

Hyeong Soo Chang, Michael C. Fu, Jiaqiao Hu,
and Steven I. Marcus

Simulation-Based Algorithms for Markov Decision Processes

SPIN Springer's internal project number, if known

– Monograph –

October 11, 2006

Springer

Berlin Heidelberg New York

Hong Kong London

Milan Paris Tokyo

Preface

Markov decision process (MDP) models are widely used for modeling sequential decision-making problems that arise in engineering, computer science, operations research, economics, and other social sciences. However, it is well known that many real-world problems modeled by MDPs have huge state and/or action spaces, leading to the well-known curse of dimensionality that makes solution of the resulting models intractable. In other cases, the system of interest is complex enough that it is not feasible to explicitly specify some of the MDP model parameters, but simulated sample paths can be readily generated (e.g., for random state transitions and rewards), albeit at a non-trivial computational cost. For these settings, we have developed various sampling and population-based numerical algorithms to overcome the computational difficulties of computing an optimal solution in terms of a policy and/or value function. Specific approaches include multi-stage adaptive sampling, evolutionary policy iteration and random policy search, and model reference adaptive search. This book brings together these algorithms and presents them in a unified manner accessible to researchers with varying interests and background. In addition to providing numerous specific algorithms, the exposition includes both illustrative numerical examples and rigorous theoretical convergence results. These approaches are distinct from but complementary to those computational approaches for solving MDPs based on explicit state-space reduction, such as neuro-dynamic programming or reinforcement learning; in fact, the computational gains achieved through approximations and parameterizations to reduce the size of the state space can be incorporated into our proposed algorithms.

Our focus is on computational approaches for calculating or estimating optimal value functions and finding optimal policies (possibly in a restricted policy space). What this book does not make any attempts at doing:

- (i) determining important theoretical and fundamental properties of the MDP models, such as existence of optimal policies and uniqueness of the optimal value function;
- (ii) teaching the use of MDPs for modeling real-world problems.

In particular, we eschew the technical mathematics associated with defining continuous state and action space MDP models. However, we do provide a rigorous theoretical treatment of convergence properties of the algorithms. Thus, this book is aimed at researchers in MDPs and applied probability modeling with an interest in numerical computation. The mathematical prerequisites are relatively mild: mainly a strong grounding in calculus-based probability theory and some familiarity with Markov decision processes or stochastic dynamic programming; as a result, this book is meant to be accessible to graduate students, particularly those in control, operations research, computer science, and economics.

We begin with a formal description of the discounted reward MDP framework in Chapter 1, including both the finite and infinite horizon settings and summarizing the associated optimality equations. We then present the well-known exact solution algorithms, value iteration and policy iteration, and outline a framework of rolling horizon control (also called receding horizon control) as an approximate solution methodology for solving MDPs, in conjunction with simulation-based approaches covered later in the book. We conclude with a brief survey of other recently proposed MDP solution techniques designed to break the curse of dimensionality.

In Chapter 2, we present simulation-based algorithms for estimating the optimal value function in finite horizon MDPs with large (possibly uncountable) state spaces, where the usual techniques of policy iteration and value iteration are either computationally impractical or infeasible to implement. We present two adaptive sampling algorithms that estimate the optimal value function by choosing actions to sample in each state visited on a finite horizon simulated sample path. The first approach builds upon the expected regret analysis of multi-armed bandit models and uses upper confidence bounds to determine which action to sample next, whereas the second approach uses ideas from learning automata to determine the next sampled action.

