

Generalized Models of Sequential Decision-Making under Uncertainty

Neustroev, G.

DOI

[10.4233/uuid:cdca9bf1-3e6b-4bfc-9d9d-b5acdd3f900d](https://doi.org/10.4233/uuid:cdca9bf1-3e6b-4bfc-9d9d-b5acdd3f900d)

Publication date

2022

Document Version

Final published version

Citation (APA)

Neustroev, G. (2022). *Generalized Models of Sequential Decision-Making under Uncertainty*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:cdca9bf1-3e6b-4bfc-9d9d-b5acdd3f900d>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Generalized Models of Sequential Decision-Making under Uncertainty

Grigory Neustroev



Generalized Models of Sequential Decision-Making under Uncertainty

Generalized Models of Sequential Decision-Making under Uncertainty

⌞ Dissertation ⌞

for the purpose of obtaining the degree of doctor
at Delft University of Technology

by the authority of the Rector Magnificus
prof.dr.ir. T.H.J.J. van der HAGEN,
Chair of the Board for Doctorates

to be defended publicly on
Monday 5 December 2022
at 15:00 o'clock

by

Grigory NEUSTROEV

mathematical economist
in mathematical methods in economics,
Udmurt State University, the Russian Federation
born in Ustinov, the Union of Soviet Socialist Republics

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus	<i>chairperson</i>
Prof. dr. M. M. de WEERDT	Delft University of Technology, <i>promotor</i>
Dr. ir. R. A. VERZIJLBERGH	Delft University of Technology, <i>copromotor</i>
Independent members:	
Prof. dr. ir. K. I. AARDAL	Delft University of Technology
Prof. dr. A. NOWÉ	Vrije Universiteit Brussel, Belgium
Dr. H. C. van HOOFF	University of Amsterdam, the Netherlands
Dr. M. KAISERS	DeepMind, France
Dr. M. T. J. SPAAN	Delft University of Technology
Prof. dr. ir. B. H. K. DE SCHUTTER	Delft University of Technology, <i>reserve member</i>

This research was financed by the Dutch Research Council (Nederlandse Organisatie voor Wetenschappelijk Onderzoek, NWO).



Keywords: sequential decision-making under uncertainty, optimization, Markov decision processes, planning, linear programming, duality, reinforcement learning, optimistic learning

Printed by: IPSKAMP printing

Cover: a color-adjusted photograph of petrol on water by Sofia NEUSTROEVA

Icons by: Zlatko NAJDENOVSKI via PixelBuddha

© G. NEUSTROEV 2022

ISBN 978-94-6366-624-4

An electronic version of this dissertation is available at <https://repository.tudelft.nl/>.

in loving memory of
Capitalina

Contents

Nomenclature	xi
Summary	xxi
Samenvatting	xxv
Автореферат	xxix
1 Introduction	1
1.1 The Decision-Making Problem	3
1.2 Planning & Reinforcement Learning	4
1.3 Examples	5
1.3.1 Frozen Lake	5
1.3.2 Inventory Management.	6
1.3.3 Active Wake Control	8
1.4 Existing Research	9
1.4.1 Planning with Markov Decision Processes.	9
1.4.2 Reinforcement Learning	12
1.4.3 Knowledge Gaps	14
1.5 Content of This Thesis	15
1.5.1 Research Questions.	15
1.5.2 Contributions of This Thesis.	15
2 A Mathematical Model of Decision-Making	17
2.1 Markov Decision Processes.	19
2.1.1 Sequentiality	20
2.1.2 Uncertainty	25
2.1.3 Optimal Behavior	33
2.2 The Existence of Optimal Policies	36
2.2.1 Limitations of the Optimality Criterion	36
2.2.2 Models with Uniformly Bounded Rewards.	38

2.3	Finding Optimal Policies	40
2.3.1	State Value Functions.	41
2.3.2	The Bellman Operators.	42
2.3.3	Occupancy Measure	45
2.3.4	Linear-Programming Formulation	48
2.4	Countably-Infinite Problems	50
2.4.1	Ill-Defined Values	50
2.4.2	Infinitely-Many Permitted Actions	51
2.4.3	Unbounded Rewards	52
2.4.4	Linear-Programming Formulation	55
2.4.5	Inventory Management (Revisited)	58
2.5	Conclusion	61
3	The Infinite-Horizon Non-Stationary Model	63
3.1	Introduction.	65
3.1.1	Truncations and Solution Horizons	65
3.1.2	Previous Work	66
3.2	Model Assumptions.	68
3.3	The Dual Formulation	69
3.3.1	Problem Truncation	71
3.4	A Stopping Rule.	73
3.4.1	Truncations with Variable Salvage Vector.	73
3.4.2	Unbounded Rewards	76
3.4.3	The Algorithm	78
3.5	Experiments.	79
3.6	Conclusion	83
4	The Countably-Infinite Model	85
4.1	Introduction.	87
4.1.1	Previous Work	89
4.2	Model Assumptions.	90
4.3	A Motivating Example	92
4.4	Policy Evaluation	93
4.4.1	Truncated Bellman Operator.	94
4.4.2	Fixed Point of the Truncated Operator	96
4.4.3	Additional Notation.	96
4.4.4	Truncation Errors.	98
4.5	Policy improvement.	100
4.5.1	Pivoting and advantages	100
4.5.2	Advantage Approximation	102

4.6	The Algorithm	106
4.7	Proofs	108
4.7.1	Theorem 4.2	108
4.7.2	Theorem 4.6	110
4.7.3	Theorem 4.8	111
4.7.4	Theorem 4.9	114
4.8	Experiments.	118
4.9	Conclusion	119
5	Generalized Optimistic Q-Learning	121
5.1	Introduction.	123
5.2	Preliminaries	125
5.2.1	Non-Stationary Episodic MDPS.	125
5.2.2	Reinforcement Learning	126
5.3	Optimism in Q-Learning	128
5.3.1	Representation of Optimism	128
5.3.2	Generalized Optimistic Q-Learning	130
5.3.3	The Total Regret Bound	135
5.4	Proof of Theorem 5.1	137
5.4.1	Properties of the Learning Rate	137
5.4.2	Bounds on Q-Value Differences	138
5.4.3	Properties of the Total Regret	140
5.5	Designing a New Optimistic Algorithm	142
5.6	Experiments.	145
5.6.1	Equipment Replacement	145
5.6.2	Frozen Lake	146
5.7	Conclusion	147
6	Reinforcement Learning for Active Wake Control	149
6.1	Introduction.	151
6.2	Preliminaries	152
6.2.1	Steady-State Wind Models	152
6.2.2	Deep Reinforcement Learning	155
6.2.3	RL for Active Wake Control	155
6.3	Active Wake Control as a RL Problem	156
6.3.1	State Space	157
6.3.2	Action Space	158
6.3.3	Rewards	160
6.3.4	Transitions	160
6.3.5	Gym Implementation	162

6.4	Experiments	163
6.4.1	Action Representations	163
6.4.2	Noisy Observations	166
6.5	Conclusion	167
7	Discussion	169
7.1	Answers to the Research Questions	171
7.2	Societal Implications	175
7.3	Future Research Directions	176
7.3.1	Theoretical and Algorithmic Extensions	176
7.3.2	Future Prospects	178
	References	181
	Acknowledgements	199
	Curriculum Vitæ	201
	List of Publications	203
	Appendices	205
A	Miscellaneous Proofs	207
A.1	Proofs of Time Augmentation Equivalence	207
A.2	Proof that Flow Conservation Induces Policies	209
A.3	Proof of Feasible Region Embedding	212
A.4	Proofs for Inventory Management Problem	213
B	Active Wake Control Implementation Details	221



Nomenclature

Abbreviations and acronyms

a.e.	almost everywhere, p. 26
AI	artificial intelligence
API	application programming interface
a.s.	almost surely, p. 26
ASPIRE	approximate salvage-based policy iteration with repeated elimination (of actions), p. 88
cf.	compare to; from Latin <i>conferatur</i>
CPU	central processing unit
DALES	<i>Dutch Atmospheric Large-Eddy Simulation</i> framework, p. 154
DDPG	deep deterministic policy gradient, p. 13
FLORIS	<i>Flow Redirection and Induction in Steady-State</i> wake modeling framework, p. 152
GPRL	Gaussian-processes reinforcement learning, p. 156
GPU	graphics processing unit
GRASP	<i>GPU-Resident Atmospheric Simulation Platform</i> , p. 154
HKNB	<i>Hollandse Kust Noord (site B)</i> dataset, p. 160
ibid.	same source; from Latin <i>ibidem</i> for <i>in the same place</i>
LES	large eddy simulation, p. 154
MDP	Markov decision process, p. 19
MISHA	multi-stage iterated solution horizon algorithm, p. 78
OPIQ	optimistic pessimistically-initialized Q-learning, p. 13
PPO	proximal policy optimization, p. 13
p.w.	pointwise
RL	reinforcement learning, p. 4
SAC	soft actor-critic, p. 14
SOWFA	<i>Simulator for On/Off-Shore Wind Farm Applications</i> , p. 154
TD3	twin delayed deep deterministic policy gradient, p. 13
TRPO	trust region policy optimization, p. 13
UCB	upper confidence bound, p. 13
UCB-B	UCB Q-learning with Bernstein-style bonus, p. 13

UCB-H ⁺	UCB-H with generalized learning rate, p. 131
UCB-H	UCB Q-learning with Hoeffding-style bonus, p. 13
∞ -UCB	infinite-horizon UCB Q-learning, p. 13

Marginalia

	main text continues
	formula explanation

Font faces

These font faces are used to distinguish objects of different classes, with a few exceptions of well-established notation. E.g., the spaces of absolutely Lebesgue-integrable measurable functions are denoted as $L^1(\cdot)$ and not $\mathbb{L}^1(\cdot)$.	FLORIS, numpy, StarCraft II, ... software
	E, Pr, supp, ... common mathematical operators
	A, a, B, b, C, c, \dots variables, constants, and functions
	$\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$ operators (functions that act on functions)
	A, B, C, ... matrices
	a, b, c, ... vectors
	$\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \dots$ tuples (ordered collections)
	$\mathbb{A}, \mathbb{B}, \mathbb{C}, \dots$ sets (unordered collections), including spaces
	$\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$ metaset (sets of sets) and distributions

Latin letters

	A	realized action, p. 20
	A	swept area of the rotor, p. 160
When surrounded by delimiters such as brackets, an argument position is denoted by an interunct (\cdot) ;	$A(\cdot)$	action coordinate map, p. 24
otherwise, a dashed square \square is used.	a	possible action, p. 22
	a	action vector, p. 58
	\mathbb{A}	action space, p. 22
	$A_p(\cdot)$	action permissibility function, p. 22
	$ax(\cdot)$	axial induction, p. 160
	B	total effect of the bonuses, p. 135
	$b(\cdot)$	confidence bonus, p. 129
	b	reward bonus function, p. 95
	b	bonus vector, p. 97
	\mathbb{B}	arbitrary subset
	$\mathcal{B}(\cdot)$	Borel σ -algebra, p. 25
	$\mathcal{C}\square$	constraint operator, p. 56
	c	vector of product prices, p. 59
	C_G	expected revenue when the inventory is infinite, p. 60
	C_H	maximum cost of holding an order, p. 60
	C_O	maximum cost of placing an order and holding it, p. 60

D	rotor diameter, p. 166
D	total demand for a given product, p. 58
d	expected demand for a given product, p. 59
d	metric, p. 43
\mathbf{d}	demand vector, p. 60
\mathbb{D}	deterministic stationary policies, Π_{DS} , p. 31
$\text{dom} \square$	domain
E	total effect of the estimation error, p. 136
$E[\cdot]$	expected value, p. 27
$e(\cdot)$	truncation error function, p. 98
\mathbf{e}	truncation error vector, p. 98
$e_\eta(\cdot, \cdot)$	absolute advantage error, p. 112
\mathbb{F}	random event in the sample space Ω , p. 25
\mathcal{F}	σ -algebra of events, p. 24
$g(\cdot)$	expected sales, p. 59
$G(\cdot, \cdot)$	expected revenue, p. 59
H	episode length, p. 125
h	time step in episodic MDPs, p. 125
$H(\cdot, \cdot)$	holding cost, p. 59
\mathbf{h}	vector of advantage bounds, p. 71
\mathbf{h}	vector of holding costs, p. 60
\mathfrak{h}	history, p. 23
$\mathfrak{h}(\cdot)$	history map, p. 24
\mathbb{H}	time space of an episode, p. 125
$H_n^{(r)}$	n -th generalized harmonic number of order r , p. 144
$\mathcal{I} \square$	the identity operator, p. 96
\mathbf{I}	identity matrix
\mathbb{I}_\square	indicator of an event
$J(\cdot)$	gain, p. 34
\mathbf{j}_α	the vector indicating the candidate optimal initial control, p. 75
J_{D}	dual solution, p. 49
J_{P}	primal solution, p. 49
K	number of episodes, p. 125
k	episode number, p. 126
\mathbf{K}	auxiliary matrix, p. 104
\mathbb{K}	space of episodes, p. 125
$\mathcal{L} \square$	Bellman operator, p. 42
\mathbf{L}	auxiliary matrix, p. 104
$\liminf \square$	limit inferior
$L^P(\cdot)$	space $L^P(\cdot, \#)$ under the counting measure $\#$, p. 44

$L^p(\cdot, \cdot)$	space of measurable functions with finite p -norm $\ \cdot\ _p$, p. 44
$L^w(\cdot)$	space of functions with finite weighted supremum norm, p. 55
$L^\infty(\cdot)$	space of uniformly bounded measurable functions, p. 44
M	maximum shipment size, p. 58
m	measure of a unit of a given product, p. 58
M	wind speed, p. 158
\mathcal{M}	maximization operator, p. 43
\mathbf{m}	vector of unit measurements, p. 58
\mathbf{m}	mean of a multivariate process, p. 161
\mathfrak{M}	Markov decision process, p. 19
N	reward-clipping bound, p. 79
n	number of products, p. 58
n	number of turbines in a wind farm, p. 158
\mathcal{N}	extension operator, p. 49
\mathbf{N}	extension matrix, p. 69
\mathbb{N}_0	non-negative integers, $\{0\} \cup \mathbb{N}$
\mathbb{N}	natural numbers
$\mathcal{N}(\cdot, \cdot)$	normal distribution, p. 166
$O(\cdot)$	ordering cost, p. 59
$\mathcal{O}(\cdot)$	BIG-O of the Bachmann–Landau notation
$\mathfrak{o}(\cdot)$	small-o of the Bachmann–Landau notation
o_f	fixed ordering cost, p. 60
\mathbf{o}_v	vector of variable ordering costs, p. 60
P	power output, p. 160
$P(\cdot)$	probability measure, p. 26
$p(\cdot \cdot)$	transition kernel (under a given policy), p. 31
$p(\cdot \cdot, \cdot)$	transition kernel, p. 28
$\mathcal{P}(\cdot)$	Poisson distribution, p. 118
$p_d(\cdot)$	probability mass of the total demand, p. 58
p_f	probability to follow, p. 28
p_p	power exponent, p. 160
$\Pr[\cdot]$	probability of an event
p_s	probability to slip, p. 29
$\mathcal{P}_{s,a}$	pivoting operator, p. 101
$q(\cdot, \cdot)$	Q-value function, p. 125
\mathcal{Q}	the value-producing operator, p. 96
\mathbf{Q}	value-producing matrix, p. 97
$q_d(\cdot)$	probability that demand reaches a given level, p. 59
R	received reward, p. 20

R	regret of an episode, p. 126
R	total regret of learning, p. 126
$r(\cdot)$	expected reward (under a given policy), p. 31
$r(\cdot, \cdot)$	expected reward, p. 29
$r(\cdot, \cdot, \cdot)$	deterministic reward function, p. 29
$r(\cdot \cdot, \cdot, \cdot)$	reward kernel, p. 29
\mathcal{R}	operator for the expected transitional change, p. 103
\mathbf{R}	matrix form of the operator \mathcal{R} , p. 104
\mathbf{r}	reward vector, p. 69
\mathbb{R}	real numbers
$\bar{\mathbb{R}}$	extended real line, $\mathbb{R} \cup \{-\infty, +\infty\}$
\mathbb{R}_+	non-negative real numbers
S	realized state, p. 20
$S(\cdot)$	state coordinate map, p. 24
s	possible state, p. 21
\mathbf{S}	diffusion matrix, p. 161
\mathbf{s}	state vector, p. 58
\mathcal{S}	state space, p. 21
sgn	signum
supp	support, p. 30
T	time horizon, p. 20
T	truncation horizon, p. 72
t	decision epoch, time step, p. 20
\mathcal{T}	transition operator, p. 42
\mathbf{T}	transition matrix, p. 69
\mathbb{T}	time space, p. 20
$u(\cdot)$	bonus for optimism, p. 128
$u(\cdot)$	salvage function, p. 72
$u(\cdot)$	value-bounding function, p. 91
$U(\cdot, \cdot, \cdot)$	update of Q-learning, p. 127
\mathbf{u}	salvage vector, p. 72
\mathbf{u}	vector of stock and order, p. 59
\mathbb{U}	salvage space, p. 74
$\mathcal{U}_d(\cdot, \cdot)$	discrete uniform distribution, p. 118
$\nu(\cdot)$	dual variable, p. 49
$\nu(\cdot)$	value, p. 41
\mathbf{v}	vector of dual variables, value vector, p. 69
w	absolute reward bound, p. 38
$w(\cdot)$	weight function, p. 52
\mathbf{W}	multivariate Wiener process, p. 161
\mathbf{w}	vector of weights, p. 69

$W_k(\cdot)$	k -th branch of the Lambert w -function, p. 61
X	size of the admissible control space, p. 135
x	admissible control pair $x = (s, a)$, p. 23
\mathbb{X}	admissible control space, p. 23
$\mathbb{X}_{\mathbb{B}}$	admissible control space restricted to states from a subspace \mathbb{B} , p. 103
$Y(\cdot)$	auxiliary function, p. 139
y	arbitrary element
\mathcal{Y}	arbitrary operator
\mathbf{y}	arbitrary vector
\mathbb{Y}	arbitrary set
$z(\cdot)$	state occupancy, p. 47
$z(\cdot)$	primal variable, p. 48
$z(\cdot, \cdot)$	occupancy measure, p. 45
$z(\cdot, \cdot)$	visitation, occupancy function, p. 46
\mathcal{Z}	policy-producing operator, p. 47
\mathbf{z}	vector of primal variables, occupancy vector, p. 69

Greek letters

$\alpha(\cdot)$	initial state distribution, p. 28
$\alpha(\cdot, \cdot)$	learning rate, p. 127
α	vector of initial distribution probabilities, p. 69
$\beta(\cdot)$	cumulative confidence bonus, p. 129
β	static wind direction, p. 159
γ	turbine yaw, p. 157
γ	discounting factor, p. 33
Δ	relative reward decrease per degradation level, p. 79
δ	arbitrary small probability for PAC-bounds, p. 129
$\delta_{\square\square}$	Kronecker delta, p. 39
Δ_h	auxiliary constant, p. 139
Δt	time increment, p. 157
ε	Gaussian noise, p. 166
ε	exploration rate, p. 131
$\zeta(\cdot)$	asymptotic of the squared learning rate, p. 133
$\eta(\cdot, \cdot)$	asymptotic of the residual learning rate, p. 133
$\eta(\cdot, \cdot)$	advantage, reduced cost, p. 70
$\boldsymbol{\eta}$	advantage vector, p. 70
θ	drift coefficient, p. 164
$\vartheta(\cdot)$	total cumulative bonus, p. 129
$\theta(\cdot)$	bonus scaling function, p. 135

Θ	drift matrix, p. 161
ι	logarithmic term of the regret, p. 136
κ	one-stage expansion coefficient, p. 52
	Lipschitz constant of an operator, p. 43
λ	intensity of a Poisson distribution $\mathcal{P}(\cdot)$, p. 118
λ	multi-stage contraction coefficient, p. 52
λ	intensity vector, p. 118
μ	value magnitude, p. 52
$\mu(\cdot)$	measure, p. 26
$\mu(\cdot, \cdot, \cdot)$	magnitude function, p. 134
ν	contraction horizon, p. 52
ξ	auxiliary constant, p. 140
π	policy, p. 31
$\pi(\cdot)$	deterministic decision rule, p. 31
$\pi(\cdot \cdot)$	stochastic decision rule, p. 30
Π	all (randomized history-dependent) policies, Π_{RH} , p. 31
Π_{D}	deterministic policies, p. 31
Π_{H}	history-dependent policies, p. 31
Π_{M}	Markovian policies, p. 31
Π_{R}	randomized policies, p. 31
Π_{S}	stationary policies, p. 31
$\rho(\cdot)$	auxiliary function, p. 141
ρ	air density, p. 160
ρ	starting reward, p. 79
σ	standard deviation, p. 166
σ	spectral radius, p. 82
Σ	covariance matrix, p. 162
τ	turbulence intensity, p. 161
ϕ	wind direction, p. 158
Φ	growth matrix, p. 82
ψ	auxiliary constant, p. 140
ψ	deterioration probability, p. 79
$\Psi(\cdot)$	auxiliary function, p. 140
$\Psi(\cdot)$	utility, p. 33
ω	exponent coefficient of the learning rate, p. 130
ω	angular velocity, p. 159
ω	sample path, p. 24
Ω	sample space, p. 23

Superscripts and diacritics

$\tilde{\square}$	optimistically augmented, p. 128
$\check{\square}$	augmented, p. 95
$\dot{\square}$	absorbing-state-augmented, p. 53
\square^b	binary version of a variable, p. 75
\square^e	empirical, that is, observation-based, p. 127
\square^t	time-augmented, p. 39
$\square^{\mathbb{B}}$	restricted to the subspace \mathbb{B} , p. 94
\square^c	complement, p. 91
\square^j	j -step, p. 32
\square^T	transpose
$\square^{\mathbb{U}}$	with salvage space \mathbb{U} , p. 102
\square^+	positive part, p. 27
\square^-	negative part, p. 27
\square', \square''	another element (usually state or action)

Subscripts

\square_H	with episode length H , p. 125
\square_h	at time step h in an episodic MDP, p. 125
\square_i	of i -th product, p. 58
\square_K	lasting for K episodes, p. 125
\square_P	with respect to probability measure P , p. 27
\square_r	rotated, p. 162
\square_T	with truncation horizon T , p. 72
\square_T	with horizon T , p. 19
\square_t	at time step t , p. 20
\square_u	with salvage vector u , p. 72
\square_π	under policy π , p. 31
\square_+	upper bound, p. 21
\square_-	lower bound, p. 21
\square_∞	infinite-horizon, p. 24
\square_∞	limiting (for monotone-increasing sequences), p. 91
\square_*	adjoint (for operators), p. 49
\square_*	dual (for spaces), p. 55
\square_\star	optimal, p. 35
\square_\uparrow	asymptotically dominant, p. 134

Numbers and other symbols

0	zero vector
\emptyset	empty set
1	unit vector
2^{\square}	power set, p. 22
\aleph_0	the cardinality of \mathbb{N}
$\langle \cdot, \cdot \rangle$	bilinear form (for functions), p. 55
$\langle \cdot, \cdot \rangle$	dot product (for vectors), $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\top \mathbf{y}$
$\langle \cdot, \cdot \rangle_{\square}$	inner product of functions, p. 49
$\lceil \cdot \rceil$	the ceiling function
$\lfloor \cdot \rfloor$	the floor function
$\ \cdot \ $	norm
$\ \cdot \ _p$	p -norm, p. 44
$\ \cdot \ _{p \rightarrow q}$	operator norm of an operator acting from a p -normed space into a q -normed one, p. 82
$\ \cdot \ _w$	w -weighted supremum norm, p. 53
$\ \cdot \ _{\infty}$	supremum norm, p. 44
$\square \xrightarrow{\text{p.w.}} \square$	converges p.w. (pointwise) to
$\square \triangleq \square$	is defined as
$\square \odot \square$	Hadamard (elementwise) multiplication
$\square \oslash \square$	Hadamard (elementwise) division
$\square \sqcup \square$	disjoint union
$\square \sim \square$	is distributed as
\diamond	absorbing state, p. 53
$\#(\cdot, \cdot)$	realized visitation function, p. 127
$\#(\cdot)$	counting measure, p. 44
\triangle_{\square}	probability simplex, p. 28
\triangle_{\square}	set of probability measures, p. 28

Summary

SEQUENTIAL DECISION-MAKING under uncertainty is an important branch of artificial intelligence research with a plethora of real-life applications. In this thesis, we generalize two fundamental properties of the decision-making process. First, we show that the theory on planning methods for finite spaces can be extended to infinite but countable spaces. Second, we propose a unified model of reinforcement learning algorithms that employ the principle of optimism in the face of uncertainty. This model is used to explain why these methods are efficient. We use the developed theory to design novel algorithms. Depending on the user's needs, these algorithms can either automate the decision-making process completely, or provide advice in decision-support systems.

We start with presenting the basic concepts from the theory of decision-making and discuss the two approaches to it: planning and reinforcement learning. We look at a few typical sequential decision-making problems of increasing difficulty. In particular, we present a game that involves grid navigation and the problems of warehouse management and wind farm operation. Next, we survey the state-of-the-art methods for solving such problems.

Chapter 1, p. 1.

Based on this analysis, we identify the following research opportunities. In planning, models with non-stationary and countably-infinite data remain relatively untreated because they are equivalent to infinitely-dimensional optimization problems, which are notoriously difficult to solve even approximately. In reinforcement learning, optimistic approaches lead to computational efficiency, yet the theory of optimism remains undeveloped. Moreover, while reinforcement learning shines at playing games, such as chess, shōgi, Go, and StarCraft II, its practical applications remain few.

Next, we overview a mathematical framework of sequential decision-making under uncertainty known as the Markov decision process. We explain how the goal of the decision-maker can be expressed as an optimization problem and present two approaches to achieving this goal. The first—more common—approach assigns so-called values to different actions. The other approach uses

Chapter 2, p. 17.

so-called occupancies that tell how often the agent should choose the actions instead of evaluating how good these actions are. In fact, the two approaches are known to be dual to each other. While this duality is well studied in the finite case, the infinite case is less explored. To address this knowledge gap, we present a new dual formulation for countable problems, both finite and infinite.

Chapter 3, p. 63.

Afterwards, we use the dual formulation to design a new planning algorithm for infinite-horizon problems with non-stationary data. These problems are essentially infinite-dimensional optimization problems and as such are impossible to solve exactly using the standard approaches. We show that they can be solved by changing what is defined as optimal behavior: instead of seeking universally optimal policies, we consider initial-decision-optimal ones. Instead of planning all of the actions beforehand, these policies can be used to plan given the currently observed data. When the next decision is required, the process can be repeated in the same manner, leading to an optimal decision-making strategy. Our approach uses the occupancy-value duality to rule out suboptimal actions based on so-called truncations: finite-time approximations of the infinite-horizon decision-making problem.

Chapter 4, p. 85.

We extend the truncation approach to a more general setting of decision-making problems with countably-infinite state spaces. Instead of time-based truncations, we consider state-based ones. This allows us to limit the amount of data required to make the decisions and to design an algorithm for a class of problems that are otherwise unsolvable to optimality. This approach belongs to a family of methods called policy iteration: starting from an initial policy, it constructs a series of improvements in the decisions while ruling out choices that are provably suboptimal.

Chapter 5, p. 121.

After that, we turn to reinforcement learning. For a long time, the only provably efficient reinforcement-learning methods were model-based ones; recently, a family of model-free optimistic methods emerged, each of them accompanied by an analysis of how sample-efficient the method is. We, too, study optimistic reinforcement learning, but in contrast to the existing research, we seek to understand not *how efficient* it is, but *why* it is efficient. Our analysis results in a formula that explains the three factors that cause regret—the efficiency loss—in optimistic reinforcement learning: the problem size, the measure of exploration, and the estimation error caused by the mismatch between the realized transitions and their true distribution. It can be applied to all

of the existing algorithms as well as new ones. We design one such new algorithm and show how our theoretical framework can facilitate the proof of its efficiency.

Finally, we consider a high-impact real-world sequential decision-making problem known as active wake control. Wind turbines can negatively impact each other with their wakes. These wake-induced losses can be reduced by changing the turbine orientations. Unfortunately, the optimal control strategy is non-trivial. To address this, existing approaches use simplified wake models in combination with numerical optimization methods; instead we propose to use model-free reinforcement learning. As a first step towards this goal, we present a wind farm simulator that is suitable for reinforcement learning and better reflects the realities of wind farm operation than other existing tools. Using this simulator, we show that previous research used a suboptimal action representation in this problem; we identify two alternatives, both of which improve the learning efficiency. Additionally, we demonstrate that reinforcement learning is robust to errors in the observations, providing further evidence that it is a fitting approach to active wake control.

Chapter 6, p. 149

Our contributions advance the state of the art in the theory of sequential decision-making under uncertainty and its applications. These advances hint at unexplored connections between countably-infinite planning and optimistic learning, which may lead to even more efficient algorithms for sequential decision-making under uncertainty in the future.

Chapter 7, p. 169.

Samenvatting

SEQUENTIËLE BESLUITVORMING onder onzekerheid is een belangrijke tak van onderzoek in het veld van kunstmatige intelligentie met een breed scala aan toepassingen. In dit proefschrift generaliseren we twee fundamentele eigenschappen van algoritmische methoden voor dit type besluitvormingsproblemen. Ten eerste laten we zien dat de theorie over planningsmethoden voor eindige ruimten kan worden uitgebreid tot oneindige maar aftelbare ruimten. Ten tweede introduceren we een uniform model voor algoritmen voor reinforcement learning die het principe van optimisme in het licht van onzekerheid gebruiken. Dit model wordt gebruikt om te verklaren waarom deze methoden efficiënt zijn. We gebruiken de ontwikkelde theorie om nieuwe algoritmen te ontwerpen. Afhankelijk van de behoeften van de gebruiker kunnen deze algoritmen ofwel het besluitvormingsproces volledig automatiseren, ofwel advies geven als onderdeel van beslissingsondersteunende systemen.

We beginnen met het presenteren van de basisconcepten uit de theorie van sequentiële besluitvorming en bespreken de twee benaderingen ervan: planning en reinforcement learning. We bekijken enkele typische sequentiële besluitvormingsproblemen van toenemende moeilijkheidsgraad. In het bijzonder presenteren we een spel over verplaatsingen over een rooster en de problemen van magazijn- en windparkbeheer. Vervolgens bekijken we de state-of-the-art methoden om dergelijke problemen op te lossen.

Op basis van deze analyse identificeren we een aantal onderzoeksmogelijkheden. Bij planning blijven modellen met niet-stationaire en aftelbaar oneindige gegevens tot nu toe relatief onbehandeld, omdat ze gelijkwaardig zijn aan oneindig-dimensionale optimalisatieproblemen, die notoir moeilijk op te lossen zijn, zelfs bij benadering. Bij reinforcement learning leiden optimistische benaderingen tot steekproef-efficiëntie, maar de theorie van optimisme bleef onderontwikkeld. Bovendien, hoewel reinforcement learning uitblinkt in het spelen van spellen, zoals schaken, shōgi, Go en StarCraft II, bleven de praktische toepassingen ervan beperkt.

Vervolgens bekijken we een veel gebruikt wiskundig raamwerk

Hoofdstuk 1, p. 1.

Hoofdstuk 2, p. 17.

van sequentiële besluitvorming onder onzekerheid, bekend als het Markov-beslissingsproces. We leggen uit hoe het doel van de beslisser kan worden uitgedrukt als een optimalisatieprobleem en presenteren de twee benaderingen om dit doel te bereiken. De eerste—meer gebruikelijke—benadering kent zogenaamde waarden toe aan verschillende acties. De andere benadering maakt gebruik van zogenaamde bezettingen die aangeven hoe vaak de agent de acties moet kiezen in plaats van te evalueren hoe goed deze acties zijn. In feite is bekend dat de twee technieken dual aan elkaar zijn. Hoewel deze dualiteit goed is bestudeerd in het eindige geval, is het oneindige geval minder onderzocht. Om deze kennislacune aan te pakken, presenteren we een nieuwe duale formulering voor aftelbare problemen, zowel eindig als oneindig.

Hoofdstuk 3, p. 63.

Daarna gebruiken we de duale formulering om een nieuw planingsalgoritme te ontwerpen voor oneindige-horizonproblemen met niet-stationaire gegevens. Deze problemen zijn in wezen oneindig-dimensionale optimalisatieproblemen en zijn als zodanig onmogelijk exact op te lossen met de standaardbenaderingen. We laten zien dat ze kunnen worden opgelost door het veranderen van wat wordt gedefinieerd als optimaal gedrag: in plaats van te zoeken naar een universeel optimale policy, beschouwen we de initiële-beslissing-optimale policy. In plaats van alle acties van tevoren te plannen, kan deze policy worden gebruikt om te plannen op basis van de tot dan toe beschikbare gegevens. Wanneer de volgende beslissing nodig is, kan het proces op dezelfde manier worden herhaald, wat leidt tot een optimale besluitvormingsstrategie. Onze aanpak maakt gebruik van de dualiteit tussen bezettingen en waarden om suboptimale acties uit te sluiten op basis van zogenaamde truncaties: eindige-tijd benaderingen van het oneindige-horizon besluitvormingsprobleem.

Hoofdstuk 4, p. 85.

We breiden de truncatiebenadering uit tot een meer algemene setting van besluitvormingsproblemen met aftelbaar oneindige toestandsruimten. In plaats van op tijd gebaseerde truncaties, beschouwen we op toestand gebaseerde truncaties. Dit stelt ons in staat om de hoeveelheid gegevens te beperken die nodig is om de beslissingen te nemen en om een algoritme te ontwerpen voor een klasse van problemen die anders niet optimaal zouden kunnen worden opgelost. Deze benadering behoort tot een groep methoden die policy iteration worden genoemd: uitgaande van een initiële policy, bouwt het een reeks verbeteringen in de beslissingen op, terwijl keuzes worden uitgesloten die aantoonbaar slecht zijn.

Daarna gaan we over op reinforcement learning. Lange tijd waren de enige aantoonbaar efficiënte methoden voor reinforcement learning modelgebaseerd; onlangs is een familie van modelvrije optimistische methoden ontstaan, elk vergezeld van een analyse van hoe steekproefefficiënt de methode is. Ook wij bestuderen optimistische methoden, maar in tegenstelling tot het bestaande onderzoek proberen we niet slechts te begrijpen *hoe efficiënt* een methode is, maar bovendien *waarom* het efficiënt is. Onze analyse resulteert in een formule die de drie factoren verklaart die het efficiëntieverlies veroorzaken: de probleemomvang, de mate van verkenning en de schattingsfout (het verschil tussen de geschatte en de werkelijke verdelingen). Deze theorie geldt niet alleen voor alle bestaande algoritmen, maar ook voor nieuwe. We ontwerpen zo'n nieuw algoritme en laten zien hoe ons theoretisch raamwerk het bewijs van de efficiëntie ervan kan vergemakkelijken.

Hoofdstuk 5, p. 121.

Tenslotte beschouwen we een real-world sequentieel besluitvormingsprobleem met grote impact dat bekend staat als het actief regelen van windparken. Windturbines kunnen elkaar negatief beïnvloeden met hun zog-effecten. De verliezen veroorzaakt door deze zog-effecten kunnen worden verminderd door de oriëntatie van de turbines iets te veranderen. Helaas is de optimale controlestrategie niet triviaal. Om dit aan te pakken, gebruiken bestaande benaderingen vereenvoudigde zogmodellen in combinatie met numerieke optimalisatiemethoden; in plaats daarvan stellen we voor om modelvrije reinforcement learning te gebruiken. Als eerste stap op weg naar dit doel presenteren we een windparksimulator die geschikt is voor reinforcement learning en die de complexiteit van windparken beter representeert dan andere bestaande tools. Met behulp van deze simulator laten we zien dat eerder onderzoek een suboptimale actierepresentatie voor dit probleem gebruikte; we onderscheiden twee alternatieven, die beide het leerrendement verbeteren. Bovendien laten we zien dat reinforcement learning robuust is tegen fouten in de waarnemingen, wat verder bewijs levert dat het een passende benadering is voor het actief regelen van windparken.

Hoofdstuk 6, p. 149

Onze bijdragen bevorderen de stand van de techniek in de theorie van sequentiële besluitvorming onder onzekerheid en de toepassingen ervan. Deze vorderingen duiden op nog onontgonnen verbanden tussen de twee fundamentele bijdragen in dit proefschrift; in de toekomst kan dit leiden tot nog efficiëntere plannings- en leeralgoritmen.

Hoofdstuk 7, p. 169.

Автореферат

ПОСЛЕДОВАТЕЛЬНОЕ принятие решений в условиях неопределённости — важная область в исследованиях искусственного интеллекта со множеством практических приложений. В данной диссертации обобщён ряд фундаментальных свойств данных процессов принятия решений. Во-первых, теория планирования в конечных дискретных средах перенесена на случай счётных пространств. Во-вторых, для обучения с подкреплением на основе принципа оптимизма перед лицом неизвестности предложена единая модель, объясняющая вычислительную эффективность такого подхода. Разработанная нами теория положена в основу нескольких инновационных алгоритмов, которые могут быть использованы для поддержки процессов принятия решений либо для их полной автоматизации.

В начале диссертации представлены основные концепции теории принятия решений и обсуждаются два подхода к нему: планирование и обучение с подкреплением. Здесь же рассматриваются несколько типичных задач последовательного принятия решений возрастающей сложности: игра, представляющая собой навигацию на сетке, а также проблемы управления складом и парком ветрогенераторов, а также описывается ряд современных методов решения задач подобных данным.

Глава 1, с. 1.

Основываясь на данном анализе, обозначены следующие направления исследований: разработка методов планирования в задачах с нестационарными и счётными средами, поскольку такие задачи подразумевают бесконечномерную оптимизацию и трудноразрешимы даже приближённо, и как следствие остаются малоизученными; исследование оптимизма в задачах обучения с подкреплением, так как теория оптимизма практически отсутствует, несмотря на то, что известно, что оптимистичные подходы обладают доказуемой вычислительной эффективностью; поиск путей практического применения обучения с подкреплением, блестяще проявляющего себя в таких играх, как шахматы, сёги, го и StarCraft II, но при этом по прежнему не имеющего серьёзных практических приложений.

Далее рассмотрена математическая модель последовательного принятия решений в условиях неопределённости, известная как марковский процесс принятия решений. Показано, как цель лица, принимающего решения, может быть выражена в виде задачи оптимизации, а также представлены два подхода к достижению этой цели. Первый подход присваивает так называемую ценность (value) различным действиям и является более распространённым. Второй подход использует понятие пребывания (occipancy), которое определяет не то, насколько хороши те или иные действия, а то, как часто лицо, принимающее решения, должно делать выбор в их пользу. Известно, что данные подходы математически двойственны друг другу, но в то время как данная двойственность хорошо исследована в случае конечномерных пространств, счётный случай изучен гораздо менее. Чтобы восполнить данный пробел в знаниях, представлена новая двойственная формулировка, которая может быть использована в задачах как с конечным, так и со счётным количеством переменных.

Данная двойственная формулировка использована для разработки нового алгоритма планирования в задачах с бесконечным временным горизонтом и нестационарными данными. Такие задачи по своему существу являются задачами бесконечномерной оптимизации, и поэтому их решение с использованием стандартных подходов представляется невозможным. Показано, что их решение возможно при изменении понятия оптимального поведения с поиска универсально-оптимального плана на поиск плана оптимального только в первоначальном решении. Вместо того, чтобы заранее планировать все возможные действия, такой план можно использовать для принятия решения на основе наблюдаемых данных; в дальнейшем же процесс может быть повторён, когда потребуется следующее решение; данный подход ведёт к оптимальному принятию решений. Для исключения субоптимальных действий в предложенном алгоритме используется двойственность пребывания и ценностей в так называемых усечениях, то есть ограниченных во времени приближениях задачи принятия решений с бесконечным временным горизонтом.

Подход, основанный на усечениях, в дальнейшем расширен на обобщённую постановку задачи принятия решений со счётными пространствами состояний. В основу данного подхода положены усечения, основанные на состоянии среды, а не

на времени, что позволило ограничить количество необходимых данных и разработать алгоритм принятия решений для класса задач, являющихся оптимально неразрешимыми иначе. Данный подход относится к семейству методов итерации по планам: начиная с некоторого плана, он последовательно улучшает решения, исключая варианты являющиеся доказуемо неоптимальными.

В следующей части диссертации рассмотрено обучение с подкреплением. Долгое время единственными доказуемо эффективными методами обучения с подкреплением были так называемые модельные методы; недавно же было предложено семейство безмодельных методов, основанных на принципе оптимизма, при этом для каждого конкретного алгоритма его вычислительная эффективность доказана математически. В данной работе также изучена эффективность оптимистического обучения с подкреплением, но, в отличие от предыдущих исследований, его целью является понять не *насколько*, а *почему* такое обучение эффективно. В результате проведённого анализа представлена формула объясняющая три фактора снижения эффективности в оптимистическом обучении с подкреплением: размер пространства управлений, необходимость исследования состояний системы и возможных действий, а также ошибка оценки, вызванная несоответствием между истинными вероятностями переходов и реализацией переходов в цепи Маркова в процессе принятия решений. Представленный анализ применим не только ко всем существующим, но и к новым алгоритмам. На его основе разработан один такой алгоритм и показано, как представленная теория облегчает доказательство его эффективности.

Глава 5, с. 121.

Наконец, рассмотрена важная практическая задача последовательного принятия решений, известная как активное управление турбулентным следом. Ветряные турбины способны негативно влиять друг на друга, создавая зоны повышенной турбулентности в процессе извлечения энергии из ветра. Потери, вызванные турбулентностью, можно уменьшить, повернув турбину и тем самым отклонив её турбулентный след. К сожалению, оптимальная стратегия управления нетривиальна, и существующие способы её нахождения используют упрощённые модели турбулентного следа в сочетании с методами численной оптимизации; вместо этого нами предложено использовать безмодельное обучение с подкреплением. В

Глава 6, с. 149.

качестве первого шага к достижению данной цели представлен симулятор ветряной электростанции, подходящий для обучения с подкреплением и отражающий реалии эксплуатации парков ветрогенераторов. Используя этот симулятор, показано, что предыдущие исследования основывались на неоптимальном представлении действий в данной задаче, а также сформулированы два альтернативных подхода, каждый из которых повышает эффективность обучения. Кроме того, продемонстрировано, что обучение с подкреплением является более устойчивым к ошибкам в наблюдаемых данных, что дополнительно указывает на его потенциал к решению задачи активного управления турбулентным следом.

Глава 7, с. 169.

Данное исследование продвигает как современную теорию последовательного принятия решений в условиях неопределённости, так и её приложения. Наш вклад обозначает неисследованные связи между планированием в счётных пространствах и оптимистическим обучением с подкреплением, что может привести к ещё более эффективным алгоритмам планирования и обучения в будущем.

1

Introduction

Man is born as a freak of nature, being within nature and yet transcending it. He has to find principles of action and decision-making which replace the principles of instincts.

— Erich Seligmann Fromm,
The Revolution of Hope

THIS CHAPTER introduces the topic and contributions of this thesis in layman's terms. We begin with a description of the class of decision-making problems that we study. Then we describe the two approaches to solving these problems: planning and reinforcement learning. Next, we look at a few examples of such problems, from a very simple frozen lake game to a complex active wake control problem. Then, we discuss the existing approaches and their limitations. Finally, we formulate the research questions that are answered in this thesis.

1.1 The Decision-Making Problem

1.1 THE DECISION-MAKING PROBLEM

This thesis presents novel algorithms for solving the problems of

SEQUENTIAL DECISION-MAKING UNDER UNCERTAINTY

[Puterman, 1994]. What kind of problems are these precisely? To explain it, let us separately examine the three parts of the term above: “decision-making,” “sequential,” and “under uncertainty.”

First, we study decision-making: we consider an agent who has to make a choice based on some data. The data observed by the agent is called a *state*. Based on this observation, the agent chooses one of the *actions* available to them. The agent knows the possible states and actions available at each state; in other words, the agent knows the *environment* they operate in. Nevertheless, this information is not enough to make informed decisions: the agent needs to know how good—or bad—each choice is. To signal this, the environment gives the agent a *reward*. The interaction between the agent and the environment is illustrated by Figure 1.1.

Next, we consider sequential problems, that is, problems that require decisions to be made repeatedly over time. Each interaction between the agent and the environment is known as a *decision epoch*. After a decision epoch takes place, the environment transitions to a new state, which the agent immediately observes, and a new decision epoch starts. This chain of interactions continues either for a predetermined number of decision epochs, known as the *horizon* of the problem, or indefinitely. In the latter case, we say that the problem has an infinite horizon.

Finally, the problems we consider may involve uncertainty of different kinds. The *decision rules* that the agent uses to determine their actions can be stochastic. For example, the agent is allowed to flip a coin and choose an action based on the result. The

An agent can be a person, a robot, or a piece of software.

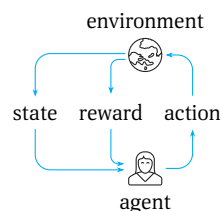
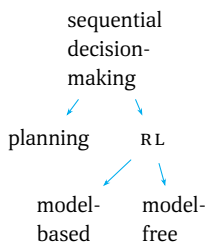


Figure 1.1: The decision-making process.

Why do we need history independence?
 When the environment is history-independent, so are the agent's optimal decisions. This substantially simplifies the problem.

rewards can be randomized as well. But the most important type of uncertainty governs the *state transitions*: the next state can also be random. The only condition that we will impose is that the transitions must be *history-independent*: the state of the next epoch must depend on the current state and action only, and not on the previous ones. While this condition may seem restrictive at first, history-dependence can often be addressed by carefully reformulating the problem. Often, the state can be redefined to include all of the data that governs the transitions, including the data from the previous decision epochs.

1.2 PLANNING & REINFORCEMENT LEARNING



Glass boxes are often called white boxes to oppose them to black ones.

The methods for solving problems of sequential decision-making under uncertainty fall under two broad categories: planning and reinforcement learning (often referred to as RL).

Planning methods assume that the environment is a so-called *glass box*, a system whose internal structure can be seen. The agent knows the environment's decision rules: how the next state and the reward are chosen for each of the possible actions. Even when the environment's behavior is random, the agent still knows the probabilities of different outcomes and can calculate the expected immediate outcome of each action.

Behaviorists further distinguish reinforcement from punishment. The latter is meant to discourage an unwanted behavior instead of promoting a desired one. In machine learning, there is no distinction between reinforcement and punishment.

When the agent has no access to the environment model, it is a *black box* problem. In this case, *reinforcement learning* can be used instead of planning [Sutton and Barto, 2018]. The term *reinforcement* comes from behavioral psychology and means a stimulus used to produce a desired response. For example, puppies are given treats after correctly performing a command such as "sit," "stay," or "heel" when they are trained. This treat is meant to induce a desired behavior and is called *positive reinforcement*. Similarly, *negative reinforcement* teaches to avoid undesirable outcomes. Sunburns are an example of negative reinforcement: to prevent them, we learn to apply sunscreen before going out. Employing the ideas from the behavioral science, reinforcement learning is a machine learning technique based on rewards. In reinforcement learning, the agent does not know the outcomes of their actions beforehand, but learns from rewards, either positive, or negative.

Reinforcement learning methods can be further divided into two subcategories. In *model-based* methods, the agent learns the model of the environment. For example, the agent can infer that

the action of not using sunscreen sometimes leads to sunburns. In *model-free* methods, the agent foregoes such a model: they simply know that applying sunscreen is good, but they do not care to reason why. While model-based reinforcement learning algorithms often perform better, they can be computationally intensive when the environment is complex and the causality is non-trivial. In extreme cases, model-based reinforcement learning can be rendered inapplicable.

1.3 Examples

In this thesis, we study both approaches to decision-making. We use planning methods for glass-box decision-making, and reinforcement learning for black-box problems. In the latter case, we focus entirely on model-free reinforcement learning, as it is applicable to a broader range of problems.

1.3 EXAMPLES

Now that we outlined the class of problems that we are interested in and the types of methods that can be used to solve them, let us consider a few examples.

We start with a simple problem known as *frozen lake*. It is a stylized example which has only a few states and actions, and is easy to solve for humans, making it a good illustration for various concepts from the theory of sequential decision-making.

The second example is a more practical problem of *inventory management* in a warehouse. This problem is interesting to us because it breaks some common assumptions used in sequential decision-making. As a result, most of the existing solution methods cannot be applied to this problem directly.

Finally, we consider a problem of *active wake control* in wind farms. This is an example of a complex problem where the environment behavior is hard to model. This makes planning particularly difficult, as it requires full knowledge of the system's dynamics. On the other hand, model-free reinforcement learning does not require such knowledge, making it a good fit for this problem.

1.3.1 Frozen Lake

Frozen lake is a game that appears in OpenAI Gym [Brockman et al., 2016], a collection of benchmark problems for reinforcement learning. It is described as follows.

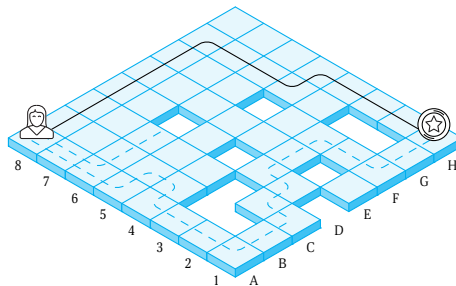
1 Introduction

“ Winter is here. You and your friends were tossing around a frisbee at the park when you made a wild throw that left the frisbee out in the middle of the lake. The water is mostly frozen, but there are a few holes where the ice has melted. If you step into one of those holes, you’ll fall into the freezing water. At this time, there’s an international frisbee shortage, so it’s absolutely imperative that you navigate across the lake and retrieve the disc. However, the ice is slippery, so you won’t always move in the direction you intend. ”

One of the default versions of this game is shown in Figure 1.2. The agent starts in the initial state A8 and the goal is to reach the frisbee state H1 without falling into any of the holes, for example, D1 or B3. The actions of the agent are movements along either of the two axes. For example, from the starting position A8 the agent can go to A7 or B8. Although the description mentions that the lake is slippery, this default version is not and the agent always moves in the intended direction. In the slippery version, agent sometimes moves to a different state than they intended.

This is a so-called *grid world* problem. Pac-Man is a more complex example that can be considered a grid world.

Figure 1.2:
FrozenLake8×8-v0.
The solid line and the dashed lines show some of the optimal and suboptimal paths respectively.



In this problem, a unit reward is given to the agent upon reaching H1 from either G1 or H2, and there is no reward for any of the remaining movements. Any sequence of actions results in a path on the lake surface. Some of them never reach the frisbee, giving no reward. Some reach it, but take longer than necessary. An optimal plan should provide the agent with a path that leads to the frisbee as fast as possible.

1.3.2 Inventory Management

Single-product inventory management motivated development of several algorithms for sequential decision-making under uncertainty [Veinott and Wagner, 1965; Tijms, 1972; Lee et al., 2017]. It is defined by Puterman [1994, Section 3.2] as follows.

“ Each month, the manager of a warehouse determines current inventory of a single product. Based on this information, he decides whether or not to order additional stock from a supplier. In doing so, he is faced with a tradeoff between the costs associated with keeping inventory and the lost sales or penalties associated with being unable to satisfy customer demand for the product. The manager’s objective is to maximize some measure of profit over the decision-making horizon. Demand for the product is random with a known probability distribution. ”

1.3 Examples

✧ In this thesis, we consider a generalization of this problem that includes multiple products. In this case, the state includes current stock on hand for each of the products. Having observed the current inventory at the beginning of the month, the warehouse manager places an order. The order tells how much of each product needs to be shipped to the warehouse. The shipment size is restricted by some measurement, such as the volume of a truck, or the maximum weight of the load. The action space includes all of the combinations of products with the total measurement not exceeding the allowed maximum.

During the month, customers place orders for the product, and the product is shipped to them, if it is still in stock. The total demand and the order for each product lead to a transition to a new state in the next month.

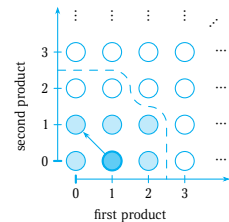
The sales revenue and the inventory costs are proportional to the demand and the total inventory including the newly ordered units, respectively. The ordering costs are usually assumed to consist of a fixed cost for placing the order, and a variable part depending on the number of units of each product that was ordered. The price of the product, the inventory holding cost per unit and ordering costs for different possible orders are all known to the manager. Based on them, the manager can calculate their reward as the sales revenue less holding and ordering costs.

A simple decision rule for a single-product model is to wait until the inventory drops below some *minimum fill* and then to re-stock to a given inventory size called the *target stock*. In fact, this decision is known to be optimal in the single-product case [Veinott, 1966], including an extended model with two suppliers [E. J. Fox et al., 2006]. A multi-product problem can be approached by considering each product in the same way independently from the other products, but we do not know if this approach is optimal. Human managers use this or similar decision rules. At the same

E.g., in a two-product model $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ is a state with one unit of the first product.

If the order weight cannot exceed five tonnes, and the products weight three and two tonnes per unit respectively, possible orders are $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\begin{bmatrix} 0 \\ 2 \end{bmatrix}$.

If the state is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and the action is $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$, the next state can be any of the shaded circles, depending on the demands. E.g., if the demand vector is $\begin{bmatrix} 2 \\ 0 \end{bmatrix}$, the next state will be $\begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$; this transition is represented by an arrow. The next state cannot be beyond the dashed line.



1 Introduction

time, algorithms for sequential decision-making can be used to automate the order placement, saving time to the warehouse manager and potentially cutting the costs of ordering and inventory holding.

1.3.3 Active Wake Control

Our final example is a complex control problem of great practical interest that arises in wind farm operation.

When a wind turbine extracts energy from the wind, it creates a wake area behind its rotor [Vermeer et al., 2003]. The wind in this area has reduced velocity and increased turbulence. If another turbine is positioned in the wake, both of these factors impact its power output. In large wind farms, these wake-induced losses can be substantial. For example, a study of an off-shore wind farm in Denmark shows a 12% energy loss due to wake effects [Barthelmie et al., 2009]. Another study conducted in Alberta, Canada reports a similar loss of 7%–13% [Howland et al., 2019]. Under certain atmospheric conditions, turbine wakes can extend so far that they even affect other wind farms [Lundquist et al., 2019]. As the number of wind farms around the world and their average size continue to grow [Jacobson and Delucchi, 2009], so do their wake-induced losses. Consequently, active wake control is important to efficient wind farm operation.

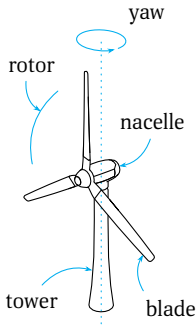


Figure 1.3: Turbine nomenclature.

Early studies of wake effects mitigation focused on per-turbine control of either pitch [Steinbuch et al., 1988; Schepers and van der Pijl, 2007; Madjidian and Rantzer, 2011] or generator torque [Johnson, 2004]. Later, joint farm-level control of turbines has been demonstrated to be an efficient strategy [Gebraad et al., 2016; Howland et al., 2019]. This is done via active control of the turbine yaws (that is, the horizontal-plane rotations, see Figure 1.3). When a turbine is yawed relative to the incoming wind, it has lower power output but the wake center shifts [Wagenaar et al., 2012]. This wake deflection can be used to improve the power output of down-wind turbines, increasing the total power production.

Figure 1.4: Overhead view of two wind turbines without (left) and with (right) yaw-based wake control. Darker areas have slower wind.

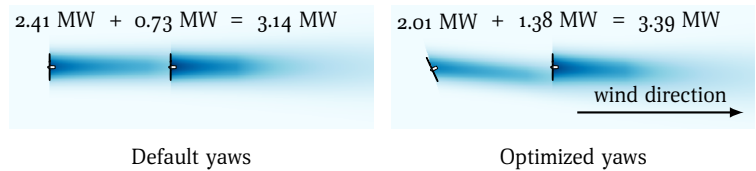


Figure 1.4 shows an example of wake effects in a two-turbine wind farm and the benefit of yaw-based active wake control. If wake effects were nonexistent, both turbines would have produced 2.41 MW for a total of 4.82 MW. Because of the wake of the upwind (left) turbine, the downwind turbine produces only 0.73 MW instead. Yawing of the upwind turbine by approximately 25° counterclockwise reduces its power output by 0.4 MW, but the wake deflection allows the downwind turbine to produce 0.65 MW more, increasing the overall power output by 0.25 MW or 8%.

1.4 Existing Research

The optimal wake control strategy primarily depends on the turbine locations relative to each other. At the same time, it is also affected by various atmospheric conditions, such as wind speed and direction, and air temperature. These conditions change over time. The optimal active wake control strategy should account for such changes in the data by repeatedly adjusting the wind farm yaws throughout the day.

In this problem, the wind farm operator is the decision-making agent. The agent's actions are the yawings of the turbines. These actions are chosen based on the current yaw angles of the turbines and the atmospheric measurements available to the agent. Therefore, the collection of these data forms the state of the problem. The state changes over time. While the yaw changes are deterministic, the atmospheric conditions change stochastically, adding uncertainty that the agent should account for.

1.4 EXISTING RESEARCH

1.4.1 *Planning with Markov Decision Processes*

There are multiple ways of modeling sequential decision-making, but the most commonly employed model is the *Markov decision process*. The theory of Markov decision processes traces back to Bellman [1954]. The work of Puterman [1994] is one of the most comprehensive compendia of this field of research.

Various definitions of an optimal behavior in Markov decision processes can be found in the literature. These are called optimality criteria. Most works—including this thesis—use the discounted expected total reward criterion, which dates back to the works of Howard [1960] and Blackwell [1965]. The expected total reward [Ornstein, 1969; van der Wal, 1981] and the expected average reward [Denardo and B. L. Fox, 1968; Dynkin and Yushkevich, 1979;

See (2.7), p. 35 for a formal definition of this criterion.

1 Introduction

For examples of
ill-posed Markov
decision processes, see
Section 2.2.1, p. 36.

Ashok et al., 2017] criteria are widely employed as well; however, these are outside of the scope of this thesis.

In general, even when an optimality criterion is given, a Markov decision process is not necessarily a well-posed problem, in the sense that the optimal solution may be unattainable or even non-existent. To guarantee that the problem has an optimal solution, additional assumptions are made about the actions, states, rewards, and transitions. These assumptions lead to various subclasses of Markov decision processes that require different algorithms to be solved.

Finite-horizon problems with finite state and action spaces

These problems form the simplest class of Markov decision processes. They can be either stationary, when the problem data do not change over time, or non-stationary otherwise. In both cases, these problems can be solved using dynamic programming [Bellman, 1954; Bellman and Dreyfus, 2016].

Stationary infinite-horizon problems with finite state and action spaces

These problems always have optimal solutions [Puterman, 1994], which can be found using various methods that fall under three main categories.

- *Value iteration* [Blackwell, 1965; Balaji et al., 2018] methods compute the so-called value function. For each state, this function shows the best possible expected total reward that the agent can collect starting from that state. When the values of all states are known, the agent can compute state-action values that show how good each action is for each state. The optimal behavior is then to take an action with the highest value in each state.
- *Policy iteration* methods [Howard, 1960; Scherrer, 2013], produce a sequence of improving policies, until no improvement can be made. The last returned policy is an optimal one, since it cannot be improved further.
- *Linear programming* methods [d'Epenoux, 1963; Malek et al., 2014] use an alternative approach to find the value function. The values can be found as a solution to an optimization problem based on the data of a Markov decision processes. The solution of the resulting problem yields the value function. Alternatively, the dual linear program can be used to find the optimal policy directly.

policies prescribe which
action should be taken
in each state

Stationary problems with countably-infinite state and action spaces

For these problems, the same theory applies: both the value functions and optimal policies exist, and they correspond to the solutions of the linear programming formulation of the problem. Unfortunately, the infinite number of states renders each of the methods useless. For example, the linear program contains infinitely many variables and constraints. Instead, these problems have to be approached with different techniques.

1.4 Existing Research

- *State-space truncation* methods [B. L. Fox, 1971; White, 1979; White, 1980] approximate the countably-infinite state space with a finite one, and use the resulting solution as an approximate solution to the original problem. The approximate problem is solved using either value iteration [White, 1982; Cavazos-Cadena, 1986] or policy iteration [Lee et al., 2017], sometimes in combination with the linear-programming approach.
- *Structured models* [White, 1981] analytically identify the structure of the value function or the policy, and use this structure to reduce the problem to a finite one.

Non-stationary infinite-horizon problems with finite state and action spaces

These problems can be reformulated as stationary infinite-horizon MDPs with a countably-infinite state space and a finite action space. Therefore, the methods described earlier can be used for these problems, for example, as done by Ghate and R. L. Smith [2013].

Truncation methods offer an alternative approach [Bès and Lasserre, 1986; Bès and Sethi, 1988; Hopp, 1989; Cheevaprawatdomrong, Schochetman, et al., 2007]. These methods seek a solution horizon, that is, a finite time horizon such that the optimal initial decision is guaranteed to be the same between the truncation and the full problem. While the policy prescribed by such methods is not guaranteed to be optimal in the future, it can be used to make an immediate decision.

Problems with continuous state and action spaces

Like in the countably-infinite state-action space, the existence of the value functions and optimal policies can be established in this case under some additional assumptions [Shreve and Bertsekas, 1978; Puterman, 1994], but the resulting equations are generally

not computationally feasible.

1 Introduction

A common remedy is to restrict the continuous functions to a class of functions defined by a finite set of parameters, such that the optimization can be performed over these parameters instead of the original functions. This idea is used in fitted value iteration [Szepesvári and Munos, 2005; Munos and Szepesvári, 2008] and fitted policy iteration [Antos et al., 2007]. Similar approximations for the linear programming approach exist as well [Hauskrecht and Kveton, 2003; Hauskrecht and Kveton, 2006].

1.4.2 Reinforcement Learning

The history of reinforcement learning based on MDPs begins with the work of Watkins [1989], who introduced the idea of learning the state-action value function from interactions with the environment, instead of computing it based on the underlying MDP. Since then, reinforcement learning has become the dominant paradigm for learning in black-box sequential decision-making problems. For a comprehensive introduction to the theory of reinforcement learning and state-of-the-art algorithms, the reader is referred to Sutton and Barto [2018].

Just like the planning methods for MDPs, reinforcement learning algorithms depend on the structure of the underlying problem, and can be divided into two broad categories: tabular methods and deep reinforcement learning.

Tabular methods and optimism

These methods are used in problems with finite state and action spaces. They estimate the so-called state-action value function—also known as the Q-value function—that can be represented by a table of values.

The seminal algorithm, Q-learning [Watkins, 1989], starts with arbitrary assigned values, and adjusts them based on the observed interactions with the environment in a manner that guarantees convergence to the optimal value function. The idea is further developed in the algorithms called temporal-difference (TD) learning [Tesauro, 1995; Sutton and Barto, 2018] and SARSA [Rummery and Niranjan, 1994; Singh et al., 2000].

SARSA stands for
state-action-reward-
state-action learning

Research of tabular reinforcement learning primarily focuses on improvement of the learning efficiency. Various techniques include variance reduction methods [Devraj and Meyn, 2017], posterior sampling [Osband and Van Roy, 2017; Agrawal and Jia,

2017], randomized value functions [Osband, Roy, et al., 2019], and optimistic learning [Szita and Lőrincz, 2008]. The latter methods use the principle of *optimism in the face of uncertainty* [ibid.], which postulates that a learning agent should assume that its actions lead to the best realistically possible outcomes. In practice, this principle is implemented in two ways:

1.4 Existing Research

- *optimistic initialization*—unencountered state-action pairs are assumed to have the best outcomes [Sutton and Barto, 2018, Chapter 2.6], and
- *action selection based on upper confidence bounds (UCBs)*—each previously encountered state-action pair is assumed to yield the best statistically plausible reward [ibid., Chapter 2.7].

Optimistic Q-learning methods are of special interest, as they are provably efficient [Jin et al., 2018]. They include upper confidence bound Q-learning that comes in two forms: with *Hoeffding-style* bonus (UCB-H) [ibid.], and with *Bernstein-style* bonus (UCB-B) [ibid.], *infinite-horizon* UCB (∞ -UCB) Q-learning [Y. Wang et al., 2020], *optimistic pessimistically-initialized* Q-learning (OPIQ) [Rashid et al., 2020], and UCB2-based methods in the context of problems with limited adaptivity [Bai et al., 2019].

Deep reinforcement learning

Recently, artificial intelligence achieved outstanding performance in various games, such as chess, shōgi, and Go [Silver, A. Huang, et al., 2016; Silver, Hubert, et al., 2017; Schrittwieser et al., 2020], as well as StarCraft II [Vinyals et al., 2019] and various Atari arcade games [Silver, Hubert, et al., 2017; Schrittwieser et al., 2020; Kapurrowski et al., 2018], in some cases going as far as winning against some the best players in the world. In all cases, this breakthrough can be attributed to deep reinforcement learning.

Deep reinforcement learning uses deep neural networks to represent the policy or the Q-value function. It is most effective in problems with high dimensional state space [François-Lavet et al., 2018], for example, learning from visual perceptual inputs made up of thousands of pixels [Mnih, Kavukcuoglu, Silver, Rusu, et al., 2015].

State-of-the-art deep RL methods include among others *trust region policy optimization* (TRPO) [Schulman, Levine, et al., 2015], *proximal policy optimization* (PPO) [Schulman, Wolski, et al., 2017], *deep deterministic policy gradient* (DDPG) [Lillicrap et al., 2015], *twin delayed deep deterministic policy gradient* (TD3) [Fujimoto

Shōgi (将棋) is a board game also known as Japanese chess.

Go (囲碁) is the oldest board game in the world that is still played today.

StarCraft II is one of the most popular real-time strategy video games.

Atari games include Breakout, Ms. Pac-Man, Qbert and other popular arcade titles.

et al., 2018], *soft actor-critic* (SAC) [Haarnoja et al., 2018], *distributional reinforcement learning* [Bellemare et al., 2017], as well as myriads of their modifications and extensions. These algorithms are included in various reinforcement learning software packages and libraries [Achiam, 2018; Moritz et al., 2018; Raffin et al., 2019; S. Huang et al., 2020] leading to their wide adoption in practice.

1.4.3 Knowledge Gaps

While the body of research in sequential decision-making under uncertainty is vast and seemingly evergrowing, it is not all-encompassing.

In planning, most of the research focuses on stationary problems with a finite state-action space, and non-stationary infinite-horizon problems remain relatively unexplored due to their infinitely-dimensional nature. Moreover, the theoretical results obtained for stationary problems cannot always be applied to non-stationary ones. At the same time, knowledge transfer in the opposite direction is always possible: stationary problems are a special case of non-stationary ones when all of the data at different time steps coincide. Can we develop more efficient algorithms for non-stationary MDPs, and use the new insights to solve the simpler problems more efficiently as well?

Similarly, the theory of planning in countably-infinite problems can be transferred to finite stationary and non-stationary problems, but not vice versa. Is it possible to extend the existing approaches for non-stationary problems to the more general case of countably-infinite ones?

In tabular reinforcement learning, the principle of optimism leads to efficient algorithms. While each particular optimistic algorithm design is proven to be efficient, why does optimism lead to such efficiency remains unclear. Instead of analyzing the efficiency of individual algorithms, can we obtain deeper insights into the efficiency of reinforcement learning by studying optimism more generally?

Deep reinforcement learning shows tremendous successes in playing video games and board games, but real-life applications remain limited. We have already mentioned one such application: active wake control. By using RL algorithms for active wake control, can we obtain new insights that can be used in solving this and other real-world problems more efficiently?

1.5 CONTENT OF THIS THESIS

Having identified some gaps in the theory of sequential decision-making under uncertainty, we formulate the *raison d'être* of this thesis. Our research goal is

1.5 *Content of This Thesis*

TO OBTAIN NEW INSIGHTS AND DEVELOP NOVEL
ALGORITHMS BY GENERALIZING THE EXISTING THEORY
AND APPLICATIONS OF SEQUENTIAL DECISION-MAKING
UNDER UNCERTAINTY.

1.5.1 *Research Questions*

In pursuing the goal, we formulate the following research questions based on the discussed open challenges.

Question 1

How can we find optimal decisions in non-stationary infinite-horizon problems with unbounded rewards?

Question 2

How can we find optimal decisions in problems with countably-infinite environments?

Question 3

Is optimistic learning efficient in non-stationary problems; if so, how can this efficiency be explained?

Question 4

How can reinforcement learning be applied to efficiently solve real-world problems such as active wake control?

1.5.2 *Contributions of This Thesis*

We address the research questions by designing several novel algorithms for sequential decision-making under uncertainty, including two planning algorithms and a reinforcement learning one, and by developing a novel RL environment for active wake control.

First, we address the first question of optimal planning in the non-stationary infinite-horizon problems. We propose an algorithm based on the linear program formulation of the problem, and show that it can identify the exact optimal initial decision, given that it is unique, in a more general setting than the existing algorithms [Neustroev, de Weerd, and Verzijlbergh, 2019].

Second, we answer the second research question by extending the theory of non-stationary MDPs to countably-infinite ones. We design an algorithm that does policy iteration in these problems and can be applied even when the state space is multidimensional. We apply the algorithm to solve the inventory management problem.

Next, we consider a general model of optimism in tabular reinforcement learning to answer the third question. Using our model, we prove the efficiency of optimistic reinforcement learning in terms of regret, and identify the three factors that produce the regret: the problem size, the optimistic overestimation, and the estimation error. We also show how this novel theory of optimism can be used to facilitate design of new RL algorithms [Neustroev and de Weerd, 2020].

Finally, we develop an RL environment for active wake control in wind farms to investigate the fourth research question. We show how the problem formulation affects the efficiency of RL methods, and find that reinforcement learning is capable of outperforming the state-of-the-art control method when there is noise in the data observed by the agent [Neustroev, Andringa, et al., 2022a].

2

A Mathematical Model of Decision-Making

*It is really true . . . that life must be
understood backwards. But . . . it must
be lived forwards.*

— Søren Aabye Kierkegaard,
Journalen JJ · 167

Translated by P. E. T. Jorgensen

THIS CHAPTER describes a mathematical model of sequential decision-making under uncertainty. This model, known as the Markov decision process, formalizes the concepts introduced in the previous chapter. Under certain conditions, policies—sequences of decisions—can be numerically evaluated. In this case, an optimal policy can be found by using either the state value function or the occupancy measure; these two approaches are dual to each other. When the admissible control space is finite, these should be well known to readers familiar with the theory of Markov decision processes. In contrast, some of the properties used in the finite case no longer hold when the environment is countably-infinite. This leads to a different treatment of such problems, resulting in a new dual formulation presented in this chapter. Finally, a look at the multi-product inventory management problem illustrates how the new theory can be applied.

2.1 MARKOV DECISION PROCESSES

There exist multiple ways of modeling problems of sequential decision-making under uncertainty. For example, they can be written as optimization problems or using graphical models like the one shown in Figure 2.1. We formalize the decision-making process as a *Markov decision process* (MDP). It is one of the most popular models that provides a mathematical formalism for a plethora of planning and RL algorithms.

Definition 2.1 | Markov decision process

A T -horizon MDP $\mathfrak{M}_T \triangleq (T, \mathbb{S}, \mathbb{A}, A_p, \alpha, p, r)$ is a tuple of:

- a time horizon T ;
- a set \mathbb{S} of possible states;
- a set \mathbb{A} of possible actions;
- a permissibility function A_p ;
- an initial state distribution α ;
- a transition kernel p that governs the state transitions;
- a reward kernel r that provides the agent's rewards.

☞ This definition is incomplete; it tells us the components of an MDP, but does not define them properly yet. We now define each of these elements, the relations between them, and some other related concepts. We divide the definitions into three parts. First, we describe the sequence of decisions by both the environment

In this chapter, known results are presented as propositions accompanied with references to the literature. Two new results are presented in Theorems 2.14 and 2.22 (see pp. 47 and 57). Because statements similar to both of these can be found in the literature, the differences with previously established results are explained thereafter. This chapter contains a few minor novel results as well; these are presented as lemmas followed by formal proofs.

Andrey MARKOV (1856–1922) studied memoryless stochastic processes. The dynamics of these processes depend on the present state and none of the past states. The adjectives *Markov* and *Markovian* are used to describe such processes.

