

# **Markov Decision Processes in Artificial Intelligence**

*MDPs, Beyond MDPs and Applications*

Edited by  
Olivier Sigaud  
Olivier Buffet

**ISTE**

 **WILEY**



## Markov Decision Processes in Artificial Intelligence



# **Markov Decision Processes in Artificial Intelligence**

*MDPs, Beyond MDPs and Applications*

Edited by  
Olivier Sigaud  
Olivier Buffet

**ISTE**

 **WILEY**

First published 2008 in France by Hermes Science/Lavoisier in two volumes entitled: *Processus décisionnels de Markov en intelligence artificielle* © LAVOISIER 2008

First published 2010 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the CLA. Enquiries concerning reproduction outside these terms should be sent to the publishers at the undermentioned address:

ISTE Ltd  
27-37 St George's Road  
London SW19 4EU  
UK

[www.iste.co.uk](http://www.iste.co.uk)

John Wiley & Sons, Inc.  
111 River Street  
Hoboken, NJ 07030  
USA

[www.wiley.com](http://www.wiley.com)

© ISTE Ltd 2010

The rights of Olivier Sigaud and Olivier Buffet to be identified as the authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

---

Library of Congress Cataloging-in-Publication Data

Markov decision processes in artificial intelligence : MDPs, beyond MDPs and applications / edited by Olivier Sigaud, Olivier Buffet.

p. cm.

Includes bibliographical references and index.

ISBN 978-1-84821-167-4

1. Artificial intelligence--Mathematics. 2. Artificial intelligence--Statistical methods. 3. Markov processes. 4. Statistical decision. I. Sigaud, Olivier. II. Buffet, Olivier.

Q335.M374 2010

006.301'509233--dc22

2009048651

---

British Library Cataloguing-in-Publication Data

A CIP record for this book is available from the British Library

ISBN 978-1-84821-167-4

---

Printed and bound in Great Britain by CPI Antony Rowe, Chippenham and Eastbourne



## Table of Contents

<b>Preface</b> . . . . .	xvii
<b>List of Authors</b> . . . . .	xix
<b>PART 1. MDPs: MODELS AND METHODS</b> . . . . .	1
<b>Chapter 1. Markov Decision Processes</b> . . . . .	3
Frédéric GARCIA and Emmanuel RACHELSON	
1.1. Introduction . . . . .	3
1.2. Markov decision problems . . . . .	4
1.2.1. Markov decision processes . . . . .	4
1.2.2. Action policies . . . . .	7
1.2.3. Performance criterion . . . . .	8
1.3. Value functions . . . . .	9
1.3.1. The finite criterion . . . . .	10
1.3.2. The $\gamma$ -discounted criterion . . . . .	10
1.3.3. The total reward criterion . . . . .	11
1.3.4. The average reward criterion . . . . .	11
1.4. Markov policies . . . . .	12
1.4.1. Equivalence of history-dependent and Markov policies . . . . .	12
1.4.2. Markov policies and valued Markov chains . . . . .	13
1.5. Characterization of optimal policies . . . . .	14
1.5.1. The finite criterion . . . . .	14
1.5.1.1. Optimality equations . . . . .	14
1.5.1.2. Evaluation of a deterministic Markov policy . . . . .	15
1.5.2. The discounted criterion . . . . .	16
1.5.2.1. Evaluation of a stationary Markov policy . . . . .	16
1.5.2.2. Optimality equations . . . . .	17
1.5.3. The total reward criterion . . . . .	22

1.5.4. The average reward criterion . . . . .	24
1.5.4.1. Evaluation of a stationary Markov policy . . . . .	24
1.5.4.2. Optimality equations . . . . .	27
1.6. Optimization algorithms for MDPs . . . . .	28
1.6.1. The finite criterion . . . . .	28
1.6.2. The discounted criterion . . . . .	28
1.6.2.1. Linear programming . . . . .	28
1.6.2.2. The value iteration algorithm . . . . .	29
1.6.2.3. The policy iteration algorithm . . . . .	30
1.6.3. The total reward criterion . . . . .	34
1.6.3.1. Positive MDPs . . . . .	34
1.6.3.2. Negative MDPs . . . . .	34
1.6.4. The average criterion . . . . .	35
1.6.4.1. Relative value iteration algorithm . . . . .	35
1.6.4.2. Modified policy iteration algorithm . . . . .	36
1.7. Conclusion and outlook . . . . .	37
1.8. Bibliography . . . . .	37
<b>Chapter 2. Reinforcement Learning</b> . . . . .	39
Olivier SIGAUD and Frédéric GARCIA	
2.1. Introduction . . . . .	39
2.1.1. Historical overview . . . . .	39
2.2. Reinforcement learning: a global view . . . . .	40
2.2.1. Reinforcement learning as approximate dynamic programming . .	41
2.2.2. Temporal, non-supervised and trial-and-error based learning . . .	42
2.2.3. Exploration versus exploitation . . . . .	42
2.2.4. General preliminaries on estimation methods . . . . .	44
2.3. Monte Carlo methods . . . . .	45
2.4. From Monte Carlo to temporal difference methods . . . . .	45
2.5. Temporal difference methods . . . . .	46
2.5.1. The TD(0) algorithm . . . . .	47
2.5.2. The SARSA algorithm . . . . .	48
2.5.3. The Q-learning algorithm . . . . .	49
2.5.4. The TD( $\lambda$ ), SARSA( $\lambda$ ) and Q( $\lambda$ ) algorithms . . . . .	51
2.5.5. Eligibility traces and TD( $\lambda$ ) . . . . .	52
2.5.6. From TD( $\lambda$ ) to SARSA( $\lambda$ ) . . . . .	55
2.5.7. Q( $\lambda$ ) . . . . .	56
2.5.8. The R-learning algorithm . . . . .	58
2.6. Model-based methods: learning a model . . . . .	59
2.6.1. Dyna architectures . . . . .	60
2.6.2. The $E^3$ algorithm . . . . .	61
2.6.3. The $R_{\max}$ algorithm . . . . .	62



