

*Reinforcement Learning
and
Optimal Control*

by

Dimitri P. Bertsekas

Massachusetts Institute of Technology

WWW site for book information and orders

<http://www.athenasc.com>



Athena Scientific, Belmont, Massachusetts

Athena Scientific
Post Office Box 805
Nashua, NH 03060
U.S.A.

Email: info@athenasc.com
WWW: <http://www.athenasc.com>

Cover photography: Dimitri Bertsekas

© 2019 Dimitri P. Bertsekas
All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

Publisher's Cataloging-in-Publication Data

Bertsekas, Dimitri P.
Reinforcement Learning and Optimal Control
Includes Bibliography and Index
1. Mathematical Optimization. 2. Dynamic Programming. I. Title.
QA402.5 .B465 2019 519.703 00-91281

ISBN-10: 1-886529-39-6, ISBN-13: 978-1-886529-39-7

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 McAfee Professor of Engineering. Starting in August 2019, he will also be Fulton Professor of Computational Decision Making at the Arizona State University, Tempe, AZ.

Professor Bertsekas' 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," "Convex Optimization Algorithms," 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 Lifetime 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

1. Reinforcement Learning and Optimal Control, by Dimitri P. Bertsekas, 2019, ISBN 978-1-886529-39-7, 388 pages
2. Abstract Dynamic Programming, 2nd Edition, by Dimitri P. Bertsekas, 2018, ISBN 978-1-886529-46-5, 360 pages
3. Dynamic Programming and Optimal Control, Two-Volume Set, by Dimitri P. Bertsekas, 2017, ISBN 1-886529-08-6, 1270 pages
4. Nonlinear Programming, 3rd Edition, by Dimitri P. Bertsekas, 2016, ISBN 1-886529-05-1, 880 pages
5. Convex Optimization Algorithms, by Dimitri P. Bertsekas, 2015, ISBN 978-1-886529-28-1, 576 pages
6. Convex Optimization Theory, by Dimitri P. Bertsekas, 2009, ISBN 978-1-886529-31-1, 256 pages
7. Introduction to Probability, 2nd Edition, by Dimitri P. Bertsekas and John N. Tsitsiklis, 2008, ISBN 978-1-886529-23-6, 544 pages
8. Convex Analysis and Optimization, by Dimitri P. Bertsekas, Angelia Nedić, and Asuman E. Ozdaglar, 2003, ISBN 1-886529-45-0, 560 pages
9. Network Optimization: Continuous and Discrete Models, by Dimitri P. Bertsekas, 1998, ISBN 1-886529-02-7, 608 pages
10. Network Flows and Monotropic Optimization, by R. Tyrrell Rockafellar, 1998, ISBN 1-886529-06-X, 634 pages
11. Introduction to Linear Optimization, by Dimitris Bertsimas and John N. Tsitsiklis, 1997, ISBN 1-886529-19-1, 608 pages
12. Parallel and Distributed Computation: Numerical Methods, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1997, ISBN 1-886529-01-9, 718 pages
13. Neuro-Dynamic Programming, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1996, ISBN 1-886529-10-8, 512 pages
14. Constrained Optimization and Lagrange Multiplier Methods, by Dimitri P. Bertsekas, 1996, ISBN 1-886529-04-3, 410 pages
15. Stochastic Optimal Control: The Discrete-Time Case, by Dimitri P. Bertsekas and Steven E. Shreve, 1996, ISBN 1-886529-03-5, 330 pages

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. 18
1.3.1. Deterministic Shortest Path Problems	p. 19
1.3.2. Discrete Deterministic Optimization	p. 21
1.3.3. Problems with a Termination State	p. 25
1.3.4. Forecasts	p. 26
1.3.5. Problems with Uncontrollable State Components	p. 29
1.3.6. Partial State Information and Belief States	p. 34
1.3.7. Linear Quadratic Optimal Control	p. 38
1.3.8. Systems with Unknown Parameters - Adaptive Control	p. 40
1.4. Reinforcement Learning and Optimal Control - Some Terminology	p. 43
1.5. Notes and Sources	p. 45
2. Approximation in Value Space	
2.1. Approximation Approaches in Reinforcement Learning	p. 50
2.1.1. General Issues of Approximation in Value Space	p. 54
2.1.2. Off-Line and On-Line Methods	p. 56
2.1.3. Model-Based Simplification of the Lookahead Minimization	p. 57
2.1.4. Model-Free Q-Factor Approximation in Value Space	p. 58
2.1.5. Approximation in Policy Space on Top of Approximation in Value Space	p. 61
2.1.6. When is Approximation in Value Space Effective?	p. 62
2.2. Multistep Lookahead	p. 64