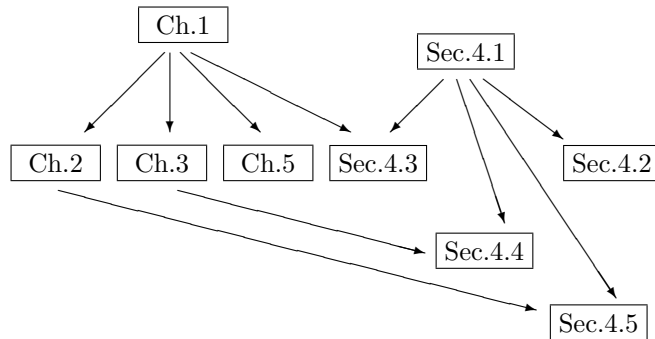
Chapter 3 considers infinite horizon problems and presents evolutionary approaches for finding an optimal policy. The algorithms in this chapter work with a population of policies — in contrast to the usual policy iteration approach, which updates a single policy — and are targeted at problems with large action spaces (again possibly uncountable) and relatively small state spaces. Although the algorithms are presented for the case where the distributions on state transitions and rewards are known explicitly, extension to the setting when this is not the case is also discussed, where finite-horizon simulated sample paths would be used to estimate the value function for each policy in the population.

In Chapter 4, we consider a global optimization approach called model reference adaptive search (MRAS), which provides a broad framework for updating a probability distribution over the solution space in a way that ensures convergence to an optimal solution. After introducing the theory and convergence results in a general optimization problem setting, we apply the MRAS approach to various MDP settings. For the finite and infinite horizon settings,

we show how the approach can be used to perform optimization in policy space. In the setting of Chapter 3, we show how MRAS can be incorporated to further improve the exploration step in the evolutionary algorithms presented there. Finally, for the finite horizon setting with both large state and action spaces, we propose a method for combining the approaches of Chapters 2 and 4 in order to sample the state and action spaces, respectively.

In Chapter 5, we consider an approximate rolling horizon control framework for solving infinite-horizon MDPs with large state/action spaces in an on-line manner by simulation. Specifically, we consider policies in which the system (either the actual system itself or a simulation model of the system) evolves to a particular state that is observed, and the action to be taken in that particular state is then computed on line at the decision time, with a particular emphasis on the use of simulation. We first present an updating scheme involving multiplicative weights for updating a probability distribution over a restricted set of policies, which can be used to estimate the optimal value function over this restricted set by sampling on the (restricted) policy space. The lower-bound estimate of the optimal value function is used for constructing on-line control policies, called (simulated) policy switching and parallel rollout. We also discuss an upper-bound based method, called hind-sight optimization.

The relationship between the chapters and/or sections of the book is shown below. After reading Chapter 1, Chapters 2, 3, and 5 can pretty much be read independently, although Chapter 5 does allude to algorithms in each of the previous chapters, and the numerical example in Section 5.1 is taken from Section 2.1. The first two sections of Chapter 4 present a general global optimization approach, which is then applied to MDPs in the subsequent Sections 4.3, 4.4 and 4.5, where the latter two build upon work in Chapters 3 and 2, respectively.



We thank Chunyue Song, Yongqiang Wang, Enlu Zhou, Jeff Heath, Scott Nestler, and Huiju Zhang for their comments on various portions and drafts of the manuscript. Finally, we acknowledge the financial support of several US Federal funding agencies for this work: the National Science Foundation (under Grants DMI-9988867 and DMI-0323220), the Air Force Office of Scientific Research (under Grants F496200110161 and FA95500410210), and the Department of Defense.

Hyeong Soo Chang, Seoul, Korea
Michael Fu, College Park, Maryland
Jiaqiao Hu, Stony Brook, New York
Steve Marcus, College Park, Maryland
September 2006

