Reinforcement Learning and Optimal Control

by

Dimitri P. Bertsekas

Massachusetts Institute of Technology

DRAFT TEXTBOOK

This is a draft of a textbook that is scheduled to be finalized in 2019, and to be published by Athena Scientific. It represents “work in progress,” and it will be periodically updated. It more than likely contains errors (hopefully not serious ones). Furthermore, its references to the literature are incomplete. Your comments and suggestions to the author at dimitrib@mit.edu are welcome. The date of last revision is given below.

January 16, 2019

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

Athena Scientific, Belmont, Massachusetts
ABOUT THE AUTHOR

Dimitri Bertsekas studied Mechanical and Electrical Engineering at the National Technical University of Athens, Greece, and obtained his Ph.D. in system science from the Massachusetts Institute of Technology. He has held faculty positions with the Engineering-Economic Systems Department, Stanford University, and the Electrical Engineering Department of the University of Illinois, Urbana. Since 1979 he has been teaching at the Electrical Engineering and Computer Science Department of the Massachusetts Institute of Technology (M.I.T.), where he is currently the McAfee Professor of Engineering.

His teaching and research have spanned several fields, including deterministic optimization, dynamic programming and stochastic control, large-scale and distributed computation, and data communication networks. He has authored or coauthored numerous research papers and seventeen books, several of which are currently used as textbooks in MIT classes, including “Dynamic Programming and Optimal Control,” “Data Networks,” “Introduction to Probability,” and “Nonlinear Programming.”

Professor Bertsekas was awarded the INFORMS 1997 Prize for Research Excellence in the Interface Between Operations Research and Computer Science for his book “Neuro-Dynamic Programming” (co-authored with John Tsitsiklis), the 2001 AACC John R. Ragazzini Education Award, the 2009 INFORMS Expository Writing Award, the 2014 AACC Richard Bellman Heritage Award, the 2014 INFORMS Khachiyan Prize for Life-Time Accomplishments in Optimization, the 2015 MOS/SIAM George B. Dantzig Prize, and the 2018 INFORMS John von Neumann Theory Prize. In 2001, he was elected to the United States National Academy of Engineering for “pioneering contributions to fundamental research, practice and education of optimization/control theory, and especially its application to data communication networks.”
ATHENA SCIENTIFIC
OPTIMIZATION AND COMPUTATION SERIES

# Contents

## 1. Exact Dynamic Programming

1.1. Deterministic Dynamic Programming .................................................. p. 2
   1.1.1. Deterministic Problems ......................................................... p. 2
   1.1.2. The Dynamic Programming Algorithm ...................................... p. 7
   1.1.3. Approximation in Value Space ................................................. p. 12
1.2. Stochastic Dynamic Programming ..................................................... p. 14
1.3. Examples, Variations, and Simplifications ........................................ p. 17
   1.3.1. Deterministic Shortest Path Problems ..................................... p. 19
   1.3.2. Discrete Deterministic Optimization ....................................... p. 21
   1.3.3. Problems with a Terminal State ............................................. p. 24
   1.3.4. Forecasts ................................................................................ p. 26
   1.3.5. Problems with Uncontrollable State Components ......................... p. 28
   1.3.6. Partial State Information and Belief States ............................... p. 33
   1.3.7. Linear Quadratic Optimal Control ............................................ p. 37
1.4. Reinforcement Learning and Optimal Control - Some Terminology ........ p. 39
1.5. Notes and Sources ............................................................................. p. 42

## 2. Approximation in Value Space

2.1. General Issues of Approximation in Value Space ................................ p. 5
   2.1.1. Methods for Computing Approximations in ................................ p. 6
   2.1.2. Off-Line and On-Line Methods ................................................ p. 7
   2.1.3. Model-Based Simplification of the Lookahead ................................ p. 8
   2.1.4. Model-Free Q-Factor Approximation in Value Space .................. p. 9
   2.1.5. Approximation in Policy Space on Top of ................................... p. 12
   2.1.6. When is Approximation in Value Space Effective? ...................... p. 13
2.2. Multistep Lookahead ........................................................................ p. 14
   2.2.1. Multistep Lookahead and Rolling Horizon ................................. p. 16
2.2. Multistep Lookahead and Deterministic Problems .......................... p. 17
2.3. Problem Approximation ......................................................... p. 19
   2.3.1. Enforced Decomposition ................................................. p. 19
   2.3.2. Probabilistic Approximation - Certainty .............................
   Equivalent Control .............................................................. p. 26
2.4. Rollout and Model Predictive Control ................................. p. 32
   2.4.1. Rollout for Deterministic Problems ................................ p. 33
   2.4.2. Stochastic Rollout and Monte Carlo Tree Search ............... p. 43
   2.4.3. Model Predictive Control .............................................. p. 51
2.5. Notes and Sources .............................................................. p. 60

