

# Markov Decision Processes and Reinforcement Learning

Martin L. Puterman and Timothy C. Y. Chan

August 7, 2025

*This material will be published by Cambridge University Press as “Markov Decision Processes and Reinforcement Learning” by Martin L. Puterman and Timothy C. Y. Chan. This pre-publication version is free to view and download for personal use only. Not for re-distribution, re-sale, or use in derivative works. ©Martin L. Puterman and Timothy C. Y. Chan, 2025.*

# Dedication

*Marty: To my wife, Dodie Katzenstein, who made helpful suggestions and encouraged me to write this book, despite claiming she has no idea what it is about.*

*Tim: To my family – mom, dad, Laura, Euclid, Escher – who support me in everything I do, and can now stop asking when the book will be done.*

# Preface

When we set out to write this book, at the onset of the COVID-19 pandemic, our goal was to provide an accessible and logically structured introduction to Markov decision process (MDP) theory, applications and algorithms. As our writing and research progressed, we came to appreciate the vast and rapidly evolving landscape of reinforcement learning as a major sub-field within artificial intelligence (AI)<sup>1</sup>. Consequently, a significant portion of this book now focuses on reinforcement learning methods. The advantage of grounding our treatment in Markov decision processes is that they offer a rigorous and unifying modeling framework, one that applies even in *model-free* reinforcement learning settings in which system dynamics are not fully known.

This book is not meant to be encyclopedic. Rather, we have chosen to focus on the core concepts of these disciplines, with the objective of providing a solid basis for further study, application, and research. Throughout, simple and transparent models illustrate how the fundamental Markov decision process entities – states, actions, transition probabilities and rewards – apply in each context. A sound grasp of these fundamentals is necessary for use in more complex settings. In our experiences reviewing journal submissions, we have observed that model formulations are frequently imprecise, resulting in ambiguity in explaining and interpreting results.

Numerous computational examples in this book illustrate how algorithms work in practice. Commentary accompanying algorithms and in computational examples provides insights into algorithmic details and guidelines for application. Hands-on coding and tuning is a necessary step in learning to use these methods, an aspect that is often missed when learners rely solely on reusing pre-written code found online.

Markov decision processes arise in many fields, most notably operations research, robotics, computer science, engineering, economics, and management. While our approach primarily reflects an operations research perspective, the concepts and tools presented here are broadly applicable. We believe this book will provide a solid foundation for anyone seeking a rigorous introduction to these topics, especially within the growing reinforcement learning community.

A word of caution: the extremely powerful methods in this book are broadly applicable yet have the potential to be misused. We encourage you to apply them responsibly. In the spirit of Google’s original guiding principle, “Don’t be evil.”

---

<sup>1</sup>Coincidentally, ChatGPT, which leveraged reinforcement learning in its development, has assisted us in editing text, generating code, and producing figures.

## Book objectives

The book aims to provide readers with a solid foundation in Markov decision process principles and algorithms, reinforcement learning methods, and the relationship between the two. Readers will learn to apply Markov decision processes, identify model components, solve models using exact or approximate methods, and interpret solutions.

We strongly believe that working through examples and coding algorithms is the best way to learn this material. Accordingly, through numerous examples, the book includes extensive numerical calculations and discussions of their implications. Additional exercises are provided at the end of each chapter to further illustrate applications and reinforce key concepts.