2.2.1. Multistep Lookahead and Rolling Horizon	p. 65
2.2.2. Multistep Lookahead and Deterministic Problems	p. 67
2.3. Problem Approximation	p. 69
2.3.1. Enforced Decomposition	p. 69
2.3.2. Probabilistic Approximation - Certainty Equivalent Control	p. 76
2.4. Rollout	p. 83
2.4.1. On-Line Rollout for Deterministic Discrete Optimization	p. 84
2.4.2. Stochastic Rollout and Monte Carlo Tree Search	p. 95
2.4.3. Rollout with an Expert	p. 104
2.5. On-Line Rollout for Deterministic Infinite-Spaces Problems - Optimization Heuristics	p. 106
2.5.1. Model Predictive Control	p. 108
2.5.2. Target Tubes and the Constrained Controllability Condition	p. 115
2.5.3. Variants of Model Predictive Control	p. 118
2.6. Notes and Sources	p. 120
 3. Parametric Approximation	
3.1. Approximation Architectures	p. 126
3.1.1. Linear and Nonlinear Feature-Based Architectures	p. 126
3.1.2. Training of Linear and Nonlinear Architectures	p. 134
3.1.3. Incremental Gradient and Newton Methods	p. 135
3.2. Neural Networks	p. 149
3.2.1. Training of Neural Networks	p. 153
3.2.2. Multilayer and Deep Neural Networks	p. 157
3.3. Sequential Dynamic Programming Approximation	p. 161
3.4. Q-Factor Parametric Approximation	p. 162
3.5. Parametric Approximation in Policy Space by Classification	p. 165
3.6. Notes and Sources	p. 171
 4. Infinite Horizon Dynamic Programming	
4.1. An Overview of Infinite Horizon Problems	p. 174
4.2. Stochastic Shortest Path Problems	p. 177
4.3. Discounted Problems	p. 187
4.4. Semi-Markov Discounted Problems	p. 192
4.5. Asynchronous Distributed Value Iteration	p. 197
4.6. Policy Iteration	p. 200
4.6.1. Exact Policy Iteration	p. 200
4.6.2. Optimistic and Multistep Lookahead Policy Iteration	p. 205
4.6.3. Policy Iteration for Q-factors	p. 208

4.7. Notes and Sources p. 209

4.8. Appendix: Mathematical Analysis p. 211

 4.8.1. Proofs for Stochastic Shortest Path Problems p. 212

 4.8.2. Proofs for Discounted Problems p. 217

 4.8.3. Convergence of Exact and Optimistic

 Policy Iteration p. 218

5. Infinite Horizon Reinforcement Learning

5.1. Approximation in Value Space - Performance Bounds p. 222

 5.1.1. Limited Lookahead p. 224

 5.1.2. Rollout p. 227

 5.1.3. Approximate Policy Iteration p. 232

5.2. Fitted Value Iteration p. 235

5.3. Simulation-Based Policy Iteration with Parametric

 Approximation p. 239

 5.3.1. Self-Learning and Actor-Critic Methods p. 239

 5.3.2. A Model-Based Variant p. 241

 5.3.3. A Model-Free Variant p. 243

 5.3.4. Implementation Issues of Parametric Policy

 Iteration p. 246

 5.3.5. Convergence Issues of Parametric Policy Iteration -

 Oscillations p. 249

5.4. Q-Learning p. 253

 5.4.1. Optimistic Policy Iteration with Parametric Q-Factor

 Approximation - SARSA and DQN p. 255

5.5. Additional Methods - Temporal Differences p. 256

5.6. Exact and Approximate Linear Programming p. 267

5.7. Approximation in Policy Space p. 270

 5.7.1. Training by Cost Optimization - Policy Gradient,

 Cross-Entropy, and Random Search Methods p. 276

 5.7.2. Expert-Based Supervised Learning p. 286

 5.7.3. Approximate Policy Iteration, Rollout, and

 Approximation in Policy Space p. 288

5.8. Notes and Sources p. 293

5.9. Appendix: Mathematical Analysis p. 298

 5.9.1. Performance Bounds for Multistep Lookahead p. 299

 5.9.2. Performance Bounds for Rollout p. 301

 5.9.3. Performance Bounds for Approximate Policy

 Iteration p. 304

6. Aggregation

6.1. Aggregation with Representative States p. 308

 6.1.1. Continuous Control Space Discretization p. 314

 6.1.2. Continuous State Space - POMDP Discretization p. 315

6.2. Aggregation with Representative Features	p. 317
6.2.1. Hard Aggregation and Error Bounds	p. 320
6.2.2. Aggregation Using Features	p. 322
6.3. Methods for Solving the Aggregate Problem	p. 328
6.3.1. Simulation-Based Policy Iteration	p. 328
6.3.2. Simulation-Based Value Iteration and Q-Learning	p. 331
6.4. Feature-Based Aggregation with a Neural Network	p. 332
6.5. Biased Aggregation	p. 334
6.6. Notes and Sources	p. 337
6.7. Appendix: Mathematical Analysis	p. 340
References	p. 345
Index	p. 369

Preface

Turning to the succor of modern computing machines, let us renounce all analytic tools.

Richard Bellman [Bel57]

From a teleological point of view the particular numerical solution of any particular set of equations is of far less importance than the understanding of the nature of the solution.