3. Parametric Approximation

3.1. Approximation Architectures ............................................... p. 2
   3.1.1. Linear and Nonlinear Feature-Based Architectures ............... p. 2
   3.1.2. Training of Linear and Nonlinear Architectures ................ p. 8
   3.1.3. Incremental Gradient and Newton Methods ............................ p. 10
3.2. Neural Networks ................................................................. p. 22
   3.2.1. Training of Neural Networks ......................................... p. 25
   3.2.2. Multilayer and Deep Neural Networks ............................... p. 28
3.3. Sequential Dynamic Programming Approximation .................... p. 31
3.4. Q-factor Parametric Approximation ...................................... p. 33
3.5. Notes and Sources .............................................................. p. 35

4. Infinite Horizon Reinforcement Learning

4.1. An Overview of Infinite Horizon Problems ............................. p. 2
4.2. Stochastic Shortest Path Problems ....................................... p. 5
4.3. Discounted Problems .......................................................... p. 14
4.4. Exact and Approximate Value Iteration .................................. p. 19
4.5. Policy Iteration ................................................................. p. 22
   4.5.1. Exact Policy Iteration ................................................. p. 22
   4.5.2. Optimistic and Multistep Lookahead Policy Iteration .......... p. 26
   4.5.3. Policy Iteration for Q-factors ....................................... p. 29
4.6. Approximate Policy Iteration ............................................... p. 31
   4.6.1. Rollout ................................................................. p. 32
   4.6.2. Convergence Issues of Approximate Policy Iteration .......... p. 34
4.7. Simulation-Based Policy Iteration with Parametric ................ p. 37
   Approximation ................................................................. p. 37
   4.7.1. Self-Learning and Actor-Critic Systems ............................. p. 38
   4.7.2. A Model-Based Variant ................................................. p. 39
   4.7.3. A Model-Free Variant ................................................ p. 41
   4.7.4. Implementation Issues of Parametric Policy Iteration ........ p. 43
4.8. Exact and Approximate Linear Programming ............................ p. 46
4.9. Q-Learning ............................................................................ p. 48
4.10. Additional Methods - Temporal Differences ....................... p. 51
Contents

4.11. Approximation in Policy Space .................................................. p. 62
   4.11.1. Training by Cost Optimization - Policy Gradient and .......... p. 62
   Random Search Methods .......................................................... p. 62
   4.11.2. Expert Supervised Training ............................................. p. 68
4.12. Notes and Sources ..................................................................... p. 69
4.13. Appendix: Mathematical Analysis ............................................. p. 73
   4.13.2. Proofs for Discounted Problems ....................................... p. 79
   4.13.3. Convergence of Exact and Optimistic Policy Iteration ...... p. 79
   4.13.4. Error Bounds for Rollout and Approximate Policy Iteration .................................................. p. 82

5. Aggregation

5.1. Aggregation Frameworks ........................................................... p.
5.2. Classical and Biased Forms of the Aggregate Problem .............. p.
5.3. Bellman’s Equation for the Aggregate Problem ....................... p.
5.5. Some Examples ................................................................. p.
5.6. Spatiotemporal Aggregation for Deterministic Problems ........ p.
5.7. Notes and Sources ............................................................... p.

References ..................................................................................... p.

Index ............................................................................................. p.
Preface

In this book we consider large and challenging multistage decision problems, which can be solved in principle by dynamic programming (DP for short), but their exact solution is computationally intractable. We discuss solution methods that rely on approximations to produce suboptimal policies with adequate performance. These methods are collectively referred to as reinforcement learning, and also by alternative names such as approximate dynamic programming, and neuro-dynamic programming.

Our subject has benefited greatly from the interplay of ideas from optimal control and from artificial intelligence. One of the aims of the book is to explore the common boundary between these two fields and to form a bridge that is accessible by workers with background in either field.

Our primary focus will be on approximation in value space. Here, the control at each state is obtained by optimization of the cost over a limited horizon, plus an approximation of the optimal future cost, starting from the end of this horizon. The latter cost, which we generally denote by $\tilde{J}$, is a function of the state where we may be at the end of the horizon. It may be computed by a variety of methods, possibly involving simulation and/or some given or separately derived heuristic/suboptimal policy. The use of simulation often allows for model-free implementations that do not require the availability of a mathematical model, a major idea that has allowed the use of DP beyond its classical boundaries.

We focus selectively on four types of methods for obtaining $\tilde{J}$:

(a) Problem approximation: Here $\tilde{J}$ is the optimal cost function of a related simpler problem, which is solved by exact DP. Certainty equivalent control and enforced decomposition schemes are discussed in some detail.

