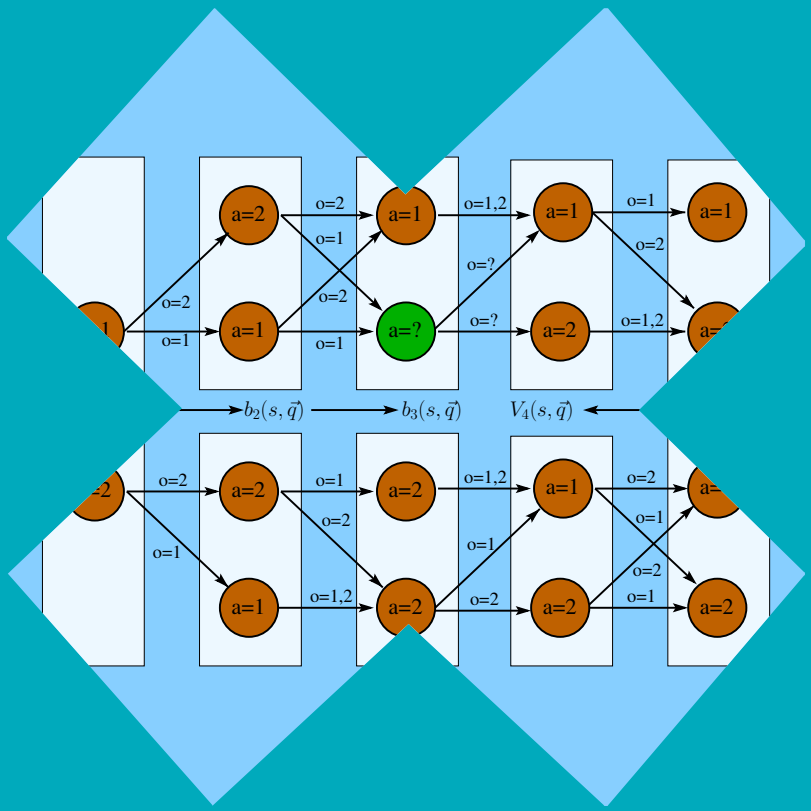
Department of Information and Computer Science

Planning under uncertainty for large-scale problems with applications to wireless networking

Joni Pajarinen



Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 20/2013

Planning under uncertainty for large-scale problems with applications to wireless networking

Joni Pajarinen

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at the lecture hall T2 of the school on 7th February 2013 at 12 noon.

Aalto University
School of Science
Department of Information and Computer Science

Supervising professor
Prof. Erkki Oja

Thesis advisor
Dr. Jaakko Peltonen

Preliminary examiners
Dr. Matthijs Spaan, Technical University of Delft, The Netherlands
Dr. Esa Hyytiä, School of Electrical Engineering, Aalto University, Finland

Opponent
Dr. François Charpillet, INRIA, France

Aalto University publication series
DOCTORAL DISSERTATIONS 20/2013

ISBN 978-952-60-4998-4 (printed)
ISBN 978-952-60-4999-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-4999-1

Unigrafia Oy
Helsinki 2013

Finland

441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Joni Pajarinen

Name of the doctoral dissertation
Planning under uncertainty for large-scale problems with applications to wireless networking

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series DOCTORAL DISSERTATIONS 20/2013

Field of research Computer and Information Science

Manuscript submitted 21 August 2012 **Date of the defence** 7 February 2013

Permission to publish granted (date) 17 October 2012 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

Planning actions into the future is a fundamental task in many real world problems. The uncertain outcome of actions and partial noisy observations often make planning difficult. Specifically, in a wireless network, wireless agents must reason whether to transmit data now or postpone transmission into the future, based only on noisy sensor readings and incomplete information about traffic patterns and the state of other devices.

In many settings of this kind, a partially observable Markov decision process (POMDP) defines optimal actions for a single agent and a decentralized POMDP (DEC-POMDP) for multiple co-operative agents. POMDPs and DEC-POMDPs are expressive but computationally demanding models. This thesis presents new efficient POMDP and DEC-POMDP methods, motivated by challenging new wireless networking problems.

The first contribution of this thesis is a method for large factored POMDPs that handles larger problems than the comparison methods. The second contribution is the first proposed method for general factored infinite-horizon DEC-POMDPs. The method solves smaller problems with similar accuracy as non-factored methods and it can solve larger problems than the comparison methods. The third contribution is a new kind of controller type for POMDPs and DEC-POMDPs, a periodic finite state controller, that allows optimization of larger controllers than previous finite state controller approaches and yields higher performance.

The fourth contribution is a POMDP model for a cognitive radio device, which served as motivation for the factored POMDP method. In the model, the cognitive radio transmits on frequency channels occupied by high priority legacy users. The model takes into account varying network traffic burst lengths and reactions of legacy users and performs better than the comparison models. The fifth contribution consists of framing wireless channel access of multiple devices with complicated spatial interference as a factored DEC-POMDP. This allows optimizing over both the spatial and time dimensions and in experiments yields higher performance than the wireless comparison methods.

The quality of wireless device decisions depends crucially on the cost and quality of sensor readings. The last contribution is a new spectrum sensing approach, that uses nanotechnology based computations and machine learning for mitigating nanoscale faults and classifying radio signals.

Keywords Planning under uncertainty, POMDP, DEC-POMDP, wireless network, WLAN, cognitive radio, nanocomputing

ISBN (printed) 978-952-60-4998-4 **ISBN (pdf)** 978-952-60-4999-1

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2013

Pages 204 **urn** http://urn.fi/URN:ISBN:978-952-60-4999-1

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Joni Pajarinen

Väitöskirjan nimi
Päätöksenteko epävarmuuden vallitessa suurissa ongelmissa ja sovelluksia langattomaan tiedonsiirtoon

Julkaisija Perustieteiden korkeakoulu

Yksikkö Tietojenkäsittelytieteen laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 20/2013

Tutkimusala Informaatiotekniikka

Käsikirjoituksen pvm 21.08.2012 **Väitöspäivä** 07.02.2013

Julkaisuluvan myöntämispäivä 17.10.2012 **Kieli** Englanti

☐ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Toimintojen suunnittelu tulevaisuuteen on tärkeä tehtävä useissa käytännön ongelmissa. Epävarmuus toimintojen lopputuloksesta ja vaillinaiset kohinaiset havainnot tekevät suunnittelusta usein vaikeaa. Erityisesti langattomissa verkoissa langattomien agenttien täytyy päättää milloin lähettää dataa, käyttäen ainoastaan kohinaisia havaintoja ja vaillinaista tietoa verkkoliikenteestä ja muiden laitteiden tilasta.

Useissa tämänkaltaisissa tilanteissa osittain havaittava Markov-päätösprosessi (POMDP)-malli määrittelee optimaaliset toiminnot yhdelle agentille ja hajautettu POMDP (DEC-POMDP)-malli usealle yhteistyötä tekevälle agentille. Tämä väitöskirja esittelee uusia tehokkaita menetelmiä näille ilmaisukykyisille, mutta laskennallisesti vaativille POMDP ja DEC-POMDP-malleille. Uudet vaativat langattomat sovellukset toimivat motivaationa menetelmille.

Tämän väitöskirjan ensimmäinen kontribuutio on faktoroitu POMDP-menetelmä, joka ratkaisee suurempia ongelmia kuin vertailumenetelmät. Toinen kontribuutio on ensimmäinen ehdotettu faktoroitu äärettömän horisontin DEC-POMDP-menetelmä, joka ratkaisee pienempiä ongelmia samalla tarkkuudella ja suurempia ongelmia kuin ei-faktoroidut vertailumenetelmät. Kolmas kontribuutio esittelee uudentyyppisen jaksollisen tilakonesäätimen POMDP ja DEC-POMDP-malleille, jonka avulla voidaan optimoida suurempia tilakonesäätimiä paremmalla suorituskyvyllä kuin aikaisemmilla menetelmillä.

Neljäs kontribuutio on POMDP-malli kognitiiviselle radiolaitteelle, joka lähettää vanhoille radiolaitteille varatuilla taajuuskanavilla. Malli ottaa huomioon vanhojen radiolaitteiden erilaiset purskepituudet ja reaktiot ja suoriutuu paremmin kokeissa kuin vertailumallit. Viidennessä kontribuutiossa usean laitteen langaton lähetys spatiaalisen häiriön alla muotoillaan faktoroiduksi DEC-POMDP:ksi. Tämä sallii optimoinnin sekä tilan että ajan suhteen, ja lähestymistavalla saavutetaan kokeissa parempia tuloksia kuin langattomaan tiedonsiirtoon käytetyillä vertailumenetelmillä.

Langattomien laitteiden päätösten laatu riippuu ratkaisevasti havaintojen hinnasta ja laadusta. Väitöskirjan viimeinen kontribuutio on uusi kaistantunnistustapa, joka käyttää nanoteknologiaan pohjautuvaa laskentaa. Uudessa tunnistustavassa koneoppimista käytetään nanomittakaavan vikojen vaimentamiseen ja signaalien luokitteluun.

Avainsanat Päätöksenteko epävarmuuden vallitessa, POMDP, DEC-POMDP, langaton verkko, WLAN, kognitiivinen radio, nanolaskenta

ISBN (painettu) 978-952-60-4998-4 **ISBN (pdf)** 978-952-60-4999-1

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2013

Sivumäärä 204 **urn** http://urn.fi/URN:ISBN:978-952-60-4999-1

Preface

This thesis has been carried out at the Department of Information and Computer Science (ICS) at Aalto University. In 2012, I also worked at Nokia Research Center. Nokia, Finnish Funding Agency for Technology and Innovation (TEKES), and the department funded the thesis work. I would like to thank these organizations very much. I am grateful for personal grants from the Nokia Foundation, the KAUTE Foundation, and the Finnish Foundation for Technology Promotion (TES). Furthermore, I am grateful for two conference trip grants from the Helsinki Graduate School in Computer Science and Engineering (HECSE).

This thesis was guided by Prof. Erkki Oja and instructed by Dr. Jaakko Peltonen. I am grateful to Prof. Oja for guidance, for advice, for always being available for discussions, and for always supporting and helping me. I am grateful to Dr. Peltonen, who is a co-author in all the publications in this thesis, of course for his great work in the publications, but also for instructing me in many things related to scientific work and especially for his very high investment of time.

I would also like to thank the other co-authors Dr. Mikko A. Uusitalo and Dr. Ari Hottinen. The thesis research work started in a machine learning and nanotechnology project initiated by Dr. Uusitalo and ended in a machine learning and wireless network project with Dr. Hottinen. In addition to joint research work, I thank Dr. Uusitalo for interesting discussions and being always available and Dr. Hottinen for the many enlightening discussions about wireless networking and many other, totally different, subjects.

I thank all the people at the ICS department, especially those I have interacted with, for a friendly working environment. I would also like to thank Nokia Research Center (NRC) for many years of collaboration and the friendly people at NRC.

On the personal front I thank my father Timo and mother Pirjo for always supporting me. I also thank my mother-in-law and father-in-law for help and all my friends for making life more enjoyable.

Finally, I would like to thank the love of my life, Anniina, for incredible support, for sad and many happy moments, and for two children.

Vantaa, January 10, 2013,

Joni Pajarinen

Contents

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Joni Pajarinen, Jaakko Peltonen, Ari Hottinen, and Mikko Uusitalo. Efficient Planning in Large POMDPs through Policy Graph Based Factorized Approximations. In *Proceedings of ECML PKDD 2010, the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Barcelona, Spain, Volume 6323, pages 1–16, Lecture Notes in Computer Science, Springer, September 2010.

II Joni Pajarinen and Jaakko Peltonen. Efficient Planning for Factored Infinite-Horizon DEC-POMDPs. In *Proceedings of IJCAI-11, the 22nd International Joint Conference on Artificial Intelligence*, Barcelona, Spain, pages 325–331, AAAI Press, July 2011.

III Joni Pajarinen and Jaakko Peltonen. Periodic Finite State Controllers for Efficient POMDP and DEC-POMDP Planning. In *Advances in Neural Information Processing Systems 24 (Proceedings of NIPS 2011)*, Granada, Spain, pages 2636–2644, December 2011.

IV Joni Pajarinen, Jaakko Peltonen, Mikko A. Uusitalo, and Ari Hottinen. Latent state models of primary user behavior for opportunistic spectrum access. In *Proceedings of PIMRC'09, the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Tokyo, Japan, pages 1267–1271, September 2009.

V Joni Pajarinen, Ari Hottinen, and Jaakko Peltonen. Optimizing spatial and temporal reuse in wireless networks by decentralized partially observable Markov decision processes. *Submitted to a journal*, 14 pages, October 1st 2012.

VI Jaakko Peltonen, Mikko A. Uusitalo, and Joni Pajarinen. Nano-scale fault tolerant machine learning for cognitive radio. In *Proceedings of MLSP 2008, the IEEE International Workshop on Machine Learning for Signal Processing*, Cancún, Mexico, pages 163–168, October 2008.

VII Joni Pajarinen, Jaakko Peltonen, and Mikko A. Uusitalo. Fault tolerant machine learning for nanoscale cognitive radio. *Neurocomputing*, Volume 74, issue 5, pages 753–764, January 2011.

Author's Contribution

Publication I: "Efficient Planning in Large POMDPs through Policy Graph Based Factorized Approximations"

The background research, the idea for the proposed method, the implementation of the proposed method and the comparison methods, and running of the experiments was by the author. The author co-designed the experiments. J. Peltonen suggested a practical implementation of one approximation in the algorithm, was involved in the discussion of other approximations, contributed to theoretical proofs, writing of the publication, and experiment design. A. Hottinen and M. A. Uusitalo took part in writing and discussions.

Publication II: "Efficient Planning for Factored Infinite-Horizon DEC-POMDPs"

The background research, the proposed method, the implementation of the proposed method and the comparison methods, and running of the experiments was by the author. J. Peltonen actively provided discussion and comments on the publication, participated in design of the experiments, and contributed to writing of the publication.

Publication III: "Periodic Finite State Controllers for Efficient POMDP and DEC-POMDP Planning"

The background research, the proposed methods, the implementation of the proposed methods and the comparison methods, and running of the experiments was by the author. J. Peltonen contributed actively to writing

of the publication, provided discussion and comments on the publication, and participated in design of the experiments.

Publication IV: "Latent state models of primary user behavior for opportunistic spectrum access"

The insufficiency of previous two-state Markov model methods for modeling traffic was identified by the author and the POMDP part was investigated mostly by the author. The proposed Markov model was designed in large part by J. Peltonen and the author. The author participated in experiment design. Experiments were implemented and run by the author. All authors participated in the problem setting. The publication was written together.

Publication V: "Optimizing spatial and temporal reuse in wireless networks by decentralized partially observable Markov decision processes"

A. Hottinen suggested optimization over the spatial dimension. It was jointly decided to optimize over both the time and spatial dimensions and the channel access problem was formulated together. The author, for the most part, designed how the continuous valued channel access problem can be efficiently formulated as a discrete factored DEC-POMDP and how existing DEC-POMDP methods can be modified to solve the channel access problem. The author designed the experiments together with the co-authors. The author implemented and ran the experiments. The publication was jointly written with the author writing the most.

Publication VI: "Nano-scale fault tolerant machine learning for cognitive radio"

All authors of the publication had equal contributions overall. The author did the background research on spectrum sensing and machine learning methods, and finding a method for spectrum sensing feature extraction. The author participated in experiment design, and implemented and ran the experiments. The design of the nano-scale approach was joint work. All authors participated in writing of the paper. M. Uusitalo suggested combining cognitive radio, nanotechnology and machine learning.

M. Uusitalo was the main contributor for the cognitive radio and nanotechnology problem setting and handled nanotechnology details. J. Peltonen was very active in the design of the fault model and in writing and participated in the application of machine learning methods.

Publication VII: "Fault tolerant machine learning for nanoscale cognitive radio"

All authors of the publication had equal contributions overall. Comparison methods were suggested by the author and additional experiments were implemented and run by the author. A general spectrum sensing enhancing change was suggested by the author. Additional experiments and additions to the fault model were discussed together. Otherwise, the contributions were similar to the conference paper "Nano-scale fault tolerant machine learning for cognitive radio".

List of Abbreviations

AWGN	Additive White Gaussian Noise
BPI	Bounded Policy Iteration
CMOS	Complementary Metal–Oxide–Semiconductor
COM-MTDP	Communicative Multiagent Team Decision Problem
CR	Cognitive Radio
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DEC-POMDP	Decentralized Partially Observable Markov Decision Process
E-step	Expectation step
EM	Expectation Maximization
FBVP	Factorized Belief Value Projection
FSC	Finite State Controller
FSVI	Forward Search Value Iteration
GPS	Global Positioning System
HSVI	Heuristic Search Value Iteration
I-POMDP	Interactive POMDP
M-step	Maximization step
MCVI	Monte Carlo Value Iteration
MDP	Markov Decision Process

MEMS	MicroElectroMechanical Systems
ND-POMDP	Network-Distributed POMDP
NEMS	NanoElectroMechanical Systems
OFDM	Orthogonal Frequency Division Multiplexing
PBPI	Point Based Policy Iteration
PBVI	Point Based Value Iteration
PI	Policy Iteration
POMDP	Partially Observable Markov Decision Process
POSG	Partially Observable Stochastic Game
PWLC	PieceWise Linear and Convex
RBFn	Radial Basis Function Network
RTDP	Real Time Dynamic Programming
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
TD-POMDP	Transition-Decoupled POMDP
TDMA	Time Division Multiple Access
VI	Value Iteration
VOIP	Voice Over IP
WLAN	Wireless Local Area Network

1. Introduction

Both humans and artificial agents, including robots and wireless devices, make decisions that may have far reaching implications. Agents make decisions based on incomplete observations about the world around them. Decision making in a complex uncertain world is a difficult problem even for advanced intelligent agents such as humans. When it is uncertain in which way the world will change in response to actions and when only incomplete information about the world is available, an optimally behaving agent has to consider a huge number of possible futures in order to find the plan that yields the greatest return. From this perspective the difficulty of optimal decision making is perhaps not surprising. Interestingly, multi-agent decision making is even more difficult: agents not only have to consider future events, but also what other agents are planning to do, which depends on past observations of the other agents.