2.7. Conclusion . . . . .	63
2.8. Bibliography . . . . .	63
<b>Chapter 3. Approximate Dynamic Programming . . . . .</b>	<b>67</b>
Rémi MUNOS	
3.1. Introduction . . . . .	68
3.2. Approximate value iteration (AVI) . . . . .	70
3.2.1. Sample-based implementation and supervised learning . . . . .	71
3.2.2. Analysis of the AVI algorithm . . . . .	73
3.2.3. Numerical illustration . . . . .	74
3.3. Approximate policy iteration (API) . . . . .	77
3.3.1. Analysis in $L^\infty$ -norm of the API algorithm . . . . .	77
3.3.2. Approximate policy evaluation . . . . .	79
3.3.3. Linear approximation and least-squares methods . . . . .	80
3.3.3.1. TD( $\lambda$ ) . . . . .	81
3.3.3.2. Least-squares methods . . . . .	82
3.3.3.3. Linear approximation of the state-action value function . . . . .	85
3.4. Direct minimization of the Bellman residual . . . . .	87
3.5. Towards an analysis of dynamic programming in $L^p$ -norm . . . . .	88
3.5.1. Intuition of an $L^p$ analysis in dynamic programming . . . . .	89
3.5.2. PAC bounds for RL algorithms . . . . .	91
3.6. Conclusions . . . . .	93
3.7. Bibliography . . . . .	93
<b>Chapter 4. Factored Markov Decision Processes . . . . .</b>	<b>99</b>
Thomas DEGRIS and Olivier SIGAUD	
4.1. Introduction . . . . .	99
4.2. Modeling a problem with an FMDP . . . . .	100
4.2.1. Representing the state space . . . . .	100
4.2.2. The <i>Coffee Robot</i> example . . . . .	100
4.2.3. Decomposition and function-specific independence . . . . .	101
4.2.3.1. Transition function decomposition . . . . .	102
4.2.3.2. Dynamic Bayesian networks in FMDPs . . . . .	103
4.2.3.3. Factored model of the transition function in an FMDP . . . . .	104
4.2.3.4. Factored model of the reward function . . . . .	105
4.2.4. Context-specific independence . . . . .	107
4.3. Planning with FMDPs . . . . .	108
4.3.1. Structured policy iteration and structured value iteration . . . . .	108
4.3.1.1. Decision trees . . . . .	108
4.3.1.2. Representation of the transition function . . . . .	108
4.3.1.3. Representation of the reward function . . . . .	110
4.3.1.4. Representation of a policy . . . . .	110
4.3.1.5. Representation of the value function . . . . .	111
4.3.1.6. Algorithms . . . . .	112

4.3.2. SPUDD: stochastic planning using decision diagrams . . . . .	112
4.3.2.1. Representing the functions of an FMDP with ADDs . . . . .	113
4.3.2.2. Algorithm . . . . .	114
4.3.3. Approximate linear programming in FMDPs . . . . .	115
4.3.3.1. Representations . . . . .	116
4.3.3.2. Representation of the transition function . . . . .	116
4.3.3.3. Representation of the reward function . . . . .	117
4.3.3.4. Policy representation . . . . .	118
4.3.3.5. Representation of the value function . . . . .	119
4.3.3.6. Algorithms . . . . .	122
4.4. Perspectives and conclusion . . . . .	122
4.5. Bibliography . . . . .	123
<b>Chapter 5. Policy-Gradient Algorithms . . . . .</b>	<b>127</b>
Olivier BUFFET	
5.1. Reminder about the notion of gradient . . . . .	128
5.1.1. Gradient of a function . . . . .	128
5.1.2. Gradient descent . . . . .	129
5.2. Optimizing a parameterized policy with a gradient algorithm . . . . .	130
5.2.1. Application to MDPs: overview . . . . .	130
5.2.1.1. First attempt to define a parameterized policy . . . . .	131
5.2.1.2. Example of definition of a parameterized policy . . . . .	131
5.2.2. Finite difference method . . . . .	132
5.2.3. Gradient estimation of $f$ in an MDP, case of the finite time horizon . . . . .	133
5.2.3.1. Time horizon 1 . . . . .	133
5.2.3.2. Time horizon $T$ . . . . .	134
5.2.4. Extension to the infinite time horizon: discounted criterion, average criterion . . . . .	137
5.2.4.1. Case of a regenerative process . . . . .	138
5.2.4.2. Using a moving window . . . . .	138
5.2.4.3. Using a discount factor . . . . .	139
5.2.5. Partially observable case . . . . .	139
5.2.6. Continuous action spaces . . . . .	141
5.3. Actor-critic methods . . . . .	143
5.3.1. A gradient estimator using the $Q$ -values . . . . .	143
5.3.2. Compatibility with an approximate value function . . . . .	144
5.3.2.1. Approximating $Q^\theta$ . . . . .	144
5.3.2.2. Compatibility of the approximators . . . . .	145
5.3.3. An actor-critic algorithm . . . . .	146
5.4. Complements . . . . .	147
5.5. Conclusion . . . . .	150
5.6. Bibliography . . . . .	150