2 A Mathematical Model of Decision-Making

Choices by:

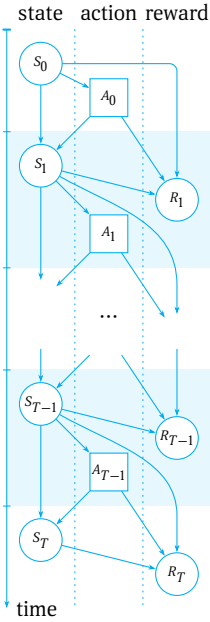
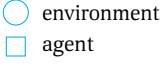


Figure 2.1: Graphical model of the problem. Each block represents a time step and contains one decision by the agent. Within a time step, lower nodes occur later.

and the agent in the model: when do these decisions occur and what information are they based on. Then, we add the uncertainty: the randomness caused by the decisions of the environment and possibly of the agent. And finally, we describe the optimality criterion: what exactly does it mean for the agent to act optimally in the decision-making problem given by an MDP.

2.1.1 Sequentiality

We start with the sequence of the decisions. To describe the outcomes of a decision-making process, we first need to define the concepts of a decision epoch, state, action, and reward.

Decision epochs and horizon

The decision-making happens between a starting point in time $t = 0$ and some final point, known as the *horizon* T , excluding the latter. It can also continue *ad infinitum*, in which case we write $T = \infty$. In general, the decisions may be taken

- continuously through time,
- at random times, when the decision-maker's attention is required,
- at specified (usually uniformly spaced) times.

Markov decision processes model the latter case only, therefore, it is the only case we consider. Without loss of generality, we let each decision epoch last one unit of time, $t \in \mathbb{T} \triangleq \{0, 1, \dots, T-1\}$, where $1 \leq T \leq \infty$ and \mathbb{T} denotes the *time space* of the problem. In this case we call each decision epoch a *time step*.

Figure 2.1 presents the probabilistic graphical model of a sequential decision-making process. Nodes show decisions made by the environment and the agent. The lower the node is, the later in time its decision is made, including within each time step. The arrows show causal connections, that is, what information is each of the decisions based on.

We use capital letters S_t , A_t , and R_t to denote the states, actions, and rewards during each time step t . The MDP starts in some initial state S_0 chosen by the environment, possibly at random. The agent observes this state and chooses their initial action A_0 . This choice can be randomized as well. Next, the environment chooses a new state S_1 , based on the initial state S_0 and action A_0 , and a reward R_1 , based on both states and the action. The process continues until the final time step $t = T$, where no action is taken. The state S_T is observed and the reward R_T is received.

The environment is assumed to be Markovian. This property means that each non-initial state $S_t, t > 0$ depends on the previous state S_{t-1} and action A_{t-1} only, and similarly with the rewards. Strictly speaking, the actions of the agent are allowed to depend on all of the information observed so far. For example, action A_1 in Figure 2.1 is chosen based on everything that happened before it, and should have incoming arrows not just from the current state S_1 , but from the initial state S_0 , action A_0 , and the reward R_1 of time step 1 as well. However, if the environment is Markovian, then the agent's decisions can be based on the current state only and remain optimal. For clarity of presentation, Figure 2.1 omits all of the causal links that do not affect the decision optimality.

Even though the rewards are allowed to be random, in many problems they are deterministic and thus are not chosen by the environment directly. Moreover, rewards may depend on the current state-action pair (S_t, A_t) only and not the transition to the next state S_{t+1} . These two assumptions lead to a new, simpler model shown in Figure 2.2.

In fact, any MDP has an equivalent MDP with deterministic rewards independent of the transitions [Puterman, 1994, p. 20].

In planning, when all of the information on the environment is available, this simplified reformulation is often readily available. In RL, this is not always the case, but a similar assumption is often made for simplicity. We, too, use the model of Figure 2.2, unless otherwise stated.

State space

The states s that the agent observes come from the state space \mathcal{S} . We consider discrete state spaces, either finite, $\mathcal{S} = \{s_0, s_1, \dots, s_{|\mathcal{S}|-1}\}$, or countably-infinite, $\mathcal{S} = \{s_0, s_1, \dots\}$, unless stated otherwise.

A notable example of a continuous state space is a closed interval $\mathcal{S} = [s_-, s_+]$ and, more generally, a multi-dimensional box. In this thesis, the active wake control problem is the only example with such states. In this problem the states are vectors of the observed atmospheric conditions and turbine yaw angles, each within a predefined interval.

Note that we use small letters s to denote all of the possible states within the state space \mathcal{S} and capital letters S to denote the actual states observed while making the decisions, as described in the previous section. For example, in FrozenLake8×8-v0, the states s_0-s_{63} are the squares A8–H1; they are numbered left-to-right

2.1 Markov Decision Processes

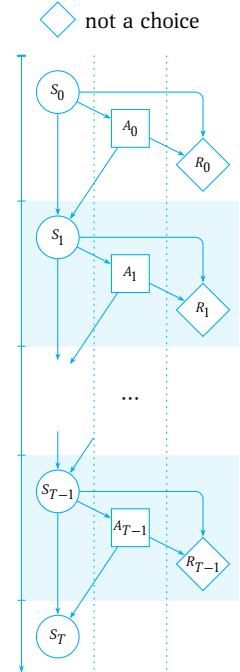


Figure 2.2: Graphical model of an MDP with deterministic rewards.

0	1	2	3	4	5	6	7	8
8	9	10	11	12	13	14	15	7
16	17	18	19	20	21	22	23	6
24	25	26	27	28	29	30	31	5
32	33	34	35	36	37	38	39	4
40	41	42	43	44	45	46	47	3
48	49	50	51	52	53	54	55	2
56	57	58	59	60	61	62	63	1
A	B	C	D	E	F	G	H	

Figure 2.3: State space of FrozenLake8×8-v0

When the action space \mathbb{A} is discrete, the action permission function A_p can return any combination of actions. The continuous case is more restrictive, but we omit the differences.

0	1	2	3	8
8	9	10	11	7
16	17	18	∅	6
24	25	26	27	5
A	B	C	D	

←	0	1	2	3	8
↓	8	9	10	11	7
←	16	17	18	19	6
→	24	25	26	27	5
A	B	C	D		

Figure 2.4: Examples of permitted actions in FrozenLake8×8-v0
Top: only some actions are permitted.
Bottom: all actions are permitted, but some of them (lighter arrows) do not change the state.

then top-to-bottom as shown in Figure 2.3. The starting state S_0 is the square A8, $S_0 = s_0$. The next state S_1 depends on the initial decision A_0 made by the agent: if the agent goes right to B8 then $S_1 = s_1$, and if they go down to A7, then $S_1 = s_8$.

Action space and action permission function

Similarly to the states, the actions a chosen by the agent belong to some action space \mathbb{A} . When the state space is discrete, we consider finite discrete actions spaces only, $\mathbb{A} = \{a_0, a_1, \dots, a_{|\mathbb{A}|-1}\}$.

While the action space \mathbb{A} describes all of the possible actions, the choices available to the agent may differ for different states. The set of actions $A_p(s)$ permitted at a state s is given by the action permission multifunction $A_p : \mathbb{S} \rightarrow 2^{\mathbb{A}}$, where $2^{\mathbb{A}}$ is the power set of the action space \mathbb{A} , that is, the set of all of its subsets. When the agent finds themselves in a state s where no actions are permitted, $A_p(s) = \emptyset$, the decision-making process terminates early. When the permissibility function is not explicitly mentioned, we assume that all actions are permitted, $A_p(s) = \mathbb{A}$.

In the frozen lake example, there are four actions corresponding to the directions that the agent can move in:

$$\mathbb{A} = \{a_0, a_1, a_2, a_3\} = \{\leftarrow, \downarrow, \rightarrow, \uparrow\}.$$

At the same time, some tiles are located along the edges of the lake, and not all actions are available in them. Additionally, stepping on either a hole or the frisbee tile terminates the decision-making process. In FrozenLake8×8-v0, we can set $A_p(s_0) = \{\downarrow, \rightarrow\}$, $A_p(s_{17}) = \mathbb{A}$, and $A_p(s_{19}) = \emptyset$, as shown in Figure 2.4, and similarly for every other state.

It is often assumed that all of the actions are permitted in each state, that is, $A_p(s) = \mathbb{A}$ for all states $s \in \mathbb{S}$, with a possible exception of terminal states, where $A_p(s) = \emptyset$. There are different ways to ensure that this assumption does not lead to a loss of generality. In FrozenLake8×8-v0 this is done by adjusting the transitions: when the agent attempts to walk off the grid, they stay in the same state and receive no reward. Other approaches are to assign a reward of $-\infty$ to forbidden actions, or to replace forbidden actions with duplicates of permitted ones. The latter method does not work when there are states with no permitted actions, however, like the hole states of the frozen lake problem.

Admissible control space

Each time step t is characterized by a state-action pair (S_t, A_t) showing the decisions made by the environment and the agent respectively. Because of the limited permissibility, not all state-action pairs (s, a) , $s \in \mathbb{S}$, $a \in \mathbb{A}$ can occur during the decision-making process. It is useful to distinguish the permitted pairs from the rest. To do so, we introduce the following space.

Definition 2.2 | admissible control space

The *admissible control space* $\mathbb{X} \subseteq \mathbb{S} \times \mathbb{A}$ is the space of all state-action pairs (s, a) such that the action $a \in \mathbb{A}$ is permitted in the state $s \in \mathbb{S}$:

$$\mathbb{X} \triangleq \{x = (s, a) \mid s \in \mathbb{S} \text{ and } a \in A_p(s)\}.$$

- ☞ When all actions are permitted in all states, the admissible control space \mathbb{X} coincides with the product space $\mathbb{S} \times \mathbb{A}$, but in general it is allowed to be a subset thereof.

Reward space

Each reward is a real number, either positive or negative. The reward signal is a form of reinforcement: higher rewards correspond to better choices by the agent. The reward space can be simply the real line \mathbb{R} or an interval $[r_-, r_+]$. In the latter case the rewards are called *uniformly bounded*. Frozen lake is an example of a problem with uniformly bounded rewards. There is a unit reward for collecting the frisbee and no reward in all other situations. Setting $r_- = 0$ and $r_+ = 1$ allows us to uniformly bound the rewards by the unit interval $[0, 1]$.

2.1 Markov Decision Processes

For further explanation why uniformly bounded rewards are important, see Condition 2.2, p. 38.

Definition of the sample space

One of the fundamental concepts in probability theory is a sample space Ω , that is, a space of all possible outcomes. Since we assume that the decisions of the environment and agent are random, we need to define the sample space. We now define the sample space of MDPs with discrete state and action spaces and deterministic rewards, as this is the case we consider most often in this thesis.

At each time step t the agent observes all of the choices made so far by both the environment and the agent itself. These choices include the previous states S_τ and actions A_τ , $\tau < t$, and the current state S_t . A sequence of such choices is called a *history* $h_t \in \Omega_t$ of the decision-making process at time step t :

$$h_t = (S_0, A_0, S_1, A_1, \dots, S_{t-1}, A_{t-1}, S_t).$$

For example, when we model a coin toss the sample space is $\Omega = \{h, t\}$, and when two coins are tossed, it is $\Omega = \{hh, ht, th, tt\}$.

Each history h_t belongs to a space Ω_t defined recursively as

$$\Omega_0 \triangleq \mathbb{S} \quad \text{and} \quad \Omega_{t+1} \triangleq \Omega_t \times \mathbb{A} \times \mathbb{S} \quad \text{for any } t \geq 0.$$

When the horizon T is finite, the sample space of the decision-making process is Ω_T . Each element $\omega \in \Omega_T$ is a sample path

$$\omega = (S_0, A_0, S_1, A_1, \dots, S_{T-1}, A_{T-1}, S_T),$$

and it shows the observed outcome at the end of the experiment, as presented in Figure 2.2. It contains all of the states S_t and actions A_t encountered by the agent in each of the decision epochs $t \in \mathbb{T}$. Additionally, it contains the final state S_T , which is observed by the agent, but no decision is made at time T . We can think of ω as the final history h_T .

E.g., when
 $\omega = (s_0, a_1, s_0)$,
 $S_0(\omega) = s_0$, $A_0(\omega) = a_1$,
and $S_1(\omega) = s_0$.

With a slight abuse of notation, we define the *coordinate maps* $S_t : \Omega_T \rightarrow \mathbb{S}$ for all $t \leq T$, and $A_t : \Omega_T \rightarrow \mathbb{A}$ for all $t < T$. Intuitively, these coordinate maps show what happened along any given sample path ω at any time step t . Similarly, we define the history map $h_t : \Omega_T \rightarrow \Omega_t$.

For infinite-horizon problems there is no final state and no final history. Nevertheless, in this case the infinite-horizon event space Ω_∞ can be defined as an infinite cartesian product

$$\Omega_\infty \triangleq \mathbb{S} \times \mathbb{A} \times \mathbb{S} \times \mathbb{A} \times \mathbb{S} \times \dots = \prod_{i=0}^{\infty} (\mathbb{S} \times \mathbb{A}).$$

Each decision-making process still follows one of these paths, but the agent is never able to observe which one.

When the rewards are random, the reward space is incorporated into the sample space in a similar way by following the model of Figure 2.1: $\Omega_T = \mathbb{S} \times \mathbb{A} \times \mathbb{S} \times \mathbb{R} \times \mathbb{A} \times \dots$. However, this sample space Ω_T is no longer discrete, complicating further presentation. The discrete nature of the sample space Ω_T is one of the reasons why we assume that the rewards are deterministic.

E.g., when flipping two
coins the outcomes are
 $\Omega = \{hh, ht, th, tt\}$.
The event $\{ht\}$ means
“1st coin landed
heads-up, and 2nd
tails-up.” If the coins are
distinguishable, this
event belongs to \mathcal{F} ,
otherwise not.

Having defined the sample space Ω_T for each horizon $T \leq \infty$, we equip it with a σ -algebra \mathcal{F}_T representing all of the events in our experiments. For example, an event “the decision-making process started in state s_0 ” is given by a set $\{\omega \mid S_0(\omega) = s_0\} \in \mathcal{F}$. Intuitively, a σ -algebra is required to show how much information is available to the agent by showing which outcomes the agent is able to distinguish from each other.

The Borel σ -algebra $\mathcal{B}(\Omega_T)$ is often used, which coincides with the power set of the sample space $\mathcal{F}_T = \mathcal{B}(\Omega_T) = 2^{\Omega_T}$ in the

discrete case. This means that any subset $\mathbb{F} \subseteq \Omega_T$ of sample paths is a valid event $\mathbb{F} \in \mathcal{F}_T$. In the continuous case, a different construction called the product σ -algebra is used instead.

Definition 2.3 | measurable space

The pair $(\mathbb{Y}, \mathcal{F})$ of a sample space and a σ -algebra on it is called a *measurable space*.

- The measurable space $(\Omega_T, \mathcal{F}_T)$ represents all of the outcomes of the decision-making process, and shows which of these outcomes are distinguishable by the observer, in our case, the agent.

2.1.2 Uncertainty

In epistemology, there are two kinds of uncertainty:

- *aleatoric uncertainty* that comes from the stochastic nature of the decision-making process: it is impossible to perfectly predict which of the sample paths $\omega \in \Omega_T$ the process will follow;
- *epistemic uncertainty* that comes from the lack of knowledge about the environment.

In the previous section, we defined the space Ω_T of possible outcomes of the decision-making process, known as sample paths. Some of these outcomes are more likely than others, depending on the choices done by the environment and—sometimes—the agent. As these choices can be done at random, the decision-making process has aleatoric uncertainty.

In planning, we assume that stochasticity is the only source of uncertainty. Therefore, we deal with aleatoric uncertainty only; knowing the MDP \mathcal{M}_T , it is possible to tell the likelihood of events $\mathbb{F} \in \mathcal{F}_T$ and base the decisions on this information. There are planning problems with epistemic uncertainty as well, for example, partially-observable MDPs; these models are outside of the scope of this thesis.

In reinforcement learning, the transition and reward kernels are not known to the agent. Therefore, epistemic uncertainty is always present.

In this section, we focus on the aleatoric uncertainty of the decision-making process. We start with the definition of probability. Then we define the transition and reward operators, and construct a probability measure on the measurable space $(\Omega_T, \mathcal{F}_T)$ of sample paths.

Named after Émile BOREL (1871–1956), a *Borel subset* is a subset that can be formed from open subsets by set differences, countable unions, and countable intersections. They are an important concept in measure theory, because measures—including probabilities—are well-defined on Borel sets.

From Latin *aleator* for *dice thrower*.

From Greek *επιστήμη* for *knowledge*.

Measure, probability, and random variables

In general, measures such as length and volume show how big different parts $\mathbb{F} \in \mathcal{F}$ of the space Ω are for a given measurable space (Ω, \mathcal{F}) . Formally, measures are defined as follows.

Definition 2.4 | measure

Given a measurable space (Ω, \mathcal{F}) , a function $\mu : \mathcal{F} \rightarrow \bar{\mathbb{R}}$ is called a *measure* if:

- it is non-negative, that is, for any $\mathbb{F} \in \mathcal{F}$, $\mu(\mathbb{F}) \geq 0$;
- the empty set has a null measure, $\mu(\emptyset) = 0$;
- it is σ -additive, that is, for any collection $(\mathbb{F}_i)_{i=0}^{\infty}$ of pair-wise disjoint sets $\mathbb{F}_i \in \mathcal{F}$, the total measure of their union is equal to the sum of measures of the individual sets \mathbb{F}_i :

$$\mu\left(\bigcup_{i=0}^{\infty} \mathbb{F}_i\right) = \sum_{i=0}^{\infty} \mu(\mathbb{F}_i).$$

Additionally, if the total measure is finite, $\mu(\Omega) < \infty$, then a measure μ is called *finite*.

Definition 2.5 | measure space

A measurable space (Ω, \mathcal{F}) equipped with a measure μ is called a *measure space* $(\Omega, \mathcal{F}, \mu)$.

Definition 2.6 | almost everywhere (a.e.)

Given a measurable space (Ω, \mathcal{F}) , a property holds *almost everywhere* with respect to a measure μ (μ -a.e.), if it holds for all elements of $\Omega \setminus \mathbb{F}$ for some set $\mathbb{F} \in \mathcal{F}$ of zero measure, $\mu(\mathbb{F}) = 0$.

Think of it this way: if something is true μ -a.e., then the elements for which it false may still exist, but they are so few that they can be neglected.

- Probability P is a particular type of measure that shows how likely different events $\mathbb{F} \in \mathcal{F}$ are, with “larger” events being more likely and the total probability $P(\Omega)$ being equal to 1. In this thesis, we mostly work with probabilities but other measures—such as the occupancy measure—are used as well occasionally.

Definition 2.7 | probability

A finite measure P is called a *probability measure* if the total measure is equal to one, $P(\Omega) = 1$. The measure space (Ω, \mathcal{F}, P) is called a *probability space*. Properties that hold P -a.e. are said to hold *almost surely* (a.s.).

- For a discrete sample space Ω , a probability measure can be defined by assigning a probability P to each sample $\omega \in \Omega$. Then for any

event $\mathbb{F} \in \mathcal{F}$ we define $P(\mathbb{F})$ as

$$P(\mathbb{F}) \triangleq \sum_{\omega \in \mathbb{F}} P(\omega), \quad \text{where } P : \Omega \rightarrow [0, 1] \text{ satisfies } \sum_{\omega \in \Omega} P(\omega) = 1.$$

Given a probability space (Ω, \mathcal{F}, P) , we can define random variables and their expected values. Intuitively, a random variable assigns some number to each sample $\omega \in \Omega$, and expected value shows the average value of a random variable, weighted by probability. An immediate reward R_t at time step t is an example of a random variable on the space Ω_T of sample paths. Random variables and expected values are formally defined as follows.

Definition 2.8 | discrete random variable

A *discrete random variable* $X : \Omega \rightarrow \mathbb{R}$ is a function that maps each sample $\omega \in \Omega$ from a discrete sample space Ω to a numeric value $X(\omega) \in \mathbb{R}$.

Definition 2.9 | expected value

Given a discrete probability space (Ω, \mathcal{F}, P) , $|\Omega| \leq \infty$, the *expected value* with respect to the probability measure P of a discrete random variable $X : \Omega \rightarrow \mathbb{R}$ is defined as

$$E_P[X] \triangleq \int_{\Omega} X \, dP = \sum_{\omega \in \Omega} X(\omega) \cdot P(\omega).$$

Remark 2.1

Expected value is not necessarily well-defined when the sample space has infinitely many elements.

When the sample space is not finite, the sum in Definition 2.9 contains infinitely many elements. Such sums are not always well-defined, as they depend on the summation order, and the sample space is not necessarily a totally ordered space. In general, expected value is well-defined only if at least one of $E_P[X^+]$ and $E_P[X^-]$ is finite, where

$$X^+(\omega) \triangleq \max\{X(\omega), 0\} \quad \text{and} \quad X^-(\omega) \triangleq \max\{-X(\omega), 0\}.$$

The four possible cases are summarized in Table 2.1. When the expected value is well-defined, it is equal to the difference between $E_P[X^+]$ and $E_P[X^-]$, $E_P[X] \triangleq E_P[X^+] - E_P[X^-]$. When both $E_P[X^+]$ and $E_P[X^-]$ are infinite, the expected value is ill-defined. This special case is often neglected because it does not arise as often as the other three cases. For example, if the sample space Ω is finite, only the first case may occur. In infinite-horizon problems the distinction becomes important.

Strictly speaking, the probability measure $P(\mathbb{F})$ is defined for events $\mathbb{F} \in \mathcal{F}$, not sample paths $\omega \in \Omega$. We should write $P(\{\omega\})$, but with a slight abuse of notation, we use $P(\omega)$ instead.

A famous example of an infinite series ill-defined in this sense is the alternating harmonic series $((-1)^{n-1}/n)_{n=1}^{\infty}$. For it, $1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots = \ln 2$, but if we follow every two positive elements with a negative one, $1 + \frac{1}{3} - \frac{1}{2} + \frac{1}{5} + \frac{1}{7} - \frac{1}{4} + \dots = \frac{3}{2} \ln 2$. The summation order changes the result.

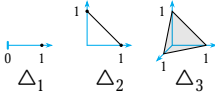
		$E[X^+]$	
$E[X^-]$		a	∞
	b	$a - b$	$+\infty$
	∞	$-\infty$	ill

Table 2.1: Definition of the expected value.

The constants a and b are some non-negative finite values.

α comes from Greek *αρχικός* for *initial*. It is also the initial letter of the Greek alphabet.

When \mathbb{Y} is a finite set of size n , $\Delta_{\mathbb{Y}}$ is homeomorphic to the n -dimensional probability simplex Δ_n , i.e., the set of vectors with non-negative coordinates summing into one.



Probabilities:

— p_t
 $p_s = \frac{1-p_t}{2}$

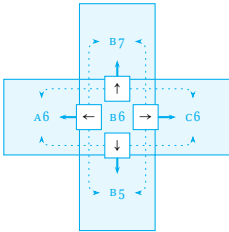


Figure 2.5: Transition probabilities in FrozenLake8×8-v0

Given an MDP, each decision-making process defines a unique probability measure P_π that depends on the decision strategy of the agent, known as a policy π . This probability also depends on the stochasticity in the environment, which is defined via the initial state distribution α and the transition p_t and reward r_t kernels. We continue with the definitions of these four objects. Then we use them to define the probability measure P_π .

Initial state distribution

Every sample path ω starts with an initial state $S_0(\omega) \in \mathbb{S}$ determined by the environment. It is drawn randomly from the *initial state distribution* $\alpha \in \Delta_{\mathbb{S}}$. $\Delta_{\mathbb{S}}$ denotes the set of all probability measures on the measurable space $(\mathbb{S}, \mathcal{B}(\mathbb{S}))$. When \mathbb{S} is discrete, the initial distribution is given by probabilities to start in each state $s \in \mathbb{S}$, also denoted by $\alpha(s) = \Pr[S_0 = s]$.

In FrozenLake8×8-v0, the agent always starts in the same state A8. Therefore, the initial state distribution is very simple:

$$\alpha(s) = \begin{cases} 1, & \text{if } s \text{ is the starting tile,} \\ 0, & \text{otherwise.} \end{cases}$$

Transition kernel

After an action A_t is chosen in a state S_t at a time step t , the environment transitions to a new state S_{t+1} according to a distribution given by a transition kernel p_t :

$$S_{t+1} \sim p_t(\cdot | S_t, A_t) \quad \text{for all } t \in \mathbb{T}, S_t \in \mathbb{S}, \text{ and } A_t \in A_p(S_t).$$

When states and actions are discrete, the transition kernel produces a probability mass function $p_t : \mathbb{X} \times \mathbb{S} \rightarrow [0, 1]$. Each $p_t(s' | s, a)$ shows the probability to transition from a state $s \in \mathbb{S}$ to a state $s' \in \mathbb{S}$ when taking an action $a \in A_p(s)$ at time step t :

$$p_t(s' | s, a) = \Pr[S_{t+1} = s' | S_t = s \text{ and } A_t = a].$$

When the transitions are stationary, we write $p(s' | s, a)$, understanding that $p_t = p$ for all decision epochs $t \in \mathbb{T}$.

Let us consider FrozenLake8×8-v0 again. When an action is permitted, the agent attempts to move in the direction indicated by this action. Unfortunately for them, the environment is slippery, and the desired state change happens only with some probability p_t to follow the desired direction. Otherwise, the agent slips and performs the movement in one of the two adjacent directions with

equal probabilities p_s , as shown in Figure 2.5. In FrozenLake8×8-v0 the probabilities to follow and slip are equal, $p_f = p_s = 1/3$. It is also possible to set $p_f = 1$, in which case the environment becomes deterministic and the desired action is always followed. We consider a more general case with arbitrary probability to follow p_f .

For example, if $p_f = 0.8$ and the agent attempts to take the action $a_0 = \leftarrow$ to move left in state $s_{17} = B6$, they will move to the left to the state $s_{16} = A6$, down to the state $s_{25} = B5$, or up to the state $s_9 = B7$ with probabilities 0.8, 0.1, and 0.1 respectively. At the edge and corner states, some of the transitions may lead to a non-existent states; in this case, the state does not change. For example, in the starting state $s_0 = A8$ if the agent takes the action $a_0 = \leftarrow$, they either stay in the same state with probability $p_f + p_s$ by attempting to either follow the desired action or to slip up, or they slip down with probability p_s .

The exact form of the transition probability function is somewhat complicated, so we do not present it here. Instead, transition probabilities can be computed and stored in a table, an excerpt from which can be seen in Table 2.2.

Reward kernel

Similarly to the transition kernel, when the rewards are stochastic, we define the reward kernels r_{t+1} . A reward R_{t+1} received by the agent after performing an action A_t in a state S_t during a decision epoch t and transitioning to a state S_{t+1} is distributed according to $r_{t+1}(\cdot \mid S_t, A_t, S_{t+1})$, that is, $R_{t+1} \sim r_{t+1}(\cdot \mid S_t, A_t, S_{t+1})$.

When the rewards are deterministic, they can be given by a function $R_{t+1} = r_{t+1}(S_t, A_t, S_{t+1})$. Moreover, for any possible state-action pair $(s, a) \in \mathbb{X}$ the distribution of a reward R_{t+1} depends on the transition distribution $p(\cdot \mid s, a)$ only, because it is the only thing that affects the next state $s' \sim p_t(\cdot \mid s, a)$. Therefore, we can use expected rewards $r_t : \mathbb{X} \rightarrow \mathbb{R}$ at time step t defined as

$$\begin{aligned} r_t(s, a) &= \mathbb{E}[r_{t+1}(s, a, s') \mid s' \sim p_t(\cdot \mid s, a)] \\ &= \sum_{s' \in \mathbb{S}} r_{t+1}(s, a, s') \cdot p_t(s' \mid s, a). \end{aligned} \quad (2.1)$$

By replacing the rewards $r_{t+1}(s, a, s')$ with the expected rewards $r_t(s, a)$, we obtain an equivalent MDP of Figure 2.2. Intuitively, we can do this because the expected reward $r_t(s, a)$ contains all of the information needed to make a decision at time t . This new MDP is equivalent to a decision-making model with deterministic

2.1 Markov Decision Processes

s	a	s'	$p(s' s, a)$
A8	\leftarrow	A8	$p_f + p_s$
		A7	p_s
		A8	p_s
	\downarrow	B8	p_s
		A7	p_f
		A8	p_s
	\rightarrow	B8	p_f
		A7	p_s
		A8	$p_f + p_s$
	\uparrow	A8	$p_f + p_s$
		B8	p_s
.....			
G1	\leftarrow	G2	p_s
		F1	p_f
		G1	p_s
	\downarrow	F1	p_s
		G1	p_f
		H1	p_s
	\rightarrow	G2	p_s
		G1	p_s
		H1	p_f
	\uparrow	G2	p_f
		F1	p_s
H1		p_s	

Table 2.2: Some of the non-zero transition probabilities in FrozenLake8×8-v0

the 2nd equality can be used for a discrete state space \mathbb{S} only

2 A Mathematical Model of Decision-Making

rewards received in the same epoch when the actions are chosen, which is presented in Figure 2.2. This justifies the simplified model, especially in planning, when the transition kernel p_t is known and the expected rewards can be computed.

In FrozenLake8×8-v0, the agent receives a unit reward when they reach the frisbee and no reward otherwise:

$$r(s, a, s') = \begin{cases} 1, & \text{if } s' \text{ is the frisbee tile,} \\ 0, & \text{otherwise.} \end{cases}$$

Using equation (2.1), the expected rewards $r(s, a)$ can be computed from the rewards $r(s, a, s')$; they are presented in Table 2.3.

s	a	$r(s, a)$
G1	↓	p_s
	→	p_f
	↑	p_s
H2	←	p_s
	↓	p_f
	→	p_s

Table 2.3: Non-zero expected rewards in FrozenLake8×8-v0.

Decision rules and policies

All of the information available to the agent during a decision epoch t is contained in a history h_t . Based on this information, the agent chooses which action $A_t \in \mathbb{A}$ to perform. This choice is called a *decision rule* π_t .

Definition 2.10 | decision rule

A (*history-dependent*) *decision rule* $\pi_t : \Omega_t \rightarrow \Delta_{\mathbb{A}}$ at time step t is a function that maps every history $h_t : \Omega_T \rightarrow \Omega_t$ to a distribution over actions such that $A_t \sim \pi_t(\cdot \mid h_t(\omega))$ for any time step $t \in \mathbb{T}$.

☞ When there are forbidden actions, the support $\text{supp } \pi_t$ of a decision rule π_t must be contained in the permitted actions,

$$\text{supp } \pi_t(\cdot \mid h_t(\omega)) \subseteq A_p(S_t(\omega)).$$

In general, a decision rule π_t of time step $t \in \mathbb{T}$ depends on the full history $h_t \in \Omega_t$, but it does not need to depend on all of the data therein. For example, a decision rule may depend on the current state $S_t(\omega)$ and decision epoch t only, in which case it is called *Markovian*.

Additionally, we distinguish between *randomized* and *deterministic* decision rules. A randomized decision rule is a probability distribution over actions, $\pi_t : \Omega_t \rightarrow \Delta_{\mathbb{A}}$. A deterministic decision rule always chooses the same action, also denoted by $\pi_t : \Omega_t \rightarrow \mathbb{A}$.

In the discrete case, randomized decision rules can be defined via probabilities to choose each action $a \in \mathbb{A}$ for each state $s \in \mathbb{S}$ during a decision epoch $t \in \mathbb{T}$. These probabilities are denoted by $\pi_t(a \mid s)$, to signify the dependence of the action choice by the agent on the observed state. With a slight abuse of notation,

The support $\text{supp } f$ of a function $f : \mathbb{Y} \rightarrow \mathbb{R}$ is a set of all points where it is non-zero, i.e., $\text{supp } f \triangleq \{y \mid f(y) \neq 0\}$.

deterministic decision rules are similarly denoted by $\pi_t(s) \in A_p(s)$. In this case, for any action a ,

$$\pi_t(a|s) = \begin{cases} 1, & \text{if } \pi_t(s) = a, \\ 0, & \text{otherwise.} \end{cases}$$

2.1 Markov Decision Processes

A combination of all of the decision rules fully defines the agent's behavior and is known as a policy.

Definition 2.11 | policy

The sequence $\pi = (\pi_t)_{t \in \mathbb{T}}$ of all decision rules is called a *policy*.

❧ Policies containing only Markovian decision rules are also called Markovian, otherwise we call them history-dependent. Furthermore, if decision rules of a Markovian policy do not depend on the time step, such a policy is called *stationary*. We denote the spaces of all history-dependent, Markovian and stationary policies by $\Pi_{\square H}$, $\Pi_{\square M}$ and $\Pi_{\square S}$ respectively. Policies containing only deterministic decision rules are also called deterministic, otherwise they are called randomized. We denote the spaces of such policies by $\Pi_{\square \square}$ and $\Pi_{\square \square}$ respectively. The hierarchy of all policy spaces is presented in Figure 2.6. The most general class of policies is randomized history-dependent policies $\Pi \triangleq \Pi_{RH}$, while deterministic stationary policies form the smallest class $\mathbb{D} \triangleq \Pi_{DS}$.

When a policy is given, it defines the way that actions a are chosen in the probability kernel $p_t(s'|s, a)$ and the expected reward $r(s, a)$. Therefore, it defines the expected rewards and transition probabilities as follows.

Definition 2.12 | expected rewards under a policy

The expected rewards $r_{\pi,t} : \mathbb{S} \rightarrow \mathbb{R}$ at time step t under a policy $\pi \in \Pi$ are functions given by

$$r_{\pi,t}(s) \triangleq \sum_{a \in A_p(s)} \pi_t(a|s) \cdot r_t(s, a).$$

Definition 2.13 | transition probabilities under a policy

The probabilities to reach state s' at time step $t+j$ by following policy π starting in state s at time step t are called the *j-step transition probabilities* $p_{\pi,t}^j(s'|s)$ and can be computed as

$$\begin{aligned} p_{\pi,t}^0(s'|s) &\triangleq \delta_{s,s'}, \\ p_{\pi,t}^1(s'|s) &\triangleq \sum_{a \in A_p(s)} \pi_t(a|s) \cdot p_t(s'|s, a), \end{aligned} \quad (2.2)$$

$$p_{\pi,t}^{j+1}(s''|s) \triangleq \sum_{s' \in \mathbb{S}} p_{\pi,t}^j(s'|s) \cdot p_{t+j,\pi}^1(s''|s'). \quad (2.3)$$

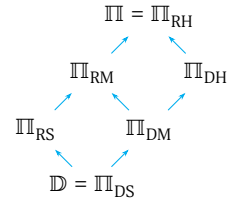


Figure 2.6: Hasse diagram for inclusion of policy spaces ($\mathbb{Y} \rightarrow \mathbb{Y}'$ means $\mathbb{Y} \subseteq \mathbb{Y}'$).

For one-step transitions, we write simply $p_{\pi,t}(s' | s)$. Similarly, in stationary MDPs we omit the time step and write r_π , p_π and p_π^i .

Policy-induced probability measure

The initial state distribution α and transition kernel p_t represent the random choices by the environment. Similarly, the policy π_t provides the random choices made by the agent. Using these, we can assign a probability P_π to each sample path $\omega = (S_0, A_0, S_1, A_1, S_2, \dots, A_{T-1}, S_T) \in \Omega_T$ as follows:

$$\begin{aligned} P_\pi(\omega) \triangleq & \alpha(S_0) \cdot \pi_0(A_0 | h_0) \cdot p_0(S_1 | S_0, A_0) \\ & \cdot \pi_1(A_1 | h_1) \cdot p_1(S_2 | S_1, A_1) \\ & \cdot \dots \\ & \cdot \pi_{T-1}(A_{T-1} | h_{T-1}) \cdot p_{T-1}(S_T | S_{T-1}, A_{T-1}). \end{aligned} \quad (2.4)$$

The fact that the function P_π is indeed a probability measure is due to the following proposition.

Proposition 2.1 * Ionescu-Tulcea extension theorem

Originally proven by Ionescu-Tulcea [1949] in a more general setting.

For both finite and infinite-horizon sample spaces Ω_T , $T \leq \infty$, the function P_π given by (2.4) exists uniquely and is a probability measure on the measurable space $(\Omega_T, \mathcal{F}_T)$.

This probability measure P_π defines the aleatoric uncertainty of any particular instance of the decision-making process: given the behavior of the environment and agent, it tells how likely each sample path is. It also allows to compare rewards collected by following different policies via their expected values, and can be used to reason about how good any given policy is.

Using this probability measure, for any random variable $X : \Omega_T \rightarrow \mathbb{R}$ its expected value on the probability space $(\Omega_T, \mathcal{F}_T, P_\pi)$ is denoted by $E_\pi[X]$ and can be computed as:

$$E_\pi[X] \triangleq E_{P_\pi}[X] = \sum_{\omega \in \Omega_T} X(\omega) \cdot P_\pi(\omega).$$

Often random variables depend not on the full sample path ω , but on a particular time step t . For example, this happens when the random variable X is the expected immediate reward,

$$X(S_t(\omega), A_t(\omega)) = r_t(S_t(\omega), A_t(\omega)).$$

In this case, the expected value can be evaluated using a simpler formula.

Proposition 2.2 * simplified expected value formula

If the random variable X depends not on the full sample path ω , but on a single state-action pair (S_t, A_t) only, its expected value can be computed as

$$\begin{aligned} E_\pi[X(S_t, A_t)] &= \sum_{(s_0, a_0, \dots, s_t) \in \Omega_t} \left(P(\mathbf{h}_t = (s_0, a_0, \dots, s_t)) \cdot \sum_{a_t \in A_p(s_t)} \pi_t(a_t | s_t) \cdot X(s_t, a_t) \right) \\ &= \sum_{s \in \mathbb{S}} \alpha(s) \cdot \sum_{s' \in \mathbb{S}} p_{\pi, 0}^t(s' | s) \cdot \sum_{a' \in A_p(s')} \pi_t(a' | s') \cdot X(s', a'). \end{aligned}$$

See Section 2.1.6 of Puterman [ibid.], where this formula is given in terms of conditional expectations.

A_p is the action permission function, see p. 22.

2.1.3 Optimal Behavior

We have defined the probabilistic model of the agent's behavior given by a policy. In this section, we show how different policies can be compared to each other. This allows us to define an optimal policy, that is, a policy such that no better policy exists.

Policy evaluation

For each sample path ω of the decision-making process, we can define some random variable $\Psi : \Omega_T \rightarrow \mathbb{R}$ that is called the *utility* [ibid.]. It measures how good any given outcome ω of the decision-making process is. The commonly used utilities are:

- the (undiscounted) total reward

$$\Psi(\omega) = \sum_{t \in \mathbb{T}} r_t(S_t(\omega), A_t(\omega));$$

- the γ -discounted total reward

$$\Psi(\omega) = \sum_{t \in \mathbb{T}} \gamma^t \cdot r_t(S_t(\omega), A_t(\omega)), \quad \text{where } 0 \leq \gamma < 1; \text{ and}$$

- the average reward

$$\Psi(\omega) = \begin{cases} \frac{1}{T} \cdot \sum_{t=0}^{T-1} r_t(S_t(\omega), A_t(\omega)) & \text{if } T < \infty, \\ \lim_{N \rightarrow \infty} \frac{1}{N} \cdot \sum_{t=0}^{N-1} r_t(S_t(\omega), A_t(\omega)), & \text{otherwise.} \end{cases}$$

Although all of these utilities are studied in the literature, the γ -discounted total reward is the most common. In this thesis, we will consider this utility only.

Having defined utilities, we can use them to evaluate policies, that is, to quantify how good—or bad—each given policy is. In the previous section, we showed that policies $\pi \in \Pi$ induce probability measures P_π on the measurable space $(\Omega_T, \mathcal{F}_T)$. Given this measures, we evaluate policies as follows.

Definition 2.14 | policy gain

The *policy gain* function $J : \Pi \rightarrow \mathbb{R}$ is equal to the expected utility Ψ on the probability space $(\Omega_T, \mathcal{F}_T, P_\pi)$, that is

$$J(\pi) \triangleq E_\pi[\Psi] \quad \text{for any } \pi \in \Pi.$$

☞ In particular, in the expected γ -discounted total reward case,

$$J(\pi) = E_\pi \left[\sum_{t \in \mathbb{T}} \gamma^t \cdot r_t(S_t, A_t) \right]. \quad (2.5)$$

Using Proposition 2.2 and Definition 2.12, the gain can be alternatively written as

$$\begin{aligned} J(\pi) &= E_\pi \left[\sum_{t \in \mathbb{T}} \gamma^t \cdot r_t(S_t, A_t) \right] = \sum_{\omega \in \Omega_T} \sum_{t \in \mathbb{T}} \gamma^t \cdot r_t(S_t(\omega), A_t(\omega)) \cdot P_\pi(\omega) \\ &\stackrel{\text{first change of the summation order}}{=} \sum_{t \in \mathbb{T}} \gamma^t \cdot \sum_{\omega \in \Omega_T} r_t(S_t(\omega), A_t(\omega)) \cdot P_\pi(\omega) = \sum_{t \in \mathbb{T}} \gamma^t \cdot E_\pi[r_t(S_t, A_t)] \\ &= \sum_{t \in \mathbb{T}} \gamma^t \cdot \sum_{s \in \mathbb{S}} \alpha(s) \cdot \sum_{s' \in \mathbb{S}} p_{\pi,0}^t(s' | s) \cdot r_{\pi,t}(s') \\ &\stackrel{\text{second change of the summation order}}{=} \sum_{s \in \mathbb{S}} \alpha(s) \cdot \sum_{t \in \mathbb{T}} \gamma^t \cdot \sum_{s' \in \mathbb{S}} p_{\pi,0}^t(s' | s) \cdot r_{\pi,t}(s'). \end{aligned} \quad (2.6)$$

It is important to note that the derivation of (2.6) involves two changes of summation order. These are only possible under one of the two conditions given by the following proposition.

Proposition 2.3 * Fubini–Tonelli theorem for sums

Adopted from
propositions 2 and 3 of
Lee et al. [2017].

For a measurable function $f : \mathbb{Y} \times \mathbb{Y}' \rightarrow \mathbb{R}$, if one of the sums

$$\sum_{(y,y') \in \mathbb{Y} \times \mathbb{Y}'} |f(y,y')|, \quad \sum_{y \in \mathbb{Y}} \sum_{y' \in \mathbb{Y}'} |f(y,y')|, \quad \text{and} \quad \sum_{y' \in \mathbb{Y}'} \sum_{y \in \mathbb{Y}} |f(y,y')|$$

is finite, then the function f is absolutely-summable. Moreover, if f is either absolutely-summable or non-negative, then

$$\sum_{(y,y') \in \mathbb{Y} \times \mathbb{Y}'} f(y,y') = \sum_{y \in \mathbb{Y}} \sum_{y' \in \mathbb{Y}'} f(y,y') = \sum_{y' \in \mathbb{Y}'} \sum_{y \in \mathbb{Y}} f(y,y').$$

☞ Proposition 2.3 holds for a wide range of models. In particular, it holds when the admissible control space \mathbb{X} is finite. In more complex settings, this may not be true. To guarantee that (2.6) holds, we introduce the notion of well-defined gain.

Definition 2.15 | well-defined gain

The gain $J(\pi)$ of a policy $\pi \in \Pi$ is *well-defined*, if the sum in (2.5) is absolutely summable or if the rewards are non-negative $r_t(s, a) \geq 0$ for all time steps $t \in \mathbb{T}$ and admissible state-action pairs $(s, a) \in \mathbb{X}$.

- By calculating the gain $J(\pi)$ of a policy $\pi \in \Pi$ we can tell how well it is expected to perform, but only when the gain is well-defined. Assuming all of the gains are well-defined, the best policies can be found by optimizing their gains.

2.1 Markov Decision Processes

Policy optimization

The problem is still missing an *optimality criterion*, that is, a definition of an optimal policy π_\star that the agent aims to find. Because each policy π has a defined gain $J(\pi)$, an optimal policy π_\star is simply a policy that achieves the highest possible gain J_\star :

$$J(\pi_\star) = J_\star \triangleq \sup_{\pi \in \Pi} J(\pi).$$

Therefore, the agent's goal is to act according to a policy π_\star that is a solution of the following optimization problem:

$$\begin{aligned} \text{find} \quad & \pi_\star \in \arg \max_{\pi \in \Pi} J(\pi), \\ \text{where} \quad & J(\pi) = \mathbb{E}_\pi \left[\sum_{t \in \mathbb{T}} \gamma^t \cdot r_t(S_t, A_t) \right]. \end{aligned} \quad (2.7)$$

Note that it is possible that multiple policies achieve the maximum gain J_\star . It is not necessary to find all of the optimal policies, knowing just one of these policies is sufficient.

Definition 2.16 | optimal policy

A solution $\pi_\star \in \Pi$ to the optimization problem (2.7) is called an *optimal policy*.

- This definition is different from what is commonly considered to be an optimal policy and what we call a universally optimal policy.

Definition 2.17 | universally optimal policy

A policy $\pi_\star \in \Pi$ is called a *universally optimal policy* if it is an optimal policy for any initial state distribution α .

Altman [1999] calls these policies uniformly optimal.

- Uniform optimality is a stronger notion in a sense that any universally optimal policy is always optimal by definition. While it is not immediately obvious that such policies would exist, this is true due to the following proposition.

Proposition 2.4 * existence of universally optimal policies

A policy $\pi_\star \in \Pi$ is *universally optimal* if and only if it is *optimal* for some initial state distribution α with full support, $\text{supp } \alpha = \mathbb{S}$.

Adopted from Lemma 1 of Lee et al. [2017].

- Proposition 2.4 allows to find a universally optimal policy by replacing the initial state distribution α with an arbitrary chosen one α' , given that it has full support, that is, $\alpha'(s) > 0$ for all states $s \in \mathbb{S}$. Because the resulting policy is invariant to changes in the initial state distribution, it provides an optimal action for

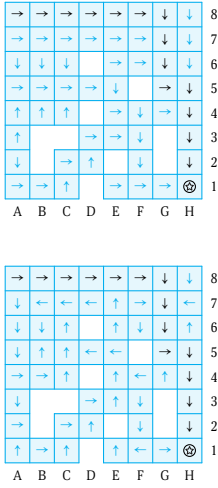


Figure 2.7: Examples of a universally optimal (top) and optimal (bottom) policies in a non-slippery frozen lake problem.

each state. Thus, universally optimal policies can be considered as more desirable than the optimal ones, and most of the research focuses on the class of universally optimal policies.

Nevertheless, when the initial state distribution is known and has a small support, it may not be necessary to find optimal actions for all of the states. For example, consider the policies presented in Figure 2.7. The second policy prescribes clearly suboptimal actions to some of the states; for example, if the agent ever finds themselves in E5, they will be stuck in an infinite loop, and starting in F3 leads to a hole. At the same time, this quality of the policy is irrelevant: the agent always starts in A8, and the policy always leads to optimal decisions from there.

The distinction between optimal and universally optimal policies is especially relevant in problems with large—for example, countably-infinite—state spaces. In these problems finding an optimal action for each state may be computationally intractable; therefore, we keep the distinction between optimal policies and universally optimal ones.

2.2 THE EXISTENCE OF OPTIMAL POLICIES

In the previous section, we formalized the decision-making problem and defined the optimal behavior as following a policy with the highest gain. But how can we guarantee that such a policy exists? And if it does, which of the policy classes does it belong to? In this section, we answer these questions.

2.2.1 Limitations of the Optimality Criterion

The definition of an optimal policy π_\star as a solution of (2.7) has limitations. In particular, it is not immediately obvious that:

- the gains $J(\pi)$ of policies $\pi \in \Pi$ are well-defined;
- the gains $J(\pi)$ are finite;
- the maximum in (2.7) is attained.

These cases are illustrated by the following examples.

Example 2.1 | an MDP with ill-defined gains

Consider an infinite-horizon γ -discounted MDP \mathcal{M}_∞ with one state $\mathbb{S} = \{s\}$ and one action $A_p(s) = \mathbb{A} = \{a\}$. The only policy π is to choose action a at every time step t , $\pi_t = a$. Therefore, it should be the optimal policy $\pi_\star = \pi$. Let the rewards be equal to

As there is only one state, $\alpha(s) = 1$ and $p(s|s, a) = 1$.

$r_t(s, a) = -1/\gamma^t$. The gain $J(\pi)$ for the only existing policy is

$$J(\pi) = \sum_{t=0}^{\infty} \gamma^t \cdot \left(-\frac{1}{\gamma^t}\right) = \sum_{t=0}^{\infty} (-1)^t = 1 - 1 + 1 - 1 + \dots$$

This is the so-called Grandi's series which is known to be divergent.

Indeed, both partial sums $J^+(\pi)$ and $J^-(\pi)$ are infinite,

$$J^+(\pi) = 1 + 0 + 1 + \dots = \infty \quad \text{and} \quad J^-(\pi) = 0 + 1 + 0 + \dots = \infty,$$

and therefore $J(\pi)$ is ill-defined as per Remark 2.1. As a result, the definition of an optimal policy π_\star cannot be used.

See p. 27.

Example 2.2 | an MDP with infinite values of all policies

Consider a γ -discounted MDP \mathfrak{M}_T with one state $\mathbb{S} = \{s\}$ and two actions $A_p(s) = \mathbb{A} = \{a_0, a_1\}$. The rewards are $r_t(s, a_0) = \gamma^{-t}$ and $r_t(s, a_1) = 2\gamma^{-t}$. In the finite-horizon case, the worst policy π_- is to choose a_0 at all time steps, and the best policy π_+ is to choose a_1 instead, while all other policies π have their gains in the range

$$J(\pi_-) = \underbrace{1 + 1 + \dots}_{T \text{ times}} = T \leq J(\pi) \leq J(\pi_+) = \underbrace{2 + 2 + \dots}_{T \text{ times}} = 2T.$$

In the infinite horizon case however, both bounds are well-defined but infinite, $J(\pi_-) = J(\pi_+) = \infty$, and therefore all of the policies have the same well-defined gain. At the same time, intuitively π_+ is still better than any other policy, because at any finite horizon it yields the largest total reward.

Example 2.3 | an MDP without an optimal policy

Consider an infinite-horizon γ -discounted MDP \mathfrak{M}_∞ with one state $\mathbb{S} = \{s\}$ and countably-infinite number of actions $A_p(s) = \mathbb{A} = \mathbb{N}$. Let the rewards be equal to $r(s, a) = 1 - a^{-1}$. The rewards are non-negative and thus the gain $J(\pi)$ is well-defined for any policy $\pi \in \Pi$ by Proposition 2.3. Moreover, the rewards are bounded by $0 < r(s, a) < 1$, and the gain $J(\pi)$ is finite:

$$\begin{aligned} \Psi(\omega) &= \sum_{t=0}^{\infty} \gamma^t \cdot r(S_t(\omega), A_t(\omega)) < \sum_{t=0}^{\infty} \gamma^t \cdot 1 = \frac{1}{1-\gamma} < \infty, \quad \text{and} \\ J(\pi) &= \sum_{\omega \in \Omega_\infty} \Psi(\omega) \cdot P_\pi(\omega) < \frac{1}{1-\gamma} \cdot \sum_{\omega \in \Omega_\infty} P_\pi(\omega) = \frac{1}{1-\gamma} \cdot P_\pi(\Omega_\infty) = \frac{1}{1-\gamma}, \quad \triangleright P_\pi(\Omega_\infty) = 1 \text{ by definition} \end{aligned}$$

and therefore the supremum J_\star exists finitely. At the same time, an optimal policy does not exist. Indeed, for any policy π , let us pick an arbitrary decision rule $\pi_t(s) = a$ and change it to $a + 1$. This increases the gain of the policy, as $r(s, a + 1) > r(s, a)$. Therefore, for any policy there exists a different policy with a greater gain, and the maximum is never attained in (2.7).

☞ These examples show that the decision-making problem can be ill-posed. In this case, optimal policies are undefined or non-existent, or—in the case of Example 2.2—all of the policies are considered optimal. It is thus necessary to be able to ensure that the decision-making problem is well-posed in the first place.

2.2.2 Models with Uniformly Bounded Rewards

Let us address the issues discussed in the previous section. The gain $J(\pi)$ is well-defined when all of the rewards are either non-negative or non-positive, as per Remark 2.1. But even in this case, it is possible that all of the policies have infinite gains and no comparison can be made between them. The following condition is one of the ways to ensure that both $E_\pi[\Psi^+]$ and $E_\pi[\Psi^-]$ are finite and therefore so is the gain $J(\pi)$ for any policy $\pi \in \Pi$.

$$\Psi^\pm(\omega) = \max\{\pm\Psi(\omega), 0\}.$$

Condition 2.1 | uniform absolute reward bound

For all time steps $t \in \mathbb{T}$, states $s \in \mathbb{S}$, and actions $a \in A_p(s)$, the reward function is absolutely bounded by a constant $w \in \mathbb{R}$:

$$|r_t(s, a)| \leq w.$$

☞ Indeed, in this case

$$\begin{aligned} \text{by the triangle inequality} \quad & |\Psi(\omega)| \leq \sum_{t \in \mathbb{T}} \gamma^t \cdot |r_t(S_t(\omega), A_t(\omega))| \leq \sum_{t \in \mathbb{T}} \gamma^t \cdot w \leq \frac{w}{1-\gamma} < \infty, \quad \text{and} \\ |a + b + \dots| \leq |a| + |b| + \dots \quad & |J(\pi)| \leq \sum_{\omega \in \Omega_T} |\Psi(\omega)| \cdot P_\pi(\omega) \leq \frac{w}{1-\gamma} \cdot \sum_{\omega \in \Omega_T} P_\pi(\omega) = \frac{w}{1-\gamma} \cdot P_\pi(\Omega_T) = \frac{w}{1-\gamma}. \end{aligned}$$

Therefore, Condition 2.1 and Proposition 2.3 guarantee that the gain $J(\pi)$ of any policy $\pi \in \Pi$ is well-defined and finite.

Sometimes, a different condition is used instead.

Condition 2.2 | uniform reward bounds

For all time steps $t \in \mathbb{T}$, states $s \in \mathbb{S}$, and actions $a \in A_p(s)$, the reward function is bounded by constants $r_-, r_+ \in \mathbb{R}$:

$$r_- \leq r_t(s, a) \leq r_+.$$

☞ The two conditions imply each other and therefore are equivalent. Indeed, one can use $r_\pm = \pm w$ given an absolute bound w , and conversely, $w = \max\{-r_-, r_+\}$ given two bounds r_\pm . Depending on the problem, either one or the other can be used.

Even if Condition 2.1 holds, the maximum in (2.7) may still be unattainable as illustrated by Example 2.3. To address this issue, we additionally impose the following condition on an MDP.

Condition 2.3 | action-space semicontinuity

Let the actions of an MDP satisfy the following three statements:

- the set of permitted actions $A_p(s)$ is compact for each state $s \in \mathbb{S}$;
- the reward function $r_t(s, \cdot)$ is upper semicontinuous for each state $s \in \mathbb{S}$ and time step $t \in \mathbb{T}$;
- the transition function $p_t(s' | s, \cdot)$ is lower semicontinuous for each pair of states $s, s' \in \mathbb{S}$ and time step $t \in \mathbb{T}$.

Under these conditions, there exists an optimal policy in both finite and infinite-horizon MDPs as stated by the following two propositions.

Proposition 2.5 * optimality in finite-horizon MDPs

Let Conditions 2.1 and 2.3 hold for a finite-horizon Markov decision process \mathfrak{M}_T with a countable state space, $|\mathbb{S}| \leq \infty$. Then there exists a deterministic Markovian policy $\pi \in \Pi_{DM}$ that is optimal.

Proposition 2.6 * optimality in stationary infinite-horizon MDPs

Let Conditions 2.1 and 2.3 hold for a stationary infinite-horizon Markov decision process \mathfrak{M}_∞ with a countable state space, $|\mathbb{S}| \leq \infty$. Then there exists a deterministic stationary policy $\pi \in \mathbb{D}$ that is optimal.

These propositions provide valuable insight into the nature of the decision-making problems. Not only do they tell us that optimal policies exist, but in both cases we know the class that they belong to: deterministic Markov and deterministic stationary policies for finite-horizon and infinite-horizon stationary problems respectively. Both of these statements allow us to narrow down the search space of the optimization problem (2.7) when searching for an optimal policy.

Propositions 2.5 and 2.6 do not directly cover infinite-horizon non-stationary problems. However, these problems can be reduced to stationary problems as follows.

Definition 2.18 | time-augmented MDP

Given a non-stationary MDP $\mathfrak{M}_T, T \leq \infty$, the *time-augmented stationary* MDP $\tilde{\mathfrak{M}}_T$ is produced by augmenting the state space \mathbb{S} with the time space \mathbb{T} . The new state space $\tilde{\mathbb{S}}$ is the cartesian product $\tilde{\mathbb{S}} \triangleq \mathbb{S} \times \mathbb{T}$. The initial state distributions $\tilde{\alpha}$, transition probabilities \tilde{p} , and rewards \tilde{r} of the new MDP for all states $\tilde{s} = (s, t)$ and $\tilde{s}' = (s', t')$ are equal to

$$\tilde{\alpha}(\tilde{s}) \triangleq \delta_{t,0} \cdot \alpha(s), \quad (2.8)$$

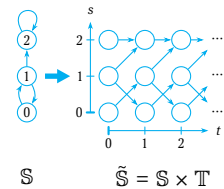
2.2 The Existence of Optimal Policies

In particular, Condition 2.3 holds if the set of permitted actions $A_p(s)$ is finite for each state $s \in \mathbb{S}$.

Adopted from Proposition 4.4.3 of Puterman [1994].

Adopted from Theorem 6.2.10 of Puterman [ibid.].

See Figure 2.6 p. 31.



$$\delta_{ij} \triangleq \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

$$\tilde{p}(\tilde{s}' | \tilde{s}, a) \triangleq \delta_{t+1, t'} \cdot p_t(s' | s, a), \quad (2.9)$$

$$\tilde{r}(\tilde{s}' | \tilde{s}, a) \triangleq \delta_{t+1, t'} \cdot r_{t+1}(s' | s, a), \quad \text{and} \quad (2.10)$$

$$\tilde{r}(\tilde{s}, a) \triangleq r_t(s, a). \quad (2.11)$$

2 A Mathematical Model of Decision-

Making

From this definition, it is easy to see that if the original discrete non-stationary MDP is finite-horizon with a finite state space \mathbb{S} , the new stationary problem also has a finite state space $\tilde{\mathbb{S}}$, otherwise it is a countably-infinite MDP. It is not immediately clear that policies of the original problem correspond to policies of its time-augmented version and vice versa. This fact is established by the following lemma.

Lemma 2.7 * stationary reformulation equivalence

All policies of a non-stationary MDP $\mathfrak{M}_T, T \leq \infty$ can be represented in time-augmented version $\tilde{\mathfrak{M}}_T$ by policies of the same gain, assuming that the gain is well-defined.

The proof of Lemma 2.7 is presented in Section A.1. Because of this equivalence, we can focus on stationary MDPs only. When necessary, we can translate their properties back to non-stationary MDPs via (2.8), (2.9), and (2.11).

For example, using the definition of the new state space $\tilde{\mathbb{S}}$, we see that every deterministic (or randomized) Markovian policy π of the original non-stationary MDP corresponds to a deterministic (respectively, randomized) stationary policy $\tilde{\pi}$ of the augmented problem: $\pi_t(s) = \tilde{\pi}(s, t)$ and $\pi_t(a | s) = \tilde{\pi}(a | s, t)$. Therefore, the following statement follows from Proposition 2.6.

Corollary 2.8 * optimality in non-stationary ∞ -horizon MDPs

Let Conditions 2.1 and 2.3 hold for a non-stationary infinite-horizon Markov decision process \mathfrak{M}_∞ with a countable state space, $|\mathbb{S}| \leq \infty$. Then there exists a deterministic Markovian policy $\pi \in \Pi_{\text{DM}}$ that is optimal.

The results of Propositions 2.5 and 2.6 and Corollary 2.8 are summarized in Table 2.4. The search space in the decision-making problem (2.7) can be reduced from all policies Π to deterministic Markovian ones $\Pi_{\text{DM}} \subseteq \Pi$ for any type of problem.

T	stationary	
	yes	no
$< \infty$	Π_{DM}	Π_{DM}
$= \infty$	\mathbb{D}	Π_{DM}

Table 2.4: Smallest policy class containing an optimal policy. \mathbb{D} and Π_{DM} are the classed of deterministic stationary and deterministic Markovian policies, see Definition 2.11, p. 31.

2.3 FINDING OPTIMAL POLICIES

In the previous section, we established that an optimal deterministic Markovian policy always exists in discrete MDPs with either

finite or countably-infinite admissible control spaces. Unfortunately, we still do not know how to find such a policy. In this section, we show how it can be done by computing either *value functions* or *occupancies*.

2.3 Finding Optimal Policies

2.3.1 State Value Functions

A common approach to policy evaluation and optimization involves computing the so-called policy values v_π and optimal values v_\star . They are defined as follows.

Definition 2.19 | value under a policy

For each state $s \in \mathbb{S}$, the *value function* $v_{\pi,t} : \mathbb{S} \rightarrow \bar{\mathbb{R}}$ under a policy $\pi \in \Pi$ shows the expected γ -discounted total reward collected when starting in that state at time step $t \in \mathbb{T}$ and following the policy π from then on:

$$v_{\pi,t}(s) \triangleq \sum_{\tau=t}^T \gamma^{\tau-t} \cdot \sum_{s' \in \mathbb{S}} p_{\pi,\tau}^{t-\tau}(s' | s) \cdot r_{\pi,\tau}(s').$$

Using the state value function $v_{\pi,t}(s)$, we can write (2.6) as

$$J(\pi) = \sum_{s \in \mathbb{S}} \alpha(s) \cdot v_{\pi,0}(s). \quad (2.12)$$

Thus, policy evaluation can be done via the policy value function. In the finite-horizon case, the policy value function can be computed recursively as

$$v_{\pi,T}(s) = 0 \quad \text{and} \quad v_{\pi,t}(s) = r_{\pi,t}(s) + \gamma \cdot \sum_{s' \in \mathbb{S}} p_{\pi,t}(s' | s) \cdot v_{\pi,t+1}(s').$$

In the infinite-horizon stationary case, the state values of stationary policies are subject to the following recurrence

$$v_\pi(s) = r_\pi(s) + \gamma \cdot \sum_{s' \in \mathbb{S}} p_\pi(s' | s) \cdot v_\pi(s'). \quad (2.13)$$

Notably, the state values do not depend on the time step t anymore. This is not the case in the finite-horizon case, because the problem is time-homogenous: for each time step t the process continues for the same number of time steps, infinitely many.

Similarly to policy values for policy evaluation, optimal values can be used for policy optimization.

Definition 2.20 | optimal value

The *optimal value function* $v_{\star,t} : \mathbb{S} \rightarrow \bar{\mathbb{R}}$ shows the highest possible expected γ -discounted total reward collected when starting in a state $s \in \mathbb{S}$ at time step $t \in \mathbb{T}$, $v_{\star,t}(s) \triangleq \sup_{\pi \in \Pi} v_{\pi,t}(s)$.

☞ If the values are maximized, the optimal policy gain is achieved:

2 A Mathematical Model of Decision-Making

$$J_{\star} = \sum_{s \in \mathbb{S}} \alpha(s) \cdot v_{\star,0}(s).$$

Similarly to the state values v_{π} under a policy π , we can find the optimal state value function v_{\star} as

$$\begin{aligned} v_{\star,T}(s) &= 0, \\ v_{\star,t}(s) &= \max_{a \in A_p(s)} \left\{ r_t(s, a) + \gamma \cdot \sum_{s' \in \mathbb{S}} p_t(s' | s, a) \cdot v_{\star,t+1}(s') \right\}. \end{aligned} \quad (2.14)$$

In the stationary infinite-horizon case this reduces to

$$v_{\star}(s) = \max_{a \in A_p(s)} \left\{ r(s, a) + \gamma \cdot \sum_{s' \in \mathbb{S}} p(s' | s, a) \cdot v_{\star}(s') \right\}. \quad (2.15)$$

Both equations assume that the number of permitted actions is always finite, $|A_p(s)| < \infty$ for any $s \in \mathbb{S}$. This allows the supremum in Definition 2.20 to be achieved; therefore, it can be replaced with the maximum. The actions that achieve the maximums in (2.14) and (2.15) are the optimal actions.

2.3.2 The Bellman Operators

While the recurrences (2.13) and (2.15) hold in the infinite-horizon case, they cannot be used to compute the value functions directly. Instead, iterative schemes presented in this section can be used.

The right-hand sides of (2.13) and (2.15) can be presented using the Bellman operator \mathcal{L}_{π} of a policy π and the optimal Bellman operator \mathcal{L}_{\star} . These operators are defined as follows.

Definition 2.21 | policy Bellman operator

For any Markov randomized policy $\pi \in \Pi_{\text{RM}}$ and function $y : \mathbb{S} \rightarrow \bar{\mathbb{R}}$, the *Bellman operator* $\mathcal{L}_{\pi} : \bar{\mathbb{R}}^{\mathbb{S}} \rightarrow \bar{\mathbb{R}}^{\mathbb{S}}$ is defined as

$$[\mathcal{L}_{\pi}y](s) \triangleq r_{\pi}(s) + \gamma \cdot \sum_{s' \in \mathbb{S}} p_{\pi}(s' | s) \cdot y(s'). \quad (2.16)$$

☞ We can write (2.16) more compactly by introducing the transition operator $\mathcal{T}_{\pi} : \bar{\mathbb{R}}^{\mathbb{S}} \rightarrow \bar{\mathbb{R}}^{\mathbb{S}}$ under a policy π :

$$[\mathcal{T}_{\pi}y](s) \triangleq \sum_{s' \in \mathbb{S}} p_{\pi}(s' | s) \cdot y(s') \quad \text{for any } y : \mathbb{X} \rightarrow \bar{\mathbb{R}}.$$

Since r_{π} is also a function in $\bar{\mathbb{R}}^{\mathbb{S}}$, $\mathcal{L}_{\pi}y = r_{\pi} + \gamma \cdot \mathcal{T}_{\pi}y$.

Definition 2.22 | optimal Bellman operator

The *optimal Bellman operator* $\mathcal{L}_{\star} : \bar{\mathbb{R}}^{\mathbb{S}} \rightarrow \bar{\mathbb{R}}^{\mathbb{S}}$ is

$$[\mathcal{L}_{\star}y](s) \triangleq \max_{a \in A_p(s)} \left\{ r(s, a) + \gamma \cdot \sum_{s' \in \mathbb{S}} p(s' | s, a) \cdot y(s') \right\}. \quad (2.17)$$

Richard BELLMAN (1920–1984) introduced dynamic programming as a way to solve optimal control problems.

$\mathbb{Y}^{\mathbb{Y}'}$ $\triangleq \{f | f : \mathbb{Y}' \rightarrow \mathbb{Y}\}$ is the space of all functions from a space \mathbb{Y}' to another space \mathbb{Y} . Equality of functions is understood pointwise, that is, it holds for any value of the argument $s \in \mathbb{S}$.

☞ To make the notation succinct, we introduce the maximization $\mathcal{M} : \bar{\mathbb{R}}^{\mathbb{X}} \rightarrow \bar{\mathbb{R}}^{\mathbb{S}}$ and transition $\mathcal{T} : \bar{\mathbb{R}}^{\mathbb{S}} \rightarrow \bar{\mathbb{R}}^{\mathbb{X}}$ operators:

$$[\mathcal{M}y](s) \triangleq \max_{a \in A_p(s)} y(s, a), \quad (2.18)$$

$$[\mathcal{T}y'](s, a) \triangleq \sum_{s' \in \mathbb{S}} p(s' | s, a) \cdot y'(s') \quad (2.19)$$

for any functions $y : \mathbb{X} \rightarrow \bar{\mathbb{R}}$ and $y' : \mathbb{S} \rightarrow \bar{\mathbb{R}}$. Using these operators, we can write equation (2.17) as

$$\mathcal{L}_\star y = \mathcal{M}(r + \gamma \cdot \mathcal{T}y). \quad (2.20)$$

The value function v_π of any randomized Markov policy π and the optimal value function v_\star are the fixed points the policy Bellman operator \mathcal{L}_π and the optimal Bellman operator \mathcal{L}_\star respectively. So far, we have implicitly assumed that v_π exists uniquely. The existence and uniqueness of the value functions v_π and v_\star can be proven using the Banach fixed-point theorem.

Definition 2.23 | fixed point

An element $y \in \mathbb{Y}$ of a space \mathbb{Y} is called a *fixed point* of an operator $\mathcal{Y} : \mathbb{Y} \rightarrow \mathbb{Y}$, if $\mathcal{Y}y = y$.

Definition 2.24 | contraction & Lipschitz constant

Consider a metric space (\mathbb{Y}, d) . An operator $\mathcal{Y} : \mathbb{Y} \rightarrow \mathbb{Y}$ is called a *contraction* in the space \mathbb{Y} with respect to the metric d if there exists a constant $0 \leq \kappa < 1$ called the *Lipschitz constant* of \mathcal{Y} , such that $d(\mathcal{Y}y - \mathcal{Y}y') \leq \kappa \cdot d(y - y')$ for any $y, y' \in \mathbb{Y}$.

Definition 2.25 | complete metric space

A metric space (\mathbb{Y}, d) is *complete* if every Cauchy sequence $(y)_{i=0}^\infty$ has a limit in the space \mathbb{Y} . A sequence is called *Cauchy* if its elements become arbitrary close to each other, that is, for any constant $\varepsilon > 0$ there exists a constant $k > 0$ such that $d(y_i, y_j) < \varepsilon$ for all $i, j > k$.

Proposition 2.9 * Banach fixed-point theorem

If \mathbb{Y} is a complete metric space, then every contraction $\mathcal{Y} : \mathbb{Y} \rightarrow \mathbb{Y}$ has a unique fixed point $y \in \mathbb{Y}$. Moreover, this fixed point y can be found as

$$y = \lim_{n \rightarrow \infty} \mathcal{Y}^n y' \quad \text{for any } y' \in \mathbb{Y}.$$

Definition 2.26 | Banach space

A normed space $(\mathbb{Y}, \|\cdot\|)$ is called a *Banach space* if it is complete with respect to the $\|\cdot\|$ -induced metric $d(y, y') \triangleq \|y - y'\|$.

2.3 Finding Optimal Policies

Rudolf LIPSCHITZ (1832–1903) proposed this property for constants $\kappa \geq 0$ as a stronger notion of continuity of functions.

Baron Augustin-Louis CAUCHY (1789–1857) introduced into calculus the infinitesimals, i.e., arbitrary small numbers.

Originally proven by Banach [1922].

Stefan BANACH (1892–1945), the founder of modern functional analysis, studied these spaces extensively.

Remark 2.2

A fortiori, Proposition 2.9 holds for Banach spaces.

2 A Mathematical Model of Decision-Making

In general, the space $\mathbb{R}^{\mathbb{S}}$ of real-valued and infinite-valued functions on \mathbb{S} is not a Banach space and therefore the Bellman operator \mathcal{L}_π is not a contraction mapping on it. It does, however, become a contraction if we restrict the space of allowed value functions to a Banach space, usually, an L^p -space.

Definition 2.27 | L^p -space

E.g., L^1 is the space of all absolutely-summable functions, and L^∞ is the space of all absolutely bounded functions.

Using L^p -spaces, Condition 2.1 can be written as $r_t \in L^\infty(\mathbb{X})$.

The counting measure $\#$ returns the number of elements in a subset $\mathbb{F} \in \mathcal{F}$: $\#(\mathbb{F}) = |\mathbb{F}|$; integration with respect to $\#$ becomes summation.