In order to optimize its behavior in a world that changes, an agent has to *plan* its actions over a sequence of time instances. When an agent or multiple agents do not know the outcome of their actions beforehand, the decision making problem is one of *planning under uncertainty*. For instance, a robot could plan its route from its home to a factory by considering in which direction to turn at each crossing. The optimal plan would consist of a sequence of direction changes. In case the hardware of the robot was of low quality, and the robot would not turn with probability 0.05, when it tried to, the robot would have to plan under uncertainty. An optimal conditional plan would have to take many different sequences of correct and false turns into account, that is, the action suggested by the conditional plan would depend on the current location of the robot. While planning under uncertainty is common in many robotics and other real world applications [141, 32, 151, 139, 63, 34, 102], it is computationally very challenging, especially when the current world state is uncertain.

Therefore, efficient computational methods are needed.

One challenging real-world application domain for planning under uncertainty is *wireless networking*. In a wireless network, agents such as mobile phones, laptops, or cell towers transmit data to each other on different frequency channels. Wireless agents decide at each time instance which channels (if any) to sense and transmit on. Consider for example two humans talking to each other over a wireless network. When the first human speaks, the voice signal is translated into data, which is put into network packets. The network packets are transmitted over a wireless channel to the wireless device of the second human and translated back into a voice signal. The wireless devices decide when to sense channels and transmit data depending on the current wireless network state and of the data to transmit.

In a wireless network, several things influence decision making. The human user, the protocol stack, and wireless device hardware influence how data is generated for transmission. In addition, the spatial location of wireless devices and the surrounding environment influence how transmitted data travels to the intended receiver. Furthermore, sensing the environment is limited by energy and hardware constraints. These and other properties of the operating environment need to be taken into account when wireless agents plan actions into the future.

1.1 Overview

Next, the three main topics of the thesis: “Planning under uncertainty”, “Wireless channel access”, and “Spectrum sensing” will be introduced.

Planning under uncertainty. This thesis presents new decision making methods for agents, that act in a world with uncertainty. Figure 1.1 illustrates the sequential action-observation cycle for a single agent and for multiple agents. In the figure, agents act, the actions change the world, the agents then observe the changed world, and the cycle starts from the beginning. In real world problems, observations usually do not reveal everything about the world, that is, observations are partial/incomplete and noisy. For instance, a wireless agent may observe that another agent transmits data, but it does not know how much data the other agent still has in its transmit buffer. Also, usually, in real world problems changes to the world state are uncertain as illustrated with the example of a robot

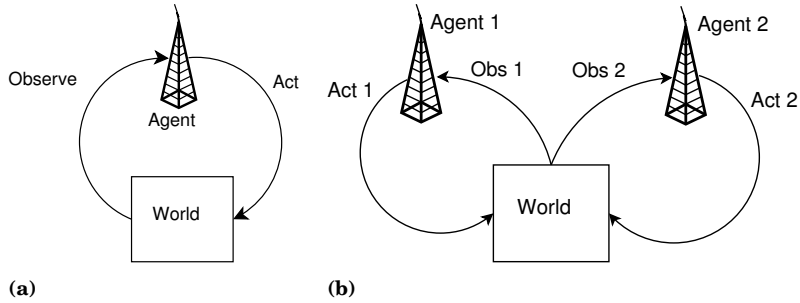


Figure 1.1. Action-observation cycles for a single agent (a) and multiple agents (b). The action-observation cycles repeat for a certain, possibly infinite, time. In (a) a single agent (depicted as a wireless device in this figure) performs an action, which influences the state of the world and then makes an observation about the new world state. In (b) two or more agents perform actions independently, which influence the state of the world and then each agent makes its own observation about the new world state.

navigating to a factory. Furthermore, actions of an agent may have far reaching implications: if the robot fails to pick up a tool at home that it needs to perform a function at the factory, it notices the failure only after arriving at the factory. A partially observable Markov decision process (POMDP) [141] is a general model for optimizing single agent decision making that takes into account both partial/incomplete observations, uncertainty in world state transitions, and the effects of the agent's actions. In a POMDP, world dynamics are defined using probabilistic Markov models [121] and the optimization objective is encoded as a reward that the POMDP assigns to the agent at each time step. Because solving POMDPs optimally is intractable except for very small problems, approximate approaches have been developed [112, 143, 140, 81].

A POMDP formalizes single agent decision making problems, but in order to optimize policies of multiple agents, such as wireless devices or robots, special problem properties need to be taken into account. In addition to uncertainty and partial observability, in many multi-agent problems agents act individually and do not share observations or actions with other agents. For instance, in a wireless network agents make individual channel access decisions based on their own channel sensing results. The agents may still optimize a joint objective, for example sum throughput in wireless networks. A decentralized POMDP (DEC-POMDP) [17] (see also [132]) is a model for optimizing co-operative multi-agent behavior that has received attention lately. A DEC-POMDP is a generalization of a POMDP to multiple agents. In addition to uncertainty and partial ob-

servability, a DEC-POMDP models agents that optimize a joint objective, but act and observe individually. These properties are crucial for many real world problems, but make policy computation hard [18].

Wireless channel access. As discussed earlier, a user that talks with another user over a wireless network connection or browses the web generates data that must be transmitted. When the user clicks a link to a web page, the web browser generates a request for the web page and encodes it as a data packet. The data packet travels through the protocol stack of the wireless device to the protocol layer that decides when and how to transmit the data. This thesis focuses on the problem of deciding whether to transmit or listen, and on which frequency channels. In wireless channel access the behavior of wireless agents corresponds to the action-observation cycle model shown in Figure 1.1: **1)** A wireless agent decides whether to access a channel, **2)** The decision of the agent influences the surrounding world, that is, the state of wireless devices, **3)** The wireless agent makes an observation on the world, that is, senses the channel.

Based on past observations an agent tries to determine, when to access a wireless channel. Many widely used standard protocols, such as IEEE 802.11 [1], specify how channel access decisions should be made. These protocols have been hand-crafted by experts and define explicitly when a wireless device may access the channel. A cognitive radio [61] on the other hand is a wireless device that can adapt to the environment it operates in and flexibly use frequency channels. One central problem in cognitive radio research is the simultaneous operation of cognitive radios and primary (high priority) users. Often a cognitive radio tries to utilize wireless channels in a new more efficient way, but at the same time has to prevent interference to primary users operating on the same channels. Some research [172, 53, 46] models cognitive radio channel access on primary user channels as a POMDP. A POMDP models naturally the stochasticity in primary user behavior and the partial observability caused by practical sensing limitations.

Wireless devices can optimize their behavior over the temporal and frequency dimensions, but also taking into account the spatial dimension. The interference from one wireless device to another depends heavily on the distance between the devices and devices far apart can transmit without causing interference to each other. Therefore, approaches for taking advantage of spatial opportunities have been proposed [174, 47, 48, 74,

73].

Spectrum sensing. As discussed previously a wireless agent first makes an observation, that is, it senses a frequency channel and then decides whether to access a channel. However, decision making is not only about deciding which channels to access, but also about when and which channels to sense. Good decisions on channel sensing at the current time help gather information that leads to better decisions in the future. The quality of actual spectrum sensing is the basis for making high value decisions. If the agent thinks a channel is free when it is actually occupied, a collision and loss of data happens. If the agent thinks a channel is occupied when it is not, a transmission opportunity is lost.

There is a wide array of research on spectrum sensing [169, 6] and different approaches for single and multi-agent sensing have been proposed. One difficult problem, wide bandwidth spectrum sensing with low power requirements, is addressed in this thesis.

1.2 Contributions

The contributions of this thesis can be divided into methodological planning under uncertainty contributions and into contributions specific to wireless networking. In the methodological category, Publication I presents a new factored POMDP method (factored means here that the problem is described in terms of several variables and how they depend on other variables) that can compute policies for very large problems and demonstrates its performance on benchmark problems and the wireless networking problem introduced in Publication IV. Publication II presents the first proposed method for planning with factored infinite-horizon DEC-POMDPs. The method solves large problems with many agents with good accuracy. Finally, Publication III shows how a new kind of policy type allows for larger policies and higher performance in both POMDPs and DEC-POMDPs.

In terms of contributions to wireless networking, Publication IV shows how to model primary user traffic in cognitive radio networks, where the goal is to choose the best channel to transmit on, with more realistic models than in earlier approaches. The new model yields significantly better performance compared to earlier simpler models when used for optimizing the POMDP policy of a cognitive radio. Publication V shows how to

take advantage of both temporal and spatial opportunities in channel access. Publication V formulates the channel access problem as a factored DEC-POMDP and computes policies for wireless agents, using the method of Publication II with modification and the new policy type introduced in Publication III. In experiments, the new approach yields higher performance than wireless channel access protocols. Publications VI and VII discuss a new passive nanoscale spectrum sensing approach for cognitive radio. The approach uses a radial basis function network to classify signals and to mitigate the effect of nanoscale faults.

1.3 Thesis structure

This thesis consists of a summary and original articles. The summary part of the thesis begins in Chapter 2 with background information on planning under uncertainty. Chapter 2 focuses on research in POMDP and DEC-POMDP methods. Next, Chapter 3 discusses the new POMDP and DEC-POMDP methods in Publication I, Publication II, and Publication III. Chapter 4 discusses cognitive radio background and the POMDP based cognitive radio approach in Publication IV, which is followed by channel access background and the DEC-POMDP based wireless channel access approach in Publication V. Chapter 5 presents background on spectrum access, nanotechnology, and fault tolerance and then the passive analog nanoscale spectrum sensing approach proposed in Publications VI and VII. Finally, the summary and discussion in Chapter 6 wraps the thesis up.

2. Background: Planning under uncertainty

As discussed in the previous chapter (see Figure 1.1) an agent has to decide on its current action based on what it has observed in the past. Optimal *decision making* is one of the fundamental questions of artificial intelligence research. The agent makes observations about the surrounding world and the agents' actions change the world. Because actions of an agent influence the world, an action may have significant repercussions in the future. A rational agent *plans* its actions into the future. In *classical planning*, [54] it is known beforehand how actions influence the world. Classical planning methods [54] try to find a sequence of deterministic actions that transitions the world, from a known initial state, to a certain predefined goal state. For example, in a factory problem the goal could be to decide on the sequence of machines to use to manufacture a car. However, in practice it is often uncertain whether a machine completes it job or not. When the world is not fully deterministic, *planning under uncertainty* is needed. The world may change in many different ways and an optimal plan is no longer a sequence of actions, but instead a conditional plan. Intuitively a conditional plan is a tree of actions: the agent performs an action prescribed by the current tree node and then follows the tree branch, which corresponds to the observation made about the world.

In classical planning, the agent tries to reach a predefined goal, but practical problems may have a goal for each time step. For example, in a wireless network problem, the goal of an agent may be to transmit in each time step as much data as possible. In *reinforcement learning* [71, 145], the agent is assigned a reward at each time step depending on the current state of the world, and the goal of an optimal agent is to collect high rewards over a long (possibly infinite) period of time. In this thesis, planning under uncertainty is discussed in the context of reinforcement learning. Reinforcement learning can be divided into model-free

and model-based approaches. In model-free reinforcement learning, the agent does not have a model of the environment it operates in, but instead tries to improve its *policy*, that is, how actions are chosen, while gathering knowledge of the environment. A model-free agent has to consider, among other things, the exploration-exploitation trade-off: the agent can collect information about the environment and emphasize maximizing rewards later, or based on current knowledge emphasize immediate reward collection. In model-based reinforcement learning, the agent optimizes its policy based on a model of the environment. When the model is known, a policy can be optimized, and then immediately used without extra information gathering. The decision making methods presented in this thesis are model-based.

As discussed above, in reinforcement learning the goal is to maximize gathered reward over time. When selecting actions it is important to consider the effects of the actions into the future: remember the example of a robot that needs to pick up a tool that is needed only later. In order to optimize the gathered reward into the future, one needs a way to reason about the current and future states of the world. One way is to assume that only the current state of the world contains all the information required for predicting the probability of the next state of the world and that no additional history information is required, that is, the world evolves according to a Markov process. This thesis focuses on models of decision making with this Markov assumption. There are also other alternative models for decision making under uncertainty. For instance, predictive state representations [88] use predictions of future observation sequences, instead of Markovian states, to model the world. In this thesis, it is assumed that the Markov model is known, that the model has a discrete set of states, actions, and observations, and that the model is stationary so that it does not change (this includes models that are descriptive enough to describe possible changes).

2.1 Hierarchy of decision processes

In this section, Markovian decision processes are categorized according to the observability of the world state and according to the number of agents. Differences between the categories are illustrated using a wireless network problem example and factored Markovian decision process models that can describe large real-world problems are discussed.

When the agent can observe the world state fully, the Markovian decision process is called a Markov decision process (MDP). Consider a wireless channel access problem, where the agent has to choose a channel to transmit on in the next time step. Each channel has a certain probability for moving between idle and occupied states. The probabilities represent traffic characteristics of other devices and depend on whether the agent accesses the channel or not. The goal is to transmit on channels that other wireless devices do not use. If the agent can always observe the state of all channels fully, then the problem is an MDP (MDPs will be discussed in Section 2.2) and it is straightforward to compute the probability of the next state of a channel. However, because of hardware limitations, wireless agents cannot in practice observe all channels at once and channel measurements are noisy. In addition to predicting how the agent's actions will influence the state of the world in the future, with partial observations, the agent has to consider how to gather information about the state of the world. With partial observations the decision process is called a partially observable MDP (POMDP). POMDPs will be discussed in Sections 2.3–2.5. An even more difficult problem arises when the goal is to optimize the behavior of two or more agents. An agent does not know what observations other agents have made or what actions the other agents have performed and thus also not what other agents intend to do next. When there are multiple decision makers sharing the same reward function, but each decision maker has its own observations and actions, the decision process is called a decentralized POMDP (DEC-POMDP). DEC-POMDPs will be discussed in more detail in Sections 2.6–2.9.

Figure 2.1 shows influence diagrams for MDPs, POMDPs, and DEC-POMDPs. The details and semantics of the models will be discussed in the relevant sections later, but for now the figure illustrates how the dependencies become more complicated for partially observable and decentralized cases. Figure 2.2 illustrates a hierarchy of Markovian decision processes. As also mentioned in the figure caption, a DEC-POMDP is a special case of a partially observable stochastic game (POSG) [60], which allows also competitive settings. A decentralized MDP (DEC-MDP) is the fully observable special case of a DEC-POMDP and a POMDP is the single agent special case of a DEC-POMDP. An MDP is the fully observable special case of a POMDP and at the same time the single agent special case of a DEC-MDP.

POMDPs and DEC-POMDPs allow for a broad range of practical appli-

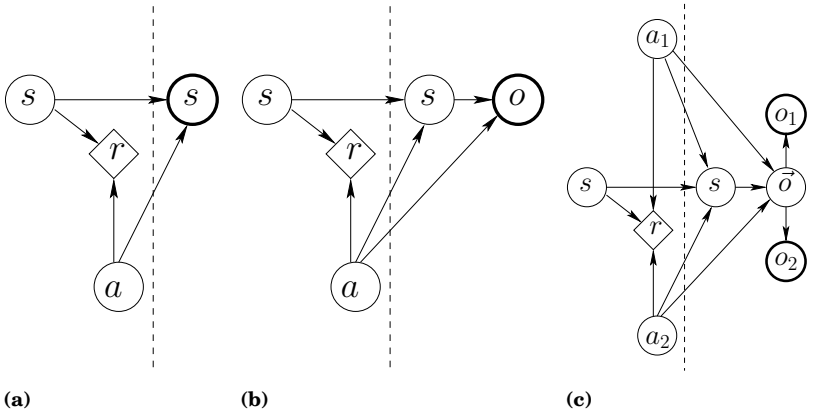


Figure 2.1. Influence diagrams of (a) a Markov decision process (MDP), (b) partially observable MDP (POMDP), and (c) a two agent decentralized POMDP (DEC-POMDP). s denotes the world state and r the reward. In an MDP and POMDP, π denotes the agent's policy, a the agent's action, and in a POMDP o denotes the observation. In a DEC-POMDP with two agents, a_1 and a_2 denote the agent actions, π_1 and π_2 denote the policies of the agents, $\vec{o}$ the joint observation, and o_1 and o_2 individual observations. Thick circles illustrate variables that the agent(s) observe. In an MDP, the agent observes the world state directly, but in a POMDP and in a DEC-POMDP the agent(s) make(s) observation(s) that only indirectly reflect(s) the real world state. An agent chooses actions based on its past observation history and according to its policy. In the figure, a dotted line separates time steps.

