Distributed and Multiagent Reinforcement Learning:
Rollout and Approximate Policy Iteration

by

Dimitri P. Bertsekas

Arizona State University
and
Massachusetts Institute of Technology

WWW site for book information and orders
http://www.athenasc.com

Athena Scientific, Belmont, Massachusetts
## Contents

1. Exact Dynamic Programming
   1.1. Deterministic Dynamic Programming
       1.1.1. Basic Finite Horizon Problem Formulation
       1.1.2. The Dynamic Programming Algorithm
       1.1.3. Approximation in Value Space
   1.2. Stochastic Dynamic Programming
       1.2.1. Finite Horizon Problems
       1.2.2. Infinite Horizon Problems - An Overview
   1.3. Examples, Variations, and Simplifications
       1.3.1. Discrete Deterministic Optimization
       1.3.2. Problems with a Termination State
       1.3.3. Partial State Information and Belief States
   1.4. Reinforcement Learning and Optimal Control - Some Terminology
   1.5. Notes and Sources

2. Rollout and Policy Improvement
   2.1. Approximation in Value and Policy Space
       2.1.1. Approximation in Value Space - One-Step and Multistep Lookahead
       2.1.2. Approximation in Policy Space
       2.1.3. Combined Approximation in Value and Policy Space
   2.2. General Issues of Approximation in Value Space
       2.2.1. Model-Based and Model-Free Implementations
       2.2.2. Off-Line and On-Line Implementations
       2.2.3. Methods for Cost-to-Go Approximation
       2.2.4. Methods for Simplification of the Lookahead Minimization
       2.2.5. Simplification of the Lookahead Minimization by Q-Factor Approximation

1.5. Notes and Sources
2.3. Rollout and the Policy Improvement Principle

2.3.1. On-Line Rollout for Deterministic Discrete Optimization
2.3.2. The Fortified Rollout Algorithm
2.3.3. Truncated Rollout Algorithms
2.3.4. Rollout with an Expert

2.4. Stochastic Rollout and Monte Carlo Tree Search

2.4.1. Simulation-Based Implementation of the Rollout Algorithm
2.4.2. Variance Reduction in Rollout - Comparing Advantages
2.4.3. Monte Carlo Tree Search
2.4.4. Randomized Policy Improvement by Monte Carlo Tree Search
2.4.5. Rollout Parallelization

2.5. On-Line Rollout for Deterministic Infinite-Spaces Problems - Optimization Heuristics

2.6. Notes and Sources

3. Specialized Rollout Algorithms

3.1. Model Predictive Control
3.1.1. Target Tubes and the Constrained Controllability Condition
3.1.2. Model Predictive Control with Terminal Cost
3.1.3. Variants of Model Predictive Control

3.2. Multiagent Rollout
3.2.1. Multiagent Coupling Through Constraints
3.2.2. Multiagent Rollout for Multiarmed Bandit Problems
3.2.3. Multiagent Model Predictive Control
3.2.4. Asynchronous Distributed Multiagent Rollout

3.3. Constrained Rollout for Deterministic Optimization
3.3.1. State-Constrained Rollout and Target Tubes
3.3.2. Rollout with Trajectory Constraints
3.3.3. Constrained Multiagent Rollout

3.4. Discrete and Combinatorial Optimization Applications
3.4.1. A General Discrete Optimization Problem
3.4.2. Multidimensional Assignment

3.5. Rollout for Minimax Control

3.6. Notes and Sources

4. Learning Values and Policies

4.1. Approximation Architectures
4.1.1. Feature-Based Architectures
4.1.2. Training of Linear and Nonlinear Architectures
## Contents

4.2. Neural Networks ................................... .............................
   4.2.1. Training of Neural Networks ............................................
   4.2.2. Multilayer and Deep Neural Networks .................................
4.3. Training of Cost Functions in Approximate DP ............................
   4.3.1. Fitted Value Iteration ...................................................
   4.3.2. Q-Factor Parametric Approximation .................................
   4.3.3. Advantage Updating ....................................................
   4.3.4. Differential Training of Rollout Policies .............................
4.4. Training of Policies in Approximate DP ...................................
   4.4.1. Perpetual Rollout with Value and Policy Networks - ..............
          Multiprocessor Parallelization ........................................
4.5. Notes and Sources ................................................................