Given a measure space $(\mathbb{Y}, \mathcal{F}, \mu)$, the L^p -space $L^p(\mathbb{Y}, \mu)$ is the space of real-valued measurable functions $f : \mathbb{Y} \rightarrow \mathbb{R}$ on \mathbb{Y} with finite p -norm defined as

$$\|f\|_p \triangleq \left(\int_{\mathbb{Y}} |f(y)|^p d\mu \right)^{1/p}.$$

For countable measure spaces $(\mathbb{Y}, \mathcal{B}(\mathbb{Y}), \#)$, $|\mathbb{Y}| \leq \aleph_0$ with the counting measure $\#$, we write $L^p(\mathbb{Y}) \triangleq L^p(\mathbb{Y}, \#)$. In this case,

$$\|f\|_p = \left(\sum_{y \in \mathbb{Y}} |f(y)|^p \right)^{1/p}, \quad \text{if } 1 \leq p < \infty, \quad \text{and} \quad \|f\|_\infty = \sup_{y \in \mathbb{Y}} |f(y)|.$$

Uniformly bounded rewards

Consider the space $L^\infty(\mathbb{S})$ of uniformly bounded functions on the state space \mathbb{S} . Under Condition 2.1, the reward r_π under policy π has a finite supremum-norm, $\|r_\pi\|_\infty \leq w$. Therefore, the state value function v_π of any policy π has a finite supremum-norm:

$$\|v_\pi\|_\infty \leq \sum_{t \in \mathbb{T}} \gamma^t \|r_\pi\|_\infty = \frac{1-\gamma^T}{1-\gamma} \cdot w, \quad \text{therefore} \quad v_\pi \in L^\infty(\mathbb{S}).$$

Moreover, the following proposition holds.

Proposition 2.10 * Bellman operators are contractions

Adopted from Proposition 6.2.4 of Puterman, 1994

Under Condition 2.1, Bellman operators \mathcal{L}_π and \mathcal{L}_\star are contractions in the space $L^\infty(\mathbb{S})$ of uniformly bounded functions.

This proposition and the fixed-point theorem prove that the state value function v_π for any policy π and the optimal state value function v_\star , given by (2.14) and (2.15), exist uniquely in the space of uniformly bounded functions $L^\infty(\mathbb{S})$. Moreover, the fixed-point theorem provides a way to compute the values v_π and v_\star iteratively. This computation scheme is known as *value iteration*.

2.3.3 Occupancy Measure

The main difficulty in evaluating policies comes from the fact that gains $J(\pi)$ are defined as expected values $E_\pi[\cdot]$ on the probability space $(\Omega_T, \mathcal{F}_T, P_\pi)$. In general, both the time space \mathbb{T} and the sample space Ω_T can be countably-infinite. Therefore, gains expand into double countably-infinite sums:

$$J(\pi) = \sum_{\omega \in \Omega_T} \sum_{t \in \mathbb{T}} \gamma^t \cdot r(S_t(\omega), A_t(\omega)) \cdot P_\pi(\omega).$$

To circumvent this limitation, the probability space $(\Omega_T, \mathcal{F}, P)$ of the problem can be converted to a different measure space $(\mathbb{S} \times \mathbb{A}, \mathcal{B}(\mathbb{S} \times \mathbb{A}), z_\pi)$. If the state-action space $\mathbb{S} \times \mathbb{A}$ is finite, this substantially simplifies the optimization problem. The new measure z_π is defined as follows.

Definition 2.28 | occupancy measure

The (γ -discounted) occupancy $z_\pi : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}_+$ of a policy $\pi \in \Pi$ is the expected γ -discounted number of visits of state-action pairs, that is, for any measurable sets $\mathbb{S}' \in \mathcal{B}(\mathbb{S})$ and $\mathbb{A}' \in \mathcal{B}(\mathbb{A})$

$$z_\pi(\mathbb{S}', \mathbb{A}') = E_\pi \left[\sum_{t \in \mathbb{T}} \gamma^t \cdot \mathbb{I}_{\{S_t \in \mathbb{S}'\}} \cdot \mathbb{I}_{\{A_t \in \mathbb{A}'\}} \right]. \quad (2.21)$$

2.3 Finding Optimal Policies

See (2.5), p. 34.

Adapted from
Definition 5 of Laroché
et al. [ibid.].

🌀 In general, the occupancy measure is not a probability measure, but it is still finite, so it can be thought of as “scaled probability” that sums into something other than one. This fact is due to the following lemma.

Lemma 2.11 * finiteness of occupancy measure

In a γ -discounted MDP $\mathfrak{M}_T, T \leq \infty$, any policy $\pi \in \Pi$ induces a finite occupancy z_π measure.

Proof. Because any state S_t belongs to the state space, $\mathbb{I}_{\{S_t \in \mathbb{S}\}} = 1$; similarly, $\mathbb{I}_{\{A_t \in \mathbb{A}\}} = 1$. Therefore, (2.21) becomes

$$z_\pi(\mathbb{S}, \mathbb{A}) = E_\pi \left[\sum_{t \in \mathbb{T}} \gamma^t \right] = E_\pi \left[\frac{1 - \gamma^T}{1 - \gamma} \right] = \frac{1 - \gamma^T}{1 - \gamma}. \quad \text{QED}$$

🌀 In problems with discrete state-action spaces, the occupancy measure can be defined per state-action pair. This results in the so-called visitation function. With a slight abuse of notation, we denote the visitation function z_π and refer to it as the occupancy.

Definition 2.29 | visitation (occupancy function)

In a γ -discounted MDP $\mathfrak{M}_T, T \leq \infty$ with a discrete state-action

space $\mathbb{S} \times \mathbb{A}$, the (γ -discounted) visitation or occupancy function z_π of a policy $\pi \in \Pi$ is defined as

$$z_\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t \in \mathbb{T}} \gamma^t \cdot \delta_{S_t, s} \cdot \delta_{A_t, a} \right].$$

Remark 2.3

If a state-action pair (s, a) is not permitted, $(s, a) \notin \mathbb{X}$, $\delta_{S_t, s} \cdot \delta_{A_t, a}$ is always equal to zero, and therefore $z_\pi(s, a) = 0$.

- ☞ Having defined the occupancy measure, we state that the gain $J(\pi)$ of any policy $\pi \in \Pi$ can be computed in terms of the occupancy measure z_π that it induces. This is a well known result formalized by the following proposition.

Proposition 2.12 * equivalence

Adopted from Lemma 1 of Larocche et al. [2022].

If $J(\pi)$ is well-defined, then $J(\pi) = \int_{\mathbb{S} \times \mathbb{A}} r(s, a) \cdot z_\pi(ds, da)$.

- ☞ For discrete state-action spaces, we can write

Note that both the switch to double summation and the change of summation order from \mathbb{A} to $A_p(s)$ followed by $\mathbb{A} \setminus A_p(s)$ require $J(\pi)$ to be well-defined.

$$\begin{aligned} J(\pi) &= \sum_{(s, a) \in \mathbb{S} \times \mathbb{A}} r(s, a) \cdot z_\pi(s, a) = \sum_{s \in \mathbb{S}} \sum_{a \in \mathbb{A}} r(s, a) \cdot z_\pi(s, a) \\ &= \sum_{s \in \mathbb{S}} \left(\sum_{a \in A_p(s)} r(s, a) \cdot z_\pi(s, a) + \sum_{a \in \mathbb{A} \setminus A_p(s)} r(s, a) \cdot z_\pi(s, a) \right) \\ &= \sum_{s \in \mathbb{S}} \sum_{a \in A_p(s)} r(s, a) \cdot z_\pi(s, a). \end{aligned}$$

When the occupancy measure z_π of a policy π is known, we can use Proposition 2.12 to compute the gain $J(\pi)$. Unfortunately, we still do not know how to find the occupancy measure for a given policy. This can be done by solving the following system of equations known as the flow-conservation recurrence.

Definition 2.30 | flow-conserving measure

Alternatively, this is known as the conservation of mass.

A measure μ on the measurable space $(\mathbb{S} \times \mathbb{A}, \mathcal{B}(\mathbb{S} \times \mathbb{A}))$ is called *flow-conserving* for a γ -discounted infinite-horizon MDP \mathfrak{M}_∞ if

$$\mu(ds', \mathbb{A}) = \alpha(ds') + \gamma \cdot \int_{\mathbb{S} \times \mathbb{A}} \mu(ds, da) \cdot p(ds' | s, a).$$

Proposition 1 of Larocche et al. [ibid.] claims this result if occupancy measures are σ -finite (including simply finite), which is true by

Lemma 2.11.

In particular, if the state-action space $\mathbb{S} \times \mathbb{A}$ is discrete,

$$\sum_{a' \in \mathbb{A}} \mu(s', a') = \alpha(s') + \gamma \cdot \sum_{s \in \mathbb{S}} \sum_{a \in \mathbb{A}} \mu(s, a) \cdot p(s' | s, a).$$

Proposition 2.13 * conservation of flow

In a γ -discounted infinite-horizon MDP \mathfrak{M}_∞ , any policy $\pi \in \Pi$ has a flow-conserving occupancy measure z_π .

Remark 2.4

By Remark 2.3, if the admissible control space \mathbb{X} is finite, then

$$\sum_{a' \in A_p(s')} z_\pi(s', a') = \alpha(s') + \gamma \cdot \sum_{s \in \mathbb{S}} \sum_{a \in A_p(s)} z_\pi(s, a) \cdot p(s' | s, a)$$

and $z_\pi(s, A') = 0$ for any $A' \in \mathbb{A} \setminus A_p(s)$.

- Conversely, given an occupancy measure, it can be transformed into a policy as follows.

Definition 2.31 | state occupancy

Given an occupancy measure $z(\cdot, \cdot)$, the *state occupancy* $z(\cdot) : \mathbb{S} \rightarrow \mathbb{R}_+$ is the expected γ -discounted number of visits of states:

$$z(s) \triangleq \int_{\mathbb{A}} z(s, da) = \sum_{a \in \mathbb{A}} z(s, a). \quad (2.22)$$

Definition 2.32 | occupancy-induced policy

Given a stationary infinite-horizon MDP \mathfrak{M}_∞ with a discrete admissible control space \mathbb{X} and an occupancy measure $z(\cdot, \cdot)$, the *occupancy-induced policy* $\pi^z \in \Pi_{\text{RS}}$ is a stationary policy produced by the following operator:

$$\pi^z(a | s) = [\mathcal{Z}z](a | s) \triangleq \begin{cases} \frac{z(s, a)}{z(s)}, & \text{if } z(s) \neq 0, \\ \text{arbitrary,} & \text{otherwise.} \end{cases} \quad (2.23)$$

2.3 Finding Optimal Policies

In the continuous case, a similar definition exists but it involves a Radon–Nikodym derivative.

- These definitions assume that the function z is known to be an occupancy measure of some policy. In fact, this requirement is not necessary and any function f induces a policy $\mathcal{Z}f$, but the resulting policy will have an occupancy that is equal to the original function only in the case established by the following theorem.

Theorem 2.14 * flow conservation induces policies

Given a stationary infinite-horizon MDP \mathfrak{M}_∞ with a discrete admissible control space \mathbb{X} , consider an absolutely-summable non-negative function $f \in L^1(\mathbb{X})$, $f \geq 0$. If the function f is a flow-conserving occupancy function, then the occupancy function of the policy $\pi^f = \mathcal{Z}f$ induced by the function f is equal to f , $z_{\pi^f} = f$, and therefore the function f is an occupancy function.

- Results similar to Theorem 2.14 have been established before for the special case when the initial state distribution α covers all states, $\text{supp } \alpha = \mathbb{S}$. For example, this setting was considered by Puterman [1994, Theorem 6.9.1] and Lee et al. [2017, Section 3]. Intuitively, when some of the states are excluded from the initial

distribution, the same result should hold. However, the existing proof of Puterman [1994] involves multiplication by $\alpha(s)$, which is guaranteed to be non-zero by the assumption $\text{supp } \alpha = \mathbb{S}$. We do not impose such restriction on the initial state distribution α . Therefore, Theorem 2.14 needs to be proven in a different way; the proof is presented in Appendix A.2.

Finally, because every absolutely bounded non-negative flow-conserving function induces a policy and vice versa, the following result follows.

Corollary 2.15 * *decision rules matter almost everywhere only*
Given a stationary infinite-horizon MDP \mathfrak{M}_∞ with a discrete admissible control space \mathbb{X} , let $\pi \in \Pi_{\text{RS}}$ be a stationary policy that induces an occupancy z_π . The policy induced by the occupancy z_π and the original policy π coincide $z_\pi(\cdot)$ -a.e.

This result holds for continuous problems as well, see Theorem 3 of Laroché et al., 2022.

For $z_\pi(\cdot)$ -a.e., see Definition 2.6.

See Figure 2.7, p. 36.

This corollary tells us that policies only need to be defined almost everywhere. Intuitively, this means that if an optimal policy does not lead to some states, it is irrelevant what the policy does in those states. Thus, optimal policies—unlike uniformly-optimal ones—are allowed to not “care” about some of states, as long as those states are not visited. This justifies our choice of the optimality criterion as opposed to the more commonly used uniform optimality.

2.3.4 Linear-Programming Formulation

The results of the previous section allow us to rewrite the policy optimization problem (2.7) as a linear program. In particular, the following formulation follows immediately from Theorem 2.14.

Corollary 2.16 * *linear programming formulation*

Consider a γ -discounted stationary infinite-horizon MDP \mathfrak{M}_∞ . Let function $z_\star : \mathbb{X} \rightarrow \mathbb{R}$ be a solution of the following linear program:

$$\begin{aligned} \max_{z \in L^1(\mathbb{X})} \quad & \sum_{(s,a) \in \mathbb{X}} r(s,a) \cdot z(s,a) \\ \text{s.t.} \quad & \sum_{a' \in A_p(s')} z(s',a') - \gamma \cdot \sum_{(s,a) \in \mathbb{X}} z(s,a) \cdot p(s' | s,a) = \alpha(s') \text{ for all } s' \in \mathbb{S}, \\ & y \geq 0, \end{aligned}$$

where $L^1(\mathbb{X})$ is the space of absolutely summable measurable functions on \mathbb{X} . Then $\pi^{z_\star} = \mathcal{L}z_\star$ is an optimal policy.

To simplify further presentation, let us equip the spaces $\mathbb{R}^{\mathbb{X}}$

and $\mathbb{R}^{\mathbb{S}}$ of real-valued functions with inner products $\langle \cdot, \cdot \rangle_{\mathbb{X}}$ and $\langle \cdot, \cdot \rangle_{\mathbb{S}}$ given by

$$\langle z, z' \rangle_{\mathbb{X}} = \sum_{(s,a) \in \mathbb{X}} z(s,a) \cdot z'(s,a) \quad \text{and} \quad \langle y, y' \rangle_{\mathbb{S}} = \sum_{s \in \mathbb{S}} y(s) \cdot y'(s)$$

2.3 Finding
Optimal Policies

for all functions $z, z' \in \mathbb{R}^{\mathbb{X}}$ and $y, y' \in \mathbb{R}^{\mathbb{S}}$. The objective of the linear program can be written as $\langle r, z \rangle_{\mathbb{X}}$.

Next, recall the transition operator \mathcal{T} of (2.19) and define the extension operator $\mathcal{N}: \mathbb{R}^{\mathbb{S}} \rightarrow \mathbb{R}^{\mathbb{X}}$ as

$$[\mathcal{N}y](s,a) \triangleq y(s) \quad (2.24)$$

for any function $y \in \mathbb{R}^{\mathbb{S}}$. While these operators do not appear in the linear program directly, their adjoints \mathcal{T}_* and \mathcal{N}_* do.

Definition 2.33 | adjoint operator

Given an operator $\mathcal{Y}: \mathbb{Y} \rightarrow \mathbb{Y}'$ between complete metric spaces \mathbb{Y} and \mathbb{Y}' equipped with inner products $\langle \cdot, \cdot \rangle_{\mathbb{Y}}$ and $\langle \cdot, \cdot \rangle_{\mathbb{Y}'}$ respectively, its *adjoint* is an operator $\mathcal{Y}_*: \mathbb{Y}' \rightarrow \mathbb{Y}$ such that

$$\langle \mathcal{Y}y, y' \rangle_{\mathbb{Y}'} = \langle y, \mathcal{Y}_*y' \rangle_{\mathbb{Y}}.$$

Using the definition, the adjoint operators \mathcal{T}_* and \mathcal{N}_* are

$$[\mathcal{T}_*y](s') \triangleq \sum_{(s,a) \in \mathbb{X}} p(s' | s, a) \cdot y(s, a) \quad \text{and} \quad [\mathcal{N}_*y](s') \triangleq \sum_{a' \in A_p(s')} y(s', a').$$

Thus, the linear program of Corrolary 2.16 can be written as

$$\begin{aligned} J_P &= \max_{z \in L^1(\mathbb{X})} \langle r, z \rangle_{\mathbb{X}} \\ \text{s.t.} \quad & \mathcal{N}_*z - \gamma \cdot \mathcal{T}_*z = \alpha, \\ & y \geq 0. \end{aligned} \quad (P)$$

The problem is in the so-called standard form and therefore it has the following well-known dual [Puterman, 1994, Section 6.9.1]:

$$\begin{aligned} J_D &= \min_{v \in L^\infty(\mathbb{S})} \langle \alpha, v \rangle_{\mathbb{S}} \\ \text{s.t.} \quad & \mathcal{N}v - \gamma \cdot \mathcal{T}v \geq r. \end{aligned} \quad (D) \quad (D.1)$$

Absolute summability of the dual variables v guarantees that the dual objective is well-defined for any initial state distribution α .

The dual problem has an interesting property: any feasible dual variable v is an upper bound on the optimal value v_\star given by (2.20), $v \geq v_\star$ [ibid., Theorem 6.2.2 a]. Since the optimal value v_\star

itself satisfies the constraint (D.1), it can be found as the solution to the dual problem (D) [Puterman, 1994, Section 6.9].

See pp. 46, 48.

By Proposition 2.12 and Corollary 2.16, $J_\star = J_P$. When the admissible control space is finite, $|\mathbb{X}| < \infty$, the number of constraints and decision variables are both finite. In this case, the two problems are known to be strongly dual, that is, $J_P = J_D$. Therefore, the decision-making problem can be solved via either the primal or the dual program, justifying the value-based approach.

2.4 COUNTABLY-INFINITE PROBLEMS

When the admissible control space \mathbb{X} is finite and the problem has a finite horizon, the results established in the previous section hold trivially. Indeed, the reward bounds of Condition 2.2 exist:

$$r_- = \min_{t \in \mathbb{T}} \min_{(s,a) \in \mathbb{X}} r_t(s,a) \quad \text{and} \quad r_+ = \max_{t \in \mathbb{T}} \max_{(s,a) \in \mathbb{X}} r_t(s,a),$$

because finite-set extrema always exist. The same is true for the stationary infinite-horizon case:

$$r_- = \min_{(s,a) \in \mathbb{X}} r_t(s,a) \quad \text{and} \quad r_+ = \max_{(s,a) \in \mathbb{X}} r_t(s,a).$$

The case of countably-infinite MDPs—including non-stationary infinite-horizon MDPs—is fundamentally different. For example, if the reward r depends linearly on the state $s \in \mathbb{N}$, $r(s,a) = c \cdot s + f(a)$, then either r_- or r_+ does not exist, depending on whether c is negative or positive. In particular, this is true for the inventory management problem of Section 1.3.2, as the holding costs are proportionate to the stock at hand of the product. In this section, we discuss this and other problems that arise in countably-infinite MDPs, as well as some of the ways they can be addressed.

2.4.1 Ill-Defined Values

Under the expected γ -discounted total reward criterion, the goal of the agent is to maximize the expected value $J(\pi) = E_\pi[\Psi]$ of (2.5). By Remark 2.1, it does not have to be well-defined. Similarly, the reformulation of the problem in terms of value functions v_π and v_\star requires the Fubini–Tonelli theorem to hold, as it involves changing the summation index set from the sample space Ω_T in (2.5) to the state \mathbb{S} and time \mathbb{T} spaces in (2.6) and Definition 2.19.

See Proposition (2.3),
P. 34

Why does this problem not arise in the finite case? To answer this question, let us inspect the value recurrence (2.13):

$$v_\pi(s) = r_\pi(s) + \gamma \cdot \sum_{s' \in \mathbb{S}} p_\pi(s' | s) \cdot v_\pi(s').$$

2.4 Countably-Infinite Problems

It involves a sum over a finite state set \mathbb{S} . As long as all of the values are finite, it is absolutely summable. Thus it can be rearranged into (2.12), which is also absolutely summable by the same logic. Because at least one of the rearrangements of summation in the gains $J(\pi)$ is absolutely summable, it is well-defined, and all of these summation rearrangements are possible due to the Fubini–Tonelli theorem.

When the state space \mathbb{S} is countably-infinite, the same is not true anymore. Even when all of the values are finite, the value recurrence does not have to be absolutely summable. Thus, absolute summability of various sums should be carefully considered every time the summation order changes are introduced.

2.4.2 Infinitely-Many Permitted Actions

Why do we study problems in which the state space \mathbb{S} can be infinite, but the action space \mathbb{A} is always finite? On the surface, both lead to countably-infinite admissible control spaces $\mathbb{X} = \mathbb{S} \times \mathbb{A}$, and the distinction may seem unnecessary.

The problem arises when for some state $s \in \mathbb{S}$ there are infinitely-many permitted actions $A_p(s)$, the action space is no longer compact and Condition 2.3 cannot be used to establish existence of optimal policies.

Moreover, the maximum in (2.15) may not be achieved. Example 2.3 shows such behavior. When the maximum in (2.15) is replaced with supremum, the policies are no longer guaranteed to achieve the optimal values v_\star . In this case, a notion of ε -optimal policies is introduced. Such policies have values v_π that differ from the optimal value function by no more than ε , $\|v_\pi - v_\star\|_\infty < \varepsilon$ for some $\varepsilon > 0$, but the existence of such policies still needs to be established based on the particular form of the action space.

See p. 42

For these reasons, we choose to study problems with finite action spaces only and impose the following condition on the action space, which is a stronger version of Condition 2.3.

Condition 2.4 | finiteness of permitted actions

For every state $s \in \mathbb{S}$, the set of permitted actions $A_p(s)$ is finite.

2.4.3 Unbounded Rewards

If either the lower r_- or the upper reward bound r_+ does not exist, we cannot establish that Bellman operators \mathcal{L}_π and \mathcal{L}_\star are contractions in the space of bounded functions $L^\infty(\mathbb{S})$.

In order to establish existence of optimal policies, alternative conditions on the rewards and transition probabilities are required. For example, we can assume that there exist variable reward bounds that are allowed to grow infinitely.

Condition 2.5 | existence of variable reward bounds

Compare to
Assumptions A1–A3 of
Lee et al. [2017].

For all states $s \in \mathbb{S}$, actions $a \in A_p(s)$, and deterministic Markov policies $\pi \in \Pi_{\text{DM}}$ there exist:

- a positive *weight function* $w : \mathbb{S} \rightarrow \mathbb{R}$ such that $\inf_{s \in \mathbb{S}} w(s) > 0$ and the rewards are absolutely bounded by w :

$$|r| \leq \mathcal{N}w, \quad \text{that is} \quad |r(s, a)| \leq w(s); \quad (2.25)$$

Inequality of functions
is understood pointwise,
i.e., $f \leq g$ means that
 $f(y) \leq g(y)$ for any
 $y \in \text{dom } f = \text{dom } g$.

- a *one-stage expansion coefficient* $\kappa > 0$ such that one-stage γ -discounted transitions expand w at most by a factor of κ :

$$\gamma \cdot \mathcal{T}_\pi w \leq \kappa \cdot w \quad \text{or equivalently} \quad \gamma \cdot \mathcal{T}w \leq \kappa \cdot \mathcal{N}w; \quad (2.26)$$

- a *contraction horizon* $v \in \mathbb{N}$ and a *v-stage contraction coefficient* $0 \leq \lambda < 1$, such that v -stage transitions and discounting contract w at least by a factor of λ :

$$\gamma^v \cdot \mathcal{T}_\pi^v w \leq \lambda \cdot w \quad \text{or} \quad \gamma^v \cdot (\mathcal{T}\mathcal{N}_*)^{v-1} \mathcal{T}w \leq \lambda \cdot \mathcal{N}w. \quad (2.27)$$

- If this condition holds, the values $v_\pi(s)$ of any Markov policy are well-defined, because they can be bounded as follows.

Proposition 2.17 * variable value bounds

Adopted from
Proposition 1 of Lee
et al. [ibid.].

Under Condition 2.5, the values v_π of any policy $\pi \in \Pi_{\text{RS}}$ are bounded by

$$|v_\pi| \leq \mu \cdot w, \quad \text{where} \quad (2.28)$$

$$\mu \triangleq \begin{cases} \frac{v}{1-\lambda}, & \kappa = 1, \\ \frac{1}{1-\lambda} \cdot \frac{1-\kappa^v}{1-\kappa}, & \text{otherwise.} \end{cases} \quad (2.29)$$

Remark 2.5

Even though Proposition 2.17 assumes that the policy is stationary, $\pi \in \Pi_{\text{RS}}$, it holds for all policies $\pi \in \Pi$ as well, as noted by Puterman [1994, p. 234].

☞ If the function w grows *ad infinitum*, v_π does not belong to $L^\infty(\mathbb{S})$ and Banach fixed-point theorem cannot be applied.

In some cases, an unbounded problem can be transformed into a bounded one. For example, we can transform the problem as follows. Let $\tilde{\mathbb{S}} \triangleq \mathbb{S} \cup \{\diamond\}$ and define

$$\begin{aligned} \tilde{r}(s, a) &\triangleq \begin{cases} w^{-1}(s) \cdot r(s, a), & s \in \mathbb{S}, \\ 0, & s = \diamond, \end{cases} \\ \tilde{p}(s' | s, a) &\triangleq \begin{cases} \frac{p(s' | s, a) \cdot w(s')}{\kappa \cdot w(s)}, & s \in \mathbb{S}, s' \in \mathbb{S}, \\ 1 - \sum_{s'' \in \mathbb{S}} p(s'' | s, a) = \diamond, & \\ 0, & s = \diamond, s' \in \mathbb{S}, \\ 1, & s = s' = \diamond, \end{cases} \quad (2.30) \\ \tilde{\gamma} &\triangleq \gamma \kappa. \end{aligned}$$

The newly added state \diamond is called the absorbing state, because once it is reached the environment remains therein.

Equation (2.26) guarantees that the probabilities in the transformed problem are less than one, and the absorbing state 0 is added so that they add up to one. The new problem is absolutely bounded, $\|r\|_\infty \leq 1$, and it is easy to check that its solution is equivalent to the solution of the original problem. Unfortunately, this method is only applicable if $\kappa < \gamma^{-1}$, otherwise the new discounting factor $\tilde{\gamma}$ is larger than one.

If this problem transformation is not possible, we can define a different Banach space to which the value functions v_π belong. We do so by using an alternative norm $\|\cdot\|_w$.

Definition 2.34 | weighted supremum norm

The *w-weighted supremum norm* $\|\cdot\|_w$ of a function $v : \mathbb{S} \rightarrow \mathbb{R}$ is a norm given by

$$\|v\|_w \triangleq \sup_{s \in \mathbb{S}} w^{-1}(s) \cdot |v(s)|. \quad (2.31)$$

Remark 2.6

Equations (2.28) and (2.31) imply each other:

$$|v_\pi| \leq \mu \cdot w \quad \Leftrightarrow \quad \|v_\pi\|_w \leq \mu.$$

☞ Indeed,

$$\|v_\pi\|_w = \sup_{s \in \mathbb{S}} w^{-1}(s) \cdot |v_\pi(s)| \leq \mu \cdot \sup_{s \in \mathbb{S}} w^{-1}(s) \cdot w(s) = \mu$$

2.4 Countably-Infinite Problems

and vice versa, for any state $s \in \mathbb{S}$

$$\begin{aligned} |v_\pi(s)| &= w(s) \cdot w^{-1}(s) \cdot |v_\pi(s)| \leq w(s) \cdot \sup_{s' \in \mathbb{S}} w^{-1}(s') \cdot |v_\pi(s')| \\ &= w(s) \cdot \|v_\pi\|_w \leq \mu \cdot w(s). \end{aligned}$$

In general, for any function u and constant c ,

$$\|u\|_w \leq c \quad \Leftrightarrow \quad |u(s)| \leq \|u\|_w \cdot w(s) \leq c \cdot w(s) \text{ for all } s \in \mathbb{S}. \quad (2.32)$$

Definition 2.35 | multi-step contraction

Consider a metric space (\mathbb{Y}, d) . An operator $\mathcal{Y} : \mathbb{Y} \rightarrow \mathbb{Y}$ is called a v -step contraction in the space \mathbb{Y} with respect to the metric d if there exists a constant $0 \leq k < 1$, such that for any $y, y' \in \mathbb{Y}$

$$d(\mathcal{Y}^v y - \mathcal{Y}^v y') \leq k \cdot d(y - y').$$

Proposition 2.18 * fixed points of multi-step contractions

Adopted from Theorem
6.10.2 of Puterman
[1994].

A multi-step contraction $\mathcal{Y} : \mathbb{Y} \rightarrow \mathbb{Y}$ in a Banach space $(\mathbb{Y}, \|\cdot\|)$ has a unique fixed point if its Lipschitz constant is finite.

Proposition 2.19 * Bellman operators are v -step contractions

Adopted from
Proposition 6.10.3
of Puterman [ibid.].

Let $(L^w(\mathbb{S}), \|\cdot\|_w)$ be the Banach space of all real-valued functions with finite w -weighted supremum norm $\|\cdot\|_w$. If Condition 2.5 holds for the weight function w , then the Bellman operators \mathcal{L}_π and \mathcal{L}_\star are v -step contractions in the space $L^w(\mathbb{S})$. Moreover, their fixed points v_π and v_\star exist uniquely in $L^w(\mathbb{S})$.

☞ These propositions explain why Condition 2.5 is used. It contains the necessary conditions for existence and uniqueness of v_π and v_\star in the space $L^w(\mathbb{S})$. Proposition 2.17 establishes that the values belong to the space $L^w(\mathbb{S})$, (2.26) guarantees that the Lipschitz constants of the Bellman operators are finite, and (2.27) implies that the Bellman operators are v -step contractions.

Therefore, in countably-infinite MDPs value functions exist under Condition 2.5. As a result, Proposition 2.6 can be replaced with the following stronger version applicable even when the rewards are unbounded.

Proposition 2.20 * optimality in stationary infinite-horizon MDPs with unbounded rewards

Adopted from Theorem
6.10.4 of Puterman
[ibid.].

Let Conditions 2.3 (or 2.4) and 2.5 hold for a stationary infinite-horizon Markov decision process \mathfrak{M}_∞ with an infinitely-countable state space, $|\mathbb{S}| = \infty$. Then there exists a deterministic stationary policy $\pi \in \mathbb{D}$ that is optimal.

2.4.4 Linear-Programming Formulation

We established the primal linear program from the definition of the occupancy measure. Since the occupancies are defined without utilizing the rewards, they do not require the uniform boundedness assumption in order to exist. Thus, the case of unbounded rewards admits the same primal formulation (P). The resulting linear program is countably infinite, and duality requires additional considerations.

First, the domains of z and r need to be $\langle \cdot, \cdot \rangle_{\mathbb{X}}$ -dual; similarly, the domains of ν and α must be $\langle \cdot, \cdot \rangle_{\mathbb{S}}$ -dual [Hernández-Lerma and Lasserre, 2002, Section 11.2.2]. This duality is defined as follows.

Definition 2.36 | bilinear form

A function $\langle \cdot, \cdot \rangle$ is called a *bilinear form* on $\mathbb{Y} \times \mathbb{Y}'$ if

- the mapping $\langle \cdot, y' \rangle$ is linear on \mathbb{Y} for every $y' \in \mathbb{Y}'$ and
- the mapping $\langle y, \cdot \rangle$ is linear on \mathbb{Y}' for every $y \in \mathbb{Y}$.

Definition 2.37 | dual spaces

Consider a tuple $(\mathbb{Y}, \mathbb{Y}', \langle \cdot, \cdot \rangle)$ of spaces \mathbb{Y} and \mathbb{Y}' equipped with a bilinear form $\langle \cdot, \cdot \rangle$ on $\mathbb{Y} \times \mathbb{Y}'$. The pair of spaces $(\mathbb{Y}, \mathbb{Y}')$ is called $\langle \cdot, \cdot \rangle$ -*dual* or simply dual, if the bilinear form separates them, that is,

- for all $y \neq 0$ in \mathbb{Y} there exist $y' \in \mathbb{Y}'$ such that $\langle y, y' \rangle \neq 0$ and
- for all $y' \neq 0$ in \mathbb{Y}' there exist $y \in \mathbb{Y}$ such that $\langle y, y' \rangle \neq 0$.

In the uniformly bounded case, the domain of the reward function r is the space $L^\infty(\mathbb{X})$ of absolutely bounded functions, and its dual is the space of absolutely summable functions $L^1(\mathbb{X})$, which is indeed the domain of the primal variable z . Similarly, the domains of the initial state distribution α and the dual variable ν are $L^1(\mathbb{S})$ and $L^\infty(\mathbb{S})$, and they are dual as well.

In the unbounded case, however, the domains of the reward function r and the dual variable ν are different. By Condition 2.5 and Proposition 2.17 they are

$$\begin{aligned} L^{\mathcal{N}w}(\mathbb{X}) &\triangleq \{y \in \mathbb{R}^{\mathbb{X}} \mid \|y\|_{\mathcal{N}w} < \infty\} && \text{and} \\ L^w(\mathbb{S}) &\triangleq \{y' \in \mathbb{R}^{\mathbb{S}} \mid \|y'\|_w < \infty\}, && \text{with duals} \\ L_*^{\mathcal{N}w}(\mathbb{X}) &\triangleq \{y \in \mathbb{R}^{\mathbb{X}} \mid \langle y, \mathcal{N}w \rangle_{\mathbb{X}} < \infty\} && \text{and} \\ L_*^w(\mathbb{S}) &\triangleq \{y' \in \mathbb{R}^{\mathbb{S}} \mid \langle y', w \rangle_{\mathbb{S}} < \infty\} \end{aligned}$$

2.4 Countably-Infinite Problems

For (P), see p. 49.

When the bilinear form $\langle \cdot, \cdot \rangle$ is the inner product on $\mathbb{R}^{\mathbb{Y}}$, the spaces $L^p(\mathbb{Y}, \mu)$ and $L^q(\mathbb{Y}, \mu)$ are dual if p and q are Hölder conjugates, i.e., $1/p + 1/q = 1$. When $p = \infty$ and $q = 1$ this is not generally true, but it is true for the counting measure.

[Hernández-Lerma and Lasserre, 2002, p. 340]. Therefore, the primal variables z and the initial state distribution α should belong to the spaces $L_*^{\mathcal{N}w}(\mathbb{X})$ and $L_*^w(\mathbb{S})$ instead of $L^1(\mathbb{X})$ and $L^1(\mathbb{S})$ respectively.

The reverse does not have to be true. Consider $\mathbb{S} = \mathbb{N}$, $\mathbb{A} = \{a_0\}$, and $y(s, a) = 6/s^2$. Because $\|y\|_1 = \pi^2 < \infty$, $v \in L^1(\mathbb{X})$. For $w(s) = 1 + s$, $\langle y, \mathcal{N}w \rangle_{\mathbb{X}}$ diverges and $v \notin L_*^{\mathcal{N}w}(\mathbb{X})$. In general, $L^1(\mathbb{X})$ and $L_*^{\mathcal{N}w}(\mathbb{X})$ do not coincide.

Mutatis mutandis means “once the necessary changes have been made.”

Condition 2.6 is also used by Lee et al. [2017].

The space $L_*^{\mathcal{N}w}(\mathbb{X})$ is a subset of $L^1(\mathbb{X})$, $L_*^{\mathcal{N}w}(\mathbb{X}) \subseteq L^1(\mathbb{X})$, because

$$\langle y, \mathcal{N}w \rangle_{\mathbb{X}} = \sum_{(s,a) \in \mathbb{X}} |y(s, a)| \cdot w(s) \geq \sum_{(s,a) \in \mathbb{X}} |y(s, a)| \cdot \inf_{s' \in \mathbb{S}} w(s') \geq \|y\|_1 \cdot w_0$$

and therefore finiteness of $\langle y, \mathcal{N}w \rangle_{\mathbb{X}}$ implies that $\|y\|_1$ is also finite. The same argument applies *mutatis mutandis* to show that $L_*^w(\mathbb{S}) \subseteq L^1(\mathbb{S})$.

In general, the initial state distribution α is not guaranteed to belong to the space $L_*^w(\mathbb{S})$. Therefore, the following condition needs to be imposed.

Condition 2.6 | finiteness of w -weighted initial distribution

The initial state distribution α satisfies $\langle \alpha, w \rangle_{\mathbb{S}} < \infty$.

Since z is a decision variable in the optimization problem, we can assert that $z \in L_*^w(\mathbb{S})$. in the definition of the primal program. At the same time, by Theorem 2.14, $z \in L^1(\mathbb{X})$. Is it possible that by restricting the search space to $L_*^{\mathcal{N}w}(\mathbb{X})$ we exclude some of the feasible occupancy measures? The following lemma alleviates this concern.

Lemma 2.21 * feasible region embedding

Under Conditions 2.4, 2.5, and 2.6, the feasible region of the primal program (P) is a subset of the space $L_^{\mathcal{N}w}(\mathbb{X})$ of functions with finite w -weighted supremum norm $\|\cdot\|_w$.*

For (P), see p. 49.

The proof is presented in Appendix A.3.

Next, duality requires the operator $\mathcal{C}_* \triangleq \mathcal{N}_* - \gamma \cdot \mathcal{T}_*$ describing the constraints to be weakly continuous [Hernández-Lerma and Lasserre, 2002, p. 342], that is, the adjoint operator \mathcal{C} must map $L^w(\mathbb{S})$ to $L^{\mathcal{N}w}(\mathbb{X})$. In our case, this means that $\|v\|_w < \infty$ should imply

$$\|\mathcal{N}v - \gamma \cdot \mathcal{T}v\|_{\mathcal{N}w} < \infty.$$

See p. 52.

This holds trivially from the triangle inequality and (2.26).

Unlike the finite case, countably-infinite linear programs do not have to be even weakly dual ($J_P \leq J_D$) in general [Romeijn, R. L. Smith, and Bean, 1992; Ghate and R. L. Smith, 2013]. Interestingly, if strong duality cannot be established, this implies that the dual

approach that involves the value functions may not return an optimal policy. Fortunately, for this pair of programs strong duality holds as per the following theorem.

2.4 Countably-Infinite Problems

Theorem 2.22 * duality in countably-infinite MDPs

Consider a stationary MDP \mathfrak{M}_T with countably-infinite state space \mathbb{S} such that Condition 2.4 holds (that is, the admissible control space \mathbb{X} is finite).

If the rewards can be uniformly bounded, that is, if Condition 2.1 holds, the MDP \mathfrak{M}_T can be solved via a dual pair of linear programs (P) and (D).

For (P) and (D), see p. 49.

If the rewards are unbounded but Conditions 2.5 and 2.6 hold, it is equivalent to the following pair of dual countably-infinite linear programs.

$$\begin{aligned} J_P = \max_{z \in L_*^{\mathcal{N}w}(\mathbb{X})} \quad & \langle r, z \rangle_{\mathbb{X}} \\ \text{s.t.} \quad & \mathcal{N}_* z - \gamma \cdot \mathcal{T}_* z = \alpha, \\ & y \geq 0. \end{aligned} \quad (\text{CI-P})$$

$$\begin{aligned} J_D = \min_{v \in L^w(\mathbb{S})} \quad & \langle \alpha, v \rangle_{\mathbb{S}} \\ \text{s.t.} \quad & \mathcal{N}v - \gamma \cdot \mathcal{T}v \geq r. \end{aligned} \quad (\text{CI-D})$$

In both cases, the problems are strongly dual, that is, $J_{\star} = J_P = J_D$.

Proof. The dual linear-programming formulation and strong duality are proven by Lee et al. [2017] under Conditions 2.1, 2.5, and 2.6 for absolutely summable primal variable $z \in L^1(\mathbb{X})$ and an initial distribution α with full support, $\text{supp } \alpha = \mathbb{S}$.

By Lemma 2.21, the domain change for the primal variable does not affect its solution and does not affect strong duality.

Lee et al. [ibid.] consider universally optimal policies only. Full support of the initial distribution α ensures that occupancies of policies $\mathcal{Z}z$ induced by the primal variable z have full support as well. By Corollary 2.15 this guarantees universal optimality. When universal optimality requirement is relaxed, the same proof applies *ceteris paribus*.

Ceteris paribus means
“all other things
unchanged.”

The special case of uniformly bounded rewards follows trivially from the unbounded one by letting the weight function w be constant. It is also proven by Ghatge [2015]. QED

🌀 The proof shows that Theorem 2.22 closely resembles Theorems 2 and 4 of Lee et al. [2017] but does not assume full support of

the initial distribution α . Other similar formulations can be found in the literature. Most authors do not impose Condition 2.5 and seek universally-optimal policies. Hernández-Lerma and Lasserre [2002] show that in this case weak duality holds under minor additional assumptions. Altman [1999, Chapters 6 and 8] imposes a different condition on the value function and considers constraint MDPs. Ghaté and R. L. Smith [2013] consider non-stationary problems which are a subclass of countably-infinite MDPs as per Lemma 2.7. Ghaté [2015] proves strong duality for uniformly bounded problems.

Theorem 2.22 differs from all of these enough to be considered on its own; it can be used to find optimal policies instead of universally optimal ones, and constraints the domain of the primal variable z to a smaller space $L_*^{\mathcal{N}w}(\mathbb{X})$.

2.4.5 Inventory Management (Revisited)

See p. 6.

To illustrate how the theory presented in this chapter can be used in practice, let us revisit the multi-product inventory management problem of Section 1.3.2. In this problem, a warehouse manager needs to make decisions about placing orders for a selection of products based on the current stock at hand.

Problem definition

Let us assume that there are n different products in the inventory management problem of Section 1.3.2. In this case, the state space is $\mathbb{S} \triangleq \mathbb{N}_0^n$ and each state $\mathbf{s} = [s_0, s_1, \dots, s_{n-1}]^\top$ is a vector of length n with elements s_i corresponding to the inventory of each product.

Having observed the inventory at the beginning of the month, the warehouse manager places an order telling how much of each product needs to be shipped to the warehouse. The shipment size is restricted by some measurement M , for example, the volume of a truck, or maximum weight. The action space $\mathbb{A} \subseteq \mathbb{N}_0^n$ includes all of the combinations of products with the total measurement adding up to M . Assuming that a unit of the i -th product has a measure of m_i , the action space is given by

The inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ is defined as $\sum_{i=0}^{n-1} x_i \cdot y_i$. It is also equivalent to the matrix product $\mathbf{x}^\top \mathbf{y}$.

$$\mathbb{A} \triangleq \{\mathbf{a} = [a_0, a_1, \dots, a_{n-1}] \in \mathbb{N}_0^n \mid \langle \mathbf{m}, \mathbf{a} \rangle \leq M\}. \quad (2.33)$$

All actions are permitted, $A_p(\mathbf{s}) = \mathbb{A}$ for all $\mathbf{s} \in \mathbb{S}$.

The total demand D_i for each product i during the month t is a random variable with the probability mass function $p_{d,i}(x)$. Additionally, we let $q_{d,i}(x)$ denote the probability that the demand

for the product i is at least x ,

$$p_{d,i}(x) \triangleq \Pr[D_i = x] \quad \text{and} \quad q_{d,i}(x) \triangleq \Pr[D_i \geq x] = \sum_{y=x}^{\infty} p_{d,i}(y).$$

2.4 Countably-Infinite Problems

We assume that the expected demand for each product is finite,

$$d_i = E[D_i] = \sum_{j=0}^{\infty} j \cdot p_{d,i}(j) < \infty.$$

Knowing the initial inventory $S_{t,i}$, the demand $D_{t,i}$ and the ordered amount $A_{t,i}$ of each product, we can compute the inventory at the beginning of the next decision epoch $S_{t+1,i}$ as

$$S_{t+1,i} = \max\{0, S_{t,i} + A_{t,i} - D_{t,i}\}.$$

From this, the marginal probabilities $p_i(s'_i | s_i, a_i)$ of having an inventory of s_i units of the product i at the beginning of the month, ordering a_i units more and having s'_i units at the end of the month are equal to

$$p_i(s'_i | s_i, a_i) = \begin{cases} 0, & \text{if } s'_i > s_i + a_i, \\ p_{d,i}(s_i + a_i - s'_i), & \text{if } s_i + a_i \geq s'_i > 0, \\ q_{d,i}(s_i + a_i), & \text{otherwise } (s'_i = 0). \end{cases} \quad (2.34)$$

We assume that the demands D_i are independent from each other. In this case, we can compute the joint transition probabilities from the marginals as

$$p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \triangleq \prod_{i=0}^{n-1} p_{d,i}(s'_i | s_i, a_i). \quad (2.35)$$

When the demands for each product are interdependent, the transition probabilities $p(\mathbf{s}' | \mathbf{s}, \mathbf{a})$ need to be explicitly defined.

Finally, we need to define the rewards of the problem. For any state-action pair (\mathbf{s}, \mathbf{a}) the expected immediate reward $r(\mathbf{s}, \mathbf{a})$ is given by

$$r(\mathbf{s}, \mathbf{a}) \triangleq G(\mathbf{s}, \mathbf{a}) - H(\mathbf{s}, \mathbf{a}) - O(\mathbf{a}), \quad (2.36)$$

where $G : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}_+$ is the expected revenue (in other words, gain), $H : \mathbb{S} \times \mathbb{A} \rightarrow \mathbb{R}_+$ is the holding cost, and $O : \mathbb{A} \rightarrow \mathbb{R}_+$ is the ordering cost.

The expected revenue depends on the prices of the products \mathbf{c} and the expected sales $g(\mathbf{u})$ when the stock including the order is equal to $\mathbf{u} = \mathbf{s} + \mathbf{a}$ units:

$$g(u_i) \triangleq \sum_{j=0}^{u_i-1} j \cdot p_{d,i}(j) + u_i \cdot q_{d,i}(u_i) \quad \text{and} \quad G(\mathbf{s}, \mathbf{a}) \triangleq \langle \mathbf{c}, g(\mathbf{s} + \mathbf{a}) \rangle,$$

where the expected sales function $g(\mathbf{s} + \mathbf{a})$ is applied elementwise.

The holding cost H also depends on the number of units in stock $\mathbf{s} + \mathbf{a}$ and is equal to

$$H(\mathbf{s}, \mathbf{a}) \triangleq \langle \mathbf{h}, \mathbf{s} + \mathbf{a} \rangle, \quad (2.37)$$

where \mathbf{h} is a vector of holding costs per unit of each product.

Finally, the ordering cost O includes a fixed component o_f that has to be paid if an order is placed, and a variable component with ordering costs per unit given by a vector \mathbf{o}_v :

$$O(\mathbf{a}) \triangleq \begin{cases} o_f + \langle \mathbf{o}_v, \mathbf{a} \rangle, & \text{if any } a_i > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2.38)$$

Finally, the initial state distribution α is known to the decision-maker. Thus, the multi-product inventory management problem is fully defined by a tuple

$$(n, M, \mathbf{m}, p_d, \mathbf{c}, \mathbf{h}, \mathbf{o}_v, o_f, \alpha)$$

that defines an MDP \mathfrak{M}_∞ via $\mathbb{S} = \mathbb{N}_0^n$, (2.33), (2.35), and (2.36).

Reward bounds

The multi-product inventory management problem is interesting to us because of the following property.

Lemma 2.23 * unbounded rewards in inventory management

If at least one holding cost h_i is positive, there exists no uniform reward bound in the multi-product inventory management problem:

$$\sup_{(\mathbf{s}, \mathbf{a}) \in \mathbb{X}} |r(\mathbf{s}, \mathbf{a})| = \infty.$$

☞ The proof of this lemma is presented in Appendix A.4. The fact that no uniform bound on the rewards exists means that the optimal values no longer belong to the space of uniformly bounded functions $L^\infty(\mathbb{S})$ and Proposition 2.10 cannot be used to establish existence and uniqueness of a solution to the problem. Nevertheless, a weight function w of Condition 2.5 still exists.

Lemma 2.24 * weight function in inventory management

In the multi-product inventory management problem, let C_G , C_O , and C_H denote the expected revenue when the inventory is infinite, the maximum cost of placing an order and holding it, and the maximum cost of holding an order.

$$C_G \triangleq \langle \mathbf{c}, \mathbf{d} \rangle \quad \text{for all } (\mathbf{s}, \mathbf{a}) \in \mathbb{X}, \quad (2.39)$$

$$C_O \triangleq o_f + M \cdot \max_{0 \leq i < n} \frac{h_i + o_{v,i}}{m_i}, \quad (2.40)$$

$$C_H \triangleq M \cdot \max_{0 \leq i < n} \frac{h_i}{m_i}. \quad (2.41)$$

2.5 Conclusion

If the expected demands \mathbf{d} are finite, then Condition 2.5 is satisfied with the weight function w , the one-stage expansion coefficient κ , the contraction horizon v and the v -stage contraction coefficient λ given by

$$\begin{aligned} w(\mathbf{s}) &\triangleq \langle \mathbf{h}, \mathbf{s} \rangle + w_0, & \kappa &\triangleq \gamma \cdot (1 + C), \\ v &\triangleq \begin{cases} 1, & \text{if } \kappa < 1, \\ \left\lfloor \frac{W_{-1}(C^{-1}\gamma^{1/C} \ln \gamma)}{\ln \gamma} - \frac{1}{C} \right\rfloor + 1, & \text{if } \kappa \geq 1, \end{cases} & \lambda &\triangleq \gamma^v \cdot (1 + Cv), \end{aligned}$$

where $w_0 = \max\{C_G, C_O\}$, $C \triangleq C_H/w_0$, and W_k is the k -th branch of the Lambert w -function.

🔗 The proof of Lemma 2.24 can be found in Appendix A.4. Note that the values of w , κ , v , and λ do not take into account the transition probabilities and can be used for any demand distribution. When the exact form of the probability transition function is known, it may be possible to find smaller values.

The existence of a weight function w allows us to establish strong duality in the multi-product inventory management problem by Theorem 2.22.

2.5 CONCLUSION

While strong duality is an important theoretical property, neither of the dual countably-infinite problems can be solved directly. In order to be solved, problems with countably-infinite state spaces require development of specialized algorithms. In the following two chapters we present such algorithms: one for non-stationary infinite-horizon problems, and another for the problems with countably-infinite state spaces.

The Lambert w -function $W_k(z)$ is a multifunction that gives the solutions of $W_k(z) \cdot e^{W_k(z)} = z$, with each branch k yielding a different solution. It cannot be expressed in terms of elementary functions. Nevertheless, it can be evaluated numerically and is available in most scientific programming packages, such as `scipy`.

3

The Infinite-Horizon Non-Stationary Model

*Let not the future trouble you; for you
will come to it, if come you must, bearing
with you the same reason which you are
using now to meet the present.*

— Marcus Aurelius Antoninus,
Meditations VII · 8

Translated by A. S. L. Farquharson

IN THIS CHAPTER, we consider the problem of decision-making in infinite-horizon non-stationary Markov environments. This problem is notoriously difficult due to its infinite dimensionality. At the same time, only the optimality of the *initial* action is of importance to the decision-maker: once it has been identified, the procedure can be repeated to generate a plan of arbitrary length. The optimal initial action can be identified by finding a time horizon so long that data beyond it has no effect on the initial decision. This horizon is known as a solution horizon and can be discovered by considering a series of truncations of the problem until a stopping rule guaranteeing initial decision optimality is satisfied. This chapter presents such a stopping rule for problems with unbounded rewards. Given a candidate policy, the rule uses a mathematical program that searches for other possibly optimal policies with different initial actions. If no better initial action can be found, the candidate action is deemed optimal.

3.1 INTRODUCTION

While infinite-horizon stationary discounted MDPs are the most commonly employed models of sequential decision-making under uncertainty, they rely on a crucial but sometimes unrealistic assumption: the data of the problem must remain constant. In order to incorporate possible temporal changes of the data in the decision-making model, non-stationary (sometimes also called non-homogeneous) MDP must be considered.

In this chapter, we study infinite-horizon discounted non-stationary MDPs with finite admissible control spaces. Corollary 2.8 establishes that for such MDPs there exists an optimal policy that is deterministic and Markovian but not necessarily stationary. This means that optimal decision rules may differ between time steps. Because there are infinitely many time steps, infinite-horizon non-stationary MDPs are infinitely-dimensional optimization problems by their nature. This means that standard solution methods (for example, value iteration and policy iteration) require an infinite number of calculations.

3.1.1 *Truncations and Solution Horizons*

To overcome the innate computational hurdle of infinite-dimensionality, a non-stationary MDP can be approximated by a finite-time

This chapter is based on the article published in the Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling [Neustroev, de Weerd, and Verzijlbergh, 2019].

Some changes were made to the text compared to the published version. First, the notation was updated to be consistent with the rest of the thesis. Second, the preliminary results were moved to Chapter 2. Third, the theoretical results were given more rigorous proofs using the results of Chapter 2. Finally, some of the text was edited to improve readability after the other changes. Major changes are discussed in the margin notes.

MDP known as a *truncation*. Ghate [2011] provides a broad survey of such methods.

A typical approach is to use a rolling-horizon procedure [Sethi and Sorger, 1991]. At each time step the original infinite-horizon problem is truncated to a chosen time horizon, known as a *study horizon*, the truncation is solved, and the first decision is made based on this solution. The process is then repeated whenever another decision has to be made. While this approach is computationally feasible, it can lead to sub-optimal decisions, as the truncation discards some of the data. Thus, it is important to identify a study horizon that is guaranteed to give the same initial decision as the infinite-horizon problem. If such a horizon exists, it is known as a *solution horizon* [Bès and Sethi, 1988].

Due to unpredictability of future data and reduced computation time for smaller truncations, the decision-maker is often interested in a solution horizon that is as short as possible. The standard procedure for discovering such a horizon is to construct a series of longer and longer truncations until a certain condition is met. This condition, called a *stopping rule*, must guarantee that the last considered study horizon is a solution horizon.

Several stopping rules have been proposed in the literature [Hopp et al., 1987; Bès and Lasserre, 1986; Hernández-Lerma and Lasserre, 1988; Cheevaprawatdomrong, Schochetman, et al., 2007]. Most of them assume that the rewards are uniformly bounded, and explicitly use these bounds. While uniform bounds are easy to work with, they can be very loose, providing inaccurate estimates of the data in the future states of the model. Moreover, for some problems the boundedness assumption may not hold at all.

Therefore, our goal is to develop a method applicable to non-stationary problems with unbounded rewards. In this chapter, we propose a new stopping rule for infinite-horizon discounted non-stationary MDPs with unbounded rewards. Our rule searches for alternative optimal initial decisions among the feasible problem truncations; if no such decision exists, the initial decision is deemed optimal, and the current horizon is a solution horizon. We show how the stopping rule can be implemented and demonstrate that it is able to find shorter solution horizons than existing methods.

E.g., consider a problem where most of the rewards are not exceeding 1, except for a single state with rewards bounded by 100. Uniform bound of 100 is too loose for most of the rewards.

3.1.2 Previous Work

Chand et al. [2002] provided an exhaustive review of literature on horizon methods. It shows that the majority of research in this

area focuses on deterministic problems: of more than two hundred papers reviewed, less than a third used stochastic models.

The most common approach is to exploit the cost (or reward) properties of a particular problem, both in deterministic and stochastic cases. Two most commonly used properties are convexity [R. L. Smith and Zhang, 1998; Cheevaprawatdomrong and R. L. Smith, 2004] and supermodularity [Nair, 1995; Cheevaprawatdomrong, Schochetman, et al., 2007]. For example, Nair [1995] considered an investment problem under technological change. The proposed method assumes that future technologies will generate higher revenues than the current ones. While this assumption is not restrictive in the particular setting, such monotonically improving environment may not exist for other problems.

In the context of MDPs, Bès and Lasserre [1986] proposed a rolling-horizon procedure and a stopping rule based on the reward differences. Their stopping rule is elegantly simple: an initial decision is deemed optimal if it outperforms all other possible decisions by a given threshold. This threshold is chosen so as to guarantee that no matter what policy is employed after the solution horizon, the difference is outweighed by the initial decision. This method was later extended to the case of MDPs with Borel state spaces [Hernández-Lerma and Lasserre, 1988].

Ergodic properties of the underlying Markov chains may be used as a source of solution horizons as well. For example, Hopp [1989] suggested the following stopping rule. For a given study horizon, approximate all of the future discounted rewards with some constants, known as salvage values. If for all feasible salvage values the resulting problems result in the same optimal initial decision, that decision must be optimal to the original infinite-horizon problem as well. Feasibility of the salvage values is established by bounding their spans using an ergodicity coefficient for discounting. The resulting space of possible salvage values forms a polyhedron, and linear programming can be used to solve the resulting problem [Bean et al., 1992].

Another linear-programming based method for solving non-stationary MDPs was proposed by Ghate and R. L. Smith [2013]. Even though their method addresses a slightly different problem and thus does not involve stopping rules, it provides useful insights on the linear-programming formulations of non-stationary MDPs.

Virtually all of the stopping rules require uniform bounds on the rewards (or their spans); the unbounded case remains relatively

Stochastic models include—but are not limited to—MDPs.

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is *convex* if
 $f(a \cdot \mathbf{x} + (1 - a) \cdot \mathbf{y}) \leq a \cdot f(\mathbf{x}) + (1 - a) \cdot f(\mathbf{y})$
 for any vectors \mathbf{x}, \mathbf{y} , and constant $0 \leq a \leq 1$.

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is *supermodular* if
 $f(\mathbf{x} \vee \mathbf{y}) + f(\mathbf{x} \wedge \mathbf{y}) \geq f(\mathbf{x}) + f(\mathbf{y})$ for all vectors \mathbf{x} , and \mathbf{y} ;
 \vee and \wedge stand for componentwise maxima and minima respectively.

Ergodic means related to the recurrence of states. It comes from Greek *έργο* for *work* and *οδός* for *way*.

A *span* $\|\cdot\|_\diamond$ is a seminorm given by $\|f\|_\diamond \triangleq \sup f - \inf f$.

untreated. Cheevaprawatdomrong, Schochetman, et al. [2007] provided a possible remedy, but necessarily introduced a different set of assumptions. To address this gap, we propose a modification of Hopp’s stopping rule based on the results of Puterman [1994] and Lee et al. [2017] for countable-state MDPs. We implement it using a modification of the linear programming method of Bean et al. [1992]. This modification is based on varying bounds instead of the uniform ones, resulting in better and faster approximations.

3.2 MODEL ASSUMPTIONS

In the published text, these assumptions were mentioned throughout different sections. We collect them together to highlight the class of problems that is considered in this chapter.

The results of this chapter rely on the following four assumptions.

Assumption 3.1 | weight function is known

A weight function $w : \mathbb{S} \times \mathbb{T} \rightarrow \mathbb{R}_+$ of Condition 2.5 is given to the agent along with its parameters κ , λ , and ν .

Assumption 3.2 | finiteness of the admissible control space

The admissible control space \mathbb{X} is finite, $|\mathbb{X}| < \infty$.

Assumption 3.3 | initial state is known

The initial state $s_\alpha \in \mathbb{S}$ is observed by the agent, that is,

$$\alpha(s) = \delta_{s, s_\alpha} \quad \text{for all states } s \in \mathbb{S} \text{ and a given state } s_\alpha \in \mathbb{S}.$$

Assumption 3.4 | optimal initial decision rule is unique

All optimal policies $\pi_\star \in \Pi_{\text{DM}}$ have the same initial decision rule $a_{\star, \alpha}$, $\pi_{\star, 0}(a|s) = \delta_{a, a_{\star, \alpha}}$ for all $(s, a) \in \mathbb{X}$.

- ☞ Assumption 3.1 is required for an optimal policy to be well-defined. Assumption 3.2 guarantees that the only infinite dimension of the problem is time and is useful because it makes the problem data that appears up to any decision epoch is guaranteed to be finite. Assumption 3.3 holds in practice because the agent makes the initial decision *after* the initial state is observed; moreover, this assumption implies that Condition 2.6 holds independent of the weight function w . Finally, Assumption 3.4 is required so that the optimal initial decision exists uniquely. It is the most restrictive of the assumptions that cannot be checked *a priori*, but it holds for many problems nevertheless.

See Figure 2.1, p. 20.

Condition 2.6 becomes
 $w(s_\alpha) < \infty$.

3.3 THE DUAL FORMULATION

Using Definition 2.18 and Lemma 2.7, a non-stationary MDP \mathfrak{M}_∞ can be converted to a problem with a countably-infinite state space $\tilde{\mathbb{S}} = \mathbb{S} \times \mathbb{T}$. Assumptions 3.1 and 3.2 guarantee that the conditions of Theorem 2.22 hold. Thus, under these assumptions the problem \mathfrak{M}_∞ is equivalent to a pair of strongly dual programs (CI-P) and (CI-D). When translated from the countably-infinite formulation back to the non-stationary formulation, these programs become

$$\begin{aligned}
 & \max_{z \in L_*^{\mathcal{N}w}(\tilde{\mathbb{X}})} \sum_{t=0}^{\infty} \langle r_t, z_t \rangle_{\mathbb{X}} \\
 & \text{s.t.} \quad \mathcal{N}_* z_0 = \alpha_0, \\
 & \quad \mathcal{N}_* z_{t+1} - \gamma \cdot \mathcal{T}_{t,*} z_t = \alpha_{t+1} \quad \text{for all } t \in \mathbb{N}_0, \\
 & \quad z_t \geq 0 \quad \text{for all } t \in \mathbb{N}_0. \\
 \\
 & \min_{v \in L^w(\tilde{\mathbb{S}})} \sum_{t=0}^{\infty} \langle \alpha_t, v_t \rangle_{\mathbb{S}} \\
 & \text{s.t.} \quad \mathcal{N} v_t - \gamma \cdot \mathcal{T}_t v_{t+1} \geq r_t \quad \text{for all } t \in \mathbb{N}_0.
 \end{aligned}$$

Here $\alpha_t(s) \triangleq \tilde{\alpha}(s, t)$ is the initial augmented-state distribution.

These programs can be simplified as follows.

First, because the decision-making process always starts at time step zero, it is easy to see that $\alpha_0 = \alpha$ and $\alpha_t = 0$ for any $t \in \mathbb{N}$. Moreover, under Assumption 3.3, only one element of α_0 is non-zero, and the dual objective function becomes simply $z_0(s_\alpha)$.

Next, under Assumption 3.2 each of the functions r_t , z_t , α , v_t , and w_t belongs to a space of measurable functions with a finite domain \mathbb{X} or \mathbb{S} equipped with the counting measure $\#$. These spaces are homeomorphic to vector spaces of dimensions $|\mathbb{X}|$ and $|\mathbb{S}|$; additionally, the bilinear forms $\langle \cdot, \cdot \rangle_{\mathbb{Y}}$, $\mathbb{Y} \in \{\mathbb{S}, \mathbb{X}\}$ coincide with the dot products. Therefore, we can replace these functions with vectors

$$\begin{aligned}
 \mathbf{r}_t &\triangleq [r_t(s, a)]_{(s,a) \in \mathbb{X}}, & \mathbf{z}_t &\triangleq [z_t(s, a)]_{(s,a) \in \mathbb{X}}, \\
 \boldsymbol{\alpha} &\triangleq [\alpha(s)]_{s \in \mathbb{S}}, & \mathbf{v}_t &\triangleq [v_t(s)]_{s \in \mathbb{S}}, & \mathbf{w}_t &\triangleq [w_t(s)]_{s \in \mathbb{S}}
 \end{aligned}$$

after equipping the state space \mathbb{S} and the admissible control space \mathbb{X} with some total orders. The operators \mathcal{T}_t and \mathcal{N} are linear and therefore in the vector reformulation they can be written as $|\mathbb{X}| \times |\mathbb{S}|$ matrices

$$\mathbf{T}_t = [p_t(s' | s, a)]_{(s,a) \in \mathbb{X}, s' \in \mathbb{S}} \quad \text{and} \quad \mathbf{N} = [\delta_{s,s'}]_{(s,a) \in \mathbb{X}, s' \in \mathbb{S}}.$$

3.3 The Dual Formulation

For (CI-P) and (CI-D), see p. 57.

The original presentation used arbitrary constants $\beta_t > 0$ instead of α_t [Neustroev, de Weerd, and Verzijlbergh, 2019]. This is because we considered uniformly-optimal policies only. By Theorem 2.22 this is not necessary. Therefore, we opt for the simpler version of the linear program; this change does not affect any of the results and makes the presentation more consistent throughout the thesis.

Total orders on the spaces \mathbb{S} and \mathbb{X} are required to know how the indexing within vectors is done. For finite sets, this can be done arbitrarily.

By definition, the adjoints of linear operators with respect to dot products are simply transposed matrices \mathbf{T}_t^\top and \mathbf{N}^\top .

Finally, the domains $L_*^{\mathcal{N}w}(\tilde{\mathbb{X}})$ and $L^w(\tilde{\mathbb{S}})$ of the primal and dual variables z and v need to be expressed in terms of the vectors \mathbf{z}_t and \mathbf{v}_t . They can be written as the following additional constraints:

$$\sum_{t=0}^{\infty} \mathbf{w}_t^\top \mathbf{N}^\top \mathbf{z}_t < \infty \quad \text{and} \quad \sup_{t \in \mathbb{N}_0} \|\mathbf{v}_t \oslash \mathbf{w}_t\|_\infty < \infty. \quad (3.1)$$

Therefore, the problem can be equivalently written as:

$$\max_{\mathbf{z}_t \in \mathbb{R}^{|\mathbb{X}|}} \sum_{t=0}^{\infty} \mathbf{r}_t^\top \mathbf{z}_t \quad (\text{NS-P})$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{N}^\top \mathbf{z}_0 = \boldsymbol{\alpha}, \\ & \mathbf{N}^\top \mathbf{z}_{t+1} - \gamma \cdot \mathbf{T}_t^\top \mathbf{z}_t = \mathbf{0} \quad \text{for all } t \in \mathbb{N}_0, \\ & \mathbf{z}_t \geq \mathbf{0} \quad \text{for all } t \in \mathbb{N}_0, \end{aligned}$$

$$\sum_{t=0}^{\infty} \mathbf{w}_t^\top \mathbf{N}^\top \mathbf{z}_t < \infty;$$

$$\min_{\mathbf{v}_t \in \mathbb{R}^{|\mathbb{S}|}} \boldsymbol{\alpha}^\top \mathbf{v}_0 = v_0(s_\alpha) \quad (\text{NS-D})$$

$$\text{s.t.} \quad \mathbf{N} \mathbf{v}_t - \gamma \cdot \mathbf{T}_t \mathbf{v}_{t+1} \geq \mathbf{r}_t \quad \text{for all } t \in \mathbb{N}_0, \quad (\text{NS-D.1})$$

$$\sup_{t \in \mathbb{N}_0} \|\mathbf{v}_t \oslash \mathbf{w}_t\|_\infty < \infty. \quad (\text{NS-D.2})$$

By Theorem 2.22, the dual programs (NS-P) and (NS-D) exhibit strong duality. Moreover, the following properties hold.

Proposition 3.1 * existence of an optimal Markovian policy

There exists a feasible solution to (NS-P) such that for all $t \in \mathbb{N}_0$ and $s \in \mathbb{S}$ there exists exactly one a for which $z_t(s, a) > 0$ and $z_t(s, a') = 0$ for all $a' \neq a$. The Markovian policy π that uses these actions is an optimal policy.

Definition 3.1 | advantages

The inverse slack in (NS-D.1) is called the *reduced cost* [ibid.] or *advantage* of state-action pair (s, a) at time t . We denote it as $\eta_t(s, a)$. Vectorized advantages $\boldsymbol{\eta}_t$ can be expressed as

$$\boldsymbol{\eta}_t \triangleq \mathbf{r}_t - \mathbf{N} \mathbf{v}_t + \gamma \cdot \mathbf{T}_t \mathbf{v}_{t+1}. \quad (3.2)$$

Advantage $\eta_t(s, a)$ of an action a represents the benefit of taking action a over the optimal action and is always non-positive for any feasible solution of the dual problem. Moreover, there exists a useful lower bound, as shown by the following lemma.

In the original presentation, the domains were still expressed in terms of the functions z and v instead of the vectors \mathbf{z}_t and \mathbf{v}_t . The new presentation is more rigorous.

$\mathbf{a} \oslash \mathbf{b}$ stands for the Hadamard (i.e., elementwise) quotient of vectors \mathbf{a} and \mathbf{b} .

Adopted from Theorems 3 and 4 of Lee et al. [2017].

The slack of an inequality is the difference between its sides.

Lemma 3.2 * advantage bounds

For any feasible combination of dual variables \mathbf{v}_t , the advantages given by (3.2) are bounded by

$$-\mathbf{h}_t \leq \boldsymbol{\eta}_t \leq \mathbf{0}, \quad \text{where} \quad \mathbf{h}_t \triangleq (1 + \mu + \gamma\kappa\mu) \cdot \mathbf{N}\mathbf{w}_t.$$

Proof. The upper bound follows from the constraints (NS-D.1). The lower bound is derived using equations (2.25), (2.26), and (2.28), and (3.2):

$$\begin{aligned} \boldsymbol{\eta}_t &= \mathbf{r}_t - \mathbf{N}\mathbf{v}_t + \gamma \cdot \mathbf{T}_t \mathbf{v}_{t+1} \\ &\geq -\mathbf{N}\mathbf{w}_t - \mu \cdot \mathbf{N}\mathbf{w}_t - \gamma\kappa\mu \cdot \mathbf{N}\mathbf{w}_t = -\mathbf{h}_t. \end{aligned} \quad \text{QED}$$

Proposition 3.3 * complementary slackness

If \mathbf{z}_t and \mathbf{v}_t , $t \in \mathbb{N}_0$ are solutions of the programs (NS-P) and (NS-D), then the complementary slackness holds:

$$\mathbf{z}_t(s, a) \odot \boldsymbol{\eta}_t(s, a) = \mathbf{0}, \quad \text{for all } t \in \mathbb{N}_0.$$

3.3 The Dual Formulation

Adopted from Theorems 2, 5 and 6 of Lee et al. [ibid.].

$\mathbf{a} \odot \mathbf{b}$ stands for the Hadamard (that is, elementwise) product of vectors \mathbf{a} and \mathbf{b} .

3.3.1 Problem Truncation

The countably-infinite linear-programming formulation is useful for analyzing mathematical properties of non-stationary problems. At the same time, it cannot be solved directly, because that requires infinite computations. For example, consider the dual program (NS-D). To find \mathbf{v}_0 one needs to know \mathbf{v}_1 , which in turn requires \mathbf{v}_2 , and so on *ad infinitum*. On the other hand, if at least one of the future value vectors \mathbf{v}_{T+1} is known, all of the previous values $\mathbf{v}_T, \mathbf{v}_{T-1}, \dots, \mathbf{v}_0$ can be computed in finite time.

This observation provides one of the ways to address the infinite dimensionality that is used for problems with uniformly bounded rewards. If the future values \mathbf{v}_{T+1} are replaced with a vector \mathbf{u} , the problem becomes finite. Even if the approximation \mathbf{u} is bad, the Bellman operator \mathcal{L}_π is a contraction in bounded problems, which means that each time it is applied to find a preceding value vector, the resulting values get closer to the fixed point, that is, the true values. When the horizon T is sufficiently large, the initial values of the approximation will be close to those of the original problem. This observation leads to the following program.

Definition 3.2 | truncation and salvage vector

A T -truncation of the problem (NS-D) with salvage vector \mathbf{u} is the

following linear program:

$$\begin{aligned} \min_{\mathbf{v}_0, \dots, \mathbf{v}_T} \quad & \boldsymbol{\alpha}^\top \mathbf{v}_0 = v_0(s_\alpha) & (\text{NS-D-ALT}) \\ \text{s.t.} \quad & \mathbf{N}\mathbf{v}_t - \gamma \cdot \mathbf{T}\mathbf{v}_{t+1} \geq \mathbf{r}_t \quad 0 \leq t < T, \\ & \mathbf{N}\mathbf{v}_T - \gamma \cdot \mathbf{T}\mathbf{u} \geq \mathbf{r}_T. \end{aligned}$$

Remark 3.1

The constraint NS-D.2 is no longer required because it holds trivially when the time domain is finite.