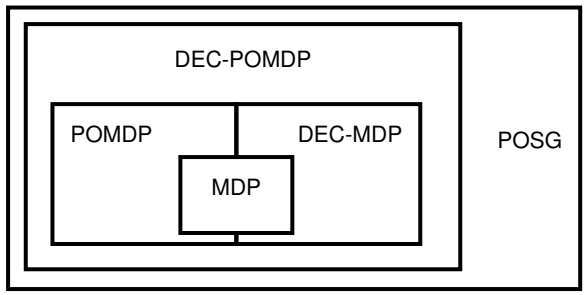


Figure 2.2. Hierarchy of Markovian decision processes. A DEC-POMDP is a special case of a partially observable stochastic game (POSG) [60], which allows also competitive settings. A decentralized MDP (DEC-MDP) is the fully observable special case of a DEC-POMDP and a POMDP is the single agent special case of a DEC-POMDP. An MDP is the fully observable special case of a POMDP and at the same time the single agent special case of a DEC-MDP.

cations, such as optimizing wireless network operation ([102] and Publication V), robot control and navigation [32, 151], rover sample collecting [139], elder care [63], tiger conservation [34], manufacturing [141], and many other kinds of problems. Cassandra [31] provides a survey of possible POMDP applications.

The POMDP and DEC-POMDP models allow a formal description of optimal decision making in complicated problems. However, in real-world POMDP and DEC-POMDP applications the size of a problem description is often huge, because the state-space grows exponentially with respect to the number of state variables. This exponential growth of the state-space is often called the state-space explosion problem. In the wireless network example discussed previously, in a realistic implementation each channel can be in 15 different internal states, but the complete world state space is then of size 15^N, where N is the number of channels. In practice, it is not possible to specify the complete probability table in such a problem. In factored (MDP, POMDP, DEC-POMDP, ...) models, state, action, observation, and controller variables are divided into sets of individual variables. For example, in the wireless network problem with multiple channels, each channel is a separate state variable and the full world state is the cross product of the individual variables, but probabilities are specified using a subset of all the state variables. The factored description of real-world problems is often compact and allows taking advantage of the factored form during planning. Publication I and Publication II present new methods for efficient planning in factored POMDPs and DEC-POMDPs, respectively.

This chapter discusses first what a Markov decision process (MDP) is and then proceeds to the more general partially observable Markov decision process (POMDP). Then the chapter discusses POMDP policy representations, solution techniques for POMDPs, and approaches for solving factored POMDPs. From POMDPs the chapter moves on to decentralized POMDPs (DEC-POMDPs). The chapter discusses solution techniques for finite-horizon and infinite-horizon DEC-POMDPs, and concludes with factored DEC-POMDPs.

2.2 Markov decision process (MDP)

This section formally defines an MDP as a starting point, which will be used in later sections as a building block for more general models. A Markov decision process (MDP) is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, s_0 \rangle$, where $\mathcal{S}$ is the set of world states, $\mathcal{A}$ is the set of actions of the agent, P denotes transition probabilities, R is a real valued reward function, and s_0 is the starting state. P and R will be defined in more detail below. In general, MDPs can be defined over continuous or discrete valued states and

actions, but in this thesis states and actions are discrete valued. Time can also be considered either continuous or discrete valued. This thesis considers discrete time denoting the current time step by t and the next time step by $t+1$. The current state at time t is denoted by s and the state at time $t+1$ by s'. Denote with a the action of the agent. $P(s'|s,a)$ is the probability to move from state s to the next state s', given the action a. $R(s,a)$ is the real-valued reward for executing action a in state s. In MDPs, the current state s is fully observable and an optimal policy π^* is a mapping $\pi^*(s) = a$ from states to actions that maximizes the reward objective. The reward objective is to maximize a cumulative sum of rewards over a certain time period. Note that the optimal policy depends on the particular reward objective chosen (possible reward objectives are discussed below). Note that rewards are a property of the computational model and there is no need for an actual reward signal. For instance, the computational model may specify that a robot is assigned a reward of twenty if the robot finds hundred dollars. In an MDP, the obtained immediate reward is always known, because the world state is fully observable. When the world state is partially observable, the obtained actual immediate reward is in general not known. However, if a robot thinks it has found hundred dollars with a probability of 0.5, it does not know the actual reward, but it knows the expected reward is ten.

For some problems, rewards are gathered over a fixed time span called the horizon. For these problems, a finite-horizon reward objective can be used:

$$E\left[\sum_{t=0}^{T-1} R(s(t), a(t))|\pi\right] , \tag{2.1}$$

where T is the horizon, $s(t)$ is the state and the action $a(t)$ at time step t is chosen by the policy π. For other problems, such as wireless network channel access, the execution of the policy can go on for an infinite time in practice. In infinite-horizon MDPs, the discounted reward objective is

$$E\left[\sum_{t=0}^{\infty} \gamma^t R(s(t), a(t))|\pi\right] , \tag{2.2}$$

where $0 < \gamma < 1$ is the discount factor. The discount factor makes computations tractable. Another optimization objective, which is used in infinite-horizon problems, is the average reward objective [13, 168].

This thesis focuses on discounted reward infinite-horizon problems, although a finite-horizon objective is used as initialization for an infinite-horizon approach in Publication III. Many solution techniques make use

of the Bellman optimality equation. Denote with $V^{\pi}(s)$ the value, that is, the expected cumulative reward, when starting from state s and following the policy π. The Bellman optimality equation

$$V^*(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right] \tag{2.3}$$

recursively defines the optimal value function $V^*(s)$. When the optimal value function is known, the optimal action is simply the one that maximizes the right-hand-side of the Bellman equation.

The computational complexity of MDPs is P-complete [105] for both of the reward objective definitions above and thus an MDP can be solved (solving refers here to finding an optimal policy) efficiently for reasonably sized state and action spaces. There are several ways for solving MDPs. In *value iteration* [16], the value function is initialized to low values and then the Bellman equation is used repeatedly to compute new value functions until convergence. Value iteration converges to the optimal value function. In *policy iteration* [64], the best policy is computed for the current value function and the new policy is then used to compute a new value function. This procedure of computing a policy and then a value function is repeated until the policy does not change. An MDP may also be solved using linear programming [21].

2.3 Partially observable Markov decision process (POMDP)

A POMDP [142, 72] allows an agent to make optimal decisions in an uncertain world with noisy and partial observations. POMDPs generalize MDPs to partially observable states. The worst case computational complexity of finite-horizon POMDPs is PSPACE-complete [105] and infinite-horizon POMDPs are undecidable [90]. Because of the high computational complexity, optimal solutions are attainable only for small problems, and state-of-the-art POMDP solvers ([112, 143, 140, 81] as well as Publication I and Publication III) use various approximations. Recent POMDP solvers have had great success, but problem size still remains a major obstacle for many real world problems. Interestingly, some POMDP problems are easier to approximate than others [65].

Formally a POMDP is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, P, R, O, b_0 \rangle$. Similarly to an MDP, $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $P(s'|s,a)$ is the probability to move from state s to the next state s', given the action a, and $R(s,a)$ is the real-valued reward for executing action a in state s. In

a POMDP, $\mathcal{O}$ is the finite set of observations and O denotes the observation probabilities $P(o|s', a)$, where o is the observation made by the agent, when action a was executed and the world moved to the state s'. Lastly, $b_0(s)$ is the initial state distribution, also known as the initial *belief*.

In this thesis, a single probability distribution over world states is called either a belief or *belief point* and the space over all possible beliefs is called the *belief space*. Although the current state is not known in a POMDP, the belief is known at each time step. After performing action a and observing o the updated belief $b' = b'(s'|b, a, o)$ can be obtained from the current belief $b = b(s)$ using the Bayes formula

$$b'(s'|b, a, o) = \frac{P(o|s', a)}{P(o|b, a)} \sum_s P(s'|s, a)b(s) \ , \tag{2.4}$$

where $P(o|b, a) = \sum_{s'} P(o|s', a) \sum_s P(s'|s, a)b(s)$ is the normalization constant. A POMDP is called a belief state MDP, because the continuous belief of a POMDP can be regarded as a continuous state in an MDP.

2.3.1 POMDP example

As a concrete POMDP example, consider a heavily simplified version of the POMDP problem discussed in Section 4.1.2, where the goal of a wireless secondary user (SU) is to transmit on a wireless channel, when the primary user (PU) of the channel is not active. The PU can either be active, meaning it is transmitting, or idle, meaning it does not transmit. In this problem, a two state Markov model describes the idle and active periods of the PU. The PU moves from an idle to an active state with probability 0.1 and from the active state to the idle state with probability 0.2. The SU can either listen to the channel or transmit on the channel. If the SU transmits when the PU is active, a negative reward (penalty) of -5 is given and the PU remains active (the PU tries to access the channel until it succeeds). Successful SU transmissions yield a positive reward of $+1$. In each time step, the SU makes a correct observation, on whether the PU is active, with probability 0.8. Because the observations are noisy, the SU should not decide on actions based only on the current observed channel state, but should make decisions using the current belief distribution which contains information also about past actions and observations. Moreover, in general, optimal decision making requires taking possible future (channel) states into account. Table 2.1 shows the formal definition of this POMDP problem and Table 2.2 illustrates how the belief would evolve over time for a series of actions and observations.

(a) $P(s'|s, a = \text{listen})$:

s \ s'	idle	active
idle	0.9	0.1
active	0.2	0.8

(b) $P(s'|s, a = \text{transmit})$:

s \ s'	idle	active
idle	0.9	0.1
active	0	1

(c) $P(o|s', a)$:

s' \ o	idle	active
idle	0.8	0.2
active	0.2	0.8

(d) $R(s, a)$:

a \ s	idle	active
listen	0	0
transmit	$+1$	-5

Table 2.1. The example POMDP, discussed in Section 2.3.1, is formally defined by the state set $\mathcal{S} = (\text{idle}, \text{active})$, the action set $\mathcal{A} = (\text{listen}, \text{transmit})$, the observation set $\mathcal{O} = (\text{idle}, \text{active})$, the transition probabilities shown in sub-figures (a) and (b), the observation probabilities shown in sub-figure (c), the reward function shown in sub-figure (d), and the initial belief $b_0(s) = 0.5$.

Time	0	1	2	3	4	5	6	7
o	–	active	active	idle	idle	active	idle	active
a	listen	listen	listen	listen	transmit	listen	listen	listen
$b(s = \text{I})$	0.50	0.23	0.13	0.62	0.87	0.48	0.82	0.46
$b(s = \text{A})$	0.50	0.77	0.87	0.38	0.13	0.52	0.18	0.54

Table 2.2. Illustration of the evolution of the belief $b(s)$ for the example POMDP problem discussed in Section 2.3.1 and formally defined in Table 2.1. The world state s (state of the primary user) is either idle (denoted with I) or active (denoted with A). In each time step, the agent makes an observation o (idle or active) and executes an action a (listen or transmit). The agent updates its belief $b(s)$ using only the action a it executed, the observation o that followed, and the current belief.

2.3.2 POMDP value function

In POMDPs, similarly to MDPs, the optimal value function yields an optimal action for the agent. By defining the value function $V(b)$ over beliefs the resulting Bellman equation for a discounted POMDP is

$$V^*(b) = \max_a \left[\sum_s R(s,a)b(s) + \gamma \sum_o P(o|b,a)V^*(b'(s'|b,a,o)) \right] . \qquad (2.5)$$

For finite-horizon POMDPs, the optimal value function is piece-wise linear and convex (PWLC) [138] over the space of beliefs and for infinite-horizon discounted POMDPs the value function can be approximated arbitrarily closely with a PWLC function. This is of practical relevance since most POMDP approaches [112, 139, 143, 81] take advantage of the PWLC property in some form or another.

Because the value function of a POMDP is PWLC, the value function can be represented as a set of vectors, commonly called α-vectors. Each

α-vector corresponds to a conditional plan (more on this later), that starts with a certain action. The value of a belief is the maximal dot product of the belief $b(s)$ with one of the α-vectors. The best α-vector α_a^* for a belief $b(s)$ is

$$\alpha_a^* = \operatorname*{argmax}_{\alpha_a^i} \sum_s b(s)\alpha_a^i(s) \ , \tag{2.6}$$

where $\alpha_a^i(s)$ is the ith α-vector with action a. The best action is the action of the best α-vector.

A common approach for solving POMDPs is to compute a policy consisting of α-vectors offline. Then, during online operation, execution starts from the initial belief and the current belief is updated at each time step using Equation 2.4. Inserting the current belief into Equation 2.6 yields the best action. In addition to policies in vector form, policies in the form of a graph (graphical policies) are commonly used. A graphical policy can be executed without maintaining a belief or an explicit value function. Figure 2.3 shows different kinds of graphical policies. How α-vectors are constructed and how α-vectors relate to graphical policies is discussed next.

2.4 POMDP approaches

As discussed above, in POMDPs the policy can be represented as a set of α-vectors. Many POMDP methods start from a small set of α-vectors and iteratively grow the set. The *backup* operation is an elementary operation for constructing new α-vectors from known α-vectors. The backup operation constructs for time step t a new α-vector from a set of α-vectors in the next time step $t+1$. The idea is to find for a given action a, in the current time step, an α-vector $\alpha_{a_o}^o(s')$ in the next time step for each possible observation o (it will be discussed shortly how POMDP methods choose $\alpha_{a_o}^o(s')$ for each observation). The new α-vector $\alpha_a^i(s)$ is then constructed from the next time step α-vectors and the immediate reward $R(s,a)$:

$$\alpha_a^i(s) = R(s,a) + \gamma \sum_o \sum_{s'} P(o|s',a)P(s'|s,a)\alpha_{a_o}^o(s') \ . \tag{2.7}$$

When performing backup for a specific belief $b(s)$, the next time step α-vector $\alpha_{a_o}^o(s')$ for observation o is chosen as

$$\alpha_{a_o}^o(s') = \operatorname*{argmax}_{\alpha_{a_j}^j(s')} \sum_{s,s'} \alpha_{a_j}^j(s')P(o|s',a)P(s'|s,a)b(s) \ . \tag{2.8}$$

α-vectors and policy graphs are directly related (see Publication I for more discussion on the subject and Figure 2.3 for a policy graph example). In-

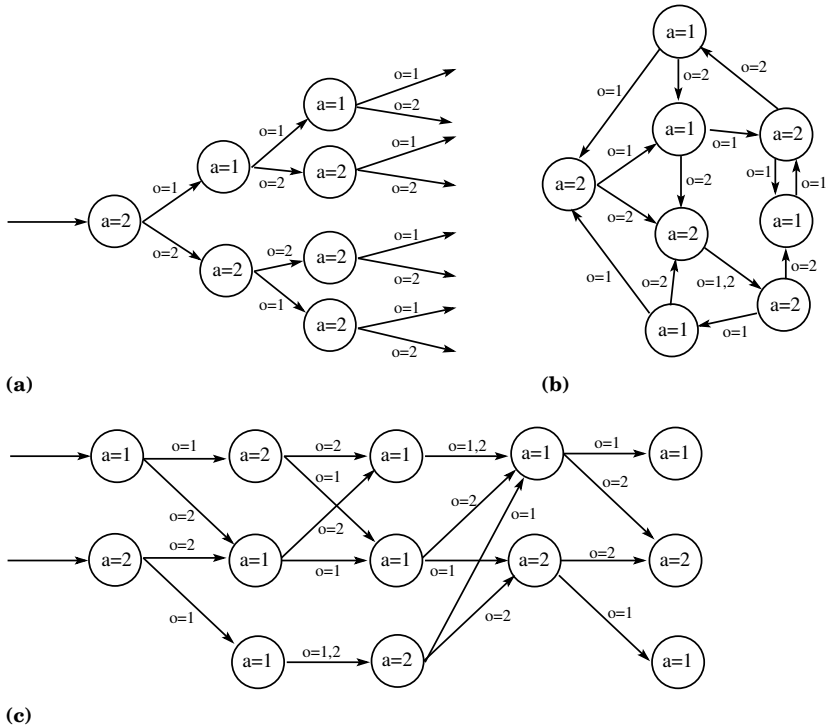


Figure 2.3. (Deterministic) examples of different types of graphical POMDP policies. A graphical policy consists of nodes and directed edges. At each time step the agent executes action a at the current node, the POMDP moves from state s to state s' with probability $P(s'|s,a)$, the agent makes observation o with probability $P(o|s',a)$, and the policy moves to a new node along the edge with o. (a) In a policy tree, the execution begins at the root node on the left and continues to the right. The execution branches at each node according to the observation made by the agent. Especially online POMDP methods discussed in Section 2.4.4 use policy tree based policy representations. (b) In a finite state controller (FSC), the execution moves from any node to any other node along the edge with the observation made. Section 2.4.3 discusses FSCs in general and methods for constructing them. Publication II and Publication III use FSCs as policy. (c) A policy graph resembles a policy tree, but a node may have several incoming edges. In Publication I the policy is represented as a policy graph only.

tuitively in an α-vector backup one finds the best action and the best previous alpha vectors for each observation. Assume that each policy graph node corresponds to an α-vector. To add a new policy graph node before the first layer of nodes, one has to find the best action, and for each observation find the best first-layer node (remember that a node corresponds to an α-vector). Adding a new node to a policy graph is thus similar to adding a new α-vector to a set of α-vectors.