(b) Rollout and model predictive control: Here $\tilde{J}$ is the cost function of some known heuristic policy. The needed cost values to implement a rollout policy are often calculated by simulation. While this method applies to stochastic problems, the reliance on simulation favors deterministic problems, including challenging combinatorial problems for which heuristics may be readily implemented. Rollout may also be combined with adaptive simulation and Monte Carlo tree search,
which have proved very effective in the context of games such as backgammon, chess, Go, and others.

Model predictive control was originally developed for continuous-space optimal control problems that involve some goal state, e.g., the origin in a classical control context. It can be viewed as a specialized rollout method that is based on an optimization algorithm for reaching a goal state.

(c) 

\textit{Parametric cost approximation:} Here $\hat{J}$ is chosen from within a parametric class of functions, including neural networks, with the parameters “optimized” or “trained” by using state-cost sample pairs and some type of incremental least squares/regression algorithm. Approximate policy iteration and its variants are covered in some detail, including several actor-critic schemes. These involve policy evaluation with temporal difference-based training methods, and policy improvement that may rely on approximation in policy space.

(d) 

\textit{Aggregation:} Here the cost function $\hat{J}$ is the optimal cost function of some approximation to the original problem, called aggregate problem, which has fewer states. The aggregate problem can be formulated in a variety of ways, and may be solved by using exact DP techniques. Its optimal cost function is then used as $\hat{J}$ in a limited horizon optimization scheme. Aggregation may also be used to provide local improvements to parametric approximation schemes that involve neural networks or linear feature-based architectures.

We have adopted a gradual expository approach, which proceeds along three directions:

(1) \textit{From exact DP to approximate DP:} We first discuss exact DP algorithms, explain why they may be difficult to implement, and then use them as the basis for approximations.

(2) \textit{From finite horizon to infinite horizon problems:} We first discuss finite horizon exact and approximate DP methodologies, which are intuitive and mathematically simple in Chapters 1-3. We then progress to infinite horizon problems in Chapters 4 and 5.

(3) \textit{From model-based to model-free approaches:} Reinforcement learning methods offer a major potential benefit over classical DP approaches, which were practiced exclusively up to the early 90s: they can be implemented by using a simulator/computer model rather than a mathematical model. In our presentation, we first discuss model-based methods, and then we identify those methods that can be appropriately modified to work with a simulator.

After the first chapter, each new class of methods is introduced as a more sophisticated or generalized version of a simpler method introduced
earlier. We often discuss methods for deterministic problems and then extend them to the stochastic setting. Moreover, we illustrate some of the methods by means of examples, which should be helpful in providing insight into their use, but may also be skipped selectively and without loss of continuity. Detailed solutions to some of the simpler examples are given, and may illustrate some of the implementation details.

The mathematical style of this book is somewhat different from the one of the author’s DP books [Ber12], [Ber17a], [Ber18a], and the 1996 neuro-dynamic programming (NDP) research monograph, written jointly with John Tsitsiklis [BeT96]. While we provide a rigorous, albeit short, account of the theory of finite and infinite horizon DP, and some fundamental approximation methods, we rely more on intuitive explanations and less on proof-based insights. Moreover, our mathematical requirements are quite modest: calculus, elementary probability, and a minimal use of matrix-vector algebra.

Several of the methods that we present are often successful in practice, but have less than solid performance properties. This is a reflection of the state of the art in the field: there are no methods that are guaranteed to work for all or even most problems, but there are enough methods to try on a given problem with a reasonable chance of success in the end. For this process to work, however, it is important to have proper intuition into the inner workings of each type of method, as well as an understanding of its analytical and computational properties. To quote a statement from the preface of the NDP monograph [BeT96]: “It is primarily through an understanding of the mathematical structure of the NDP methodology that we will be able to identify promising or solid algorithms from the bewildering array of speculative proposals and claims that can be found in the literature.”

Another statement from a recent NY Times article [Str18], in connection with DeepMind’s remarkable AlphaZero chess program, is also worth quoting: “What is frustrating about machine learning, however, is that the algorithms can’t articulate what they’re thinking. We don’t know why they work, so we don’t know if they can be trusted. AlphaZero gives every appearance of having discovered some important principles about chess, but it can’t share that understanding with us. Not yet, at least. As human beings, we want more than answers. We want insight. This is going to be a source of tension in our interactions with computers from now on.” To this we may add that human insight can only develop within some structure of human thought, and it appears that mathematical reasoning with algorithmic models is the most suitable structure for this purpose.

Dimitri P. Bertsekas

Winter 2018