## Book structure

The book is organized as follows:

- **Introduction:** Chapter 1 describes the foundations and applications of Markov decision processes and reinforcement learning. In addition it provides a historical perspective on the evolution of Markov decision processes in operations research and control theory, and reinforcement learning in behavioural science and artificial intelligence, setting the stage for the integration of these perspectives in contemporary research and applications.
- **Part I - Fundamentals:** Chapters 2-3 develop and apply a rigorous modeling foundation. Chapter 2 introduces basic model components and optimality criteria, illustrates elementary calculations that form the building blocks of more complex algorithms, and includes a numerical example that appears throughout the book to illustrate new ideas and methods. A diverse set of applications in Chapter 3 provides the reader with real-world examples of Markov decision process formulations.
- **Part II - Classical Markov decision process models:** Chapter 4 covers finite horizon MDPs, while Chapters 5-7 address infinite horizon MDPs under several optimality criteria. Chapter 5 focuses on discounted models, which are most widely used and theoretically complete. Chapter 6 characterizes and analyses expected total reward models, also known as episodic models, which are foundational in reinforcement learning. Chapter 7 examines models under the long-run average reward criteria. Finite and infinite horizon partially observable Markov decision processes (POMDPs) are the subject of Chapter 8.
- **Part III - Reinforcement learning:** Chapters 9-11 describe methods for solving large-scale, model-based and model-free sequential decision processes. Chapter 9 combines value function approximation with value iteration, policy iteration and linear programming to find approximate solutions for Markov decision process. This material is often referred to as *approximate dynamic programming*.

Simulation-based methods that apply in both a model-free and model-based environment are the subject of Chapter 10. This chapter considers *tabular models* which are of a size that allows value functions, state-action value functions, and policies to be represented explicitly in look-up tables. Chapter 11 extends these results to large-scale models by combining the approximation methods of Chapters 9 and 10.

- **Appendices:** Book appendices summarize key mathematical notation and conventions used throughout the book, as well as background on Markov chains and linear programming. Some chapters have their own appendices, which provide supplementary technical details and relevant background. Topics include stochastic approximation, regression, and gradient descent.
- **Bibliographic remarks:** Each chapter includes a brief section that provides historical commentary and references.
- **Exercises:** Exercises at the end of each chapter offer opportunities to formulate models, apply methods, or delve into theoretical details. Many are open-ended. Alternatively, readers are encouraged to choose their own models to formulate and try out the methods in the book, as we have done with the recurring two-state model, queuing control model and coffee-delivering robot model (featured on the book cover). Starred exercises indicate increased difficulty.
- **Starred sections:** Sections with an asterisk (\*) contain important, technically demanding or supplementary material. They may be skipped on first reading.
- **Gray boxes:** In addition to theoretical results, algorithms, and examples, important equations and comments are placed in gray boxes throughout the book.

## Relationship to Puterman [1994]

This book is not a revision of Puterman [1994] which was written to be a comprehensive and state-of-the-art reference targeted at researchers and advanced graduate students with strong mathematics backgrounds. Our goal in writing this book is to provide a rigorous yet more accessible introduction to Markov decision processes, reinforcement learning and their extensions that will be accessible to advanced undergraduate students, early graduate students, and practitioners. Since the field has rapidly expanded since the early 1990s, especially in the area of reinforcement learning, this book develops these concepts using the common notation and language of Markov decision processes.

Some specific differences with Puterman [1994] include:

- A broad discussion of reinforcement learning methods, many of which were not widely known, especially in the operations research community, in the early 1990s.

- Formulation and analysis of partially observable Markov decision processes (POMDPs).
- Descriptions of state-action value functions permeate the book in anticipation of their fundamental role in reinforcement learning methods.
- Increased emphasis on randomized policies in light of their fundamental role in linear programming, policy gradient and actor-critic methods.
- Illustrating all algorithms with numerical calculations applied to common examples. All computations were done from scratch using R [\[R Core Team, 2024\]](#). With the rapidly advancing proficiency of AI models to generate useful code and translate code between languages, readers are encouraged to engage hands-on in the coding process (perhaps with AI assistance) in order to truly understand the workings of each algorithm.
- Omission of material on countable state models, multi-chain average reward MDPs and sensitive optimality criteria including Blackwell optimality.
- Limited bibliographic references. With the rapid development of internet resources as well as the presence of some excellent books, historical perspectives are readily available elsewhere.
- Representing component-wise maxima by “c-max” to avoid misinterpretation of “max” when using vector notation.
- Using  $r(s, a, j)$  as the basic definition of the reward function instead of  $r(s, a)$ .
- Emphasizing transient and stochastic shortest path models in the chapter on infinite horizon models with the expected total reward criterion. The advantage of this categorization is that it applies easily to what have become known as *episodic* models. The earlier book adopted a classic approach based on positive and negative models.