Contents

Selected Notation and Abbreviations	xiii
1 Markov Decision Processes	1
1.1 Optimality Equations	3
1.2 Policy Iteration and Value Iteration	5
1.3 Rolling Horizon Control	7
1.4 Survey of Previous Work on Computational Methods	8
1.5 Simulation	11
1.6 Preview of Coming Attractions	14
1.7 Notes	15
2 Multi-stage Adaptive Sampling Algorithms	17
2.1 Upper Confidence Bound Sampling	19
2.1.1 Regret Analysis in Multi-armed Bandits	19
2.1.2 Algorithm Description	20
2.1.3 Alternative Estimators	21
2.1.4 Convergence Analysis	24
2.1.5 Numerical Example	31
2.2 Pursuit Learning Automata Sampling	40
2.2.1 Algorithm Description	41
2.2.2 Convergence Analysis	42
2.2.3 Application to POMDPs	51
2.3 Notes	53
3 Population-based Evolutionary Approaches	55
3.1 Evolutionary Policy Iteration	57
3.1.1 Policy Switching	57
3.1.2 Policy Mutation and Population Generation	59
3.1.3 Stopping Rule	59
3.1.4 Convergence Analysis	60
3.1.5 Parallelization	61

3.2	Evolutionary Random Policy Search	61
3.2.1	Policy Improvement with Reward Swapping	62
3.2.2	Exploration	65
3.2.3	Convergence Analysis	67
3.3	Numerical Examples	70
3.3.1	A One-dimensional Queueing Example	70
3.3.2	A Two-dimensional Queueing Example	78
3.4	Extension to Simulation-based Setting	81
3.5	Notes	81
4	Model Reference Adaptive Search	83
4.1	The Model Reference Adaptive Search Method	85
4.1.1	The MRAS ₀ Algorithm (Idealized Version)	87
4.1.2	The MRAS ₁ Algorithm (Adaptive Monte Carlo Version)	90
4.1.3	The MRAS ₂ Algorithm (Stochastic Optimization)	92
4.2	Convergence Analysis	95
4.3	Application to MDPs via Direct Policy Learning	123
4.3.1	Finite Horizon MDPs	124
4.3.2	Infinite Horizon MDPs	124
4.3.3	MDPs with Large State Spaces	126
4.3.4	Numerical Examples	126
4.4	Application to Infinite Horizon MDPs in Population-based Evolutionary Approaches	135
4.4.1	Algorithm Description	135
4.4.2	Numerical Examples	137
4.5	Application to Finite Horizon MDPs using Adaptive Sampling	140
4.6	Notes	142
5	On-line Control Methods via Simulation	143
5.1	Simulated Annealing Multiplicative Weights Algorithm	147
5.1.1	Basic Algorithm Description	148
5.1.2	Convergence Analysis	149
5.1.3	Convergence of the Sampling Version of the Algorithm	152
5.1.4	Numerical Example	154
5.1.5	Simulated Policy Switching	158
5.2	Rollout	159
5.2.1	Parallel Rollout	160
5.3	Hindsight Optimization	162
5.3.1	Numerical Example	163
5.4	Notes	169
	References	171
	Index	181

Index

- acceptance-rejection method, 12
- action selection distribution, 55, 56, 58, 65, 67, 70, 80, 135–137
- adaptive multi-stage sampling (AMS), 18, 53, 83, 140
- aggregation, 9, 10, 16
- ant colony optimization, 142
- approximate dynamic programming, 15
- asymptotic, 17, 19, 20, 23, 24, 72, 111, 148
- Azuma’s inequality, 153

- backward induction, 7, 169
- base-stock, 33
- basis function representation, 9
- Bellman optimality principle, 4, 55
- bias, 24, 30
- Borel-Cantelli lemma, 104, 109, 112, 116, 119, 121, 153

- Chebyshev’s inequality, 111, 115
- common random numbers, 13, 140, 146, 160
- composition method, 12
- conditional Monte Carlo, 13
- control variates, 13
- convergence, 6–8, 10, 12, 15, 16, 24, 27–29, 34, 36–39, 56, 59–62, 65, 67, 68, 70, 71, 74, 81, 85, 88–90, 95, 97, 100–102, 111, 113–115, 128, 130, 131, 139, 150, 152, 154, 155, 157–159, 170
- convex(ity), 9, 12, 16, 71, 102, 105, 106, 110, 133

- convolution method, 12
- counting measure, 96, 105, 116
- cross-entropy (CE) method, 142