- The definition of truncation involves only one salvage vector \mathbf{u} . However, if we choose to consider truncations of different lengths, we may want to use different salvage vectors \mathbf{u}_t at different time steps t . To address this, instead of a single salvage vector \mathbf{u} , we introduce a *salvage function* u . If $u \in L^w(\mathbb{S})$ the solutions of these truncations will be feasible solutions of the original problem (NS-D).

Given such a function u , we obtain a *series of truncations* with different salvage vectors $\mathbf{u}_{T+1} \triangleq [u_{T+1}(s)]_{s \in \mathbb{S}}$ at different study horizons T .

To analyze the truncations, we use a new Bellman operator.

Definition 3.3 | (u, T) -truncated Bellman operators

The (u, T) -truncated Bellman operator $\mathcal{L}_{\pi, u, T} : \bar{\mathbb{R}}^{\mathbb{S}} \rightarrow \bar{\mathbb{R}}^{\mathbb{S}}$ under policy π and the (u, T) -truncated optimal Bellman operator $\mathcal{L}_{\star, u, T} : \bar{\mathbb{R}}^{\mathbb{S}} \rightarrow \bar{\mathbb{R}}^{\mathbb{S}}$ are given by:

$$[\mathcal{L}_{\pi, u, T} v_t](s) \triangleq \begin{cases} r_{\pi, t}(s) + \gamma \cdot [\mathcal{T}_{\pi, t} v_{t+1}](s), & t < T, \\ r_{\pi, t}(s) + \gamma \cdot [\mathcal{T}_{\pi, t} u_{t+1}](s), & t = T, \\ u_t(s), & t > T. \end{cases}$$

$$\mathcal{L}_{\star, u, T} v \triangleq \sup_{\pi \in \mathbb{D}} \mathcal{L}_{\pi, u, T} v.$$

- For any salvage function $u \in L^w(\mathbb{S})$, both operators are multi-stage contractions, therefore, they have unique fixed points by Proposition 2.19. We denote these points as $v_{\pi, u, T}$ and $v_{\star, u, T}$, and their vectors of their values at time t as $\mathbf{v}_{\pi, u, T, t}$ and $\mathbf{v}_{\star, u, T, t}$. By properly choosing a salvage function u , we can obtain convergent upper or lower bounds on $\mathbf{v}_{\star, t}$ using the following proposition.

Proposition 3.4 * optimal value bounds

If there exist functions u_- and u_+ in $L^w(\mathbb{S})$ such that $\mathcal{L}_{\pi, u, T} u_- \geq u_-$ and $\mathcal{L}_{\pi, u, T} u_+ \leq u_+$ for all $\pi \in \mathbb{D}$, then

$$\mathbf{v}_{\star, u_-, T, t} \leq \mathbf{v}_{\star, u_-, T+1, t} \leq \mathbf{v}_{\star, t} \leq \mathbf{v}_{\star, u_+, T+1, t} \leq \mathbf{v}_{\star, u_+, T, t}.$$

Adopted from
Corollary 6.10.10
of Puterman [1994]

Definition 3.4 | value-bounding functions and approximations

Functions u_- and u_+ of Proposition 3.4 are called lower and upper *value-bounding functions*; the values $\mathbf{v}_{\star, u_-, T, t}$ and $\mathbf{v}_{\star, u_+, T, t}$ are called *lower and upper value approximations* respectively.

3.4 A Stopping Rule

- Any function u_+ of Proposition 3.4 provides an upper bound on optimal value function v_{\star} , and thus the operator $\mathcal{L}_{\pi, u, T}$ can be used in linear formulation for the approximated problem (NS-D-ALT) the same way as \mathcal{L}_{π} is used in the original problem (CI-D). The optimal values $\mathbf{v}_{\star, u, T, t}$ are equal to \mathbf{u}_t if $t > T$. The constraints beyond the horizon T will become $\mathbf{v}_t - \mathbf{u}_t \geq 0$, and can be discarded resulting in a truncation with salvage vector $\mathbf{u} = \mathbf{u}_{T+1}$.

3.4 A STOPPING RULE

Section 3.3 shows that non-stationary MDPs can be represented by countably-infinite linear programs. Even though these representations cannot be solved with finite computations, they can be approximated by truncations. As an approximation, a truncation may result in a solution with an immediate decision π_0 that is different from the optimal immediate decision of the original non-stationary MDP. Therefore, we are interested in a method that allows us to check optimality of this decision without solving the countably-infinite linear program. In this section we design such a method for non-stationary MDPs with unbounded rewards.

We start by presenting a problem formulation with variable salvage vector. This formulation was introduced by Hopp [1989] for the uniformly bounded case. We demonstrate how it can be solved using a linear program of Bean et al. [1992]. Then we extend the results to non-stationary MDPs with unbounded rewards by introducing different salvage spaces based on bounding functions instead of uniform bounds. Finally, we present a new algorithm for discovery of optimal solution horizons that employs our stopping rule and exploits the fact that the Bellman operator of the unbounded problem is a multi-stage contraction.

3.4.1 Truncations with Variable Salvage Vector

Assume that for a given study horizon T and salvage function u we have solved a truncation and found the optimal initial action $\pi_{\star, u, T, 0}(s_{\alpha})$. We want to check if this action is equal to the optimal initial action $\pi_{\star, 0}(s_{\alpha}) = a_{\star, \alpha}$ of the original problem.

Suppose that we know that values $\mathbf{v}_{\star, T+1}$ belong to some sets $\mathbb{U}_{T+1} \subseteq \mathbb{R}^{|\mathbb{S}|}$. For example, if the values are non-negative and bounded from above by a constant w , \mathbb{U}_t can be $|\mathbb{S}|$ -dimensional cubes: $\mathbb{U}_t = \{\mathbf{v} \mid \mathbf{0} \leq \mathbf{v} \leq w \cdot \mathbf{1}\}$. If all of the salvage vectors $\mathbf{v} \in \mathbb{U}$ of a given subspace $\mathbb{U} \subseteq \mathbb{R}^{|\mathbb{S}|}$ result in T -truncations with the same optimal initial decision and optimal values $\mathbf{v}_{\star, T+1}$ also belong to that set, then the original problem has the same optimal initial decision $\pi_{\star, 0}$ as the truncation. The following proposition formalizes this observation.

Proposition 3.5 * generalized Hopp's stopping rule

Generalization of
Theorem 1c of Hopp
[1989].

Study horizon T is a solution horizon if the initial optimal action is the same for all $\mathbf{u} \in \mathbb{U}_{T+1}$, where the sequence $(\mathbb{U}_t)_{t \in \mathbb{N}_0}$ of subspaces $\mathbb{U}_t \subseteq \mathbb{R}^{|\mathbb{S}|}$ is chosen so that $\mathbf{v}_{\star, t} \in \mathbb{U}_t$.

Proposition 3.5 was used in Hopp's stopping rule [Hopp, 1989] for constant sequence $\mathbb{U}_t = \mathbb{U}$ based on the uniform bounds of the value vector spans. Given this stopping rule, solution horizons can be discovered by starting with a study horizon $T = 0$, checking the stopping rule, and incrementing T until the stopping rule is satisfied. However, in order for the rule to be of any practical use, we need to guarantee that this solution horizon discovery method terminates in finite time.

The salvage subspaces $(\mathbb{U}_t)_{t \in \mathbb{N}_0}$ must be chosen so that the stopping rule is able to find a solution horizon. This condition can be satisfied due to the following lemma. If $\mathbb{U} \subseteq L^w(\mathbb{S})$, where \mathbb{U} is the set of all salvage functions u providing salvage vectors $\mathbf{u}_t \in \mathbb{U}_t$, then the stopping rule terminates due to the following lemma.

Lemma 3.6 * existence of solution horizons

Generalized Lasserre
and Bès [1984].

Under Assumptions 3.1–3.4, there exists a finite horizon T_{\star} such that for any salvage function $u \in L^w(\mathbb{S})$ all T -truncations with $T \geq T_{\star}$ have the same optimal initial decision.

Proof. This lemma is proposed by Lasserre and Bès [1984] for zero salvage function $u = 0$ and uniformly bounded rewards. Both conditions are only required to guarantee that the objective functions of the linear programs are well-defined, and the proof holds *mutatis mutandis* under Assumption 3.1 for $u \in L^w(\mathbb{S})$. The only crucial assumptions are that the action space \mathbb{A} is finite, the initial state is known and the optimal initial action is unique. **QED**

Moreover, we need to ensure that the condition of the stopping rule can be checked in finite time. When \mathbb{U}_t are polytopes, it can be done by solving a mixed integer linear program of Bean et al. [1992] as follows.

Polytopes can be expressed by sets of linear constraints.

First, we find a candidate optimal initial decision rule $a_\alpha = \pi_{\star, \mathbf{u}, T, 0}(s_\alpha)$ by solving the truncation (NS-D-ALT) for an arbitrary $\mathbf{u} \in \mathbb{U}_{T+1}$. Then we allow the salvage vector \mathbf{v} to vary within \mathbb{U}_{T+1} and seek a decision rule $\pi_{\star, \mathbf{v}, T, 0}(s_\alpha) \neq a_\alpha$ by solving the following program:

$$\begin{aligned}
 \min_{\mathbf{v}, \mathbf{v}_t, \mathbf{z}_t} \quad & \eta_0 = \mathbf{j}_\alpha^\top (\mathbf{r}_0 - \mathbf{N}\mathbf{v}_0 - \gamma \cdot \mathbf{T}_0 \mathbf{v}_1) & (\text{NS-D3}) \\
 \text{s.t.} \quad & \\
 & -\mathbf{h}_t \odot (\mathbf{1} - \mathbf{z}_t) \leq \mathbf{r}_t - \mathbf{N}\mathbf{v}_t + \gamma \cdot \mathbf{T}_t \mathbf{v}_{t+1} \leq \mathbf{0}, \quad 0 \leq t < T, \\
 & -\mathbf{h}_T \odot (\mathbf{1} - \mathbf{z}_T) \leq \mathbf{r}_T - \mathbf{N}\mathbf{v}_T + \gamma \cdot \mathbf{T}_T \mathbf{u} \leq \mathbf{0}, \\
 & \mathbf{N}^\top \mathbf{z}_T = \mathbf{1}, \\
 & \mathbf{j}_\alpha^\top \mathbf{z}_0 = 0, \\
 & \mathbf{v}_t \in \mathbb{U}_t, & 0 \leq t \leq T, \\
 & \mathbf{u} \in \mathbb{U}_{T+1}, \\
 & \mathbf{z}_t \in \{0, 1\}^{|\mathbb{X}|}, & 0 \leq t \leq T,
 \end{aligned}$$

where $\mathbf{j}_\alpha \triangleq [\delta_{(s,a), (s_\alpha, a_\alpha)}]_{(s,a) \in \mathbb{X}}$ is a vector of length $|\mathbb{X}|$, with all elements equal to zero except for the element corresponding to the state-action pair (s_α, a_α) , which is equal to one, so that $\eta_0 = \eta_0(s_\alpha, a_\alpha)$ is the advantage of the candidate optimal action a_α ; constants \mathbf{h}_t are defined in Lemma 3.2.

Program (NS-D3) is derived as follows. By Proposition 3.3, if an optimal decision rule $a_{\star, \alpha} \neq a_\alpha$ exists for some salvage vector \mathbf{u} , the reduced cost $\eta_0 \triangleq \eta_0(s_\alpha, a_\alpha)$ in the original program NS-D will be negative. We can check if η_0 can be made less than zero by minimizing it for all feasible values of \mathbf{v} and variables of the primal-dual program pair (NS-P)–(NS-D). By Propositions 3.1 and 3.3, only the sign of \mathbf{z}_t is important: if $z_t(s, a) > 0$, then $\eta_t(s, a) = 0$, and if $z_t(s, a) = 0$, $\eta_t(s, a) < 0$. Thus, we can replace $z_t(s, a)$ with binary variables $\hat{z}_t(s, a) \triangleq \text{sgn } z_t(s, a)$. The integer variables $\hat{z}_t(s, a)$ ensure that the found solution is a feasible solution to the dual program that corresponds to a deterministic policy.

The constraints of the program serve the following purposes. The expressions $\mathbf{r}_t - \mathbf{N}\mathbf{v}_t + \gamma \mathbf{T}_t \mathbf{v}_{t+1}$ in the first two constraints are equal to the advantages η_t . Whenever $\hat{z}_t(s, a) = 1$, the corresponding constraint becomes tight and ensures that $\eta_t(s, a) = 0$.

When $\dot{z}_t(s, a) = 0$, the left-hand side of the corresponding constraint becomes equal to $-h_t(s, a)$, and $\eta_t(s, a) > -h_t(s, a)$ always holds as per Lemma 3.2. Constraint $\mathbf{N}^\top \dot{\mathbf{z}}_T = \mathbf{1}$ is equivalent to $\sum_{a \in A_p(s)} \dot{z}_t(s, a) = 1$ for all $s \in \mathbb{S}$. It ensures that Proposition 3.1 holds.

Next, $\mathbf{j}_\alpha^\top \dot{\mathbf{z}}_0 = 0$ forces the program to search for policies with $\pi_{\star, \mathbf{v}, T, 0}(s_\alpha) \neq a_\alpha$. This constraint makes the program infeasible if no actions other than a_α are available for s_α , $A_p(s_\alpha) = \{a_\alpha\}$. In this case a_α is also optimal as the only possible action.

We add constraints $\mathbf{v}_t \in \mathbb{U}_t$ to the formulation of Bean et al. [1992], because we assume that the optimal value vector $\mathbf{v}_{\star, t}$ is known to belong to the space \mathbb{U}_t . These new constraints help with speeding up computations by reducing the search space for variables \mathbf{v}_t . For an appropriate choice of the spaces \mathbb{U}_t , they can also guarantee that the constraint NS-D.2 holds.

We exclude constraints $\mathbf{v}_t \geq \mathbf{0}$ from the formulation of Bean et al. [ibid.], as this assumption does not hold in our case. The non-negativity assumption was used to show that η_0 is zero only when a_α is optimal, but Proposition 3.1 already guarantees this.

Finally, we would like to note that it is not strictly necessary to solve the optimization problem: if at any iteration the solver finds a feasible solution with negative value of the objective function, it can proceed to the next study horizon.

In order to implement the program (NS-D3) the salvage spaces \mathbb{U}_t need to be polytopes (that is, we should be able to express them using sets of linear constraints). In the next subsection we provide such subspaces under Assumption 3.1.

3.4.2 Unbounded Rewards

To implement the program (NS-D3) we need to be able to construct the salvage subspaces \mathbb{U}_t so that:

- they can be expressed via linear constraints,
- they contain the optimal values $\mathbf{v}_{\star, t} \in \mathbb{U}_t$, and
- $\mathbf{v}_t \in \mathbb{U}_t$ implies that (3.1) holds.

If the value bounding functions u_+ and u_- of Definition 3.4 exist and are known, we can consider a sequence of spaces

$$(\mathbb{U}_t)_{t \in \mathbb{N}_0}, \mathbb{U}_t \subseteq \mathbb{R}^{|\mathbb{S}|} \quad \text{where} \quad \mathbb{U}_t = \{\mathbf{v} \mid \mathbf{u}_{-, t} \leq \mathbf{v} \leq \mathbf{u}_{+, t}\}. \quad (3.3)$$

These spaces \mathbb{U}_t are indeed defined by linear constraints. By Proposition 3.4, they contain the optimal values $\mathbf{v}_{\star, t} \in \mathbb{U}_t$. Finally,

since u_- and u_+ belong to the space $L^w(\mathbb{S})$, so does \mathbf{v} by the squeeze theorem. As the truncation horizon T increases, the ranges of possible optimal initial values shrink monotonically by Proposition 3.4, until eventually all of the truncations begin to agree in the optimal initial decision as per Proposition 3.6.

3.4 A Stopping Rule

Unfortunately, existence of such functions is guaranteed only in the uniformly bounded case. Unless additional information can be exploited to obtain such bounding functions, the only bounds on $\mathbf{v}_{\star,t}$ are provided by (2.28) and the only salvage spaces that we can use are

$$\mathbb{U}_t = \{\mathbf{v} \mid -\mu \cdot \mathbf{w}_t \leq \mathbf{v} \leq \mu \cdot \mathbf{w}_t\}.$$

These bounds are no longer value bounding functions in the sense of Proposition 3.4, because the conditions $\mathcal{L}_{\pi,u,T}u_- \geq u_-$ and $\mathcal{L}_{\pi,u,T}u_+ \leq u_+$ are not guaranteed to hold. While a series of truncations with salvage spaces $(\mathbb{U}_t)_{t \in \mathbb{N}_0}$ will provide an optimal solution eventually, the convergence is no longer monotone, and larger truncations may no longer tighten the constraints in (NS-D3). This is undesirable, as it may lead to unnecessary approximations that are worse than already considered ones.

Nonetheless, the functions $\pm \mu \cdot w$ are the only information about the problem available under Assumption 3.1, so we want to establish similar convergence properties for them. To do so, we show the following property of the Bellman operator \mathcal{L}_π .

Lemma 3.7 * monotonicity of the multi-stage Bellman operator

For all $\pi \in \mathbb{D}$, functions $u_\pm = \pm \mu \cdot w$ satisfy $\mathcal{L}_\pi^\nu u_- \geq u_-$ and $\mathcal{L}_\pi^\nu u_+ \leq u_+$.

Proof. We prove the statement for u_+ ; the proof for u_- is identical. By applying \mathcal{L}_π to the function u_+ consecutively ν times, we obtain

$$[\mathcal{L}_\pi^\nu u_+]_t(s) = \sum_{i=0}^{\nu-1} \gamma^i \cdot [\mathcal{T}_{\pi,t}^i r_\pi]_{t+i}(s) + \gamma^\nu \mu \cdot [\mathcal{T}_{\pi,t}^\nu w]_{t+\nu}(s).$$

Note that $\mu = \sum_{i=0}^{\nu-1} \kappa^i + \lambda \mu$ by rearranging the terms in (2.29). Then for all $t \in \mathbb{N}_0$, by recalling (2.26) and (2.27),

$$\begin{aligned} [\mathcal{L}_\pi^\nu u_+]_t(s) &\leq \sum_{i=0}^{\nu-1} \kappa^i \cdot w_t(s) + \lambda \mu \cdot w_t(s) \\ &= \mu \cdot w_t(s) = u_{+,t}(s). \end{aligned} \quad \text{QED}$$

Proposition 3.4 uses the operators \mathcal{L}_π to show that one-stage increments in study horizons lead to monotone convergence. In

3 The Infinite-Horizon Non-Stationary Model

In Russian, *Muuua* (Misha) is a diminutive form of the name *Muxaun* (Michael). It also means *a little bear*; the mascot of the 1980 Olympic Games being a famous example.

Figure 3.1:
MISHA—multi-step-
iterated solution-horizon
algorithm.

the unbounded case, Lemma 3.7 shows that similar properties hold if instead of looking only one stage ahead, the decision-maker chooses v -stage increments in study horizons, as the Bellman operator \mathcal{L}_π is now a v -stage contraction instead of a contraction. This is a crucial property leveraged by our algorithm; it ensures that the space of possible initial values decreases with each iteration, and the algorithm converges monotonically.

3.4.3 The Algorithm

Summarizing the aforementioned results, we present MISHA: the *multi-stage iterated solution-horizon algorithm*. of Figure 3.1. It is guaranteed to terminate in a finite number of steps if the optimal policy is unique. Moreover, when better value bounding functions are known, they can be used instead of $\pm\mu \cdot w$ to provide smaller salvage subspaces \mathbb{U}_t , resulting in faster convergence.

Data: an non-stationary MDP with a bounding function w .

Result: an optimal initial action $a_{\star,\alpha}$ and a solution horizon T_\star .

```

1  Let  $u_+ \leftarrow \mu \cdot w$  and  $u_- \leftarrow -\mu \cdot w$ ;
2  for  $N \leftarrow 1, 2, \dots$  do
3       $T \leftarrow N \cdot v - 1$ ;
4      find a candidate optimal initial action  $a_\alpha$  by. solving
        (NS-D-ALT) with any salvage vector  $\mathbf{u} \in \mathbb{U}_{T+1}$ ;
5      solve (NS-D3) with the spaces  $\mathbb{U}_t$  given by (3.3);
6      if (NS-D3) is infeasible or  $\eta_0 = 0$  then
7           $a_{\star,\alpha} \leftarrow a_\alpha$ ;
8           $T_\star \leftarrow T$ ;
9          break
10 for  $n \leftarrow 1, \dots, v - 1$  do
11      $T \leftarrow T_\star - n$ ;
12     solve (NS-D3) with  $\mathbb{U}_t$  given by (3.3);
13     if (NS-D3) is feasible and  $\eta_0 < 0$  then
14          $T_\star \leftarrow T + 1$ ;
15         return  $a_{\star,\alpha}$  and  $T_\star$ ;
16     break
```

- ☞ The algorithm searches for a solution horizon by doing v -stage increments in study horizons. For each of these horizons it checks if all of the feasible truncations agree in the initial optimal decision. Once a solution horizon has been identified, the algorithm returns back in time, up to the previous considered study horizon. It does

so in order to identify possible shorter solution horizons.

3.5 EXPERIMENTS

3.5 Experiments

To demonstrate the performance of our stopping rule, we implemented Algorithm 3.1 for the following problem, known as an equipment replacement problem of Bean et al., 1992.

Consider a piece of equipment subject to deterioration. The state space $\mathbb{S} = \{0, \dots, |\mathbb{S}| - 1\}$ represents the state of its decay, with 0 being “new.” At each time step, the agent chooses between two actions: “replace” (action 0) and “keep” (action 1).

Transition probabilities of the problem are given by

$$p_t(s' | s, 0) = \begin{cases} 1, & s' = 0, \\ 0, & \text{otherwise;} \end{cases}$$

$$p_t(s' | s, 1) = \begin{cases} 1 - \psi, & s' = s < S, \\ \psi, & s' = s + 1, s' < S, \\ 1, & s' = s = S, \\ 0, & \text{otherwise,} \end{cases}$$

where ψ is the deterioration probability. If the equipment is replaced, the state always changes to 0 (that is, “new”). Otherwise, it either deteriorates to the next state (if there is one) with probability ψ , or remains the same state with probability $1 - \psi$.

In the first experiment we used the following rewards:

$$r_t(s, 0) = \rho \cdot (-0.5N^{\min\{t/T, 1\}} + (|\mathbb{S}| - s + 1)/\Delta);$$

$$r_t(s, 1) = \rho \cdot (N^{\min\{t/T, 1\}} - s/\Delta).$$

Figure 3.2 outlines the general reward structure. When the equipment is kept, it generates revenue which depends on the

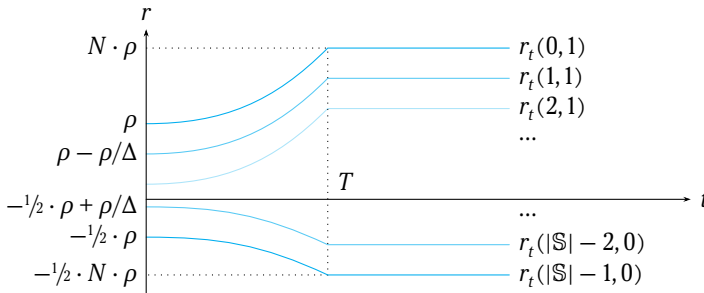


Figure 3.2: Rewards in the equipment replacement problem

3 The Infinite-Horizon Non-Stationary Model

state of deterioration and grows over time. If the equipment is new, the initial revenue $r_0(0,1)$ is equal to ρ and it grows exponentially (for example, due to inflation). For each stage of deterioration the revenue decreases by ρ/Δ . When the equipment is replaced, it generates no revenue, and a replacement cost needs to be paid. The costs behave similarly to revenues, and the worse is the state of the equipment, the larger are the costs. We limit the data at time step T , when it becomes equal to $N \cdot \rho$ to add uniform bounds so that the method of Hopp [1989] can be applied as well for comparison.

Function $w_t = \rho \cdot N^{\min\{t/T, 1\}}$ satisfies Condition 2.5 with $\kappa = \gamma \cdot N^{1/T}$. Assuming $T > -\log_\gamma N$, $\lambda = \kappa$ and $v = 1$, so functions $\pm \mu \cdot w_t$ can be used as bounds for the state values. These are *loose bounds*, as they don't use any additional information. The following functions can be used as tighter bounding functions:

$$u_{+,t} = \sum_{\tau=0}^{\infty} \gamma^\tau \cdot \max_{(s,a) \in \mathbb{X}} r_{t+\tau}(s,a) = \sum_{\tau=0}^{\infty} \gamma^\tau \cdot r_{t+\tau}(0,1),$$

$$u_{-,t} = \sum_{\tau=0}^{\infty} \gamma^\tau \cdot \min_{(s,a) \in \mathbb{X}} r_{t+\tau}(s,a) = -\frac{1}{2} \cdot u_{+,t}.$$

These *tighter bounds* are easy to compute and result in smaller search spaces \mathbb{U}_t , making the problem easier to solve. In practice, for a truly non-stationary problem such closed-form bounds will not be available, therefore they can be seen as a bound of what can be achieved without exploiting any additional information on the exact reward structure.

Stationarity of the rewards after the capping horizon T allowed us to find the exact solution of the problem. We started at time horizon T and solved the problem using value iteration, then used dynamic programming to obtain the initial optimal decision.

We compared our stopping rule for both choices of the bounding functions to Hopp's rule. We ran the experiments for different combinations of parameters. For all of them, both stopping rules identified the optimal initial action correctly but discovered different solution horizons. In almost all of the experiments, our stopping rule was able to find a significantly shorter solution horizon. The default values are listed in Table 3.1. These values were used in all of the experiments, unless stated otherwise.

Figure 3.3 shows how the solution horizons and run times scale with respect to the number of states $|\mathbb{S}|$. Both algorithms need to look further into the future as the problem size grows, however, our stopping rule identifies significantly shorter solution horizons.

s_α	1
S	10
N	10
T	1000
γ	0.95
Δ	45
ψ	0.4
ρ	1

Table 3.1: Values of the hyperparameters.

Shorter horizons mean that less mixed-integer programs need to be solved, which substantially reduces the run-time.

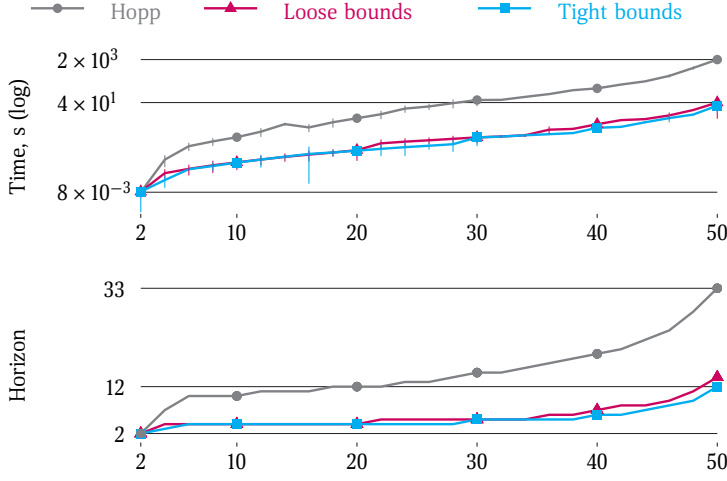


Figure 3.3: Performance with respect to the state space size $|\mathbb{S}|$.

Figure 3.4 presents the effect of the model uncertainty ψ on the algorithm. The largest difference in performance is exhibited when $\psi = 0.5$, that is, when the system's entropy is the largest.

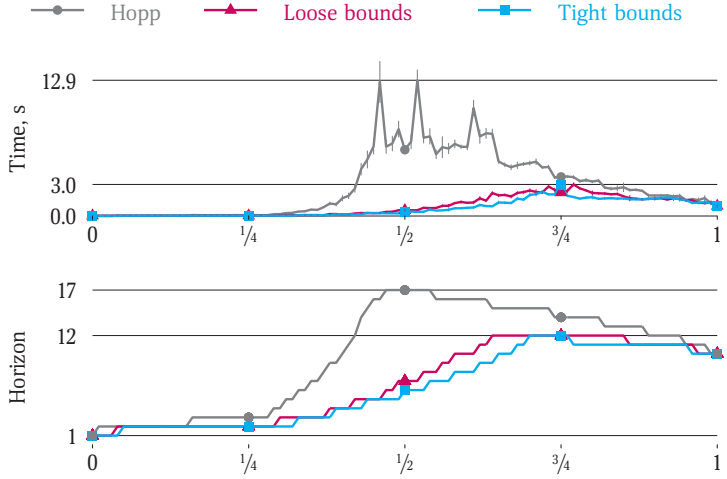


Figure 3.4: Performance with respect to the deterioration rate ψ .

In the second experiment, we set $|\mathbb{S}| = 5$ and used the same transition matrices but different rewards. We randomly generated the initial rewards $\mathbf{r}_{a,0}$ from the following sets

$$\mathbf{r}_{0,0} \in [-0.5N, 0)^{|\mathbb{S}|}, \quad \mathbf{r}_{1,0} \in [0, N)^{|\mathbb{S}|}.$$

3 The Infinite-Horizon Non-Stationary Model

Subsequent rewards were given by $\mathbf{r}_{a,t+1} = \Phi_a \mathbf{r}_{a,t} = \Phi_a^t \mathbf{r}_{a,0}$, where Φ_a are tri-diagonal matrices with non-zero elements drawn from uniform distribution on $[-1, 1)$, and then scaled so that spectral radii σ_a of Φ_a were less than one. The latter condition was added to ensure that the problem has a bounding function w . These spectral radii are similar to discounting factors for matrices, because they indicate the rate of growth of the matrix power series; therefore for problems with $\sigma = \max\{\sigma_1, \sigma_2\} \geq 1$ the values \mathbf{v}_t may not be well-defined.

The rewards of this problem are bounded by the function

$$w_t(s) = N \cdot w_t, \quad \text{where} \quad w_t \triangleq \max\{\|\Phi_1^t\|_{\infty \rightarrow \infty}, \|\Phi_2^t\|_{\infty \rightarrow \infty}\}.$$

$\|\mathcal{Y}\|_{p \rightarrow q}$ is the norm of the operator $\mathcal{Y} : \mathbb{Y} \rightarrow \mathbb{Y}'$, that is, the smallest constant that satisfies $\|\mathcal{Y}y\|_q \leq \|\mathcal{Y}\|_{p \rightarrow q} \cdot \|y\|_p$ for any $y \in \mathbb{Y}$.

For matrices, $\|\cdot\|_{\infty \rightarrow \infty}$ is equal to the maximum of the absolute row sums.

with the following coefficients:

$$\nu = \min_{j \in \mathbb{N}_0} \{j \mid \gamma^j \cdot \|\Phi_1^j\|_{\infty \rightarrow \infty} < 1 \wedge \gamma^j \cdot \|\Phi_2^j\|_{\infty \rightarrow \infty} < 1\},$$

$$\kappa = \gamma \cdot w_1, \quad \lambda = \gamma^\nu \cdot w_\nu.$$

Existence of ν is guaranteed by the following property of spectral radii: $\sigma_a = \lim_{t \rightarrow \infty} \|\Phi_a^t\|_{\infty \rightarrow \infty}^{1/t}$. As a result, for any $\sigma < 1$ the operator norm $\|\Phi_a^t\|_{\infty \rightarrow \infty}$ becomes less than one eventually.

When $\kappa < 1$, the problem can be transformed into a bounded problem by using (2.30). In this case we are able to solve the problem using Hopp's stopping rule as well. In this experiment the data was truly non-stationary, and it was impossible to compute value functions exactly. When Hopp's stopping rule was able to find a solution horizon, we knew that the action it identified was indeed optimal and used it as a benchmark for MISHA.

The results are presented in Figure 3.5. In all of the experiments our method was able to identify the optimal initial action. In these cases MISHA always returned the same horizon as Hopp's stopping rule. This can be explained by the fact that the methods are similar: after the transformation is applied to the problem the salvage spaces \mathbb{U}_t become identical at all time steps, just like in the case of Hopp's stopping rule.

Nevertheless, MISHA runs faster, as, on the one hand, it does not require the data transformation, and on the other hand, it uses large steps ν when searching for the solution horizon, reducing the number of iterations by a factor of ν . Moreover, it is applicable to a wider range of problems; for example, Hopp's stopping rule cannot be used in problems with a large spectral radius, as illustrated by Figure 3.5.

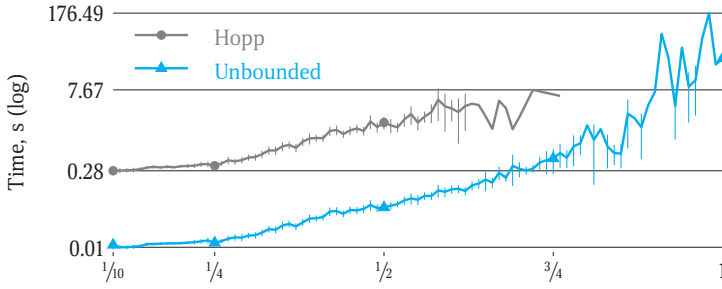


Figure 3.5: Performance with respect to the spectral radius σ .

3.6 CONCLUSION

Infinite-horizon non-stationary Markov decision processes cannot be solved using traditional methods because they seek universally optimal policies. Finding such a policy would require infinite computations, because the optimality needs to be established for each of the infinitely many state-time pairs.

At the same time, the decision-maker may not be interested in optimality of all of the decision rules at all. Often, the decision involves only a single state: the one that is observed right now and requires immediate action. By seeking initial-decision optimal policies instead of the universally optimal ones, we can overcome the curse of infinite dimensionality of the decision-making problem. Once an optimal initial decision was identified, the same procedure can be used to find an optimal action for the next decision epoch.

We propose one such algorithm for finding optimal initial decision rules. This algorithm uses a stopping rule to discover a solution horizon: a time horizon removed so far into the future, that the data beyond it does not affect the decision that needs to be made right now. The rule is applicable to problems with unbounded rewards and does not require any additional assumptions on the reward structure, such as convexity of rewards, making it applicable to a broad class of problems.

An experimental study shows that our stopping rule was able to find better solution horizons and did it faster even when the rewards can be uniformly bounded.

Future research directions include an extension to problems with countably-infinite base state spaces, as the problem is already countably-infinite in the time domain. Additionally, the rate of convergence may be improved by considering span-based bounds in combination with weighted-supremum ones.

See the epigraph to this chapter.

4

The Countably-Infinite Model

*I am incapable of conceiving infinity,
and yet I do not accept finity.*

— Simone de Beauvoir,
The Coming of Age

Translated by P. O'Brian

THE PREVIOUS CHAPTER presented an algorithm for non-stationary infinite-horizon MDP that is based on a reformulation of the original problem as a countably-infinite stationary one. This is a much broader class of decision processes that remains almost unexplored due to computational hurdles associated with countably-infinite optimization. In this chapter, we further develop the theory of truncations in countably-infinite MDPs, and design an algorithm for such problems. Like MISHA—the algorithm of Chapter 3—it is applicable to problems with unbounded rewards; unlike it, the new algorithm belongs to the family of policy iteration methods. It performs sequential policy-improving updates while eliminating provably suboptimal actions; this procedure continues until unique optimal decision rules are obtained for each state in the support of the initial distribution. This allows the decision-maker to plan ahead of time and be prepared for different possible states of the environment at the moment when the decision needs to be made.

4.1 Introduction

4.1 INTRODUCTION

In the previous chapter, we considered non-stationary MDPs and showed that they can be written as countably-infinite stationary MDPs. Similarly to the non-stationary case, solution methods for such decision-making processes employ approximations known as truncations. A truncation contains only a finite subspace of states, and policy optimization is performed over these; for the remaining states, the policy is set arbitrarily. This approach is used in most of the existing methods for countably-infinite MDPs [Lasserre and Bès, 1984; Cavazos-Cadena, 1986; Hopp et al., 1987; Hopp, 1989; Bean et al., 1992; Lee et al., 2017].

Once a truncation is solved, the quality of the resulting approximate solution can be evaluated. If the approximation needs improvement, a larger truncation is considered, and the process repeats until a sufficiently good truncation is found. While the basic idea of this scheme sounds simple, the infinite-dimensional nature of the underlying problem presents unique challenges. For example, Proposition 2.3 presented in Chapter 2 tells us that even something as simple as a change of summation order may not be possible and requires careful consideration.

Because of the underlying challenges, research in countably-infinite MDPs remains limited and relies on some additional as-

For a more detailed explanation, the reader is referred to the works of Ghatge [2015] and Lee et al. [2017].

sumptions about the underlying problem.

See Lemma 2.23, p. 2.23.

First, uniformly boundedness of rewards may be assumed to guarantee existence of optimal policies. Unfortunately, this condition does not hold for some countably-infinite MDPs, including the inventory management problem of Sections 1.3.2 and 2.4.5.

MISHA uses multi-step truncation enlargements to guarantee improvements; this is possible because of the special state-space structure in non-stationary problems, where the time dimension is not recurrent.

Next, a desirable property of a solution method is monotonic improvement of the policy it produces [Lee et al., 2017]: each iteration should result in a policy that *strictly improves* the previous one. In problems with unbounded rewards, an increase in the truncation size no longer guarantees that the resulting solution is better than the previously found one. This happens because the Bellman operator is no longer a contraction, and additional data may result in a solution that is further from the optimum. It is therefore desirable to have methods that are monotonically improving the solution.

The contraction horizon is defined in Condition 2.5, p. 52.

Finally, many of the existing truncation-based methods assume—mostly implicitly—that the state space is totally ordered and lower bounded. For example, non-negative integer numbers $\mathbb{S} = \mathbb{N}_0$ satisfy these conditions. This assumption produces a natural truncation improvement scheme: whenever the current truncation is not sufficiently good, add the smallest missing state to it. For example, Lee et al. [ibid.] use this approach. A similar scheme is used in the previous chapter for non-stationary MDP, but instead of adding one state at a time, we added sets of all states reachable within the contraction horizon. In practice, the state space can have a more complex structure. For example, in the multi-product inventory management problem, the states are multi-dimensional vectors. This calls for truncation enlargement schemes that do not rely on a total order, but still improve the approximation.

To summarize, our goal is to design an iterative truncation-based solution scheme for countably-infinite MDPs that:

- is applicable to problems with unbounded rewards;
- monotonically improves the baseline policy;
- does not require the state space to be totally ordered.

In this chapter, we present such an algorithm. We name it *ASPIRE—approximate salvage-based policy iteration with repeated elimination (of actions)*. It combines the ideas of Chapter 3 and Lee et al. [ibid.] and uses the duality of occupancies and values in countably-infinite MDPs and the theory of contractions in Banach spaces.

See Figure 3.1, p. 78.

Like the previously discussed MISHA, ASPIRE is based on a

search of salvage function that may result in better initial decisions. In the case of non-stationary MDPs this problem is a linear program with salvage vectors as the optimization variables. In the countably-infinite case, this approach leads to programs with countably many variables and thus it is rendered useless. Instead, ASPIRE utilizes an analytical solution to this linear program, which is computable under some additional conditions.

Like the simplex method of Lee et al. [ibid.], ASPIRE is a policy-iterating algorithm that utilizes approximate advantages to reason about optimality of actions. While the method of Lee et al. [ibid.] uses zero salvages only, ASPIRE reasons about all of the possible salvages simultaneously, which allows us to introduce a rule for action elimination: if an action is suboptimal for all possible salvages, this includes the case when the salvage coincides with the true value function; as a result, the action does not need to be considered anymore.

4.1.1 Previous Work

Solution methods for countably-infinite MDPs with unbounded rewards can be traced back to the works of Harrison [1972], Lippman [1975], and Wessels [1977]. Each of these papers considered a different set of assumptions on rewards and transition probabilities. These sets of assumptions are referred to as *settings* by Lee et al. [2017]. White [1982] proposed a generalization of the three settings, which in turn was even further generalized by the setting of Cavazos-Cadena [1986].

Unfortunately, the setting of Cavazos-Cadena [ibid.] utilizes a value bounding function which cannot be computed finitely [Lee et al., 2017]. Due to this observation, the setting of White [1982], as presented in Puterman [1994, Section 6.10], remains the most commonly employed one.

For this setting, which was formally introduced as Condition 2.5, Lee et al. [2017] proposed a simplex-based algorithm mentioned earlier. Starting with an arbitrary policy, their algorithm evaluates it approximately. Then, it estimates how much each alternative action can improve the current policy. If an improvement in the approximate value is large enough, it is guaranteed to improve the true values as well, and the policy is updated. Otherwise, the truncation is expanded and the policy is re-evaluated. Thus, this method can be seen as an extension of the policy iteration algorithm to the countably-infinite MDP case.

The salvage function approximates the value function outside of the truncation.

The value bounding function is denoted by \mathfrak{B} by Cavazos-Cadena [ibid.].

See p. 52.

To the extent of our knowledge, the method of Lee et al. [2017] is the only method applicable to countably-infinite MDPs in general, but other method exists for problems with special structures. For example, Ghate and R. L. Smith [2013] considers non-stationary MDPs with uniformly bounded rewards and proves that strong duality holds in this case.

In addition to these algorithmic approaches to countably-infinite MDPs, some authors study theoretical aspects of such problems. For example, Puterman [1994, Section 6.10] employs the idea of multi-stage contractions to show that the value function exists uniquely under Condition 2.5 even though the Bellman operator is no longer a contraction; Hernández-Lerma and Lasserre [2002] provide conditions for duality of occupancies and values in countably-infinite MDPs.

4.2 MODEL ASSUMPTIONS

Similarly to non-stationary problems of Chapter 3, a set of assumptions is required to establish existence of optimal policies in countably-infinite MDPs. The first two assumptions guarantee that the problem is well-posed.

Assumption 4.1 | weight function exists

Cf. Assumption 4.1,
p. 90.

A weight function $w : \mathbb{S} \rightarrow \mathbb{R}_+$ of Condition 2.5 exists but does not have to be known to the agent.

Assumption 4.2 | finiteness of actions

For any state $s \in \mathbb{S}$, the set of permitted actions $A_p(s)$ is finite, $|A_p(s)| < \infty$.

☞ In Chapter 3 we required that the initial state is observed by the agent and the decision is made based on this observation. While this allows the agent to act optimally by re-planning every time a decision needs to be made, there may not be enough time to plan in such a reactive way. Instead, if the initial state distribution is known, the agent can plan proactively by finding an optimal decision for each of the possible states. The following assumption guarantees that this is possible.

Assumption 4.3 | finite support of initial state distribution

Cf. Assumption 3.3
p. 68.

The initial state distribution α has a finite support, $|\text{supp } \alpha| < \infty$.

☞ Conditions 2.5, 2.4 and 2.6 hold under Assumptions 4.1, 4.2, and

4.3 respectively. Together, they ensure that Theorem 2.22 (the duality theorem) holds.

When the weight function w exists, the value functions are absolutely bounded by $\mu \cdot w$. In Chapter 3 these bounds were used to design the salvage spaces, that is, the spaces of potential future values. Sometimes, tighter bounds may exist and be easier to identify. For example, if the rewards are positive, the lower bound of zero is trivial and tighter than $\mu \cot w$. Therefore, we assume that some bounds are available to the agent that may not be related to the weight function w .

4.2 Model Assumptions

Assumption 4.4 | value bounds exist in $L^w(\mathbb{S})$

The agent knows the value-bounding functions u_{\pm} such that $u_{-} \leq v_{\pi} \leq u_{+}$ for any stationary deterministic policy $\pi \in \mathbb{D}$. Moreover, these functions are assumed to have finite w -weighted supremum norms, $u_{\pm} \in L^w(\mathbb{S})$.

Assumption $u_{\pm} \in L^w(\mathbb{S})$ is required so that the possible value functions are restricted to the space $L^w(\mathbb{S})$ of functions with finite w -weighted supremum norm, which is required for the duality to hold.

The following assumption introduces the necessary properties for initial optimal decisions to be identifiable starting with one the truncations. It requires two additional definitions.

Definition 4.1 | monotone-increasing sequence

A sequence $(\mathbb{Y}_k)_{k=0}^{\infty}$ of subsets $\mathbb{Y}_k \in \mathbb{Y}$ is called *monotone increasing* if each element is a superset of the previous one:

$$\mathbb{Y}_0 \subset \mathbb{Y}_1 \subset \dots \subset \mathbb{Y}_{k-1} \subset \mathbb{Y}_k \subset \mathbb{Y}_{k+1} \subset \dots$$

Definition 4.2 | limiting set

The *limiting set* \mathbb{Y}_{∞} of a monotone-increasing sequence $(\mathbb{Y}_k)_{k=0}^{\infty}$ is defined as

$$\mathbb{Y}_{\infty} = \lim_{k \rightarrow \infty} \mathbb{Y}_k \triangleq \bigcup_{i=0}^{\infty} \mathbb{Y}_i.$$

Assumption 4.5 | properties of the salvage spaces

There exists a monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^{\infty}$ of finite subspaces of the state space, $\mathbb{S}_k \in \mathbb{S}$, such that its limiting set \mathbb{S}_{∞} is a superset of the the initial distribution support, $\text{supp } \alpha \subseteq \mathbb{S}_{\infty}$. Additionally, for any stationary deterministic policy $\pi \in \mathbb{D}$

$$\lim_{k \rightarrow \infty} \sup_{s \in \mathbb{S}_{\infty}} \left(\sum_{s' \in \mathbb{S}_k^c} p_{\pi}(s' | s) \cdot w(s') \right) = 0. \quad (4.1)$$

For a subset $\mathbb{B} \subseteq \mathbb{Y}$ of a universe \mathbb{Y} , \mathbb{B}^c denotes its complement, $\mathbb{B}^c \triangleq \mathbb{Y} \setminus \mathbb{B}$.

☞ The last part of this assumption requires that the truncations increase in such a manner that the probabilities to transition outside of the truncation decrease faster than the weight function w grows there.

Finally, the following assumption guarantees that the optimal initial decisions are unique and therefore identifiable as optimal by our algorithm.

Assumption 4.6 | optimal initial decision rules are unique

All optimal policies $\pi_\star \in \mathbb{D}$ have the same decision rules $a_{\star,\alpha}(s)$ in the support of the initial distribution, $\pi_\star(a|s) = \delta_{a,a_{\star,\alpha}(s)}$ for all $s \in \text{supp } \alpha$ and $a \in A_p(s)$.

4.3 A MOTIVATING EXAMPLE

We now re-examine the multi-product inventory management problem of Sections 1.3.2 and 2.4.5.

In Section 2.4.5, we established that no uniform reward bound exists in this problem, as stated by Lemma 2.23. At the same time, the weight function w exists by Lemma 2.24, and therefore Assumption 4.1 holds. Assumption 4.2 holds as well because the order size is limited. Assumption 4.3 can be checked easily when the initial distribution is known.

Assumption 4.4 holds due to the following lemma.

Lemma 4.1 * value bounds in inventory management

In the multi-product inventory management problem, for any policy $\pi \in \Pi$ the value $v_\pi(\mathbf{s})$ of each state $\mathbf{s} \in \mathbb{S}$ is bounded by the functions $u_\pm \in L^w(\mathbb{S})$:

$$-\mu \cdot w(\mathbf{s}) \leq u_-(\mathbf{s}) \leq v_\pi(\mathbf{s}) \leq u_+(\mathbf{s}) \leq \mu \cdot w(\mathbf{s}),$$

$$\text{where } u_-(\mathbf{s}) \triangleq -\frac{1}{1-\gamma} \cdot \langle \mathbf{h}, \mathbf{s} \rangle - \frac{C_O - \gamma \cdot (C_O - C_H)}{(1-\gamma)^2} \quad (4.2)$$

$$\text{and } u_+(\mathbf{s}) \triangleq \frac{C_G}{1-\gamma}. \quad (4.3)$$

The constants C_G , C_O , and C_H are defined in Lemma 2.24

☞ The proof of Lemma 4.1 is presented in Appendix A.4.

Next, Assumption 4.5 requires us to construct a sequence of finite truncation sets. Let $\mathbb{S}_0 = \text{supp } \alpha$. It is finite by Assumption 4.3. For any \mathbb{S}_k , define the next truncation set \mathbb{S}_{k+1} as the set

of all states in \mathbb{S}_k as well as all states reachable from \mathbb{S}_k in one step:

$$\mathbb{S}_{k+1} \triangleq \mathbb{S}_k \cup \{s' \in \mathbb{S} \mid p(s' \mid s, a) > 0 \text{ for some } a \in A_p(s)\}. \quad (4.4)$$

4.4 Policy Evaluation

In this case, the limiting set coincides with the state space $\mathbb{S}_\infty = \mathbb{S}$. For any state $s \in \mathbb{S}$ the complement \mathbb{S}_k^c becomes unreachable starting with some N , and the sum in (4.1) becomes equal to zero for any $k \geq N$, therefore, (4.1) holds.

Assumption 4.6 is the only assumption that is cannot be guaranteed to hold without further analysis.

The inventory management problem belongs to a class of problems with *limited state reachability*: for any state $s \in \mathbb{S}$, only finitely many other states can be reached in the next time step.

Countably-infinite reformulations of non-stationary MDPs used in Chapter 3 have the same property: the resulting augmented state space $\tilde{\mathbb{S}}$ is indeed countably infinite, but for any augmented state $\tilde{s} = (s, t), s \in \mathbb{S}$, only the augmented states of the next time step $\tilde{s}' = (s', t + 1), s' \in \mathbb{S}$ are reachable.

Similarly to the inventory management problem, the iterative procedure given by (4.4) can be used for any limited-reachability problem to produce a monotone-increasing sequence of truncation spaces starting with $\mathbb{S}_0 = \text{supp } \alpha$.

4.4 POLICY EVALUATION

To design an algorithm for countably-infinite MDPs, let us first consider the task of evaluating a given policy π . This is necessary to ensure that we can evaluate policies and guarantee that policy iteration improves them.

In finite-dimensional problems, policy evaluation can be done by finding the fixed point v_π of the Bellman operator \mathcal{L}_π . In the countably-infinite case, this cannot be done. A common approach is to consider a finitely computable approximation to the Bellman operator \mathcal{L}_π to obtain approximate values. In this section, we present one such operator, and show that it can be used to create a series of approximations converging to v_π pointwise within a given subspace.

The presentation of this section relies heavily on the theory of contraction in Banach spaces. For a reminder of the notation and properties of contractions, see Section 2.3.2 p. 42.

4.4.1 Truncated Bellman Operator

4 The Countably-Infinite Model

In the previous chapter, we considered an approximation called a (u, T) -truncation. In this approximation, the evaluation is done up to time T , and is approximated with a function u called the salvage afterwards. Essentially, the truncation horizon T separated the time-augmented state space $\mathbb{S} = \{\tilde{s} = (s, t) \mid s \in \mathbb{S}, t \in \mathbb{T}\}$ into two subspaces. We can use the same approach in countably-infinite spaces by defining a finite subspace $\mathbb{B} \subseteq \mathbb{S}$ in which the evaluation takes place. We call the resulting approximation a (u, \mathbb{B}) -truncation.


Definition 4.3 | (u, \mathbb{B}) -truncated Bellman operator

For a given function $u : \mathbb{S} \rightarrow \mathbb{R}$ and a finite subspace of states $\mathbb{B} \subseteq \mathbb{S}$, $|\mathbb{B}| < \infty$, a (u, \mathbb{B}) -truncated Bellman operator $\mathcal{L}_{\pi, u}^{\mathbb{B}} : \mathbb{R}^{\mathbb{S}} \rightarrow \mathbb{R}^{\mathbb{S}}$ of a policy π is an operator given by

$$[\mathcal{L}_{\pi, u}^{\mathbb{B}} v](s) \triangleq \begin{cases} r_{\pi}(s) + \gamma \cdot \sum_{s' \in \mathbb{S}} p_{\pi}(s' | s) \cdot (v(s') \cdot \mathbb{I}_{\{s' \in \mathbb{B}\}} \\ \quad + u(s') \cdot \mathbb{I}_{\{s' \notin \mathbb{B}\}}), & \text{if } s \in \mathbb{B}, \\ u(s), & \text{otherwise.} \end{cases}$$

We call u a *salvage function* and \mathbb{B} a *truncation set* of a (u, \mathbb{B}) -truncation, and the resulting MDP a (u, \mathbb{B}) -truncation of the original problem.

Compare to
Definition 3.3 p. 72.

When $\mathbb{S} = \mathbb{N}_0$ and $\mathbb{B} = \{0, 1, \dots, N\}$ such an approximation is known as an N -state approximation to v_{π} [Puterman, 1994]. Even though each countably-infinite space is denumerable in this way, we want to develop an algorithm that does not rely on such a denumeration. 

By first summing over the states s' in the subspace \mathbb{B} and then over the rest of the states, we can rewrite the (u, \mathbb{B}) -truncated Bellman operator $\mathcal{L}_{\pi, u}^{\mathbb{B}}$ as

$$[\mathcal{L}_{\pi, u}^{\mathbb{B}} v](s) \triangleq \begin{cases} r_{\pi}(s) + \gamma \cdot \sum_{s' \in \mathbb{B}} p_{\pi}(s' | s) \cdot v(s') \\ \quad + \gamma \cdot \sum_{s' \in \mathbb{B}^c} p_{\pi}(s' | s) \cdot u(s'), & \text{if } s \in \mathbb{B}, \\ u(s), & \text{otherwise;} \end{cases} \quad (4.5)$$

however, this involves a change of summation order and therefore requires either absolutely summability or non-negativity of the sum by the Fubini–Tonelli theorem. We use absolutely summability and therefore require that the following condition holds.

Condition 4.1 | salvage has a finite weighted supremum norm

The salvage function u has a finite w -weighted supremum norm, $u \in L^w(\mathbb{S})$.

See Proposition 2.3.
P. 34.

Remark 4.1

Under Condition 4.1, the sum $\sum_{s' \in \mathbb{B}} p_\pi(s' | s) \cdot u(s')$ can be split into sums over disjoint sets \mathbb{B}_i (finite or infinite) for any subset \mathbb{B} :

$$\sum_{s' \in \mathbb{B}} p_\pi(s' | s) \cdot u(s') = \sum_{i=0}^n \sum_{s' \in \mathbb{B}_i} p_\pi(s' | s) \cdot u(s'),$$

where $\bigsqcup_{i=0}^n \mathbb{B}_i = \mathbb{B}$, and $\mathbb{B} \subseteq \mathbb{S}$.

4.4 Policy Evaluation

Indeed, under Condition 4.1, the sum $\sum_{s' \in \mathbb{S}} p_\pi(s' | s) \cdot u(s')$ is absolutely summable, because

$$\begin{aligned} \sum_{s' \in \mathbb{S}} |p_\pi(s' | s) \cdot u(s')| &\leq \sum_{s' \in \mathbb{S}} p_\pi(s' | s) \cdot |u(s')| &> p_\pi(s' | s) \geq 0 \\ &\leq \sum_{s' \in \mathbb{S}} p_\pi(s' | s) \cdot w(s') \cdot \|u\|_w &> \text{by Remark 2.6} \\ &\leq \kappa \|u\|_w \cdot w(s) < \infty. &> \text{by (2.26)} \end{aligned}$$

By the Fubini–Tonelli theorem, if an infinite sum is absolutely summable, so is any of its sub-sums, and the summation order can be chosen arbitrary.

We can rearrange the terms in (4.5) even further by combining all of the summands that do not depend on the value function v . To do this, we define the following auxiliary functions.

Definition 4.4 | (u, \mathbb{B}) -truncated reward bonus

The (u, \mathbb{B}) -truncated reward bonus $b_{\pi, u}^{\mathbb{B}}$ under policy π is a function given by

$$b_{\pi, u}^{\mathbb{B}}(s) \triangleq \gamma \cdot \sum_{s' \in \mathbb{B}^c} p_\pi(s' | s) \cdot u(s'). \quad (4.6)$$

Definition 4.5 | untruncated reward bonus

The untruncated reward bonus $b_\pi^{\mathbb{B}}$ under policy π is given by

$$b_\pi^{\mathbb{B}}(s) \triangleq \gamma \cdot \sum_{s' \in \mathbb{B}^c} p_\pi(s' | s) \cdot v_\pi(s'). \quad (4.7)$$

Definition 4.6 | bonus-augmented reward

The $b_{\pi, u}^{\mathbb{B}}$ -augmented reward is a function given by

$$r_{\pi, u}^{\mathbb{B}}(s) \triangleq r_\pi(s) + b_{\pi, u}^{\mathbb{B}}(s).$$

Using this notation, the (u, \mathbb{B}) -truncated Bellman operator $\mathcal{L}_{\pi, u}^{\mathbb{B}}$ can be written as

$$[\mathcal{L}_{\pi, u}^{\mathbb{B}} v](s) = r_{\pi, u}^{\mathbb{B}}(s) + \gamma \cdot \sum_{s' \in \mathbb{B}} p_\pi(s' | s) \cdot v(s'), \text{ for each } s \in \mathbb{B}. \quad (4.8)$$

It is now equivalent to a Bellman operator of an MDP $\tilde{\mathfrak{M}}$ with a finite state space $\tilde{\mathbb{S}} \triangleq \mathbb{B}$ and rewards $\tilde{r}_\pi(s) \triangleq r_{\pi, u}^{\mathbb{B}}(s)$.

Unlike this smaller, finite-state problem $\check{\mathfrak{M}}$, we can still transition outside of the subspace \mathbb{B} in the original countably-infinite MDP \mathfrak{M} . It is thus important to distinguish between transitions within the subspace \mathbb{B} , and those that can lead outside of it. The probability $p_\pi^{j,\mathbb{B}}(s' | s)$ to transition from some state $s \in \mathbb{B}$ to another state $s' \in \mathbb{B}$ in j steps when following policy π and never leaving the subspace \mathbb{B} can be computed as

$$\begin{aligned} p_\pi^{0,\mathbb{B}}(s' | s) &\triangleq p_\pi^0(s' | s) = \delta_{s,s'}, \\ p_\pi^{j,\mathbb{B}}(s'' | s) &\triangleq \sum_{s' \in \mathbb{B}} p_\pi(s'' | s') \cdot p_\pi^{j-1,\mathbb{B}}(s' | s). \end{aligned}$$

4.4.2 Fixed Point of the Truncated Bellman Operator

While the resulting truncated Bellman operator $\mathcal{L}_{\pi,u}^\mathbb{B}$ can be used as an approximation to the exact Bellman operator \mathcal{L}_π , it is not immediately obvious that this new operator has a fixed point, like \mathcal{L}_π does, nor that this fixed point is unique. The following theorem shows, that this is indeed true.

Theorem 4.2 * fixed point of the truncated Bellman operator
Under Condition 2.5, the (u, \mathbb{B}) -truncated Bellman operator $\mathcal{L}_{\pi,u}^\mathbb{B}$ has a unique fixed point $v_{\pi,u}^\mathbb{B} \in L^w(\mathbb{S})$ for any stationary policy $\pi \in \mathbb{D}$, salvage function $u \in L^w(\mathbb{S})$, and truncation set $\mathbb{B} \subset \mathbb{S}$.

See p. 108. The proof of this theorem is presented in Section 4.7.1.

4.4.3 Additional Notation

In order to simplify further presentation, let us introduce the following notation.

First, let $\mathcal{I} : \mathbb{R}^\mathbb{S} \rightarrow \mathbb{R}^\mathbb{S}$ denote the identity operator

$$[\mathcal{I}y](s) = y(s) \quad \text{for all } s \in \mathbb{S}.$$

When the argument y is restricted to the space $L^w(\mathbb{S})$, $\mathcal{I}y \in L^w(\mathbb{S})$ trivially.

Using this operator, we can write $v_\pi = r_\pi + \gamma \cdot \mathcal{T}_\pi v_\pi$ as $r_\pi = (\mathcal{I} - \gamma \cdot \mathcal{T}_\pi)v_\pi$, or $v_\pi = (\mathcal{I} - \gamma \cdot \mathcal{T}_\pi)^{-1}r_\pi$. It is useful to have a notation for this inverse operator.

Definition 4.7 | value-producing operator

The *value-producing operator* $\mathcal{Q}_\pi : \mathbb{R}^\mathbb{S} \rightarrow \mathbb{R}^\mathbb{S}$ is the inverse of $\mathcal{I} - \gamma \cdot \mathcal{T}_\pi$:

$$\mathcal{Q}_\pi \triangleq (\mathcal{I} - \gamma \cdot \mathcal{T}_\pi)^{-1} = \sum_{i=0}^{\infty} (\gamma \cdot \mathcal{T}_\pi)^i.$$

Lemma 4.3 * the operator \mathcal{Q}_π maps $L^w(\mathbb{S})$ to itself

The value-producing operator \mathcal{Q}_π maps the space $L^w(\mathbb{S})$ of functions with finite w -weighted supremum norm to itself.

4.4 Policy
Evaluation

Proof. Proposition 2.17 states that $\|\mathcal{Q}_\pi r_\pi\|_w \leq \mu < \infty$ for the reward function $r_\pi \in L^w(\mathbb{S})$. The same argument holds for any function $y \in L^w(\mathbb{S})$, showing that $\|\mathcal{Q}_\pi y\|_w$ is finite. QED

Next, because we restrict the evaluation to a finite subset \mathbb{B} , we can use the following vector notation. For any subsets $\mathbb{B} \subseteq \mathbb{S}$ and $\mathbb{B}' \subseteq \mathbb{S}$, let $\mathbf{y}^\mathbb{B}$ and $\mathbf{T}_\pi^{\mathbb{B} \rightarrow \mathbb{B}'}$ denote a vector of all elements of a function restricted to the subspace \mathbb{B} and a transition matrix from \mathbb{B} to \mathbb{B}' :

$$\begin{aligned} \mathbf{y}^\mathbb{B} &\triangleq [y(s)]_{s \in \mathbb{B}} & \text{where } \mathbf{y} \in \{r_\pi, v_\pi, u, w, \text{ etc.}\}, \\ \mathbf{T}_\pi^{\mathbb{B} \rightarrow \mathbb{B}'} &\triangleq [p_\pi(s' | s)]_{s \in \mathbb{B}, s' \in \mathbb{B}'} \end{aligned}$$

When the transition operator given by a matrix $\mathbf{T}_\pi^{\mathbb{B} \rightarrow \mathbb{B}}$ acts from a subset \mathbb{B} to itself, we write simply $\mathbf{T}_\pi^\mathbb{B}$, and similarly for other linear operators.

As a consequence of Theorem 4.2, the (u, \mathbb{B}) -truncated value function $v_{\pi, u}^\mathbb{B}$ exists and is unique. Moreover, $v_{\pi, u}^\mathbb{B}(s) = u(s)$ for all $s \in \mathbb{B}^c$ (we can write this as $(\mathbf{v}_{\pi, u}^\mathbb{B})^{\mathbb{B}^c} = \mathbf{u}^{\mathbb{B}^c}$). For $s \in \mathbb{B}$, we write the vector $(\mathbf{v}_{\pi, u}^\mathbb{B})^\mathbb{B}$ of (u, \mathbb{B}) -truncated values $v_{\pi, u}^\mathbb{B}(s)$ simply as $\mathbf{v}_{\pi, u}^\mathbb{B}$. It can be found as follows:

$$\mathbf{v}_{\pi, u}^\mathbb{B} = \mathbf{r}_\pi^\mathbb{B} + \mathbf{b}_{\pi, u}^\mathbb{B} + \gamma \cdot \mathbf{T}_\pi^\mathbb{B} \mathbf{v}_{\pi, u}^\mathbb{B}. \quad (4.9)$$

By moving the second summand to the left-hand side and multiplying both sides by $(\mathbf{I}^\mathbb{B} - \gamma \cdot \mathbf{T}_\pi^\mathbb{B})^{-1}$ we obtain the following exact formula for the (u, \mathbb{B}) -truncated values:

$$\mathbf{v}_{\pi, u}^\mathbb{B} = \mathbf{Q}_\pi^\mathbb{B} (\mathbf{r}_\pi^\mathbb{B} + \mathbf{b}_{\pi, u}^\mathbb{B}) = \mathbf{Q}_\pi^\mathbb{B} \mathbf{r}_{\pi, u}^\mathbb{B}, \quad \text{where} \quad (4.10)$$

$$\mathbf{Q}_\pi^\mathbb{B} \triangleq (\mathbf{I}^\mathbb{B} - \gamma \cdot \mathbf{T}_\pi^\mathbb{B})^{-1}. \quad (4.11)$$

This is possible because the spectral radius of matrix $\gamma \cdot \mathbf{T}_\pi^\mathbb{B}$ is less than one. Therefore the matrix $\mathbf{Q}_\pi^\mathbb{B}$ exists and is finite [Puterman, 1994, Corollary C.4]. Assuming that the bonus vector $\mathbf{b}_{\gamma, \pi}^\mathbb{B}$ is known, the (u, \mathbb{B}) -truncated values $\mathbf{v}_{\pi, u}^\mathbb{B}$ on the truncation set \mathbb{B} can be computed in finite time.

The spectral radius of the transition matrix $\mathbf{T}_\pi^\mathbb{B}$ is equal to one, because it is a stochastic matrix.

As a result, (4.10) is a more compact form of

$$v_{\pi, u}^\mathbb{B}(s) = \begin{cases} \sum_{i=0}^{\infty} \gamma^i \cdot \sum_{s' \in \mathbb{B}} p_\pi^{i, \mathbb{B}}(s' | s) \cdot r_{\pi, u}^\mathbb{B}(s'), & s \in \mathbb{B}, \\ u(s), & s \in \mathbb{B}^c, \end{cases}$$

following the notation of Theorem 4.2. It can also be derived from (4.29) directly by using $v_{\pi,u}^{\mathbb{B}} = \mathcal{L}_{\pi,u}^{\mathbb{B}} v_{\pi,u}^{\mathbb{B}}$. In the remainder of this chapter we will use vector notation in this way. We would like to stress that all of the results can be derived element-wise as well, so vector notation is just a shorthand used for clarity of presentation.

Remark 4.2

In vector-matrix form, Remark 4.1 can be expressed as

$$\mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{B}'} \mathbf{u}^{\mathbb{B}'} = \sum_{i=0}^n \mathbf{T}_{\pi}^{\mathbb{B}_i \rightarrow \mathbb{B}'} \mathbf{u}^{\mathbb{B}'} \quad \text{where} \quad \bigcap_{i=0}^{\infty} \mathbb{B}_i = \mathbb{B}.$$

Similarly, given a finite set of functions $u_i \in L^w(\mathbb{S})$, $0 \leq i \leq n < \infty$, we can see that $u = \sum_{i=0}^n u_i$ belongs to $L^w(\mathbb{S})$ as well, and therefore

$$\mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{B}'} \mathbf{y}^{\mathbb{B}'} = \sum_{i=0}^n \mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{B}'} \mathbf{y}_i^{\mathbb{B}'} \quad (4.12)$$

In particular, $\mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{B}'} (\mathbf{y}_0^{\mathbb{B}'} \pm \mathbf{u}_1^{\mathbb{B}'}) = \mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{B}'} \mathbf{y}_0^{\mathbb{B}'} \pm \mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{B}'} \mathbf{y}_1^{\mathbb{B}'}$.

Under Condition 2.5, rewards r_{π} , values v_{π} , salvages u and truncated values $v_{\pi,u}^{\mathbb{B}}$ all belong to $L^w(\mathbb{S})$. Remark 4.2 tell us that multiplication of $\mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{B}'}$ by either $\mathbf{r}_{\pi}^{\mathbb{B}'}$, $\mathbf{v}_{\pi}^{\mathbb{B}'}$, $\mathbf{u}^{\mathbb{B}'}$, or $\mathbf{v}_{\pi,u}^{\mathbb{B}'}$ can be block-partitioned and is left-distributive. This observation will be used in the following proofs without being explicitly mentioned.

4.4.4 Truncation Errors

As (u, \mathbb{B}) -truncation is an approximation, the truncated values $v_{\pi,u}^{\mathbb{B}}$ differ from the true values $v_{\pi}^{\mathbb{B}}$. In this section, we examine the difference between these two values. We begin with the following definition.

Definition 4.8 | truncation error

Given a (u, \mathbb{B}) -truncation, its *error* $e_{\pi,u}^{\mathbb{B}}$ under a policy $\pi \in \mathbb{D}$ is the difference between the true values $v_{\pi}^{\mathbb{B}}$ and the approximated values $v_{\pi,u}^{\mathbb{B}}$:

$$e_{\pi,u}^{\mathbb{B}}(s) \triangleq v_{\pi,u}^{\mathbb{B}}(s) - v_{\pi}^{\mathbb{B}}(s). \quad (4.13)$$

Theorem 4.4 * truncation error

If Condition 2.5 holds, within the truncation set \mathbb{B} , the values $\mathbf{v}_{\pi,u}^{\mathbb{B}}$ of a (u, \mathbb{B}) -truncation differ from the exact values $\mathbf{v}_{\pi}^{\mathbb{B}}$ by

$$\mathbf{e}_{\pi,u}^{\mathbb{B}} = \mathbf{Q}_{\pi}^{\mathbb{B}} (\mathbf{b}_{\pi,u}^{\mathbb{B}} - \mathbf{b}_{\pi}^{\mathbb{B}}) \quad (4.14)$$

for any truncation set $\mathbb{B} \subseteq \mathbb{S}$ and salvage function $u \in L^w(\mathbb{S})$. The matrix $\mathbf{Q}_{\pi}^{\mathbb{B}}$ is defined by (4.11).

Moreover, for any choice of salvage functions $u_{\pm} \in L^w(\mathbb{S})$ such that $u_- \leq v_{\pi} \leq u_+$, the exact values v_{π} are bounded by

$$v_{\pi, u_-}^{\mathbb{B}} \leq v_{\pi} \leq v_{\pi, u_+}^{\mathbb{B}}.$$

4.4 Policy Evaluation

Proof. Values $\mathbf{v}_{\pi}^{\mathbb{B}}$ can be written as

$$\mathbf{v}_{\pi}^{\mathbb{B}} = \mathbf{r}_{\pi}^{\mathbb{B}} + \gamma \cdot \mathbf{T}_{\pi}^{\mathbb{B} \rightarrow \mathbb{S}} \mathbf{v}_{\pi} = \mathbf{r}_{\pi}^{\mathbb{B}} + \gamma \cdot \mathbf{T}_{\pi}^{\mathbb{B}} \mathbf{v}_{\pi}^{\mathbb{B}} + \mathbf{b}_{\pi}^{\mathbb{B}}. \quad \triangleright \text{by (4.7)}$$

The error $\mathbf{e}_{\pi, u}^{\mathbb{B}}$ is then equal to

$$\begin{aligned} \mathbf{e}_{\pi, u}^{\mathbb{B}} &= \mathbf{r}_{\pi}^{\mathbb{B}} + \gamma \cdot \mathbf{T}_{\pi}^{\mathbb{B}} \mathbf{v}_{\pi, u}^{\mathbb{B}} + \mathbf{b}_{\pi, u}^{\mathbb{B}} - \mathbf{r}_{\pi}^{\mathbb{B}} - \gamma \cdot \mathbf{T}_{\pi}^{\mathbb{B}} \mathbf{v}_{\pi}^{\mathbb{B}} - \mathbf{b}_{\pi}^{\mathbb{B}} \\ &= \gamma \cdot \mathbf{T}_{\pi}^{\mathbb{B}} (\mathbf{v}_{\pi, u}^{\mathbb{B}} - \mathbf{v}_{\pi}^{\mathbb{B}}) + (\mathbf{b}_{\pi, u}^{\mathbb{B}} - \mathbf{b}_{\pi}^{\mathbb{B}}) \\ &= \gamma \cdot \mathbf{T}_{\pi}^{\mathbb{B}} \mathbf{e}_{\pi, u}^{\mathbb{B}} + (\mathbf{b}_{\pi, u}^{\mathbb{B}} - \mathbf{b}_{\pi}^{\mathbb{B}}). \end{aligned} \quad \begin{aligned} &\triangleright \text{by (4.9) and (4.13)} \\ &\triangleright \text{reordering, see (4.12)} \\ &\triangleright \text{again by (4.13)} \end{aligned}$$

By moving the first summand to the left-hand side and multiplying both sides by $\mathbf{Q}_{\pi}^{\mathbb{B}}$ we obtain (4.14).