<b>Chapter 6. Online Resolution Techniques</b>	153
Laurent PÉRET and Frédéric GARCIA	
6.1. Introduction	153
6.1.1. Exploiting time online	153
6.1.2. Online search by simulation	154
6.2. Online algorithms for solving an MDP	155
6.2.1. Offline algorithms, online algorithms	155
6.2.2. Problem formalization	155
6.2.2.1. Forward search over a reasoning horizon	156
6.2.2.2. Tree or graph?	157
6.2.2.3. Complexity and efficiency of the forward search	157
6.2.3. Online heuristic search algorithms for MDPs	158
6.2.3.1. General principles	158
6.2.3.2. The RTDP algorithm	159
6.2.3.3. The <i>LAO*</i> algorithm	160
6.2.4. Kearns, Mansour and Ng's simulation-based algorithm	163
6.2.4.1. Complexity and convergence of Kearns <i>et al.</i> 's algorithm	165
6.2.4.2. Efficiency and practical considerations	165
6.2.5. Tesauro and Galperin's rollout algorithm	165
6.2.5.1. Complexity and convergence of the rollout algorithm	166
6.2.5.2. Efficiency of the rollout algorithm	166
6.3. Controlling the search	167
6.3.1. Error bounds and pathology of the forward search	168
6.3.1.1. Pathology of the simulation-based forward search	168
6.3.1.2. Searching for a good compromise between depth and width	170
6.3.2. Iterative allocation of simulations	170
6.3.2.1. Multi-armed bandits and exploration in MDPs	171
6.3.3. Focused reinforcement learning	173
6.3.3.1. Global sampling error estimation	174
6.3.3.2. Organizing the search in successive trajectories	175
6.3.3.3. Convergence and practical considerations	176
6.3.4. Controlled rollout	178
6.3.4.1. Choice of the horizon	178
6.3.4.2. Iterative allocation of simulations	179
6.4. Conclusion	180
6.5. Bibliography	180
<b>PART 2. BEYOND MDPs</b>	185
<b>Chapter 7. Partially Observable Markov Decision Processes</b>	187
Alain DUTECH and Bruno SCHERRER	
7.1. Formal definitions for POMDPs	188
7.1.1. Definition of a POMDP	188

7.1.2. Performance criteria . . . . .	189
7.1.3. Information state . . . . .	190
7.1.3.1. Definition . . . . .	190
7.1.3.2. Complete information state . . . . .	191
7.1.3.3. Sufficient statistics . . . . .	191
7.1.3.4. Belief states . . . . .	191
7.1.4. Policy . . . . .	193
7.1.5. Value function . . . . .	195
7.2. Non-Markovian problems: incomplete information . . . . .	196
7.2.1. Adapted policies . . . . .	196
7.2.2. Discounted reward . . . . .	197
7.2.2.1. Adapted stochastic policies . . . . .	197
7.2.2.2. Adapted value function . . . . .	198
7.2.2.3. Convergence of adapted algorithms . . . . .	199
7.2.3. Adapted algorithms and adapted average reward criterion . . . . .	201
7.3. Computation of an exact policy on information states . . . . .	202
7.3.1. The general case . . . . .	202
7.3.1.1. Finite horizon . . . . .	202
7.3.1.2. Infinite horizon . . . . .	203
7.3.2. Belief states and piecewise linear value function . . . . .	203
7.3.2.1. Choice of the $\theta$ vectors . . . . .	206
7.3.2.2. Infinite horizon . . . . .	207
7.4. Exact value iteration algorithms . . . . .	207
7.4.1. Steps for the dynamic programming operator . . . . .	207
7.4.2. A parsimonious representation of $V$ . . . . .	210
7.4.2.1. Region . . . . .	210
7.4.2.2. Parsimonious representation . . . . .	212
7.4.2.3. Pruning of dominated vectors . . . . .	213
7.4.2.4. Pruning . . . . .	213
7.4.2.5. Choice of a vector for a belief state . . . . .	215
7.4.3. The WITNESS algorithm . . . . .	216
7.4.3.1. Neighborhood of a vector . . . . .	216
7.4.3.2. The algorithm . . . . .	218
7.4.4. Iterative pruning . . . . .	219
7.4.4.1. Complete enumeration . . . . .	219
7.4.4.2. Incremental enumeration . . . . .	220
7.5. Policy iteration algorithms . . . . .	222
7.6. Conclusion and perspectives . . . . .	223
7.7. Bibliography . . . . .	225
<b>Chapter 8. Stochastic Games . . . . .</b>	<b>229</b>
Andriy BURKOV, Laëtitia MATIGNON and Brahim CHAIB-DRAA	
8.1. Introduction . . . . .	229

8.2. Background on game theory . . . . .	230
8.2.1. Some basic definitions . . . . .	230
8.2.1.1. Criteria distinguishing different forms of games . . . . .	230
8.2.2. Static games of complete information . . . . .	232
8.2.2.1. Games in strategic form, pure strategy, mixed strategy . . . . .	232
8.2.2.2. Zero-sum game and minimax . . . . .	234
8.2.2.3. Equilibrium in dominating strategies . . . . .	236
8.2.2.4. Nash equilibrium . . . . .	238
8.2.3. Dynamic games of complete information . . . . .	240
8.2.3.1. Games in extensive form with perfect information . . . . .	241
8.2.3.2. Repeated games . . . . .	243
8.3. Stochastic games . . . . .	245
8.3.1. Definition and equilibrium of a stochastic game . . . . .	246
8.3.2. Solving stochastic games . . . . .	248
8.3.2.1. Game model available . . . . .	249
8.3.2.2. Game model unavailable, $\mathbf{A}_{-i}$ observed, equilibrium learners . . . . .	252
8.3.2.3. Game model unavailable, $\mathbf{A}_{-i}$ observed, opponent modeling . . . . .	254
8.3.2.4. Game model unavailable, $\mathbf{A}_{-i}$ not observed . . . . .	256
8.3.3. Complexity and scalability of multi-agent learning algorithms . . . . .	262
8.3.4. Beyond the search for an equilibrium . . . . .	264
8.3.4.1. Efficient play . . . . .	264
8.3.4.2. Regret minimization . . . . .	265
8.3.4.3. Metastrategies . . . . .	267
8.3.5. Discussion . . . . .	267
8.4. Conclusion and outlook . . . . .	269
8.5. Bibliography . . . . .	270
<b>Chapter 9. DEC-MDP/POMDP . . . . .</b>	<b>277</b>
Aur�lie BEYNIER, Fran�ois CHARPILLET, Daniel SZER and Abdel-Allah MOUADDIB	
9.1. Introduction . . . . .	277
9.2. Preliminaries . . . . .	278
9.3. Multiagent Markov decision processes . . . . .	279
9.4. Decentralized control and local observability . . . . .	280
9.4.1. Decentralized Markov decision processes . . . . .	281
9.4.2. Multiagent team decision problems . . . . .	283
9.4.3. Complexity . . . . .	285
9.5. Sub-classes of DEC-POMDPs . . . . .	285
9.5.1. Transition and observation independence . . . . .	285
9.5.2. Goal oriented DEC-POMDPs . . . . .	287
9.5.3. DEC-MDPs with constraints . . . . .	288

