

UNIVERSITY OF NEWCASTLE

DOCTORAL THESIS

**Cooperative Reinforcement Learning
for Independent Learners**

Author:

Bilal Hashem Kalil Abed-Alguni

BCS, MCS

*A thesis submitted to the
University of Newcastle, NSW, Australia
for the degree of
Doctor of Philosophy
(Computer Science)*

October 2014

Statement of Originality

The thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to the final version of my thesis being made available worldwide when deposited in the University's Digital Repository**, subject to the provisions of the Copyright Act 1968. **Unless an Embargo has been approved for a determined period.

(Bilal Hashem Abed-Alguni)

Contents

Statement of Originality	ii
List of Figures	ix
Abstract	xi
Acknowledgements	xiv
Abbreviations	xv
1 Introduction	1
1.1 Cooperative Reinforcement Learning for Independent Learners	2
1.2 Motivation	4
1.3 Research Objectives	6
1.4 Research Methodology	9
1.5 Contributions	10
1.6 Structure of the Thesis	12
2 Agent-based Model	15
2.1 Motivating Problem: Hunter Prey Problem	15
2.2 Intelligent Agents	16
2.3 Agent Types	17
2.4 Single-agent vs. Multi-agent Systems	19
2.4.1 Single-agent Systems	20
2.4.2 Multi-agent Systems	20
2.5 Multi-agent Systems Agendas	21
2.6 Multi-agent Systems Classifications	23
2.7 Multi-agent Learning	23
2.7.1 Cooperative Multi-agent Learning Techniques	23
2.7.2 Team Learning	24
2.7.3 Concurrent Learning	26
2.7.4 Learning and Communication	29

2.8	Famous Agent-based Models	29
2.8.1	Belief-Desire-Intention Agent-based Model	30
2.8.2	Swarm Intelligent Agent-based Model	30
2.8.3	Hierarchical Multi-agent Learning	31
2.8.4	Layered Multi-agent Learning	31
2.8.5	Distributed Multi-agent Systems	32
2.8.6	Distributed Hunter Prey Problem	33
2.8.7	Advantages of Agents-based Models	34
2.8.8	Limitations of Agent-based Models	35
2.9	Learning and Teaching	35
3	Markov Decision Process	37
3.1	Markov Decision Processes	38
3.2	Partially Observable Markov Decision Processes	40
3.3	Stochastic Games	41
3.4	Finite-horizon MDP vs Infinite-horizon MDP	44
3.5	Bellman's Equation	44
3.6	Application of Markov Decision Processes	45
3.7	Limitations of Markov Decision Processes	46
3.7.1	Curse of dimensionality	46
3.7.2	Memory Requirement	46
3.7.3	Stationary Assumption	47
3.8	Large Markov Decision Processes	48
3.9	Decomposition Approaches for Markov Decision Process	48
3.9.1	Factored Approaches	48
3.9.2	Hierarchical Approaches	54
3.10	Merging Markov Decision Processes	59
3.11	Summary	62
4	Reinforcement Learning	65
4.1	Introduction	65
4.2	Reinforcement Learning Model	67
4.3	Types of Policy	68
4.4	Episodic vs. Continuous Tasks	68
4.5	Discounted Reward	69
4.6	Applications of Reinforcement Learning	70
4.7	The Trade-off Between Exploration and Exploitation	71
4.8	Action Selection Policies	71
4.9	Temporal Credit Assignment	72
4.10	Estimation of Value Function	73
4.10.1	Dynamic Programming Methods	73
4.10.2	Monte Carlo Methods	74
4.10.3	Temporal Difference Learning	75
4.11	Q-learning Algorithm	76