## Neural networks

The book intentionally omits the extensive and rapidly evolving body of research and application of *deep reinforcement learning*, a subfield of reinforcement learning that leverages neural network approximations. While neural networks have played a central role in many breakthroughs in reinforcement learning, we have chosen not to cover neural network design or training. Readers interested in learning about and applying neural networks are encouraged to consult dedicated texts on deep learning that describe network architectures, backpropagation, and training strategies.

Our focus is on foundational principles and the core Markov decision process and reinforcement learning theory and algorithms. Understanding these foundations is essential regardless of whether function approximation is through tabular representations, linear functions of features, or deep neural networks. For this reason, we believe

a mastery of the basics should precede the study of more complex approximation techniques.

## How to use this book

Our intention in writing this book is that readers find the material self-explanatory and accessible so that independent study is possible. The following description of course structures might provide guidance in working through it. The requisite background includes a solid foundation in probability, linear algebra, real analysis and statistical estimation.

Every course should at a minimum cover Chapters [1-4](#). Subsequent chapters should be chosen to reflect course and learning objectives.

- An introductory Markov decision process course with an operations research orientation should supplement the above chapters with selected material on optimality equations and algorithms from Chapter [5](#) on infinite horizon discounted models and Chapter [6](#) on episodic models. The model formulation in Chapter [8](#) of partially observable MDPs and the material on simulation of episodic and infinite horizon discounted models in Chapter [10](#) nicely complements the above material.
- A more advanced MDP course should delve more deeply into theoretical issues in Chapters [5](#) and [6](#), and also include material on average reward models in Chapter [7](#), supplemented by material in Chapter 9 of [Puterman 1994](#).
- A reinforcement learning course with a computer science orientation should supplement the material in Chapters [1-4](#) with methods chosen from Chapters [9-11](#). Such a course should delve more deeply into the topics in Section [11.7](#), especially neural networks. Instructors should add material on neural network approximations of value functions, state-action value functions and policies, as well as focusing on examples arising in robotics, vehicle guidance and generative models.

## Additional resources

Additional resources and errata can be found at <https://www.cambridge.org/9781009098410>.

## Our cover image

Our life-long friend, Flora Gordon, designed our whimsical and aptly-themed cover with assistance from her colleague Sergio Lopez. The robot, who we refer to as “Flow”, appears in the Gridworld model of Section [3.2](#), an application that recurs throughout the book to illustrate Markov decision process and reinforcement learning algorithms. This example is inspired by the Hungarian mathematician Alfréd Rényi who is attributed with the quote:



*A mathematician is a machine for turning coffee into theorems.*

Moreover, robotics has now become one of the principal application areas of reinforcement learning.

## Acknowledgments

We would like to acknowledge many individuals who assisted us on this journey. Significant contributors include Abhijit Gosavi and Nicolas Gast, whose insights helped shape Chapters 10 and 11 (AG) and Chapter 7 (NG). Special thanks go to Antoine Sauré who provided extensive and insightful comments on Chapters 9-11.

We are also grateful to those who have provided suggestions and guidance including Alan Mackworth, Rich Sutton, Reid Swanson, Steven Shechter, Dmitri Bertsekas, John Tsitsiklis, Sergey Levine, Bruno Scherrer and Gergely Neu. Others have offered minor editorial improvements and suggestions, and checked calculations.