2.4.1 Optimal POMDP methods

There are several ways for constructing optimal α-vector policies. The brute force approach iteratively constructs a set of α-vectors, whose size grows fast. Some optimal methods [33, 72] avoid redundant α-vectors. An α-vector is redundant if another set of α-vectors yields higher value over the complete belief space. Some methods [33] generate all possible α-vector backups and then prune redundant α-vectors. Some methods [72] try to identify beliefs, for which a backup generates a redundant α-vector. Optimal methods scale to POMDPs with only a few states. Point based approximate methods, discussed next, scale to larger problems.

2.4.2 Point based methods

For practical problems, it is computationally too demanding to construct α-vectors over the complete belief space. Equation 2.8 shows how to perform an α-vector backup efficiently for a single belief point. Point based POMDP solvers [112, 143, 140, 134, 81] do not try to find a policy over the complete belief space, but instead restrict policy search to a limited number of belief points. Point based POMDP solvers scale to state spaces with thousands of states [140, 134, 81].

The point based method Perseus by Spaan et al. [143] samples a set of beliefs and then improves the policy, a set of α-vectors, for the fixed set of beliefs until convergence. In an iteration, Perseus tries to improve the value of all sampled beliefs by backing up α-vectors for each belief. Identical α-vectors are pruned. Because Perseus improves all beliefs in one iteration, backup operations can be done in parallel resulting in a computationally efficient implementation. The complexity of one Perseus iteration is polynomial, because the number of α-vectors is bounded by the size of the belief set.

Point based value iteration (PBVI) [112] starts from an initial set of beliefs, performs backups for the beliefs, and then adds new beliefs to the belief set and starts a new backup round. The beliefs to be added are beliefs selected from a set of sampled beliefs, such that the beliefs have a large distance to the previous beliefs. Heuristic search value iteration (HSVI) [139, 140] maintains a set of α-vectors as a lower bound and a set of belief-value pairs as a value function upper bound. HSVI uses the bounds in an efficient (deterministic) belief generation heuristic. Forward search value iteration (FSVI) [134] samples belief trajectories from the

initial belief by selecting actions based on the optimal MDP policy and computes an α-vector for each sampled belief using the backup operation discussed earlier. Similar to HSVI, SARSOP [81] also maintains lower and upper bounds and uses those for generating new beliefs. SARSOP's new sampling and pruning procedures (see [81] for details) are motivated by the notion of optimally reachable belief spaces introduced in [65]. SARSOP yields better results than earlier POMDP methods (see [81] and Publication III).

2.4.3 Finite state controllers

A set of α-vectors is a common policy representation for POMDPs. Another kind of policy representation is the finite state controller (FSC). An FSC is a finite state machine that takes as input observations and outputs actions. Figure 2.3b displays an example FSC. During online operation an FSC does not need to update a belief about the world state, and it does not require a search for the best action for the current belief, thus online use of an FSC is extremely light weight. Humans can also interpret compact FSCs more easily than a set of vectors.

Formally an FSC is defined by a set of controller states (also called nodes, because of the graphical structure of the policy), the start distribution $P(q)$, where the distribution is over the controller state variable q, the conditional action distribution $P(a|q)$, and the conditional transition distribution $P(q'|q, o)$, where q' is the controller state in the next time step. In time step zero, the FSC starts from state q according to the probability distribution $P(q)$. In time step t, the FSC is in some known controller state q, and the agent then performs action a according to probability $P(a|q)$, the action influences the world and the agent makes observation o about the world. The FSC then transitions from current state q to the next time step state q' according to the conditional probability $P(q'|q, o)$.

There are several FSC based POMDP methods ([58, 96, 115, 69, 9, 12, 11, 152, 14] and Publication III). The policy iteration (PI) method in [58] transforms the current FSC policy to α-vectors, backs the α-vectors up, and then modifies the FSC using the backed up α-vectors. Point based policy iteration (PBPI) [69] is similar to PI, but does backups at a selected set of beliefs. Bounded policy iteration (BPI) [115] uses linear programming to improve a stochastic FSC. Gradient ascent [96] methods represent the FSC in a form (softmax parameter representation for example) for which the gradient can be computed and improve the policy by following the

gradient.

In the non-linear programming approaches [9, 12, 11], the optimization problem is written as a non-linear program and an off-the-shelf solver is used to optimize the FSC parameters. Expectation maximization (EM) [152, 153, 79] transforms the optimization problem into an inference problem and improves the FSC in each iteration. Non-linear programming and EM are applicable to both POMDPs and DEC-POMDPs. Because a POMDP is the single agent special case of a DEC-POMDP, non-linear programming and EM will be discussed in more detail in the context of DEC-POMDPs in Section 2.8.

Publication III introduces a new type of FSC, periodic FSCs, for both POMDPs and DEC-POMDPs. A periodic FSC is composed of layers of controller nodes, which are connected only to the next layer. The last layer is connected to the first layer. The periodic FSC approach yields state-of-the-art results. Periodic FSCs are discussed in more detail in Section 3.3.

2.4.4 Other POMDP approaches

Bonet and Geffner [25] show how discounted POMDPs can be transformed into Goal POMDPs (Goal POMDPs have absorbing goal states with zero immediate reward) and compare the real time dynamic programming (RTDP-Bel) approach for Goal POMDPs with point-based solvers for discounted POMDPs. The RTDP-Bel approach discretizes the belief space and uses a hash table as value function approximation. It samples beliefs using the current best action and updates the value function approximation for each sampled belief.

Online methods [160, 92, 133, 106, 122, 123, 124] do not compute a policy offline, meaning in advance, but instead make decisions during online operation. One advantage of online planning is that the current belief can be used as a starting point for planning and planning can concentrate on a small part of the belief space. A disadvantage is that planning is needed at each time step, which can be prohibitive in some applications (mobile phones for example may have hard restrictions on computing resources). During offline planning, the planner has to consider all beliefs that the agent with an optimal policy can visit. A simple online method would construct a complete policy tree (see Figure 2.3 for a policy tree example), where each node is an action, and each edge an observation. This tree has exponential size with respect to the planning depth. State-of-the-art POMDP solvers improve the simple method substantially, but a more

detailed discussion is out of the scope of this thesis.

2.5 Factored POMDP

Factored POMDPs ([113] and Publication I) provide a framework for making optimal decisions in settings, where the world state and observations can be described using several state and observation variables. A factored POMDP description can be compact even though the actual state space is huge. The compactness is possible because the probability for a single variable depends only on a subset of all variables. For example, in a wireless network problem with multiple frequency channels, the next state of a single channel depends on the previous state of the channel and on the action of the agent, but not on the state of other channels. A factored POMDP allows a compact description of many real-world (and benchmark) problems for which a "flat" description would not fit into memory.

More formally, in a factored POMDP the state consists of several state variables $s_1, s_2, \ldots, s_M$ and an observation of several observation variables $o_1, o_2, \ldots, o_M$. The probability for state variable s_i to transition to s'_i is $P(s'_i|\mathrm{Parents}(s_i))$ where $\mathrm{Parents}(s_i) \subset (a, s_1, s_2, \ldots, s_M)$ denotes the parent variables. $P(o_i|\mathrm{Parents}(o_i))$ denotes the probability for the agent to observe o_i and $\mathrm{Parents}(o_i) \subset (a, s'_1, s'_2, \ldots, s'_M)$ denotes the parent variables of o_i. The reward function $R(s, a)$ is a sum of reward sub-functions $R(s, a) = \sum_i R_i(S_i)$, where $S_i \subset (a, s_1, s_2, \ldots, s_M)$. Figure 2.4 shows the influence diagram of an example factored POMDP.

In a factored POMDP, variables depend directly only on a subset of other variables, but over several time steps the influence of a variable may spread to all other variables. Intuitively, variable dependencies form a graph, in which a path from one variable to another means that the variables depend, at least over time, on each other. Because influence spreads over time, the major problem that most factored POMDP methods try to solve is how to keep the belief and the policy in a compact form during planning and execution. There are several ways for making beliefs tractable. One popular approximation for beliefs is the Boyen-Koller [27] approximation, which breaks dependencies between all state variables resulting in a product of probabilities of individual state variables: $b(s) = \prod_i b_i(s_i)$.

Linear value functions are one choice for an approximate policy. The

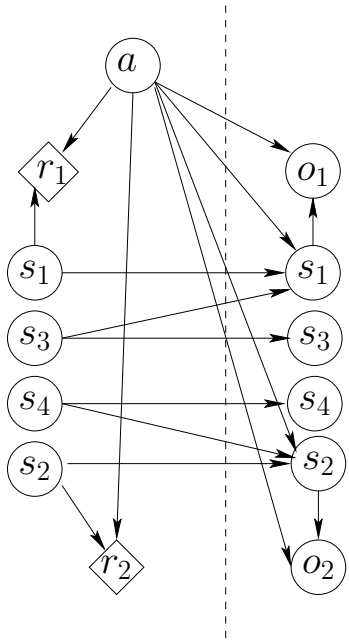


Figure 2.4. Influence diagram of an example factored POMDP. The dotted line separates two time slices. a is an action, r_1 and r_2 rewards, $s_1, \ldots, s_4$ state variables, and o_1 and o_2 observation variables.

paper [57] proposes to approximate the policy with a linear value function $V(s) = \sum_i V_i(S_i)$, where $S_i \subset (s_1, s_2, \ldots, s_M)$, that is, each sub-function $V_i(S_i)$ is restricted to a small subset of variables. Guestrin et al. [57] propose linear programming methods to optimize the policy for factored POMDPs, but does not present experimental results.

Boutilier and Poole [26] discuss how to represent α-vectors as value trees and probability tables compactly as Bayesian networks. Concerning graph based probability and value function representations, an algebraic decision diagram (ADD) [62, 59, 113] is a directed graph, where leaf nodes represent values or probabilities. The transition and observation probabilities and the reward function of a factored POMDP can be represented compactly as ADDs. Additionally, operations between ADDs are computationally efficient.

Symbolic Perseus [113] is similar to the Perseus [143] method, but it uses ADDs to represent probability tables, rewards, and α-vectors. Symbolic Perseus scales to large problems and has been successfully used for solving complicated problems in elder care [63]. Symbolic HSVI [137] adds an additional ADD based upper bound for the value function.

Several POMDP methods compute policies for large POMDPs without

considering factored structure explicitly. Compression methods [114, 116, 113, 126, 86] compress the probability and reward matrices and solve the POMDP problem for the compressed matrices (Publication I shows experimental results for an ADD implementation of the truncated Krylov iteration compression method discussed in [113]). Monte-Carlo value iteration (MCVI) [14, 87] represents the policy as a finite state controller and uses sampling to estimate values in value function backups. Sampling circumvents problems with large (and continuous) state spaces. Finally, the POMCP method in [136] uses Monte-Carlo tree search with Monte-Carlo sampling to plan online in large POMDPs. POMCP requires only a black-box simulator.

2.6 Decentralized partially observable Markov decision process (DEC-POMDP)

The policies of multiple agents, such as wireless devices [102] or robotic fire fighters [103], in a partially observable environment, can be optimized using a decentralized POMDP (DEC-POMDP; [17, 132]). A DEC-POMDP is a generalization of a POMDP to multiple co-operative agents. The agents share the same reward function, but each agent performs actions on its own and makes its own observations. There is no explicit communication in a DEC-POMDP among agents. The worst case computational complexity for finite-horizon DEC-POMDPs is NEXP-complete [18] and infinite-horizon DEC-POMDPs are undecidable (this is because infinite-horizon POMDPs are undecidable [90] and a POMDP is a special case of a DEC-POMDP). Contrary to POMDPs, in DEC-POMDPs it is not possible for an agent to make optimal decisions based only on a probability distribution over world states. In order to construct optimal plans, the action-observation histories of all agents have to be considered and one has to consider possible future action-observation sequences. This explains why DEC-POMDP problems are computationally so challenging. The complexity will be illustrated further in the context of finite-horizon DEC-POMDPs in Section 2.7.

Formally a DEC-POMDP with N agents is defined by the tuple $\langle \mathcal{S}, \mathcal{A}_1, \ldots, \mathcal{A}_N, \mathcal{O}_1, \ldots, \mathcal{O}_N, P, R, O, b_0 \rangle$. $\mathcal{S}$ is the set of world states, $b_0(s)$ is the initial probability distribution over the world states, and $P(s'|s, \vec{a})$ is the probability to move from state s to state s', given the actions $\vec{a} = (a_1, \ldots, a_N)$ of all agents. $R(s, \vec{a})$ is the real-valued reward for executing

actions $\vec{a}$ in state s. In a DEC-POMDP, $\mathcal{A}_i$ and $\mathcal{O}_i$ are the finite sets of actions and observations of agent i and O the observation probabilities $P(\vec{o}|s', \vec{a})$, where $\vec{o} = (o_1, \ldots, o_N)$ are the respective observations made by each agent, when actions $\vec{a}$ were executed and the world moved to state s'.

In some DEC-POMDP problems, the policy is executed over a limited time span, that is, a finite horizon which is known in advance. The optimization goal for a finite-horizon DEC-POMDP is

$$E\left[\sum_{t=0}^{T-1} R(s(t), \vec{a}(t))|\pi\right] , \tag{2.9}$$

where T is the horizon, $s(t)$ is the state and $\vec{a}(t)$ agents' actions at time step t, and π denotes the agents' policies. In some other DEC-POMDP problems, such as in wireless networks with multiple agents, the execution of policies of agents can continue over an infinite time period in practice. In discounted infinite-horizon DEC-POMDPs, the optimization goal is

$$E\left[\sum_{t=0}^{\infty} \gamma^t R(s(t), \vec{a}(t))|\pi\right] , \tag{2.10}$$

where $0 < \gamma < 1$ is the discount factor.

Other models than the DEC-POMDP model have been proposed for cooperative multiagent planning under uncertainty. In interactive POMDPs (I-POMDPs) [55, 42], the agent's belief includes probabilities for the current world state, other agent capabilities, other agents' beliefs about beliefs of agents, etc. An I-POMDP captures the idea that an agent needs to reason about the beliefs of other agents recursively, because other agents consider the agent's belief. An I-POMDP allows for modeling also adversial situations. The Communicative Multiagent Team Decision Problem (COM-MTDP) [119, 120] framework and the DEC-POMDP-COM framework [56] are similar to a DEC-POMDP, but they include explicit communication and may be useful when communication is analyzed. Online DEC-POMDP methods [166] also utilize communication. When agents communicate their observations, they reduce uncertainty about the current world state and about the policies of other agents, and the agents need to consider only a small part of the policy and belief space while planning future actions. When agents are allowed to communicate fully at every time step, a DEC-POMDP is reduced to a centralized multiagent POMDP [120], which has been exploited for computationally lighter multiagent planning [95].

In a POMDP, the value function is piecewise linear and convex and the agent can maintain a belief, which is a sufficient statistic, over the world states. In a POMDP, the policy can be efficiently represented in vector form in both finite-horizon and discounted infinite-horizon POMDP problems. However, in a DEC-POMDP an agent does not know what other agents have observed, and in order to act optimally, it must reason about the possible action-observations histories of other agents. Because the policy cannot be kept in vector form, many state-of-the-art finite-horizon DEC-POMDP methods [148, 147, 104, 165, 78, 144] use policy trees or graphs as policy representation. In infinite-horizon DEC-POMDPs, a policy tree or graph would require infinite space, and thus state-of-the-art infinite-horizon methods ([9, 12, 11, 79] and Publications II and III) use finite state controllers as policy (see Figure 2.3 for policy tree, policy graph, and finite state controller examples).

2.7 Finite-horizon DEC-POMDP

The computational challenges in DEC-POMDPs are illustrated nicely by optimal finite-horizon DEC-POMDP methods [60, 148, 104, 144]. In finite-horizon POMDPs, the optimal policy can be represented as a policy tree (see Figure 2.3 for a policy tree example). Similarly, in DEC-POMDPs the optimal policy can be represented as a set of policy trees, one policy tree for each of the N agents. A simple optimal finite-horizon DEC-POMDP algorithm constructs a search tree, where each node is a set of policy trees. The first level of the search tree contains $|\mathcal{A}_i|$ nodes for one agent and $|\mathcal{A}_i|^N$ nodes for N agents (without loss of generality assume that all N agents have the same number of actions $|\mathcal{A}_i|$ and observations $|\mathcal{O}_i|$), that is, a set of policy trees for each action combination. When considering the second level in addition to the first level, there are in total $(|\mathcal{A}_i|^{1+|\mathcal{O}_i|})^N$ policy trees, because all actions have to be considered for each observation. For horizon T, there are in general $(|\mathcal{A}_i|^{\frac{|\mathcal{O}_i|^T-1}{|\mathcal{O}_i|-1}})^N$ possible policy trees (see Section 3.1 in [132] for details). The search tree is doubly exponential in the horizon and exponential in the number of agents. However, state-of-the-art optimal methods [144] can solve some two agent DEC-POMDP problems to large horizons. Methods based on A* search [148, 104, 144] use an upper bound for pruning the search tree and for choosing which search nodes to expand. Techniques such as history clustering [104] make policy search more efficient. Another interesting optimal method is the

dynamic programming approach for partially observable stochastic games [60], which is one of the first algorithms for solving DEC-POMDPs optimally.