If $u \geq v_{\pi}$, then $\mathbf{u}^{\mathbb{B}^c} - \mathbf{v}_{\pi}^{\mathbb{B}^c} \geq 0$ and consequently $(\mathbf{e}_{\pi, u}^{\mathbb{B}})^{\mathbb{B}^c} \geq 0$ (as $(\mathbf{v}_{\pi, u}^{\mathbb{B}})^{\mathbb{B}^c} = \mathbf{u}^{\mathbb{B}^c}$ by definition). At the same time, by comparing Definitions 4.4 and 4.5, $\mathbf{b}_{\pi, u}^{\mathbb{B}} \geq \mathbf{b}_{\pi}^{\mathbb{B}}$ trivially. Therefore, $\mathbf{e}_{\pi, u}^{\mathbb{B}} \geq 0$ as a product of a non-negative matrix and a non-negative vector. Thus $u_+ \geq v_{\pi}$ implies $v_{\pi, u_+}^{\mathbb{B}} \geq v_{\pi}$. The case of u_- follows *mutatis mutandis*. QED


Corollary 4.5 * truncated values are monotone in salvages

For any two salvage functions $u', u'' \in L^w(\mathbb{S})$, if $u' \geq u''$, then $v_{\pi, u'}^{\mathbb{B}} \geq v_{\pi, u''}^{\mathbb{B}}$.

Proof. We write $\mathbf{v}_{\pi, u'}^{\mathbb{B}} - \mathbf{v}_{\pi, u''}^{\mathbb{B}} = (\mathbf{v}_{\pi, u'}^{\mathbb{B}} - \mathbf{v}_{\pi}^{\mathbb{B}}) - (\mathbf{v}_{\pi, u''}^{\mathbb{B}} - \mathbf{v}_{\pi}^{\mathbb{B}})$ and use Theorem 4.4 to obtain

$$\mathbf{v}_{\pi, u'}^{\mathbb{B}} - \mathbf{v}_{\pi, u''}^{\mathbb{B}} = \mathbf{Q}_{\pi}^{\mathbb{B}} (\mathbf{b}_{\pi, u'}^{\mathbb{B}} - \mathbf{b}_{\pi, u''}^{\mathbb{B}}).$$

By definition of the truncated bonus function, it is a linear combination of values of the salvage function with positive coefficients. Therefore, if $u' \geq u''$, then $b_{\pi, u'}^{\mathbb{B}} \geq b_{\pi, u''}^{\mathbb{B}}$ and $v_{\pi, u'}^{\mathbb{B}}(s) \geq v_{\pi, u''}^{\mathbb{B}}(s)$ if $s \in \mathbb{B}$. For $s \in \mathbb{B}^c$, $v_{\pi, u'}^{\mathbb{B}}(s) - v_{\pi, u''}^{\mathbb{B}}(s) = \mathbf{u}'(s) - \mathbf{u}''(s) \geq 0$; as a result, $v_{\pi, u'}^{\mathbb{B}}(s) \geq v_{\pi, u''}^{\mathbb{B}}(s)$ for any $s \in \mathbb{S}$. QED

 Theorem 4.4 allows us to estimate how good an approximation provided by (u, \mathbb{B}) -truncation is. Moreover, as the salvage u provides an estimate of values over the complement \mathbb{B}^c , we can say that if $u \leq v_{\pi}$ (or $u \geq v_{\pi}$) it “underestimates” (or “overestimates”) v_{π} . By Corollary 4.5, so do all the truncation values $v_{\pi, u}^{\mathbb{B}}$. Therefore, we can obtain bounds on possible ranges of true values v_{π} .

When bounds from a truncation set \mathbb{B} are too loose to reason about optimality of the policy, we want to be able to improve them by choosing a different truncation set \mathbb{B} . We also want the approximations obtained this way to converge to the true values v_π . The following theorem shows that it is possible, but requires an additional condition on the weight function.

Condition 4.2 | vanishing effect of the weight function


There exists a monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^\infty$ of states such that for any stationary deterministic policy $\pi \in \mathbb{D}$

$$\lim_{k \rightarrow \infty} \sup_{s \in \mathbb{S}_\infty} \left(\sum_{s' \in \mathbb{S}_k^c} p_\pi(s' | s) \cdot w(s') \right) = 0.$$

Theorem 4.6 * convergence of the truncated values

Under Conditions 2.5 and 4.2, the sequence of absolute errors $(e_{\pi,u}^{\mathbb{S}_k}(s))_{k=0}^\infty$ of (u, \mathbb{S}_k) -truncations converges to zero for any salvage function $u \in L^w(\mathbb{S})$, stationary deterministic policy $\pi \in \mathbb{D}$, and state $s \in \mathbb{S}_\infty$ in the limiting set of the monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^\infty$. As a result, the sequence of (u, \mathbb{S}_k) -truncated values $(v_{\pi,u}^{\mathbb{S}_k}(s))_{k=0}^\infty$ converges to the exact values v_π over \mathbb{S}_∞ :

$$\lim_{k \rightarrow \infty} v_{\pi}^{u, \mathbb{S}_k}(s) = v_\pi(s) \quad \text{for all } s \in \mathbb{S}_\infty.$$

See p. 110  The proof of Theorem 4.6 is presented in Section 4.7.2.

Theorem 4.6 generalizes Lemma 3 of Lee et al. [2017], where $\mathbb{S} = \mathbb{N}$, $\mathbb{B}_k = \{1, 2, \dots, k\}$, and $u = 0$.

Theorem 4.6 establishes that in countably-infinite MDPs policy evaluation can be done approximately by considering a truncation and that the approximation quality improves as the truncation size grows.

4.5 POLICY IMPROVEMENT

In the previous section, we showed that policies can be evaluated arbitrary close in the limiting set of the truncation sequence. We now design a policy improvement procedure that uses such evaluations to optimize the decisions prescribed by a policy.

4.5.1 Pivoting and advantages

We begin with a procedure known as policy pivoting. It can be used to create a new policy by changing a single decision.

Definition 4.9 | policy pivoting

Given a policy $\pi \in \mathbb{D}$, and a *pivot pair* $(s, a) \in \mathbb{X}$, *pivoting* of the policy π at the state-action pair (s, a) is a procedure of obtaining a new policy $\pi' \in \mathbb{D}$ that is identical to the policy π everywhere except for the state s , where it uses the action a instead of $\pi(s)$. In other words, the new policy is produced by the *pivoting operator* $\mathcal{P}_{s,a} : \mathbb{D} \rightarrow \mathbb{D}$ (that is, $\pi' = \mathcal{P}_{s,a}\pi$) defined as

$$[\mathcal{P}_{s,a}\pi](s') \triangleq \begin{cases} a, & \text{if } s' = s, \\ \pi(s'), & \text{otherwise.} \end{cases}$$

4.5 Policy improvement

- Whether a pivoting improves or worsens the policy, can be established by computing advantages of different actions.

Definition 4.10 | advantage over policy

Advantage $\eta_\pi(s, a)$ of an action a over a policy π for state s is defined as

$$\begin{aligned} \eta_\pi(s, a) &\triangleq r(s, a) + \gamma \cdot \sum_{s' \in \mathbb{S}} p(s' | s, a) \cdot v_\pi(s') - v_\pi(s) \\ &= r(s, a) + \gamma \cdot \sum_{s' \in \mathbb{S} \setminus \{s\}} p(s' | s, a) \cdot v_\pi(s') \\ &\quad - (1 - \gamma \cdot p(s | s, a)) \cdot v_\pi(s). \end{aligned} \quad (4.15)$$

In this notation, the advantages $\eta(s, a)$ of Definition 3.1 (p. 70) are equal to advantages over an optimal policy π_\star : $\eta(s, a) = \eta_{\pi_\star}(s, a)$.

- Any policy π induces occupancies z_π that are feasible to the primal program CI-P. Advantages represent negative slacks of the complementary dual solution v_π to the dual program CI-D. They can be used in both policy improvement and action elimination as follows.

For a reminder of the dual formulation of the problem, see Theorem 2.22, p. 57.

Lemma 4.7 * salvage effect on policy pivots

Given a policy $\pi \in \mathbb{D}$ and a pivot pair (s, a) , let π' denote the new policy after pivoting π at (s, a) , $\pi' = \mathcal{P}_{s,a}\pi$.

- If $\eta_\pi(s, a) < 0$, then $v_{\pi'}(s') \leq v_\pi(s')$ for all states $s' \in \mathbb{S}$ and $v_{\pi'}(s) < v_\pi(s)$, so the pivoting worsens the policy π .
- If $\eta_\pi(s, a) > 0$, then $v_{\pi'}(s') \geq v_\pi(s')$ for all states $s' \in \mathbb{S}$ and $v_{\pi'}(s) > v_\pi(s)$, so the pivoting improves the policy π .
- If $\eta_\pi(s'', a'') \leq 0$ for all state-action pairs $(s'', a'') \in \mathbb{X}$, then the policy π cannot be improved and is optimal.

Proof. The values $v_{\pi'}$ of the policy π' are equal to

$$v_{\pi'} = r_{\pi'} + \gamma \cdot \mathcal{T}_{\pi'} v_{\pi'} = r_{\pi'} + \gamma \cdot \mathcal{T}_{\pi'} (v_{\pi'} - v_\pi) + \gamma \cdot \mathcal{T}_{\pi'} v_\pi - v_\pi + v_\pi.$$

Also see the proof of Lee et al. [ibid., Proposition 6].

Therefore,

4 The Countably-Infinite Model

$$(\mathcal{J} + \gamma \cdot \mathcal{T}_{\pi'}) (v_{\pi'} - v_{\pi}) = r_{\pi'} + \gamma \cdot \mathcal{T}_{\pi'} v_{\pi} - v_{\pi}, \quad \text{and} \\ v_{\pi'} - v_{\pi} = \mathcal{Q}_{\pi'} (r_{\pi'} + \gamma \cdot \mathcal{T}_{\pi'} v_{\pi} - v_{\pi}).$$

The function $f = r_{\pi'} + \gamma \cdot \mathcal{T}_{\pi'} v_{\pi} - v_{\pi}$ is equal to zero everywhere except for $f(s)$, which is equal to $\eta_{\pi}(s, a)$.

If $\eta_{\pi}(s, a) > 0$, then for any function y

$$[\mathcal{Q}_{\pi'} y](s) = \sum_{i=0}^{\infty} \gamma^i \cdot [\mathcal{T}_{\pi'}^i y](s) = y(s) + \sum_{i=1}^{\infty} \gamma^i \cdot [\mathcal{T}_{\pi'}^i y](s) \geq y(s),$$

and therefore $v_{\pi'} - v_{\pi} \geq r_{\pi'} + \gamma \cdot \mathcal{T}_{\pi'} v_{\pi} - v_{\pi} \geq 0$ and $v_{\pi'}(s) - v_{\pi}(s) \geq \eta_{\pi}(s, a) > 0$. Similarly, if $\eta_{\pi}(s, a) < 0$ then $v_{\pi'} - v_{\pi} \leq 0$ and $v_{\pi'}(s) - v_{\pi}(s) < 0$.

See Definition 3 of Lee et al. [2017] for details.

The last statement follows trivially from the complementary slackness conditions. QED

Lemma 4.7 serves as a basis for both policy improvement and action elimination steps if the exact advantages are known. Unfortunately, just like in the case of values, they cannot be evaluated finitely and we have to rely on approximations.

4.5.2 Advantage Approximation

We now introduce the approximate advantages and show that they approach the exact advantages $\eta_{\pi}(s, a)$ as the truncation set grows. We call such approximations upper and a lower (\mathbb{U}, \mathbb{B}) -approximate advantages $\eta_{\pi, \pm}^{\mathbb{U}, \mathbb{B}}(s, a)$, and show that they converge to the exact advantage $\eta_{\pi}(s, a)$ from above and below respectively.

Definition 4.11 | (u, \mathbb{B}) -approximate advantage

For any salvage function $u \in L^w(\mathbb{S})$ and truncation set $\mathbb{B} \subset \mathbb{S}$, the (u, \mathbb{B}) -approximate advantage of action a over policy π for state s is a function $\eta_{\pi, u}^{\mathbb{B}} : \mathbb{X} \rightarrow \mathbb{R}$ defined as

$$\eta_{\pi, u}^{\mathbb{B}} \triangleq r + \gamma \cdot \mathcal{T} v_{\pi, u}^{\mathbb{B}} - v_{\pi, u}^{\mathbb{B}}. \quad (4.16)$$

Definition 4.12 | upper and lower (\mathbb{U}, \mathbb{B}) -approximate advantages

Given a subspace of functions $\mathbb{U} \subset L^w(\mathbb{S})$ called a *salvage space* and a subspace of states $\mathbb{B} \subset \mathbb{S}$, the *upper* and *lower* (\mathbb{U}, \mathbb{B}) -approximate advantages $\eta_{\pi, +}^{\mathbb{U}, \mathbb{B}}(s, a)$ and $\eta_{\pi, -}^{\mathbb{U}, \mathbb{B}}(s, a)$ of action $a \in A_p(s)$ over policy $\pi \in \mathbb{D}$ for state $s \in \mathbb{S}$ are defined as

$$\eta_{\pi, +}^{\mathbb{U}, \mathbb{B}}(s, a) \triangleq \max_{u \in \mathbb{U}} \eta_{\pi, u}^{\mathbb{B}}(s, a), \quad \eta_{\pi, -}^{\mathbb{U}, \mathbb{B}}(s, a) \triangleq \min_{u \in \mathbb{U}} \eta_{\pi, u}^{\mathbb{B}}(s, a). \quad (4.17)$$

🌀 The upper and lower (\mathbb{U}, \mathbb{B}) -approximate advantages may be ill-defined, because the attainability of the extrema in (4.17) depends on the salvage space \mathbb{U} . For now, let us assume that the approximate advantages $\eta_{\pi, \pm}^{\mathbb{U}, \mathbb{B}}(s, a)$ are indeed well-defined. We will establish this fact later for a particular choice of the salvage space \mathbb{U} .

4.5 Policy improvement

Theorem 4.8 * convergence of the approximate advantages

Under Conditions 2.5 and 4.2, if the upper and lower $(\mathbb{U}, \mathbb{S}_k)$ -approximate advantages $\eta_{\pi, \pm}^{\mathbb{U}, \mathbb{S}_k}(s, a)$ are well-defined, then for any stationary deterministic policy $\pi \in \mathbb{D}$, state $s \in \mathbb{S}_\infty$ in the limiting set of the monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^\infty$, and permitted action $a \in A_p(s)$, they converge to the true advantages $\eta_\pi(s, a)$ from above and below respectively:

$$\eta_{\pi, -}^{\mathbb{U}, \mathbb{S}_k}(s, a) \uparrow \eta_\pi(s, a) \quad \text{and} \quad \eta_{\pi, +}^{\mathbb{U}, \mathbb{S}_k}(s, a) \downarrow \eta_\pi(s, a).$$

🌀 The proof of this theorem is presented in Section 4.7.4.

Even if we assume that the extrema in (4.17) are attained, it is still not immediately clear how these optimization problems can be solved, because the (u, \mathbb{B}) -approximate advantage function $\eta_{\pi, u}^{\mathbb{B}}$ of 4.16 is expressed in terms of the truncated value function $v_{\pi, u}^{\mathbb{B}}$. To address this challenge, we now show the approximate advantage $\eta_{\pi, u}^{\mathbb{B}}$ can be written as a function of the salvage u .

First, we rewrite the approximate advantage function $\eta_{\pi, u}^{\mathbb{B}}$ as

$$\begin{aligned} \eta_{\pi, u}^{\mathbb{B}}(s, a) &= r(s, a) + \gamma \cdot \sum_{s' \in \mathbb{S}} p(s' | s, a) \cdot v_{\pi, u}^{\mathbb{B}}(s') - v_{\pi, u}^{\mathbb{B}}(s) \\ &= r(s, a) + b_{\pi, u}^{\mathbb{B}}(s) \\ &\quad + \left(\gamma \cdot \sum_{s' \in \mathbb{B}} p(s' | s, a) \cdot v_{\pi, u}^{\mathbb{B}}(s') - v_{\pi, u}^{\mathbb{B}}(s) \right) \\ &= r(s, a) + [\mathcal{N}b_{\pi, u}^{\mathbb{B}}](s, a) + [\mathcal{R}^{\mathbb{B}}v_{\pi, u}^{\mathbb{B}}](s, a), \end{aligned} \quad (4.18)$$

▶ \mathcal{N} is the extension operator that changes the argument of a function from s to (s, a) , see (2.24).

where the operator $\mathcal{R}^{\mathbb{B}} : \mathbb{R}^{\mathbb{X}} \rightarrow \mathbb{R}^{\mathbb{S}}$ shows how much a function y is expected to change after a one-step transition into the subspace \mathbb{B} ; it is defined as

$$[\mathcal{R}^{\mathbb{B}}y](s, a) \triangleq \gamma \cdot \sum_{s' \in \mathbb{B}} p(s' | s, a) \cdot y(s') - y(s).$$

Since our goal is to evaluate the advantages for some states $s \in \mathbb{B}$, we can assemble them into a vector that includes these states only. To do so, we define the set $\mathbb{X}_{\mathbb{B}} \subseteq \mathbb{X}$ of admissible controls in the subset $\mathbb{B} \subseteq \mathbb{S}$ of states as

$$\mathbb{X}_{\mathbb{B}} \triangleq \{(s, a) \in \mathbb{X} \mid s \in \mathbb{B}\},$$

and write the operators $\mathcal{R}^{\mathbb{B}}$ and \mathcal{N} as $|\mathbb{X}_{\mathbb{B}}| \times |\mathbb{B}|$ matrices

4 The Countably-Infinite Model

$$\mathbf{R}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \triangleq [\gamma \cdot p(s' | s, a) - \delta_{s,s'}]_{(s,a) \in \mathbb{X}_{\mathbb{B}}, s' \in \mathbb{B}}, \quad (4.19)$$

$$\mathbf{N}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \triangleq [\delta_{s,s'}]_{(s,a) \in \mathbb{X}_{\mathbb{B}}, s' \in \mathbb{B}}. \quad (4.20)$$

Finally, using this notation, the (u, \mathbb{B}) -approximate advantages $\eta_{\pi,u}^{\mathbb{B}}(s, a)$ for $(s, a) \in \mathbb{X}_{\mathbb{B}}$ can be written in matrix form as

$$\begin{aligned} \text{by (4.18)} \quad \eta_{\pi,u}^{\mathbb{X}_{\mathbb{B}}} &\triangleq (\eta_{\pi,u}^{\mathbb{B}})^{\mathbb{X}_{\mathbb{B}}} = \mathbf{r}^{\mathbb{X}_{\mathbb{B}}} + \mathbf{N}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{b}_{\pi,u}^{\mathbb{B}} + \mathbf{R}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{v}_{\pi,u}^{\mathbb{B}} \\ \text{by (4.10)} \quad &= \mathbf{r}^{\mathbb{X}_{\mathbb{B}}} + \mathbf{N}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{b}_{\pi,u}^{\mathbb{B}} + \mathbf{R}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{Q}_{\pi}^{\mathbb{B}} (\mathbf{r}_{\pi}^{\mathbb{B}} + \mathbf{b}_{\pi,u}^{\mathbb{B}}) \\ &= \mathbf{r}^{\mathbb{X}_{\mathbb{B}}} + \mathbf{R}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{Q}_{\pi}^{\mathbb{B}} \mathbf{r}_{\pi}^{\mathbb{B}} + (\mathbf{N}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} + \mathbf{R}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{Q}_{\pi}^{\mathbb{B}}) \mathbf{b}_{\pi,u}^{\mathbb{B}} \\ &= \check{\mathbf{r}}_{\pi}^{\mathbb{X}_{\mathbb{B}}} + \mathbf{L}_{\pi}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{b}_{\pi,u}^{\mathbb{B}}, \end{aligned} \quad (4.21)$$


where the augmented reward vector $\check{\mathbf{r}}_{\pi}^{\mathbb{X}_{\mathbb{B}}}$ and the auxiliary matrices $\mathbf{L}_{\pi}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}}$ and $\mathbf{K}_{\pi}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}}$ are defined as follows:

$$\check{\mathbf{r}}_{\pi}^{\mathbb{X}_{\mathbb{B}}} \triangleq \mathbf{r}^{\mathbb{X}_{\mathbb{B}}} + \mathbf{K}_{\pi}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{r}_{\pi}^{\mathbb{B}}, \quad (4.22)$$

$$\mathbf{L}_{\pi}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \triangleq \mathbf{N}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} + \mathbf{K}_{\pi}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}}, \quad \text{and} \quad (4.23)$$

$$\mathbf{K}_{\pi}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \triangleq \mathbf{R}^{\mathbb{X}_{\mathbb{B}} \rightarrow \mathbb{B}} \mathbf{Q}_{\pi}^{\mathbb{B}}. \quad (4.24)$$

Note that all of these matrices and vectors are finite-dimensional, justifying the various changes of the operations order in (4.21).

In finite dimensions, the space \mathbb{U}_{\pm} is compact, because it is bounded and complete; therefore, by the extreme-value theorem Aliprantis and Border [2006, Theorem 2.43], the extrema are attained. In the countably-infinite case, boundedness and closeness are not sufficient to establish compactness. 

The (u, \mathbb{B}) -approximate advantages $\eta_{\pi,u}^{\mathbb{B}}(s, a)$ of (4.21) are expressed in a way that does not require to compute the values v_{π} of the policy $\pi \in \mathbb{D}$. Instead, they depend on the salvage function u directly via the (u, \mathbb{B}) -truncated bonus function $b_{\pi,u}^{\mathbb{B}}$ of Definition 4.4. The bonus function is an affine transformation of the salvage function u presented in (4.6), and therefore the bonus is linear in the salvage and the objective function of (4.16) is linear.

To ensure that the optimization problems (4.16) are linear, we consider the following salvage space, which can be expressed with linear constraints.

Definition 4.13 | (u_{-}, u_{+}) -bounded salvage space

Given salvage functions $u_{\pm} \in L^w(\mathbb{S})$ such that $u_{-} \leq u_{+}$, the (u_{-}, u_{+}) -bounded salvage space $\mathbb{U}_{\pm} \subset L^w(\mathbb{S})$ is the set of all functions in $L^w(\mathbb{S})$ bounded by u_{-} and u_{+} from below and above, that is,

$$\mathbb{U}_{\pm} \triangleq \{u \in L^w(\mathbb{S}) \mid u_{-} \leq u \leq u_{+}\}. \quad (4.25)$$

Combining the results of this section, we solve the problems (4.17) and show that the approximate advantages $\eta_{\pi,\pm}^{\mathbb{U}_{\pm},\mathbb{B}}(s, a)$ are well-defined and can be computed as follows.

Theorem 4.9 * the approximate advantage formula

The upper and lower $(\mathbb{U}_\pm, \mathbb{B})$ -approximate advantages $\eta_{\pi, \pm}^{\mathbb{U}_\pm, \mathbb{B}}(s, a)$ of Definition 4.12 are well-defined and equal to

$$\eta_{\pi, \pm}^{\mathbb{U}_\pm, \mathbb{B}}(s, a) = \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a) + \gamma \cdot \sum_{s' \in \mathbb{B}^c} \check{p}^+(s' | s, a) \cdot u_\pm(s') - \gamma \cdot \sum_{s' \in \mathbb{B}^c} \check{p}^-(s' | s, a) \cdot u_\mp(s'), \quad (4.26)$$

for each stationary policy $\pi \in \mathbb{D}$, bounding functions $u_\pm \in L^w(\mathbb{S})$, truncation set $\mathbb{B} \subset \mathbb{S}$, state $s \in \mathbb{B}$, and action $a \in A_p(s)$. The space \mathbb{U}_\pm and the augmented reward $\check{r}_\pi^{\mathbb{X}_\mathbb{B}}$ are defined by (4.25) and (4.22); the function \check{p} is given by

$$\check{p}(s'' | s, a) \triangleq \sum_{s' \in \mathbb{B}} l_\pi(s' | s, a) \cdot p_\pi(s'' | s'), \quad (4.27)$$

where $l_\pi(s' | s, a)$ are the elements of the matrix $\mathbf{L}_\pi^{\mathbb{X}_\mathbb{B} \rightarrow \mathbb{B}}$ of (4.23):

$$\mathbf{L}_\pi^{\mathbb{X}_\mathbb{B} \rightarrow \mathbb{B}} = [l_\pi(s' | s, a)]_{(s, a) \in \mathbb{X}_\mathbb{B}, s' \in \mathbb{B}}.$$

🌀 The proof of this theorem is presented in Section 4.7.4.

Strictly speaking, the coefficients $\check{p}(s'' | s, a)$ of Theorem 4.9 are not transition probabilities. However, we denote these quasi-probabilities as $\check{p}(s'' | s, a)$ because the formula (4.26) bears resemblance to the truncated Bellman operator equation (4.5)

The quasi-probabilities $\check{p}(s'' | s, a)$ of (4.27) are not necessarily computable finitely, as they involve an infinite sum over the completion \mathbb{B}^c . Therefore, we impose an additional assumption to ensure that they are computable.

Assumption 4.7 | computability of the quasi-probabilities

For the monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^\infty$ of Condition 4.2, the sums in (4.26) are computable in finite time.

🌀 In particular, Assumption 4.7 holds in problems with limited reachability because there are only finitely many reachable states in the complement \mathbb{S}_k^c of any truncation set \mathbb{S}_k . Depending on the particular form of transition kernel and the bounding functions u_\pm , these sum may be computable in other problems as well. For example, in problems with non-negative rewards, a zero lower bound $u_- = 0$ means that the second sum in (4.26) is equal to zero.

Summarizing the results of this section, we now present the policy improvement and action elimination rules.

4.5 Policy improvement

y^+ and y^- denote the positive and negative parts of a function y : $y^+ = \max\{y, 0\}$ and $y^- = \max\{-y, 0\}$.

The negative values in $\check{p}(s'' | s, a)$ arise because the elements of the matrix $\mathbf{R}^{\mathbb{X}_\mathbb{B} \rightarrow \mathbb{B}}$ can be negative, see (4.26).

For the truncated Bellman operator, see p. 94

Corollary 4.10 * policy improvement and action elimination

Given a monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^{\infty}$ of Condition 4.2, and bounding functions $u_{\pm} \in L^w(\mathbb{S})$ such that $u_- \leq v_{\pi} \leq u_+$, consider a stationary deterministic policy $\pi \in \mathbb{D}$.

- If for some state $s \in \mathbb{S}_k$ and action $a \in A_p(s)$ the upper approximate advantage $\eta_{\pi,+}^{\mathbb{U},\mathbb{S}_k}(s,a)$ is negative, then pivoting the policy π at (s,a) worsens it.
- If for some state $s \in \mathbb{S}_k$ and action $a \in A_p(s)$ the lower approximate advantage $\eta_{\pi,-}^{\mathbb{U},\mathbb{S}_k}(s,a)$ is positive, then pivoting the policy π at (s,a) improves it.
- Moreover, if the policy π is not optimal, there exist $k \in \mathbb{N}_0$, such that for some $s \in \mathbb{S}_k$ and $a \in A_p(s)$ the lower approximate advantage $\eta_{\pi,-}^{\mathbb{U},\mathbb{S}_k}(s,a)$ is positive.

Proof. Corollary 4.10 is a direct consequence of Lemma 4.7 and Theorem 4.9. QED

Corollary 4.10 serves as the basis for policy improvement and action elimination steps. If $\eta_{\pi,+}^{\mathbb{U},\mathbb{B}_k}(s,a) < 0$, then action a is worse than $\pi(s)$ for state s and can be eliminated. If $\eta_{\pi,-}^{\mathbb{U},\mathbb{B}_k}(s,a) > 0$, then $\pi(s)$ should be changed to a ; this is policy improvement. The last statement of the lemma guarantees that an improvement can always be found in this way as long as the policy π is not optimal (i.e., as long as it can be improved).

4.6 THE ALGORITHM

We now introduce our algorithm **ASPIRE**—approximate salvage-based policy iteration with repeated elimination of actions.

The idea behind **ASPIRE** is simple: starting with an arbitrary policy, the algorithm improves it until no better policy can be found for any state in the support of the initial distribution. In order to do so, it computes the policy values (policy evaluation), then finds the upper and lower advantages of alternative actions.

If the lower advantage approximation is positive, then so is the true advantage, and pivoting the policy on the given state-action pair increases its values (improvement).

If for some state-action pair the upper advantage approximation is negative, so is the advantage itself, and this action cannot be optimal for the given state (action elimination). The algorithm

Data: a countably-infinite MDP \mathfrak{M}_∞ ,
a monotone-increasing sequence of salvage spaces
 $(\mathbb{S}_k)_{k=0}^\infty$, and value bounding functions u_\pm

Result: initial-distribution optimal policy π

```

1 initialize the iteration counter  $i \leftarrow 0$ ;
2 initialize the truncation counter  $k \leftarrow 0$ ;
3 initialize the policy  $\pi_0 \in \mathbb{D}$  arbitrarily;
4 initialize the feasible control set  $\mathbb{W} \leftarrow \mathbb{X}_{\mathbb{S}_0}$ ;
5 repeat
6   loop
7     foreach  $(s, a) \in \mathbb{W}$  do
8       compute the upper and lower approximate
          advantages  $\eta_{\pi_i, \pm}^{\mathbb{U}, \mathbb{S}_k}(s, a)$  given by (4.26);
9       if  $\eta_{\pi_i, +}^{\mathbb{U}, \mathbb{S}_k}(s, a) < 0$  then
10        eliminate the pair  $(s, a)$ ,  $\mathbb{W} \leftarrow \mathbb{W} \setminus \{(s, a)\}$ ;
11    pick  $(s, a) \in \arg \max_{(s', a') \in \mathbb{W}} \{\eta_{\pi_i, -}^{\mathbb{U}, \mathbb{S}_k}(s', a')\}$ ;
12    if  $\eta_{\pi_i, -}^{\mathbb{U}, \mathbb{S}_k}(s, a) > 0$  then
13      pivot the policy  $\pi_{i+1} \leftarrow \mathcal{P}_{s, a} \pi_i$ ;
14      break;
15    else
16      enlarge the truncation set  $k \leftarrow k + 1$ ;
17      add new state-action pairs to the feasible
          control set  $\mathbb{W} \leftarrow \mathbb{W} \cup \mathbb{X}_{\mathbb{S}_k}$ ;
18   $i \leftarrow i + 1$ ;
19 until  $\mathbb{W}$  has one action for every state  $s \in \text{supp } \alpha$ ;
```

Figure 4.1:
ASPIRE—approximate
salvage-based policy
iteration with repeated
elimination (of actions).

► approximation

► repeated elimination

► policy iteration

terminates when all of the alternative actions are eliminated for the initial states.

Under Assumption 4.6, the initial states have unique optimal actions. Since all other actions are suboptimal, their advantages are negative. Because upper advantages converge to the true advantages, they eventually become negative and the alternative actions are eliminated.

Note that if Assumption 4.6 does not hold, ASPIRE can still be used for policy improvement. However, a different terminal condition should be used in this case, such as a time-based constraint.

4.7 PROOFS

In this section, we present the proofs of several theorems from the previous sections. For the reader's convenience, we restate these theorems before proving them.

4.7.1 Proof of Theorem 4.2

Theorem 4.2 * fixed point of the truncated Bellman operator
Under Condition 2.5, the (u, \mathbb{B}) -truncated Bellman operator $\mathcal{L}_{\pi, u}^{\mathbb{B}}$ has a unique fixed point $v_{\pi, u}^{\mathbb{B}} \in L^w(\mathbb{S})$ for any stationary policy $\pi \in \mathbb{D}$, salvage function $u \in L^w(\mathbb{S})$, and truncation set $\mathbb{B} \subset \mathbb{S}$.

See p. 54 *Proof.* By Proposition 2.19, we need to show that the (u, \mathbb{B}) -truncated Bellman operator $\mathcal{L}_{\pi, u}^{\mathbb{B}}$ is indeed a v -stage contraction, and that it has a finite Lipschitz constant.

To prove the first statement, we show that the operator $\mathcal{L}_{\pi, u}^{\mathbb{B}}$ maps the space $L^w(\mathbb{S})$ to itself, that is, $\mathcal{L}_{\pi, u}^{\mathbb{B}} v \in L^w(\mathbb{S})$ for any $v \in L^w(\mathbb{S})$, and that

$$\|(\mathcal{L}_{\pi, u}^{\mathbb{B}})^v v' - (\mathcal{L}_{\pi, u}^{\mathbb{B}})^v v''\|_w < \|v' - v''\|_w$$

for any $v', v'' \in L^w(\mathbb{S})$. Then, we find the Lipschitz constant of the truncated Bellman operator $\mathcal{L}_{\pi, u}^{\mathbb{B}}$ and verify that it is finite, concluding the proof.

Outside of the truncation set \mathbb{B} , when $s \in \mathbb{B}^c$,

by definition and (2.32), \triangleleft $|\mathcal{L}_{\pi, u}^{\mathbb{B}} v(s)| = |u(s)| \leq \|u\|_w \cdot w(s)$
see p. 54

For any state $s \in \mathbb{B}$ in the truncation set \mathbb{B} ,

by definition \triangleleft $|\mathcal{L}_{\pi, u}^{\mathbb{B}} v(s)| = |r_{\pi}(s) + \gamma \cdot \sum_{s' \in \mathbb{B}} p_{\pi}(s' | s) \cdot v(s')|$

$$\begin{aligned}
& + \gamma \cdot \sum_{s' \in \mathbb{B}^c} p_\pi(s' | s) \cdot u(s') \\
& \leq |r_\pi(s)| + \gamma \cdot \sum_{s' \in \mathbb{B}} p_\pi(s' | s) \cdot |v(s')| \quad \triangleright \text{using the triangle inequality} \\
& \quad + \gamma \cdot \sum_{s' \in \mathbb{B}^c} p_\pi(s' | s) \cdot |u(s')| \\
& \leq w(s) + \max\{\|v\|_w, \|u\|_w\} \cdot \gamma \cdot \sum_{s' \in \mathbb{S}} p_\pi(s' | s) \cdot w(s') \quad \triangleright \text{by (2.25) and (2.32)} \\
& \leq (1 + \kappa \cdot \max\{\|v\|_w, \|u\|_w\}) \cdot w(s). \quad \triangleright \text{by (2.26)}
\end{aligned}$$

Therefore, it follows from (2.32) that

$$\|\mathcal{L}_{\pi,u}^{\mathbb{B}} v\|_w \leq \max\{1 + \kappa \cdot \max\{\|v\|_w, \|u\|_w\}, \|u\|_w\} < \infty.$$

which proves that $\mathcal{L}_{\pi,u}^{\mathbb{B}} v \in L^w(\mathbb{S})$ for any $v \in L^w(\mathbb{S})$. By induction, $(\mathcal{L}_{\pi,u}^{\mathbb{B}})^n v \in L^w(\mathbb{S})$ for any $n \in \mathbb{N}$.

Next, we show that the operator $\mathcal{L}_{\pi,u}^{\mathbb{B}}$ is a v -stage contraction mapping on $L^w(\mathbb{S})$. Note that

$$p_\pi^{j,\mathbb{B}}(s' | s) \leq p_\pi^j(s' | s) \quad \text{for any } j \geq 0 \text{ and } s, s' \in \mathbb{B}, \quad (4.28)$$

because the former is the probability to transition from a state s to a state s' in j steps while never leaving the subspace \mathbb{B} , and the latter allows stepping out of the subspace \mathbb{B} and returning.

By chain-substituting $(\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v = \mathcal{L}_{\pi,u}^{\mathbb{B}}((\mathcal{L}_{\pi,u}^{\mathbb{B}})^{v-1} v)$, from (4.8) we obtain

$$\begin{aligned}
[(\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v](s) &= \begin{cases} \sum_{j=0}^{v-1} \gamma^j \cdot \sum_{s' \in \mathbb{B}} p_\pi^{j,\mathbb{B}}(s' | s) \cdot r_{\pi,u}^{\mathbb{B}}(s') \\ \quad + \gamma^v \cdot \sum_{s' \in \mathbb{B}} p_\pi^{v,\mathbb{B}}(s' | s) \cdot v(s'), & \text{if } s \in \mathbb{B}, \\ u(s), & \text{otherwise.} \end{cases} \quad (4.29)
\end{aligned}$$

Let v' and v'' be two functions in $L^w(\mathbb{S})$. Then $(\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v'(s) - (\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v''(s) = u(s) - u(s) = 0$ for any $s \in \mathbb{B}^c$. Note that the first summand in the first case of (4.29) does not depend on v . Thus, for all states $s \in \mathbb{B}$

$$\begin{aligned}
|(\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v'(s) - (\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v''(s)| &= \gamma^v \cdot \sum_{s' \in \mathbb{B}} p_\pi^{v,\mathbb{B}}(s' | s) \cdot |v'(s') - v''(s')| \quad \triangleright \text{by (4.29)} \\
&\leq \gamma^v \cdot \sum_{s' \in \mathbb{B}} p_\pi^v(s' | s) \cdot |v'(s') - v''(s')| \quad \triangleright \text{by (4.28)} \\
&\leq \gamma^v \cdot \sum_{s' \in \mathbb{S}} p_\pi^v(s' | s) \cdot |v'(s') - v''(s')| \quad \triangleright \text{by adding positive summands over } \mathbb{B}^c \\
&\leq \gamma^v \cdot \sum_{s' \in \mathbb{S}} p_\pi^v(s' | s) \cdot w(s') \cdot \|v' - v''\|_w \quad \triangleright \text{by (2.32), see p. 54}
\end{aligned}$$

by (2.27), see p. 52 ◀

$$\leq \lambda \|v' - v''\|_w \cdot w(s).$$

Therefore, by applying (2.32) we obtain

$$\|(\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v' - (\mathcal{L}_{\pi,u}^{\mathbb{B}})^v v''\|_w \leq \lambda \|v' - v''\|_w. \quad (4.30)$$

Because $\lambda < 1$ by its definition, the (u, \mathbb{B}) -truncated Bellman operator $\mathcal{L}_{\pi,u}^{\mathbb{B}}$ is a v -stage contraction with respect to the w -weighted norm $\|\cdot\|_w$.

Following the same steps that we used to derive (4.30) but with $\mathcal{L}_{\pi,u}^{\mathbb{B}}$ instead of $(\mathcal{L}_{\pi,u}^{\mathbb{B}})^v$, we can show that

$$\|\mathcal{L}_{\pi,u}^{\mathbb{B}} v' - \mathcal{L}_{\pi,u}^{\mathbb{B}} v''\|_w \leq \kappa \|v' - v''\|_w,$$

where κ is the one-stage expansion coefficient defined by (2.26). Therefore the Lipschitz constant of $\mathcal{L}_{\pi,u}^{\mathbb{B}}$ is equal to the coefficient κ , which is finite by its definition. QED

4.7.2 Proof of Theorem 4.6

Theorem 4.6 * convergence of the truncated values

Under Conditions 2.5 and 4.2, the sequence of absolute errors $(e_{\pi,u}^{\mathbb{S}_k}(s))_{k=0}^{\infty}$ of (u, \mathbb{S}_k) -truncations converges to zero for any salvage function $u \in L^w(\mathbb{S})$, stationary deterministic policy $\pi \in \mathbb{D}$, and state $s \in \mathbb{S}_{\infty}$ in the limiting set of the monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^{\infty}$. As a result, the sequence of (u, \mathbb{S}_k) -truncated values $(v_{\pi,u}^{\mathbb{S}_k}(s))_{k=0}^{\infty}$ converges to the exact values v_{π} over \mathbb{S}_{∞} :

$$\lim_{k \rightarrow \infty} v_{\pi}^{u, \mathbb{S}_k}(s) = v_{\pi}(s) \quad \text{for all } s \in \mathbb{S}_{\infty}.$$

Proof. Note that $w_0 = \inf_{s \in \mathbb{S}} w(s)$ is positive by definition of the weight function w . Then for any $s \in \mathbb{B} \subseteq \mathbb{Y} \subseteq \mathbb{S}$

$$\begin{aligned} |e_{\pi,u}^{\mathbb{B}}(s)| &= \sum_{i=0}^{\infty} \sum_{s' \in \mathbb{B}} \gamma^i \cdot p_{\pi}^i(s' | s) \cdot \sum_{s'' \in \mathbb{B}^c} \gamma \cdot p_{\pi}(s'' | s') \cdot |u(s'') - v_{\pi}(s'')| \\ &\stackrel{\text{by adding positive summands over } \mathbb{Y} \setminus \mathbb{B}}{\leq} \sum_{i=0}^{\infty} \sum_{s' \in \mathbb{Y}} \gamma^i \cdot p_{\pi}^i(s' | s) \cdot \sum_{s'' \in \mathbb{B}^c} \gamma \cdot p_{\pi}(s'' | s') \cdot |u(s'') - v_{\pi}(s'')| \\ &\stackrel{\text{by (2.32), see p. 54}}{\leq} \sum_{i=0}^{\infty} \sum_{s' \in \mathbb{Y}} \gamma^i \cdot p_{\pi}^i(s' | s) \cdot \sum_{s'' \in \mathbb{B}^c} \gamma \cdot p_{\pi}(s'' | s') \cdot w(s'') \\ &\quad \cdot \|u - v_{\pi}\|_w \\ &\stackrel{\text{by the triangle inequality}}{\leq} \sum_{i=0}^{\infty} \sum_{s' \in \mathbb{Y}} \gamma^i \cdot p_{\pi}^i(s' | s) \cdot \sum_{s'' \in \mathbb{B}^c} \gamma \cdot p_{\pi}(s'' | s') \cdot w(s'') \\ &\quad \cdot (\|u\|_w + \|v_{\pi}\|_w) \\ &\stackrel{\text{by (2.28), see p. 52}}{\leq} \mu \cdot w(s) \cdot \sup_{s' \in \mathbb{Y}} \left(w(s')^{-1} \cdot \sum_{s'' \in \mathbb{B}^c} \gamma \cdot p_{\pi}(s'' | s') \cdot w(s'') \right) \end{aligned}$$

$$\begin{aligned}
& \cdot (\mu + \|u\|_w) \\
& \leq c_u \cdot w(s) \cdot \sup_{s' \in \mathbb{Y}} \left(\sum_{s'' \in \mathbb{B}^c} p_\pi(s'' | s') \cdot w(s'') \right), \quad (4.31) \quad \triangleright \text{combining the constants} \\
& \quad \text{into } c_u
\end{aligned}$$

where c_u is a finite constant for any $s \in \mathbb{S}$ given by

$$c_u \triangleq \gamma \mu w_0 \cdot (\mu + \|u\|_w). \quad (4.32)$$

Because the state space is discrete, any state $s \in \text{supp } \alpha$ will belong to all \mathbb{S}_n , $n \geq N$ starting with some index N . Then for any $s \in \mathbb{S}_\infty$

$$\begin{aligned}
\lim_{i \rightarrow \infty} |e_{\pi, u}^{\mathbb{S}_i}(s)| &= \lim_{k \rightarrow \infty} |e_{\pi, u}^{\mathbb{S}_{N+k}}(s)| \\
&\leq c_u \cdot w(s) \cdot \lim_{k \rightarrow \infty} \sup_{s' \in \mathbb{S}_\infty} \left(\sum_{s'' \in \mathbb{S}_{N+k}^c} p_\pi(s'' | s') w(s'') \right) = 0.
\end{aligned}$$

Therefore, $\lim_{i \rightarrow \infty} v_{\pi, u}^{\mathbb{S}_i}(s) = v_\pi(s)$ for any $s \in \mathbb{S}_\infty$. QED

4.7.3 Proof of Theorem 4.8

Theorem 4.8 states that the (\mathbb{U}, \mathbb{B}) -approximate advantage functions $\eta_{\pi, \pm}^{\mathbb{U}, \mathbb{B}}$ converge to the true advantage function η_π . To prove it, we first consider an auxiliary approximation, to which we refer as the (u', u'', \mathbb{B}) -approximate advantage function $\eta_{\pi, u', u''}^{\mathbb{B}}$.

Definition 4.14 | (u', u'', \mathbb{B}) -approximate advantage

For any functions $u', u'' \in L^w(\mathbb{S})$ and truncation set $\mathbb{B} \subset \mathbb{S}$, a (u', u'', \mathbb{B}) -approximate advantage of action a over policy π for state s is defined as

$$\begin{aligned}
\eta_{\pi, u', u''}^{\mathbb{B}}(s, a) &\triangleq r(s, a) + \gamma \cdot \sum_{s' \in \mathbb{S} \setminus \{s\}} p(s' | s, a) \cdot v_{\pi, u'}^{\mathbb{B}}(s') \\
&\quad - (1 - \gamma \cdot p(s | s, a)) \cdot v_{\pi, u''}^{\mathbb{B}}(s). \quad (4.33)
\end{aligned}$$

As the (u', \mathbb{B}) -truncated value function $v_{\pi, u'}^{\mathbb{B}}$ belongs to $L^w(\mathbb{S})$ by Theorem 4.2, the (u', u'', \mathbb{B}) -approximate advantage function $\eta_{\pi, u', u''}^{\mathbb{B}}$ is well-defined and belongs to $L^w(\mathbb{S})$ as well.

Next, we proof the following bounds on the exact advantage function $\eta_\pi(s, a)$.

Lemma 4.11

For any truncation set $\mathbb{B} \subset \mathbb{S}$ and salvage functions $u_\pm \in L^w(\mathbb{S})$ such that $u_- \leq v_\pi \leq u_+$, the advantages $\eta_\pi(s, a)$ are bounded by

$$\eta_{\pi, u_-, u_+}^{\mathbb{B}}(s, a) \leq \eta_\pi(s, a) \leq \eta_{\pi, u_+, u_-}^{\mathbb{B}}(s, a).$$

Proof. By definition of $\eta_\pi(s, a)$ and $\eta_{\pi, u_-, u_+}^{\mathbb{B}}(s, a)$,

$$\begin{aligned} \text{by (4.13), (4.15) and (4.33)} \quad \eta_\pi(s, a) - \eta_{\pi, u_-, u_+}^{\mathbb{B}}(s, a) &= \gamma \cdot \sum_{s' \in \mathbb{S} \setminus \{s\}} p(s' | s, a) \cdot (-e_{\pi, u_-}^{\mathbb{B}}(s')) \\ &\quad + (1 - \gamma \cdot p(s | s, a)) \cdot e_{\pi, u_+}^{\mathbb{B}}(s). \end{aligned} \quad (4.34)$$

Note that (4.34) involves a change of summation order, which is possible because the errors are absolutely bounded by (4.31) and the sum in (4.34) is finite by (2.26).

For any probability $p(s | s, a)$ the multiplier $1 - \gamma \cdot p(s | s, a)$ is positive. By Theorem 4.4, the differences $v_\pi(s') - v_{\pi, u_-}^{\mathbb{B}}(s')$ and $e_{\pi, u_+}^{\mathbb{B}}(s) = v_{\pi, u_+}^{\mathbb{B}}(s) - v_\pi(s)$ are non-negative. Therefore, the right-hand side is non-negative, and $\eta_{\pi, u_-, u_+}^{\mathbb{B}}(s, a) \leq \eta_\pi(s, a)$. The proof for $\eta_{\pi, u_+, u_-}^{\mathbb{B}}(s, a)$ holds *mutatis mutandis*. QED

Similarly to the approximate values, approximate advantages converge to the exact advantages as the truncation set \mathbb{B} grows and approaches the state space \mathbb{S} . More formally, this statement can be formulated as follows.

Lemma 4.12

Given a function w of Condition 2.5 and a monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^\infty$, the sequence $(\eta_{\pi, u', u''}^{\mathbb{S}_k}(s, a))_{k=0}^\infty$ of (u', u'', \mathbb{S}_k) -approximate advantages converges to the exact advantage $\eta_\pi(s, a)$ for any salvage functions $u', u'' \in L^w(\mathbb{S})$, stationary deterministic policy $\pi \in \mathbb{D}$, and state s in the limiting set of the sequence, $s \in \mathbb{S}_\infty$, and permitted action $a \in A_p(s)$.

Proof. Similarly to the proof of Theorem 4.6, we observe that every state $s \in \mathbb{S}_\infty$ will belong to all $\mathbb{S}_n, n \geq N$ starting with some index N . Then for any $s \in \mathbb{S}_\infty$ the difference $e_\eta(s, a) \triangleq |\eta_\pi(s, a) - \eta_{\pi, u', u''}^{\mathbb{S}_{N+k}}(s, a)|$ is bounded by

$$\begin{aligned} \text{by (4.34) and the triangle inequality} \quad e_\eta(s, a) &\leq \gamma \cdot \sum_{s' \in \mathbb{S} \setminus \{s\}} p(s' | s, a) \cdot |e_{\pi, u'}^{\mathbb{S}_{N+k}}(s')| \\ &\quad + (1 - \gamma \cdot p(s | s, a)) \cdot |e_{\pi, u''}^{\mathbb{S}_{N+k}}(s)| \\ 1 - \gamma \cdot p(s | s, a) &\leq 1 \quad \leq \gamma \cdot \sum_{s' \in \mathbb{S} \setminus \{s\}} p(s' | s, a) \cdot |e_{\pi, u'}^{\mathbb{S}_{N+k}}(s')| + |e_{\pi, u''}^{\mathbb{S}_{N+k}}(s)| \\ \text{by (4.31)} \quad &\leq \gamma c_{u'} \cdot \sum_{s''' \in \mathbb{S} \setminus \{s\}} p(s''' | s, a) \cdot w(s''') \\ &\quad \cdot \sup_{s' \in \mathbb{S}_\infty} \left(\sum_{s'' \in \mathbb{S}_{N+k}^c} p_\pi(s'' | s') \cdot w(s'') \right) \\ &\quad + c_{u''} \cdot w(s) \cdot \sup_{s' \in \mathbb{S}_\infty} \left(\sum_{s'' \in \mathbb{S}_{N+k}^c} p_\pi(s'' | s') \cdot w(s'') \right) \end{aligned}$$

$$\leq c_{u',u''} \cdot w(s) \cdot \sup_{s' \in \mathbb{S}_\infty} \left(\sum_{s'' \in \mathbb{S}_{N+k}^c} p_\pi(s'' | s') w(s'') \right), \quad \triangleright \text{ by (2.26), see p. 52}$$

where the constant $c_{u',u''}$ is defined as

$$c_{u',u''} \triangleq \kappa c_{u'} + c_{u''}$$

and the constants c_u are given by (4.32). The multiplier $c_{u',u''} \cdot w(s)$ is a finite constant for any s ; thus, under Condition 4.2,

$$\lim_{i \rightarrow \infty} \eta_{\pi,u',u''}^{\mathbb{B}_i}(s, a) = \eta_\pi(s, a) \quad \text{for any } (s, a) \in \mathbb{X}_{\mathbb{S}_\infty}. \quad \text{QED}$$

Finally, we show how the approximate advantages $\eta_{\pi,u',u''}^{\mathbb{B}}(s, a)$ can be used to bound the upper and lower (\mathbb{U}, \mathbb{B}) -approximate advantages $\eta_{\pi,\pm}^{\mathbb{U},\mathbb{B}}(s, a)$.

Lemma 4.13

For any salvage space $\mathbb{U} \subseteq \mathbb{U}_\pm$ such that $v_\pi \in \mathbb{U}$, the upper and lower (\mathbb{U}, \mathbb{B}) -approximate advantages $\eta_{\pi,\pm}^{\mathbb{U},\mathbb{B}}(s, a)$ are bounded by

$$\eta_{\pi,u_-,u_+}^{\mathbb{B}}(s, a) \leq \eta_{\pi,-}^{\mathbb{U},\mathbb{B}}(s, a) \leq \eta_\pi(s, a) \leq \eta_{\pi,+}^{\mathbb{U},\mathbb{B}}(s, a) \leq \eta_{\pi,u_+,u_-}^{\mathbb{B}}(s, a).$$

Proof. The fact that $\eta_{\pi,u_-,u_+}^{\mathbb{B}}(s, a) \leq \eta_\pi(s, a) \leq \eta_{\pi,u_+,u_-}^{\mathbb{B}}(s, a)$ follows from Lemma 4.11. Consider the lower approximate advantage $\eta_{\pi,-}^{\mathbb{U},\mathbb{B}}(s, a)$. By (4.33) and (4.25), $\eta_{\pi,u_-,u_+}^{\mathbb{B}}(s, a) \leq \eta_{\pi,-}^{\mathbb{U},\mathbb{B}}(s, a)$. Similarly, $\eta_{\pi,-}^{\mathbb{U},\mathbb{B}}(s, a) \leq \eta_\pi(s, a)$ follows immediately from (4.17) and the fact that $v_\pi \in \mathbb{U}$ by the definition of \mathbb{U} . The case of the upper (\mathbb{U}, \mathbb{B}) -approximate advantage $\eta_{\pi,+}^{\mathbb{U},\mathbb{B}}(s, a)$ holds *mutatis mutandis*. QED

We are now ready to prove Theorem 4.8.

Theorem 4.8 * convergence of the approximate advantages

Under Conditions 2.5 and 4.2, if the upper and lower $(\mathbb{U}, \mathbb{S}_k)$ -approximate advantages $\eta_{\pi,\pm}^{\mathbb{U},\mathbb{S}_k}(s, a)$ are well-defined, then for any stationary deterministic policy $\pi \in \mathbb{D}$, state $s \in \mathbb{S}_\infty$ in the limiting set of the monotone-increasing sequence $(\mathbb{S}_k)_{k=0}^\infty$, and permitted action $a \in A_p(s)$, they converge to the true advantages $\eta_\pi(s, a)$ from above and below respectively:

$$\eta_{\pi,-}^{\mathbb{U},\mathbb{S}_k}(s, a) \uparrow \eta_\pi(s, a) \quad \text{and} \quad \eta_{\pi,+}^{\mathbb{U},\mathbb{S}_k}(s, a) \downarrow \eta_\pi(s, a).$$

Proof. By Lemma 4.13, the lower approximate advantage $\eta_{\pi,-}^{\mathbb{U},\mathbb{B}}(s, a)$ is bounded from below by $\eta_{\pi,u_-,u_+}^{\mathbb{B}}(s, a)$ and from above by $\eta_\pi(s, a)$. By Lemma 4.12, the lower bound converges to the upper bound for every state $s \in \mathbb{S}_\infty$ in the limiting set; by the squeeze theorem

so does the lower approximate advantage $\eta_{\pi,-}^{\mathbb{U},\mathbb{B}}(s,a)$, and the convergence is strictly from below. The same argument applies to $\eta_{\pi,+}^{\mathbb{U},\mathbb{B}}(s,a)$ *mutatis mutandis*. QED

4.7.4 Proof of Theorem 4.9

To prove Theorem 4.9, we need to find analytical solutions to the optimization problems 4.17. We do so by employing their dual formulations. Unlike finite linear programs, neither weak nor strong duality is guaranteed to hold in countably-infinite linear programs, so we introduce the following lemma, which provides sufficient conditions for strong duality in this case.

Lemma 4.14 * staircase programs are strongly dual

See Romeijn and
R. L. Smith [1998,
Theorem 3.7].

Consider the following countably-infinite linear program, known as a lower-staircase countably-infinite linear program, and its upper-staircase dual:

$$\begin{aligned} J_P(\mathbf{y}) = \min_{\mathbf{y}} \quad & \sum_{i=1}^{\infty} \mathbf{c}_i^{\top} \mathbf{y}_i & (\text{LS-P}) \\ \text{s.t.} \quad & \mathbf{A}_{i,i-1} \mathbf{y}_{i-1} + \mathbf{A}_{i,i} \mathbf{y}_i \geq \mathbf{b}_i, \\ & \mathbf{y}_i \geq \mathbf{0}, \\ & \mathbf{y} \in \mathbb{Y} \end{aligned}$$

$$\begin{aligned} \text{and } J_D(\mathbf{x}) = \max_{\mathbf{x}} \quad & \sum_{i=1}^{\infty} \mathbf{b}_i^{\top} \mathbf{x}_i & (\text{US-D}) \\ \text{s.t.} \quad & \mathbf{A}_{i,i}^{\top} \mathbf{x}_i + \mathbf{A}_{i+1,i}^{\top} \mathbf{x}_{i+1} \leq \mathbf{c}_i, \\ & \mathbf{x}_i \geq \mathbf{0}, \\ & \mathbf{x} \in \mathbb{X}, \end{aligned}$$

where \mathbb{Y} and \mathbb{X} are the sets of all possible values of optimization variables \mathbf{y}_i and \mathbf{x}_i , $i \in \mathbb{N}$ for which the objective function is well-defined and finite.

If for all $\mathbf{x} \in \mathbb{X}$ and $\mathbf{y} \in \mathbb{Y}$

$$\liminf_{k \rightarrow \infty} \mathbf{x}_{k+1}^{\top} \mathbf{A}_{k+1,k} \mathbf{y}_k \geq 0, \quad (4.35)$$

then for any pair $\mathbf{x}' \in \mathbb{X}$ and $\mathbf{y}' \in \mathbb{Y}$ the following two statements are equivalent:

- \mathbf{x}' is primal-feasible, \mathbf{y}' is dual-feasible, and they satisfy the complementary slackness

$$(\mathbf{A}_{i,i-1} \mathbf{y}'_{i-1} + \mathbf{A}_{i,i} \mathbf{y}'_i - \mathbf{b}_i)^{\top} \mathbf{x}'_i = 0 \quad \text{and}$$

$$(\mathbf{c}_i - \mathbf{A}_{i,i}^\top \mathbf{x}'_i - \mathbf{A}_{i+1,i}^\top \mathbf{x}'_{i+1})^\top \mathbf{y}'_i = 0 \quad \text{for all } i \in \mathbb{N},$$

and the transversality condition

$$\liminf_{k \rightarrow \infty} (\mathbf{x}'_{k+1})^\top \mathbf{A}_{k+1,k} \mathbf{y}'_k = 0;$$

- \mathbf{x}' and \mathbf{y}' are optimal solutions of the primal and dual programs respectively, and the programs are strongly dual, that is, $J_p(\mathbf{y}) = J_D(\mathbf{x})$.

Theorem 4.9 * the approximate advantage formula

The upper and lower $(\mathbb{U}_\pm, \mathbb{B})$ -approximate advantages $\eta_{\pi,\pm}^{\mathbb{U}_\pm, \mathbb{B}}(s, a)$ of Definition 4.12 are well-defined and equal to

$$\begin{aligned} \eta_{\pi,\pm}^{\mathbb{U}_\pm, \mathbb{B}}(s, a) &= \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a) + \gamma \cdot \sum_{s' \in \mathbb{B}^c} \check{p}^+(s' | s, a) \cdot u_\pm(s') \\ &\quad - \gamma \cdot \sum_{s' \in \mathbb{B}^c} \check{p}^-(s' | s, a) \cdot u_\mp(s'), \end{aligned} \quad (4.26)$$

for each stationary policy $\pi \in \mathbb{D}$, bounding functions $u_\pm \in L^w(\mathbb{S})$, truncation set $\mathbb{B} \subset \mathbb{S}$, state $s \in \mathbb{B}$, and action $a \in A_p(s)$. The space \mathbb{U}_\pm and the augmented reward $\check{r}_\pi^{\mathbb{X}_\mathbb{B}}$ are defined by (4.25) and (4.22); the function \check{p} is given by

$$\check{p}(s'' | s, a) \triangleq \sum_{s' \in \mathbb{B}} l_\pi(s' | s, a) \cdot p_\pi(s'' | s'), \quad (4.27)$$

where $l_\pi(s' | s, a)$ are the elements of the matrix $\mathbf{L}_\pi^{\mathbb{X}_\mathbb{B} \rightarrow \mathbb{B}}$ of (4.23):

$$\mathbf{L}_\pi^{\mathbb{X}_\mathbb{B} \rightarrow \mathbb{B}} = [l_\pi(s' | s, a)]_{(s,a) \in \mathbb{X}_\mathbb{B}, s' \in \mathbb{B}}.$$

Proof. We prove the theorem for the lower $(\mathbb{U}_\pm, \mathbb{B})$ -approximate advantage $\eta_{\pi,-}^{\mathbb{U}_\pm, \mathbb{B}}(s, a)$ only. The proof for the upper $(\mathbb{U}_\pm, \mathbb{B})$ -approximate advantage $\eta_{\pi,+}^{\mathbb{U}_\pm, \mathbb{B}}(s, a)$ follows *mutatis mutandis*.

First, by (4.21) and (4.25) the optimization problem (4.17) that defines $\eta_{\pi,-}^{\mathbb{U}_\pm, \mathbb{B}}(s, a)$ becomes

$$\begin{aligned} \min_{u \in L^w(\mathbb{S})} \quad & \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a) + \sum_{s' \in \mathbb{B}} l_\pi(s' | s, a) \cdot \gamma \cdot \sum_{s'' \in \mathbb{B}^c} p_\pi(s'' | s') \cdot u(s'') \\ \text{s.t.} \quad & u_- \leq u \leq u_+. \end{aligned}$$

The augmented reward $\check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a)$ is independent of the optimization variable u and can be removed from the program without affecting the solution. We can also move the discounting factor outside of the summation and remove it as a positive constant.

The inner infinite sum is absolutely summable

$$\sum_{s'' \in \mathbb{B}^c} p_\pi(s'' | s') \cdot |u(s'')| \leq \sum_{s'' \in \mathbb{S}} p_\pi(s'' | s') \cdot |u(s'')|$$

by (2.32), see p. 54 ◀

$$\leq \sum_{s'' \in \mathbb{S}} p_\pi(s'' | s') \cdot w(s'') \cdot \|u\|_w$$

by (2.26), see p. 52 ◀

$$\leq \kappa \|u\|_w \cdot w(s') < \infty. \quad (4.36)$$

Because $\mathbf{L}_\pi^{\mathbb{X}_\mathbb{B} \rightarrow \mathbb{B}}$ is a finite-dimensional matrix with finite elements by its definition, (4.23), the outer sum is a finite sum of absolutely summable sums, and therefore it is absolutely summable as well.

Next, we exchange the summation order to write the objective function as

$$\begin{aligned} & \sum_{s' \in \mathbb{B}} l_\pi(s' | s, a) \cdot \sum_{s'' \in \mathbb{B}^c} p_\pi(s'' | s') \cdot u(s'') \\ &= \sum_{s'' \in \mathbb{B}^c} \left(\sum_{s' \in \mathbb{B}} l_\pi(s' | s, a) \cdot p_\pi(s'' | s') \cdot u(s'') \right) \\ &= \sum_{s'' \in \mathbb{B}^c} \check{p}(s'' | s, a) \cdot u(s''). \end{aligned} \quad (4.37)$$

by (4.27) ◀

The optimization problem (4.17) is then equivalent to

$$\begin{aligned} \frac{\eta_{\pi, -}^{\mathbb{U}_{\pm, \mathbb{B}}}(s, a) - \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a)}{\gamma} &= \min_{u \in L^w(\mathbb{S})} \sum_{s' \in \mathbb{B}^c} \check{p}(s' | s, a) \cdot u(s') \\ \text{s.t.} \quad & u_- \leq u \leq u_+. \end{aligned}$$

After the change of variable to $y \triangleq u - u_-$, the optimization problem becomes

$$\begin{aligned} J_P(y) &= \min_{y \in L^w(\mathbb{S})} \sum_{s' \in \mathbb{B}^c} \check{p}(s' | s, a) \cdot y(s') \\ \text{s.t.} \quad & -y \geq u_- - u_+, \\ & y \geq 0. \end{aligned} \quad (\text{P})$$

with its objective function equal to

$$J_P(y) = \frac{\eta_{\pi, -}^{\mathbb{U}_{\pm, \mathbb{B}}}(s, a) - \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a)}{\gamma} - \sum_{s' \in \mathbb{B}^c} \check{p}(s' | s, a) \cdot u_-(s'),$$

and therefore

$$\eta_{\pi, -}^{\mathbb{U}_{\pm, \mathbb{B}}}(s, a) = \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a) + \gamma \cdot \left(J_P + \sum_{s' \in \mathbb{B}^c} \check{p}(s' | s, a) \cdot u_-(s') \right). \quad (4.38)$$

Again, there is a change of summation order but it is possible because absolute summability can be established by applying the same argument used in (4.36) to the lower bound u_- instead of the function u .

Note that y_f given by

$$y_f(s') \triangleq \begin{cases} 0, & \check{p}(s' | s, a) \geq 0, \\ u_+(s') - u_-(s'), & \text{otherwise} \end{cases}$$

is feasible to the primal problem (that is, it satisfies both constraints).

The dual program is

4.7 Proofs

$$J_D(y) = \max_{x \in L_*^{\mathcal{N}w}(\mathbb{X})} \sum_{s' \in \mathbb{B}^C} (u_-(s') - u_+(s')) \cdot x(s') \quad (\text{D})$$

$$\text{s.t.} \quad -x \leq \check{p}(\cdot | s, a),$$

$$x \geq 0.$$

The function $x_f(s') \triangleq \check{p}^-(s' | s, a)$ is feasible to the dual program.

The complementary slackness conditions for the primal-dual pair are

$$(-y(s') - u_-(s') + u_+(s')) \cdot x(s') = 0 \quad \text{and}$$

$$(\check{p}(s' | s, a) + x(s')) \cdot y(s') = 0;$$

they are satisfied by $x_f(s')$ and $y_f(s')$.

Indeed, if $\check{p}(s' | s, a) \geq 0$, then $y_f(s') = 0$ and $x_f(s') = 0$, so the second multipliers in the complementary slackness conditions are equal to zero. Otherwise $y_f(s') = u_+(s') - u_-(s')$ and $x_f(s') = -\check{p}(s' | s, a)$, and the first multipliers are equal to zero.

If there are only finitely many optimization variables in the objective function (that is, $|\mathbb{B}^C| < \infty$), then the programs (P) and (D) are strongly dual. Otherwise we need to additionally establish that the transversality condition of Lemma 4.14 holds.

Assuming that the complement \mathbb{B}^C of the truncation set \mathbb{B} is countably infinite, there exists a bijection $f : \mathbb{B}^C \rightarrow \mathbb{N}$. Let

$$y_i \triangleq y(f^{-1}(i)), \quad \mathbf{A}_{i,i-1} = 0, \quad \mathbf{b}_i = u_-(f^{-1}(i)) - u_+(f^{-1}(i)),$$

$$x_i \triangleq x(f^{-1}(i)), \quad \mathbf{A}_{i,i} = -1, \quad \mathbf{c}_i = \check{p}(f^{-1}(i) | s, a).$$

Using the new variables, the programs (P) and (D) are exactly in the lower-staircase form (LS-P) and upper-staircase form (US-D). Because all of the off-diagonal matrices $\mathbf{A}_{i,i-1}$ are equal to zero, both the transversality condition and (4.35) are satisfied. Therefore, y_f and x_f are optimal and the strong duality holds by Lemma 4.14.

Strong duality means that

$$J_P(y_f) = J_D(x_f) = \sum_{s' \in \mathbb{B}^C} \check{p}^-(s' | s, a) \cdot (u_-(s') - u_+(s')).$$

Substitution of this objective value into (4.38) yields

$$\eta_{\pi, \pm}^{\mathbb{U}, \mathbb{B}}(s, a) = \check{r}_{\pi}^{\mathbb{X}, \mathbb{B}}(s, a) + \gamma \cdot \left(\sum_{s' \in \mathbb{B}^C} \check{p}^-(s' | s, a) \cdot (u_-(s') - u_+(s')) \right)$$

4 The Countably-Infinite Model

$$\begin{aligned} y^- + y &= \max\{-y, 0\} + y = \triangleleft \\ \max\{-y + y, 0 + y\} &= \\ \max\{0, y\} &= y^+ \end{aligned}$$

$$\begin{aligned} &+ \sum_{s' \in \mathbb{B}^C} \check{p}(s' | s, a) \cdot u_-(s') \\ &= \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a) - \gamma \cdot \sum_{s' \in \mathbb{B}^C} \check{p}^-(s' | s, a) \cdot u_+(s') \\ &\quad + \gamma \cdot \sum_{s' \in \mathbb{B}^C} (\check{p}^-(s' | s, a) + \check{p}(s' | s, a)) \cdot u_-(s') \\ &= \check{r}_\pi^{\mathbb{X}_\mathbb{B}}(s, a) + \gamma \cdot \sum_{s' \in \mathbb{B}^C} \check{p}^+(s' | s, a) \cdot u_-(s') \\ &\quad - \gamma \cdot \sum_{s' \in \mathbb{B}^C} \check{p}^-(s' | s, a) \cdot u_+(s'). \end{aligned}$$

Once again, the derivation involves a change of summation order that is possible by the argument used in the derivation of (4.37).

QED

	low	high
<i>M</i>	5	10
m	1	5
λ	1	10
c	10	20
h	1	3
o_v	1	5
<i>o_f</i>	1	10

Table 4.1: Bounds of the parameters of the problem. See Section 2.4.5 p. 58 for details.

4.8 EXPERIMENTS

To illustrate the performance of **ASPIRE**, we applied it to thirty randomly generated two-product inventory management problems of Sections 1.3.2 and 2.4.5.

For all of the problems, the discounting rate was set to $\gamma = 0.95$. The initial states were distributed uniformly between all possible with up to nine units of each product, that is,

$$\alpha(\mathbf{s}) = 0.01 \quad \text{if} \quad \|\mathbf{s}\|_\infty \leq 9 \quad \text{and} \quad 0 \quad \text{otherwise.}$$

This way, each problem has 100 states for which the algorithm needs to identify the optimal actions. The demands $p_{d,i}$ for the products were Poisson-distributed, $p_{d,i} \sim \mathcal{P}(\lambda_i)$, with randomly chosen intensities $\lambda_i \sim \mathcal{U}_d(\lambda_{i,-}, \lambda_{i,+})$. The intensities and other remaining parameters were drawn uniformly from the intervals presented in Table 4.1.

In all problems, the initial policy provided to **ASPIRE** prescribed the same action in all states, namely:

$$\mathbf{a} = (a_0, a_1), \quad \text{where} \quad a_i = \left\lfloor \min\{\lambda_i, \frac{M}{2m_i}\} \right\rfloor.$$

This way, the agent tries to match the expected demands that are equal to the intensities **λ**. At the same time, because the data is generated at random, this initial action may not satisfy the maximum-shipment-measurement constraint $\langle \mathbf{m}, \mathbf{a} \rangle \leq M$. To ensure that this constraint holds, the actions were capped at $(M/2m_0, M/2m_1)$.

In all of the experiments, **ASPIRE** produced sequences of improving policies, as shown in Figure 4.2. Because the true values

are not computable finitely, the data presented in Figure 4.2 is only approximate. We evaluated the policies by computing their gains $J(\pi)$ as follows. For each problem, we used a series of zero-salvage truncations to compute approximate values until the difference between two successive approximations was less than 0.1%. We then employed the largest truncation obtained this way for policy evaluation after each pivot.

See Definition 2.14, p. 34

All of the instances terminated within approximately 100 pivots, which is equal to the number of the initial states. Intuitively, this happens because to be initial-decision optimal, the policy at least needs to change to optimal states in all of the 100 states in the support of the initial distribution. In some cases, the initial policy already prescribed optimal actions for some of those states and fewer pivots happened. In other cases, pivoting of only the initial actions only was not sufficient and more pivots took place.

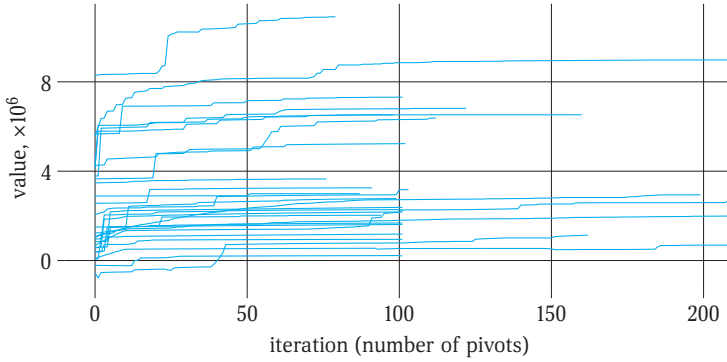


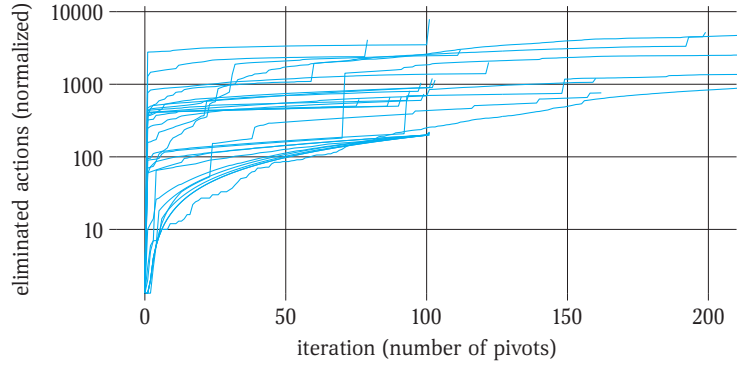
Figure 4.2: Policies produced by `ASPIRE` improve monotonically with each pivot. Each line represents one of the thirty problem instances. The horizontal axis shows the number of performed pivots.