Elise Liu, while still in high school and now at The University of St. Andrews, produced many of the book's figures. Discussions with Pritam Dash provided us with insight into a computer science graduate student's perspective on this material. Moreover, he developed a Github repository for code and text. Neal Kaw and Michael Gimelfarb developed exercise solutions and several figures for the early chapters. Ken Wong also contributed to the figures, as well as the bibliography.

In addition, we thank Lauren Cowles and Arman Chowdhury for their timely encouragement, attention to detail, and expert advice on manuscript preparation, and Diana Gillooly who was our initial contact with Cambridge University Press.

# Contents

<b>1</b>	<b>Introduction</b>	<b>19</b>
1.1	What is a Markov decision process?	21
1.1.1	A historical perspective	22
1.1.2	Markov decision process applications	23
1.2	What is reinforcement learning?	25
1.2.1	Trial-and-error learning	25
1.2.2	A historical perspective	26
1.3	Book structure	31
<b>2</b>	<b>Markov Decision Process Fundamentals</b>	<b>35</b>
2.1	Basic model components	36
2.1.1	Planning horizons and decision epochs	36
2.1.2	States	38
2.1.3	Actions	38
2.1.4	Transition probabilities	39
2.1.5	Rewards	39
2.1.6	Markov decision processes and decision trees	40
2.2	Derived objects	41
2.2.1	Decision rules	42
2.2.2	Policies	44
2.2.3	Derived stochastic processes	45
2.2.4	Reward processes	47
2.2.5	Assigning a value to a reward process	48
2.3	Optimality criteria: Transforming a Markov decision process into a Markov decision problem	50
2.3.1	Expected total reward	51
2.3.2	Expected discounted reward	56
2.3.3	Long-run average reward for an infinite horizon model	61
2.4	The one-period problem: A fundamental building block	63
2.4.1	Optimal value functions	64
2.4.2	Optimal policies	66
2.4.3	State-action value functions	68
2.4.4	Summary of results for a one-period problem	69

CONTENTS	10
2.5 A two-state model . . . . .	70
2.5.1 A one-period version . . . . .	71
2.5.2 Policies in a two-period version of the two-state model . . . . .	72
<b>3 Examples and Applications</b>	<b>80</b>
3.1 Inventory management . . . . .	80
3.1.1 Inventory management with backlogged demand . . . . .	82
3.1.2 The newsvendor problem . . . . .	85
3.2 Gridworld navigation . . . . .	88
3.3 Revenue management: Using price to manage demand . . . . .	92
3.4 Discrete-time queuing models . . . . .	94
3.4.1 Service rate control . . . . .	95
3.4.2 Admission control . . . . .	97
3.5 Behavioral decision making: When should a lion hunt? . . . . .	103
3.6 Clinical decision making: An application to liver transplantation . . . . .	105
3.7 Advance appointment scheduling . . . . .	108
3.8 Optimal stopping . . . . .	113
3.8.1 Model formulation . . . . .	114
3.8.2 Examples . . . . .	115
3.9 Sports strategy . . . . .	117
3.9.1 When to pull the goalie in ice hockey . . . . .	118
3.9.2 A tennis handicap system . . . . .	120
3.10 The art of modeling . . . . .	122
<b>4 Finite Horizon Models</b>	<b>135</b>
4.1 The value of a policy in a finite horizon model . . . . .	136
4.1.1 Backwards induction . . . . .	137
4.1.2 Evaluating a Markovian deterministic policy . . . . .	138
4.1.3 Evaluating a history-dependent randomized policy* . . . . .	143
4.2 Optimal policies and their values . . . . .	144
4.2.1 Definitions . . . . .	144
4.2.2 Computing optimal value functions and finding optimal policies . . . . .	146
4.2.3 A numerical example . . . . .	150
4.3 Applications . . . . .	150
4.3.1 Queuing service rate control . . . . .	151
4.3.2 Revenue management . . . . .	154
4.3.3 Online dating: Finding the best match . . . . .	157
4.4 Interpretability: Optimality of structured policies . . . . .	162
4.4.1 A general approach . . . . .	163
4.4.2 Example: Preventive maintenance . . . . .	164
4.4.3 Monotonicity of optimal policies in service rate control queuing systems* . . . . .	166
4.5 State-action value functions . . . . .	171