2.7.1 Bounded width policy graph methods

Because DEC-POMDPs are computationally so demanding, a large part of recent research work [44, 131, 132, 79, 165] propose approximate methods. Bounded policy graph approaches scale to large horizons and have received significant interest lately [131, 132, 79, 165]. Bounded policy graph methods use a policy graph with bounded width (see Figure 2.3 for a policy graph example) as policy. Bounded width policy graph methods can scale linearly with respect to the horizon. The main algorithmic idea behind the bounded policy graph methods is discussed next.

The basic idea in a bounded policy graph method is to build the policy graph starting from the last layers of nodes in horizon T and to go backwards towards the first layers of nodes in time step zero. The policy graphs of all agents have the same size and structure. A point based bounded policy graph method updates the policies for the nodes in all agents' graphs, which are in the same layer and in the same position within the layer, at the same time. In the update, the method samples a centralized belief, assumes that the agents are in the "same" nodes, and computes a policy for the nodes using the value function of the next layers. The value function is maintained by backing it up, when the policy for a layer has been computed. In more detail, at the last time step T a value function $V_T(s, \vec{q})$, over the world state s and the policy graph nodes $\vec{q}$, is initialized to maximize the immediate reward $R(s, \vec{a})$. When the policy for policy graph layer t has been determined, $V_t(s, \vec{q})$ is computed using the new policy, the immediate reward, and the next layer value function $V_{t+1}(s, \vec{q})$. Point based methods commonly determine the policy for a node by sampling a shared centralized belief for all agents. Denote with t the layer of the policy graph and with (t, i) the ith node in layer t. A centralized belief $b(s)$ is sampled for layer t, and then it is assumed that all agents are in the same ith policy graph node: $P_t(s, q_1 = i, q_2 = i, \ldots, q_N = i) = b(s)$. The goal is then to find for all agents a policy for the (t, i) node, which maximizes the expected sum of the immediate reward and of $V_{t+1}(s, \vec{q})$ for $P_t(s, q_1 = i, q_2 = i, \ldots, q_N = i)$.

2.8 Infinite-horizon DEC-POMDP

Finite-horizon DEC-POMDP methods use policy trees or policy graphs as policy representation. In infinite-horizon DEC-POMDP problems, the policy of each agent must control the agent over an infinite length of time. Using a policy graph or tree in an infinite-horizon DEC-POMDP would either require that the policy has infinite length (which is not possible in real methods) or that actions could be chosen using the current belief state (possible in POMDPs, but not in DEC-POMDPs). Instead of using policy graphs or trees state-of-the-art infinite-horizon DEC-POMDP methods (see [146, 19, 9, 20, 12, 11, 79] and Publications II and III) use finite state controllers as policy. Figure 2.3b shows an example finite state controller and Figure 2.5 shows an influence diagram of a two agent DEC-POMDP controlled by finite state controllers. In many problems, finite state controllers allow compact representations of very good policies. Furthermore, recent results indicate that controllers can be computed for very large problems with a good performance (see [80], Publication II, and Publication V) and that large controllers with high performance can be optimized (see Publication III).

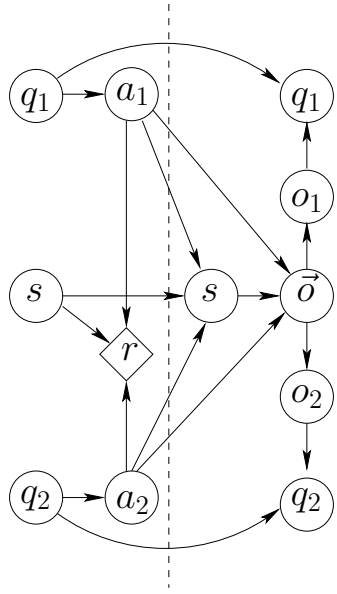


Figure 2.5. Influence diagram for a two agent DEC-POMDP. The agents are controlled by finite state controllers (FSCs). The action of each agent (a_1 and a_2) depends on the current FSC state (q_1 and q_2). The next FSC state depends on the current FSC state and on the observation of the agent (o_1 and o_2).

Different techniques have been used to optimize finite state controllers

for DEC-POMDPs. The method in [19] initializes the stochastic FSC of each agent and then iteratively improves each FSC by linear programming while keeping other FSCs fixed. Because only one FSC is improved at a time, the approach may not converge to the global optimum. The best-first search method in [146] finds deterministic FSCs with a fixed size. The dynamic programming policy iteration approach in [20] converges in the limit to an optimal solution, but for practical results a heuristic policy iteration approach is presented. Bernstein et al. [20] also discuss how a correlation device can be used to improve synchronization between agents without communication. If a DEC-POMDP problem has a specific goal state, then a goal-directed [10] approach can be used.

In the *non-linear programming* approach [9, 12, 11], the optimization problem is formulated as a non-linear program, where the program tries to find finite state controller action and transition probabilities that maximize the value function for the initial belief. The non-linear programming approach tries to optimize the policies of all agents at the same time. Similar to the Bellman equation in POMDPs (see Equation 2.5), the value function is defined recursively to be the sum of the immediate reward and the backed up next time step value function. Another approach for optimizing stochastic FSCs, expectation maximization, is discussed next.

2.8.1 Expectation maximization

In the expectation maximization (EM) method for (PO)MDPs [152] and DEC-POMDPs [79], the optimization is written as an inference problem and EM is used to improve stochastic FSCs in each iteration. Because a POMDP is a single agent DEC-POMDP, EM will be presented for DEC-POMDPs only. In general, EM [39] tries to iteratively maximize the likelihood of a probabilistic model over latent parameters. One iteration of EM consists of one E-step and one M-step. In the E-step, EM computes the expected log-likelihood of the probabilistic model for the current parameter values (where the expectation is taken over the latent variables), and in the M-step EM finds parameters that maximize the expected log-likelihood found in the E-step. In the EM algorithm for DEC-POMDPs, the DEC-POMDP problem is transformed into a probabilistic modeling form where the likelihood of the resulting probabilistic model is directly proportional to the expected discounted reward of the DEC-POMDP. EM is then used to find latent FSC parameters that maximize the likelihood. More details of the approach follow.

In the EM approach, the reward function is scaled into a probability distribution for a stochastic binary reward variable r, so that

$$\hat{R}(r = 1|s, \vec{a}) = (R(s, \vec{a}) - R_{min})/(R_{max} - R_{min}) \ , \tag{2.11}$$

where R_{min} and R_{max} are the minimum and maximum rewards possible and $\hat{R}(r = 1|s, \vec{a})$ is the conditional probability for the binary reward r to be 1.

Denote with $L = (Q_1, \ldots, Q_N, O_1, \ldots, O_N, A_1, \ldots, A_N, S)$ sequences of latent FSC variables $Q_1, \ldots, Q_N$, observation variables $O_1, \ldots, O_N$, action variables $A_1, \ldots, A_N$, and state variables S. EM maximizes the following complete log likelihood in the M-step:

$$Q(\theta, \theta_{new}) = \sum_{T=0}^{\infty} \sum_{L} P(r = 1, L, T|\theta) \log P(r = 1, L, T|\theta_{new}) \ , \tag{2.12}$$

where θ and θ_{new} are current and new FSC parameters. The maximization is done with respect to the new parameters θ_{new}. When the initial belief is projected T time steps forward in time using the latent variables L and the given FSC parameters θ, the probability for the binarized immediate reward to happen at time step T is $P(r = 1, L, T|\theta)$:

$$\begin{aligned} P(r = 1, L, T|\theta) \quad = \quad & P(T)\hat{R}(r_T = 1|s_T, \vec{a}_T) \prod_i P(a_{(i,T)}|q_{(i,T)}) \prod_{t=1}^{T} P(\vec{o}_t|s_t, \vec{a}_{t-1}) \\ & P(s_t|s_{t-1}, \vec{a}_{t-1}) \prod_i P(a_{(i,t-1)}|q_{(i,t-1)}) \\ & \prod_i P(q_{(i,t)}|q_{(i,t-1)}o_{(i,t)}) P(s_0, \vec{q}_0) \ , \end{aligned} \tag{2.13}$$

where $P(T)$ is the prior probability for time step T. Subscripts denote either the time step (x_t for time step t) or the agent and time step ($x_{(i,t)}$ for agent i at time step t). In order to make the likelihood proportional to the discounted reward, $P(T) = \gamma^T(1 - \gamma)$.

In the E-step, the EM method computes discounted sums of beliefs projected forward in time and discounted sums of reward probabilities projected backwards in time. In the M-step, the EM method uses these discounted sums to find FSC parameters to maximize the log-likehood in Equation 2.12. Because the $\log$ in $Q(\theta, \theta_{new})$ makes a sum out of the product of the new FSC parameters θ_{new} and DEC-POMDP probabilities, FSC parameters can be maximized separately in the M-step (see Section 4.2 in [79] for details). Therefore, the FSC parameters of each agent can be updated in parallel in the M-step, making efficient parallel algorithmic implementations possible.

2.9 Factored DEC-POMDP

Similar to factored POMDPs, factored DEC-POMDPs ([102] and Publication II) provide a model for making optimal decisions in settings where the world state, observations, and agents, can be described using several variables. All DEC-POMDP problems are to some degree inherently factored: because agents are decentralized, that is, make decisions independently in a DEC-POMDP, the policies of agents do not "directly" influence each other (an agent influences the state of the world, which influences other agents "indirectly", but agents do not decide together at each time step which actions to perform. See the DEC-POMDP influence diagram in Figure 2.1.).

The wireless network problem in Publication V with multiple wireless agents provides an example of a general factored DEC-POMDP. In wireless networks, the interference from a wireless agent to another agent depends on the distance between the agents. When an agent tries to transmit a packet, only wireless agents that are close enough can cause a packet drop. Therefore, in the factored DEC-POMDP model, the next state of an agent's packet buffer depends only on the agent's action and on the actions of those agents that can cause interference, but not on agents that cannot cause interference. A factored DEC-POMDP describes the wireless network problem compactly.

Formally, in a factored DEC-POMDP the state consists of several state variables $s_1, s_2, \ldots, s_M$ and the observation o_i, of agent i, of several observation variables $o_i = o_{i,1} \times o_{i,2} \times \cdots \times o_{i,N}$. The probability for state variable s_i to transition to s'_i is $P(s'_i|\text{Parents}(s_i))$, where $\text{Parents}(s_i) \subset (a_1, a_2, \ldots, a_N, s_1, s_2, \ldots, s_M)$ denotes the parent variables. The probability for agent i to observe o_i is $P(o_i|\text{Parents}(o_i)) = \prod_j P(o_{i,j}|\text{Parents}(o_{i,j}))$, where $\text{Parents}(o_{i,j}) \subset (a_1, a_2, \ldots, a_N, s'_1, s'_2, \ldots, s'_M)$. The reward function $R(s, \vec{a})$ is a sum of reward sub-functions $R(s, \vec{a}) = \sum_i R_i(S_i)$, where $S_i \subset (a_1, a_2, \ldots, a_N, s_1, s_2, \ldots, s_M)$. Figure 2.6 shows the influence diagram of an example factored DEC-POMDP.

2.9.1 Special cases of factored DEC-POMDPs

The computational complexity of general factored DEC-POMDPs is the same as that of non-factored DEC-POMDPs [8]. This is perhaps not surprising, because the influence of variables spreads over several time steps to all other variables, when variable dependencies are not restricted. Be-

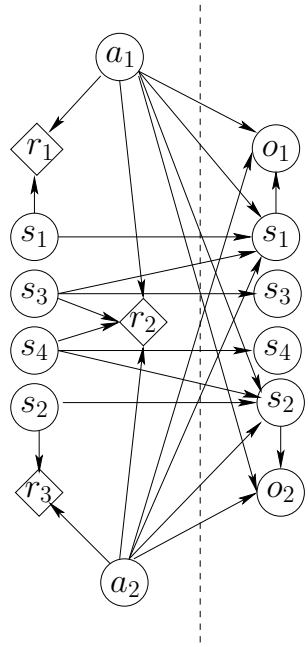


Figure 2.6. Influence diagram of an example factored DEC-POMDP with two agents. The dotted line separates two time slices. a_1 and a_2 denote actions of agent one and two, r_1, r_2 and r_3 rewards, $s_1, \ldots, s_4$ state variables, and o_1 and o_2 observation variables of agent one and two.

cause of the computational complexity, special cases of factored DEC-POMDPs have been investigated in the literature.

In transition-independent DEC-MDPs [15], it is assumed that an agent's local state evolves independently of other agents. In Network-Distributed POMDPs (ND-POMDPs) [100], both the agent's local observation and the agent's state do not depend on other agents' actions or states. In a ND-POMDP, the reward is a sum of sub-rewards and each sub-reward depends on a limited subset of all agents' actions and states. Because of the restricted structure, the influence of variables does not spread to other state variables over time and efficient algorithms are easier to construct (transition-independent DEC-MDPs are NP-complete). Another special case of a factored DEC-POMDP is the Transition-Decoupled POMDP (TD-POMDP) [162], which has local state and observation variables for each agent. In a TD-POMDP, the joint reward is composed of individual rewards and the local state variable of an agent can be directly influenced by one other agent, which allows influence to spread over time, while still making more efficient algorithms possible [163].

Most work on special cases of factored DEC-POMDPs concentrates on

finite-horizon problems. Kumar et al. [80] present an expectation maximization approach for both finite-horizon and infinite-horizon problems. The approach optimizes finite state controllers for models, such as transition independent DEC-MDPs and ND-POMDPs, in which the value function can be decomposed into factors (see [80] for more details).

2.9.2 General factored DEC-POMDPs

The models described above offer solutions to problems such as sensor networks [100] or autonomous exploration [162]. However, in many real-world problems a general factored DEC-POMDP approach is required. Examples of such problems are the factored fire-fighting problem in [103] and the wireless network problems in [102], Publication II, and Publication V. General *factored finite-horizon DEC-POMDPs* are investigated in [103] and approximations for scaling to large number of agents are presented in [102]. Publication II provides an expectation maximization based approach for optimizing finite state controllers in *factored infinite-horizon DEC-POMDPs*. In the experiments in Publication II, the approach has similar performance as non-factored methods in problems with few agents and scales to large problems with many agents for which non-factored methods fail. The approach is discussed in more detail in Section 3.2.

3. New methods: Efficient planning for POMDPs and DEC-POMDPs

Many real-world problems require more scalable and better performing POMDP and DEC-POMDP methods. This chapter motivates and presents the factored POMDP method in Publication I, the method for factored infinite-horizon DEC-POMDPs in Publication II, and periodic finite state controllers for POMDPs and DEC-POMDPs in Publication III.

Exact optimal planning [33] algorithms exist for POMDPs, but such algorithms can handle only problems with few states. For larger problems approximate algorithms [115, 113, 86, 136, 87] are needed. DEC-POMDP algorithms [103, 144] can find optimal solutions to some problems with a limited planning horizon in reasonable time (for complex problems the maximum feasible horizon is very short). A flat format definition of a very large POMDP or DEC-POMDP problem, such as the wireless networking problems described in Publication IV or in Publication V, often exceeds practical memory limits. Luckily most real-world problems are factored in some way and a factored description is possible. However, finding the optimal policy for a factored problem is in general not simpler than for a non-factored problem. Factored finite-horizon DEC-POMDPs have the same computational complexity of NEXP-complete than non-factored DEC-POMDPs [8]. In order to scale to larger problems, new kinds of solution methods are needed.

The new factored POMDP method in Publication I finds approximate solutions to larger POMDP problems than comparison methods and gives good results on smaller problems. Publication I is motivated by the opportunistic spectrum access problem in Publication IV, which will be discussed in detail in Chapter 4. Publication II discusses the first factored infinite-horizon DEC-POMDP method. The method performs as well as non-factored methods on smaller problems and finds approximate solutions to much larger problems than comparison methods. The method is

used in Publication V for optimizing wireless channel access policies (see Chapter 4 for a discussion on Publication V).

There are at least two problems, that make policy optimization computationally intractable: 1) large problem size and 2) large policy size. Many factored methods ([57, 116], Publications I and II) focus on the problem size. Complex large problems most often require complex large policies for high performance. Because of the computational complexity of DEC-POMDP problems, state-of-the-art infinite-horizon DEC-POMDP methods can optimize larger policies only in small problems. Publication III presents periodic finite state controllers (FSCs) that allow large policies and new kinds of optimization algorithms. Periodic FSCs yield state-of-the-art results for both POMDPs and DEC-POMDPs. The practical relevance of the developed methods is demonstrated in Publication V. Publication V uses periodic policies together with the factored infinite-horizon DEC-POMDP method for optimizing wireless controllers in a real-world problem with many agents.

3.1 Efficient planning for factored POMDPs

Factored POMDPs have been used on benchmark problems such as the computer network problem [116], but also on important real-world problems such as assisting persons with dementia during handwashing [63] or optimizing spectrum access in a wireless network (Publication IV). This chapter discusses the factored POMDP method factorized belief value projection (FBVP) described in Publication I. The practical motivation for the new approach comes from Publication IV which investigates opportunistic spectrum access in a wireless network. A secondary user, the wireless agent, has to choose a channel to transmit on, while balancing the tasks of collecting information about channel states, avoiding collisions with primary users, and transmitting as much data as possible (see Section 4.1.2 for a more detailed discussion on the wireless networking problem). Because the state space of the POMDP grows exponentially with the number of wireless channels, the problem requires a scalable POMDP approach.