4.12	SARSA Algorithm	77
4.13	Multi-agent Reinforcement Learning	78
4.14	Hierarchical Reinforcement Learning	79
4.15	Model-based Learning Methods	80
4.15.1	Dyna Architecture	80
4.15.2	Prioritised Sweeping	81
4.16	The Actor-critic Architecture	82
4.17	Performance Evaluation	83
4.18	Summary	84
5	Multi-agent Reinforcement Learning	87
5.1	Introduction	87
5.2	Benefits of Multi-agent Reinforcement Learning	88
5.2.1	Parallel Computation	89
5.2.2	Sharing of Knowledge	89
5.2.3	Robustness	89
5.2.4	Distributed Learning	90
5.3	Challenges in Multi-agent Reinforcement Learning	90
5.3.1	The Trade-off Between Exploration and Exploitation	90
5.3.2	Structural Credit Assignment Problem	91
5.3.3	Curse of Dimensionality	91
5.3.4	Non-stationary Environment	92
5.3.5	Coordination of Actions	92
5.3.6	Specifying The Goal	92
5.4	Recent Advances in Multi-agent Reinforcement Learning	92
5.4.1	Combinational Reinforcement Learning	93
5.4.2	Swarm Reinforcement Learning	95
5.4.3	Cooperation and Coordination in MARL	98
5.4.4	Reinforcement Learning for Stochastic Cooperative Multi-agent Systems	102
5.4.5	Reducing Joint Action Space in Cooperative Multi-agent Reinforcement Learning	106
5.4.6	Teacher Learner Model in Reinforcement Learning	109
5.4.7	Rewards Modifications in Reinforcement Learning	111
5.5	Summary	112
6	Hierarchical Reinforcement Learning Model	115
6.1	Introduction	116
6.2	Related Work	118
6.2.1	Hierarchical Decomposition in the RL Domain	118
6.2.2	Cooperative Hunting Strategy	119
6.3	Distributed Hierarchical Learning Model	120
6.3.1	Problem Model	120
6.3.2	Example	121

6.3.3	Design of DHLM	122
6.3.4	Migration of Agents	123
6.3.5	Task Scheduling	128
6.3.6	Roles of Worker Agents at Different Levels of Distributed Systems	135
6.4	Intelligent Distributed Q-learning Algorithm	141
6.4.1	Independent Learners	141
6.4.2	The Q-Functions of IDQL	142
6.4.3	Modifications of Policy	143
6.4.4	Limitations vs Benefits	143
6.5	Experiments	144
6.5.1	Setup	144
6.5.2	Results and Discussion	146
6.6	Summary	149
7	Hierarchical Cooperative Policy Construction for Independent Q-Learners	151
7.1	Introduction	152
7.2	Related Work	153
7.3	Motivating Example: Nearest Emergency Exit Problem	154
7.4	QA-learning Algorithm	155
7.4.1	Two Cooperative Roles	155
7.4.2	Problem Model	156
7.4.3	QA-learning Algorithm	158
7.5	Experiments	162
7.5.1	Setup	162
7.5.2	Results and Discussion	163
7.6	Summary	171
8	Cooperative Q-learning Algorithms for Independent Learners	175
8.1	Introduction	176
8.2	Q-value Sharing Strategies	177
8.2.1	BEST-Q	177
8.2.2	AVE-Q	178
8.2.3	PSO-Q	178
8.2.4	WSS	179
8.2.5	Aggregate Sharing Strategy	180
8.3	Experiments 1	180
8.3.1	Equal Levels of Experience	182
8.3.2	Different Experiences	190
8.4	Combined approach of IDQL, QA-learning and BEST-Q	197
8.5	Experiments 2	198
8.5.1	Setup	198
8.5.2	Results and Discussion	199

8.6	Summary	203
9	Final Discussion	207
9.1	Review of Existing Research	209
9.1.1	Reinforcement Learning	209
9.1.2	Multi-agent Reinforcement Learning	210
9.1.3	Hierarchical Structure of Complex Systems	211
9.2	Conclusion	212
9.2.1	Research Question 1	212
9.2.2	Research Question 2	214
9.2.3	Research Question 3	217
9.3	Future Directions	219
9.3.1	Cooperative Dyna Architecture	219
9.3.2	QA-learning for Single-goal Hierarchical Systems	220
9.3.3	Hierarchical Reinforcement Learning algorithms	220
	 Bibliography	 221