5. Infinite Horizon: Distributed and Multiagent Algorithms

5.1. Stochastic Shortest Path and Discounted Problems ........................
   5.1.1. Policy Iteration .........................................................
   5.1.2. Optimistic and Multistep Policy Iteration ...........................
   5.1.3. Policy Iteration for Q-Factors ....................................... 792
   5.1.4. Abstract Notation for Infinite Horizon DP ..........................
5.2. Multiagent Rollout and Policy Iteration ...................................
5.3. Multiagent Value Iteration ....................................................
5.4. Approximation in Value Space - Performance Bounds ...................
   5.4.1. Limited Lookahead .....................................................
   5.4.2. Rollout, Parallel Rollout, and Multiagent Rollout ............
   5.4.3. Approximate Policy Iteration ....................................... 950
5.5. Asynchronous Distributed Value Iteration ................................
5.6. Asynchronous Distributed Policy Iteration ................................
5.7. Approximate Policy Iteration - Partitioned Architectures ...........
5.8. Approximate Policy Iteration - Partial State Observations .......... 950
   Problems ...........................................................................
5.9. Notes and Sources ................................................................

References ..............................................................................

Index ....................................................................................
Preface

In this research monograph we discuss the solution of large and challenging multistage decision problems using methods of reinforcement learning (RL for short), also referred to by other names such as approximate dynamic programming, and neuro-dynamic programming. We will focus on a subset of methods which are based on the idea of policy iteration, i.e., starting from some policy and generating one or more improved policies. If just one improved policy is generated, this is called rollout, which based on broad and consistent computational experience, appears to be the most reliable of all reinforcement learning methods. If multiple policies are generated, we obtain forms of approximate policy iteration, which we will also refer to as perpetual rollout. This is one of the most prominent type of reinforcement learning methods. It can be implemented using data generated by the system itself, a process known as self-learning, and value and policy approximation architectures, including neural networks.

An important difference between rollout and policy iteration is that the former, while less ambitious, it is not only more reliable, but it is also well-suited for on-line implementation and on-line replanning. Policy iteration is a strictly off-line method, and generally far more computationally intensive (of course rollout may also require a lot of on-line computation). This motivates the use of parallel and distributed computation. One of the purposes of this monograph is to discuss distributed (possibly asynchronous) methods that relate to rollout and policy iteration, both in the context of an exact and an approximate implementation involving neural networks. Moreover, we develop variants of rollout and policy iteration for problems with a multiagent structure, where the control consists of multiple components, each associated with a separate agent. In particular, we introduce a new approach to lookahead simplification through the use of multiagent rollout, which allows the dramatic reduction of the computational requirements for one-step lookahead when the control consists of multiple components. Multiagent rollout is also well suited for on-line autonomous decision making by multiple agents each coordinating in varying degrees with the other agents.

Several of the ideas that we develop in some depth in this monograph have been central in the implementation of recent high profile successes,
such as the AlphaGo and AlphaZero programs. In addition to the fundamental idea of successive policy iteration/improvement, they include the use of neural networks for representation of both value functions and policies, the extensive use of large scale parallelization, and the simplification of lookahead minimization, through methods involving Monte Carlo tree search and pruning of the lookahead tree. In this monograph, we also focus on policy iteration, value and policy network representations, and parallel and distributed computation.

Another subject that we deal with in some depth is model predictive control (MPC for short), one of the most prominent control system design methods at present. One of the reasons is that classical forms of MPC can be viewed as rollout algorithms, thereby providing a connection with reinforcement learning, which is beneficial in two ways. On one hand the MPC context provides rich crossfertilization opportunities with the analytical and algorithmic ideas of rollout and RL; for example the notion of sequential improvement in rollout is intimately connected to Lyapounov stability analysis in MPC, and the target tube ideas that are central in MPC may prove useful in the context of constrained rollout and policy iteration. On the other hand the dynamic programming and RL methodologies point the way to extensions of MPC based on self-learning, approximate policy iteration, simulation, the treatment of stochastic and set membership uncertainty, and the use of distributed computation.

In our development of several of the topics of this book we rely on methodology that is covered in greater depth in the 1996 neuro-dynamic programming book [BeT96] (jointly written with J. Tsitsiklis) as well as the author’s recent RL book [Ber19a]. However, we will aim to develop rollout and approximate policy iteration beyond the books [BeT96] and [Ber19a]. In particular, we will present new research, relating to distributed asynchronous computation, and systems involving multiple agents and partitioned architectures. We will also apply our methods to the solution of several types of challenging large scale optimization problems, such as combinatorial/discrete optimization and partially observed Markov decision problems (POMDP).

Dimitri P. Bertsekas
ASU, May 2020