In addition to policy improvement via pivoting, `ASPIRE` performs action elimination. Figure 4.3 shows the jumpy nature of the elimination process: when the truncation is enlarged, the approximate advantages used in the elimination procedure improve, allowing `ASPIRE` to eliminate a bulk of suboptimal actions at once.

4.9 CONCLUSION

In this chapter, we considered MDPs with countably-infinite state spaces. Because convergence in infinite-dimensional spaces is non-trivial, we proposed a set of assumptions to guarantee that further results hold. Then we illustrated how these assumptions can be checked in the inventory management problem.

Figure 4.3: Action elimination by ASPIRE. Each line represents one of the thirty problem instances. The horizontal axis shows the number of performed pivots. Note the logarithmic scale of the vertical axis.



We then demonstrated how approximate policy evaluation can be done in countably-infinite MDPs by augmenting the reward with a bonus function representing the unevaluated states outside of the truncation.

Next, we showed how this approximate evaluation procedure can be used in policy improvement. Based on the duality of policy values and policy-induced occupancies, we established that the advantages—negative dual slacks—can be used to identified policy-improving pivots. Since in countably-infinite problems the true advantages cannot be evaluated exactly, we proposed a method of evaluating salvage bounds and showed how these bounds can be used in both policy improvement and action elimination.

Using these theoretical development, we designed an algorithm called ASPIRE for solving countably-infinite MDPs via a series of increasing truncations. Our previous truncation-based algorithm MISHA prescribes the actions reactively, after the state is observed but before the action needs to be taken. ASPIRE is able to identify optimal actions in the support of the initial distribution and can be used to plan proactively, before the actual initial state is observed. Like MISHA, it is applicable to problems with unbounded rewards and produces strictly improving policies.

Even though the traditional approach of finding a universally optimal policy is computationally intractable in countably-infinite MDPs, ASPIRE can be used for provably optimal decision-making: once the action prescribed by ASPIRE is taken, the procedure can be repeated for the distribution of the next state, leading to optimal behavior.

5

Generalized Optimistic Q-Learning

*Optimism is essential to achievement
and it is also the foundation of courage
and true progress.*

— Nicholas Murray Butler,
*Commencement Addresses,
The Responsibility of Youth*

REINFORCEMENT LEARNING, like any on-line learning method, inevitably faces the exploration-exploitation dilemma. When a learning algorithm requires as few data samples as possible, it is called sample efficient. The design of sample-efficient algorithms is an important area of research. Interestingly, all currently known provably efficient model-free RL algorithms utilize the same well-known principle of optimism in the face of uncertainty. We unite these existing algorithms into a single general model-free optimistic RL framework. Using the proposed framework, we study sample-efficiency of optimistic reinforcement learning in terms of regret, that is, the value loss in the learning process. We show how this facilitates the design of new optimistic model-free RL algorithms by simplifying the analysis of their efficiency. Finally, we propose one such new algorithm and demonstrate its performance in an experimental study.

This chapter is based on the article published in the Proceedings of the Nineteenth International Conference on Autonomous Agents and Multiagent Systems [Neustroev and de Weerd, 2020].

Minor changes were made to the text to improve readability.

5.1 INTRODUCTION

Reinforcement learning [Sutton and Barto, 2018] is a popular framework for sequential decision-making problems in an unknown environment, applicable to a wide range of problems. In general, RL methods fall into two categories: model-based and model-free. Model-based approaches build an approximate model of the environment and use it to reason about optimality of actions. Model-free approaches, in contrast, estimate optimality of actions directly. To find the best possible course of actions, reinforcement learning requires many repeated trials, which is effective but costly. Therefore, one of the important challenges in reinforcement learning is the design of *sample-efficient* algorithms, that is, algorithms utilizing as much information from each interaction as possible. Sample efficiency of model-based reinforcement learning has been studied extensively, and several methods were proven to be sample efficient [Azar, Osband, et al., 2017; Kakade et al., 2018].

Even though most RL breakthroughs—from seminal Q-learning [Watkins, 1989] to state-of-the-art deep Q-networks [Mnih, Kavukcuoglu, Silver, Graves, et al., 2013; Hessel et al., 2018]—are of the model-free paradigm, theory on sample efficiency of model-free reinforcement learning remains limited. Only recently some dispersed results have appeared for a few model-free methods. For proper understanding of the potential of model-free reinforcement learning, and thus of the design of optimal RL algorithms, we need

to identify the relation between the efficiency of these methods and various components of their design.

The first provably efficient model-free RL algorithm was introduced by Jin et al. [2018]. It is called upper confidence bound Q-learning and comes in two forms: UCB-H and UCB-B. Its conception sparked interest in sample complexity of model-free reinforcement learning; as a result, several similar methods have been proposed, namely, ∞ -UCB, OPIQ Q-learning [Y. Wang et al., 2020; Rashid et al., 2020]. All of these methods attribute their success to the use of the same learning rate proposed by Jin et al. [2018]. Another factor that allows these (both model-based and model-free) algorithms to achieve sample efficiency is their use of *optimism in the face of uncertainty* [Szita and Lőrincz, 2008]. We aim to better understand how optimism affects the efficiency of reinforcement learning.

The main contribution of this chapter is a generalized theory on optimistic Q-learning which unifies the existing algorithms. In the context of model-based methods, there already exists a generalization known as optimistic initial model [ibid.]. Instead, we focus on model-free methods because they have better space complexity and can be adapted to deep learning, which is arguably the most promising direction of future work.

We also perform a generalized theoretical analysis of sample efficiency. In order to establish efficiency of an algorithm, two related techniques are used. Some authors provide PAC-bounds on the *time* required to achieve near-optimal performance [Strehl, Li, Wiewiora, et al., 2006; Strehl, Li, and Littman, 2009; Kakade et al., 2018; Y. Wang et al., 2020]. We employ another approach and establish efficiency by showing that the *regret* of the algorithm—the total loss of reward incurred while learning—grows sub-linearly with respect to the number of interactions [Jin et al., 2018; Bai et al., 2019; Rashid et al., 2020]. The two approaches are similar; in fact, it is known that one implies the other, and *vice versa* [Jin et al., 2018; Osband and Van Roy, 2017].

PAC stands for *probably approximately correct*, and means that an equation holds with high probability and low absolute error, both of which can be chosen *a priori* in an arbitrary way.

To summarize, in this work, we study the effects of optimism on the regret of model-free RL algorithms. We start with examining the existing sample-efficient Q-learning methods and identifying their common features. Then we propose a generalized model of optimistic Q-learning, which encompasses these methods. Next, we perform a theoretical regret analysis and derive a regret bound for the generalized model, which allows us to identify the sources of

regret. We show how these general results can be used to facilitate the design of new optimistic model-free algorithms by proposing one such algorithm, and evaluate its performance experimentally.

5.2 Preliminaries

5.2 PRELIMINARIES

This section introduces the underlying model and our notation.

5.2.1 Non-Stationary Episodic Markov Decision Processes

We use episodic non-stationary Markov decision process (non-stationary MDP) as an underlying model because the total regret is a well-defined value in episodic learning [Y. Wang et al., 2020] but is not as clearly defined in other settings. An episodic non-stationary MDP is defined as a tuple

$$\mathfrak{M}_{H,K} \triangleq (H, K, \mathbb{S}, \mathbb{A}, A_{p,h}, p_h, r_h).$$

In this setting, the agent interacts with the environment for K episodes, each consisting of H time steps for the total number of $T \triangleq HK$ interactions. We denote the sets of all episodes and steps of each episode as $\mathbb{K} \triangleq \{1, \dots, K\}$ and $\mathbb{H} \triangleq \{1, \dots, H\}$. At each time step h , an agent observes the state of the environment $s_h \in \mathbb{S}$ and chooses one of the available actions $a_h \in A_{p,h}(s_h) \subseteq \mathbb{A}$. The environment transitions to a new state s_{h+1} with probability $p_h(s_{h+1} | x_h)$; the agent observes this transition and receives a reward $r_h(x_h)$. We use $x_h \triangleq (s_h, a_h)$ for state-action pairs and

$$\mathbb{X}_h \triangleq \{(s, a) \mid s \in \mathbb{S} \text{ and } a \in A_{p,h}(s)\}$$

for the admissible control space in time step h .

Given the state s at time step h , the value $v_{\pi,h}(s)$ of a policy $\pi \in \mathbb{II}$ that can be found using the Bellman policy equations:

$$v_{\pi,h}(s) = q_{\pi,h}(s, \pi_h(s)), \quad v_{\pi,H+1}(s) = 0, \quad (5.1)$$

$$\begin{aligned} q_{\pi,h}(x) &= [r_h + \gamma \cdot \mathcal{T}_h v_{\pi,h+1}](x), \\ [\mathcal{T}_h y](x) &\triangleq \sum_{s' \in \mathbb{S}} p_h(s' | x) \cdot y(s') \quad \text{for all } y : \mathbb{S} \rightarrow \mathbb{R}. \end{aligned} \quad (5.2)$$

The agent needs to learn an optimal policy, that is, a policy π_\star with the highest possible values

$$v_{\pi_\star,h}(s) = v_{\star,h}(s) \triangleq \max_{\pi \in \mathbb{II}} v_{\pi,h}(s).$$

The optimal values $v_{\star,h}(s)$ satisfy the Bellman optimality equations

$$\begin{aligned} v_{\star,h}(s) &= [\mathcal{M}_h q_{\star,h}](s), & v_{\star,H+1}(s) &= 0, \\ q_{\star,h}(x) &= [r_h + \gamma \cdot \mathcal{T}_h v_{\star,h+1}](x), \end{aligned} \quad (5.3)$$

$$\text{where } [\mathcal{M}_h y](s) \triangleq \max_{a \in A_{p,h}(s)} y(s, a) \quad \text{for all } y : \mathbb{X}_h \rightarrow \bar{\mathbb{R}}.$$

In each episode k , the agent follows some policy π_k . When these policies are suboptimal, they cause a loss of the total γ -discounted reward, known as the regret.

Definition 5.1 | total regret

The (expected) *total regret* R of such agent in an episodic non-stationary MDP $\mathfrak{M}_{H,K}$ is defined as

$$R \triangleq \sum_{k=1}^K R_k = \sum_{k=1}^K (v_{\star,1}(s_{1,k}) - v_{\pi_k,1}(s_{1,k})).$$

Finally, in this chapter we assume that the rewards and values are bounded, but the bounds may vary between steps, that is, $r_h(x) \in [r_{-,h}, r_{+,h}]$ and $v_{\pi,h}(x) \in [v_{-,h}, v_{+,h}]$ for all $x \in \mathbb{X}$ and π . For simplicity, we use deterministic rewards; however, our results can be extended to randomized rewards. We denote the reward bounds of the whole episode as $r_{\pm}(H)$, that is, $r_{-}(H) \leq \min_{h \in \mathbb{H}} r_{-,h}$ and $r_{+}(H) \geq \max_{h \in \mathbb{H}} r_{+,h}$. We denote the reward span of a step as $r_{\Delta,h} \triangleq r_{+,h} - r_{-,h}$, and of an episode as $r_{\Delta}(H) \triangleq r_{+}(H) - r_{-}(H)$. We define the value bounds $v_{\pm,H}$ and spans $v_{\Delta,h}$ and $v_{\Delta}(H)$ similarly.

5.2.2 Reinforcement Learning

In reinforcement learning, the transition and reward functions of an MDP are not known, so the Bellman optimality equation (5.4) cannot be applied directly. Instead, the optimal Q-values are learned through interactions with the environment. The initial Q-values $q_{h,0}(x)$ are chosen arbitrarily, and at each episode $k + 1$ they are gradually updated from the previous Q-values $q_{h,k}(x)$. In Q-learning [Watkins, 1989], the update rule is:

$$q_{h,k+1}(x) = \begin{cases} (1 - \alpha_t) \cdot q_{h,k}(x) + \alpha_t \cdot U_{h,k}(x, s_{h+1}), & \text{if } x = x_{h,k+1}, \\ q_{h,k}(x), & \text{otherwise;} \end{cases} \quad (5.5)$$

we call the term $U_{h,k}(x, s)$ the update.

Definition 5.2

The *update* $U_{h,k}(x, s)$ of Q-learning is a function defined as

$$U_{h,k}(x, s) \triangleq r_h(x) + \gamma \cdot [\mathcal{M}_{h+1} q_{h+1,k}](s).$$

5.2 Preliminaries

- To easier relate these values to the optimal Q-values $q_{\star,h}(x)$, we define the following operator.

Definition 5.3

The *empirical transition operator* $\hat{\mathcal{T}}_{h,k}$ for each $k \in \mathbb{K}$ and $h \in \mathbb{H}$:

$$[\hat{\mathcal{T}}_{h,k} y](x) \triangleq y(s_{h+1,k}) \quad \text{if } h < H, \quad \text{and} \quad [\hat{\mathcal{T}}_{H,k} y](x) \triangleq 0. \quad (5.6)$$

- Using this operator, the update term can be written similarly to the Bellman equations (5.3) and (5.4):

$$\begin{aligned} U_{h,k}(x, s_{h+1,k}) &\triangleq [r_h + \gamma \cdot \hat{\mathcal{T}}_{h,k} v_{h+1,k}](x) && \text{with} \\ v_{h,k}(s_{h,k}) &\triangleq [\mathcal{M}_h q_{h,k}](s_{h,k}). \end{aligned}$$

Definition 5.4 | learning rate

The function α_t in (5.6) is called the *learning rate*.

- We use t as a shorthand for the *realized visitation function* $\#_{h,k}(x)$, which gives the number of times the state-action pair x has been visited in time step h of the first k episodes.

The learning rate is used to balance the newly acquired information $U_{h,k}(x, s)$ with the old experiences $q_{h,k}(x)$. For an appropriate choice of the learning rate, the sequence $(q_{h,k}(x))_{k=1}^{\infty}$ converges to $q_{\star,h}(x)$ with probability one, if the state-action space \mathbb{X} is finite $|\mathbb{X}| < \infty$ and the rewards function r is uniformly bounded [Jaakkola et al., 1994].

In particular, the conditions on the learning rate are:

$$\sum_{t=1}^{\infty} \alpha_t(x) = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} \alpha_t^2(x) < \infty \quad \text{for all } x \in \mathbb{X}. \quad (5.7)$$

The first condition ensures that the updates remain large enough to affect Q-values, while the second condition guarantees that the variance of the resulting iterative stochastic process remains bounded (i.e., that it converges).

Using the notation of Jin et al. [2018], we introduce the following values.

Definition 5.5 | cumulative learning rates

Given a learning rate α , the *cumulative learning rates* are given by

$$\alpha_{t,0} = \prod_{j=1}^t (1 - \alpha_j), \quad \text{and} \quad \alpha_{t,i} = \alpha_i \cdot \prod_{j=i+1}^t (1 - \alpha_j). \quad (5.8)$$

For $t = 0$, we define $\alpha_{t,0} \triangleq 1$ and $\sum_{i=1}^t \alpha_{t,i} \triangleq 0$. If a state-action pair $x = (s, a)$ was previously visited in time step h of episodes $k_1, \dots, k_t < k$, then by the update equation (5.5) on k_i we can write

$$q_{h,k}(x) = \alpha_{t,0} \cdot q_{h,0}(x) + \sum_{i=1}^t \alpha_{t,i} \cdot U_{h,k_i}(x, s_{h+1,k_i}). \quad (5.9)$$

5.3 OPTIMISM IN Q-LEARNING

This section presents our main contribution. We start with an overview of optimism in model-free RL methods. Then we propose a generalized framework of optimistic reinforcement learning. Next, we formulate the conditions under which the total regret of optimistic Q-learning can be bounded and present an intuitive interpretation of the bound.

5.3.1 Representation of Optimism

As briefly mentioned in Section 5.1, the principle of optimism in the face of uncertainty is usually applied in two ways: optimistic initialization, and use of UCBs in action selection. We looked at UCB-H [Jin et al., 2018], UCB-B [ibid.], ∞ -UCB [Y. Wang et al., 2020], and OPIQ [Rashid et al., 2020] to see how they incorporate these two aspects of optimism.

For initialization, all of the methods use $q_{h,0}(x) = v_+(H) = v_+$ except for OPIQ. The latter uses $q_{h,0}(x) = v_-$, but additionally augments Q-values with a *bonus for optimism* $u(t)$, depending on the visitation counter t . These *augmented* Q-values

$$\bar{q}_h(x) \triangleq q_h(x) + u(t)$$

overestimate the true Q-values (i.e., they are optimistic) and are used for action selection. The particular choice of this bonus is $u(t) = C/(t+1)^M$, where $C \geq v_\Delta$ and M is a sufficiently large number. It ensures that the augmented Q-values $\bar{q}_{h,0}(x)$ of unvisited state-action pairs are optimistic:

$$\bar{q}_{h,0}(x) = q_{h,0}(x) + u(t) \geq v_- + v_\Delta/1^M = v_+.$$

If $t > 0$, however, the bonus for optimism becomes close to zero as $\lim_{M \rightarrow \infty} C/(t+1)^M = 0$ and the effect of the augmentation vanishes fast. This bonus for optimism is motivated by deep learning models, where it is hard to ensure optimistic initialization,

but an addition of an extra summand is easier to implement [ibid.]. As deep learning represents an interesting area of study, we choose to keep the bonus for optimism in our model and allow arbitrary initialization. We allow this bonus for optimism $u_h(t)$ to differ with time step h , and therefore define the augmented Q-values as follows.

5.3 Optimism in Q-Learning

Definition 5.6 | augmented Q-values

The *augmented* Q-values and *augmented values* are equal to

$$\bar{q}_h(x) \triangleq q_h(x) + u_h(t), \quad \bar{v}_h(s) \triangleq \min\{v_{+,h}, [\mathcal{M}_h \bar{q}_h](s)\}. \quad (5.10)$$

- For exploration, all of the models store UCB Q-values and explore greedily based on them. Compared to regular Q-learning, these Q-values include an additional term that we call the confidence bonus.

Definition 5.7 | confidence bonus

The *confidence bonus* $b(t)$ is a UCB-based term added to the updates,

$$U_{h,k}(x, s) \triangleq r_h(x) + \gamma \cdot [\mathcal{M}_{h+1} q_{h+1,k}](s) + b(t).$$

- The goal of this bonus is to ensure that the learned Q-values $q_{h,k}(x)$ are the UCB-estimates of the optimal Q-values $q_{\star,h}(x)$. The exact form of the bonus depends on which concentration inequalities are used in the method's design. These concentration inequalities provide probabilistic bounds on the total regret, and the bonuses are carefully crafted to ensure that the resulting bounds hold with high probability $1 - \delta$. Instead of designing bonuses to guarantee the probability that regret bound holds, we do the reverse, that is, we allow arbitrary bonuses $b_h(t)$, and see how they affect the probability δ .

Additionally, we introduce the following two auxiliary functions.

Definition 5.8 | cumulative confidence bonus

The *cumulative confidence bonus* $\beta_h(t)$ is given by

$$\beta_h(t) \triangleq \sum_{i=1}^t \alpha_{t,i} b_h(i).$$

Definition 5.9 | total cumulative bonus

The *total cumulative bonus* $\vartheta_h(t)$ is equal to

$$\vartheta_h(t) \triangleq \beta_h(t) + u_h(t).$$

Figure 5.1: Generalized optimistic Q-learning

Data: episodic non-stationary MDP $\mathfrak{M}_{H,K}$, initial Q-values $q_{h,0}$, bonuses $u_h(t)$ and $\beta_h(t)$, learning rate α_t , and exploration rate ϵ .

- 1 Initialize Q-table $q_h(x) \leftarrow q_{h,0}$ and visitation counter $\#_h(x) \leftarrow 0$ for all $h \in \mathbb{H}, x \in \mathbb{X}_h$;
- 2 **for** episode $k \leftarrow 1, \dots, K$ **do**
- 3 observe initial state s_1 ;
- 4 **for** step $h \leftarrow 1, \dots, H$ **do**
- 5 take action $a_h \leftarrow \text{Greedy}_\epsilon(\bar{q}_h, s_h)$, where $\bar{q}_h(x) \triangleq q_h(x) + u_h(t)$;
- 6 receive reward r_h , observe next state s_{h+1} , and let $x_h = (s_h, a_h)$ denote the current state-action pair;
- 7 increment visitation counter $t = \#_h(x_h)$ by 1;
- 8 compute confidence bonus $b_h(t) \leftarrow \alpha_t^{-1} \cdot \beta_h(t) + (1 - \alpha_t^{-1}) \cdot \beta_h(t - 1)$;
- 9 compute update $U_h(x_h, s_{h+1}) \leftarrow r_h(x_h) + b_h(t) + \gamma \cdot \bar{v}_{h+1}(s_{h+1})$, where $\bar{v}_h(s) \triangleq \min\{v_{+,h}, [\mathcal{M}_h \bar{q}_h](s)\}$ update Q-table $q_h(x_h) \leftarrow (1 - \alpha_t) \cdot q_h(x_h) + \alpha_t \cdot U_h(x_h, s_{h+1}, t)$;

🌀 The total cumulative bonus $\vartheta_h(t)$ represents all of the optimistic bias of an algorithm and which plays an important role in our analysis.

Summarizing the aforementioned, a generalization of the UCB-based methods should include two kind of bonuses: a bonus for optimism $u_h(t)$ and a confidence bonus $b_h(t)$ (or its cumulative form $\beta_h(t)$), and use the augmented Q-values $\bar{q}_h(x)$.

5.3.2 Generalized Optimistic Q-Learning

Following the discussion of Section 5.3.1, the existing sample-efficient optimistic Q-learning methods differ with respect to three hyperparameters: initial Q-values $q_{h,0}$, bonus for optimism $u_h(t)$, and cumulative confidence bonus $\beta_h(t)$. We unify these methods into a single algorithm, which we name *Generalized optimistic Q-learning*. It is presented in Figure 5.1. Table 5.1 summarizes how the existing methods fit into this framework.

Algorithm 5.1 has two extra hyperparameters, a learning rate α_t and an exploration rate ϵ . It is shown in [Jin et al., 2018] that the learning rate $\alpha_t = (H+1)/(H+t)$ offers significant improvements in performance compared to previously considered rates $\alpha_t = t^{-1}$ and $t^{-\omega}$, where $0.5 < \omega \leq 1$ is a constant. Therefore, it is possible that

other learning rates may offer similar, or even better improvements.

We want generalized optimistic Q-learning to be as general as (reasonably) possible, so we include the exploration rate ε as a parameter. This allows us to represent several other methods in our framework as well, as shown at the top of Table 5.1. In our theoretical study, however, we assume greedy action selection, that is, $\varepsilon = 0$, as is the case for all variants of UCB, and we leave the analysis of regret for $\varepsilon > 0$ as an interesting future direction.

Following the discussion of Section 5.3.1, we would like to point out that the update equation (5.5) of Algorithm 5.1 uses a slightly different update term (see step 9) by adding a bonus term $u_h(t)$:

$$U_{h,k}(x, s) \triangleq r_h(x) + b_h(\#_{h,k}(x)) + \gamma \cdot \bar{v}_{h+1,k}(s), \quad \text{where} \quad (5.11)$$

$$\bar{v}_{h+1,k}(s) \triangleq \min\{v_{+,h+1}, [\mathcal{M}_{h+1}\bar{q}_{h+1,k}](s)\} \quad \text{and} \quad (5.12)$$

$$\bar{q}_{h,k}(x) \triangleq q_{h,k}(x) + u_h(\#_{h,k}(x)). \quad (5.13)$$

New optimistic model-free RL algorithms can be expressed by Algorithm 5.1 with different hyperparameter combinations. Below we present a novel algorithm, which is designed using this framework.

Example 5.1 | UCB-H with generalized learning rate, UCB-H⁺

UCB-H⁺ follows the flow of Algorithm 5.1 with the hyperparameters presented in the last row of Table 5.1. In particular, UCB-H⁺ utilizes a new learning rate

$$\alpha_t \triangleq \frac{\lambda H + 1}{\lambda H + t^\omega}, \quad \text{where } \lambda \geq 0 \text{ and } \frac{1}{2} < \omega \leq 1. \quad (5.14)$$

The learning rate of UCB-H⁺ generalizes the previously used learning rates, complies with the learning rate conditions (5.7), and is motivated by two observations. Firstly, for the discounted problems the learning rate $t^{-\omega}$ outperforms $1/t$, and the best performance is achieved for $\omega \approx 0.8$ [Even-Dar, Mannor, et al., 2006; Azar, Munos, et al., 2011]. Secondly, switching from $\alpha_t = 1/t$ to $(H+1)/(H+t)$ allowed Jin et al. [2018] to bound the regret blow-up with respect to H and achieve efficiency. We would like to note that our generalized framework does not rely on this particular learning rate, instead, this example serves as an illustration.

🌀 The generality of our framework complicates the theoretical analysis of Algorithm 5.1. To achieve interesting, interpretable results, we need to impose at least some conditions on the hyperparameters of the model. None of the conditions we use are particularly

Q-learning variant	$q_{h,0}$	α_t	ε	$u_h(t)$	$\beta_h(t)$	regret
regular	[Watkins, 1989; Even-Dar, Mannor, et al., 2006]	any	$t^{-\omega}$	ε	0	$\Omega_{HX}(T)$
optimistic	[Even-Dar and Mansour, 2002]	$v_+/\alpha_{T,0}$	$t^{-\omega}$	ε	0	?
speedy	[Azar, Munos, et al., 2011]	any	t^{-1}	ε	0	$\tilde{\mathcal{O}}_{HX}(T^{2/3})$
UCB-H	[Jin et al., 2018]	v_+	$\frac{H+1}{H+t}$	0	0	$\tilde{\mathcal{O}}(H^2\sqrt{TX})$
UCB-B	[Jin et al., 2018]	v_+	$\frac{H+1}{H+t}$	0	0	$\frac{1}{2} \min \left\{ c_1 \cdot \left(\sqrt{\frac{H(W_t+H)^1}{t}} + \sqrt{\frac{HTX}{t}} \right), \right.$
					$\left. c_2 \sqrt{\frac{H^3}{t}} \right\}$	$\tilde{\mathcal{O}}(H\sqrt{HTX})$
∞ -UCB	[Y. Wang et al., 2020]	v_+	$\frac{H+1}{H+t}$	0	0	$\tilde{\mathcal{O}}_H(\sqrt{TX})$
OP1Q	[Rashid et al., 2020]	v_-	$\frac{H+1}{H+t}$	0	$\frac{C}{(t+1)^M}$	$\tilde{\mathcal{O}}(H^2\sqrt{TX})$
UCB-H ⁺	this chapter	$v_{+,h}$	$\frac{\lambda H+1}{\lambda H+t^\omega}$	0	0	$\tilde{\mathcal{O}}(\mu\sqrt{H^{\omega-1}T^{2-\omega}X^\omega})$

5 Generalized Optimistic Q-Learning

Table 5.1: Different Q-learning algorithms as generalized optimistic Q-learning. Below the line are provably efficient methods.

restrictive, and they—sometimes trivially—hold for all of the existing optimistic methods, albeit without being explicitly mentioned. At the same time, these conditions encompass a broader class of models, including the aforementioned UCB-H^+ .

5.3 Optimism in Q-Learning

Conditions on the learning rate

We start with conditions on the learning rate α_t . By inspection of various proofs involving the learning rates presented in Table 5.1, we identified that their successful application can be attributed to the following condition.

Condition 5.1 | initial learning rate is one

The learning rate satisfies $\alpha_1 = 1$.

- Intuitively, Condition 5.1 means that when a state-action pair is visited for the first time, the update equation becomes

$$q_k = (1 - \alpha_1) \cdot q_0 + \alpha_1 \cdot U = U,$$

and the initial value q_0 becomes “forgotten”, being replaced by a UCB -based update U . Thus, under a condition $\alpha_1 = 1$ the initialization affects the optimistic view of unencountered state-action pairs only.

Iterative approximation of optimal Q-values via (5.9) leads to a scaling factor of $\sum_{i=1}^t \alpha_{t,i}$. As the learning process is stochastic, we want to ensure that its variance remains bounded similarly to (5.7). Moreover, as UCB depends on this variance, we need to be able to quantify it in order to compare the bonus terms we use to the actual confidence bounds. This observation leads us to the following condition.

Condition 5.2 | asymptotic of squared α

There exists a function $0 \leq \zeta(t) \leq 1$ such that

$$\sum_{i=1}^t (\alpha_{t,i})^2 \leq \zeta^2(t).$$

- Next, to quantify the total regret, we need to be able to express its propagation from one time step to another; we see from Corollary 5.10 that the total regret inflates by a factor of $\gamma \cdot \eta(H, K)$ with each step, where $\eta(H, K)$ satisfies the following condition.

Condition 5.3 | asymptotic of residual α

There exists a function $\eta(H, K) \geq 1$ such that

$$\sum_{n=t}^K \alpha_{t,n} \leq \eta(H, K).$$

We omit the arguments of η and other functions introduced later for brevity of notation, if it does not lead to ambiguity.

- Knowing the learning rate, it is possible to express η analytically. For example, Jin et al. [2018] show that $\sum_{n=t}^{\infty} \alpha_{t,n} \leq 1 + 1/H = \eta(H)$ in their analysis, which implies Condition 5.3. However, without any assumptions on the form of the learning rate, we have to fall back to η as a generalized term.

Function η serves as a “scaling factor” for the total regret, but there are other scale parameters, for example, the discounting factor, the lower $r_{-,h}$ and the upper $r_{+,h}$ reward functions affect the total regret scale as well. We want to be able to quantify their effect and combine all of the scale parameters together as follows.

Condition 5.4 | asymptotic of the values

Let $v_{\uparrow,h}$ denote the asymptotically dominant term between the upper value function $v_{+,h}$ and the value span $v_{\Delta,h}$, that is,

$$v_{\uparrow,h} \triangleq \begin{cases} v_{\Delta,h} & \text{if } v_{+,h} = \mathcal{O}(v_{\Delta,h}), \\ v_{+,h} & \text{otherwise,} \end{cases}$$

and similarly for the reward bound $r_{\uparrow}(H)$ and the value bound $v_{\uparrow}(H)$. Then there exists a function $\mu(H, K, \gamma)$ such that

$$\sum_{h=1}^H (\gamma\eta)^{h-1} v_{\uparrow,h} = \mathcal{O}(\mu(H, K, \gamma)). \quad (5.15)$$

- We call the function μ of Condition 5.4 the *magnitude function*, because it quantifies the asymptotic behavior of the total regret blowup in all H time steps. Intuitively, regret of each time step is at most $v_{\Delta} = \mathcal{O}(v_{\uparrow})$, which means that the total regret grows at most at a rate of $\sum_{h=1}^H (\gamma\eta)^{h-1} v_{\uparrow,h}$ as H grows. The magnitude function quantifies this rate.

All of the existing UCB-based methods utilize the same learning rate $\alpha_t = (H+1)/(H+t)$ as showed in Table 5.1. It is easy to check that this learning rate satisfies Conditions 5.1–5.4. In particular, $\zeta(t) = 2H/t$ and $\eta(H) = 1+1/H$ are proposed by Jin et al. [ibid.] and used by other authors [Jin et al., 2018; Y. Wang et al., 2020; Rashid et al., 2020]. Due to the fact that $(1 + 1/H) < e$, the magnitude function equal to $\mu(H) = V_{\uparrow} = H$ is used.

Conditions on the bonuses

All of the remaining conditions are rather intuitive. The first one addresses the initialization and was already discussed in Section 5.3.1. We require that the initial values are not too high or too low, and that the augmented initial values $\bar{q}_{h,0}$ used in action selection are optimistic.

Condition 5.5 | initial values are optimistic

The initial values $q_{h,0}$ belong to intervals $[\nu_{-,h}, \nu_{+,h}]$, and the bonus for optimism $u_h(t)$ is such that $q_{h,0} + u_h(0) \geq \nu_{+,h}$.

5.3 Optimism in Q-Learning

Finally, we present two conditions (5.6 and 5.7) on the bonuses.

Condition 5.6 | bonuses decrease with visitations

The total bonus function is non-negative and non-increasing in t , $\vartheta_h(t) \geq \vartheta_h(t+1) \geq 0$ for all $t \in \mathbb{N}$.

As t represents the number of visitations of a state-action pair, we want the bonus to decrease as it grows, that is, as we collect more samples and build higher confidence. Non-negativity ensures that the bonuses are optimistic.

Condition 5.7 | asymptotic of the bonuses

There exists a *bonus scaling* function $\theta(t)$ such that

$$\sum_{n=1}^t \vartheta_h(n) = \mathcal{O}(\nu_{\uparrow,h} \cdot \theta(t)).$$

This condition is used to quantify the effect of the total bonus $\vartheta_h(t)$ on the regret by a function $\theta(t)$, similarly to how the magnitude function μ quantifies the other effects.

The existing methods satisfy Conditions 5.5 and 5.6 trivially. Condition 5.7 depends on the particular bonus design, and also holds for all of the methods. For example, UCB-H and OPIQ both use $\theta(t) = \sqrt{Ht}$ as the bonus scaling function, although implicitly.

5.3.3 The Total Regret Bound

Finally, we are ready to give a high-probability bound on the total regret, which is our main theoretical contribution. The total regret is bounded by the sum of three different terms, each amplified by the *magnitude* function μ of Condition 5.4. These terms are:

- the *size* of the admissible control space

$$X \triangleq |\mathbb{X}|,$$

- the total effect of the *bonuses*

$$B \triangleq X \cdot \theta(K/X),$$

which depends on the bonus scaling function of Condition 5.7, and

- the total effect of the *estimation error*

$$E \triangleq c\sqrt{Kl},$$

where

$$l \triangleq \ln(TX/\delta)$$

is the logarithmic term.

The state-action space size X represents the effect of the *optimistic initialization*, as the number of initial values is proportionate to X . The bonus effect B relates to *optimistic action selection*.

The third factor E is caused by replacing the unknown transition operator (5.2) with its empirical counterpart (5.6). The constant c depends on how much uncertainty there is in the transitions, and is formally introduced later. An important property is that for deterministic problems $c = 0$, and the estimation term disappears. The probability δ used in the estimation error term E depends on our confidence in the total regret bound, that is, the bound holds with probability at least $1 - 2\delta$. It depends on the choice of the cumulative confidence bonus $\beta_h(t)$ as follows:

$$\delta = \begin{cases} 2KX \cdot \sum_{h \in \mathbb{H}} \exp\left(-\frac{1}{2} \left(\frac{\beta_h(t)}{\gamma c v_{\Delta, h+1} \cdot \zeta(t)}\right)^2\right), & \text{if } c > 0, \\ 0, & \text{if } c = 0, \end{cases} \quad (5.16)$$

The following theorem formalizes these results.

Theorem 5.1 * total regret in optimistic Q-learning

Let Conditions 5.1–5.7 hold. Then for some constant $0 \leq c \leq 1$, with probability at least $1 - 2\delta$ the total regret of generalized optimistic Q-learning with no exploration (i.e., when $\varepsilon = 0$) is bounded by

$$R(\mathfrak{M}_{H,K}, \alpha, \vartheta) = \mathcal{O}(\mu \cdot (X + B + E)), \quad (5.17)$$

- If there are no random transitions in the non-stationary MDP, the learning process becomes fully deterministic as well (we assume no random exploration). This leads us to the following corollary.

Corollary 5.2 * total regret in deterministic MDPs

If the transitions of the underlying non-stationary MDP $\mathfrak{M}_{H,K}$ are deterministic, the total effect of the estimation error is equal to zero, $E = 0$. Moreover, the bound of Theorem 5.1 holds with probability one.

5.4 PROOF OF THEOREM 5.1

We prove Theorem 5.1 by using a recurrent decomposition of the regret of a time step h in terms of the next time step $h + 1$. We bound the regret of each time step using the differences between augmented Q-values $\bar{q}_h(x)$ of generalized optimistic Q-learning and the optimal Q-values $q_{\star,h}(x)$. To derive these bounds, we employ some properties of the learning rate.

5.4 Proof of Theorem 5.1

5.4.1 Properties of the Learning Rate

We prove two lemmas, both relying on Condition 5.1 only.

Lemma 5.3 * learning rate sums into one

If $\alpha_1 = 1$, then

- $\alpha_{t,0} = 0$ and $\sum_{i=1}^t \alpha_{t,i} = 1$ for $t \geq 1$;
- $\sum_{i=0}^t \alpha_{t,i} = 1$ for any $t \geq 0$.

Proof. By definition, $\alpha_{t,0} = (1 - \alpha_1) \cdot \prod_{j=2}^t (1 - \alpha_j) = 0$.

We prove that $\sum_{i=1}^t \alpha_{t,i} = 1$ by induction. For $t = 1$, $\sum_{i=1}^1 \alpha_{1,i} = \alpha_1 = 1$. Assume that $\sum_{i=1}^t \alpha_{t,i} = 1$. Then using the definition of $\alpha_{t,i}$,

$$\begin{aligned} \sum_{i=1}^{t+1} \alpha_{t+1,i} &= \sum_{i=1}^t \alpha_i \prod_{j=i+1}^{t+1} (1 - \alpha_j) + \alpha_{t+1} \\ &= \left(\sum_{i=1}^t \alpha_i \cdot \prod_{j=i+1}^t (1 - \alpha_j) \right) \cdot (1 - \alpha_{t+1}) + \alpha_{t+1} \end{aligned}$$

where the expression in the first brackets is equal to $\sum_{i=1}^t \alpha_{t,i} = 1$ by the induction hypothesis, and therefore $\sum_{i=1}^{t+1} \alpha_{t+1,i} = 1$.

The second statement follows trivially from the first for $t \geq 1$ and from the definition of $\alpha_{t,i}$ for $t = 0$. QED

Lemma 5.3 allows us to write

$$q_{\star,h}(x) = \sum_{i=1}^t \alpha_{t,i} \cdot q_{\star,h}(x)$$

similarly to the decomposition (5.9) of $q_{h,k}(x)$ in order to relate them to each other.

We also prove the following relation between the confidence bonus $b(t)$ and the cumulative confidence bonus $\beta(t)$, justifying our choice of the bonus in step 8 of Algorithm 5.1.

Lemma 5.4 * sums of bonuses

If for some function $\beta(t)$

$$b(t) \triangleq \alpha_t^{-1} \cdot \beta(t) + (1 - \alpha_t^{-1}) \cdot \beta(t-1)$$

and either $\alpha_1 = 1$ or $\beta(0) = 0$, then $\sum_{i=1}^t \alpha_{t,i} \cdot b(i) = \beta(t)$.

Proof. By induction. For $t = 1$, $\sum_{i=1}^1 \alpha_1^i \cdot b(i) = \alpha_1 \cdot b(1) = \beta(1) + (\alpha_1 - 1) \cdot \beta(0) = \beta(1)$. Assume $\sum_{i=1}^t \alpha_{t,i} \cdot b(i) = \beta(t)$ for some t . Then

$$\begin{aligned} \sum_{i=1}^{t+1} \alpha_{t+1,i} \cdot b(i) &= \sum_{i=1}^t \alpha_{t+1,i} \cdot b(i) + \alpha_{t+1} \cdot b(t+1) \\ &= (1 - \alpha_{t+1}) \cdot \beta(t) + \alpha_{t+1} \cdot b(t+1) \\ &= (1 - \alpha_{t+1}) \cdot \beta(t) + \beta(t+1) \\ &\quad + \alpha_{t+1} \cdot (1 - \alpha_{t+1}^{-1}) \cdot \beta(t) \\ &= \beta(t+1). \end{aligned}$$

QED

5.4.2 Bounds on Q-Value Differences

First, we show that the augmented Q-values $\bar{q}_h(x)$ are related to the augmented values $\bar{v}_{h+1}(s)$ of previous episodes as follows.

Lemma 5.5 * recursion on \bar{q}

This is a generalization
of Lemma 4.2 of Jin et al.
[2018].

For any step $h \in \mathbb{H}$, state-action pair $x = (s, a) \in \mathbb{X}_h$ and episode $k \in \mathbb{K}$, let $t \triangleq \#_{h,k}(x)$ and suppose that for state s action a was previously taken in time step h of episodes $k_1, \dots, k_t < k$. Then under Condition 5.1

$$\begin{aligned} [\bar{q}_{h,k} - q_{\star,h}](x) &= \alpha_{t,0} [q_{h,0} - q_{\star,h}](x) \\ &\quad + \sum_{i=1}^t \alpha_{t,i} \left(\gamma \cdot [\bar{v}_{h+1,k_i} - v_{\star,h+1}](s_{h+1,k_i}) \right. \\ &\quad \left. + \gamma \cdot [(\hat{\mathcal{T}}_{h,k_i} - \mathcal{T}_h) v_{\star,h+1}](x) \right) + \vartheta_h(t). \end{aligned} \quad (5.18)$$

Proof sketch. Similarly to the proof of Lemma 4.2 of Jin et al. [ibid.], we use (5.13) and (5.9) to express $\bar{q}_{h,k}(x)$ in terms of the initial values $q_{h,0}$. Then we apply Lemma 5.3 and the Bellman optimality equation (5.4) to do a similar decomposition for $q_{\star,h}(x)$. QED

Next, we introduce the parameter c that quantifies the difference between the empirical transition operator (5.6) and the true transition operator (5.2), both of which appear in (5.18).

Proposition 5.6 * estimation error bounds

Let $y(x) : \mathbb{X}_{h+1} \rightarrow [a, b]$. There exists a constant $0 \leq c \leq 1$ such that

$$c(a - b) \leq [(\hat{\mathcal{T}}_{h,k} - \mathcal{T}_h)y](x) \leq c(b - a).$$

5.4 Proof of
Theorem 5.1

Remark 5.1

Note that while the case $c = 1$ holds trivially for any problem, a smaller constant possibly exists. For example, if the transitions of an non-stationary MDP $\mathcal{M}_{H,K}$ are not random, operators $\hat{\mathcal{T}}_{h,k}$ and \mathcal{T}_h coincide and $c = 0$ provides a sharper bound.

Using Proposition 5.6 and Lemma 5.5, we bound the difference between the augmented Q-values $\bar{q}_{h,k}(x)$ and the optimal Q-values $q_{\star,h}(x)$. The bound consists of four summands, three of which correspond to the three factors of the total regret discussed in Section 5.3.3. The fourth term, $\gamma\Delta_h\zeta(t)$, disappears from the regret bound because it is asymptotically dominated by the total bonus $\vartheta_h(t)$.

Lemma 5.7 * bound on $\bar{q}^k - q_{\star}$

Let Conditions 5.1, 5.2, 5.5, and 5.6 hold. Given constants $\delta_h > 0$ such that $\beta_h(t) \geq \gamma\Delta_h \cdot \zeta(t)$, where

This is a generalization of Lemma 3 of Rashid et al. [ibid.].

$$\Delta_h \triangleq cv_{\Delta,h+1} \cdot \sqrt{2 \ln \frac{2}{\delta_h}},$$

and c is a constant from Proposition 5.6, the following holds with probability at least $1 - \delta$, where $\delta \triangleq KX \cdot \sum_{h \in \mathbb{H}} \delta_h$:

$$\begin{aligned} 0 \leq [\bar{q}_{h,k} - q_{\star,h}](x) &\leq \alpha_{t,0} \cdot (q_{h,0} - v_{-,h}) \\ &\quad + \gamma \cdot \sum_{i=1}^t \alpha_{t,i} \cdot [\bar{v}_{h+1,k_i} - v_{\star,h+1}](s_{h+1,k_i}) \\ &\quad + \vartheta_h(t) + \gamma\Delta_h \cdot \zeta(t). \end{aligned} \quad (5.19)$$

Proof sketch. Let

$$Y_{t,i}(x) \triangleq \alpha_{t,i} \cdot [(\hat{\mathcal{T}}_{h,k_i} - \mathcal{T}_h)v_{\star,h+1}](x).$$

Note that $|Y_{t,i}(x)| \leq \alpha_{t,i}cv_{\Delta,h+1}$. Follow the argument of the proof of Lemma 4.3 of Jin et al. [2018], we apply the Azuma–Hoeffding inequality [McDiarmid, 1998, Theorem 3.13] to see that with probability at least $1 - \delta$

$$\left| \sum_{i=1}^t Y_{t,i}(x) \right| \leq \sqrt{2 \cdot \sum_{i=1}^t (\alpha_{t,i}cv_{\Delta,h+1})^2 \ln \frac{2}{\delta_h}} \leq \Delta_h \cdot \zeta(t), \quad (5.20)$$

for all $x \in \mathbb{X}$, $h \in \mathbb{H}$, and $k \in \mathbb{K}$. The right-hand side of inequality (5.19) follows from Lemma 5.5 and the fact that $q_{\star,h}(x) \geq v_{-,h}$. The left-hand side proof follows the existing proof of Rashid et al. [2020] using (5.20). QED

✎ A direct consequence of Lemma 5.7 is that for an arbitrary chosen bonus function we can lower-bound the probability that inequalities (5.19) hold (note that sometimes the bound can be zero though).

Corollary 5.8 * PAC-bounds under Proposition 5.6

Under Conditions 5.1, 5.2, 5.5, and 5.6, for an arbitrary chosen cumulative confidence bonus function $\beta_h(t)$, inequalities (5.19) hold with probability at least $1 - \delta$, where δ is given by (5.16) for c introduced in Proposition 5.6.

Proof. The special case $c = 0$ trivially follows from Condition 5.6 and Lemma 5.7. Otherwise δ can be obtained by solving for δ_h the following equation:

$$\beta_h(t) = c\gamma v_{\Delta,h+1} \cdot \zeta(t) \cdot \sqrt{2 \ln \frac{2}{\delta_h}}. \quad \text{QED}$$

5.4.3 Properties of the Total Regret

We are now ready to provide an upper bound on total regret of generalized optimistic Q-learning using the results of the previous sections. We start by introducing the following proposition, generalizing the arguments used in the literature [Jin et al., 2018; Rashid et al., 2020].

Proposition 5.9 * recursion on total regret bound

Denote

$$\begin{aligned} \psi_{h,k} &\triangleq [\bar{v}_{h,k} - v_{\pi_k,h}](s_{h,k}) & \text{and} \\ \xi_{h,k} &\triangleq [(\hat{\mathcal{T}}_{h,k} - \mathcal{T}_h)(\bar{v}_{h+1,k} - v_{\star,h+1})](x_{h,k}). \end{aligned}$$

Let Conditions 5.1–5.3, 5.5 and 5.6 hold. Using notation of Lemma 5.7, the following two statements hold with probability at least $1 - \delta$:

- the total regret R is upper-bounded by $R \leq \sum_{k=1}^K \psi_1^k$.
- for any $h \in \mathbb{H}$ and $k \in \mathbb{K}$, $\psi_{h,k}$ is upper-bounded by

$$\psi_{h,k} \leq \gamma \eta \psi_{h+1,k} + \Psi_{h,k}(t), \quad \text{where} \quad (5.21)$$

$$\Psi_{h,k}(t) \triangleq \alpha_{t,0}(q_{h,0} - v_{-,h}) + \vartheta_h(t) + \gamma(\Delta_h \zeta(t) + \xi_{h,k}). \quad (5.22)$$

Next, applying the bounds (5.21) iteratively on $h = 1, 2, \dots, H + 1$ and noticing that $\psi_{H+1}^k = 0$ by (5.3) and (5.1), we bound R .

Corollary 5.10 * recursive regret bound

Under Conditions 5.1–5.3, 5.5 and 5.6 with probability at least $1 - \delta$ the total regret is upper-bounded by

$$R \leq \sum_{k=1}^K \sum_{h=1}^H (\gamma\eta)^{h-1} \cdot \Psi_{h,k}(t), \quad (5.23)$$

where δ and $\Psi_{h,k}(t)$ are given by (5.16) and (5.22).

Finally, we are ready to prove Theorem 5.1.

Proof of Theorem 5.1. We study the right-hand side of inequality (5.23) by rewriting it as

$$\begin{aligned} R(K) &\leq \rho_K(\alpha_{t,0}(q_{h,0} - q_{-,h})) + \rho_K(\vartheta_h(t)) \\ &\quad + \gamma \cdot \rho_K(\Delta_h \cdot \zeta(t)) + \gamma \cdot \rho_K(\xi_{h,k}), \end{aligned}$$

where

$$\rho_K(g_{h,k}(t)) \triangleq \sum_{h=1}^H (\gamma\eta)^{h-1} \sum_{k=1}^K g_{h,k}(t).$$

For the first element $\rho_K(\alpha_{t,0}(q_{h,0} - q_{-,h}))$, by changing the summation order and using the fact that $q_{h,0} - q_{-,h} \leq \nu_{\Delta,h}$ we write

$$\rho_K(\alpha_{t,0}(q_{h,0} - q_{-,h})) \leq \sum_{k=1}^K \sum_{h=1}^H (\gamma\eta)^{h-1} \alpha_{t,0} \nu_{\Delta,h}.$$

In this sum $\alpha_{t,0} = \mathbb{I}_{\{t=0\}}$ by Lemma 5.3 and $\alpha_{0,0} = 1$. In this sum, $\mathbb{I}_{\{\#_{h,k}(x_{h,k})=0\}} \neq 0$ means that x has never been visited in step h before episode k , and the number of such state-action pairs is $\mathcal{O}(X)$ independent of K and H ; therefore, we have $\mathcal{O}(X)$ summands $(\gamma\eta)^{h-1} \nu_{\Delta,h}$, and each of them is $\mathcal{O}(\mu)$, so $\rho_K(\alpha_{t,0} \nu_{\Delta,h}) = \mathcal{O}(\mu X)$.

For $\rho_K(\xi_{h,k})$ we use the fact that $\{\xi_{h,k}\}_{k \in \mathbb{K}}$ is a martingale difference sequence [Jin et al., 2018, proof of Theorem 1]. Note that

$$\nu_{-,h+1} \leq \nu_{\star,h+1}(x) \leq \bar{\nu}_{h+1,k}(x) \leq \nu_{+,h+1},$$

therefore

$$[\bar{\nu}_{h+1,k} - \nu_{\star,h+1}](x) \in [0, \nu_{\Delta,h+1}].$$

Using these bounds, Proposition 5.6, an argument similar to the proof of Lemma 5.7, and Azuma–Hoeffding inequality, we see that with probability at least $1 - \delta$,

$$\left| \sum_{k=1}^K \xi_{h,k} \right| \leq \sqrt{2 \sum_{k=1}^K (c\nu_{+,h+1})^2 \cdot \ln \frac{2HX}{\delta}} = \mathcal{O}\left(c\nu_{+,h+1} \cdot \sqrt{K \ln \frac{HX}{\delta}}\right),$$

5.4 Proof of
Theorem 5.1

for all $h \in \mathbb{H}$ and $x \in \mathbb{X}$. Note that $\ln(HX)/\delta = \mathcal{O}(\iota)$, therefore

$$\rho_K(\xi_{h,k}) = \mathcal{O}\left(c \sum_{h=1}^H (\gamma\eta)^{h-1} v_{+,h+1} \cdot \sqrt{Ki}\right) = \mathcal{O}(c\mu\sqrt{Ki}) = \mathcal{O}(\mu E).$$

Finally, for the last two terms we notice that $\vartheta_h(t) \geq \gamma \Delta_h \zeta(t) \geq 0$ and thus $\vartheta_h(t)$ is the asymptotically dominant term, that is, $\Delta_h \zeta(t) = \mathcal{O}(\vartheta_h(t))$. We write

$$\rho_K(\vartheta_h(t)) = \sum_{h=1}^H (\gamma\eta)^{h-1} \cdot \sum_{k=1}^K \vartheta_h(\#_{h,k}(x_{h,k})).$$

First, we consider the inner sum

$$\Sigma_h^\vartheta \triangleq \sum_{k=1}^K \vartheta_h(\#_{h,k}(x_{h,k})).$$

Instead of summing in order of episodes $k \in \mathbb{K}$, we can sum the total bonuses $\vartheta_h(\#_{h,k}(x_{h,k}))$ separately for each state-action pair $x \in \mathbb{X}_h$ first, and add all visitations $n = 1, \dots, \#_h^K(x)$ of x in all episodes. This yields

$$\Sigma_h^\vartheta = \sum_{x \in \mathbb{X}_h} \sum_{n=1}^{\#_h^K(x)} \vartheta_h(n) \quad \text{where} \quad \sum_{x \in \mathbb{X}_h} \#_h^K(x) = K.$$

Because $\vartheta_h(t)$ is decreasing in t by Condition 5.6, Σ_h^ϑ is maximized when as many state-action pairs x are visited, which happens when $\#_h^K(x) = K/X$ for all $x \in \mathbb{X}$:

$$\Sigma_h^\vartheta \leq \sum_{x \in \mathbb{X}} \sum_{n=1}^{K/X} \vartheta_h(n) = X \cdot \sum_{n=1}^{K/X} \vartheta_h(n) = \mathcal{O}(v_{\uparrow,h} X \cdot \theta(K/X)),$$

where $\theta(t)$ is defined in Condition 5.7. Thus, $\rho_K(\vartheta_h(t)) = \mathcal{O}(\mu B)$.

Adding the three factors together, the bound (5.17) holds with probability at least $1 - 2\delta$. QED

5.5 DESIGNING A NEW OPTIMISTIC ALGORITHM

In this section, we apply Theorem 5.1 to prove efficiency of UCB-H^+ presented in Example 5.1. We show how the proposed generalized learning rate (5.14) satisfies the required condition, and how the bonus design is based on it. We only consider the case $\lambda > 0$, as inclusion of H is required to achieve sub-linear regret [Jin et al., 2018], but similar analysis can be performed for $\lambda = 0$, yielding worse bounds.

Conditions on the Learning Rate

First, we want to ensure that the generalized learning rate (5.14) satisfies the Conditions 5.1–5.4. Condition 5.1 holds trivially. We now show that so do the other ones.

5.5 Designing a New Optimistic Algorithm

Proposition 5.11 * auxiliary property of the exponent ω

$t^\omega + j \geq (t+j)^\omega$ for any $t \in \mathbb{N}_0$ and $j \in \mathbb{N}_0$.

Lemma 5.12 * Condition 5.2 holds for UCB-H^+

For the generalized learning rate given by (5.14), Condition 5.2 holds with

$$\zeta(t) = \sqrt{\frac{\lambda H + 1}{\lambda H + t^\omega}}.$$

Proof. Notice that

$$\sum_{i=1}^t (\alpha_{t,i})^2 \leq \max_{i \in \{1,2,\dots,t\}} \alpha_{t,i} \cdot \sum_{i=1}^t \alpha_{t,i},$$

which by Lemma 5.3 is equal to $\max_{i \in \{1,2,\dots,t\}} \alpha_{t,i}$. By definition,

$$\begin{aligned} \alpha_{t,i} &= \frac{\lambda H + 1}{\lambda H + i^\omega} \cdot \left(\frac{(i+1)^\omega - 1}{\lambda H + (i+1)^\omega} \cdot \frac{(i+2)^\omega - 1}{\lambda H + (i+2)^\omega} \cdots \frac{t^\omega - 1}{\lambda H + t^\omega} \right) \\ &= \frac{\lambda H + 1}{\lambda H + t^\omega} \cdot \left(\frac{(i+1)^\omega - 1}{\lambda H + i^\omega} \cdot \frac{(i+2)^\omega - 1}{\lambda H + (i+1)^\omega} \cdots \frac{t^\omega - 1}{\lambda H + (t-1)^\omega} \right). \end{aligned}$$

By Proposition 5.11 for $j = 1$ each fraction in the brackets is less than 1, so

$$\alpha_{t,i} \leq \frac{\lambda H + 1}{\lambda H + t^\omega} \triangleq \zeta^2(t). \quad \text{QED}$$

Proposition 5.13 * an auxiliary inequality

For any $m \geq k$,

See [Jin et al., 2018 Equation B.1].

$$\frac{m}{k} = 1 + \sum_{i=1}^{\infty} \prod_{j=1}^i \frac{m - k + j - 1}{m + j}.$$

Lemma 5.14 * Condition 5.3 holds for UCB-H^+

For the learning rate given by (5.14), Condition 5.3 holds with $\eta(H) = 1 + (\lambda H)^{-1}$ if $\lambda > 0$.

Proof. By Proposition 5.13 with $m = \lambda H + t^\omega$ and $k = \lambda H$,

$$\begin{aligned} \sum_{n=t}^k \alpha_{n,t} &\leq \sum_{n=t}^{\infty} \alpha_{n,t} = \alpha_t \cdot \left(1 + \sum_{i=1}^{\infty} \prod_{j=1}^i (1 - \alpha_{t+j}) \right) \\ &\leq \frac{\lambda H + 1}{\lambda H + t^\omega} \cdot \left(1 + \sum_{i=1}^{\infty} \prod_{j=1}^i \frac{t^\omega + j - 1}{\lambda H + t^\omega + j} \right) \end{aligned}$$

$$= \frac{\lambda H + 1}{\lambda H + t^\omega} \cdot \frac{\lambda H + t^\omega}{\lambda H} = 1 + \frac{1}{\lambda H},$$

5 *Generalized
Optimistic
Q-Learning*

where the second inequality holds by Proposition 5.11, because

$$1 - \alpha_{t+j} = \frac{(t+j)^\omega - 1}{\lambda H + (t+j)^\omega} \leq \frac{t^\omega + j - 1}{\lambda H + t^\omega + j}. \quad \text{QED}$$

Lemma 5.15 * Condition 5.4 holds for UCB-H^+

For the generalized learning rate Condition 5.4 holds with

$$\mu(H, v_\uparrow, \gamma) = H v_\uparrow$$

if $\gamma = 1$ and $v_\uparrow / (1 - \gamma)$ otherwise.

☞ We omit the proof of Lemma 5.15. It is straightforward as the sum in the definition (5.15) can easily be computed directly.

Conditions on the Bonuses

Lemmas 5.7 and 5.12 explain our choice of the bonuses, namely,

$$\beta_h(t) \triangleq c \gamma v_{\Delta, h+1} \sqrt{8 \cdot \frac{\lambda H + 1}{\lambda H + t^\omega} \cdot \ln \frac{2TX}{\delta}} \quad \text{and} \quad u_h(t) = 0 \quad (5.24)$$

for the constant c of Proposition 5.6. By Corollary 5.8, Lemma 5.7 holds with probability at least $1 - \delta$ for this cumulative bonus for any δ . Conditions 5.5 and 5.6 both hold trivially.

Lemma 5.16 * Condition 5.7 holds for UCB-H^+

For the bonuses given by (5.24), Condition 5.7 holds with

$$\theta(t) = \sqrt{H t^{2-\omega}}.$$

Proof. Note that

$$\sum_{n=1}^t (\lambda H + n^\omega)^{-1/2} \leq \sum_{n=1}^t n^{-\omega/2} = H_t^{(\omega/2)},$$

where $H_n^{(r)}$ denotes the generalized harmonic number of n of order r . By Euler–Maclaurin sum [Abramowitz, 1972, formula 3.6.28], for a given $r \neq 1$, $H_n^{(r)} = \zeta(r) + (1 - r)^{-1} n^{1-r} + o(n^{1-r}) = \mathcal{O}(n^{1-r})$. Thus

$$\sum_{n=1}^t \beta_h(n) = \mathcal{O}\left(\sqrt{H} \cdot \sum_{n=1}^t \frac{1}{\sqrt{\lambda H + n^\omega}}\right) = \mathcal{O}(\sqrt{H} \cdot t^{1-\omega/2}). \quad \text{QED}$$

Regret Bound

Combining the aforementioned results, we prove the efficiency of UCB-H^+ .

Theorem 5.17 * UCB-H⁺ is efficient in terms of regret

For any $\delta > 0$ with probability at least $1 - \delta$ the total regret of UCB-H⁺ with $\lambda > 0$ is bounded by $\mathcal{O}(\mu\sqrt{H^{\omega-1}T^{2-\omega}X^{\omega_l}})$, where the magnitude μ is given by Lemma 5.15.

5.6 Experiments

Proof. Using $\theta = \sqrt{Ht^{2-\omega_l}}$, we write the sum in Theorem 5.1 as

$$X + B + E = X + \sqrt{H^{\omega-1}T^{2-\omega}X^{\omega_l}} + c\sqrt{Kl}.$$

The last term is trivially dominated by the second one, so it can be omitted. Now we show that the first term is also dominated in the total regret bound.

Assume $T \leq \sqrt{H^{1+\omega}T^{2-\omega}X^{\omega_l}}$. The total regret is bounded by

$$\sum_{k=1}^K \psi_1^k \leq v_+ K \leq v_+ T/H = \mathcal{O}(v_+ \sqrt{H^{\omega-1}T^{2-\omega}X^{\omega_l}}),$$

which is dominated by the second term multiplied by μ . The opposite assumption implies that $T > H^{1+1/\omega}X^{1/\omega}$, and

$$\sqrt{H^{\omega}T^{2-\omega}X^{\omega_l}} > \sqrt{H^{\omega}(H^{1+1/\omega}X^{1/\omega})^{2-\omega}X^{\omega_l}} \geq \sqrt{H^3X^2} > HX.$$

In either case $\mu\sqrt{H^{\omega-1}T^{2-\omega}X^{\omega_l}}$ is the dominant term.

QED

5.6 EXPERIMENTS

To illustrate the performance of UCB-H⁺, we consider two problems, one stochastic and one deterministic. The latter is less interesting in the context of reinforcement learning, but allows us to alleviate the regret caused by the estimation error, highlighting the effect of optimism.

5.6.1 Equipment Replacement

We start with a classical problem known as *the automobile replacement problem* [Howard, 1960]. This problem is based on real data and is considered as a benchmark by different authors [Puterman, 1994; Even-Dar, Mannor, et al., 2006; Bellman and Dreyfus, 2016]. In the replacement problem, the agent operates an automobile, which can be in one of the forty states, from brand new one to ten years old, quantified quarterly. At the beginning of each quarter, the agent chooses to either keep the automobile, or to replace it with a different one, which can be in any of the forty available

Two years correspond to
 $H = 8$ steps.

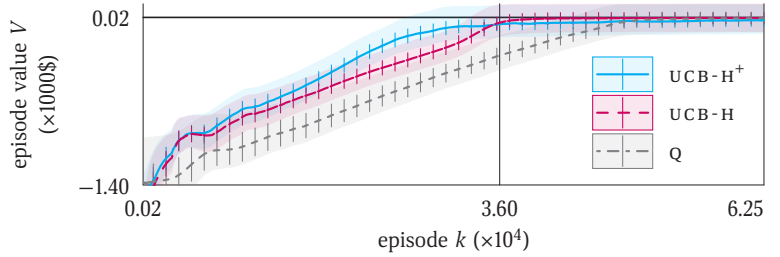
This state s_α means that
the car is brand new.

states. The detailed description of the problem can be found in the original paper by Howard [1960].

We consider a two-year plan, and $K = 62\,500$ episodes, each starting with state $s_\alpha = 1$. Therefore, the problem size is equal to $HX = 13\,120$, and the total duration of the learning is $T = 5 \times 10^5$ time steps. We assume no discounting $\gamma = 1$, and use the same values $\delta = c = 10^{-3}$ for UCB-based algorithms. As a baseline, we use regular Q-learning optimistically initialized with $q_{h,0}(x) = v_+$ with an exponentially decaying exploration rate $\varepsilon = 0.9999^{k-1}$ and the same learning rate $\alpha_t = (H+1)/(H+t)$ as UCB-H. For UCB-H⁺ we use $\omega = 0.8$ and $\lambda = 1$ as the learning rate parameters.

The experiment was repeated fifty times. The results are presented in Figure 5.2.

Figure 5.2: Replacement problem. UCB-H⁺ offers a 41% total regret improvement over Q-learning and 13% over UCB-H. The horizontal line at the top represents the optimal value v_\star , the vertical bars show 95%-confidence intervals on mean estimates, and the ribbons show the interquartile range. Data is smoothed using a moving average with a bandwidth of $0.05 \cdot K$.



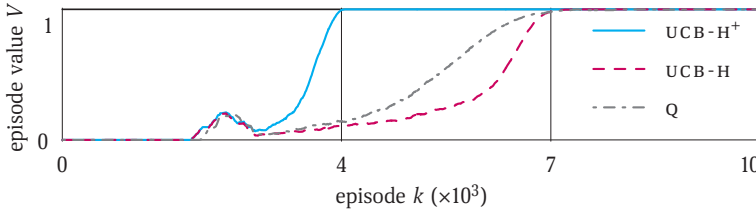
This experiment shows that the total regret of Q-learning, equal to the area between the line and the optimal line above it, is 1525 ± 2 thousand dollars on average. While the plot lines may seem close to each other, UCB-H was able to achieve a regret of 1037 ± 2 thousands, showing a 32% reduction over the naïve approach. Finally, UCB-H⁺ incurred a regret of 907 ± 3 thousand dollars, enjoying a reduction of 41% compared to Q-learning and 13% when compared to UCB-H. Interestingly, UCB-H⁺ has only a slightly higher variance, which we expect to increase as the exponent ω approaches 0.5 (with $\omega = 0.5$ preventing convergence by violating conditions (5.7)).

5.6.2 Frozen Lake

Our second experiment is the FrozenLake8x8-v0 problem [Brockman et al., 2016]. The agent navigates a grid world searching for a goal state. The world has holes, stepping into one terminates the current episode. All states give no rewards, except for the goal with a unit reward. We consider $K = 10^4$ episodes of up to $H = 16$ time steps. The problem size is $HX = 16 \times 64 \times 4 = 4096$, and the

total duration of the learning is $T = 1.6 \times 10^5$ time steps. Because this problem is simpler, we can use a faster decaying exploration rate $\varepsilon = 0.99^{k-1}$ for Q-learning. For UCB-H and UCB-H^+ we use zero constant c as per Remark 5.1. The rest of the parameters remain the same.

The results are presented in Figure 5.3. Interestingly, UCB-H suffered from the largest regret of 5503, while Q-learning and UCB-H^+ achieved the regret of approximately 4900 and 3144 respectively. UCB-H^+ offers a 43% improvement over UCB-H . As mentioned earlier, this problem has no stochasticity in transitions, and thus the last term of the regret is zero. Moreover, all algorithms use the same initialization, therefore, the only reasons for the performance difference is the choice of the learning rate and the optimism representation.



5.7 Conclusion

Figure 5.3: Frozen lake. UCB-H^+ offers a 43% total regret improvement over UCB-H .

5.7 CONCLUSION

This chapter presents generalized optimistic Q-learning, a novel framework for optimistic model-free reinforcement learning that incorporates many existing methods, such as Q-learning, UCB-H , and OPIQ . We showed that under some mild conditions the total regret of optimistic model-free methods is driven by three distinct terms multiplied by the magnitude of the problem:

- the size of the state-action space,
- the total effect of the bonuses, and
- the total effect of the estimation error.

To the extent of our knowledge, this is the first study of RL performance that does not rely on a particular form of the learning rate. This high level of abstraction facilitates transfer of our results to new algorithms within the generalized optimistic Q-learning framework. As an example, we present one such algorithm, UCB-H^+ , prove its efficiency in terms of regret, and illustrate

its performance in experiments. Our analysis shows that the regret is driven by the bonuses and the learning rate, therefore, their choice is a promising direction for the design of more efficient optimistic RL algorithms.

Future work includes further relaxations of the conditions used, and extensions of generalized optimistic Q-learning to other settings such as infinite-horizon non-episodic learning, deep reinforcement learning, and models with continuous state and/or action space. The algorithm UCB-H^+ can be extended to the continuous setting as well. One of the possible ways to do this is to employ deep Q-networks and pseudo-visitation counters similarly to [Rashid et al., 2020].

6

Reinforcement Learning for Active Wake Control

*For the things we have to learn before we
can do them, we learn by doing them.*

— Aristotle, *The Nicomachean
Ethics* II · 1

Translated by W. D. Ross

WIND FARMS suffer from so-called wake effects: when turbines are located in the wind shadows of other turbines, their power output is substantially reduced. These losses can be partially mitigated via actively changing the yaw from the individually optimal direction. Most existing wake control techniques have two major limitations: they use simplified wake models to optimize the control strategy, and they assume that the atmospheric conditions remain stable. In this chapter, we address these limitations by applying reinforcement learning. Reinforcement learning forgoes the wake model entirely and learns an optimal control strategy based on the observed atmospheric conditions and a reward signal, in this case the power output of the farm. It also accounts for random transitions in the observations, such as turbulent fluctuations in the wind. To evaluate the benefits of reinforcement learning for active wake control, we implement a simulator based on the state-of-the-art `FLORIS` model in the Gym format, making it readily available to the RL community. Next, we propose three different state-action representations of the active wake control problem and investigate their effect on the performance of RL-based wake control. Finally, we compare reinforcement learning to a state-of-the-art wake control strategy based on `FLORIS` and show that reinforcement learning is less sensitive to changes in unobservable data.

6.1 INTRODUCTION

In this chapter, we investigate the benefits of using reinforcement learning for active wake control in dynamic atmospheric conditions. To do so, we implement a dynamic wind farm simulator. Our simulator uses `FLORIS` for each stable state, and supports arbitrary transition models between such states, defined by the user. Its design is driven by real-life wind farm operation: it supports varying angular velocities of the turbines and imitates installations not seen in other studies, such as meteorological masts and nacelle-mounted lidar systems. A detailed list of differences with the existing work on reinforcement learning for active wake control is given in Table 6.1. We implement our simulator in the Gym format [Brockman et al., 2016], which is the industry standard for representing RL problems, so as to facilitate future research in active wake control from the RL community.

Additionally, we discuss two alternative representations of

This chapter is based on the article published in the Proceedings of the Twenty-First International Conference on Autonomous Agents and Multiagent Systems [Neustroev, Andringa, et al., 2022a].

Minor changes were made to the text compared to the published version: some of the preliminary results were moved to Chapter 1 and minor text edits were made to improve readability.

The active wake control problem is described in Section 1.3.3, p. 8.

Lidar stands for *light detection and ranging*. It is sometimes spelled as “LiDAR.”

the control actions in this problem, not used in other studies. We compare the performances of state-of-the-art reinforcement learning algorithms for each representation. Our experiments show that action encoding has a significant impact on the performance of reinforcement learning methods, with one of the alternatives being preferable.

Finally, we demonstrate the benefits of reinforcement learning compared to model-based optimization. Having no explicit model, it is more robust to changes in unobserved data and to observation noise, which is especially important for real-life applications.

6.2 PRELIMINARIES

We begin with providing background information both on state-of-the-art model-based active wake control and on the principles of reinforcement learning itself.

6.2.1 *Steady-State Wind Models*

Flow Redirection and Induction in Steady-State (FLORIS) wake modeling framework [NREL, 2021] includes many of the state-of-the-art steady-state wake models, and various tools for analysis and optimization of wind farm layout and operation. It is fully open-source, computationally cheap, and implemented in Python, a popular language among RL researchers. FLORIS is maintained, updated, studied and put into practice by a large community. It was originally based on works by Jensen [1983] and Jiménez et al. [2010], but it is being improved frequently, in particular by adding newly developed wake models. For a more detailed overview of FLORIS, the reader is referred to the work by Annoni et al. [2018].

Various studies highlight applicability of FLORIS. For example, Gebraad et al. [2016] apply FLORIS-based control strategy in a high-fidelity computational fluid dynamics simulator. Wind tunnel tests were performed by Schreiber et al. [2017]. A field trial on a commercial offshore wind farm is presented by Fleming, Annoni, et al. [2017]. These and other applications allow us to consider FLORIS as the state of the art in both wake modeling and model-based active wake control.