List of Figures

2.1	An example of hunter prey problem on a 5×5 grid.	16
2.2	Model of an agent based on Russell and Norvig [2003].	17
2.3	Mobile agent based on Lange and Oshima [1999]	18
2.4	Distributed hunter prey problem composed of two sub-grids.	19
2.5	The hunter prey problem with a single hunter agent.	20
2.6	The hunter prey problem with multiple agents.	21
2.7	The hunter prey problem with a heterogeneous team of hunters.	25
2.8	A team of hunters composed of two disjoint groups of hunters.	26
2.9	Hierarchical multi-agent systems.	31
2.10	Distributed multi-agent system composed of three subsystems.	32
2.11	Distributed hunter prey problem.	34
2.12	A game of two players.	36
3.1	A view of an MDP based on Russell and Norvig [2003, Chapter 17].	38
3.2	An example of hunter prey problem on a 5×5 grid.	40
3.3	The visual area of hunter agent H_3 with depth 1.	42
3.4	An example of hunter prey problem on a 4×4 grid.	47
3.5	Distributed hunter prey problem.	50
3.6	Distributed hunter prey problem composed of four sub-grids.	51
4.1	Reinforcement learning model from Sutton and Barto [1998].	66
4.2	The problem formulation used in Dyna based on Sutton [1990].	81
4.3	The Actor-critic architecture based on Raicevic [2006].	82
5.1	The aggregation architecture based on Jiang and Kamel [2006].	93
5.2	A reinforcement learner model based on Liu and Zeng [2006].	98
6.1	Distributed RL Hierarchical Model.	117
6.2	Generalisation of DHLM agents.	122
6.3	Example of the two migration procedures.	124
6.4	Migration Algorithm.	125
6.5	Partially finished hunting.	126
6.6	Service queue and inactive list.	126
6.7	Redistribution of worker agents.	127
6.8	Task scheduling.	130
6.9	Sequential scheduling procedure.	131

6.10	Priority scheduling procedure.	132
6.11	Parallel scheduling procedure.	133
6.12	Combined scheduling procedure.	134
6.13	Shortest Manhattan distance.	135
6.14	Specialised hunter agents at system level.	136
6.15	Roles in a distributed hunter prey problem (system level).	138
6.16	Ambush algorithm (system level).	139
6.17	Specialised hunter agents at sub-system level.	140
6.18	Roles in a distributed hunter prey problem (sub-system level).	140
6.19	Ambush algorithm (subsystem level).	141
6.20	Experiment 1: Performance of IDQL vs Q-learning.	147
6.21	Experiment 2: Performance of IDQL vs Q-learning.	148
7.1	Nearest emergency exit problem.	154
7.2	Q-table models for multiple goal systems.	159
7.3	QA-learning flowchart.	159
7.4	The second stage of the QA-learning algorithm.	160
7.5	An example of the Merge procedure of QA-learning.	162
7.6	Experiment 1: QA-learning with three integration scenarios.	164
7.7	Experiment 1: Single agent Q-learning.	165
7.8	Experiment 2: QA-learning with three integration scenarios.	166
7.9	Experiment 2: Single agent Q-learning.	167
7.10	Experiment 3: QA-learning with three integration scenarios.	168
7.11	Experiment 3: Single agent Q-learning.	169
8.1	Average number of moves per one episode in 5×5 grid.	183
8.2	Average number of moves per 10 episodes in 5×5 grid.	183
8.3	Average number of moves per 100 episodes in 5×5 grid.	184
8.4	Average number of moves per one episode in 10×10 grid.	185
8.5	Average number of moves per 10 episodes in 10×10 grid.	185
8.6	Average number of moves per 100 episodes in 10×10 grid.	186
8.7	Average number of moves per one episode in 20×20 grid.	187
8.8	Average number of moves per 10 episodes in 20×20 grid.	188
8.9	Average number of moves per 100 episodes in 20×20 grid.	189
8.10	Average number of moves per 10 episodes in 5×5 grid.	191
8.11	Average number of moves per 100 episodes in 5×5 grid.	192
8.12	Average number of moves per 10 episodes in 10×10 grid.	193
8.13	Average number of moves per 100 episodes in 10×10 grid.	193
8.14	Average number of moves per 10 episodes in 20×20 grid.	195
8.15	Average number of moves per 100 episodes in 20×20 grid.	196
8.16	Experiment 1: EQA-learning vs Q-learning.	200
8.17	Experiment 2: EQA-learning vs Q-learning.	201
8.18	Experiment 3: EQA-learning vs Q-learning	202