4.5.1	State-action value function recursion for a fixed policy	171
4.5.2	Revisiting the two-state model from the state-action value function perspective	173
4.5.3	Optimal state-action value functions	174
4.5.4	An example	177
4.5.5	Optimality equations in the queuing admission control model*	178
4.6	Technical appendix	180
4.6.1	Finding optimal policies in $\Pi^{\text{MD}}$	180
4.6.2	Finding optimal values in $\Pi^{\text{HR}}$	181
4.6.3	Optimality of Markov deterministic policies in $\Pi^{\text{HR}}$	182
<b>5</b>	<b>Infinite Horizon Models: Expected Discounted Reward</b>	<b>189</b>
5.1	Preliminaries	191
5.1.1	Reward functions	191
5.1.2	Key assumptions	191
5.1.3	Markovian policies are sufficient	192
5.1.4	Matrix and vector representation for stationary policies	192
5.1.5	Partial orders	194
5.1.6	Norms	195
5.2	The expected discounted reward of a policy	197
5.2.1	Insights into how the expected discounted reward is evaluated	197
5.2.2	Representing value functions as vectors	199
5.2.3	Evaluating stationary policies	201
5.3	Optimal policies and the Bellman equation	204
5.3.1	Optimal policies	204
5.3.2	The Bellman equation for a discounted model	205
5.3.3	The maximum return operator	207
5.3.4	Existence of an optimal value function	210
5.3.5	Existence of an optimal stationary policy	212
5.4	State-action value functions	215
5.4.1	State-action value function recursions for a fixed policy	216
5.4.2	Bellman equation for state-action value functions	216
5.4.3	An example	217
5.4.4	Existence of solutions to the Bellman equation for state-action value functions*	218
5.5	Spans, bounds and other technical concepts	219
5.5.1	Spans and span contractions	220
5.5.2	Properties of $L\mathbf{v} - \mathbf{v}$	225
5.5.3	Bounds	227
5.5.4	Stopping criteria	230
5.5.5	Action elimination	231
5.6	Value iteration	233
5.6.1	A value iteration algorithm	233

5.6.2	Gauss-Seidel value iteration	238
5.6.3	Value iteration with action elimination	244
5.7	Policy iteration	246
5.7.1	The policy iteration algorithm	246
5.7.2	Convergence of policy iteration	251
5.7.3	Policy iteration and Newton's method*	253
5.8	Modified policy iteration	256
5.8.1	Motivation	257
5.8.2	A modified policy iteration algorithm	258
5.8.3	Convergence of modified policy iteration*	263
5.9	Linear programming	265
5.9.1	The primal linear program	266
5.9.2	The dual linear program	271
5.9.3	Dual feasible solutions and stationary policies	273
5.9.4	Solving linear programs	282
5.10	Optimality of structured policies	284
5.10.1	The fundamental result	284
5.10.2	A preventive maintenance example	285
5.11	Application: Queuing service rate control	286
5.11.1	The model	286
5.11.2	Value iteration	287
5.11.3	Gauss-Seidel value iteration	288
5.11.4	Policy iteration	289
5.11.5	Modified policy iteration	292
5.11.6	Linear Programming	294
5.11.7	Concluding remarks on queuing control application	296
5.12	Application: Clinical decision making	297
5.12.1	The Bellman equation	298
5.12.2	A numerical example	298
5.12.3	Numerical solution	299
5.12.4	Interpretation of results	301
5.12.5	Concluding remarks on clinical decision making application	301
5.13	Technical appendix	302
5.13.1	Proof of Theorem 5.1	302
5.13.2	Bounds for Gauss-Seidel iteration	306
5.13.3	Convergence of policy iteration in non-finite models*	310
<b>6</b>	<b>Infinite Horizon Models: Expected Total Reward</b>	<b>320</b>
6.1	Introduction	320
6.1.1	Motivating examples	321
6.2	Model classification	323
6.2.1	Preliminaries	324
6.2.2	Transient models	326