9.5.3.1. Event-driven DEC-MDPs . . . . .	289
9.5.3.2. Opportunity-cost DEC-MDPs . . . . .	289
9.5.3.3. Problem statement . . . . .	289
9.5.3.4. The OC-DEC-MDP model . . . . .	290
9.5.4. Communication and DEC-POMDPs . . . . .	292
9.5.4.1. DEC-POMDPs with communication . . . . .	293
9.5.4.2. Complexity results . . . . .	294
9.5.4.3. Discussion . . . . .	294
9.6. Algorithms for solving DEC-POMDPs . . . . .	295
9.6.1. Optimal algorithms . . . . .	296
9.6.1.1. Dynamic programming for DEC-POMDPs . . . . .	296
9.6.1.2. Heuristic search for DEC-POMDPs . . . . .	299
9.6.1.3. Optimal algorithms for sub-classes of DEC-POMDPs . . . . .	303
9.6.2. Approximate algorithms . . . . .	303
9.6.2.1. Heuristics and approximate dynamic programming . . . . .	304
9.6.2.2. Bounded memory . . . . .	305
9.6.2.3. Co-evolutionary algorithms . . . . .	305
9.6.2.4. Gradient descent for policy search . . . . .	307
9.6.2.5. Bayesian games . . . . .	307
9.6.2.6. Heuristics for communicating agents . . . . .	308
9.6.2.7. Approximate solutions for OC-DEC-MDPs . . . . .	308
9.7. Applicative scenario: multirobot exploration . . . . .	310
9.8. Conclusion and outlook . . . . .	312
9.9. Bibliography . . . . .	313
<b>Chapter 10. Non-Standard Criteria . . . . .</b>	<b>319</b>
Matthieu BOUSSARD, Maroua BOUZID, Abdel-Ilhah MOUADDIB,	
Régis SABBADIN and Paul WENG	
10.1. Introduction . . . . .	319
10.2. Multicriteria approaches . . . . .	320
10.2.1. Multicriteria decision-making . . . . .	321
10.2.2. Multicriteria MDPs . . . . .	322
10.2.2.1. Operators for multicriteria decision-making . . . . .	323
10.3. Robustness in MDPs . . . . .	327
10.4. Possibilistic MDPs . . . . .	329
10.4.1. Possibilistic counterpart of expected utility . . . . .	330
10.4.2. Possibilistic dynamic programming . . . . .	333
10.4.2.1. Finite horizon . . . . .	333
10.4.2.2. Value iteration . . . . .	334
10.4.2.3. Policy iteration . . . . .	338
10.4.3. Extensions of possibilistic MDPs . . . . .	338
10.4.3.1. Possibilistic reinforcement learning . . . . .	339
10.4.3.2. Possibilistic partially observable MDPs . . . . .	340
10.4.3.3. Possibilistic influence diagrams (PID) . . . . .	342

10.5. Algebraic MDPs . . . . .	342
10.5.1. Background . . . . .	343
10.5.1.1. Semirings . . . . .	343
10.5.1.2. Plausibility measures . . . . .	344
10.5.1.3. Generalized expected utility . . . . .	345
10.5.2. Definition of an algebraic MDP . . . . .	345
10.5.3. Value function of a policy . . . . .	347
10.5.4. Conditions . . . . .	348
10.5.5. Examples of AMDPs . . . . .	349
10.5.5.1. Probabilistic multicriteria AMDP . . . . .	349
10.5.5.2. Possibilistic multicriteria AMDPs . . . . .	350
10.5.5.3. AMDPs whose rewards are non-decreasing functions . . . .	351
10.6. Conclusion . . . . .	354
10.7. Bibliography . . . . .	355
<b>PART 3. APPLICATIONS . . . . .</b>	<b>361</b>
<b>Chapter 11. Online Learning for Micro-Object Manipulation . . . . .</b>	<b>363</b>
Guillaume LAURENT	
11.1. Introduction . . . . .	363
11.2. Manipulation device . . . . .	364
11.2.1. Micro-positioning by pushing . . . . .	364
11.2.2. Manipulation device . . . . .	365
11.2.3. Control loop . . . . .	366
11.2.4. Representation of the manipulation task as an MDP . . . . .	366
11.2.4.1. Definition of the state space . . . . .	366
11.2.4.2. Definition of the action space . . . . .	367
11.2.4.3. Definition of the reward function . . . . .	367
11.2.4.4. Definition of an episode . . . . .	367
11.3. Choice of the reinforcement learning algorithm . . . . .	367
11.3.1. Characteristics of the MDP . . . . .	367
11.3.2. A suitable algorithm: <i>STM-Q</i> . . . . .	368
11.4. Experimental results . . . . .	370
11.4.1. Experimental setup . . . . .	370
11.4.2. Results . . . . .	370
11.5. Conclusion . . . . .	373
11.6. Bibliography . . . . .	373
<b>Chapter 12. Conservation of Biodiversity . . . . .</b>	<b>375</b>
Iadine CHADÈS	
12.1. Introduction . . . . .	375
12.2. When to protect, survey or surrender cryptic endangered species . . .	376
12.2.1. Surveying and managing the Sumatran tiger . . . . .	376