- differential training, 160, 169
- direct policy search, 123
- discrete measure, 95, 98
- dominated convergence theorem, 96, 103
- dynamic programming, 3, 8, 9, 15, 33

- elite, 87, 140
- elite policy, 55, 57, 62, 82, 135
- ergodicity, 169
- estimation of distribution algorithms (EDA), 142
- evolutionary policy iteration (EPI), 56–59, 61, 62, 64, 65, 70, 74–76, 81, 82
- evolutionary random policy search (ERPS), 56, 61–65, 67–83, 135–139, 162
- exploitation, 18–20, 63, 65–67, 71, 75, 77, 80, 83, 135, 139
- exploration, 18–20, 55–57, 59, 63, 65, 80, 83

- finite horizon, 3, 4, 7, 8, 14, 15, 17, 31, 81, 124–126, 143, 144, 147, 159, 161, 163, 169
- fixed-point equation, 4

- Gaussian, 12, 67, 84, 90, 126
- Gaussian elimination, 65

- genetic algorithms (GA), 55, 59, 81, 142
- genetic search, 81
- global optimal/optimizer/optimum, 85, 90, 92, 95, 101, 113, 131
- global optimization, 66, 83, 85, 141
- heuristic, 34, 62, 63, 65, 66, 75, 147, 159, 161, 164, 168, 169
- hidden Markov model (HMM), 163–165
- hindsight optimization, 147, 162–168, 170
- Hoeffding inequality, 26, 103, 108
- importance sampling, 13
- infinite horizon, 3–5, 7, 8, 14, 15, 17, 53, 81, 83, 123, 124, 126, 127, 130, 137, 142–144, 146, 161, 162, 168
- information-state, 51, 53
- inventory, 15, 23, 31, 33, 35–39, 66, 126, 128, 129, 133, 134, 154–157
- inverse transform method, 12
- Jensen’s inequality, 162, 170
- Kullback-Leibler (KL) divergence, 86, 88, 89, 93, 149
- large deviations principle, 111, 113
- learning automata, 40, 53
- Lebesgue measure, 95, 96, 98, 105, 116
- linear congruential generator (LCG), 11
- linear programming, 15
- Lipschitz condition, 67, 113
- local optima, 59, 65, 74
- low discrepancy sequence, 12
- Markov chain, 53, 169
- Markov reward process, 170
- Markovian policy, 2, 3
- metric, 65–67
- model-based methods, 83, 84, 142
- modularity, 9, 16
- monotonicity, 6, 9, 16, 30, 55–57, 64, 116, 135, 159
- multi-armed bandit, 18, 19, 21, 53, 140
- multi-policy improvement, 162, 169
- multi-policy iteration, 169
- multivariate normal distribution, 84, 90, 100
- mutation, 55–63, 74, 75
- natural exponential family (NEF), 89, 90, 95, 96, 100, 101, 114, 130
- nearest neighbor heuristic, 62, 63, 65, 66, 75
- neighborhood, 95, 104, 113, 130
- nested partitions method, 142
- neural network, 10, 16
- neuro-dynamic programming, 9, 15
- newsboy problem, 33
- nonstationary, 1, 33, 124
- nonstationary policy, 2, 7, 17, 31, 124, 125, 128, 144, 146, 155
- norm, 66–68
- off-line, 143, 164, 166, 167
- on-line, 7, 15, 81, 143, 146, 147, 159–161, 168
- optimal, 17
- optimal policy, 2–6, 8, 10, 15, 32, 33, 46, 55, 56, 58–63, 65, 67–70, 81, 124, 126, 130, 131, 133, 136, 144, 147–149, 155, 159
- optimal reward-to-go value, 3, 5
- optimal value, 3, 4, 7, 10, 15, 18, 20, 24, 34, 40–42, 143, 144, 150, 152
- optimal value function, 2–5, 8, 10, 17, 18, 20, 24, 68, 71, 74, 78, 82, 123, 147, 155
- optimality equation, 3, 7, 14, 17, 64
- order statistic, 92, 93, 126
- ordinal comparison, 158, 170
- parallel rollout, 147, 161, 162, 164–170
- parallelization, 61, 156
- parameterized, 16, 51, 83, 84, 126, 128, 142
- parameterized distribution, 84, 85, 89, 90, 95, 101, 123, 124, 130, 135, 141, 142
- partially observable Markov decision process (POMDP), 40, 51–53, 169
- Pinsker’s inequality, 151
- policy evaluation, 6, 14, 16, 58, 65, 137
- policy improvement, 6, 16, 55, 57, 58, 61, 62, 64, 65, 160
- policy improvement with reward swapping (PIRS), 61–65, 71, 75, 77, 81, 135, 136, 162