As discussed in Section 2.5 the problem in factored POMDPs is how to keep the belief and the policy in compact form. FBVP avoids problems with the exponentially sized state space by maintaining the policy as a policy graph only (see Figure 2.3 for a policy graph example). More information on why this helps is given below. The belief is kept in an approxi-

mate fully factored form, that is, as a product of individual state variable distributions. Although the state space has exponential size with respect to the number of state variables N, the FBVP algorithm requires only polynomial time with respect to N.

3.1.1 Factorized belief value projection (FBVP)

The main algorithm of FBVP is similar to Perseus [143] in that FBVP first samples a set of beliefs and then improves belief values incrementally using the Bellman equation. FBVP starts from an empty policy graph and adds a new policy graph layer, which improves the value of each belief, in each iteration. The policy graph consists of nodes, each associated with a certain action and connected for each observation to a next layer policy graph node (see Figure 2.3 for a policy graph example).

Input: Initial policy G_0, beliefs B, factored POMDP specification
Output: Policy graph G
Initialize policy graph layer counter $n = 0$
repeat
    Initialize current belief set $\tilde{B}$ to B
    Initialize new policy graph layer G_{n+1} to $\emptyset$
    repeat
        Remove random belief b from $\tilde{B}$
        Backup belief b using G_n yielding new graph node α that could become part of G_{n+1}
        if α *does not exist yet in* G_{n+1} *(there is no node with the same action and observation connections)* **then**
            Add α to G_{n+1}
            Remove beliefs from $\tilde{B}$ for which α increased value
    until $\tilde{B}$ *is empty*
    $n = n + 1$
until *convergence*

Algorithm 1: Pseudocode for the main loop of factorized belief value projection (FBVP). FBVP adds iteratively new policy graph layers at the beginning of the policy graph. In order to add a new layer, FBVP randomly selects beliefs from a fixed belief set B and backs up each belief for a new node in the layer. G_n denotes the nth policy graph layer of the complete policy graph G.

Algorithm 1 shows an outline of the main loop in FBVP. The algorithm invokes two sub-routines: *backup* a belief and *evaluate* a belief. Because backup uses evaluate, evaluate is described first.

Evaluate a belief. FBVP uses the policy graph to compute the value of a belief, that is, the expected discounted reward when following the policy graph starting from the belief. FBVP computes the belief value for each policy graph root node separately and selects the root node with the highest value. The value for a root node is computed by projecting the belief through the policy graph (discussed below) and summing the discounted immediate rewards at the policy graph nodes to get the expected reward.

FBVP projects the belief forward one policy graph layer at a time, starting from the initial policy graph node. The end result of the projection is the probability for visiting each node, and the belief over world states at the node, when visiting it. During projection FBVP conditions the belief at each node on the node's action, and on each observation using Equation 2.4 (without normalization over observations). FBVP computes the visiting probability for an outgoing observation edge by multiplying the observation probability with the visiting probability of the current node. At the next policy graph layer FBVP computes the belief for a node by summing all beliefs, that arrive at the node through incoming observation edges, and normalizes it. The visiting probability for the next layer node is computed as the sum of incoming observation edge probabilities.

In order to make computations tractable, FBVP keeps beliefs in a fully factored form, which corresponds to a product of probability distributions $b(s) = \prod_i b(s_i)$ of individual state variables s_i. When a belief is projected through the policy graph, the fully factored form is maintained by employing approximations that minimize the Kullback-Leibler divergence to the current non-factored form. Approximations are done: 1) when a belief is conditioned on an action, 2) when a belief is conditioned on an observation, and 3) for converting a sum of several beliefs into a single belief. As discussed above beliefs are summed when a node has several incoming observation edges. Section 3.1 in Publication I discusses the formulas for these belief approximations.

Backup a belief. In FBVP, the *backup* operation constructs, for a certain belief, a new policy graph node at the beginning of the policy graph. FBVP chooses an action and for each observation chooses an edge to the next layer node so that the choices yield highest value for the belief. In more detail, the approach projects a belief for each action-observation pair us-

ing Equation 2.4, finds for the projected belief the next layer node that gives the highest value using the *evaluate* procedure described above, and then sums together the immediate reward and the value following each observation to get a value for the action. The right hand side of the Bellman-equation in Equation 2.5 shows how the value of an action is computed. The backup operation in FBVP corresponds to the elementary backup operation in other point based POMDP algorithms [112, 143, 140], but usually other POMDP algorithms backup previous α-vectors in order to construct a new α-vector, instead of backing up graph nodes.

3.1.2 Pruning

The evaluation of a belief is a computationally heavy operation, because FBVP projects the belief through the whole policy graph. To speed up computations FBVP computes for each policy graph node an approximate upper value bound, that is a sum of linear functions of single state variables. The bounds can be computed by either applying quadratic programming or least squares on computed belief values. In the experiments quadratic programming was used. The bounds are used in the *backup* operation during belief evaluation: if the upper bound for a belief's value is below the best value found so far, evaluation for the belief can be stopped. Pruning can be made even more efficient: process most likely observations first and use the upper bounds to determine if it is impossible to improve the current best solution using the remaining observations. The observation based pruning works well in problems where observation probabilities are unevenly distributed.

3.1.3 Implementation

In an efficient implementation of FBVP, computations are done in parallel for multiple beliefs at a time. Also, in FBVP it can happen that a new node in the new graph layer $n + 1$ does not increase the value for a belief compared to the highest value node in the previous policy graph layer n. In order to prevent a value decrement, FBVP inserts a virtual dummy node into layer $n + 1$ that redirects the belief directly into the highest value node in layer n during belief projection.

3.1.4 Results

In the experiments, shown in Table 1 in Publication I, FBVP was compared to four POMDP methods in four different benchmark problems. Background information about the used comparison methods can be found in Section 2.4.2 for the non-factored Perseus [143] and HSVI2 [140] methods and in Section 2.5 for the truncated Krylov iteration [113]) and Symbolic Perseus [113]) methods. In the experiments, FBVP found approximate solutions to larger problems than comparison methods, with good performance, and yielded adequate performance in smaller problems. Noteworthy is the much higher performance of FBVP, compared to the comparison methods, in the large scale real-world spectrum access problem.

3.2 Factored infinite-horizon DEC-POMDPs

Publication II appears to be the first publication on general factored infinite-horizon DEC-POMDPs (see Section 2.9 for a discussion about factored DEC-POMDPs). The new factored infinite-horizon DEC-POMDP method in Publication II allows the use of different efficient approximations that can scale polynomially with respect to the number of agents and state variables. The policies of agents are stochastic finite state controllers that are improved using a specially modified version of the expectation-maximization (EM) method for non-factored DEC-POMDPs [79]. In comparison to state-of-the-art algorithms, the method performs well and finds approximate solutions to much larger infinite-horizon DEC-POMDP problems than the comparison methods.

Wireless controller optimization, described in detail in Section 4.2 and Publication V, serves as a real-world application for the method. Publication V models a wireless network as a factored infinite-horizon DEC-POMDP. Because interference decreases quadratically or faster with distance, a device receives significant interference only from a small subset of devices in a wireless network. Therefore, the next state of a device depends only on a small subset of other devices and the problem is naturally modeled as a factored DEC-POMDP. Because wireless controllers can be used in practice for an infinitely long time, the problem needs to be treated as an infinite-horizon problem.

3.2.1 Expectation maximization for factored DEC-POMDPs

This thesis discusses now how to apply EM to large factored DEC-POMDPs, what problems need to be solved, and how the new factored infinite-horizon approach addresses these problems.

In the expectation maximization (EM) methods for POMDPs [152] and DEC-POMDPs [79], the optimization is written as an inference problem and EM is used to improve stochastic finite state controllors (FSCs) in each iteration (see Section 2.8.1 for more details). The reward function is scaled into a probability of a binary reward and EM tries to maximize the expected reward likelihood. The EM algorithm performs expectation and maximization steps (E- and M-steps) iteratively. In the E-step, EM computes, using the current policy, α messages by projecting probabilities forward in time and β messages by projecting reward probabilities backward in time. In the M-step, EM uses the α and β messages computed in the E-step for finding new policy parameters that improve expected reward.

In practice, EM cannot be applied to large factored DEC-POMDPs, because the computation time of one EM iteration scales exponentially with the number of agents and exponentially with the number of state variables. The main idea in Publication II is to keep probabilities always factored, that is, in a form that has polynomial size with respect to the number of agents and state variables (see Section 3.5 in Publication II for more details on the properties of the approach). In Publication II, the EM algorithm is transformed into a form in which probabilities are projected only forward, which eliminates the need to project rewards, which are sums of reward sub-functions, backwards in time. In brief, the method substitutes backward projection of reward probabilities with forward projection of probabilities and computation of immediate reward probabilities. Below, subsection 3.2.2 discusses how probabilities are kept factored and subsection 3.2.3 discusses how rewards are kept factored.

3.2.2 Keeping probabilities factored

In a general factored DEC-POMDP, probabilities become non-factored during forward and backward projection. Consider for example the two agent dynamic Bayesian networks in Figure 3.1. Assume that all variables are independent at the beginning and let's consider state variable s_4 of the set of all state and FSC variables $(s_1, s_2, s_3, s_4, q_1, q_2)$. s_4 in the first time

step directly influences s_2 and s_4 in the second time step. Indirectly it influences also q_2 in the second time step. The influence spreads in each time step. In this wireless network problem, the number of agents affects how many time steps it takes for all variables to depend on each other, because an agent influences only neighboring agents inside one time step, but in the end all variables depend on each other. This spreading of influence intuitively explains why factored finite-horizon DEC-POMDPs have the same computational complexity as non-factored finite-horizon DEC-POMDPs [8]: even if only part of the variables depend on each other in one time step, the influence will spread over all variables over multiple time steps. The same principle applies to reward sub-functions. The paper [103] contains a nice illustration on how influence spreads in a factored finite-horizon DEC-POMDP, when reward sub-functions are projected backwards in time. The same kind of influence spreading occurs with reward probabilities, when projected backwards in time. The spreading of variable influence in dynamic Bayesian networks in general is discussed in [98].

Intuitively, the method in Publication II projects probabilities one time step forward and then breaks dependencies between variables in different variable clusters. Publication II explains in more detail how to break the dependencies with two different variable clusterings and how the approximation error caused by the forced variable clustering behaves. The first approach, fully factored clustering, keeps all variables in separate clusters. The second approach, overlapping clustering, clusters variables using one time step DEC-POMDP variable dependencies. Figure 3.1 shows examples of both clusterings in a wireless network problem.

3.2.3 Keeping rewards factored

There are two problems with keeping rewards factored. Firstly, when a factored reward function is projected in time, the scope of the reward sub-functions grows (Oliehoek et al. [103] illustrate this for finite-horizon DEC-POMDPs). In order to keep rewards factored, the method in Publication II does not project rewards, but instead projects factored probability distributions forward in time and computes only immediate reward probabilities. This approach circumvents problems with keeping the reward function in a computationally efficient form during projection. Secondly, the standard EM algorithm for POMDPs and DEC-POMDPs scales rewards into probabilities using the minimum and maximum of the reward

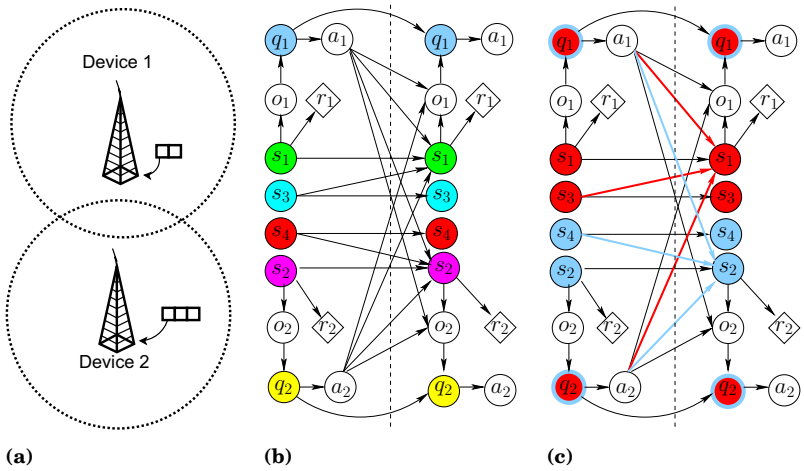


Figure 3.1. Examples of the two different variable clusterings in the wireless network problem shown in subfigure (a). Two wireless agents try to transmit packets from their buffers s_1 and s_2, which are filled by traffic sources s_3 and s_4. q_1 and q_2 are FSC states of agent one and two, a_1, a_2 are actions, o_1, o_2 are observations, and r_1, r_2 are rewards. Dotted lines separate time steps. In the dependency diagrams (b) and (c), nodes with the same color are in the same cluster. In the fully factored clustering in subfigure (b), all variables are in different clusters. In the overlapping variable clustering in subfigure (c), state variables s_1 and s_3 are in the same cluster, state variables s_2 and s_4 are in another cluster, and the FSC variables q_1 and q_2 are in both clusters. The colored arrows explain the reasoning behind overlapping clustering: s_1 depends directly on s_3, s_2 depends directly on s_4, and both s_1 and s_2 depend through an action on both q_1 and q_2.

function. In a factored DEC-POMDP the reward function is a sum of reward sub-functions. While the scopes of reward sub-functions are small, the scope of the non-factored reward function can be huge and it can be prohibitively expensive to compute the minimum and maximum of the reward function. Instead, as described in Section 3.4 in Publication II, the new approach uses the sums of reward sub-function minimums and maximums as lower and upper bounds for the minimum and maximum rewards, respectively. The approach then scales the reward function with these lower and upper bounds.

3.2.4 Results

Publication II compared the proposed factored infinite-horizon DEC-POMDP method, experimentally, to a random baseline method and to the non-factored non-linear programming [9] and EM [79] methods. The experiments included two factored DEC-POMDP benchmark problems: a robotic fire fighting problem and a wireless networking problem (see Publication

II for details). Problem size was increased from two to ten agents. In the fire fighting problem the proposed factored infinite-horizon method had equal or better performance than comparison methods in problems with four or less agents. In problems with five or more agents, the non-factored methods ran out of memory, but the proposed method produced better results than the baseline. In the wireless networking problem, the proposed method had equal or better performance than the comparison methods with only two agents. With three or more agents the non-factored methods ran out of memory, but the proposed method produced, again, good results compared to the baseline.

3.3 Periodic finite state controllers for (DEC)-POMDPs

While the factored infinite-horizon method discussed in the previous section focuses on the problem of finding solutions to large DEC-POMDP problems, Publication III focuses on optimizing large policies (which are often needed in large problems). State-of-the-art infinite-horizon DEC-POMDP methods ([9, 79, 11] and Publication II) store agent policies as finite state controllers (FSCs). One problem with these methods is that DEC-POMDP problems are computationally very demanding and policy size is thus restricted. The novel approach in Publication III for both POMDPs and DEC-POMDPs uses *periodic* FSCs, which are composed of layers connected only to the next layer and the last layer connected to the first layer. It enables the design of new kinds of algorithms and computation of much larger controllers.

In more detail, periodic FSCs allow construction of new algorithms that improve only a single policy layer at a time while keeping other layers fixed. In contrast, a regular aperiodic FSC can be in any state at any given time. Therefore, improving only a subset of the states of an aperiodic FSC is not possible in the same way as in periodic FSCs. Furthermore, in general, in a DEC-POMDP, agents do not communicate or jointly decide on actions. Therefore, synchronizing agent behavior is difficult. However, in a periodic FSC, only a subset of the FSC states is active at each time step; it is computationally easier to find policies that work together for smaller sets of FSC states than for complete FSCs.

The periodic methods in Publication III are general and applicable to any POMDP or DEC-POMDP problems. The real-world wireless network problem in Publication V serves as a practical application. In Publica-

tion V, the factored infinite-horizon DEC-POMDP approach, discussed in the previous section, optimizes stochastic finite state controllers for wireless agents. Because the dependencies among the agents are complicated, policies of adequate complexity are required to achieve high value. The periodic expectation maximization (EM) approach, discussed later in this section, scales linearly with the number of FSC layers. Although periodic EM is based on the EM formulation for non-factored problems, it is straightforward to use it together with the factored infinite-horizon DEC-POMDP approach. The two approaches are used together in Publication V to optimize large policies for large wireless network problems. Using periodic EM together with the factored infinite-horizon DEC-POMDP approach will be discussed in more detail in Section 3.3.4.

Publication III presents first a new method for improving deterministic finite-horizon finite state controllers monotonically (a finite-horizon finite state controller is a policy graph). Next a method for transforming the finite-horizon policy to an infinite-horizon periodic FSC is described. Finally, the finite-horizon improvement method is adapted to the infinite-horizon case to optimize periodic FSCs. In addition to the deterministic FSC methods, formulas for a periodic expectation maximization method for optimizing stochastic FSCs are derived. The periodic EM method can be used to optimize policies starting from deterministic FSCs with noise added, which is done in Publication III, or to optimize policies starting from random stochastic FSCs as is done in Publication V.

The text proceeds next to discuss what a periodic FSC actually is. Then it presents the new optimization method for finite-horizon DEC-POMDPs and shows how the optimization method can be adapted to optimize deterministic periodic FSCs. Lastly, a periodic expectation maximization approach for stochastic FSCs is discussed.