12.2.2. The model . . . . .	377
12.2.3. Results . . . . .	377
12.2.4. Extension to more than one population . . . . .	379
12.3. Can sea otters and abalone co-exist? . . . . .	381
12.3.1. Abalone and sea otters: two endangered species . . . . .	381
12.3.2. The models . . . . .	382
12.3.2.1. Population dynamics of abalone . . . . .	382
12.3.2.2. Sea otter population model . . . . .	383
12.3.2.3. States . . . . .	384
12.3.2.4. Decisions . . . . .	385
12.3.2.5. Interaction between sea otters and abalone . . . . .	386
12.3.2.6. Multicriteria objective and reward function . . . . .	386
12.3.3. Methods . . . . .	387
12.3.4. Results . . . . .	387
12.3.4.1. Scenario 1: sea otter reintroduction and anti-poaching enforcement . . . . .	387
12.3.4.2. Scenario 2: control of sea otters . . . . .	389
12.3.4.3. Scenario 3: combined action of sea otter control and anti-poaching . . . . .	389
12.3.5. Conclusion . . . . .	389
12.4. Other applications in conservation biology and discussions . . . . .	391
12.5. Bibliography . . . . .	392
 <b>Chapter 13. Autonomous Helicopter Searching for a Landing Area in an Uncertain Environment . . . . .</b>	 395
Patrick FABIANI and Florent TEICHTEIL-KÖNIGSBUCH	
13.1. Introduction . . . . .	395
13.2. Exploration scenario . . . . .	397
13.2.1. Planning problem . . . . .	398
13.2.2. States and actions . . . . .	399
13.2.3. Uncertainties . . . . .	400
13.2.4. Optimization criterion . . . . .	400
13.2.5. Formalization of the decision problem . . . . .	401
13.3. Embedded control and decision architecture . . . . .	401
13.3.1. Global view . . . . .	401
13.3.2. Multi-thread planning triggered by the supervisor . . . . .	403
13.3.2.1. Policy optimization . . . . .	403
13.3.2.2. Dialogue with the supervisor . . . . .	404
13.4. Incremental stochastic dynamic programming . . . . .	404
13.4.1. Obtaining the initial safe policy quickly . . . . .	405
13.4.2. Generating the sub-space of reachable states . . . . .	405
13.4.3. Local policy optimization . . . . .	406
13.4.4. Launching local replanning processes . . . . .	407



13.5. Flight tests and return on experience . . . . .	407
13.6. Conclusion . . . . .	410
13.7. Bibliography . . . . .	410

## **Chapter 14. Resource Consumption Control for an Autonomous Robot . . 413**

Simon LE GLOANNEC and Abdel-Ilah MOUADDIB

14.1. The rover's mission . . . . .	414
14.2. Progressive processing formalism . . . . .	415
14.3. MDP/PRU model . . . . .	416
14.3.1. States . . . . .	416
14.3.2. Actions . . . . .	417
14.3.3. Transition function . . . . .	418
14.3.4. Reward function . . . . .	418
14.4. Policy calculation . . . . .	418
14.4.1. Value function . . . . .	419
14.4.2. Propagation algorithm . . . . .	419
14.5. How to model a real mission . . . . .	419
14.6. Extensions . . . . .	422
14.7. Conclusion . . . . .	423
14.8. Bibliography . . . . .	423

## **Chapter 15. Operations Planning . . . . . 425**

Sylvie THIÉBAUX and Olivier BUFFET

15.1. Operations planning . . . . .	425
15.1.1. Intuition . . . . .	425
15.1.1.1. Problem features . . . . .	426
15.1.1.2. Plans . . . . .	427
15.1.2. Formal definitions . . . . .	428
15.1.2.1. Planning problem, operations . . . . .	429
15.1.2.2. Execution . . . . .	431
15.1.2.3. Decision epochs, states, plans . . . . .	432
15.1.2.4. Objective . . . . .	432
15.2. MDP value function approaches . . . . .	433
15.2.1. Formalizations, CoMDP . . . . .	433
15.2.1.1. States, actions, transitions . . . . .	433
15.2.1.2. Rewards, costs . . . . .	434
15.2.2. Algorithms . . . . .	435
15.2.2.1. (L)RTDP . . . . .	435
15.2.2.2. Memory management . . . . .	435
15.2.2.3. Reduction of the number of updates . . . . .	436
15.2.2.4. Hybrid algorithms . . . . .	436
15.2.2.5. Algorithms with upper bounds . . . . .	437

15.2.3. Heuristics . . . . .	438
15.2.3.1. Basic heuristics . . . . .	438
15.2.3.2. Heuristics obtained by relaxation of the CoMDP . . . . .	439
15.2.3.3. Planning graph heuristics . . . . .	440
15.3. Reinforcement learning: FPG . . . . .	442
15.3.1. Employing approximate methods . . . . .	442
15.3.2. Parameterized policy . . . . .	443
15.3.2.1. Inputs . . . . .	443
15.3.2.2. Outputs . . . . .	443
15.3.2.3. Function approximator . . . . .	444
15.3.3. Gradient methods . . . . .	445
15.3.3.1. Terminating an execution . . . . .	445
15.3.3.2. Choice of $OL_{pomdp}$ . . . . .	445
15.3.3.3. Optimized criterion . . . . .	445
15.3.4. Improving FPG . . . . .	446
15.4. Experiments . . . . .	446
15.5. Conclusion and outlook . . . . .	448
15.6. Bibliography . . . . .	450
<b>Index . . . . .</b>	<b>453</b>

## Preface

The present book discusses sequential decision-making under uncertainty and reinforcement learning, two classes of problems in artificial intelligence which can be formalized in the framework of Markov decision processes. It has been written for students, engineers and researchers likely to be interested in these fields and models.