Abstract

Cooperative Reinforcement Learning for Independent Learners

by Bilal ABED-ALGUNI, BCS, MCS

University of Newcastle

Faculty of Engineering and Built Environment

School of Electrical Engineering and Computer Science

A thesis submitted for the degree of Doctor of Philosophy

Machine learning in multi-agent domains poses several research challenges. One challenge is how to model cooperation between reinforcement learners. Cooperation between independent reinforcement learners is known to accelerate convergence to optimal solutions. In large state space problems, independent reinforcement learners normally cooperate to accelerate the learning process using decomposition techniques or knowledge sharing strategies. This thesis presents two techniques to multi-agent reinforcement learning and a comparison study. The first technique is a formal decomposition model and an algorithm for distributed systems. The second technique is a cooperative Q-learning algorithm for multi-goal decomposable systems. The comparison study compares the performance of some of the best known cooperative Q-learning algorithms for independent learners.

Distributed systems are normally organised into two levels: system and subsystem levels. This thesis presents a formal solution for decomposition of Markov Decision Processes (MDPs) in distributed systems that takes advantage of the organisation of distributed systems and provides support for migration of learners. This is accomplished by two proposals: a Distributed, Hierarchical Learning Model (DHLM) and an Intelligent Distributed Q-Learning algorithm (IDQL) that are based on three specialisations of agents: workers, tutors and consultants. Worker agents are the actual learners and performers of tasks, while tutor agents and consultant agents are coordinators at the subsystem level and the system level, respectively. A main duty of consultant and tutor agents is the assignment of problem space to worker agents. The experimental results in a distributed hunter prey problem suggest that IDQL converges to a solution faster than the single agent Q-learning algorithm. An important feature of DHLM is that it provides a solution for migration of agents.

This feature provides support for the IDQL algorithm where the problem space of each worker agent can change dynamically. Other hierarchical RL models do not cover this issue.

Problems that have multiple goal-states can be decomposed into sub-problems by taking advantage of the loosely-coupled bonds among the goal states. In such problems, each goal state and its problem space form a sub-problem. This thesis introduces Q-learning with Aggregation algorithm (QA-learning), an algorithm for problems with multiple goal-states that is based on two roles: learner and tutor. A learner is an agent that learns and uses the knowledge of its neighbours (tutors) to construct its Q-table. A tutor is a learner that is ready to share its Q-table with its neighbours (learners). These roles are based on the concept of learners reusing tutors' sub-solutions. This algorithm provides solutions to problems with multiple goal-states. In this algorithm, each learner incorporates its tutors' knowledge into its own Q-table calculations. A comprehensive solution can then be obtained by combining these partial solutions. The experimental results in an instance of the shortest path problem suggest that the output of QA-learning is comparable to the output of a single Q-learner whose problem space is the whole system. But the QA-learning algorithm converges to a solution faster than a single learner approach.

Cooperative Q-learning algorithms for independent learners accelerate the learning process of individual learners. In this type of Q-learning, independent learners share and update their Q-values by following a sharing strategy after some episodes learning independently. This thesis presents a comparison study of the performance of some famous cooperative Q-learning algorithms (BEST-Q, AVE-Q, PSO-Q, and WSS) as well as an algorithm that aggregates their results. These algorithms are compared in two cases: equal experience and different experiences cases. In the first case, the learners have equal learning time, while in the second case, the learners have different learning times. The comparison study also examines the effects of the frequency of Q-value sharing on the learning speed of independent learners.