6.2.3	Stochastic shortest path models	329
6.2.4	Positive models	330
6.2.5	Negative models	330
6.2.6	Comparison of model classes	331
6.3	Value functions, optimal policies and the Bellman equation	333
6.3.1	The Bellman equation in an expected total reward model	334
6.3.2	The transient Bellman equation	335
6.3.3	Solutions of the Bellman equation in expected total reward models	336
6.3.4	Existence of optimal stationary policies in expected total reward models*	338
6.3.5	Identification of optimal policies	340
6.4	State-action value functions	342
6.5	Value iteration	342
6.5.1	Examples	343
6.5.2	Transient models	346
6.5.3	Stochastic shortest path models	356
6.5.4	Positive models	356
6.5.5	Negative models	360
6.6	Policy iteration	362
6.6.1	Transient models	363
6.6.2	Stochastic shortest path models	365
6.7	Modified policy iteration	365
6.7.1	Transient models	365
6.7.2	Convergence in transient models	367
6.7.3	An example	367
6.7.4	Stochastic shortest path models	368
6.8	Linear programming in transient models	368
6.8.1	The primal linear program	369
6.8.2	The dual linear program	369
6.8.3	Examples	371
6.9	Optimality of structured policies	373
6.9.1	The fundamental result	374
6.10	Applications	374
6.10.1	Gridworld navigation	374
6.10.2	Optimal stopping	376
<b>7</b>	<b>Infinite Horizon Models: Long-Run Average Reward</b>	<b>390</b>
7.1	Preliminaries	391
7.2	The long-run average reward or gain	392
7.2.1	The gain of a stationary policy	392
7.2.2	The gain of a non-stationary policy need not exist	393
7.2.3	A more general definition of the gain	395
7.2.4	Average optimality	396

7.3	Chain structure	397
7.4	Bias of a stationary policy	403
7.4.1	Interpreting the bias	404
7.4.2	Computing the gain and bias of a stationary policy	410
7.5	The Bellman equation and its properties	418
7.5.1	The Bellman equation in recurrent and unichain models	418
7.5.2	Bounds on the optimal gain	420
7.5.3	Existence of solutions of the Bellman equation	423
7.5.4	The Bellman equation and optimal policies	424
7.5.5	State-action value functions	427
7.5.6	The Bellman equation in multi-chain models*	430
7.6	Value iteration	430
7.6.1	Examples	431
7.6.2	Removing periodicity	433
7.6.3	A value iteration algorithm	435
7.6.4	Relative value iteration	440
7.7	Policy iteration	442
7.7.1	An algorithm	442
7.7.2	An example	444
7.7.3	Convergence of recurrent and unichain policy iteration	445
7.8	Linear programming	451
7.8.1	Linear programming formulation	452
7.8.2	Recurrent models	454
7.8.3	Unichain models	459
7.8.4	Constrained Markov decision processes	460
7.9	Optimality of structured policies	461
7.10	Queuing service rate control	463
7.10.1	The model	463
7.10.2	Value iteration	463
7.10.3	Policy iteration	464
7.10.4	Linear programming	466
7.10.5	Concluding remarks on computation	468
<b>8</b>	<b>Partially Observable Markov Decision Processes</b>	<b>474</b>
8.1	Model overview	475
8.1.1	POMDP dynamics	476
8.1.2	A two-state POMDP model	480
8.2	Information and belief states	484
8.3	Transforming a POMDP into a Markov decision process	488
8.3.1	The derived Markov decision process	489
8.3.2	The two-state model revisited	493
8.4	A finite horizon model	496
8.4.1	Optimality criterion	496