The book is organized as follows:

- Part 1 provides an introduction to this domain and to efficient resolution techniques (Markov decision processes, reinforcement learning, approximate representations, factored representations, policy gradients and online resolution).
- Part 2 presents important extensions of Markov decision processes that make it possible to solve more complex sequential decision-making problems (partially observable Markov decision processes, Markov games, multi-agent approaches and non-classical criteria).
- Part 3 completes the book with example applications showing how Markov decision processes can be employed for various problems (micro-object manipulation, biodiversity preservation, high-level control of a helicopter, control of an exploration mission and operations planning).

It was not possible for this book to cover all research directions in this very active field. We give here some references to point the reader to some uncovered aspects. For example, we have decided not to cover continuous time reinforcement learning [MUN 01], relational reinforcement learning [DZE 01], hierarchical reinforcement learning [BAR 03], learning classifier systems [SIG 07] or predictive state representations [LIT 02].

In addition, we endeavor in each chapter to provide the reader with references to related work.

Additional information related to this book (e.g. *errata*) can be found at the following website: <http://www.loria.fr/projets/PDMIA/Book/>.

## Bibliography

- [BAR 03] BARTO A. and MAHADEVAN S., “Recent advances in hierarchical reinforcement learning”, *Discrete Event Dynamic Systems*, vol. 13, no. 4, pp. 341–379, 2003.
- [DZE 01] DZEROSKI S., DE RAEDT L. and DRIESSENS K., “Relational reinforcement learning”, *Machine Learning*, vol. 43, no. 1-2, pp. 7–53, 2001.
- [LIT 02] LITTMAN M., SUTTON R. and SINGH S., “Predictive representations of state”, *Advances in Neural Information Processing Systems 14 (NIPS’01)*, MIT Press, Cambridge, MA, pp. 1555–1561, 2002.
- [MUN 01] MUNOS R. and MOORE A., “Variable resolution discretization in optimal control”, *Machine Learning*, vol. 49, pp. 291–323, 2001.
- [SIG 07] SIGAUD O. and WILSON S. W., “Learning classifier systems: a survey”, *Soft Computing*, vol. 11, no. 11, pp. 1065–1078, 2007.

## List of Authors

### **Aurélie Beynier**

Aurélie Beynier is currently an associate professor of computer science at the LIP6 Laboratory. Her PhD thesis has been defended in November 2006 at the University of Caen.

### **Matthieu Boussard**

Matthieu Boussard holds a PhD in computer science from the University of Caen. He is a member of the MAD Group of the GREYC Laboratory.

### **Maroua Bouzid**

Maroua Bouzid is an associate professor at the University of Caen. She is a member of the MAD Group of the GREYC Laboratory.

### **Olivier Buffet**

Olivier Buffet is an INRIA junior research scientist. He is member of the Autonomous Intelligent Machines (MAIA) Team of the LORIA Laboratory.

### **Andriy Burkov**

Andriy Burkov is a PhD student working under supervision of Professor Brahim Chaib-Draa at Laval University, Canada. His main research interests include multiagent learning and game theory.

### **Iadine Chadès**

Iadine Chadès is currently a research fellow at CSIRO Sustainable Ecosystems, Australia, on leave from the French National Institute for Agricultural Research (INRA).

**Brahim Chaib-Draa**

Professor Brahim Chaib-Draa is the leader of the DAMAS Research Group on Agents and Multiagent Systems at the Computer Science and Software Engineering Department of Laval University, Canada.

**François Charpillet**

François Charpillet is an INRIA senior research scientist. He is the head of the Autonomous Intelligent Machines (MAIA) Team of the LORIA Laboratory.

**Thomas Degris**

Thomas Degris holds a PhD in computer science. After working in the video game industry, he is now a postdoctoral fellow at the University of Alberta.

**Alain Dutech**

Alain Dutech is an INRIA experienced research scientist. He is a member of the Autonomous Intelligent Machines (MAIA) Team of the LORIA Laboratory.

**Patrick Fabiani**

Patrick Fabiani is the director of the Systems Control and Flight Dynamics Department at ONERA. His research interests include models, methods and algorithms for sequential decision making and planning under uncertainty, applied to autonomous aerial robots in the ReSSAC project at ONERA.

**Frédéric Garcia**

Frédéric Garcia is a researcher in artificial intelligence at the Department of Applied Mathematics and Informatics at INRA (the French National Institute for Agricultural Research).

**Guillaume Laurent**

Guillaume Laurent is an associate professor at the ENSMM Graduate School at Besançon. He is a member of the Automatic Control and Micro-Mechatronic Systems Department of the FEMTO-ST Research Institute.

**Simon Le Gloannec**

Simon Le Gloannec holds a PhD from the University of Caen. He has worked with both the MAIA Team (LORIA laboratory) and the MAD Group (GREYC Laboratory). He currently holds a postdoctoral position at GREYC.

**Laëtitia Matignon**

Laëtitia Matignon holds a PhD from the University of Franche-Comté at Besançon. She is now a postdoctoral fellow of the Cooperative Decision-Theoretic Autonomous Agent System Group (MAD) of the GREYC Laboratory in Caen.