The experimental results in the equal experience case indicate that sharing of Q-values is not beneficial and produces similar results to single agent Q-learning. While, the experimental results in the different experiences case suggest that each of the cooperative Q-learning algorithms performs similarly, but better than single agent Q-learning. In both cases, high-frequency sharing of Q-values accelerates the convergence to solutions compared to low-frequency sharing. Low-frequency Q-value

sharing degrades the performance of the cooperative Q-learning algorithms in the equal experience and different experiences cases.

Acknowledgements

This Ph.D thesis would not have seen the light without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here.

First and foremost I offer my sincerest gratitude to my supervision team: A/Prof. Stephan Chalup, A/Prof. Frans Henskens, A/Prof. Huilin Ye, Dr David Paul, and Dr Yuqing Lin for the continuous support of my Ph.D research. Their patience, advice, encouragement, motivation, and enthusiasm have been of great value to me. Indeed, without their guidance, I would not be able to put the topic together.

I would like to acknowledge the financial, academic and technical support of the University of Newcastle and its staff, particularly in the award of University of Newcastle International Postgraduate Research Scholarship (UNIPRS) and University of Newcastle Research Scholarship Central 5050 that provided the necessary financial support for this research. The library and computer facilities of the university have been indispensable for the completion of my research and writing my doctoral thesis.

I would like to thank everybody who has been involved in the Interdisciplinary Machine Learning Research Group (IMLRG), and the Distributed Computing Research Group (DCRG).

I would like to thank my confirmation committee: A/Prof. Frans Henskens, Dr Yuqing Lin, and Dr Alexandre Mendes. I admit that my research would be much poorer without the constructive criticism and advice that were provided during my confirmation exam.

Finally, and most importantly, I would like to thank my family for their unconditional support, especially my wonderful wife Flordeliza, my lovely daughter Rua, my parents Hashem and my departed mother Sukiba, my sisters Yara Rua Lara Maha, and my brother Tareq. Your encouragement and emotional support helped me to overcome the hard moments that I have been through during my research. I will always owe them a debt of gratitude for believing in me, even when I did not. A smile from baby Rua was enough to encourage me and meant the world to me.

Abbreviations

ABM	Agent-Based Model
AC	Actor-Critic
AMRLS	Aggregated Multiple Reinforcement Learning System
AOEs	Areas Of Expertise
AVE-Q	Average Q-learning
BA	Boltzman Addition
BEST-Q	BEST Q-learning
BM	Boltzman Multiplication
CBR	Case-Based Reasoning
CG	Coordination Graph
CST	Constructing Skill Trees
DFG	Dissolution and Formation of Groups
DHLM	Distributed Hierarchical Learning Model
FMDP	Factored Markov Decision Process
FMQ	Frequent Maximum Q-values
FIRL	Feature Construction for Inverse Reinforcement Learning
MARL	Multi-agent Reinforcement Learning
MAS	Multi-agent System
MCLA	Multi-agent Cooperative Learning algorithm
MCLM	Multi-agent Cooperative Learning Model
MDP	Markov Decision Process
MV	Majority Voting
IDQL	Intelligent Distributed Q-Learning

POMDP	Partially Observable Markov Decision Process
PSO-Q	Particle Swarm Optimisation Q-learning
QA-learning	Q-learning by Aggregation
RL	Reinforcement Learning
RV	Rank Voting
SARSA	State-Action-Reward-State-Action
SCMARL	State-clusters Multi-agent Reinforcement Learning
SRL	Selfish Reinforcement Learning
WSS	Weighted Strategy Sharing
WMV	Weighted Majority Voting
2LRL	Two-Level Reinforcement Learning

Dedicated to my family.