8.4.2	The Bellman equation and its solution	498
8.4.3	Structure of optimal value functions in a POMDP	501
8.4.4	An exact algorithm for finite horizon POMDPs	506
8.4.5	A finite-grid approximation for finite horizon POMDPs	513
8.5	Infinite horizon models	524
8.5.1	Optimality criterion	524
8.5.2	Computing optimal values and finding optimal policies	525
8.6	Examples	528
8.6.1	Preventive maintenance and inspection	528
8.6.2	Models with unknown parameters: Bayesian decision problems	533
8.6.3	Multi-armed bandits	536
8.6.4	Breast cancer screening	542
8.6.5	Controlling autonomous robots with noisy sensors	545
<b>9</b>	<b>Value Function Approximation</b>	<b>559</b>
9.1	Introduction	559
9.2	Approximating policy value functions	561
9.2.1	Features	563
9.2.2	Feature-based value function approximators	566
9.2.3	State-action value functions	571
9.3	Parameter estimation for policy value functions: Linear architectures	571
9.3.1	Least squares policy evaluation	575
9.3.2	Linear programming approximations for policy value functions	581
9.4	Parameter estimation for a fixed policy: Nonlinear architectures	584
9.5	Combining approximation and optimization	585
9.5.1	Iterative methods for linear architectures	586
9.5.2	Iterative methods for nonlinear architectures	593
9.5.3	Optimization using linear programming: Linear architectures	595
9.6	Application: strategic scheduling	600
9.6.1	A numerical example	604
9.6.2	Value iteration without approximation	604
9.6.3	LSVI with linear value function approximation	605
9.6.4	LSVI with quadratic value function approximation	607
9.6.5	Comparison of policies and values	608
9.6.6	Further insights	609
9.7	Application: Golf strategy	612
9.7.1	Value functions and strokes gained in golf	612
9.8	Technical appendix: Regression and nonlinear optimization	619
9.8.1	Linear regression	619
9.8.2	Matrix formulation	621
9.8.3	Nonlinear regression	625
9.8.4	Nonlinear optimization	625



<b>10 Simulation in Tabular Models</b>	<b>634</b>
10.1 Preliminaries	636
10.1.1 Data generation	637
10.1.2 Trade-offs	640
10.2 Methods overview: The learning newsvendor	641
10.2.1 Analysis by simulation	642
10.2.2 A numerical study	646
10.2.3 Conclusions	649
10.3 Policy evaluation: Episodic models	651
10.3.1 Monte Carlo methods	652
10.3.2 Temporal differencing	656
10.3.3 TD( $\gamma$ )	667
10.4 Policy evaluation: Infinite horizon discounted models	677
10.4.1 Monte Carlo methods	677
10.4.2 Online TD( $\gamma$ ) in an infinite horizon discounted model	680
10.5 Policy evaluation: Infinite horizon average reward models	685
10.5.1 Monte Carlo methods	686
10.5.2 Temporal differencing (TD(0))	688
10.6 Policy optimization	692
10.6.1 Two perspectives	692
10.6.2 Model selection	692
10.6.3 Preliminaries	693
10.6.4 Selecting actions: Exploration vs. exploitation	695
10.7 Q-learning and SARSA	697
10.7.1 Episodic models	699
10.7.2 Infinite horizon discounted models	706
10.7.3 Infinite horizon average reward models	712
10.8 Policy iteration type algorithms*	719
10.8.1 Hybrid policy improvement	720
10.8.2 Hybrid policy iteration bounds	722
10.8.3 Online policy improvement	725
10.9 Queuing service rate control revisited	728
10.9.1 Monte Carlo policy evaluation	728
10.9.2 TD( $\gamma$ ) policy evaluation	729
10.9.3 Q-learning	732
10.9.4 Policy iteration type algorithms	735
10.9.5 Computational summary: Discounted model	738
10.9.6 Average reward queuing control	739
10.10 Conclusion	741
10.11 Technical appendix	741
10.11.1 Choosing good estimators	741
10.11.2 Stochastic approximation	745
10.11.3 Stochastic approximation and optimization	750

