Reinforcement Learning and Optimal Control

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

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

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

Ebook version (includes minor revisions)

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. In 1979 he joined the Electrical Engineering and Computer Science Department of the Massachusetts Institute of Technology (M.I.T.), where he retired in 2016 as McAfee Professor of Engineering. In 2019, he became Fulton Professor of Computational Decision Making in the School of Computing, Informatics, and Decision Systems Engineering 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 eighteen books, several of which are currently used as textbooks in MIT and ASU classes, including "Dynamic Programming and Optimal Control," "Data Networks," "Introduction to Probability," "Convex Optimization Algorithms," "Nonlinear Programming," and the present book.

Professor Bertsekas was awarded the INFORMS 1997 Prize for Research Excellence in the Interface Between Operations Research and Computer Science, 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 2022 IEEE Control Systems Award. In 2018, he was awarded, jointly with his coauthor John Tsitsiklis, the INFORMS John von Neumann Theory Prize, for the contributions of the research monographs "Parallel and Distributed Computation" and "Neuro-Dynamic Programming." 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. Rollout, Policy Iteration, and Distributed Reinforcement Learning, by Dimitri P. Bertsekas, 2020, ISBN 978-1-886529-07-6, 480 pages
2. Reinforcement Learning and Optimal Control, by Dimitri P. Bertsekas, 2019, ISBN 978-1-886529-39-7, 388 pages
3. Abstract Dynamic Programming, 2nd Edition, by Dimitri P. Bertsekas, 2018, ISBN 978-1-886529-46-5, 360 pages
4. Dynamic Programming and Optimal Control, Two-Volume Set, by Dimitri P. Bertsekas, 2017, ISBN 1-886529-08-6, 1270 pages
5. Nonlinear Programming, 3rd Edition, by Dimitri P. Bertsekas, 2016, ISBN 1-886529-05-1, 880 pages
6. Convex Optimization Algorithms, by Dimitri P. Bertsekas, 2015, ISBN 978-1-886529-28-1, 576 pages
7. Convex Optimization Theory, by Dimitri P. Bertsekas, 2009, ISBN 978-1-886529-31-1, 256 pages
8. Introduction to Probability, 2nd Edition, by Dimitri P. Bertsekas and John N. Tsitsiklis, 2008, ISBN 978-1-886529-23-6, 544 pages
9. Convex Analysis and Optimization, by Dimitri P. Bertsekas, Angelia Nedić, and Asuman E. Ozdaglar, 2003, ISBN 1-886529-45-0, 560 pages
10. Network Optimization: Continuous and Discrete Models, by Dimitri P. Bertsekas, 1998, ISBN 1-886529-02-7, 608 pages
11. Network Flows and Monotropic Optimization, by R. Tyrrell Rockafellar, 1998, ISBN 1-886529-06-X, 634 pages
12. Introduction to Linear Optimization, by Dimitris Bertsimas and John N. Tsitsiklis, 1997, ISBN 1-886529-19-1, 608 pages
13. Parallel and Distributed Computation: Numerical Methods, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1997, ISBN 1-886529-01-9, 718 pages
14. Neuro-Dynamic Programming, by Dimitri P. Bertsekas and John N. Tsitsiklis, 1996, ISBN 1-886529-10-8, 512 pages
15. Constrained Optimization and Lagrange Multiplier Methods, by Dimitri P. Bertsekas, 1996, ISBN 1-886529-04-3, 410 pages
16. Stochastic Optimal Control: The Discrete-Time Case, by Dimitri P. Bertsekas and Steven E. Shreve, 1996, ISBN 1-886529-03-5, 330 pages

Contents

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 improve-

ment 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 al-

gebra, and elementary probability (mathematically complicated arguments involving laws of large numbers and stochastic convergence are bypassed in favor of intuitive explanations).