- policy iteration (PI), 5–9, 15, 16, 55–59, 62, 70–74, 78–82, 123, 143, 160, 162, 169
- policy switching, 56–58, 61, 64, 65, 81, 147, 158, 159, 169, 170
- population, 14, 55–58, 62–65, 69, 71, 72, 74, 75, 78, 79, 82, 83, 127, 135–137, 139–141
- probability collectives, 142
- projection, 84, 86
- pursuit algorithm, 40, 53
- pursuit learning automata (PLA)
 - sampling algorithm, 19, 40–43, 45, 47, 49–53, 146
- Q -function, 3, 5, 9, 10, 16–18, 20–22, 24, 25, 34, 41, 42, 52, 140, 141, 163, 168
- Q -learning, 9, 16
- quantile, 87, 88, 90–94, 102, 103, 105, 125–127, 140
- quasi-Monte Carlo sequence, 12
- queue(ing), 70, 71, 77–79, 82, 126, 130, 132, 137, 163, 164, 169
- random number, 2, 11, 12, 17, 18, 21, 22, 24, 25, 41, 47, 146, 148, 154, 158, 160, 163, 165
- random search method, 66
- random variate, 11, 12, 16
- randomized policy, 9, 146
- receding horizon control, 7, 15
- reference distribution, 84
- regret, 19, 20, 24, 53
- reinforcement learning, 9, 15
- rolling horizon control, 7, 8, 15, 143, 144, 146, 147, 159, 162, 163, 168, 169
- rollout, 159, 169
- (s, S) policy, 33, 126, 133
- sample path, 8, 12–14, 106, 114, 121, 124, 131, 142, 147, 148, 158
- sampled tree, 18, 40, 41
- scheduling, 13, 163–165, 168–170
- simulated annealing (SA), 128–135, 142, 148
- simulated annealing multiplicative weight (SAMW), 147–157, 162, 169
- simulated policy switching, 158
- stationary, 1, 7, 126, 165
- stationary policy, 3, 7, 8, 15, 55, 124, 127, 131, 140, 144, 170
- stochastic approximation, 9, 10, 142
- stochastic matrix, 128
- stopping rule, 34, 56, 58, 59, 63, 72, 75, 87, 91, 94, 125, 127, 136
- stratified sampling, 13
- sub-MDP, 61–65
- successive approximation, 7, 15
- supermartingale, 44
- tabu search, 142
- threshold, 33, 87, 88, 91–93, 111, 113, 116, 126, 133, 155, 157, 169
- total variation distance, 150
- unbiased, 19, 23, 24, 65, 75, 103
- uniform distribution, 34, 41, 59, 61, 66, 70, 130, 131, 133, 137, 148, 150, 155, 165
- upper confidence bound (UCB) sampling algorithm, 18–24, 29, 30, 32, 33, 40–42, 53, 146
- validation, 11, 13
- value function, 3, 4, 6–9, 15, 16, 35, 81
- value iteration (VI), 5, 7, 8, 15, 55, 81, 82, 123, 143, 144
- variance reduction, 11, 13
- verification, 11, 13
- Wald’s equation, 29
- weighted majority algorithm, 169