10.11.4 Offline TD( $\gamma$ ): Technical details . . . . .	757
<b>11 Simulation with Function Approximation</b> . . . . .	<b>768</b>
11.1 The challenge of large models . . . . .	769
11.1.1 Features . . . . .	769
11.1.2 Policy value function approximation . . . . .	770
11.1.3 State-action value function approximation . . . . .	771
11.1.4 Policy approximation . . . . .	774
11.2 Policy value function approximation . . . . .	775
11.2.1 Monte Carlo policy value function approximation . . . . .	775
11.2.2 Temporal differencing with policy value function approximation . . . . .	781
11.2.3 TD( $\gamma$ ) with linear policy value function approximation . . . . .	789
11.3 Optimization based on value function approximation: Q-learning . . . . .	792
11.3.1 Motivation . . . . .	793
11.3.2 Discounted Q-learning with function approximation . . . . .	794
11.3.3 Q-learning in an episodic model . . . . .	798
11.4 Q-policy iteration . . . . .	802
11.4.1 Motivation . . . . .	803
11.4.2 An algorithm . . . . .	803
11.4.3 Newsvendor model . . . . .	806
11.4.4 Queuing service rate control model . . . . .	810
11.4.5 Concluding remarks on examples . . . . .	812
11.5 Policy space methods . . . . .	812
11.5.1 Motivation: A policy gradient algorithm for a one-state one- period model . . . . .	813
11.5.2 A policy gradient algorithm for a Markov decision process . . . . .	817
11.5.3 The gradient of the policy value function for an undiscounted infinite horizon Markov decision process . . . . .	819
11.5.4 A simple policy gradient algorithm . . . . .	820
11.5.5 An example . . . . .	823
11.5.6 Policy gradient in a discounted model . . . . .	828
11.5.7 Policy gradient: Concluding remarks . . . . .	830
11.6 Actor-critic algorithms: Combining policy and policy value function ap- proximation . . . . .	830
11.6.1 An online actor-critic algorithm . . . . .	833
11.6.2 A batch actor-critic algorithm . . . . .	837
11.6.3 Actor-critic methods: Concluding remarks and enhancements . . . . .	841
11.7 Further topics . . . . .	843
11.7.1 Simultaneous learning and control . . . . .	843
11.7.2 Experience replay . . . . .	844
11.7.3 Deep learning . . . . .	845
11.7.4 Importance sampling . . . . .	845
11.7.5 Monte Carlo tree search . . . . .	847

11.7.6 Partially observable models . . . . .	850
11.8 Technical appendix: Derivation of policy gradient representation . . . .	850
<b>A Notation and Conventions</b>	<b>858</b>
A.1 General math notation . . . . .	858
A.2 Notation specific to MDPs . . . . .	860
A.3 Conventions . . . . .	861
A.4 Abbreviations . . . . .	862
<b>B Markov Chains</b>	<b>863</b>
B.1 What is a finite Markov chain? . . . . .	863
B.2 Classifying states . . . . .	864
B.3 Classes and class structure . . . . .	865
B.3.1 Chain structure . . . . .	866
B.4 Limiting behavior . . . . .	867
B.5 An important lemma . . . . .	870
B.6 The deviation matrix . . . . .	871
B.7 Structure of $\mathbf{P}^*$ and $\mathbf{H}$ . . . . .	872
B.8 Some examples . . . . .	874
B.9 Eigenvalues and eigenvectors of a transition matrix* . . . . .	876
B.10 Absorbing chains . . . . .	878
B.11 Countable state chains* . . . . .	880
<b>C Linear Programming</b>	<b>883</b>
C.1 Duality . . . . .	885