Richard Bellman [Bel57]

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 known by several essentially equivalent names: *reinforcement learning*, *approximate dynamic programming*, and *neuro-dynamic programming*. We will use primarily the most popular name: reinforcement learning.

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. Another aim is to organize coherently the broad mosaic of methods that have proved successful in practice while having a solid theoretical and/or logical foundation. This may help researchers and practitioners to find their way through the maze of competing ideas that constitute the current state of the art.

There are two general approaches for DP-based suboptimal control. The first is *approximation in value space*, where we approximate in some way the optimal cost-to-go function with some other function. The major alternative to approximation in value space is *approximation in policy space*, whereby we select the policy by using optimization over a suitably restricted class of policies, usually a parametric family of some form. In some schemes these two types of approximation may be combined, aiming to capitalize on the advantages of both. Generally, approximation in value

space is tied more closely to the central DP ideas of value and policy iteration than approximation in policy space, which relies on gradient-like descent, a more broadly applicable optimization mechanism.

While we provide a substantial treatment of approximation in policy space, most of the book is focused 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. The latter cost, which we generally denote by \tilde{J} , is a function of the state where we may be. 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 implementations that do not require a mathematical model, a major idea that has allowed the use of DP beyond its classical boundaries.

We discuss selectively 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 a suboptimal optimization for reaching a goal state.

- (c) *Parametric cost approximation*: Here \tilde{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 and critic schemes. These involve policy evaluation with simulation-based training methods, and policy improvement that may rely on approximation in policy space.
- (d) *Aggregation*: Here \tilde{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 \tilde{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 four directions:

- (1) *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) *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 4-6.
- (3) *From deterministic to stochastic models:* We often discuss separately deterministic and stochastic problems. The reason is that deterministic problems are simpler and offer special advantages for some of our methods.
- (4) *From model-based to model-free implementations:* 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 implementations, and then we identify schemes 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. 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.

The mathematical style of this book is somewhat different from the one of the author's DP books [Ber12], [Ber17], [Ber18a], and the 1996 neuro-dynamic programming (NDP) research monograph, written jointly with John Tsitsiklis [BeT96]. While we provide a rigorous, albeit short, mathematical 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, a minimal use of matrix-vector algebra, and elementary probability (mathematically complicated arguments involving laws of large numbers and stochastic convergence are bypassed in favor of intuitive explanations).

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. To aid in this process, we place primary emphasis on developing intuition into the inner workings of each type of method. Still, however, it is important to have a foundational understanding of the analytical principles of the field and of the mechanisms underlying the central computational methods. 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.

I would like to express my appreciation to the many students and colleagues that contributed directly or indirectly to the book. A special thanks is due to my principal collaborators on the subject, over the last 25 years, particularly John Tsitsiklis, Janey (Huizhen) Yu, and Mengdi Wang.

[†] The two 1957 Bellman quotations at the beginning of this preface also express this tension, although the first of these, while striking and widely cited, is admittedly taken a little out of context (throughout his work on practical applications, Bellman remained a mathematical analyst at heart). Bellman’s fascinating autobiography [Bel84] contains a lot of information on the origins of DP (and approximate DP as well!); selected quotations from this autobiography have been compiled by his collaborator Dreyfus [Dre02]. Among others, Bellman states that “In order to make any progress, it is necessary to think of approximate techniques, and above all, of numerical algorithms. Finally, having devoted a great deal of time and effort, mostly fruitless, to the analysis of many varieties of simple models, I was prepared to face up to the challenge of using dynamic programming as an effective tool for obtaining numerical answers to numerical questions.” He goes on to attribute his motivation to work on numerical DP to the emergence of the (then primitive) digital computer, which he calls “Sorcerer’s Apprentice.”

Moreover, sharing insights with Ben Van Roy over the years has been important in shaping my thinking. Interactions with Ben Recht regarding policy gradient methods were also very helpful. The projects that my students worked on as part of DP courses I taught at MIT inspired many ideas that indirectly found their way into the book. I want to express my thanks to the many readers, who proofread parts of the book. In this respect I would like to single out Yuchao Li who made many helpful comments, and Thomas Stahlbuhk, who went through the entire book with great care, and offered numerous insightful suggestions.

The book took shape while teaching a course on the subject at the Arizona State University (ASU) during a two-month period starting in January 2019. Videlectures and slides from this class are available from my website

<http://web.mit.edu/dimitrib/www/RLbook.html>

and provide a good supplement and companion resource to the book. The hospitable and stimulating environment at ASU contributed much to my productivity during this period, and for this I am very thankful to Stephanie Gil, as well as other colleagues from ASU, including Heni Ben Amor, Esmā Gel, Subbarao (Rao) Kambhampati, Angelia Nedic, Giulia Pedrielli, Jennie Si, and Petr Sulc. Moreover, Stephanie together with her students, Sushmita Bhattacharya and Thomas Wheeler, collaborated with me on research and implementation of various methods, contributed many insights, and tested out several variations.

Dimitri P. Bertsekas

June 2019