Still in our use of a more intuitive but less proof-oriented expository style, we have followed a few basic principles. The most important of these is to maintain rigor in the use of natural language. The reason is that with fewer mathematical arguments and proofs, precise language is essential to maintain a logically consistent exposition. In particular, we have aimed to define terms unambiguously, and to avoid using multiple terms with essentially identical meaning. Moreover, when circumstances permitted, we have tried to provide enough explanation/intuition so that a mathematician can find the development believable and even construct the missing rigorous proofs.

We note that 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

† While reinforcement learning rests on the mathematical principles of DP, it also relies on multiple interacting approximations whose effects are hard to predict and quantify in practice. It may be hoped that with further theoretical and applications research, the state of the subject will improve and clarify. However, it can be said that in its current form, reinforcement learning is an exploding field, which is complicated, unclean, and somewhat confusing (something that the front cover image of the book also tries to convey). Reinforcement learning is not unique in this. One may think of other important optimization areas where a similar state of the art has prevailed for a long time.

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. 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. Videolectures 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, Esma Gel, Subbarao (Rao) Kambhampati, Angelia Nedic, Giulia Pedrielli, Jennie Si, and Petr Sulc. Moreover, Stephanie together with her

† 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."

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

NOTE ABOUT THE EBOOK EDITION

This 2021 ebook edition contains minor editorial changes to the 2019 original version, which were prompted by the publication of my companion research monograph

> Rollout, Policy Iteration, and Distributed Reinforcement Learning, Athena Scientific, 2020, ISBN 978-1-886529-07-6

This latter monograph focuses more closely on several topics related to rollout, approximate policy iteration, multiagent problems, discrete and Bayesian optimization, and distributed computation, which are either discussed in less detail or not covered at all in the present book. Moreover, the monograph's development follows somewhat more narrowly the AlphaZero and TD-Gammon paradigms, and their off-line training/on-line play algorithmic structure, which is the key to their success.

On the other hand, the present book provides a more comprehensive coverage of reinforcement learning, and includes the development of topics that are not covered at all in the 2020 book, such as approximation in policy space, aggregation, and temporal difference methods. It also contains, in Chapters 4 and 5, a proof-based development of some of the infinite horizon exact and approximate DP theory.

My website

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

contains class notes, and a series of videolectures and slides from my 2021 course, which address a selection of topics from both books. The videolectures are also available at

> https://www.youtube.com/playlist?list
> =PLmH30BG15SIp79JRJ-MVF12uvB1qPtPzn

and at

> https://space.bilibili.com/2036999141

Dimitri P. Bertsekas

July 2021

Neuro-Dynamic Programming
Dimitri P. Bertsekas and John N. Tsitsiklis

Athena Scientific, 1996, 512 pp., hardcover, ISBN 1-886529-10-8

This is the first textbook that fully explains the neuro-dynamic programming/reinforcement learning methodology, which is a recent breakthrough in the practical application of neural networks and dynamic programming to complex problems of planning, optimal decision making, and intelligent control.

From the review by George Cybenko for IEEE Computational Science and Engineering, May 1998:

"Neuro-Dynamic Programming is a remarkable monograph that integrates a sweeping mathematical and computational landscape into a coherent body of rigorous knowledge. The topics are current, the writing is clear and to the point, the examples are comprehensive and the historical notes and comments are scholarly."

"In this monograph, Bertsekas and Tsitsiklis have performed a Herculean task that will be studied and appreciated by generations to come. I strongly recommend it to scientists and engineers eager to seriously understand the mathematics and computations behind modern behavioral machine learning."

Among its special features, the book:

- Describes and unifies a large number of NDP methods, including several that are new
- Describes new approaches to formulation and solution of important problems in stochastic optimal control, sequential decision making, and discrete optimization
- Rigorously explains the mathematical principles behind NDP
- Illustrates through examples and case studies the practical application of NDP to complex problems from optimal resource allocation, optimal feedback control, data communications, game playing, and combinatorial optimization
- Presents extensive background and new research material on dynamic programming and neural network training

Neuro-Dynamic Programming is the winner of the 1997 INFORMS CSTS prize for research excellence in the interface between Operations Research and Computer Science