FLORIS offers various analytical models to compute the wakes, but it does not explicitly model rapidly changing conditions due to turbulence and other small-scale atmospheric phenomena. Higher

	Verstraeten et al. [2020]	Stanfel et al. [2021]	Dong et al. [2021]	this chapter
simulator	software OpenAI Gym API elements	FLORIS yes wind & turbines	FLORIS no wind & turbines	SOWFA no wind & turbines
state	space per-turbine observ. per-mast observ. farm-wide observ. atm. measurements measurment. sources partial observability noisy observations	continuous yaw & power — — wind speed turbines no no	discrete yaw — — wind speed & dir. turbines yes yes	continuous yaw & lidar measurem. atm. measurements atm. measurements multiple, see Table 6.2 turbines, MM, extern. yes yes
action	space representation	discrete, $\{-1, 0, 1\}^n$ yaw-based	discrete, $\{-1, 1\}^n$ yaw-based	continuous, $[-1, 1]^n$ yaw & 2 more
reward	based on	power deficit	power increase	average power total power
transition	time-varying data stochastic model	— —	T1, wind speed Gaussian noise	internal simul. data —
learning	algorithm is deep?	GPRL no	Q-learning no	DDPG yes

6.2 Preliminaries

Table 6.1: Comparison of the existing studies of reinforcement learning for active wake control. T1 and MM stand for turbulence intensity and meteorological masts respectively.

fidelity tools based on computational fluid dynamics such as large eddy simulation (LES) can be used for this. Examples of LES include *Simulator for On/Off-Shore Wind Farm Applications* (SOWFA) [Fleming, Gebraad, et al., 2013], *Dutch Atmospheric Large-Eddy Simulation* (DALES) [Heus et al., 2010], and *GPU-Resident Atmospheric Simulation Platform* (GRASP) [Gilbert et al., 2020].

Unfortunately, LES require substantial computational power. To apply their learning method, Dong et al. [2021] performed 90 simulations, each of which took approximately 44 hours on 256 CPU cores for a total of 921 600 core-hours. Each simulation consisted of just 1000 seconds of simulated time. This computational power is far beyond the reach of an average researcher. Moreover, not all of the LES models are open source, further limiting their applicability. As a result, steady-state computationally efficient simulators like FLORIS are more commonly used. Interestingly, even though LES are dynamic, optimization is often done per a steady incoming wind direction [Gebraad et al., 2016; Dong et al., 2021], which is another argument for using steady-state simulator such as FLORIS as a simpler alternative to LES.

Figure 1.4 on p. 8 shows a simulation in FLORIS with the default parameters and two turbines positioned at a distance of six rotor diameters ($6 \cdot D$).

It is important to distinguish FLORIS the simulator and FLORIS the controller. A simulation in FLORIS is based on turbine specifications, such as the amount of power they produce at different wind speeds, turbine locations in the wind farm, and a set of atmospheric conditions presented in Table 6.2.

These atmospheric conditions are used together with one of the wake models to predict steady-state wake locations and the wind flow throughout the farm. Based on this information, the total power output of the farm is represented as a function of the yaws [Rott et al., 2018]. This function is then maximized by an optimizer to improve the yaws.

FLORIS considers atmospheric conditions as steady, therefore an atmosphere in FLORIS can be represented by a vector of num-

Table 6.2: Atmospheric conditions in FLORIS

measurement	default value	description
wind speed	8 m/s	
direction	270°	from north clockwise
shear	0.12 s^{-1}	change of speed with height
veer	$0^\circ/\text{m}$	change of direction with height
turbulence intensity	0.06	coefficient of variation of wind speed
air density	1.225 kg/m^3	the default is at 101 325 Pa and 15°C

bers, like the second column of Table 6.2. Since atmospheric conditions change over time, one of the possible ways to use FLORIS for control in a dynamic system is to use long-time averages. Another approach is to reinitialize it each time new conditions are observed, using either historical data or a simulated multivariate stochastic process.

6.2.2 Deep Reinforcement Learning

Among the various RL algorithms, we are interested in so-called actor-critic methods. They use deep neural networks to concurrently learn a policy that prescribes actions to take in each state, and state-action values that tell how good the actions chosen by the policy are.

DDPG updates both the actor and the critic using gradient descent. Its successor TD3 adds a few tricks to stabilize the learning process. Namely, it uses two critics (hence twin learning) and updates the policy less frequently than DDPG (delayed). Additionally it slightly perturbs the actions to avoid a phenomenon known as catastrophic forgetting which may happen in deep neural networks when they stop receiving novel inputs.

SAC uses similar tricks, but has a non-deterministic policy with entropy regularization. The entropy coefficient controls how much exploration the policy does, and is usually automatically tuned, making SAC more adaptive. Since its conception, this algorithm has been one of the best performing deep-RL methods.

6.2.3 Reinforcement Learning for Active Wake Control

Table 6.1 provides an overview of RL methods applied to active wake control problems.

The works of Verstraeten et al. [2020] and Stanfel et al. [2021] both use discrete actions with ± 1 standing for counterclockwise and clockwise rotations at a fixed angular velocity. Instead of directly using the power output of the wind farm as the reward signal, both use some form of reward shaping to construct a different reward signal. Both of these methods use non-deep reinforcement learning. Both methods use steady-state simulations, with learning done separately per wind speed and direction. The optimal action is chosen based on current yaw. Therefore, in both cases transitions between different atmospheric conditions are not modelled, but transitions between yaw are taken into account.

6.2 Preliminaries

For an overview of RL methods, see Section 1.4.2, p. 12.

The neural network that produces a policy is called the *actor*, because it prescribes how to act. The neural network that produces the values is called the *critic* because it tells how good the outputs of the actor are.

Actor-critic approach is similar to the dual formulation of MDPs, where the primal program (C1-P) (p. 57) produces a policy and the dual program (C1-D) evaluates the states.

Instead of neural networks, Verstraeten et al. [2020] use Gaussian-processes reinforcement learning (GPRL) for Q-value approximation. This is paired with knowledge transfer between similarly positioned turbines to learn the optimal control strategy. This is the only article that uses multi-agent reinforcement learning, showing its high efficiency. Stanfel et al. [2021] use simple Q-learning, but combine it with domain knowledge. For example, they apply Gaussian blur to the state-action value function, so that similar states do not have vastly different values.

Research on *deep* reinforcement learning for active wake control in *non-stationary* environments remains limited. To the extent of our knowledge, the only such application of deep reinforcement learning was by Dong et al. [2021]. They use an offline version of DDPG to learn from examples generated in a high-fidelity (LES) simulator, and then use the simulator to evaluate the resulting policy. Even though the wind speed is steady, LES accounts for fluctuations in the atmosphere caused by turbulence, creating stochastic transitions. As mentioned above, this required substantial computational power, but the results are sufficiently promising to further explore the use of deep reinforcement learning for wake control. To do so, in this chapter we analyze alternative action representations, two different deep-RL algorithms, and the performance with respect to changes in unobserved data and to observation noise.

6.3 ACTIVE WAKE CONTROL AS A REINFORCEMENT LEARNING PROBLEM

To be able to apply RL algorithms to the active wake control problem, we need to define it in terms of time steps, states, actions, rewards and one-step transitions. While this has been done in previous studies, the resulting formulations are usually highly abstract and do not reflect the realities of wind farm operation. For example, atmospheric measurements are captured directly at the turbine locations, or are assumed to be uniform across the wind farm. In practice, various measurement tools positioned throughout the farm can be used to provide atmospheric information, such as free-standing meteorological masts or lidar systems. We aim for a more realistic problem formulation that reflects this.

As mentioned earlier, we treat each time step as having a steady-state atmospheric conditions. At the end of a time step,

the atmospheric conditions change and the control chosen by the agent is executed, causing a transition to a new state, which is again assumed to be steady. This process is repeated for a predefined number of steps T . In our definition of the problem, we allow arbitrary chosen (but equal) time intervals Δt between observations and control events, typically a few seconds.

6.3 Active Wake Control as a Reinforcement Learning Problem

6.3.1 State Space

In reinforcement learning, states describe the current environment as observed by the agent and contain all the information used by the agent to choose an action. At any single point of time, the wind farm can be assumed steady and thus can be represented by a FLORIS simulation. Nevertheless, not all of the simulation data is observable by the agent. It is thus important to consider what kind of information is available to the wind farm controller and include only this information in the state description.

First, we assume that the current yaw angles γ_i of all of the turbines are known, otherwise controlling them may prove difficult. Additionally, a FLORIS simulation allows to measure atmospheric conditions presented in Table 6.2, and the control strategy may depend on these.

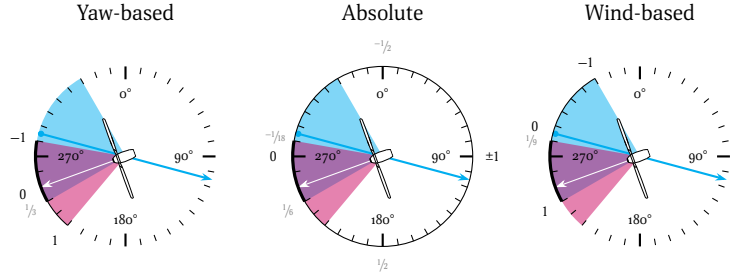
In the current implementation of FLORIS, wind speed, direction, and turbulence intensity vary across the wind farm and therefore should be measured at specific points in space. In a real-world wind farm such measurements come from meteorological masts or from sensors on the turbines. For example, these can be nacelle-mounted lidar systems. They are installed behind the turbine rotor and can measure the wind in front of it at a distance of 10–300 meters [M. Smith et al., 2014; Bot, 2016].

In contrast, wind shear, veer, and atmospheric density remain constant across the wind farm and can be defined for the wind farm as a whole. In real-life systems, this type of measurement exists as well. For example, some data may come from an external source, such as a meteorological forecast.

When creating a simulation, the user can specify:

- the positions of meteorological masts and the measurements collected there;
- which of the turbines are equipped with lidars and what is measured by these lidars;
- a list of per-farm measurements from an external data source.

Figure 6.1: Action representations. The blue arrow shows the wind direction (coming from 285°). The white arrow indicates the turbine orientation (250°). The blue sector (top) shows the desired yaw range ($\pm 45^\circ$ from the wind), and the purple sector (bottom) shows the reachable yaws for an angular velocity of $30^\circ/\text{step}$. The overlap shows the reachable yaws, which are the same in all cases.



At runtime, the simulator registers the data according to this specification, arranges them into a numeric state vector $s \in \mathbb{R}^k$ and returns this vector to the user. For example, if a simulation includes three turbines that register their yaws γ_i , $i \in \{0, 1, 2\}$ and two masts that register the wind speed M and direction ϕ at their locations, the state is $s = [\gamma_0, \gamma_1, \gamma_2, M_0, \phi_0, M_1, \phi_1]^\top \in \mathbb{R}^7$.

It is common to normalize states in reinforcement learning. We define ranges of possible values for each measurement and include an option to rescale each observation to an interval between zero and one.

Finally, to account for imperfections in the measuring equipment (including yaw measurements), we allow state vector perturbations by a zero-mean Gaussian noise. This noise is independently drawn at each time step with a scale parameter defined by the user for each of the observed variables from a given list. The normalized observations are then clamped between zero and one. If the observations are not normalized, the noise is rescaled accordingly for each observed measurement.

6.3.2 Action Space

Each action $a = [a_0, a_1, \dots, a_{n-1}]^\top \in [-1, 1]^n$ is a vector of length n , where n is the number of turbines. Each coordinate a_i encodes a yaw change of the i -th turbine. In FLORIS, when a turbine is rotated counterclockwise relative to the incoming wind, its yaw is positive, otherwise it is negative. We use the same convention.

The way that the yaw of the i -th turbine changes based on the coordinate a_i can be different. We consider three possible interpretations of actions, visualized in Figure 6.1.

Yaw-based action representation

The action tells how much the turbine yaw should change with respect to the current position. Zero action means that the tur-

bine should remain still, and ± 1 correspond to maximum possible rotations, that is $\pm \omega_+$ degrees from the current position, where ω_+ is the maximum angular velocity of the turbine in degrees per time step. This is the representation used in the previous research on reinforcement learning for active wake control. In this representation, if the current yaw angle of the i -th turbine is γ_i , the new yaw γ'_i will be $\gamma'_i = \gamma_i + a_i \cdot \omega_+$.

6.3 Active Wake Control as a Reinforcement Learning Problem

Absolute angle representation

The action tells what the optimal yaw should be relative to some static direction. For example, the most prevalent wind direction can be used. In Figure 6.1 it is west. In this case, 0.5 corresponds to south, -1 and +1 to east, and -0.5 to north. If this desired new yaw is outside of the operational zone of the turbine, it will turn as far towards it as it can, either clockwise or counterclockwise, depending on which direction is closer. For example, if the static direction β is 270° as in Figure 6.1, the next step yaw will be $\gamma'_i = \beta - a_i \cdot 180^\circ$.

Wind-based action representation

The action is represented as the optimal yaw relative to the current wind direction ϕ measured at the turbine's location. The actions of ± 1 correspond to the maximum (desired) yaw relative to the wind. The new yaw is computed as $\gamma'_i = \phi + \frac{1}{2}(a_i + 1) \cdot (\gamma_+ - \gamma_-) + \gamma_-$.

After the new yaw angles γ'_i are calculated, they are adjusted to satisfy two constraints.

First, turbines cannot rotate faster than their maximum angular velocity ω_+ . This constraint is based on physical limitations and should always be satisfied. In Figure 6.1, the possible yaws in the next time step are shown in purple. If the agent selects the new yaw γ'_i to be outside of the interval $[\gamma_i - \omega_+, \gamma_i + \omega_+]$, it is clipped to fit inside this interval. For example, if in the absolute representation the agent chooses any action a_i smaller than $-1/8$, it will result in the same new turbine orientation of 280° .

Second, the turbine's yaw relative to the wind should not be too large. This is because its power output is proportional to the cosine of the yaw, and drops fast as it turns away from the wind. To ensure a reasonable operational range, we define minimum γ_- and maximum γ_+ yaws. This constraint is shown in blue in Figure 6.1. If the turbine is within the desired yaw limits and attempts to leave them, it will stop. The new yaw is clipped to

satisfy this constraint. In rare cases the turbine may end up outside of the desired yaw range, for example due to a sudden change in the wind direction. In the notation of Figure 6.1, this will result in blue and purple sectors not overlapping. In this case, the turbine should attempt to return as fast as possible to the operational yaw range (the blue sector), but may stay outside of it temporarily. No matter which action is chosen by the agent, the turbine will perform the same rotation: the one that minimizes the angle with the wind direction.

After these constraints are applied, the range of the possible next-step yaws is the same between the three representations. For example, in Figure 6.1, the new yaw can only be between 240° and 280° . The only thing that differs between the three representations is how the new yaws are computed based on the action vector.

6.3.3 Rewards

At each time step, FLORIS simulator calculates the total power output P of the wind farm in watts, which is the sum $P = \sum_{i=1}^n P_i$ of power outputs P_i of the individual turbines,

$$P_i = \frac{1}{2} \rho \cdot A_i \cdot M_i^3 \cdot 4ax_i(M_i) \cdot (1 - ax_i(M_i))^2 \cdot \eta \cdot \cos^{p_p} \gamma_i.$$

Here ρ is the air density and M_i is the wind speed at the turbine, both of which are atmospheric conditions that may be included into the state vector and γ_i is the yaw of the i -th turbine, which depends on the i -th coordinate of the action vector. This equation shows that the reward is dependent both on the state and the action in a non-linear manner.

6.3.4 Transitions

When the environment transitions to a new steady state, two things change in the FLORIS simulator. First, the yaws are adjusted according to the action chosen by the agent.

Next, the atmospheric conditions change, resulting in changes in both the wind flow in the simulation, and in the atmospheric measurements registered at the next time step. The most obvious approach is to find a dataset of atmospheric conditions at the desired granularity and use it to generate transitions.

To create a simple yet realistic wind simulation, we looked at a publicly available dataset from the *Hollandse Kust Noord (site B)* (HKNB) wind farm zone in the Netherlands [RVO, 2019]. This

For a more detailed description of the remaining parameters and the function $ax_i(M_i)$, called the *axial induction factor*, see the paper by Gebraad et al. [2016].

data	parameter		
	$\log \tau$	$\log M$	ϕ_r
mean	-2.2×10^0	2.3×10^0	
drift	$\log \tau$	2.5×10^{-3}	5.5×10^{-4}
	$\log M$	-2.1×10^{-5}	4.8×10^{-5}
	ϕ_r	3.1×10^{-3}	-3.6×10^{-3}
diffusion	$\log \tau$	1.3×10^{-2}	-2.1×10^{-4}
	$\log M$		2.2×10^{-3}
	ϕ_r		1.6×10^{-1}

Table 6.3: Estimated parameters of the wind process. Empty cells correspond to zeros. τ , M , and ϕ_r stand for turbulence intensity, wind speed, and relative wind direction respectively.

dataset was chosen because it includes all atmospheric parameters used by FLORIS. Furthermore, it is a practically relevant case, as active wake control will be investigated for the wind farm at this location [Crosswind, 2021].

The data is measured at ten-minute intervals, which is typical for such datasets. Unfortunately, this means that it cannot be used directly in turbine control experiments, as control is typically more frequent. To address this issue, we fit a continuous-time stochastic process to the data. This allows us to use one time step for estimation and a different one for simulation.

We use the multivariate Ornstein–Uhlenbeck process [Vatutipong and Phewchean, 2019]. It is a mean-reverting process, meaning that its parameters tend to return to long-term average values, for example, single prevalent direction or mean wind speed. Moreover, many commonly used stochastic processes can be seen as particular cases of the multivariate Ornstein–Uhlenbeck process [Meucci, 2009]. For these reasons, Ornstein–Uhlenbeck processes are used in wind modeling [Arenas-López and Badaoui, 2020; Obukhov et al., 2021]. Additionally, by increasing the mean-reversion coefficient of wind direction, we can force it to stay stable, emulating a popular experimental setup with a wind tunnel.

Formally, the multivariate Ornstein–Uhlenbeck process is defined by the following stochastic differential equation

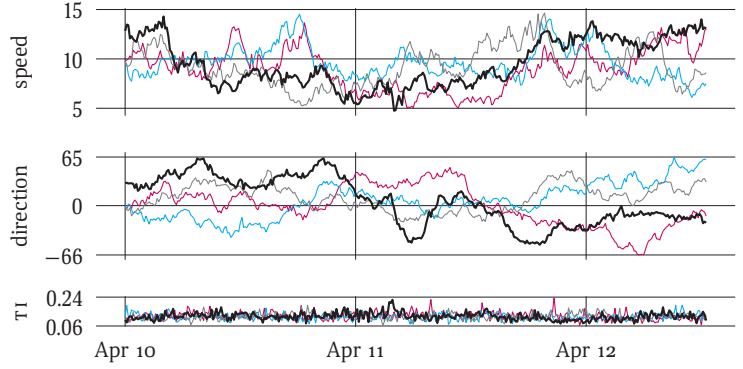
$$dy = \Theta(\mathbf{m} - \mathbf{y}) dt + \mathbf{S} d\mathbf{W}_t.$$

In simpler terms, this process can be described as follows. \mathbf{m} is a vector of mean values to which the process tends to revert. Θ is the drift matrix. It determines the speed of reversion to the means \mathbf{m} . \mathbf{W}_t is a multivariate Wiener process that adds random noise. \mathbf{S} is the diffusion matrix that determines the noise covariance matrix

$\Sigma = \mathbf{S}\mathbf{S}^\top$. A procedure described by Meucci [2005] can be used to estimate the parameters of this process. When the parameters are known, a simulation procedure for arbitrary chosen time steps is provided by Vatiwutipong and Phewchean [2019].

We then estimated a process for three atmospheric measurements: turbulence intensity τ , wind speed M , and wind direction ϕ . Because turbulence intensity and wind speed cannot be negative, we applied a logarithmic transformation. For the wind direction, we applied a rotation so that the mean m_{ϕ_r} of the rotated process ϕ_r is equal to zero. This transformation means that the wind direction is measured relative to some prevalent direction, which becomes easier to set in the simulation. Figure 6.2 shows the wind data used in estimation and three simulated paths.

Figure 6.2: Sample paths of the simulated atmospheric conditions. Black line shows historical data used in estimation.



After the data transformation, we fitted a multivariate Ornstein–Uhlenbeck process for $\mathbf{y} = [\log \tau, \log M, \phi_r]^\top$. The estimation procedure requires data points at equal time intervals. To achieve this, we cropped the HKNB dataset to the first missing entry. The resulting wind parameters are presented in Table 6.3. For wind shear and veer, we used mean values in the dataset, 0.0094 s^{-1} and $-0.025^\circ/\text{m}$ respectively.

6.3.5 Gym Implementation

OpenAI Gym [Brockman et al., 2016] is an open-source Python library of benchmark problems for reinforcement learning. Each problem in Gym is represented by an environment which provides a unified API for RL algorithms to communicate with, making it the field standard for RL problems. For this reason, we implement our simulator as a Gym environment to make it eas-

ier for other RL researchers to use [Neustroev, Andringa, et al., 2022b]. This environment supports all of the elements of state and action representation mentioned in this section, as well as an arbitrary transition function. We provide four basic variants of the transition model, but users can define their own, more sophisticated transition models, for example, using time-varying multivariate Ornstein–Uhlenbeck processes, or entirely different stochastic models of the wind. If the wind process is not specified, the environment uses steady wind from FLORIS.

For the multivariate Ornstein–Uhlenbeck process, the user can provide a list of k measurement names, whether the logarithmic transformation needs to be taken for each of the measurements, the mean vector of length k , and two $k \times k$ matrices of drift and diffusion. For the wind direction, we additionally use the principal wind direction relative to which it has been measured. After the direction data is generated, it is rotated by that angle. This direction is 270° by default, meaning that the wind comes primarily from the west. This is a common practice in wake control experiments.

6.4 Experiments

6.4 EXPERIMENTS

Using our Gym environment, we performed two experiments where we compare reinforcement learning to two control strategies. The baseline strategy is to ignore the wake effects, turning the turbines to face the incoming wind. The second strategy is given by the FLORIS optimizer. It optimizes the yaws numerically based on the wind flow model in the simulation. In contrast, reinforcement learning needs no such model.

6.4.1 Action Representations

In this experiment, we test the effect of action representation on the performance of two state-of-the-art RL algorithms: TD3 and SAC. We omit DDPG even though it is used by Dong et al., 2021 because TD3 is its direct successor. The hyperparameters used for each method are available in Appendix B. We use a setup where one meteorological mast and three turbines are positioned in a line. This single-line layout is commonly used in evaluation of wake control strategies in a wind tunnel, as it represents the worst possible scenario because of the many wake interactions.

We use a multivariate Ornstein–Uhlenbeck process to simulate the wind as described in Section 6.3.4, but with a single adjustment: we increase the mean-reversion rate of wind direction by changing the drift coefficient of the wind θ_{ϕ_r, ϕ_r} from -8.3×10^{-7} to 10^{-2} . This forces wind direction to stay within $270^\circ \pm 5^\circ$ but still change with time. Other parameters remained as listed in Table 6.3. The dependencies of turbulence intensity and wind speed on the wind direction are unchanged.

The state space is a vector of length five that includes the yaw angles and two measurements from the meteorological mast: wind speed and direction. While turbulence intensity changes over time, it is not observed by the wind farm operator. For FLORIS, we used turbulence intensity of 0.12 and wind veer and shear presented in Table 6.3. We allowed the turbines to turn at the maximum angular velocity of $1^\circ/\text{sec}$. Further, we used the parameters from the NREL 5 MW reference turbines. These and other FLORIS parameters are taken from the default multi-zone wake model.

For each RL method, we trained ten agents on different random seeds. To evaluate the performance of the learned policies, we separated training from evaluation as follows. For training, we simulated a week of wind farm operation with time intervals of ten seconds. The evaluation of the momentarily learned policy of each agent is done every twelve hours of simulated time (that is, fourteen times) in five randomly generated environments. Each such evaluation lasts for eight hours of simulated time (2880 steps), during which the total reward is compared against two benchmark strategies: a *baseline* in which each turbine faces the incoming wind, and a model-based control strategy offered by FLORIS.

Because different evaluation environments contain different atmospheric conditions, the total power output of these benchmark strategies changes across environments. To compare, we normalize the results so that in each evaluation the total reward of the baseline policy is equal to zero, and of FLORIS to one. In this experiment, FLORIS has access to the exact simulation model sans turbulence intensity, justifying how we use it to indicate a 100% performance.

The rescaled results are presented in Figure 6.3. While the yaw-based representation may seem to be the most intuitive one, it performs poorly. Because either positive or negative actions are chosen too often, it often fluctuates between the extreme yaws, leading to a drift in the turbine yaw.

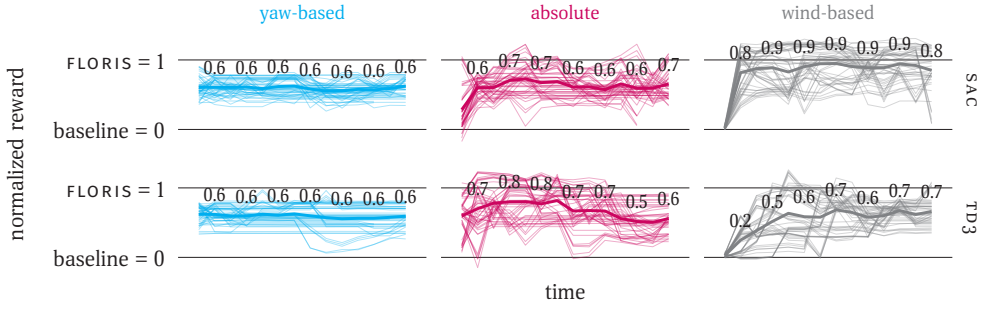


Figure 6.3:
FLORIS-normalized
reward of RL agents for
different action
representations over
one month of simulated
time for SAC (top) and
TD3 (bottom). Thin and
thick lines represent
individual evaluations
and means respectively.

To better understand this effect, consider a situation where the wind is steady. In the other two representations, the optimal action is the same for any current yaw. In the yaw-based representation, however, this is not the case, and if the same action is performed at all time steps, the turbine keeps turning either clockwise or counterclockwise until it reaches the end of the desired yaw sector. Therefore different actions need to be learned for different states. Assuming that learning constant values is easier for a deep neural network, other representations will lead to better performance.

The wind-based representation is the best performing one. To understand why, consider the baseline strategy of always facing the wind. For any down-wind turbine this is the optimal strategy. In the wind-based representation, this strategy is yaw-independent, making it easy to learn. In other representations, the optimal action depends on the incoming wind direction. These results show how the performance of RL methods depends on action representation in the active wake control problem.

Of the two RL agents, SAC performs better than TD3, and learns almost a perfect strategy in the given timeframe. Interestingly, SAC sometimes outperforms FLORIS. This is possible because of the interactions between wind speed, direction and turbulence intensity. While the latter is not observed, its changes can be derived (up to a noise parameter) from other wind data. We speculate that in some of the experiments SAC performed so well because it was able to find a better turbulence representation than the average turbulence intensity known to FLORIS.

Table 6.4: Performance improvement in percent over the baseline in the noisy observations benchmark. For SAC, the final learned strategy is used.

noise, σ	FLORIS		SAC	
	mean	95% conf. int.	mean	95% conf. int.
0.01	9.54	9.01 – 10.11	8.46	7.04 – 9.65
0.03	1.24	1.04 – 1.42	5.15	0.96 – 10.04
0.05	0.42	0.32 – 0.50	9.53	8.07 – 11.02
0.07	0.23	0.18 – 0.29	7.35	5.85 – 9.01

6.4.2 Noisy Observations

Of the two benchmarks in the previous experiment, SAC outperformed TD3, but FLORIS offered a better control strategy most of the time. This is because it has a perfect model of the environment, which is not true in practical applications. In this experiment, we compare reinforcement learning to FLORIS in the presence of imperfect observations.

To illustrate the capabilities of our simulation environment, we slightly adjust the experimental setup of the previous section. First, we remove the mast. Instead, we use per-turbine measurements of wind speed and direction, and a farm-wide measurement of turbulence intensity for both FLORIS-based controller and SAC. Next, we move the second and third turbines by $\frac{1}{4} \cdot D$ south and north respectively. This makes the problem harder, as it no longer has two symmetric solutions. Finally, in this experiment the time step is one second instead of ten for a more realistic control.

To generate faulty observations, we use four different levels of noise: $\sigma \in \{0.01, 0.03, 0.05, 0.07\}$, that is, after the observations are normalized between zero and one, we perturb them with a Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma)$. Only the wind measurements (speed, direction, turbulence intensity) are perturbed, and the yaws are unchanged. We train five agents for one day of simulated time (86 400 steps). The evaluations are performed every two hours of simulated time (7200 steps) and last for thirty minutes of simulated time (1800 steps). Each evaluation uses five different environments.

The results of this experiment are presented in Figure 6.4 and Table 6.4. FLORIS-based optimization struggles to outperform the baseline strategy as the noise scale grows, dropping from 9.5% improvement over the baseline to just 0.2%. While SAC also suffers from the noise in the observations, its performance improvement is between 8.5% and 7.4%, giving a statistically significant improvement over FLORIS-based control in noisy environments.

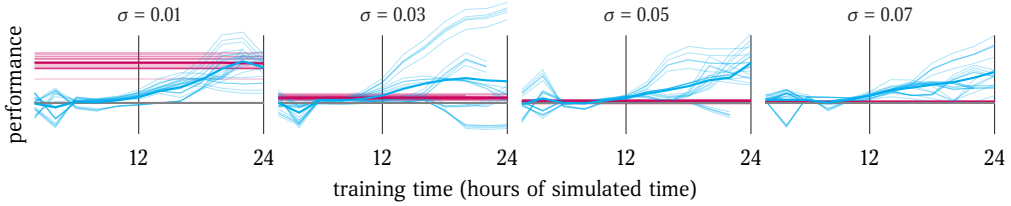


Figure 6.4: Rewards in the noisy observations benchmark. The gray line (bottom) is the baseline method, the purple horizontal line is `FLORIS`, and the blue lines show the learning progress of `SAC`.

6.5 CONCLUSION

Active wake control is a promising real-life application of reinforcement learning. On the one hand, this problem can be very difficult to solve. Its states are only partially observable, the observations are noisy, and the state-action space can be extremely large for large wind farms. On the other hand, emerging research in this domain indicates that the RL community is well equipped to solve this problem, potentially saving millions of dollars in energy losses due to wake effects.

To facilitate future research in this direction, we have presented a new simulator for this problem. It is based on the state-of-the-art steady-state atmospheric simulator called `FLORIS`. Our simulator includes many aspects of the problem not seen in the RL research of active wake control before, such as decoupling of measurement devices from turbines and changes in wind conditions. Our simulator is implemented as an OpenAI Gym environment, is easy to use off the shelf, and is completely open source.

While previous RL approaches for this wake control all use the same action encoding, we identified two possible alternatives. We then experimentally showed that the choice of such an encoding has a great impact on the performance of learning methods. Interestingly, the most common one—yaw-based—performed the worst in our experiments. Soft actor-critic, while a golden standard in RL research, has never been applied to active wake control before, and we demonstrated that it shows better performance than `TD3`.

Finally, we showed that in the presence of imperfect observations, a deep-RL agent is capable of learning a better strategy than the state-of-the-art model-based one.

Deep reinforcement learning for active wake control holds great promise for further refinement: first, `FLORIS` is steady state, which means it optimizes yaw only for the particular time the state

was measured. RL methods have techniques to predict the next state and would therefore pick an action that is best suited for the duration until the next course of action can be taken. Second, where FLORIS has a fixed set of parameters, RL techniques can easily be augmented with other potentially relevant data picked up by sensors. Especially deep-RL techniques seem to be promising when data gets highly-dimensional. Third, we expect deep-RL techniques to outperform model-based optimal control such as FLORIS in terms of computational efficiency, which is especially relevant for big windfarms.

Besides further exploring potential benefits of reinforcement learning, also some more technical questions remain. Is there an even better action encoding system? Or a different state representation? Are there alternative reward shaping methods? While we investigated some state-of-the-art deep-RL methods, sophisticated alternatives exist. *Rainbow* Hessel et al., 2018 combines aspects of many existing RL algorithms. Distributional reinforcement learning Bellemare et al., 2017 provides an alternative learning paradigm by using distributions instead of deterministic state-action values.

Practical implementation of active wake control methods comes with challenges as well. The wind farm operator needs to maximize power production, but also to minimize structural loads on the turbines. This can be done via safe reinforcement learning [García and Fernández, 2015] or multi-objective reinforcement learning [Liu et al., 2014]. Another problem is scalability; perhaps multi-agent reinforcement learning [Hernandez-Leal et al., 2019] can learn to perform active wake control in large-scale wind farms. Finally, reinforcement learning requires exploration, which will inevitably cost money to the wind farm owner. This can be addressed by using offline reinforcement learning [Levine et al., 2020; Agarwal et al., 2020] and learning from the past data, or by using more sample-efficient methods, such as optimistic reinforcement learning [Ciosek et al., 2019; Neustroev and de Weerd, 2020]. Finally, the evaluation of the performance of reinforcement learning vs. model-based wind farm control in more realistic atmospheric environments as present in field tests and atmospheric LES models remains an open topic.

We hope that this work sparks interest of the RL community in this problem, and that our results will make it easier for other researchers to develop new methods for active wake control.

7

Discussion

*What I propose, therefore, is very simple:
it is nothing more than to think what we
are doing.*

— Hannah Arendt,
The Human Condition

SEQUENTIAL DECISION-MAKING under uncertainty is one of the key research areas in artificial intelligence. It studies the ways for the agent to operate under uncertain or incomplete information. We began with a goal

7.1 Answers to the Research Questions

TO OBTAIN NEW INSIGHTS AND DEVELOP NOVEL
ALGORITHMS BY GENERALIZING THE EXISTING THEORY
AND APPLICATIONS OF SEQUENTIAL DECISION-MAKING
UNDER UNCERTAINTY.

We now conclude with a discussion of how this goal was achieved. First, we re-examine the research questions. Next, we address the implications of our findings for society. Finally, we discuss possible future research directions.

7.1 ANSWERS TO THE RESEARCH QUESTIONS

We begin with a retrospection of the research questions posed in the introduction and explain how they were addressed.

Research question 1

First, we asked ourselves,

HOW CAN WE FIND OPTIMAL DECISIONS
IN NON-STATIONARY INFINITE-HORIZON PROBLEMS
WITH UNBOUNDED REWARDS?

To answer this question, we surveyed the existing research. We found that when the agent is able to identify an optimal initial decision, a so-called rolling-horizon procedure can be employed to act optimally. We then examined solution horizon methods that seek initial-decision optimal policies and found that they require the rewards of the problem to be uniformly bounded, including the methods based on linear programming. At the same time, we saw that under the universal optimality criterion, the dual linear-programming approach is applicable in the unbounded case as well.

Because universal optimality implies initial-decision optimality, we combined these approaches to design a novel solution-horizon algorithm of Chapter 3. Moreover, in our algorithm we were able to forego the universal optimality entirely, showing that a weaker notion of occupancy-based optimality is sufficient to establish initial-solution optimality. While the resulting problem is still infinitely-dimensional, it can be approximated by a finite problem

7 Discussion

that stops at a certain horizon. This approximation is known as a truncation. By using the multi-stage contraction properties of the new formulation, we showed that the truncation-based approach can be made monotonically convergent when using multi-step horizon increments instead of the previously used single-step ones.

The algorithm we proposed is applicable to non-stationary problems with unbounded rewards, providing an answer to this research question. As a result, it can be used to find optimal initial decisions in problems where this was not possible before. Additionally, even in problems with bounded rewards it is able to outperform the original solution-horizon method by better utilizing the knowledge of the reward function.

Research question 2

The answer to the first question involved a reformulation of the non-stationary problem as a countably-infinite one. This gave rise to the following question:

HOW CAN WE FIND OPTIMAL DECISIONS IN PROBLEMS
WITH COUNTABLY-INFINITE ENVIRONMENTS?

We answered this research question in Chapter 4 by generalizing the theory and method of Chapter 3. We showed that the truncation-based approach can be extended to countably-infinite problems with unbounded rewards in general. Then we designed a different algorithm that performs policy iteration in these problems to eliminate provably suboptimal actions. Additionally, the proposed algorithm can be applied even when the state space is multidimensional. One of such problems is the inventory management problem with countably-infinite state space. We showed how our approach can be used to solve this problem.

E.g., in FrozenLake8×8-v0 with the discounting factor of 0.99 the uniform reward bound of one leads to the upper value bound of $100 = 1 + 0.99 + 0.99^2 + 0.99^3 + \dots$. In practice, the state values are never greater than one, and for some of the states—the holes—they are equal to zero.

Additionally, some of the existing methods for planning and learning rely on the uniform reward bounds which exist trivially in the finite case. These bounds can be incorporated in the algorithms—often implicitly—to reason about possible outcomes of the future actions. Countably-infinite problems lead to unbounded rewards; therefore, we must forego this uniform bound in the reasoning. Instead, we use a concept of a weight function. This function makes the information on the future states more realistic, in the uniformly bounded case, it can be used to accelerate convergence of the decision-making algorithms. In general, the weighted-supremum approach leads to better performance by in-

corporating more information into the decision-making process. Therefore, it should be employed in favor of the uniformly bounded assumption.

Research question 3

Next, we focused our attention on reinforcement learning. We saw that the existing research shows that optimistic Q-learning is provably efficient in stationary problems and asked ourselves,

IS OPTIMISTIC LEARNING EFFICIENT IN NON-STATIONARY PROBLEMS; IF SO, HOW CAN THIS EFFICIENCY BE EXPLAINED?

To answer this question, we considered non-stationary episodic Markov decision processes. We showed that they can be analyzed the same way that stationary models can. Then we proved sample-efficiency of optimistic Q-learning in terms of the asymptotic behavior of the total regret it generates, providing a positive answer to the first part of the research question.

The second part of the research question is addressed by the novel regret analysis of Chapter 5. It is fundamentally different from previous results in this field; while other studies present concrete formulae for regret as a function of time and space dimensions of the problem, we derived a first-of-its-kind high-level result on sample-efficiency of optimistic Q-learning. Instead of showing *how* the regret behaves asymptotically, we explain *why* it behaves in this way. Additionally, unlike any of the previous results, our analysis does not depend on any particular learning rate function, generalizing some results previously known only for specific learning rates.

Our regret bound provides new insights into the nature of optimistic reinforcement learning. For example, one of the identified regret sources is the estimation error: an RL agent estimates the true transitions and expected rewards of the underlying MDP from the observed ones. In model-based methods, this estimation happens explicitly and is therefore hard to overlook. In model-free learning, however, the estimation occurs implicitly. Our result shows that this is detrimental to applications with no aleatoric uncertainty. In this case, the estimation element of learning can and should be removed, leading to both faster convergence and fewer computations. The distinction between stochastic and deterministic environments should be more commonly adopted in algorithm design; its inclusion is not arduous and leads to more

7.1 Answers to the Research Questions

See Corollary 5.2. p. 136.

See Section 5.6.2. p. 146.

efficient algorithms.

7 Discussion

Finally, we showed how the theoretical result of Chapter 5 can be used to facilitate design of new optimistic RL methods. We gave an example of one such method, UCB-H⁺. We proved its efficiency and demonstrated that it is capable of outperforming UCB-H, supplementing the theoretical findings of Chapter 5 with more practical results.

Research question 4

Our final research question was

HOW CAN REINFORCEMENT LEARNING BE APPLIED TO
EFFICIENTLY SOLVE REAL-WORLD PROBLEMS SUCH AS
ACTIVE WAKE CONTROL?

To answer it, we surveyed the existing body of research on reinforcement learning for active wake control. We found that it is rather scarce, but most of the proposed solutions use wind farm simulations in lieu of field studies. This is especially important given the trial-and-error nature of reinforcement learning. At the same time, we saw that the existing simulations are far detached from the reality of a wind farm operation. To provide a remedy for this problem, we designed a simulation toolbox for active wake control based on the state-of-the-art FLORIS framework. Our simulator is highly configurable and easy to employ.

Unlike wind farm models previously used in RL research, we included a way to add additional data sources such as nacelle-mounted sensors and external information providers. The measurement data in our problem is aggregated into a state space vector, which can be given as an input to an RL method including off-the-shelf solutions.

All of the previous research in RL for active wake control used the same action description based on the maximum angular velocity of the turbines. We considered two alternative ways to encode the actions in this problem: as a desired angle from either a fixed direction—for example, north—or from the wind direction. In the experimental evaluation, we found that the proposed action representations can lead to improved performance of RL algorithms; therefore, future research should consider using one of these action encodings for more efficient reinforcement learning.

See Section 6.4.1 p. 163.

Additionally, to illustrate the potential of reinforcement learning compared to other state-of-the-art control methods, we investigated the impact of information noise on the learning process

in active wake control. Our findings revealed that reinforcement learning can be more robust to distorted inputs than model-based control methods. This property is especially useful in real-world applications, as sensor readings rarely provide perfect information.

See Section 6.4.2, p. 166.

7.2 SOCIETAL IMPLICATIONS

The first part of this thesis focuses on planning in countably-infinite Markov decision processes. The reader may wonder what is the goal of studying these models. After all, is not infinity but a mathematical abstraction?

For example, let us consider the inventory management problem of Section 1.3.2. The reason why the sample space of this problem is infinite is twofold: on the one hand, stock at hand can be unlimited, on the other, the decision-making continues *ad infinitum*. Of course, there is no infinite storage space in reality, nor is the agent expected to operate a warehouse eternally.

See Section 1.3.2, p. 6.

While countably-infinite problems present a unique mathematical challenge, there is a practical implication as well. When a hyperparameter of the problem is known to be finite, this information is often embedded in the solution methods. The same is simply not possible for infinite values, and different methods need to be developed without such hyperparameters. As a result, these new methods can be applied to finitely-countable problems where the aforementioned hyperparameters are unknown.

Businesses and governments alike make many of their decisions by choosing an arbitrary planning horizon. In the European Union, for example, investment in research and innovation is currently planned for 2021–2027 [EU, 2021], and the climate policy is laid down until 2030 [EU, 2022]. When choosing planning horizons like these, the solution-horizon approach of Chapter 3 can be used to reason whether the selected horizon is chosen appropriately.

Similarly, when non-temporal parts of the state space are infinite, we can think of them as either unknown or irrelevant to the solution. In the inventory management example, the maximum warehouse capacity is always limited. At the same time the manager of a warehouse is probably not interested in filling the whole storage space with just pens or staplers. Traditional methods based on Markov decision processes require a complete state space specification, including many irrelevant warehouse states like these. In Chapter 4, we propose to increase the considered

7 Discussion

See Theorem 5.1 p. 136.

state of the problem—the stock in this case—incrementally, until its sufficiently large for the decision to be made. Like in the time horizon case, this approach does not require the truncation to be chosen by a human *a priori*.

The regret analysis of Chapter 5 provides a new viewpoint on the efficiency analysis of reinforcement learning. It can be used in the design of future algorithms, resulting in faster, more sample-efficient training, which is crucial for applications of reinforcement learning to many real-life problems.

One of such problems is active wake control in wind farms. Wake effects account for substantial losses in energy production, and wake control strategies can be used to boost the efficiency of wind farms. More efficient energy production in wind farms can facilitate their adoption, aiding the transition from fossil-based fuel to renewable energy sources. This is especially important in achieving the United Nations resolution to limit the rise in global temperatures by 2050 to 1.5 °C above pre-industrial levels [UN, 2015]; a strenuous undertaking that requires the share of solar and wind power to increase to 74% of the total power generation capacity [IRENA, 2022, Chapter 1].

7.3 FUTURE RESEARCH DIRECTIONS

The work presented in this thesis answers some questions about the nature of sequential decision-making; at the same time, it presents new challenges and opens opportunities for future research. In this section, we discuss the potential future research directions. We group them into two categories: possible extensions and speculative future prospects.

7.3.1 Theoretical and Algorithmic Extensions

In this section, we discuss some of the more straightforward research directions. Most of these are possible extensions of the theory and methodology presented in this thesis.

Span-based salvage spaces

The proposed planning methods for non-stationary and continuous MDPs search for possible value approximations in the dual problems that can lead to alternative solutions to the primal problems. The search is performed within what we call salvage spaces.

Naturally, the smaller the salvage space is, the better its points approximate the true optimal values of the problem.

In this thesis, we defined the salvage spaces in terms of absolute bounds only. However, these spaces can be made smaller by introducing additional span-based constraints. In fact, in the case of uniformly bounded rewards, this was done by Bean et al. [1992]. In the unbounded case, however, such a result is not available. Similarly to weighted-supremum norms that we use, Scherrer [2007] introduced weighted spans. With some additional analysis, these can be used to extend the method of Bean et al. [1992] to problems with unbounded rewards and to improve the algorithms of Chapters 3 and 4.

Continuous (Borel) models

Many of the reinforcement learning problems have continuous elements in their sample spaces. For example, in the active wake control problem, both the states and the actions can be continuous.

The theory of MDPs with Borel spaces—both continuous and discrete—is well established [Hernández-Lerma and Lasserre, 2012]. While value- and policy-based methods can be applied to such problems [Yu and Bertsekas, 2015], including problems with unbounded rewards [Hernández-Lerma and Muñoz de Ozak, 1992], the dual linear-programming becomes impossible.

At a glance, this may look as an unsolvable challenge, but other notions of duality can be used where linear programming fails. For example, a recent study by Nachum and Dai [2020] provides a connection between Fenchel–Rockafellar duality and reinforcement learning, and Laroche et al. [2022] extends the theory of occupancy measures in Borel spaces.

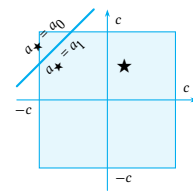
By combining these results with the theory presented in this thesis, it should be possible to extend the proposed methods for planning and reinforcement learning to continuous MDPs, making them applicable to a larger class of problems.

Sample-efficient reinforcement learning for active wake control

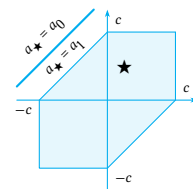
In this thesis, we studied optimistic RL algorithm. It is very eager in its exploration: after all, an optimistic agent always assumes that unencountered states hold a great promise. This property makes it not suitable for field studies where the cost of failure can be high. At the same time, if exploration can be done at little to no cost—for example, in simulations—optimistic learning becomes

7.3 Future Research Directions

Consider the following geometric illustration. In the value space, the optimal actions above and below the thick line are a_0 and a_1 . The true optimal values (\star) result in a_1 being optimal, but they are not known to the agent. If the agent knows that the value norm is absolutely bounded by c , the search space is a square and both actions may be optimal.



If the agent knows that the value span (the absolute difference) is also bounded by c , the search space becomes smaller and a_1 can be guaranteed to be optimal without knowing the exact optimal values.



7 Discussion

especially promising due to its provable efficiency.

In this thesis, we presented a simulator for the active wake control problem. And yet we did not use it in combination with optimistic reinforcement learning.

The main reason for this is that active wake control is a continuous problem. While it can be solved via discretization, algorithms tailored to such problems—such as SAC considered in Chapter 6—tend to perform better.

Another reason is the so-called curse of dimensionality. Optimistic learning keeps track of state-action visitations. In our attempts to apply optimistic Q-learning to active wake control, we saw that the visitation function becomes hard to approximate as the number of the problem’s dimensions grow. For example, a simple approach is to divide a state space into bins and count visitations within those bins. If the state space is ten-dimensional, even 5 bins per dimension lead to almost ten million bins overall. As a result, most of them contain zeros, making the agent explore unnecessarily aggressively. We considered other approximations as well [H. Tang et al., 2017; Simão and Spaan, 2019], but none of them yielded satisfactory results.

Of course, one of the possible research directions is to explore even more pseudo-visitation approaches [B. Tang, 1993; Martin et al., 2017]. Alternatively, if a Borel model of optimism is designed in the context of the previous section, it can be applied to active wake control as well.

7.3.2 Future Prospects

The research directions presented in this section are more long-term. They do not yet have an immediately obvious way to address them and pose more significant scientific challenges.

From sufficient to necessary conditions

The weight function is a key component of the planning methods in countably-infinite domains. We use the properties of this function to establish existence of policy values via multi-stage contractive properties of the Bellman operators. At the same time, weight functions with different properties are sometimes used to achieve similar results [Cavazos-Cadena, 1986; Altman, 1999]. Thus, the conditions we impose on the problems are sufficient, but not necessary.

The necessary conditions for duality in countably-infinite MDPs

are still not known. Their discovery can be an important theoretical contribution that can shed light on the nature of such MDPs.

Similarly, our analysis of optimism in Q-learning relies on a few sufficient conditions on the learning rate and the problem's data. Necessary conditions for sample efficiency of Q-learning are not known; their discovery will be a significant contribution to the field of reinforcement learning.

7.3 Future Research Directions

Connections between truncations and optimism

There are possible connections between the notions of truncations and optimism. In the analysis of Chapter 4, we added a bonus term to the rewards to represent the uncertainty of the states we are not yet taking into account directly. In optimistic Q-learning, we add a bonus to the reward function as well. This bonus is based on the number of visitations of a particular state-action pair; it represents our uncertainty about the state-action pair and decreases as the algorithm continues to encounter that state-action pair. Perhaps, there exists a deeper link between the two approaches that can be explored to better understand the nature of both of them.

Connections between duality and actor-critic methods

The dual linear-programming approach utilizes two problems. One of them—the one that we call primal—is based on occupancies and its solution provides the agent with a policy. The other problem—the dual—seeks the optimal state values.

In actor-critic reinforcement learning, two neural networks work in tandem. The actor estimates the policy. The critic estimates the values under the actor's policy. Both use gradient descent to improve their estimates.

The actor and the critic resemble the primal and the dual programs. Moreover, their alternating learning is similar to the primal-dual gradient descent method [Du and Hu, 2019]. Further exploration of this idea can advance the theory of actor-critic reinforcement learning.

* * *

In conclusion, the results presented in this thesis advance both the theory of sequential decision-making under uncertainty and its potential for real-world applications, paving the path for further development of efficient algorithms for planning and learning.

References

A

- Abramowitz, Milton. “Elementary Analytical Methods”. In: *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Ed. by Milton Abramowitz and Irene A. Stegun. 10th ed. National Bureau of Standards Applied Mathematics Series 55. Washington, D. C., USA: U. S. Government Printing Office, Dec. 1972. Chap. 3, pp. 9–63. URL: <http://www.worldcat.org/oclc/25644903>.
- Achiam, Joshua S. *Spinning Up in Deep Reinforcement Learning*. <https://github.com/openai/spinningup>. 2018.
- Agarwal, Rishabh, Dale Schuurmans, and Mohammad Norouzi. “An Optimistic Perspective on Offline Reinforcement Learning”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 104–114. ACM DL: 10.5555/3524938.3524949.
- Agrawal, Shipra and Randy Jia. “Optimistic Posterior Sampling for Reinforcement Learning: Worst-Case Regret Bounds”. In: *Advances in Neural Information Processing Systems* 30. Ed. by Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett. Long Beach, California, USA: Curran Associates, Inc., 2017, pp. 1184–1194. ISBN: 978-1-510-86096-4.
- Aliprantis, Charalambos D. and Kim C. Border. *Infinite Dimensional Analysis: A Hitchhiker’s Guide*. 3rd ed. Heidelberg, Germany: Springer Berlin, 2006. xxii, 704 pp. ISBN: 978-3-540-29586-0. DOI: 10.1007/3-540-29587-9.
- Altman, Eitan. *Constrained Markov Decision Processes: Stochastic Modeling*. New York, USA: Routledge, 1999. 256 pp. ISBN: 9781315140223. DOI: 10.1201/9781315140223.
- Annoni, Jennifer, Paul A. Fleming, Andrew K. Scholbrock, J. Roadman, S. Dana, C. Adcock, F. Porte-Agel, S. Raach, F. Haizmann, and D. Schlipf. “Analysis of control-oriented wake modeling tools using lidar field results”. In: *Wind Energy Science* 3.2 (2018), pp. 819–831. DOI: 10.5194/wes-3-819-2018.

- Antos, András, Csaba Szepesvári, and Rémi Munos. “Value-Iteration Based Fitted Policy Iteration: Learning with a Single Trajectory”. In: *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. IEEE. 2007, pp. 330–337. DOI: 10.1109/ADPRL.2007.368207.
- Arenas-López, J. Pablo and Mohamed Badaoui. “The Ornstein–Uhlenbeck process for estimating wind power under a memoryless transformation”. In: *Energy* 213 (2020), pp. 1–15. ISSN: 0360-5442. DOI: 10.1016/j.energy.2020.118842.
- Ashok, Pranav, Krishnendu Chatterjee, Przemysław Daga, Jan Křetínský, and Tobias Meggendorfer. “Value iteration for long-run average reward in Markov decision processes”. In: *International Conference on Computer Aided Verification*. Springer. 2017, pp. 201–221.
- Azar, Mohammad G., Rémi Munos, Mohammad Ghavamzadeh, and Hilbert J. Kappen. “Speedy Q-learning”. In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Red Hook, New York, USA: Curran Associates, Inc., 2011, pp. 2411–2419.
- Azar, Mohammad G., Ian Osband, and Rémi Munos. “Minimax regret bounds for reinforcement learning”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. ICML’17. Sydney, NSW, Australia: JMLR.org, Aug. 2017, pp. 263–272.
- Bai, Yu, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. “Provably efficient Q-learning with low switching cost”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, Edward J. Fox, and R. Garnett. Red Hook, New York, USA: Curran Associates, Inc., 2019, pp. 8002–8011.
- Balaji, Nikhil, Stefan Kiefer, Petr Novotný, Guillermo A. Pérez, and Mahsa Shirmohammadi. “On the Complexity of Value Iteration”. In: *arXiv preprint arXiv:1807.04920* (2018).
- Banach, Stefan. “Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales”. In: *Fundamenta Mathematicæ* 3.1 (1922), pp. 133–181. DOI: 10.4064/fm-3-1-133-181.

- Barthelmie, R., S. Frandsen, K. Hansen, Jan G. Schepers, K. Rados, W. Schlez, A. Neubert, L. Jensen, and S. Neckelmann. “Modelling the impact of wakes on power output at Nysted and Horns Rev”. In: *European Wind Energy Conference*. Vol. 2. WindEurope, 2009, pp. 1–10.
- Bean, James C., Wallace J. Hopp, and Izak Duenyas. “A Stopping Rule for Forecast Horizons in Nonhomogeneous Markov Decision Processes”. In: *Operations Research* 40.6 (1992), pp. 1188–1199. ISSN: 0030364X, 15265463.
- Bellemare, Marc G., Will Dabney, and Rémi Munos. “A Distributional Perspective on Reinforcement Learning”. In: *Proceedings of Machine Learning Research* 70 (Aug. 2017). Ed. by Doina Precup and Yee Whye Teh, pp. 449–458. URL: <https://proceedings.mlr.press/v70/bellemare17a.html>.
- Bellman, Richard E. “The Theory of Dynamic Programming”. In: *Bulletin of the American Mathematical Society* 60.6 (1954), pp. 503–515.
- Bellman, Richard E. and Stuart E. Dreyfus. *Applied Dynamic Programming*. Princeton Legacy Library. Princeton, New Jersey, USA: Princeton University Press, 2016. Chap. 3. 390 pp. ISBN: 9780691651873.
- Bès, Christian and Jean B. Lasserre. “An on-line procedure in discounted infinite-horizon stochastic optimal control”. In: *Journal of Optimization Theory and Applications* 50.1 (1986), pp. 61–67.
- Bès, Christian and Suresh P. Sethi. “Concepts of Forecast and Decision Horizons: Applications to Dynamic Stochastic Optimization Problems”. In: *Mathematics of Operations Research* 13.2 (1988), pp. 295–310. ISSN: 0364765X, 15265471.
- Blackwell, David. “Discounted Dynamic Programming”. In: *The Annals of Mathematical Statistics* 36.1 (1965), pp. 226–235.
- Bot, Edwin T. G. *Flow Analysis with Nacelle-Mounted LiDAR*. Tech. rep. ECN-E-16-041. ECN, 2016. URL: <https://publications.tno.nl/publication/34629395/v931oD/e16041.pdf>.
- Brockman, Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. *OpenAI Gym*. 2016. arXiv: 1606.01540 [cs.LG].

C

- Cavazos-Cadena, Rolando. "Finite-state approximations for denumerable state discounted markov decision processes". In: *Applied Mathematics & Optimization* 14.1 (Apr. 1986), pp. 1–26. ISSN: 0095-4616. DOI: 10.1007/BF01442225.
- Chand, Suresh, Vernon N. Hsu, and Suresh P. Sethi. "Forecast, Solution, and Rolling Horizons in Operations Management Problems: A Classified Bibliography". In: *Manufacturing & Service Operations Management* 4.1 (2002), pp. 25–43.
- Cheevaprawatdomrong, Torpong, Irwin E. Schochetman, Robert L. Smith, and Alfredo Garcia. "Solution and Forecast Horizons for Infinite-Horizon Nonhomogeneous Markov Decision Processes". In: *Mathematics of Operations Research* 32.1 (2007), pp. 51–72.
- Cheevaprawatdomrong, Torpong and Robert L. Smith. "Infinite horizon production scheduling in time-varying systems under stochastic demand". In: *Operations Research* 52.1 (2004), pp. 105–115.
- Ciosek, Kamil, Quan Vuong, Robert Loftin, and Katja Hofmann. "Better Exploration with Optimistic Actor-Critic". In: *arXiv preprint arXiv:1910.12807* (2019).
- Corless, Robert M., Gaston H. Gonnet, David E. G. Hare, David J. Jeffrey, and Donald E. Knuth. "On the Lambert W Function". In: *Advances in Computational mathematics* 5.1 (1996), pp. 329–359.
- Crosswind. *Crosswind Innovations*. <https://www.crosswindhkn.nl/innovations>. Accessed: 2021-10-25. 2021.

D

- Denardo, Eric V. and Bennett L. Fox. "Multichain Markov Renewal Programs". In: *SIAM Journal on Applied Mathematics* 16.3 (1968), pp. 468–487.
- Devraj, Adithya M. and Sean Meyn. "Zap Q-learning". In: *Advances in Neural Information Processing Systems* 30. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Red Hook, New York, USA: Curran Associates, Inc., 2017, pp. 2235–2244.
- Dong, Hongyang, Jincheng Zhang, and Xiaowei Zhao. "Intelligent wind farm control via deep reinforcement learning and high-fidelity simulations". In: *Applied Energy* 292 (2021), p. 116928. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2021.116928.

Du, Simon S. and Wei Hu. “Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity”. In: *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR. 2019, pp. 196–205.

Dynkin, Evgeniy Borisovich and Alexander Adolphovich Yushkevich. *Controlled Markov Processes*. Vol. 235. Springer, 1979.

d’Epenoux, F. “A Probabilistic Production and Inventory Problem”. In: *Management Science* 10.1 (1963), pp. 98–108. ISSN: 00251909, 15265501. DOI: 10.1287/mnsc.10.1.98.

E

European Union. “Decision (EU) 2022/591 of the European Parliament and of the Council of 6 April 2022 on a General Union Environment Action Programme to 2030”. In: *Official Journal of the European Union* L 114, 65 (Apr. 2022), pp. 22–36. ISSN: 1977-0677.

—— “Regulation (EU) 2021/695 of the European Parliament and of the Council of 28 April 2021 establishing Horizon Europe – the Framework Programme for Research and Innovation, laying down its rules for participation and dissemination, and repealing Regulations (EU) No 1290/2013 and (EU) No 1291/2013 (Text with EEA relevance)”. In: *Official Journal of the European Union* L 170, 64 (May 2021), pp. 1–68. ISSN: 1977-0677.

Even-Dar, Eyal, Shie Mannor, and Yishay Mansour. “Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems”. In: *Journal of Machine Learning Research* 7 (Dec. 2006), pp. 1079–1105. ISSN: 1532-4435.

Even-Dar, Eyal and Yishay Mansour. “Convergence of Optimistic and Incremental Q-Learning”. In: *Advances in Neural Information Processing Systems* 14. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. Cambridge, Massachusetts, USA: The MIT Press, 2002, pp. 1499–1506.

Fleming, Paul A., Jennifer Annoni, Jigar J. Shah, Linpeng Wang, Shreyas Ananthan, Zhijun Zhang, Kyle Hutchings, Peng Wang, Weiguo Chen, and Lin Chen. “Field Test of Wake Steering at an Offshore Wind Farm”. In: *Wind Energy Science* 2.1 (2017), pp. 229–239. DOI: 10.5194/wes-2-229-2017.

F

Fleming, Paul A., Pieter M. O. Gebraad, Jan-Willem van Wingerden, Sang Lee, Matt Churchfield, Andrew K. Scholbrock, John Michalakes, Kathryn Johnson, and Pat Moriarty. *SOWFA Super-Controller: A High-Fidelity Tool for Evaluating Wind Plant Control Approaches*. Tech. rep. Golden, Colorado, USA: National Renewable Energy Lab. (NREL), 2013.

Fox, Bennett L. “Finite-state approximations to denumerable-state dynamic programs”. In: *Journal of Mathematical Analysis and Applications* 34.3 (1971), pp. 665–670. ISSN: 0022-247X. DOI: 10.1016/0022-247X(71)90106-5.

Fox, Edward J., Richard Metters, and John Semple. “Optimal Inventory Policy with Two Suppliers”. In: *Operations Research* 54.2 (2006), pp. 389–393. DOI: 10.1287/opre.1050.0229.

François-Lavet, Vincent, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. “An Introduction to Deep Reinforcement Learning”. In: *Foundations and Trends in Machine Learning* 11.3–4 (2018), pp. 219–354. ISSN: 1935-8237. DOI: 10.1561/22000000071.

Fujimoto, Scott, Herke van Hoof, and David Meger. “Addressing Function Approximation Error in Actor-Critic Methods”. In: *CoRR* abs/1802.09477 (2018), pp. 1582–1591. arXiv: 1802.09477. URL: <http://arxiv.org/abs/1802.09477>.

G

García, Javier and Fernando Fernández. “A comprehensive survey on safe reinforcement learning”. In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1437–1480.

Gebraad, Pieter M. O., Floris W. Teeuwisse, J. W. Wingerden, Paul A. Fleming, Shalom D. Ruben, Jason R. Marden, and Lucy Y. Pao. “Wind plant power optimization through yaw control using a parametric model for wake effects—A CFD simulation study”. In: *Wind Energy* 19 (Dec. 2016), pp. 95–114. DOI: 10.1002/we.1822.

Ghate, Archis. “Circumventing the Slater Conundrum in Countably Infinite Linear Programs”. In: *European Journal of Operational Research* 246.3 (2015), pp. 708–720. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2015.04.026.

———. “Infinite Horizon Problems”. In: *Wiley Encyclopedia of Operations Research and Management Science* (2011).

Ghate, Archis and Robert L. Smith. “A Linear Programming Approach to Nonstationary Infinite-Horizon Markov Decision Processes”. In: *Operations Research* 61.2 (2013), pp. 413–425.

Gilbert, Ciaran, Jakob Messner, Pierre Pinson, Pierre-Julien Trombe, Remco Verzijlbergh, Pim Dorp, and Harmen Jonker. “Statistical post-processing of turbulence-resolving weather forecasts for offshore wind power forecasting”. In: *Wind Energy* 23 (Apr. 2020). DOI: 10.1002/we.2456.

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor”. In: *CoRR abs/1801.01290* (2018). arXiv: 1801.01290. URL: <http://arxiv.org/abs/1801.01290>.

H

Harrison, J. Michael. “Discrete dynamic programming with unbounded rewards”. In: *The Annals of Mathematical Statistics* 43.2 (1972), pp. 636–644.

Hauskrecht, Milos and Branislav Kveton. “Approximate Linear Programming for Solving Hybrid Factored MDPs.” In: *AI & M.* 2006.

—— “Linear program approximations for factored continuous-state Markov decision processes”. In: *Advances in Neural Information Processing Systems* 16 (2003).

Hernandez-Leal, Pablo, Bilal Kartal, and Matthew E Taylor. “A survey and critique of multiagent deep reinforcement learning”. In: *Autonomous Agents and Multi-Agent Systems* 33.6 (2019), pp. 750–797.

Hernández-Lerma, Onésimo and Jean B. Lasserre. “A forecast horizon and a stopping rule for general Markov decision processes”. In: *Journal of Mathematical Analysis and Applications* 132.2 (June 1988), pp. 388–400. ISSN: 0022247X. DOI: 10.1016/0022-247X(88)90069-8.

—— *Discrete-Time Markov Control Processes: Basic Optimality Criteria*. Vol. 30. Springer Science & Business Media, 2012.

—— “The Linear Programming Approach”. In: *Handbook of Markov Decision Processes: Methods and Applications*. Ed. by Eugene A. Feinberg and Adam Schwartz. 1st ed. Vol. 40. International Series in Operations Research & Management Science (ISOR). New York, USA: Springer, 2002. Chap. 11,

pp. 377–407. ISBN: 978-0-7923-7459-6. DOI: 10.1007/978-1-4615-0805-2.

Hernández-Lerma, Onésimo and Myriam Muñoz de Ozak. “Discrete-Time Markov Control Processes with Discounted Unbounded Costs: Optimality Criteria”. In: *Kybernetika* 28.3 (1992), pp. 191–212.

Hessel, Matteo, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad G. Azar, and David Silver. “Rainbow: combining improvements in deep reinforcement learning”. In: *The 32nd AAAI Conference on Artificial Intelligence*. New Orleans, Louisiana, USA: AAAI Press, Feb. 2018, pp. 3215–3222.

Heus, T., C. C. van Heerwaarden, H. J. J. Jonker, A. Pier Siebesma, S. Axelsen, et al. “Formulation of the Dutch Atmospheric Large-Eddy Simulation (DALES) and Overview of Its Applications”. In: *Geoscientific Model Development* 3.2 (2010), pp. 415–444. DOI: 10.5194/gmd-3-415-2010.

Hopp, Wallace J. “Identifying Forecast Horizons in Nonhomogeneous Markov Decision Processes”. In: *Operations Research* 37.2 (1989), pp. 339–343.

Hopp, Wallace J., James C. Bean, and Robert L. Smith. “A New Optimality Criterion for Nonhomogeneous Markov Decision Processes”. In: *Operations Research* 35.6 (Dec. 1987), pp. 875–883.

Howard, Ronald A. *Dynamic Programming and Markov Processes*. New York, USA: Technology Press of the Massachusetts Institute of Technology and Wiley, 1960. 136 pp. ISBN: 978-0262080095.

Howland, Michael F., Sanjiva K. Lele, and John O. Dabiri. “Wind farm power optimization through wake steering”. In: *Proceedings of the National Academy of Sciences* 116.29 (2019), pp. 14495–14500. ISSN: 0027-8424. DOI: 10.1073/pnas.1903680116.

Huang, Shengyi, Rousslan Dossa, and Chang Ye. *CleanRL: High-Quality Single-File Implementation of Deep Reinforcement Learning Algorithms*. <https://github.com/vwxyzjn/cleanrl/>. 2020.

- International Renewable Energy Agency (IRENA). *World Energy Transitions Outlook: 1.5 °C pathway*. Report. Abu Dhabi, 2022. 352 pp. URL: <https://www.irena.org/publications/2022/Mar/World-Energy-Transitions-Outlook-2022>.
- Ionescu-Tulcea, Cassius T. “Mesures dans les espaces produits”. In: *Atti della Accademia nazionale dei Lincei. Rendiconti. Classe di scienze fisiche, matematiche e naturali*. 8.7 (1949), pp. 208–211.
- Jaakkola, Tommi, Michael I. Jordan, and Satinder P. Singh. “Convergence of Stochastic Iterative Dynamic Programming Algorithms”. In: *Advances in Neural Information Processing Systems* 6. Ed. by J. D. Cowan, Gerald Tesauro, and J. Alspector. Burlington, Massachusetts, USA: Morgan–Kaufmann, 1994, pp. 703–710.
- Jacobson, Mark Z. and Mark A. Delucchi. “A Path to Sustainable Energy by 2030”. In: *Scientific American* 301.5 (2009), pp. 58–65. ISSN: 00368733, 19467087.
- Jensen, Niels O. *A note on wind generator interaction*. 1983.
- Jiménez, Ángel, Antonio Crespo, and Emilio Migoya. “Application of a LES technique to characterize the wake deflection of a wind turbine in yaw”. In: *Wind Energy* 13.6 (2010), pp. 559–572. DOI: 10.1002/we.380.
- Jin, Chi, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I. Jordan. “Is Q-Learning Provably Efficient?” In: *Advances in Neural Information Processing Systems* 31. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Red Hook, New York, USA: Curran Associates, Inc., 2018, pp. 4863–4873.
- Johnson, Kathryn E. *Adaptive Torque Control of Variable Speed Wind Turbines*. Tech. rep. Golden, Colorado, USA: National Renewable Energy Lab. (NREL), 2004.
- Kakade, Sham, Mengdi Wang, and Lin F. Yang. *Variance Reduction Methods for Sublinear Reinforcement Learning*. 2018. arXiv: 1802.09184 [cs.AI].
- Kapturowski, Steven, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. “Recurrent Experience Replay in Distributed Reinforcement Learning”. In: *International conference on learning representations*. 2018.

L

- Laroche, Romain, Remi Tachet des Combes, and Jacob Buckman. *Non-Markovian Policies Occupancy Measures*. 2022. DOI: 10.48550/arXiv.2205.13950.
- Lasserre, Jean B. and Christian Bès. “Infinite horizon nonstationary stochastic optimal control problem: A planning horizon result”. In: *IEEE Transactions on Automatic Control* 29.9 (Sept. 1984), pp. 836–837. ISSN: 0018-9286. DOI: 10.1109/TAC.1984.1103671.
- Lee, Ilbin, Marina A. Epelman, H. Edwin Romeijn, and Robert L. Smith. “Simplex Algorithm for Countable-State Discounted Markov Decision Processes”. In: *Operations Research* 65.4 (2017), pp. 1029–1042.
- Levine, Sergey, Aviral Kumar, George Tucker, and Justin Fu. “Offline reinforcement learning: Tutorial, review, and perspectives on open problems”. In: *arXiv preprint arXiv:2005.01643* (2020).
- Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).
- Lippman, Steven A. “On Dynamic Programming with Unbounded Rewards”. In: *Management Science* 21.11 (1975), pp. 1225–1233.
- Liu, Chunming, Xin Xu, and Dewen Hu. “Multiobjective reinforcement learning: A comprehensive overview”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45.3 (2014), pp. 385–398.
- Lundquist, Julie K., Katharine K. DuVivier, Daniel Kaffine, and Jessica M. Tomaszewski. “Costs and consequences of wind turbine wake effects arising from uncoordinated wind energy development”. In: *Nature Energy* 4.1 (Jan. 2019), pp. 26–34. DOI: 10.1038/s41560-018-0281-2.

M

- Madjidian, Daria and Anders Rantzer. “A Stationary Turbine Interaction Model for Control of Wind Farms”. In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 4921–4926. ISSN: 1474-6670. DOI: 10.3182/20110828-6-IT-1002.00267.

- Malek, Alan, Yasin Abbasi-Yadkori, and Peter Bartlett. “Linear programming for large-scale Markov decision problems”. In: *International Conference on Machine Learning*. PMLR. 2014, pp. 496–504.
- Martin, Jarryd, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. “Count-based exploration in feature space for reinforcement learning”. In: *arXiv preprint arXiv:1706.08090* (2017).
- McDiarmid, Colin. “Concentration”. In: *Probabilistic Methods for Algorithmic Discrete Mathematics. Algorithms and Combinatorics*. Ed. by M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed. Vol. 16. Berlin, Heidelberg, Germany: Springer, 1998, pp. 195–248.
- Meucci, Attilio. “Review of Statistical Arbitrage, Cointegration, and Multivariate Ornstein–Uhlenbeck”. In: *SSRN Electronic Journal* (May 2009), p. 20. DOI: 10.2139/ssrn.1404905.
- . *Risk and Asset Allocation*. Vol. 1. New York: Springer, 2005.
- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. *Playing Atari with Deep Reinforcement Learning*. 2013. arXiv: 1312.5602 [cs.LG].
- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518 (Feb. 2015), pp. 529–33. DOI: 10.1038/nature14236.
- Moritz, Philipp, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, et al. “Ray: A Distributed Framework for Emerging AI Applications”. In: *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. USENIX association, 2018, pp. 561–577.
- Munos, Rémi and Csaba Szepesvári. “Finite-Time Bounds for Fitted Value Iteration.” In: *Journal of Machine Learning Research* 9.5 (2008).
- Nachum, Ofir and Bo Dai. *Reinforcement Learning via Fenchel–Rockafellar Duality*. 2020. DOI: 10.48550/ARXIV.2001.01866.