**Abdel-Ilah Mouaddib**

Abdel-Ilah Mouaddib is a full professor at the University of Caen. He is the head of the Cooperative Decision-Theoretic Autonomous Agent System Group (MAD) of the GREYC Laboratory.

**Rémi Munos**

Rémi Munos is senior researcher at INRIA Lille, France. He works in the fields of reinforcement learning, optimal control and decision theory.

**Laurent Péret**

Laurent Péret holds a PhD in artificial intelligence, focused on search methods for MDPs. Since 2005, he has been involved in various space programs as a flight dynamics engineer.

**Emmanuel Rachelson**

Emmanuel Rachelson is a postdoctoral fellow at the University of Liège, Belgium, working on reinforcement and statistical learning. He also works with Pr. Lagoudakis in Greece and with the Electricité de France (EDF) Research Center in Paris.

**Régis Sabbadin**

Régis Sabbadin has been a research scientist at INRA-Toulouse since 1999. His research focuses on planning under uncertainty, applied to natural resources management and agriculture.

**Bruno Scherrer**

Bruno Scherrer is an INRIA experienced research scientist. He is member of the Autonomous Intelligent Machines (MAIA) Team of the LORIA Laboratory.

**Olivier Sigaud**

Olivier Sigaud is a professor of computer science at the UPMC-Paris 6 University. He is the head of the “Motion” Group at the Institute of Intelligent Systems and Robotics (ISIR).

**Daniel Szer**

Daniel Szer obtained a Master's degree in computer science from UMass Dartmouth and a PhD in artificial intelligence from Henri-Poincaré University, Nancy. He works as an IT analyst in Paris.

**Florent Teichteil-Königsbuch**

Florent Teichteil-Königsbuch is a researcher in probabilistic sequential decision-making at ONERA. He got a PhD in artificial intelligence from SUPAERO in 2005.

**Sylvie Thiébaux**

Sylvie Thiébaux is an associate professor at the Australian National University and principal researcher at the National ICT Australia, specializing in automated planning in artificial intelligence.

**Paul Weng**

Paul Weng has been an associate professor at Paris 6 University since September 2007. Before his PhD obtained in 2006, he worked in finance in London.



Part 1

## MDPs: Models and Methods



# Chapter 1

## Markov Decision Processes

### 1.1. Introduction

This book presents a decision problem type commonly called *sequential decision problems under uncertainty*. The first feature of such problems resides in the relation between the current decision and future decisions. Indeed, these problems do not consist of one, but several decision problems, presented in a sequence. At each step of this sequence, the *agent* (actor or decision-maker) needs to decide on the current action by taking into account its effect on the solution of future problems. This sequential feature is also typical of *planning* problems in artificial intelligence and is often linked with shortest path methods in graph theory. The second characteristic of the problems discussed in these pages is the uncertainty in the consequences of all available decisions (actions). Knowledge of its decision's effects is not available in advance to the agent in a deterministic form. As such, this problem deals with the various theories of decision under uncertainty which suggest different formalization and resolution approaches. Among these approaches, we need to mention specifically the standard theory of expected utility maximization.

Consequently, problems of sequential decision under uncertainty couple the two problematics of sequential decision and decision under uncertainty. *Markov decision problems* (MDPs) are a general mathematical formalism for representing shortest path problems in stochastic environments. This formalism is based on the theory of *Markov decision processes* (also written as MDPs). A Markov decision process relies on the notions of *state*, describing the current situation of the agent, *action* (or decision), affecting the dynamics of the process, and *reward*, observed for each

transition between states. Such a process describes the probability of triggering a transition to state  $s'$  and receiving a certain reward  $r$  when taking decision  $a$  in state  $s$ . Hence, an MDP can be described as a controlled Markov chain, where the control is given at each step by the chosen action. The process then visits a sequence of states and can be evaluated through the observed rewards. Solving an MDP consists of controlling the agent in order to reach an optimal behavior, i.e. to maximize its overall revenue. Because action effects are stochastic and, thus, can result in different possible states at the next stage of the decision process, the optimal control strategy cannot necessarily be represented as a single sequence of actions.<sup>1</sup> Consequently, solutions of an MDP are usually given as *universal plans* or *policies* (*strategies* or *decision rules*) specifying which action to undertake at each step of the decision process and for every possible state reached by the agent. Due to the uncertainty in actions' results, applying a given policy can result in different sequences of states/actions.

**EXAMPLE 1.1.** Let us illustrate these concepts with a simple car maintenance example. According to the current state of the car (breakdown, wear, age, etc.), an agent wishes to decide which is its best strategy (do nothing, replace parts preventively, repair, change car, etc.) in order to minimize the maintenance cost in the long run. Assuming the agent knows the consequences and the cost of each separate action in every possible state (e.g. we know the failure probability of an engine if the oil leak is not fixed), we can model this problem as an MDP. Solving this MDP will provide the agent with a policy indicating which is the optimal action to undertake in every state of the problem. This way, the sequence of actions performed as the car's state changes will allow the agent to always minimize the expected maintenance cost.

The theory of Markov decision processes and its generalizations will be developed in the next chapters. These models have become the most popular framework for representing and solving problems of sequential decision under uncertainty. This chapter presents the basics of MDP theory and optimization, in the case of an agent having a perfect knowledge of the decision process and of its state at every time step, when the agent's goal is to maximize its global revenue over time.