3.3.1 Periodic finite state controller

A periodic FSC is composed of layers of nodes and policy execution cycles through each layer. In layer m, the agent takes action a_i when in node q_i with probability $P^{(m)}(a_i|q_i)$ and the controller moves from node q_i to a node q_i' in the next layer with probability $P^{(m)}(q_i'|q_i, o_i)$, when observing o_i. Figure 3.2 shows a periodic FSC. Note that the size of the probability distribution over nodes depends only on the width of the FSC, that is, the number of nodes in a layer, not on the total number of nodes. The periodic FSC structure makes it possible to create efficient algorithms

that improve one layer at a time. Intuitively a regular FSC is the special single-layer case of a periodic FSC or alternatively a periodic FSC is a regular FSC with some transition probabilities set to zero.

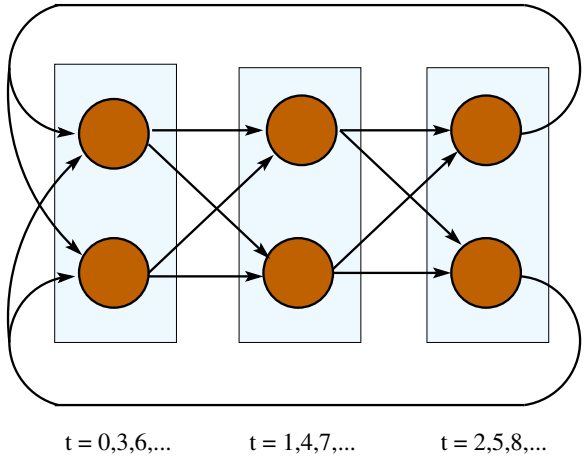


Figure 3.2. A periodic finite state controller composed of layers of nodes. The policy cycles through each layer of nodes periodically. The agent executes an action according to the action probabilities of the current node, makes an observation and transitions to a node in the next layer according to the transition probability for the current node and the observation made. From the last layer the controller transitions to the first layer. As a result, for this example 3-layer controller, at time steps $0, 3, 6, \ldots$ the controller will be in layer 1, at time steps $1, 4, 7, \ldots$ in layer 2, and at time steps $2, 5, 8, \ldots$ in layer 3.

3.3.2 Monotonic policy graph value improvement

The method in Publication III first initializes the policy graphs and then applies the procedure illustrated in Figure 3.3 to improve the policy graph value monotonically. The initialization procedure starts from random policy graphs. It computes a policy (an action and next layer node for each observation) for each node proceeding from last layer to the first layer. When in layer t, it samples beliefs $b_t(s)$ for each policy graph node assuming that all agents are in the same node (the policy graphs of all agents are of the same size and same node means here the ith node in all policy graphs) and improves the policy of the agents' nodes. Using $b_t(s)$ and the next time step value function $V_{t+1}(s, \vec{q})$ the method optimizes the policy of each agent's node while holding the other agents policies fixed until no improvement is possible. The initialization procedure is similar to the PBPG method in [164], but instead of linear programming direct search is used with the same end result.

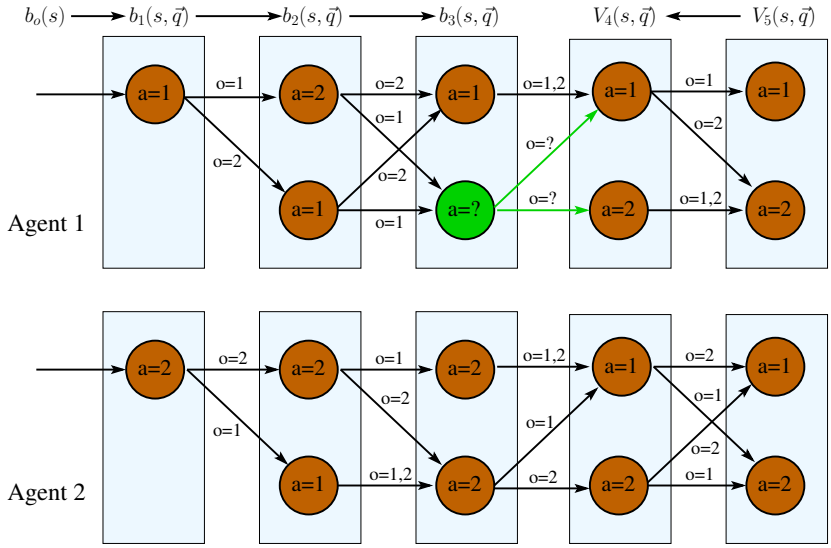


Figure 3.3. Illustration of the monotonic policy graph value improvement method for two agents. Each graph node is labeled with an action and outbound edges denote the next node for each observation. Belief $b_t(s,\vec{q})$ is the probability distribution and $V_t(s,\vec{q})$ is the value function over world state s and graph nodes $\vec{q}$ at graph layer t. The method projects the belief from left to right and starts optimization from the right-most graph layer and proceeds left. When optimizing a layer, the method optimizes each node at a time and after all nodes have been optimized in the layer, it backups the value function.

In order to improve the value of the policy graph, the new improvement method in Figure 3.3 projects $b(s,\vec{q})$ from time step zero to the end of the horizon T using the current policy. The value function $V_{T+1}(s,\vec{q})$ is initialized to zero. The method optimizes each layer starting from the last graph layer and ending at the first layer. In each graph layer, the method optimizes for the current belief $b_t(s,\vec{q})$, at time step t, the action and observation edges of an agent's graph node (the policy for an agent's graph node) by holding other agents' policies fixed. The formulas for selecting the action and the observation edges are shown in Algorithm 1 at lines 6–9 in Publication III. If the found policy for the node (action and observation edges) is identical to the policy of another node of the agent, then the new redundant policy is discarded and a new node policy is optimized for a sampled belief. When there are two or more agents and a policy is discarded, the incoming observation edges to the optimized node need to be redirected to the node, which had the identical policy. Without this redirection, optimization of other agents' nodes may result in lower value. This happens because the optimization of another agent's node's policy depends on the policies of other agents' nodes. For only one agent, this is

not a problem, because a node's policy does not depend on the policies of other nodes of the same agent. After all the nodes in a graph layer have been optimized, the value function is backed up:

$$V_t(s,\vec{q}) = \sum_{s',\vec{a},\vec{o},\vec{q}'} \left[R(s,\vec{a}) \prod_i P_t(a_i|q_i) + \gamma \prod_i P_t(q_i'|q_i,o_i) P(s',\vec{o}|s,\vec{a}) V_{t+1}(s',\vec{q}') \right] . \tag{3.1}$$

3.3.3 Periodic FSC improvement

In order to construct periodic deterministic FSCs, a finite-horizon policy graph is optimized first and then the last layer of the policy graph is connected to the first layer (see Publication III for details on how the connection is done). The resulting periodic FSC could be used as such, but would most likely not yield high value without further optimization. The infinite-horizon optimization method for periodic FSCs is adapted from the monotonic policy graph value improvement method discussed in the previous section. The infinite-horizon optimization method is deterministic and produces deterministic FSCs.

The main idea of the infinite-horizon periodic FSC improvement method is to optimize one FSC layer at a time similar to the finite-horizon monotonic policy graph value improvement method. This can be achieved by dividing the reward sum over time periodically. In more detail, when the world starts from belief $b_0(s,\vec{q})$ and the agents follow a periodic controller policy with period M, the expected discounted reward is

$$\begin{aligned} &\sum_{t=0}^{\infty} \sum_{s,\vec{q},\vec{a}} \gamma^t b_t(s,\vec{q}) R(s,\vec{a}) \prod_i P_t(a_i|q_i) = \\ &\quad \sum_{s,\vec{q},\vec{a}} \sum_{t=0,M,2M,\ldots} \gamma^t b_t(s,\vec{q}) \prod_i P_t(a_i|q_i) \\ &\qquad \left[R(s,\vec{a}) + V_{(t+1,t+M)}(s',\vec{q}') P(\vec{o},s',|s,\vec{a}) \prod_i P_t(q_i'|q_i,o_i) \right] , \end{aligned} \tag{3.2}$$

where $b_t(s,\vec{q})$ is the belief projected t time steps forward using the current policy and $\sum_{t=0,M,2M,\ldots}$ is the periodic sum over time. The value function $V_{(t+1,t+M)}(s,\vec{q})$ corresponds here to the expected discounted reward, when starting from world and controller states s and $\vec{q}$ in time step $t+1$ and following the current policy for $M-1$ steps into time step $t+M$. Because of policy periodicity, $V_{(t+1,t+M)}(s,\vec{q}) = V_{((t+1) \mod M,(t+M) \mod M)}(s,\vec{q})$, and thus $V_{(t+1,t+M)}(s,\vec{q})$ is identical for all $t = 0, M, 2M, \ldots$.

In practice, the periodic FSC method computes $V_{(t+1,t+M)}(s,\vec{q})$ by backing up $M-1$ steps periodically (at the last layer the first layer is backed

up) starting from the layer $t+M$, initialized to zero value, and finishing at the layer $t+1$. Because $V_{(t+1,t+M)}(s,\vec{q})$ does not depend on the policy of the periodic FSC layer in time step t, the method can efficiently optimize the policy for layer t similarly to the finite-horizon monotonic improvement algorithm.

Using the computation of value functions as described above, the computational procedure of the complete periodic FSC improvement method is as follows. First, the method projects a belief to a sufficient horizon. Then the method computes the discounted sum of the beliefs for each periodic layer l: $\hat{b}_l(s,\vec{q}) = \sum_{t=l,M+l,2M+l,\ldots} \gamma^t b_t(s,\vec{q})$. Then the method optimizes each FSC layer l like the monotonic improvement algorithm does, using the current belief $\hat{b}_l(s,\vec{q})$ and the next time step value function $V_{(l+1,l+M)}(s,\vec{q})$. Note that this method does not guarantee monotonic value improvement, since policy changes affect beliefs beyond one policy period, but because of discounting the approximation error nevertheless decreases exponentially with the period M. Note that in the case when the approximation is exact (no approximation error), the finite state controller policy converges to a (local) optimum in the limit, because of monotonic policy improvement.

3.3.4 Periodic expectation maximization

The expectation maximization (EM) approach for POMDPs and DEC-POMDPs discussed in Section 2.8.1 optimizes stochastic FSCs. Periodic EM, presented in Section 3.2 of Publication III, optimizes periodic stochastic FSCs. Periodic EM retains the theoretical properties of the standard EM approach (monotonic convergence to a local optimum). A periodic deterministic controller is a local optimum for EM, but a periodic deterministic controller plus a small amount of noise can be used as initialization for periodic EM.

The EM algorithm performs an E- and M-step in one iteration. The new E-step computes α and β messages for each layer separately. In order to derive the new M-step, Publication III transforms the log-likelihood into a sum, where each sum component contains only parameters from the same periodic FSC layer. The update for each layer is then independent of the other layers and one iteration of the periodic EM method has linear complexity with respect to the number of layers.

Using periodic EM with the factored infinite-horizon DEC-POMDP approach. Publication V discusses how wireless channel access can be described as a DEC-POMDP and how policies for the DEC-POMDP can be computed. Therefore, Publication V combines periodic EM described in Section 3.2 of Publication III with the factored EM approach for infinite-horizon DEC-POMDPs introduced in Publication II. The combination is straightforward, because although the factored EM approach is based on a form of EM in which probabilities are projected only forwards in time, the fundamental idea how the log in the log-likelihood separates FSC parameters into components of a sum remains unchanged. Therefore, in the combined approach, the M-step update for a layer of a periodic FSC can be performed independently of the other layers.

4. Spectrum access in wireless networks

In a wireless network, an agent (mobile phone, laptop, access point, cell tower, ...) accesses the frequency spectrum in order to transmit data to its intended receiver. The quality of the transmission depends on the interference level at the receiver. Interference means that signals of other transmitters, and background noise, decrease the relative signal power of the transmitter at the intended receiver. Thus, to maximize the probability of a successful transmission an agent should access the spectrum when interfering agents are not transmitting, access the spectrum at an interference free spatial location, or use an unused frequency channel.

There is a large body of research that focuses on optimizing and analyzing wireless agent behavior in these time [40, 158, 36, 94, 172, 53, 35, 4, 89, 154, 66], spatial [75, 174, 47, 48, 74, 7, 73], and frequency [172, 53, 35, 4, 89, 154, 66] dimensions. Publication IV optimizes the policy of a wireless agent in the time and frequency dimensions and Publication V optimizes the policies of multiple agents in the time and spatial dimensions.

Cognitive radio refers to the ideal radio device that makes intelligent decisions based on information sources such as the current frequency spectrum, other radio devices, and the user. In a large part of cognitive radio research, a cognitive radio differs from other wireless devices in that it tries to avoid interference to legacy primary users, while it tries to maximize its throughput or another performance measure. In Publication IV, the spectrum access policy of a cognitive radio [97, 61] is optimized in an environment with multiple frequency channels on which primary users operate.

In a wireless network, the interference from one wireless agent to another depends on the distance between the agents, because of the signal attenuation. Therefore, an agent does not interfere with other far away

agents. On the other hand, the traffic that users of wireless devices generate is bursty. Idle periods of agents with bursty traffic provide transmission opportunities for other agents. It follows, that wireless channel access should be optimized over both the temporal and spatial dimensions. In order to take both time and spatial dimensions into account, the channel access problem is formulated as a factored DEC-POMDP in Publication V. As discussed in Section 2.6 DEC-POMDPs optimize policies over time. Moreover, because of the weak interaction of far away wireless agents, that is, the spatial dimension of the problem, the transmission success of an agent depends only on a subset of other agents. This makes it possible to encode the spatial interaction of agents into the factored DEC-POMDP in Publication V.

Section 4.1 discusses cognitive radio background and the cognitive radio approach in Publication IV. Section 4.2 presents first background on wireless channel access and then the channel access approach in Publication V.

4.1 Cognitive radio

There is a pressing need for more wireless bandwidth as mobile users begin to use new applications and data centric services. Different radio frequencies have been assigned to wireless devices to transmit on, but currently the radio frequency spectrum is congested, especially in high-density urban areas. Historically radio frequencies have been statically allocated to licensed users, that is primary users, such as TV-stations or mobile operators. However, studies [5, 157] show that frequencies are heavily under-utilized.

Currently in both urban and rural areas only a fraction of all spectrum opportunities are used. Radio frequency legislators are opening up parts of the spectrum under conditions that secondary users do not interfere with the licensed primary users. In order to optimally utilize the unused spectrum in the time, space, and frequency dimensions, a *cognitive radio* [97, 61] is needed.

Traditional wireless protocols have been manually designed by domain experts using assumptions about the radio environment in which the wireless devices operate. Because of these assumptions, traditional protocols do not adapt to conditions, which the wireless protocol designers did not take into account. Hence, traditional protocols cannot take advantage of

all the opportunities in the current radio environment, but an ideal cognitive radio device can. In order to achieve a high performance level, a cognitive radio must be able to adapt to changes in the operating environment and be able to make decisions based on noisy observations. The spatial locations of primary and secondary users, their temporal access patterns, bandwidth needs, battery levels, sensor and transmitter constraints, and the wealth of many other parameters make cognitive radio tasks, such as dynamic access of frequencies and time slots, challenging.

This section first discusses cognitive radio background, with a focus on dynamic spectrum access, and then discusses opportunistic spectrum access formulated as a POMDP.

4.1.1 Background

The cognitive radio (CR) literature comprises a wide variety of topics (see [5, 171, 157] for comprehensive surveys), but this thesis focuses on *dynamic spectrum access* [61, 171] in cognitive radio networks.

Dynamic spectrum access [61, 171] in cognitive radio networks means dynamically allocating a spatially and temporally varying limited set of channels to several cognitive radios (in this discussion the radio spectrum is divided into distinct frequency channels and a channel refers to one of these channels). Note that from the point of view of a cognitive radio, wireless channel quality varies over time, because the interference from other wireless devices evolves. Furthermore, the interference level of a channel may change when the spatial location of the cognitive radio changes, because interference depends on spatial distance and on other properties of the wireless network, such as signal strengths and (moving) objects affecting signal propagation. Cognitive radios sense the state of the spectrum and decide which channels to use. The aim is to dynamically allocate unused spectrum efficiently to cognitive radios. Dynamic spectrum access is a difficult problem, because of the complex combinatorial aspect of selecting an optimal set of channels for each of multiple cognitive radios, while obeying interference and fairness constraints. In a decentralized ad hoc network, which is an appropriate model [172] for cognitive radio networks, the cognitive radios have to perform spectrum access decisions independently based on partial/incomplete information at hand.

In general, cognitive radio networks can be categorized [171] into networks in which users are entitled to use a part of the spectrum exclu-

sively, to networks where anyone can use the license-free spectrum (regular WLAN is an example of this), and into networks where the users are divided into primary and secondary users. The last network type can be further divided into *overlay* and *underlay* networks. In underlay networks, simultaneous transmissions by cognitive radios do not cause collisions with primary users, but in underlay networks cognitive radios usually have to restrict their transmit power to low levels to prevent collisions and use spread spectrum techniques. Because of the limited transmit power, the distance between cognitive radio transmitters and receivers is restricted. In overlay networks cognitive radio transmissions interfere with primary user transmissions if they occur at the same time. Therefore, such cognitive radios usually listen to primary user transmissions in order to predict when primary users do not transmit.