- Nair, Suresh K. “Modeling Strategic Investment Decisions Under Sequential Technological Change”. In: *Management Science* 41.2 (Feb. 1995), pp. 282–297. ISSN: 0025-1909. DOI: 10.1287/mnsc.41.2.282.
- National Renewable Energy Lab. (NREL). FLORIS. *Version 2.4*. <https://github.com/NREL/floris>. Golden, Colorado, USA, 2021.
- Neustroev, Grigory, Sytze P. E. Andringa, Remco A. Verzijlbergh, and Mathijs M. de Weerdt. “Deep Reinforcement Learning for Active Wake Control”. In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. Ed. by Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud, and Matthew E. Taylor. Auckland, New Zealand, virtual event: International Foundation for Autonomous Agents and Multiagent Systems, May 2022, pp. 944–953.
- *The Wind Farm Gym*. <https://github.com/AlgTUDelft/wind-farm-env>. 2022. DOI: 10.4121/19107257.
- Neustroev, Grigory and Mathijs M. de Weerdt. “Action Elimination in Countably-Infinite Markov Decision Processes”. To appear; based on Chapter 4 of this thesis. Manuscript in progress. 2022.
- “Generalized Optimistic Q-Learning with Provable Efficiency”. In: *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*. Ed. by Bo An, Neil Yorke-Smith, Amal El Fallah Seghrouchni, and Gita Sukthankar. Auckland, New Zealand, virtual event: International Foundation for Autonomous Agents and Multiagent Systems, May 2020, pp. 913–921.
- Neustroev, Grigory, Mathijs M. de Weerdt, and Remco A. Verzijlbergh. “Discovery of Optimal Solution Horizons in Non-Stationary Markov Decision Processes with Unbounded Rewards”. In: *Proceedings of the 29th International Conference on Automated Planning and Scheduling*. Ed. by J. Benton, Nir Lipovetzky, Eva Onaindia, David E. Smith, and Siddharth Srivastava. Vol. 29. 1. Berkeley, California, USA: PKP Publishing Services, July 2019, pp. 292–300.

- Obukhov, Sergey, Emad M. Ahmed, Denis Y. Davydov, Talal Alharbi, Ahmed Ibrahim, and Ziad M. Ali. “Modeling Wind Speed Based on Fractional Ornstein–Uhlenbeck Process”. In: *Energies* 14.5561 (2021), pp. 1–15. ISSN: 1996-1073. DOI: 10.3390/en14175561. O
- Ornstein, Donald. “On the existence of stationary optimal strategies”. In: *Proceedings of the American Mathematical Society* 20.2 (1969), pp. 563–569.
- Osband, Ian, Benjamin Van Roy, Daniel J. Russo, and Zheng Wen. “Deep Exploration via Randomized Value Functions”. In: *Journal of Machine Learning Research* 20.124 (2019). Ed. by P. Auer, pp. 1–62.
- Osband, Ian and Benjamin Van Roy. “Why is Posterior Sampling Better than Optimism for Reinforcement Learning?” In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Vol. 70. ICML’17. Sydney, NSW, Australia: JMLR.org, Aug. 2017, pp. 2701–2710.
- Puterman, Martin L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Hoboken, New Jersey, USA: John Wiley & Sons, Inc., 1994. xvii, 649 pp. ISBN: 0471619779. P
- Raffin, Antonin, Ashley Hill, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, and Noah Dormann. Stable Baselines 3. <https://github.com/DLR-RM/stable-baselines3>. 2019. R
- Rashid, Tabish, Bei Peng, Wendelin Böhmer, and Shimon Whiteson. “Optimistic Exploration Even with a Pessimistic Initialisation”. In: *International Conference on Learning Representations*. Addis Ababa, Ethiopia: OpenReview.net, Apr. 2020. URL: <https://openreview.net/forum?id=r1xGP6VYwH>.
- Rijksdienst voor Ondernemend Nederland (RVO). *Hollandse Kust Noord (Site B) Dataset* (HKNB). <https://offshorewind.rvo.nl/file/view/55040229/Processed+data+HKNB>. Accessed: 2021-09-30. Ministry of Economic Affairs and Climate Policy of the Netherlands, Aug. 2019.
- Romeijn, H. Edwin and Robert L. Smith. “Shadow Prices in Infinite-Dimensional Linear Programming”. In: *Mathematics of Operations Research* 23.1 (1998), pp. 239–256. DOI: 10.1287/moor.23.1.239.

- Romeijn, H. Edwin, Robert L. Smith, and James C. Bean. “Duality in Infinite Dimensional Linear Programming”. In: *Mathematical Programming* 53.1 (1992), pp. 79–97. ISSN: 1436-4646. DOI: 10.1007/BF01585695.
- Rott, A., B. Doekemeijer, J. K. Seifert, Jan-Willem van Wingerden, and M. Kühn. “Robust active wake control in consideration of wind direction variability and uncertainty”. In: *Wind Energy Science* 3.2 (2018), pp. 869–882. DOI: 10.5194/wes-3-869-2018.
- Rummery, Gavin A. and Mahesan Niranjan. *On-Line Q-Learning Using Connectionist Systems*. Vol. 37. Citeseer, 1994.
- Schepers, Jan G. and Sander P. van der Pijl. “Improved Modelling of Wake Aerodynamics and Assessment of New Farm Control Strategies”. In: *Journal of Physics: Conference Series* 75 (July 2007), p. 012039. DOI: 10.1088/1742-6596/75/1/012039.
- Scherrer, Bruno. “Improved and generalized upper bounds on the complexity of policy iteration”. In: *Advances in Neural Information Processing Systems* 26 (2013).
- *Performance Bounds for Lambda Policy Iteration and Application to the Game of Tetris*. 2007. DOI: 10.48550/ARXIV.0711.0694.
- Schreiber, Johannes, Emmanouil M. Nanos, Filippo Campagnolo, and Carlo L. Bottasso. “Verification and Calibration of a Reduced Order Wind Farm Model by Wind Tunnel Experiments”. In: *Journal of Physics: Conference Series* 854 (May 2017), p. 012041. DOI: 10.1088/1742-6596/854/1/012041.
- Schrittwieser, Julian, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, et al. “Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model”. In: *Nature* 588.7839 (2020), pp. 604–609. ISSN: 1476-4687. DOI: 10.1038/s41586-020-03051-4.
- Schulman, John, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. “Trust region policy optimization”. In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897.
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

- Sethi, Suresh P. and Gerhard Sorger. "A theory of rolling horizon decision making". In: *Annals of Operations Research* 29.1 (Dec. 1991), pp. 387–415. ISSN: 1572-9338.
- Shreve, Steven E. and Dimitri P. Bertsekas. "Dynamic programming in Borel spaces". In: *Dynamic programming and its applications*. Elsevier, 1978, pp. 115–130.
- Silver, David, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, et al. "Mastering the Game of Go with Deep Neural Networks and Tree Search". In: *Nature* 529 (Jan. 2016), pp. 484–489. DOI: 10.1038/nature16961.
- Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. DOI: 10.48550/ARXIV.1712.01815.
- Simão, Thiago D. and Matthijs T. J. Spaan. "Safe policy improvement with baseline bootstrapping in factored environments". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 4967–4974.
- Singh, Satinder, Tommi Jaakkola, Michael L. Littman, and Csaba Szepesvári. "Convergence results for single-step on-policy reinforcement-learning algorithms". In: *Machine learning* 38.3 (2000), pp. 287–308.
- Smith, Matt, Michael Harris, John Medley, and Chris Slinger. "Necessity is the Mother of Invention: Nacelle-mounted Lidar for Measurement of Turbine Performance". In: *Energy Procedia* 53 (2014), pp. 13–22. ISSN: 1876-6102. DOI: 10.1016/j.egypro.2014.07.211.
- Smith, Robert L. and Rachel Q. Zhang. "Infinite Horizon Production Planning in Time-Varying Systems with Convex Production and Inventory Costs". In: *Management Science* 44.9 (1998), pp. 1313–1320. DOI: 10.1287/mnsc.44.9.1313.
- Stanfel, P., K. Johnson, C. J. Bay, and J. King. "Proof-of-concept of a reinforcement learning framework for wind farm energy capture maximization in time-varying wind". In: *Journal of Renewable and Sustainable Energy* 13.4 (2021), p. 043305. DOI: 10.1063/5.0043091.

- Steinbuch, M., W. W. de Boer, O. H. Bosgra, S. A. W. M. Peters, and J. Ploeg. “Optimal control of wind power plants”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 27.1 (1988), pp. 237–246. ISSN: 0167-6105. DOI: 10.1016/0167-6105(88)90039-6.
- Strehl, Alexander L., Lihong Li, and Michael L. Littman. “Reinforcement Learning in Finite MDPs: PAC Analysis”. In: *Journal of Machine Learning Research* 10 (Dec. 2009). Ed. by S. Mahadevan, pp. 2413–2444. ISSN: 1532-4435.
- Strehl, Alexander L., Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. “PAC Model-Free Reinforcement Learning”. In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML’06. Pittsburgh, PA, USA: Association for Computing Machinery, 2006, pp. 881–888. ISBN: 1595933832.
- Sutton, Richard S. and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, Massachusetts, USA: The MIT Press, 2018. xxii, 526 pp. ISBN: 9780262039246. URL: <http://incompleteideas.net/book/the-book-2nd.html>.
- Szepesvári, Csaba and Rémi Munos. “Finite time bounds for sampling based fitted value iteration”. In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 880–887.
- Szita, István and András Lőrincz. “The many faces of optimism: a unifying approach”. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML’08. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 1048–1055. ISBN: 9781605582054.
- T Tang, Boxin. “Orthogonal array-based Latin hypercubes”. In: *Journal of the American statistical association* 88.424 (1993), pp. 1392–1397.
- Tang, Haoran, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. “# exploration: A study of count-based exploration for deep reinforcement learning”. In: *Advances in neural information processing systems* 30 (2017).
- Tesauro, Gerald. “Temporal difference learning and TD-Gammon”. In: *Communications of the ACM* 38.3 (1995), pp. 58–68. ISSN: 0001-0782. DOI: 10.1145/203330.203343.

Tijms, Henk C. *Analysis of (s, S) Inventory Models*. Tech. rep. Mathematisch Centrum Amsterdam, 1972.

United Nations. *Paris Agreement*. Treaty No. xxvii-7-d. C.N.63.2016.TREATIES-XXVII.7.d of 16 February 2016 (Opening for signature) and C.N.92.2016.TREATIES-XXVII.7.d of 17 March 2016 (Issuance of Certified True Copies). Dec. 2015. URL: https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27.

U

Vatiwutipong, Pat and Nattakorn Phewchean. “Alternative Way to Derive the Distribution of the Multivariate Ornstein–Uhlenbeck Process”. In: *Advances in Difference Equations* 2019.1, 276 (2019), pp. 1–7. ISSN: 1687-1847. DOI: 10.1186/s13662-019-2214-1.

V

Veinott Jr., Arthur F. “On the Optimality of (s, S) Inventory Policies: New Conditions and a New Proof”. In: *SIAM Journal on Applied Mathematics* 14.5 (1966), pp. 1067–1083.

Veinott Jr., Arthur F. and Harvey M. Wagner. “Computing Optimal (s, S) Inventory Policies”. In: *Management Science* 11.5 (1965), pp. 525–552.

Vermeer, Nord-Jan (L. J.), Jens N. Sørensen, and Antonio Crespo. “Wind Turbine Wake Aerodynamics”. In: *Progress in Aerospace Sciences* 39.6 (2003), pp. 467–510.

Verstraeten, Timothy, Pieter J. K. Libin, and Ann Nowé. “Fleet Control using Coregionalized Gaussian Process Policy Iteration”. In: *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*. Ed. by Giuseppe De Giacomo, Alejandro Catala, Bistra Dilkina, Michela Milano, Senen Barro, Alberto Bugarin, and Jerome Lang. Vol. 325. Frontiers in Artificial Intelligence and Applications. Netherlands: IOS Press, Aug. 2020, pp. 1571–1578. ISBN: 978-1-64368-100-9. DOI: 10.3233/FAIA200266.

Vinyals, Oriol, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, et al. *AlphaStar: Mastering the Real-Time Strategy Game StarCraft II*. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>. 2019.

Wagenaar, Jan W., Leo A. H. Machielse, and Jan G. Schepers. “Controlling Wind in ECN’s Scaled Wind Farm”. In: *Proc. Europe Premier Wind Energy Event* 1 (2012), pp. 685–694.

W

- Wal, Jan van der. *On Uniformly Nearly Optimal Stationary Strategies*. Memorandum COSOR 8111. Jan. 1981.
- Wang, Yuanhao, Kefan Dong, Xiaoyu Chen, and Liwei Wang. “Q-learning with UCB Exploration Is Sample Efficient for Infinite-Horizon MDP”. In: *International Conference on Learning Representations*. Addis Ababa, Ethiopia: OpenReview.net, Apr. 2020. URL: <https://openreview.net/forum?id=BkgISTNFDB>.
- Watkins, Christopher J. C. H. “Learning from Delayed Rewards”. PhD thesis. Cambridge, United Kingdom: King’s College, May 1989. vii, 234. URL: <https://www.cs.rhul.ac.uk/~chrisw/thesis.html>.
- Wessels, Jaap. “Markov programming by successive approximations with respect to weighted supremum norms”. In: *Journal of mathematical analysis and applications* 58.2 (1977), pp. 326–335.
- White, Douglas J. “Finite state approximations for denumerable state infinite horizon discounted Markov decision processes with unbounded rewards”. In: *Journal of Mathematical Analysis and Applications* 86.1 (1982), pp. 292–306.
- “Finite state approximations for denumerable-state infinite horizon contracted Markov decision processes: The policy space method”. In: *Journal of Mathematical Analysis and Applications* 72.2 (1979), pp. 512–523.
- “Finite-state approximations for denumerable-state infinite-horizon discounted Markov decision processes”. In: *Journal of Mathematical Analysis and Applications* 74.1 (1980), pp. 292–295.
- “Isotone optimal policies for structured Markov decision processes”. In: *European Journal of Operational Research* 7.4 (1981), pp. 396–402.
- Y
Yu, Huizhen and Dimitri P. Bertsekas. “A Mixed Value and Policy Iteration Method for Stochastic Control with Universally Measurable Policies”. In: *Mathematics of Operations Research* 40.4 (2015), pp. 926–968. ISSN: 0364765X, 15265471.

Acknowledgements

This book would have never been possible without the amazing people in my life to whom I would like to express my gratitude.

First and foremost, this endeavor would not have been possible without my promotors Prof. dr. Mathijs de Weerdt and Dr. ir. Remco Verzijlbergh. Both of them have always had faith in me—even when I could not find it myself—and showed me the amount of support, trust, and patience that I could only dream of. I'm honored to have travelled this journey under their guidance.

Next, I would like to thank the other members of the defense committee: prof. dr. ir. Karen Aardal, prof. dr. Ann Nowé, dr. Herke van Hoof, dr. Michael Kaisers (with whom I also had a pleasure of working at CWI), dr. Matthijs Spaan, and prof. dr. ir. Bart De Schutter. Thank you for your interest in my work and the insightful comments and questions.

I would like to thank dr. Scott Sanner from University of Toronto for all the advice and encouragement (and reviewing requests!) he gave me during my PhD as my mentor.

My special thanks to prof. dr. Andrei Letchikov, whom I first met at one of the mathematical contests for middle and high school students in Udmurtia that he used to organize and I used to participate in. Later, I became a student at his department at Udmurt State University, and later still, started working there. Without him, my academic career would not have been the same. I would also like to express my gratitude to the other professors and lecturers I had a privilege to study under and work with at Udmurt State University: Valentina Shulikovskaya, Anastasia Merzlyakova, Aleksei Lashkarev, Leonid Romanov, Irina Korepanova, Galina Slesarenko, and many others.

When my life brought me to CWI, I was blessed to have Felix Claessen as my officemate. He taught me a lot about the Dutch society, academia, and the beautiful city I now live in. Later, we were joined by Iliana Pappi and Shantanu Chakraborty. Our office was always full of both academic discussions and fun small talk, both of which I enjoyed immensely.

Many thanks to my fellow PhD students from the Algorithmics group in Delft. Thiago Dias Simão, Canmanie Ponnambalam, Anna Stawska, and Qisong Yang all started their PhDs around the same time as me; with all of them we have been through thick and thin; all of them are dear friends to me. Erwin Walraven, Rens Philipsen, Grigorii Vevurko, Junhan Wen, and Leonard Volarić Horvat used to be my officemates in Delft at different times; as such, they offered me help with and respite from my research—both often needed so badly! Finally, I would like to thank my other colleagues for inspiring and thought-provoking conversations, including Sytze Andrijga, Koos van der Linden, Lei He, Longjian Piao, Natalia Romero Lane, and Wendelin Böhmer among others.

Words cannot express my gratitude to my best friends, the “bored game people”—Tim Baarslag, Evgeny Rezunenko, Brinn Hekkelman, Zerline Henning, Christina Katsimerou, and Valentin Robu—who are rooting for me no matter what. I am also grateful to the many other friends I have met in the Netherlands, including Marie-Claire Dangerfield, Mariane Urias, Carmel Freeman, Marloes Kunst, Lotte Baltussen, Hay Kranen, Adrian Ajami, Arno Jägers, Trevor Grahl, and Ginger da Silva.

I thank my dear friends in Russia and all over the globe, who always tell me how proud they are of me and all of whom I am immensely proud of myself. First of all, my dear Vasilyev clan: Alexander, Ekaterina, Vladimir, and Natalya, but also Marianne McPherson and Damon Sidel, Natalia Nekludova, Irina and Dmitriy Kolesnikov, Mikhail Stepanov and Tatiana Nizhegorodova, Mariia Koroleva, Anton Orlov, Svetlana Lepikhina, Faina Blinova, Dana Bessolitsyna, Pablo Hernandez-Leal, Mariya Galimova, Diana Suleymanova, Roza Badaeva, and many others. I hope we have a chance to see each other again soon.

Finally, I would like to express my deepest appreciation to the Burgoyne family: Nancy, John, Adam, and especially Ashley, whose belief in me was my motivation all these years. Thank you all for making me part of your family. As for my own family, I would like to address them in Russian now.

Спасибо моим дорогим маме Свете и сестре Соне, а также (двоюродному) брату Орхану, его жене Алёне и нашей тёте Марине за их любовь и поддержку, которые часто так нужны были мне. Спасибо тёте Оле, Давиду и бабушке Рае, тёте Наде и дяде Лёне, Славе, Наташе и Лёше за то, что вы есть у меня.

G. N.

Curriculum Vitæ

≡ Grigory NEUSTROEV ≡

born in Ustinov, USSR on August 28, 1986

EDUCATION

- 1993–2003 *General education*
Municipal school № 16, Izhevsk, Russia
- 2003–2008 *Specialist, mathematical economist*
Udmurt State University, Izhevsk, Russia

WORK EXPERIENCE

- 2008–2009 *Lead specialist in international
documentary operations*
Bank ВТБ, Izhevsk, Russia
- 2009–2015 *Specialist in distance learning*
Udmurt State University, Izhevsk, Russia
- 2010–2016 *Lecturer*
Udmurt State University, Izhevsk, Russia
- 2015–2016 *Software developer*
ELMA, Izhevsk, Russia
- 2016–2017 *Project researcher*
Dutch national research institute for mathematics
and computer science (Centrum Wiskunde &
Informatica, CWI) Amsterdam, the Netherlands
- 2017–2021 *Doctoral researcher*
Delft University of Technology, the Netherlands
- 2021– *Postdoctoral researcher*
Delft University of Technology, the Netherlands

List of Publications

RELATED TO THIS THESIS

6. Grigory Neustroev and Mathijs M. de Weerdt. “Action Elimination in Countably-Infinite Markov Decision Processes”. To appear; based on Chapter 4 of this thesis. Manuscript in progress. 2022.
5. Grigory Neustroev, Sytze P. E. Andringa, Remco A. Verzijlbergh, and Mathijs M. de Weerdt. “Deep Reinforcement Learning for Active Wake Control”. In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. Ed. by Piotr Faliszewski, Viviana Mascardi, Catherine Pelachaud, and Matthew E. Taylor. Auckland, New Zealand, virtual event: International Foundation for Autonomous Agents and Multiagent Systems, May 2022, pp. 944–953.
4. Grigory Neustroev and Mathijs M. de Weerdt. “Generalized Optimistic Q-Learning with Provable Efficiency”. In: *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*. Ed. by Bo An, Neil Yorke-Smith, Amal El Fallah Seghrouchni, and Gita Sukthankar. Auckland, New Zealand, virtual event: International Foundation for Autonomous Agents and Multiagent Systems, May 2020, pp. 913–921.
3. Grigory Neustroev, Mathijs M. de Weerdt, and Remco A. Verzijlbergh. “Discovery of Optimal Solution Horizons in Non-Stationary Markov Decision Processes with Unbounded Rewards”. In: *Proceedings of the 29th International Conference on Automated Planning and Scheduling*. Ed. by J. Benton, Nir Lipovetzky, Eva Onaindia, David E. Smith, and Siddharth Srivastava. Vol. 29. 1. Berkeley, California, USA: PKP Publishing Services, July 2019, pp. 292–300.

OTHER

2. Grigory Neustroev, Canmanie T. Ponnambalam, Mathijs M. de Weerdt, and Matthijs T. J. Spaan. “Interval Q-Learning: Balancing

Deep and Wide Exploration”. In: *19th International Conference on Autonomous Agents and Multiagent Systems*. Adaptive and Learning Agents Workshop. Ed. by Roxana Radulescu, Felipe Leno da Silva, Fernando P. Santos, and Patrick MacAlpine. 7 pp. Auckland, New Zealand, virtual event, May 2020.

1. Stanislav Vladimirovich Puchkovskiy, Marina Stanislavovna Bui-novskaya, Dzhamal Karimovna Voronetskaya, and Grigory Vladimirovich Neustroev. “On the Studies of Marking Behavior of Brown Bear in Terms of Tree Diameter Selectivity”. In: *Contemporary Problems of Ecology* 5.1 (2012), pp. 104–109. DOI: 10.1134/S1995425512010146.

Appendices



Miscellaneous Proofs

THIS APPENDIX contains proofs of auxiliary results presented in this thesis. Section A.1 contains proofs of equivalence of non-stationary MDPs to countably-infinite ones. Section A.2 presents the proof that any flow-conserving function induces a policy. Section A.2 concerns feasible solutions of the occupancy-based linear program. Finally, Section A.4 contains proofs of various properties of the multi-product inventory management problem.

A.1 PROOFS OF TIME AUGMENTATION EQUIVALENCE

Lemma 2.7 * stationary reformulation equivalence

All policies of a non-stationary MDP $\mathfrak{M}_T, T \leq \infty$ can be represented in time-augmented version $\tilde{\mathfrak{M}}_T$ by policies of the same gain, assuming that the gain is well-defined.

Proof. Consider the sample space $\tilde{\Omega}_T = \tilde{\mathbb{S}} \times \prod_{i=1}^T (\mathbb{A} \times \tilde{\mathbb{S}})$ of the augmented MDP $\tilde{\mathfrak{M}}_T$. It consists of sample paths $\tilde{\omega}$ of the following structure:

$$\tilde{\omega} = (S_0, \tau_0, A_0, S_1, \tau_1, A_1, \dots, S_{T-1}, \tau_{T-1}, A_{T-1}, S_T, \tau_T).$$

Let \tilde{h}_t denote the history mapping in the augmented problem. Let $\tilde{h}_t^{\downarrow \tau_t}$ denote the history with all of the times τ_t removed:

$$(S_0, \tau_0, A_0, S_1, \tau_1, A_1, \dots, S_t, \tau_t)^{\downarrow \tau_t} = (S_0, A_0, S_1, A_1, \dots, S_t),$$

and similarly for sample paths. For any policy $\pi \in \Pi$, choose an augmented policy $\tilde{\pi}$ as

$$\tilde{\pi}_t(a|\tilde{h}_t) = \pi_t(a|\tilde{h}_t^{\downarrow t}). \quad (\text{A.1})$$

By this construction, every policy in the original problem \mathfrak{M}_T has a counterpart in the augmented problem $\tilde{\mathfrak{M}}_T$.

We now show that the gains of these two policies are equal.

We begin with the probability measure $\tilde{P}_{\tilde{\pi}}$ induced by the augmented policy $\tilde{\pi}$. It is equal to

$$\begin{aligned} \text{by (2.4)} \quad \tilde{P}_{\tilde{\pi}}(\tilde{\omega}) &= \tilde{\alpha}(\tilde{S}_0) \cdot \tilde{\pi}_0(A_0|\tilde{h}_0) \cdot \tilde{p}_0(\tilde{S}_1|\tilde{S}_0, A_0) \\ &\quad \cdot \tilde{\pi}_1(\tilde{A}_1|\tilde{h}_1) \cdot \tilde{p}_1(\tilde{S}_2|\tilde{S}_1, A_1) \cdot \dots \\ \text{by (2.8), (2.9), and (A.1)} \quad &= \delta_{\tau_0,0} \cdot \alpha(S_0) \cdot \pi_0(A_0|S_0) \cdot \delta_{\tau_0+1,\tau_1} \cdot p_0(S_1|S_0, A_0) \\ &\quad \cdot \pi_1(A_1|S_0, A_0, S_1) \cdot \delta_{\tau_1+1,\tau_2} \cdot p_1(S_2|S_1, A_1) \cdot \dots \\ \text{For the product of Kronecker deltas to be non-zero, } \tau_0 \text{ must be equal to 0. Then } \tau_1 \text{ must be equal to } \tau_0 + 1 = 1 \text{ and so forth. Instead of sequentially comparing } \tau_{t+1} \text{ to each of the previous values } \tau_t, \text{ we rewrite the product by comparing them to the values of time directly.} \quad &= P_{\pi}(\tilde{\omega}^{\downarrow t}) \cdot \delta_{\tau_0,0} \cdot \prod_{t=0}^{T-1} \delta_{\tau_t+1,\tau_{t+1}} = P_{\pi}(\tilde{\omega}^{\downarrow t}) \cdot \prod_{t=0}^{T-1} \delta_{\tau_t,t}. \quad (\text{A.2}) \end{aligned}$$

Let $\tilde{\Omega}_T^{\rightarrow}$ be the set of all sample paths where all the times τ_t occur sequentially:

$$\tilde{\Omega}_T^{\rightarrow} = \{\tilde{\omega} \in \tilde{\Omega} \mid \tilde{\omega} = (S_0, 0, A_0, S_1, 1, A_1, \dots, S_{T-1}, T-1, A_{T-1}, S_T, T)\}.$$

For any sample path $\tilde{\omega} \in \tilde{\Omega}_T^{\rightarrow}$ in this set, the product $\prod_{t=0}^{T-1} \delta_{\tau_t,t}$ is equal to one. For all other sample paths it is equal to zero. Thus, the augmented probability measure $\tilde{P}_{\tilde{\pi}}(\tilde{\omega})$ is equal to the original one $\tilde{P}_{\pi}(\omega)$ for all $\tilde{\omega} \in \tilde{\Omega}_T^{\rightarrow}$ and $\omega = \tilde{\omega}^{\downarrow t}$.

Therefore, the gain of the augmented policy $J(\tilde{\pi})$ is equal to

$$\begin{aligned} J(\tilde{\pi}) &= \sum_{\tilde{\omega} \in \tilde{\Omega}_T} \gamma^t \cdot \tilde{r}(S_t(\tilde{\omega}), \tau_t(\tilde{\omega}), A_t(\tilde{\omega})) \cdot \tilde{P}_{\tilde{\pi}}(\tilde{\omega}) \\ \text{by removing zero summands} \quad &= \sum_{\tilde{\omega} \in \tilde{\Omega}_T^{\rightarrow}} \gamma^t \cdot \tilde{r}(S_t(\tilde{\omega}), t, A_t(\tilde{\omega})) \cdot \tilde{P}_{\tilde{\pi}}(\tilde{\omega}) \\ \text{by (A.2)} \quad &= \sum_{\tilde{\omega} \in \tilde{\Omega}_T^{\rightarrow}} \gamma^t \cdot r_t(S_t(\tilde{\omega}^{\downarrow t}), A_t(\tilde{\omega}^{\downarrow t})) \cdot P_{\pi}(\tilde{\omega}^{\downarrow t}). \end{aligned}$$

The right-hand side uses only augmented sample paths with the times removed from them. Since there is only one combination of times in $\tilde{\Omega}_T^{\rightarrow}$, we can remove the times from the definition of the sample space as well, and each summand will still appear exactly once. Therefore,

$$\begin{aligned} \text{The switch of the summation index set from } \tilde{\Omega}_T^{\rightarrow} \text{ to } \Omega_T \text{ is only possible for well-defined gains.} \quad &J(\tilde{\pi}) = \sum_{\omega \in \Omega_T} \gamma^t \cdot r_t(S_t(\omega), A_t(\omega)) \cdot P_{\pi}(\omega) = J(\pi), \end{aligned}$$

and the gains of the augmented policy $\tilde{\pi}$ and the original policy π are the same. QED

Corollary 2.8 * optimality in non-stationary ∞ -horizon MDPs

Let Conditions 2.1 and 2.3 hold for a non-stationary infinite-horizon Markov decision process \mathfrak{M}_∞ with a countable state space, $|\mathbb{S}| \leq \infty$. Then there exists a deterministic Markovian policy $\pi \in \Pi_{\text{DM}}$ that is optimal.

Proof. By Proposition 2.6, the augmented version $\tilde{\mathfrak{M}}_T, T \leq \infty$ of a non-stationary MDP \mathfrak{M}_T has an optimal deterministic stationary policy $\tilde{\pi}_\star \in \tilde{\mathbb{D}}$ with some gain J_\star . Consider a policy $\pi_{\star,t}(a|s) \triangleq \tilde{\pi}_\star(a|s, t)$. By construction (A.1), its augmentation is equal to

$$\tilde{\pi}_{\star,t}(a|s, \tau_t) = \pi_{\star,t}(a|s) = \tilde{\pi}_\star(a|s, t) \quad \text{if } \tau_t = t.$$

► instead of the full history, only its last element is required

Thus, the augmented version of $\pi_{\star,t}(a|s)$ may differ from the optimal augmented policy $\tilde{\pi}_\star$ when $\tau_t \neq t$. At the same time, all such cases happen with zero probability, because $\delta_{\tau_t, t}$ appears in the construction of the probability measure (A.2) and it is equal to zero when $\tau_t \neq t$. Therefore,

$$\tilde{\pi}_{\star,t}(a|s, \tau_t) \stackrel{\text{a.s.}}{=} \tilde{\pi}_\star$$

and their induced probability measures are equivalent. The policy $\pi_{\star,t}(a|s)$ has the same gain as the optimal augmented policy J_\star .

Assume that J_\star is the optimal gain in the augmented problem $\tilde{\mathfrak{M}}_T$, but not in the original one \mathfrak{M}_T . In this case there is a policy π' with a gain J' such that $J' > J_\star$. But that means that the augmented policy $\tilde{\pi}'$ has the same gain J' . Therefore, in the augmented problem $\tilde{\mathfrak{M}}_T$ the policy $\tilde{\pi}_\star$ is not optimal. By contradiction, J_\star is the optimal gain in the original problem \mathfrak{M}_T as well. Since $\pi_{\star,t}$ achieves this gain, it is optimal. QED

A.2 PROOF THAT FLOW CONSERVATION INDUCES POLICIES

Theorem 2.14 * flow conservation induces policies

Given a stationary infinite-horizon MDP \mathfrak{M}_∞ with a discrete admissible control space \mathbb{X} , consider an absolutely-summable non-negative function $f \in L^1(\mathbb{X}), f \geq 0$. If the function f is a flow-conserving occupancy function, then the occupancy function of the policy $\pi^f = \mathcal{Z}f$ induced by the function f is equal to f , $z_{\pi^f} = f$, and therefore the function f is an occupancy function.

Proof. First, by Definition 2.29, we can express the occupancies z_{π^f} of policy π^f for arbitrary chosen $s'' \in \mathbb{S}$ and $a'' \in \mathbb{A}$ as

$$\begin{aligned}
z_{\pi^f}(s'', a'') &= \mathbb{E}_{\pi^f} \left[\sum_{t=0}^{\infty} \gamma^t \cdot \delta_{S_t, s''} \cdot \delta_{A_t, a''} \right] = \sum_{t \in \mathbb{T}} \gamma^t \cdot \mathbb{E}_{\pi^f} [\delta_{S_t, s''} \cdot \delta_{A_t, a''}] \\
&= \sum_{t=0}^{\infty} \gamma^t \cdot \sum_{s \in \mathbb{S}} \alpha(s) \cdot \sum_{s' \in \mathbb{S}} p_{\pi^f}^t(s' | s) \\
&\quad \cdot \sum_{a' \in A_p(s')} \pi^f(a' | s') \cdot \delta_{s', s''} \cdot \delta_{a', a''} \\
&= \sum_{t=0}^{\infty} \gamma^t \cdot \sum_{s \in \mathbb{S}} \alpha(s) \cdot \sum_{s' \in \mathbb{S}} p_{\pi^f}^t(s' | s) \cdot \pi^f(a'' | s') \cdot \delta_{s', s''} \\
&= \sum_{t=0}^{\infty} \gamma^t \cdot \sum_{s \in \mathbb{S}} \alpha(s) \cdot p_{\pi^f}^t(s'' | s) \cdot \pi^f(a'' | s'') \\
&= \sum_{s \in \mathbb{S}} \alpha(s) \cdot \sum_{t=0}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s) \cdot \pi^f(a'' | s'').
\end{aligned}$$

Next, we express the initial state distribution α in terms of the function z . If the state occupancy is zero, $z(s) = 0$, then the occupancy for all actions $a \in A_p(s)$ permitted in that state must be zero as well, $z(s, a) = 0$. This follows directly from (2.22) and the non-negativity assumption. From the flow-conservation recurrence,

$$\begin{aligned}
\alpha(s') &= z(s') - \gamma \cdot \sum_{s \in \mathbb{S}} \sum_{a \in A_p(s)} z(s, a) \cdot p(s' | s, a) \\
\text{by removing zero} &\quad = z(s') - \gamma \cdot \sum_{s \in \text{supp } z} \sum_{a \in A_p(s)} z(s, a) \cdot p(s' | s, a) \\
\text{summands} &\quad \\
\text{by (2.22) and } z(s) \neq 0 &\quad = z(s') - \gamma \cdot \sum_{s \in \text{supp } z} \sum_{a \in A_p(s)} z(s, a) \cdot p(s' | s, a) \cdot \frac{z(s)}{\sum_{a'' \in A_p(s)} z(s, a'')} \\
&\quad \\
\text{reordering} &\quad = z(s') - \gamma \cdot \sum_{s \in \text{supp } z} \sum_{a \in A_p(s)} z(s) \cdot \frac{z(s, a)}{\sum_{a'' \in A_p(s)} z(s, a'')} \cdot p(s' | s, a) \\
&\quad \\
\text{by (2.23)} &\quad = z(s') - \gamma \cdot \sum_{s \in \text{supp } z} \sum_{a \in A_p(s)} z(s) \cdot \pi^f(a | s) \cdot p(s' | s, a) \\
&\quad \\
\text{by (2.2)} &\quad = z(s') - \gamma \cdot \sum_{s \in \text{supp } z} z(s) \cdot p_{\pi^f}(s' | s) \\
&\quad \\
\text{by adding zero} &\quad = z(s') - \gamma \cdot \sum_{s \in \mathbb{S}} z(s) \cdot p_{\pi^f}(s' | s). \\
\text{summands} &\quad
\end{aligned}$$

Therefore,

$$\begin{aligned}
z_{\pi^f}(s'', a'') &= \sum_{s' \in \mathbb{S}} \left(z(s') - \gamma \cdot \sum_{s \in \mathbb{S}} z(s) \cdot p_{\pi^f}(s' | s) \right) \\
&\quad \cdot \sum_{t=0}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'')
\end{aligned}$$

$$\begin{aligned}
&= \sum_{s' \in \mathbb{S}} z(s') \cdot \sum_{t=0}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'') \\
&\quad - \sum_{s' \in \mathbb{S}} \gamma \cdot \sum_{s \in \mathbb{S}} z(s) \cdot p_{\pi^f}(s' | s) \\
&\quad \cdot \sum_{t=0}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'').
\end{aligned}$$

► By expanding the brackets. This is possible by Proposition 2.3, because the quantity in the brackets is equal to $\alpha(s')$ and therefore is non-negative.

Consider the subtrahend only. It can be simplified to

$$\begin{aligned}
&\sum_{s' \in \mathbb{S}} \gamma \cdot \sum_{s \in \mathbb{S}} z(s) \cdot p_{\pi^f}(s' | s) \cdot \sum_{t=0}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'') \\
&= \sum_{t=0}^{\infty} \gamma^{t+1} \cdot \sum_{s \in \mathbb{S}} z(s) \cdot \sum_{s' \in \mathbb{S}} p_{\pi^f}(s' | s) \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'') \\
&= \sum_{s \in \mathbb{S}} z(s) \cdot \sum_{t=1}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s) \cdot \pi^f(a'' | s'') \\
&= \sum_{s' \in \mathbb{S}} z(s') \cdot \sum_{t=1}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'').
\end{aligned}$$

► Changing the summation order is possible because all of the values are non-negative.
(A.3) ► by changing indexing variable from s to s'

Therefore,

$$\begin{aligned}
z_{\pi^f}(s'', a'') &= \sum_{s' \in \mathbb{S}} \left(z(s') - \gamma \cdot \sum_{s \in \mathbb{S}} z(s) \cdot p_{\pi^f}(s' | s) \right) \\
&\quad \cdot \sum_{t=0}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'') \\
&= \sum_{s' \in \mathbb{S}} z(s') \cdot \sum_{t=0}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'') \\
&\quad - \sum_{s' \in \mathbb{S}} z(s') \cdot \sum_{t=1}^{\infty} \gamma^t \cdot p_{\pi^f}^t(s'' | s') \cdot \pi^f(a'' | s'') \\
&= \sum_{s' \in \mathbb{S}} z(s') \cdot \delta_{s'', s'} \cdot \pi^f(a'' | s').
\end{aligned}$$

► by (A.3)

If the state s'' is not in the support of the function z , then $z(s'') = 0$ and $\sum_{s' \in \mathbb{S}} z(s') \cdot \delta_{s'', s'} = 0$ because all of the summands are equal to zero. Therefore, $z_{\pi^f}(s'', a'') = z(s'', a'') = 0$ no matter what the policy is. Otherwise, $\sum_{s' \in \mathbb{S}} z(s') \cdot \delta_{s'', s'} = z(s'')$ and the occupancy $z_{\pi^f}(s'', a'')$ can be simplified as follows:

$$\begin{aligned}
z_{\pi^f}(s'', a'') &= z(s'') \cdot \pi^f(a'' | s'') \\
&= \sum_{a'' \in A_p(s'')} z(s'', a'') \cdot \frac{z(s'', a'')}{\sum_{a \in A_p(s'')} z(s'', a)} = z(s'', a'').
\end{aligned}$$

In both cases, $z_{\pi^f}(s'', a'') = z(s'', a'')$.

QED

A.3 PROOF OF FEASIBLE REGION EMBEDDING

Lemma 2.21 * feasible region embedding

Under Conditions 2.4, 2.5, and 2.6, the feasible region of the primal program (P) is a subset of the space $L_*^w(\mathbb{X})$ of functions with finite w -weighted supremum norm $\|\cdot\|_w$.

Proof. First, note the following two properties of inner products:

$$\begin{aligned} \langle z, z' \rangle_{\mathbb{X}} &= \sum_{(s,a) \in \mathbb{X}} z(s,a) \cdot z'(s,a) = \sum_{s \in \mathbb{S}} \sum_{a \in A_p(s)} z(s,a) \cdot z'(s,a) \\ &\leq \sum_{s \in \mathbb{S}} \left(\sum_{a \in A_p(s)} z(s,a) \right) \cdot \left(\sum_{a \in A_p(s)} z'(s,a) \right) \\ &= \langle \mathcal{N}_* z, \mathcal{N}_* z' \rangle_{\mathbb{S}}, \end{aligned} \quad (\text{A.4})$$

$$\begin{aligned} \langle \mathcal{N}_* \mathcal{N} z, z' \rangle_{\mathbb{S}} &= \sum_{s \in \mathbb{S}} \left(\sum_{a \in A_p(s)} z(s) \right) \cdot z'(s) \\ &= \sum_{s \in \mathbb{S}} |A_p(s)| \cdot z(s) \cdot z'(s) \leq |\mathbb{A}| \cdot \langle z, z' \rangle_{\mathbb{S}}. \end{aligned} \quad (\text{A.5})$$

Next, the feasible region is given by the constraint $\mathcal{N}_* y - \gamma \cdot \mathcal{T}_* y = \alpha$ or alternatively $\mathcal{N}_* y = \gamma \cdot \mathcal{T}_* y + \alpha$. Then, by Definition 2.33 and linearity of the operators,

$$\begin{aligned} \langle y, \mathcal{N} w \rangle_{\mathbb{X}} &= \langle \mathcal{N}_* y, w \rangle_{\mathbb{S}} = \langle \gamma \cdot \mathcal{T}_* y + \alpha, w \rangle_{\mathbb{S}} \\ &= \gamma \cdot \langle \mathcal{T}_* y, w \rangle_{\mathbb{S}} + \langle \alpha, w \rangle_{\mathbb{S}} = \langle y, \gamma \cdot \mathcal{T} w \rangle_{\mathbb{X}} + \langle \alpha, w \rangle_{\mathbb{S}} \\ &\leq \gamma \cdot \langle \mathcal{N}_* y, \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{S}} + \langle \alpha, w \rangle_{\mathbb{S}} \\ &= \gamma \cdot \langle \gamma \cdot \mathcal{T}_* y + \alpha, \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{S}} + \langle \alpha, w \rangle_{\mathbb{S}} \\ &= \gamma^2 \cdot \langle y, \mathcal{T} \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{X}} + \gamma \cdot \langle \alpha, \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{S}} + \langle \alpha, w \rangle_{\mathbb{S}} \\ &= \gamma^2 \cdot \langle y, \mathcal{T} \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{X}} + \gamma \cdot \langle \mathcal{N} \alpha, \mathcal{T} w \rangle_{\mathbb{X}} + \langle \alpha, w \rangle_{\mathbb{S}} \\ &\leq \gamma^2 \cdot \langle y, \mathcal{T} \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{X}} + \gamma \kappa \cdot \langle \mathcal{N} \alpha, \mathcal{N} w \rangle_{\mathbb{X}} + \langle \alpha, w \rangle_{\mathbb{S}} \\ &= \gamma^2 \cdot \langle y, \mathcal{T} \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{X}} + \gamma \kappa \cdot \langle \mathcal{N}_* \mathcal{N} \alpha, w \rangle_{\mathbb{S}} + \langle \alpha, w \rangle_{\mathbb{S}} \\ &\leq \gamma^2 \cdot \langle y, \mathcal{T} \mathcal{N}_* \mathcal{T} w \rangle_{\mathbb{X}} + (\gamma \kappa |\mathbb{A}| + 1) \cdot \langle \alpha, w \rangle_{\mathbb{S}}. \end{aligned}$$

Repeating this process $v - 2$ more times we obtain

$$\begin{aligned} \langle y, \mathcal{N} w \rangle_{\mathbb{X}} &\leq \gamma^v \cdot \langle y, (\mathcal{T} \mathcal{N}_*)^{v-1} \mathcal{T} w \rangle_{\mathbb{X}} + \sum_{i=0}^{v-1} (\gamma \kappa |\mathbb{A}|)_i \cdot \langle \alpha, w \rangle_{\mathbb{S}} \\ &\leq \lambda \cdot \langle y, w \rangle_{\mathbb{X}} + C \cdot \langle \alpha, w \rangle_{\mathbb{S}}, \end{aligned}$$

where $C \triangleq \sum_{i=0}^{v-1} (\gamma \kappa |\mathbb{A}|)_i$ is a finite constant. Thus,

$$\begin{aligned} (1 - \lambda) \cdot \langle y, \mathcal{N} w \rangle_{\mathbb{X}} &\leq C \cdot \langle \alpha, w \rangle_{\mathbb{S}} \quad \text{and} \\ \langle y, \mathcal{N} w \rangle_{\mathbb{X}} &\leq \frac{C}{1 - \lambda} \cdot \langle \alpha, w \rangle_{\mathbb{S}} < \infty. \quad \text{QED} \end{aligned}$$

A.4 PROOFS FOR INVENTORY MANAGEMENT PROBLEM

A.4.1 Proof that Rewards Are Unbounded

Lemma 2.23 * unbounded rewards in inventory management

If at least one holding cost h_i is positive, there exists no uniform reward bound in the multi-product inventory management problem:

$$\sup_{(\mathbf{s}, \mathbf{a}) \in \mathbb{X}} |r(\mathbf{s}, \mathbf{a})| = \infty.$$

Proof. We prove this statement by showing that for any constant $w \in \mathbb{R}_+$, there exists a state \mathbf{s} and action \mathbf{a} such that $r(\mathbf{s}, \mathbf{a}) < -w$ and therefore $|r(\mathbf{s}, \mathbf{a})| > w$ for any w .

Let us assume that no order is placed, $\mathbf{a} = \mathbf{0}$. In this case,

$$r(\mathbf{s}, \mathbf{0}) = G(\mathbf{s}, \mathbf{0}) - H(\mathbf{s}, \mathbf{0}) - O(\mathbf{0}) = G(\mathbf{s}, \mathbf{0}) - \langle \mathbf{h}, \mathbf{s} \rangle \leq C_G - \langle \mathbf{h}, \mathbf{s} \rangle.$$

Choose an arbitrary product k with a positive holding cost h^k . Let the inventory of each other product be zero, $s_i = 0$ if $i \neq k$. For the k -th product, consider an inventory s^k that is greater than $(w + C_G)/h^k$, for example, let $s^k = \lfloor (w + C_G)/h^k \rfloor + 1$. For this state, the expected immediate reward $r(\mathbf{s}, \mathbf{0})$ is bounded from above by

$\lfloor a \rfloor + 1$ is the smallest integer greater than a .

$$r(\mathbf{s}, \mathbf{0}) \leq C_G - \sum_{i=0}^{n-1} h_i \cdot s_i = C_G - h_k \cdot s_k < C_G - h_k \cdot \frac{w + C_G}{h_k} = -w. \quad \text{QED} \triangleright s_i = 0 \text{ if } i \neq k$$

A.4.2 Proof of Weight Function Existence

Lemma 2.24 * weight function in inventory management

In the multi-product inventory management problem, let C_G , C_O , and C_H denote the expected revenue when the inventory is infinite, the maximum cost of placing an order and holding it, and the maximum cost of holding an order.

$$C_G \triangleq \langle \mathbf{c}, \mathbf{d} \rangle \quad \text{for all } (\mathbf{s}, \mathbf{a}) \in \mathbb{X}, \quad (2.39)$$

$$C_O \triangleq o_f + M \cdot \max_{0 \leq i < n} \frac{h_i + o_{v,i}}{m_i}, \quad (2.40)$$

$$C_H \triangleq M \cdot \max_{0 \leq i < n} \frac{h_i}{m_i}. \quad (2.41)$$

If the expected demands \mathbf{d} are finite, then Condition 2.5 is satisfied with the weight function w , the one-stage expansion coefficient κ , the contraction horizon v and the v -stage contraction coefficient λ

given by

$$w(\mathbf{s}) \triangleq \langle \mathbf{h}, \mathbf{s} \rangle + w_0, \quad \kappa \triangleq \gamma \cdot (1 + C),$$

$$v \triangleq \begin{cases} 1, & \text{if } \kappa < 1, \\ \left\lfloor \frac{W_{-1}(C^{-1}\gamma^{1/C} \ln \gamma)}{\ln \gamma} - \frac{1}{C} \right\rfloor + 1, & \text{if } \kappa \geq 1, \end{cases} \quad \lambda \triangleq \gamma^v \cdot (1 + Cv),$$

where $w_0 = \max\{C_G, C_O\}$, $C \triangleq C_H/w_0$, and W_k is the k -th branch of the Lambert w -function.

Proof. Indeed, if the expected demands \mathbf{d} are finite, the expected revenue when the inventory of each product is infinite is equal to $C_G \triangleq \langle \mathbf{c}, \mathbf{d} \rangle$ and is also finite. When the inventory is finite, the expected revenue can not exceed C_G ,

$$G(\mathbf{s}, \mathbf{a}) < C_G \triangleq \langle \mathbf{c}, \mathbf{d} \rangle \quad \text{for all } (\mathbf{s}, \mathbf{a}) \in \mathbb{X}. \quad (\text{A.6})$$

We can formally prove this statement as follows. Let $g_i(s_i, a_i)$ denote the expected sales of the i -th product when the total inventory of that product is equal to $s_i + a_i$.

$$\begin{aligned} g_i(s_i, a_i) &= \sum_{s' \in \mathbb{S}} p_i(s'_i | s_i, a_i) \cdot \max\{0, s_i + a_i - s'_i\} \\ \text{by (2.34)} \quad &= \sum_{k=1}^{s_i+a_i} (s_i + a_i - k) \cdot p_i(s_i + a_i - k) + (s_i + a_i) \cdot q_i(s_i + a_i) \\ \text{using } j = s_i + a_i - k \quad &= \sum_{j=0}^{s_i+a_i-1} j \cdot p_i(j) + (s_i + a_i) \cdot q_i(s_i + a_i) \\ \text{by definition of } q_i \quad &= \sum_{j=0}^{s_i+a_i-1} j \cdot p_i(j) + \sum_{j=s_i+a_i}^{\infty} (s_i + a_i) \cdot p_i(j) \\ &\leq \sum_{j=0}^{s_i+a_i-1} j \cdot p_i(j) + \sum_{j=s_i+a_i}^{\infty} j \cdot p_i(j) = \sum_{j=0}^{\infty} j \cdot p_i(j) = d_i. \end{aligned} \quad (\text{A.7})$$

The total expected revenue $G(\mathbf{s}, \mathbf{a}) = \langle \mathbf{c}, g(\mathbf{s}, \mathbf{a}) \rangle$ is then indeed bounded from above by a constant $C_G = \langle \mathbf{c}, \mathbf{d} \rangle$.

Using this bound, we can see that the rewards are bounded from above by

$$\text{both } H(\mathbf{s}, \mathbf{a}) \geq 0 \text{ and } O(\mathbf{a}) \geq 0 \quad r(\mathbf{s}, \mathbf{a}) = G(\mathbf{s}, \mathbf{a}) - H(\mathbf{s}, \mathbf{a}) - O(\mathbf{a}) \leq G(\mathbf{s}, \mathbf{a}) \leq C_G \quad (\text{A.8})$$

and similarly from below by

$$\begin{aligned} G(\mathbf{s}, \mathbf{a}) \geq 0 \quad & -r(\mathbf{s}, \mathbf{a}) = H(\mathbf{s}, \mathbf{a}) + O(\mathbf{a}) - G(\mathbf{s}, \mathbf{a}) \leq H(\mathbf{s}, \mathbf{a}) + O(\mathbf{a}) \\ \text{by (2.37) and (2.38)} \quad & \leq \langle \mathbf{h}, \mathbf{s} + \mathbf{a} \rangle + \langle \mathbf{o}_v, \mathbf{a} \rangle + o_f \\ & = \langle \mathbf{h}, \mathbf{s} \rangle + \langle \mathbf{h} + \mathbf{o}_v, \mathbf{a} \rangle + o_f. \end{aligned} \quad (\text{A.9})$$

The second product $\langle \mathbf{h} + \mathbf{o}_v, \mathbf{a} \rangle$ does not depend on the state \mathbf{s} and we can show that it is bounded from above by some constant. To find this bound, we solve the following optimization problem

$$\begin{aligned} \max_{\mathbf{a} \geq \mathbf{0}} \quad & \langle \mathbf{h} + \mathbf{o}_v, \mathbf{a} \rangle \\ \text{s.t.} \quad & \langle \mathbf{m}, \mathbf{a} \rangle \leq M. \end{aligned}$$

This is a finite-dimensional linear program, therefore strong duality holds between it and its dual

$$\begin{aligned} \min_{y \geq 0} \quad & M \cdot y \\ \text{s.t.} \quad & y \cdot m_i \geq h_i + o_{v,i} \quad \text{for all } 0 \leq i < n. \end{aligned}$$

The dual program has a single variable y and has a trivial solution

$$y = \max_{0 \leq i < n} \frac{h_i + o_{v,i}}{m_i}.$$

Indeed, any y that is smaller violates at least one of the constraints, and any one that is larger yields a larger objective $M \cdot y$. Thus, if we let

$$C_O \triangleq o_f + M \cdot \max_{0 \leq i < n} \frac{h_i + o_{v,i}}{m_i}, \quad (\text{A.10})$$

then the constant C_O is an upper bound on ordering costs and holding costs for the order. By combining the bounds (A.8) and (A.9), we find that

$$|r(\mathbf{s}, \mathbf{a})| \leq \max\{C_G, \langle \mathbf{h}, \mathbf{s} \rangle + C_O\} \leq \langle \mathbf{h}, \mathbf{s} \rangle + \max\{C_G, C_O\} = w(\mathbf{s}).$$

We denote $\max\{C_G, C_O\}$ by w_0 because $w(\mathbf{0}) = w_0$. The affine function $w(\mathbf{s}) \triangleq \langle \mathbf{h}, \mathbf{s} \rangle + w_0$ is a weight function that satisfies (2.25). Moreover, we can show that the remaining parts of Condition 2.5 hold for this weight function.

The weight function w was chosen so that (2.25) holds. To complete the proof, we need to show that the constants κ , v , and λ of Lemma 2.24 satisfy the requirements of Condition 2.5.

We start with the one-stage expansion coefficient κ . First, note that for any two states $\mathbf{s} \in \mathbb{S}$ and $\mathbf{s}' \in \mathbb{S}$

$$w(\mathbf{s}') = \langle \mathbf{h}, \mathbf{s}' \rangle + w_0 = \langle \mathbf{h}, \mathbf{s} \rangle + w_0 + \langle \mathbf{h}, \mathbf{s}' - \mathbf{s} \rangle = w(\mathbf{s}) + \langle \mathbf{h}, \mathbf{s}' - \mathbf{s} \rangle.$$

Therefore, for any state-action pair $(\mathbf{s}, \mathbf{a}) \in \mathbb{X}$

$$[\mathcal{T}w](\mathbf{s}, \mathbf{a}) = \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \cdot w(\mathbf{s}') \quad \triangleright \text{by definition}$$

$$\begin{aligned}
&= \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \cdot w(\mathbf{s}) + \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \cdot \langle \mathbf{h}, \mathbf{s}' - \mathbf{s} \rangle \\
&= w(\mathbf{s}) + \sum_{\mathbf{s}' \in \mathbb{S}} \sum_{i=0}^{n-1} p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \cdot h_i \cdot (s'_i - s_i).
\end{aligned}$$

$\sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) = 1$ \triangleleft

Additionally, the following inequality holds.

$$\begin{aligned}
\sum_{\mathbf{s}' \in \mathbb{S}} p_i(s'_i | s_i, a_i) \cdot (s'_i - s_i) &= \sum_{k=1}^{s_i+a_i} (k - s_i) \cdot p_i(s_i + a_i - k) \\
&\quad - s_i \cdot q_i(s_i + a_i) \\
&\leq \sum_{k=1}^{s_i+a_i} (k - s_i) \cdot p_i(s_i + a_i - k) \\
&= \sum_{j=0}^{s_i+a_i-1} (a_i - j) \cdot p_i(j) \\
&\leq \sum_{j=0}^{s_i+a_i-1} a_i \cdot p_i(j) \leq a_i \cdot \sum_{j=0}^{\infty} p_i(j) = a_i. \quad (\text{A.11})
\end{aligned}$$

by (2.34) \triangleleft

$s_i \cdot q_i(s_i + a_i) \geq 0$ \triangleleft

using $j = s_i + a_i - k$ \triangleleft

$j \geq 0$ \triangleleft

The transition probabilities $p(\mathbf{s}' | \mathbf{s}, \mathbf{a})$ are given by (2.35). Because each of the probabilities $p_i(s'_i | s_i, a_i)$ is between zero and one, for any product i

$$p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) = p_i(s'_i | s_i, a_i) \cdot \prod_{j \neq i} p_j(s'_j | s_j, a_j) \leq p_i(s'_i | s_i, a_i). \quad (\text{A.12})$$

Therefore,

$$\begin{aligned}
[\mathcal{T}w](\mathbf{s}, \mathbf{a}) &\leq w(\mathbf{s}) + \sum_{\mathbf{s}' \in \mathbb{S}} \sum_{i=0}^{n-1} p_i(s'_i | s_i, a_i) \cdot h_i \cdot (s'_i - s_i) \\
&= w(\mathbf{s}) + \sum_{i=0}^{n-1} h_i \cdot \sum_{\mathbf{s}' \in \mathbb{S}} p_i(s'_i | s_i, a_i) \cdot (s'_i - s_i) \\
&\leq w(\mathbf{s}) + \langle \mathbf{h}, \mathbf{a} \rangle \\
&= w(\mathbf{s}) + C_H \left(1 + \frac{C_H}{w(\mathbf{s})} \right) \cdot w(\mathbf{s}) \quad (\text{A.13}) \\
&\leq \left(1 + \frac{C_H}{\inf_{\mathbf{s}' \in \mathbb{S}} w(\mathbf{s}')} \right) \cdot w(\mathbf{s}) = \left(1 + \frac{C_H}{w_0} \right) \cdot w(\mathbf{s}). \quad (\text{A.14})
\end{aligned}$$

changing the summation order \triangleleft

The change of summation order is possible because the summands are all positive and Proposition 2.3 can be used. The constant C_H is chosen so that $\langle \mathbf{h}, \mathbf{a} \rangle \leq C_H$ for any action \mathbf{a} . The derivation of C_H is identical to the derivation of C_O in equation (2.40).

By putting inequality (A.14) into the definition of the transition operator \mathcal{T} , we see that for any state-action pair $(\mathbf{s}, \mathbf{a}) \in \mathbb{X}$

$$\gamma \cdot [\mathcal{T}w](\mathbf{s}, \mathbf{a}) \leq \kappa \cdot w(\mathbf{s}),$$

and therefore $\gamma \cdot [\mathcal{T}_\pi w](\mathbf{s}) = \gamma \cdot [\mathcal{T}w](\mathbf{s}, \pi(\mathbf{s})) \leq \kappa \cdot w(\mathbf{s})$ for any Markov deterministic policy $\pi \in \Pi_{\text{DM}}$.

Finally, we consider the contraction horizon v and the v -stage contraction coefficient λ . If $\kappa < 1$, we can set $v = 1$ and $\lambda = \kappa$. Otherwise we still need to show that Condition 2.5 holds. We start this part of the proof with showing that the following inequality holds:

$$\mathcal{T}_\pi^k w \leq w + k \cdot C_H \quad \text{for any } k \geq 1. \quad (\text{A.15})$$

We prove it by induction. The base case $k = 1$ holds due to the inequality (A.13). Assuming that the inequality (A.15) holds for some $k - 1$, we show that it holds for k . Indeed, for any state $\mathbf{s} \in \mathbb{S}$

$$\begin{aligned} [\mathcal{T}_\pi^k w](\mathbf{s}) &= [\mathcal{T}_\pi \mathcal{T}_\pi^{k-1} w](\mathbf{s}) = \sum_{\mathbf{s}' \in \mathbb{S}} p_\pi(\mathbf{s}' | \mathbf{s}) \cdot [\mathcal{T}_\pi^{k-1} w](\mathbf{s}') \\ &\leq \sum_{\mathbf{s}' \in \mathbb{S}} p_\pi(\mathbf{s}' | \mathbf{s}) \cdot (w(\mathbf{s}') + (k-1) \cdot C_H) &> \text{by the inductive hypothesis (A.15)} \\ &= (k-1) \cdot C_H + \sum_{\mathbf{s}' \in \mathbb{S}} p_\pi(\mathbf{s}' | \mathbf{s}) \cdot w(\mathbf{s}') \\ &= (k-1) \cdot C_H + [\mathcal{T}w](\mathbf{s}, \pi(\mathbf{s})) \\ &\leq (k-1) \cdot C_H + w(\mathbf{s}) + C_H &> \text{by (A.13)} \\ &= w(\mathbf{s}) + k \cdot C_H. \end{aligned}$$

Now that we have proven the inequality (A.15), we use it to show that

$$[\mathcal{T}_\pi^k w](\mathbf{s}) \leq w(\mathbf{s}) + k \cdot C_H \leq (1 + k \cdot \frac{C_H}{w_0}) \cdot w(\mathbf{s}).$$

Thus, if $\lambda = \gamma^v \cdot (1 + Cv) < 1$ for some v , then these values of v and λ satisfy Condition 2.5. To find such a contraction horizon v , we solve the inequality

$$\begin{aligned} \gamma^v \cdot (1 + Cv) &< 1, \\ \gamma^{v+C^{-1}} \cdot (v + C^{-1}) \cdot \ln \gamma &> C^{-1} \gamma^{1/C} \cdot \ln \gamma, &> x C^{-1} \cdot \gamma^{1/C} \cdot \ln \gamma < 0 \\ \exp((v + C^{-1}) \cdot \ln \gamma) \cdot (v + C^{-1}) \cdot \ln \gamma &> C^{-1} \gamma^{1/C} \cdot \ln \gamma. &> a^b = \exp(b \cdot \ln a) \end{aligned}$$

Let $x = C^{-1} \cdot \gamma^{1/C} \cdot \ln \gamma$ and $y = (v + C^{-1}) \cdot \ln \gamma$. The inequality becomes $y \cdot e^y > x$.

Let us solve the equation $y \cdot e^y = x$ first. For real-valued x , this problem has a solution only if $x \geq -e^{-1}$ [Corless, Gonnet, Hare, Jeffrey, and Knuth, 1996], which holds if $\gamma \neq e^{-C}$. By substituting $\gamma = e^{-C}$ into the formula for κ , it is easy to check that in this case $\kappa < 1$ for any $C > 0$ and we can use $v = 1$. If $\gamma \neq e^{-C}$, x is a negative number because $\ln \gamma < 0$ for any $0 < \gamma < 1$. For $-e^{-C} \leq x < 0$, the equation $y \cdot e^y = x$ has two solutions, $y = W_{-1}(x)$ and $y = W_0(x)$

such that $W_{-1}(x) < W_0(x) < 0$ [Corless, Gonnet, Hare, Jeffrey, and Knuth, 1996]. Moreover, $y \cdot e^y > x$ if $y < W_{-1}(x)$ or $y > W_0(x)$.

By substitution of x and y back into the inequality, we obtain $(v + C^{-1}) \cdot \ln \gamma < W_{-1}(C^{-1} \cdot \gamma^{1/C} \cdot \ln \gamma)$ or $(v + C^{-1}) \cdot \ln \gamma > W_0(C^{-1} \cdot \gamma^{1/C} \cdot \ln \gamma)$ and therefore

Recall that $\ln \gamma$ is a negative number, hence the change in the inequality signs.

$$v < \frac{W_0(C^{-1} \cdot \gamma^{1/C} \cdot \ln \gamma)}{\ln \gamma} - \frac{1}{C} \quad \text{or} \quad v > \frac{W_{-1}(C^{-1} \cdot \gamma^{1/C} \cdot \ln \gamma)}{\ln \gamma} - \frac{1}{C}.$$

The first case yields non-positive values of v . The value of v used in the statement of Lemma 2.24 is the smallest positive value for which the second inequality holds. This concludes the proof. **QED**

A.4.3 Proof of Value Bounds

Lemma 4.1 * value bounds in inventory management

In the multi-product inventory management problem, for any policy $\pi \in \Pi$ the value $v_\pi(\mathbf{s})$ of each state $\mathbf{s} \in \mathbb{S}$ is bounded by the functions $u_\pm \in L^w(\mathbb{S})$:

$$-\mu \cdot w(\mathbf{s}) \leq u_-(\mathbf{s}) \leq v_\pi(\mathbf{s}) \leq u_+(\mathbf{s}) \leq \mu \cdot w(\mathbf{s}),$$

$$\text{where} \quad u_-(\mathbf{s}) \triangleq -\frac{1}{1-\gamma} \cdot \langle \mathbf{h}, \mathbf{s} \rangle - \frac{C_O - \gamma \cdot (C_O - C_H)}{(1-\gamma)^2} \quad (4.2)$$

$$\text{and} \quad u_+(\mathbf{s}) \triangleq \frac{C_G}{1-\gamma}. \quad (4.3)$$

The constants C_G , C_O , and C_H are defined in Lemma 2.24

Proof. From equations (A.8), (A.9) and (2.40) we know that the rewards belong to intervals

$$-w(\mathbf{s}) \leq r_-(\mathbf{s}) \leq r(\mathbf{s}, \mathbf{a}) \leq r_+(\mathbf{s}) \leq w(\mathbf{s}), \quad \text{where} \\ r_-(\mathbf{s}) \triangleq -\langle \mathbf{h}, \mathbf{s} \rangle - C_O \quad \text{and} \quad r_+(\mathbf{s}) \triangleq C_G.$$

The upper bound $u_+ = C_G/(1-\gamma)$ then immediately follows from $r \leq r_+$:

$$\begin{aligned} v_\pi(\mathbf{s}) &= \sum_{i=0}^{\infty} \gamma^i \cdot \sum_{\mathbf{s}' \in \mathbb{S}} p_{\pi,i}(\mathbf{s}' | \mathbf{s}) \cdot r_\pi(\mathbf{s}') \\ &\leq C_G \cdot \sum_{i=0}^{\infty} \gamma^i \cdot \sum_{\mathbf{s}' \in \mathbb{S}} p_{\pi,i}(\mathbf{s}' | \mathbf{s}) = C_G \cdot \sum_{i=0}^{\infty} \gamma^i = \frac{C_G}{1-\gamma} = u_+(\mathbf{s}). \end{aligned}$$

Similarly, for the lower bound,

$$v_\pi(\mathbf{s}) \geq -\frac{C_O}{1-\gamma} - \sum_{i=0}^{\infty} \gamma^i \cdot \sum_{\mathbf{s}' \in \mathbb{S}} p_{\pi,i}(\mathbf{s}' | \mathbf{s}) \cdot \langle \mathbf{h}, \mathbf{s}' \rangle. \quad (\text{A.16})$$

To simplify this expression, we show by induction that

$$\sum_{\mathbf{s}' \in \mathbb{S}} p_{\pi}^k(\mathbf{s}' | \mathbf{s}) \cdot \langle \mathbf{h}, \mathbf{s}' \rangle \leq \langle \mathbf{h}, \mathbf{s} \rangle + k \cdot C_H \quad \text{for any } k \geq 0. \quad (\text{A.17})$$

For $k = 0$,

$$\sum_{\mathbf{s}' \in \mathbb{S}} p_{\pi}^k(\mathbf{s}' | \mathbf{s}) \cdot \langle \mathbf{h}, \mathbf{s}' \rangle = \delta_{\mathbf{s}, \mathbf{s}'} \cdot \langle \mathbf{h}, \mathbf{s}' \rangle = \langle \mathbf{h}, \mathbf{s} \rangle = \langle \mathbf{h}, \mathbf{s} \rangle + k \cdot C_H. \quad \triangleright \text{by (2.3)}$$

Additionally, for $k = 1$,

$$\begin{aligned} \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \cdot \langle \mathbf{h}, \mathbf{s}' \rangle &= \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \cdot \sum_{i=0}^{n-1} h_i \cdot s'_i &> \text{by inner product definition} \\ &\leq \sum_{\mathbf{s}' \in \mathbb{S}} \sum_{i=0}^{n-1} p_i(s'_i | s_i, a_i) \cdot h_i \cdot s'_i &> \text{by (A.12)} \\ &= \sum_{j=0}^{n-1} h_j \cdot \sum_{\mathbf{s}' \in \mathbb{S}} p_j(s'_j | s_j, a_j) \cdot s'_j &> \text{changing summation order} \\ &= \sum_{j=0}^{n-1} h_j \cdot \left(s_j + \sum_{\mathbf{s}' \in \mathbb{S}} p_j(s'_j | s_j, a_j) \cdot (s'_j - s_j) \right) &> \sum_{\mathbf{s}' \in \mathbb{S}} p_j(s'_j | s_j, a_j) = 1 \\ &\leq \sum_{j=0}^{n-1} h_j \cdot (s_j + a_j) = \langle \mathbf{h}, \mathbf{s} \rangle + \langle \mathbf{h}, \mathbf{a} \rangle &> \text{by (A.11)} \\ &\leq \langle \mathbf{h}, \mathbf{s} \rangle + C_H. &(\text{A.18}) \triangleright \text{by definition of } C_H, \text{ also see (A.13)} \end{aligned}$$

Assuming that inequality A.17 holds for $k - 1$, we write

$$\begin{aligned} \sum_{\mathbf{s}'' \in \mathbb{S}} p_{\pi}^k(\mathbf{s}'' | \mathbf{s}) \cdot \langle \mathbf{h}, \mathbf{s}'' \rangle &= \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \pi(\mathbf{s})) \cdot p_{\pi}^{k-1}(\mathbf{s}'' | \mathbf{s}') \cdot \langle \mathbf{h}, \mathbf{s}'' \rangle &> \text{by (2.3)} \\ &\leq \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \pi(\mathbf{s})) \cdot (\langle \mathbf{h}, \mathbf{s}' \rangle + (k - 1) \cdot C_H) &> \text{by inductive hypothesis (A.17)} \\ &= \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \pi(\mathbf{s})) \cdot \langle \mathbf{h}, \mathbf{s}' \rangle \\ &\quad + (k - 1) \cdot C_H \cdot \sum_{\mathbf{s}' \in \mathbb{S}} p(\mathbf{s}' | \mathbf{s}, \pi(\mathbf{s})) \\ &\leq \langle \mathbf{h}, \mathbf{s} \rangle + C_H + (k - 1) \cdot C_H \cdot 1 &> \text{by (A.18)} \\ &= \langle \mathbf{h}, \mathbf{s} \rangle + k \cdot C_H. \end{aligned}$$

Thus, inequality (A.17) holds by induction. Combining it with inequality (A.16), we obtain the lower bound:

$$\begin{aligned} v_{\pi}(\mathbf{s}) &\geq -\frac{C_O}{1-\gamma} - \sum_{i=0}^{\infty} \gamma^i \cdot (\langle \mathbf{h}, \mathbf{s} \rangle + i \cdot C_H) \\ &= -\frac{C_O}{1-\gamma} - \frac{\langle \mathbf{h}, \mathbf{s} \rangle}{1-\gamma} - \frac{\gamma \cdot C_H}{(1-\gamma)^2} &> \sum_{i=0}^{\infty} \gamma^i \cdot i = \frac{\gamma}{(1-\gamma)^2} \\ &= -\frac{1}{1-\gamma} \cdot \langle \mathbf{h}, \mathbf{s} \rangle - \frac{C_O - \gamma \cdot (C_O - C_H)}{(1-\gamma)^2}. \quad \text{QED} \end{aligned}$$

B

Active Wake Control Implementation Details

	parameter	I	II
sampling	discounting factor	0.99	
	size of the replay buffer	10^5	
	batch size	128	
	start learning at step	4321	7201
actor	learning rate	10^{-3}	10^{-5}
	layers	2	
	neurons per layer	128	
	activations	ReLU	
critic	learning rate	10^{-2}	10^{-4}
	layers	2	
	neurons per layer	128	
	activations	ReLU	
target updates	Polyak τ	0.05	
	frequency	60	
TD3	policy noise	0.2	
	policy update frequency	60	
	noise clipped at	± 0.5	
	gradient norm clipped at	± 0.5	
SAC	initial α	1.0	
	α learning rate	10^{-2}	10^{-4}

Table B.1:
Hyperparameters of the deep reinforcement learning agents. The second experiment uses the same parameters unless explicitly listed. In both experiments learning starts after the first evaluation. TD3 uses noise only in training, but not in evaluation. The parameter α of SAC is auto-tuned starting with the initial value. For the second experiment, the learning rates of SAC were additionally tuned using a grid search.