## 1.2. Markov decision problems

### 1.2.1. Markov decision processes

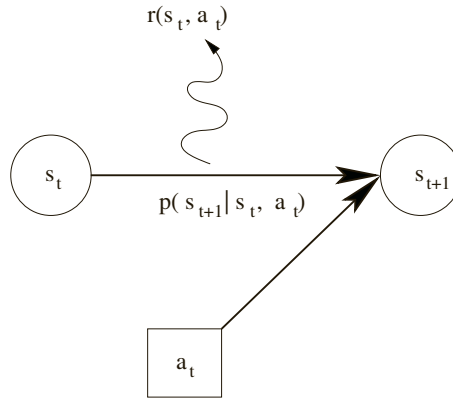
Markov decision processes are defined as *controlled stochastic processes* satisfying the Markov property and assigning reward values to state transitions [BER 87, PUT 94]. Formally, they are described by the 5-tuple  $(S, A, T, p, r)$  where:

---

1. Contrary to the deterministic approaches of classical planning.

- $S$  is the state space in which the process' evolution takes place;
- $A$  is the set of all possible actions which control the state dynamics;
- $T$  is the set of time steps where decisions need to be made;
- $p()$  denotes the state transition probability function;
- $r()$  provides the reward function defined on state transitions.

Figure 1.1 represents an MDP, drawn as an influence diagram. At every time step  $t$  in  $T$ , action  $a_t$  is applied in the current state  $s_t$ , affecting the process in its transition to the next state  $s_{t+1}$ . Reward  $r_t$  is then obtained for this transition.



**Figure 1.1.** Markov decision process

The set  $T$  of decision epochs is a discrete set, subset of  $\mathbb{N}$ , which can either be finite or infinite (then we talk, respectively, about finite horizon or infinite horizon). A third case corresponds to the existence of a set of terminal states (or goal states). In this case, the process stops as soon as one of these states is encountered. Then, the horizon is then said to be indefinite. These problems are often related to stochastic shortest path problems. This case, however, can be seen as a specific case of infinite horizon MDPs with absorbing states and will not be presented in detail in this chapter (see Chapter 6, section 6.2.3 and Chapter 15).

In the most general case, the  $S$  and  $A$  sets are supposed finite, even though many results can be extended to countable or even continuous sets (see [BER 95] for an introduction to the continuous case). Generally, the set  $A$  of applicable actions can also depend on the current state: we define a subset  $A_s$  of applicable actions in state  $s$ . Similarly,  $S$  and  $A$  can change based on the time step  $t$  ( $S_t$  and  $A_t$ ). However, in this chapter, for clarity of presentation, we will restrict ourselves to the standard case where  $A$  and  $S$  are constant throughout the process.

The transition probabilities  $p()$  characterize the state dynamics of the system, i.e. indicate which states are likely to appear after the current state. For a given action  $a$ ,  $p(s' \mid s, a)$  represents the probability for the system to transit to state  $s'$  after undertaking action  $a$  in state  $s$ . Because the  $p()$  values are probabilities, we classically have  $\forall s, a, \sum_{s'} p(s' \mid s, a) = 1$ . This  $p()$  function is usually represented in matrix form, where we write  $P_a$  the  $|S| \times |S|$  matrix containing elements  $\forall s, s' P_{a,s,s'} = p(s' \mid s, a)$ . Consequently, the probability transitions of the decision process are given as  $|A|$  matrices  $P_a$ . Since each line of these matrices sums to one, the  $P_a$  are said to be *stochastic* matrices.

The  $p()$  probability distributions over the next state  $s'$  follow the fundamental property which gives their name to Markov decision processes. If we write  $h_t = (s_0, a_0, \dots, s_{t-1}, a_{t-1}, s_t)$  the history of states and actions until time step  $t$ , then the probability of reaching state  $s_{t+1}$  consecutively to action  $a_t$  is only a function of  $a_t$  and  $s_t$ , and not of the entire history  $h_t$ . Let us write  $P(x \mid y)$  the conditional probability of event  $x$ , provided that  $y$  is true, then we have

$$\forall h_t, a_t, s_{t+1} \quad P(s_{t+1} \mid h_t, a_t) = P(s_{t+1} \mid s_t, a_t) = p(s_{t+1} \mid s_t, a_t).$$

We should note here that the previous condition does not necessarily imply that the resulting stochastic process  $(s_t)_{t \in T}$  itself respects the Markov property: this also depends on the action choice policy for  $a_t$ .

As a result of choosing action  $a$ , in state  $s$ , at time  $t$ , the deciding agent receives a reward  $r_t = r(s, a) \in \mathbb{R}$ . We can consider positive values of  $r_t$  as gains and negative values as costs. We also sometimes use a cost function  $c()$  instead of the reward function  $r()$ . This reward can be received instantaneously at time  $t$  or accumulated between  $t$  and  $t + 1$ . The important feature is that this reward only depends on the simple input of the current state  $s$  and the current action  $a$ . A vector representation of the  $r(s, a)$  reward function consists of  $|A|$  vectors  $r_a$  of length  $|S|$ .

A common extension consists of considering random rewards. In this case, we will use their average value for the reward function  $r(s, a) = \bar{r}(s, a)$ . In particular, the reward obtained at time step  $t$  can depend on the final state  $s'$  of the transition. We then have a reward specified as  $r(s, a, s')$ . The value used for the reward vectors is  $\bar{r}(s, a) = \sum_{s'} p(s' \mid s, a) r(s, a, s')$ . In all cases,  $r_t$  is supposed bounded.

Finally, as for  $S$  and  $A$ , the transition and reward functions can vary across time. In this case they are written, respectively, as  $p_t$  and  $r_t$ . When these functions do not change from one step to the other, the process is said to be *stationary*:  $\forall t \in T$ ,  $p_t() = p()$ ,  $r_t() = r()$ . In the rest of this chapter, we will keep this stationarity hypothesis in the study of infinite horizon MDPs.