Temporal exploitation of frequency spectrum, that is *opportunistic spectrum access*, has received considerable attention [172, 53, 46]. In opportunistic spectrum access, the goal is to utilize time slots that are not used by primary users. A primary user can be a legacy protocol user that uses for example a common WLAN protocol for transmissions, a TV-station, or a mobile operator. In opportunistic spectrum access, the network corresponds to an overlay network. Figure 4.1 shows an example of primary user traffic (traffic used in Publication IV) on four frequency channels. There are long unoccupied periods, but the complicated traffic patterns of the primary users indicate that predicting unoccupied channels is nontrivial.

There exists research on predicting unoccupied channels. For example, in [53] a secondary user predicts primary user channel access using a traffic pattern model estimated from experimental data. Zhao et al. [172] model primary user network traffic with a two-state Markov model that has an idle and a busy state. Because current radios can only sense a portion of the spectrum at a time, in the model in [172] the secondary user transmits or senses a single channel at a time. Sensing a channel reveals the state of the channel: “idle” or “busy”, but the state of the other channels remains hidden. Because the complete state is not observable, Zhao et al. [172] model the problem as a POMDP (see Section 2.3 for a definion and description of a POMDP in the general case). The problem can also be defined as a special case of a restless bandit problem [4, 150].

The simple two-state Markov model used in [172] allows modeling bursty traffic where primary users do not react to collisions and because of the

simplicity of the model, it is amenable for analysis. Most research on this problem [35, 4, 89, 154, 66] assumes that primary users do not react to collisions, that Markov models have two states, and that each channel evolves independently. The next section discusses our new POMDP approach in Publication IV that models the effects of collisions with primary users and that models idle times and packet bursts of differing length using several states.

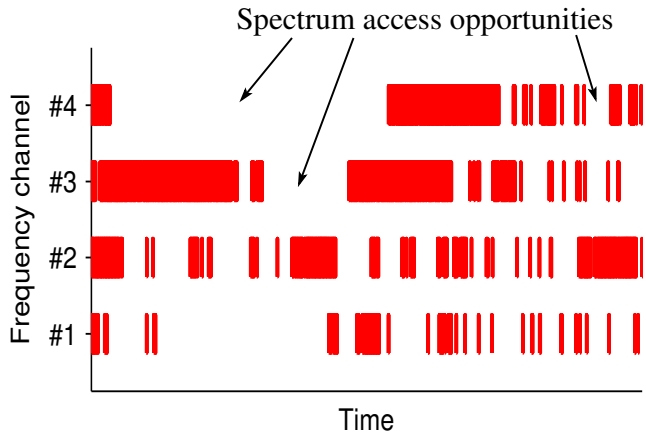


Figure 4.1. Snapshot of idle and busy periods on wireless channels. On each of the four frequency channels a primary user transmits web and voice over IP (VOIP) traffic. The traffic was generated by the NS-2 network simulator [93]. Red lines denote traffic, white space denotes idle periods, and arrows show examples of spectrum access opportunities. Note the complicated bursty nature of primary user traffic patterns.

4.1.2 Opportunistic spectrum access as a POMDP

Publication IV presents a new approach for opportunistic spectrum access. Legacy primary users, who are not aware of cognitive radios, operate according to their legacy protocol on several frequency channels. The goal is to optimize the policy of a secondary user, so that the secondary user can transmit as much data as possible, while at the same time not disturbing primary users.

The transmission of a secondary user may interfere with the current transmission of a primary user and disrupt the primary user's transmission. Furthermore, even if the primary user is not transmitting currently, the transmission of the secondary user may affect the future behavior of the primary user. For example, in the commonly used IEEE 802.11 WLAN network wireless devices use carrier sense multiple access (CSMA) pro-

tocols, which prevent primary user transmission until the primary user senses the channel idle. Therefore, if a secondary user does not take into account primary user behavior, it can inadvertently hijack a channel. To model different packet and idle burst lengths, and to model primary user responses to secondary user traffic, Publication IV presents a new kind of Markov process model. The primary user traffic shown in Figure 4.1 concretely illustrates the need for explicit modeling of long and short bursts and both idle and busy bursts. In addition, the new Markov model takes into account explicitly primary user reactions to secondary user (cognitive radio) channel access, which is crucial for operating in wireless networks such as regular WLAN networks.

Figure 4.2 illustrates the operation of the wireless system described in Publication IV for five frequency channels. In each time step, the secondary user senses three adjacent channels and may transmit on one of them. Each primary user is modeled with a Markov process, which is discussed in more detail in Section 4.1.2. The goal of the secondary user is to predict when a primary user listens to a channel or when a primary user transmits on a channel. If the secondary and primary users transmit at the same time, then there is a collision and both transmissions typically fail. If the secondary user transmits when a primary user listens to the channel, then primary user transmission is postponed. The secondary user cannot distinguish between primary-user-listening and channel-idle situations using sensing, but must instead predict the intentions of the primary user. Next, the primary user channel model will be discussed in more detail and then the computation of policies for many frequency channels.

Realistic model for primary users

The main idea of the primary user Markov model in Publication IV is to model both short and long traffic bursts and to take into account the interference that the secondary user causes to a primary user that transmits or intends to transmit data. The primary user Markov model consists of sets of listen, listen collision, transmit, transmit collision, and idle states. When the primary user is in an idle state and decides to transmit data, it moves to a listen state. If it senses the channel unoccupied it moves next to a transmit state. When the primary user listens to its channel with the intention to transmit, but the secondary user transmits on the channel, the primary user moves to the listen collision state. Similarly,

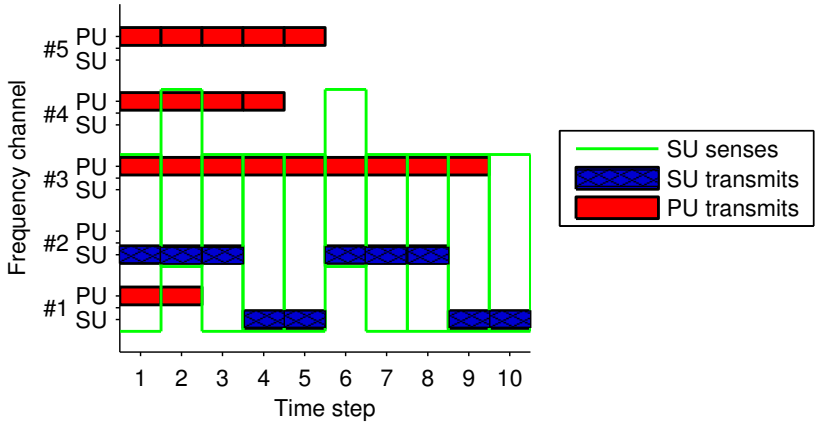


Figure 4.2. Illustration of secondary user spectrum access in a wireless network with five primary user (PU) frequency channels denoted $\#1, \ldots, \#5$. At each time step the secondary user (SU) senses three adjacent channels and may transmit on one of them. The sensed channels are denoted by a green rectangle and the transmission is denoted by a blue filled box. Primary user transmissions are denoted by a red filled box. The goal of the secondary user is to transmit without causing interference to any of the primary users.

the primary user moves to the transmit collision state, if the secondary user transmits, when the primary user would have transmitted.

There are J idle and K transmit states (and the corresponding listen and collision states). Intuitively, each state corresponds to a certain burst length. Figure 1 in Publication IV shows the states and possible state transitions when the secondary user transmits and also when it does not transmit. In Publication IV, the probability to move from a certain idle state to another transmit state and vice versa was determined from the simulated network data. The other transition probabilities are deterministic and can be derived directly from the Markov model specification for primary user behavior, that is illustrated in Figure 1 in Publication IV.

The secondary user policy is optimized within the POMDP framework described in Section 2.3. In a POMDP, the Markov probability model describes how the system evolves over time in response to agent actions, but the optimization objective is specified using rewards. In POMDPs, the agent is assigned at each time step a real valued reward that depends on the current world state and the action of the agent. In wireless spectrum access, the reward can be based on performance measures such as throughput or packet delay, but also for instance monetary rewards (or penalties), negotiated between secondary and primary users, could be assigned for situations where a secondary user interferes with

primary users. Because collisions with primary users have highest priority, in Publication IV, primary user listen and transmission collisions were penalized with the reward of -10. Secondary user transmissions were rewarded with $+1$, and because of the energy consumption, the sensing was penalized with -0.01. Note that only the relative values of the rewards matter when solving POMDPs.

Publication IV tests the new more realistic Markov model approach in experiments with four, five, and six channels using a combination of voice-over-IP (VOIP) and web traffic. In numerical comparisons (see Publication IV for details), the new approach clearly outperforms several two-state Markov model approaches.

Computational difficulty

Because the state of a channel does not directly depend on the state of all other channels (here the state of a channel does not depend on any other channel state), the opportunistic spectrum access problem is defined as a factored POMDP in Publication IV. A factored POMDP definition makes it possible to fit the POMDP model in a limited amount of memory and also makes it possible to find solutions efficiently (see Section 2.5 for details on factored POMDPs and solution methods for them). In Publication IV, Symbolic Perseus [113], a factored POMDP method, is used to compute the policy of a secondary user. In the experiments in Publication IV, the Markov model of a channel has 15 states and thus the size of the complete state space is 15^N, where N is the number of channels. Symbolic Perseus cannot handle large numbers of channels, but Publication I (discussed in Section 3.1) introduces a new factored POMDP method, which scales to larger problems and finds solutions to the opportunistic spectrum access problem with a larger number of channels.

Summary

This section presented a new approach for modeling primary user traffic and making channel access decisions based on the model. In contrast to traditional two-state Markov models, the new model takes into account packet and idle bursts of varying lengths. Furthermore, the new model takes into account the primary user's reactions to cognitive radio transmissions. Experiments in Publication IV demonstrate that these properties are crucial for transmitting large amounts of data and for limiting the number of collisions with primary users.

4.2 Wireless channel access

The previous section discussed optimizing the behavior of a single wireless agent. In wireless networks, such as in widely used wireless local area networks (WLANs), multiple agents transmit data. In wireless networks, the transmission of one agent may interfere with the transmissions of other agents. Therefore, one widely studied problem in wireless networks is how to ensure that agents that interfere with each other do not transmit at the same time.

The next section discusses background on wireless network channel access. After that, Section 4.2.2 discusses the new channel access approach based on factored DEC-POMDPs, presented in Publication V.

4.2.1 Background

Channel access is a wide and active research field with diverse approaches. Figure 4.3 illustrates how the interaction of wireless agents, that access the same channel, depends on spatial locations. Each agent has a transmit queue from which it transmits data. The transmit queue gets new data when, for example, the user of a mobile wireless device clicks on a new web page and the corresponding web page request is inserted into the transmit queue of the mobile device. The agents must decide, when to transmit data based on whether they have data in the transmit queue and whether they anticipate that interfering agents will be transmitting. This section will proceed with channel access methods that take into account temporal dynamics, then continue with spatial methods that take advantage of interference diminishing with distance, discuss multi-agent techniques for channel access, and finally present the DEC-POMDP based approach proposed in Publication V for channel access.

Wireless channel access protocols

Wireless channel access can be divided into *scheduled* and *contention based* access. In scheduled access, someone else than the wireless devices themselves decides when transmissions occur. That is, someone schedules the channel access. In scheduled access, the decision maker often receives information from the wireless devices and can thus make more informed decisions than individual decision makers could. However, communication of information consumes network bandwidth. In *time division multiple access* (TDMA), channel access is statically distributed among wireless

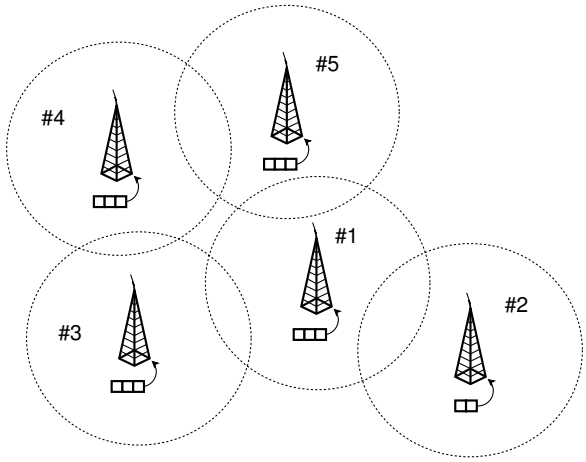


Figure 4.3. Example of interfering wireless agents. Five wireless agents (wireless networks or devices) with overlapping operating ranges. Agents whose operating ranges overlap may interfere with each other. For example, agent 1 may interfere with agents 2, 3, and 5. Each agent has a transmit queue with a certain amount of data. The problem is how to decide which agents should transmit at which time.

devices: each time slot is assigned to a specific wireless device and channel access by other wireless devices in that time slot is forbidden. In mobile networks, the base station decides when mobile phones are allowed to transmit. In WiMAX [2, 3], a central controller allocates a time slot to each transmitter. In contrast to scheduled systems where a central controller tells wireless devices when and how to transmit, in polling based systems [101], a central controller polls wireless devices for data.

In contention based systems, wireless devices listen to a channel and decide on their own when to access the channel, based on observations about the channel state and their own needs. In effect, wireless devices contend for channel access, because the number of devices on the channel is limited. The most widely known set of standards for wireless networking is IEEE 802.11 [1], that specifies how wireless devices should access channels in the frequency bands that are widely available for contention based access. For example, most personal computers use IEEE 802.11 for wireless access and improvements to 802.11 are a widely researched topic. Channel access of IEEE 802.11 is based on carrier sense multiple access with collision avoidance (CSMA/CA). In CSMA/CA, when a device wants to transmit data, it has to monitor the channel until it detects it to be free. In order to prevent a single device from occupying the channel all the time, CSMA/CA includes randomization. When a listen or packet

collision happens a device has to wait a uniformly random number of idle slots. A listen collision happens when the channel is occupied at the end of a waiting period, and a packet collision happens when two or more devices transmit at the same time. The maximum waiting time of a device doubles after every collision in order to make it less likely for another collision to occur. The doubling of the maximum waiting time is commonly referred to as the exponential backoff and the waiting time is referred to as the contention window.

In principle, CSMA/CA makes contention based communication in a network with many wireless devices possible, but unfortunately basic CSMA/CA does not guarantee high channel usage, that is, there may be wasted idle periods on the channels. Therefore, enhancements to CSMA/CA that improve the channel allocation over time have been widely researched. Especially tuning of CSMA/CA parameters, such as tuning the backoff mechanism [158, 36, 94] yields increased performance. Another problem with basic CSMA/CA is that high priority traffic has to compete with low priority traffic. This has been addressed by using different contention window sizes for different priority classes [40].

Improving the behavior of CSMA/CA and other wireless protocols over time allows higher network performance. Next, performance improvement through spatial reuse will be discussed.

Spatial reuse

In wireless networks, the interference caused by a wireless device to other wireless devices depends on the distance between the devices. This creates transmission opportunities: two devices far away from each other can transmit successfully at the same time, that is, a device can *spatially reuse* [75, 7] spectrum when non-interfering devices are transmitting. Spatial reuse is related to the so-called hidden and exposed terminal problems. The hidden terminal problem refers to the situation where a wireless device interferes with the transmissions of a second wireless device, but the second device can not observe transmissions of the first device, leading to packet collisions. The exposed terminal problem refers to a situation where a wireless device does not interfere with a second wireless device, but the second device does observe transmissions of the first device, leading to wasted transmission opportunities. Figure 4.3 provides examples of spatial reuse possibilities in a wireless network.

In contention based channel access protocols, such as IEEE 802.11, a

wireless device determines whether another device is transmitting data by measuring the signal power level on a channel. If the power level is over the carrier sense threshold, the channel is assumed occupied. In basic IEEE 802.11, the carrier sense threshold is identical for every wireless device and set to a very low value in order to prevent packet collisions. However, research shows that network performance can be increased by tuning carrier sense thresholds of wireless devices to take advantage of spatial reuse [174, 47, 48, 73]. Another parameter that can be tuned for spatial reuse is the transmit power [47, 48, 74]. The transmit power of a device is strongly linked to the carrier sense threshold in the sense that the transmit power and the carrier sense thresholds of other devices determine whether other devices detect the device's transmissions.

Another approach that has been investigated for maximizing spatial reuse is to use directional [77, 76] or MIMO antennas [107], that allow interference only to selected directions (the IEEE 802.11n wireless standard has support for multiple antennas [156]). Transmission rate tuning [22] has also been investigated in the literature.

Multi-agent approaches to channel access

A wireless network consists of a set of wireless agents that transmit data. Each agent monitors channel(s) and uses past observations to decide on which channel(s) to transmit. A question that arises is whether multi-agent techniques could be used for planning wireless channel access. As discussed in Section 2.6, in order to compute optimal policies in a multi-agent system with imperfect sensing and uncertain dynamics, one has to plan actions into the future by considering observation histories of all agents for all possible action-observation sequences. The decentralized partially observable Markov decision process (DEC-POMDP) model (see Section 2.6 for background information on DEC-POMDPs) is a model for formalizing this and solving a DEC-POMDP yields optimal policies for the wireless agents. "Optimal" here means the (global) optimum of the DEC-POMDP when the Markov model assumptions are true. In practical applications the performance of the DEC-POMDP policy depends on how well the Markov model describes the wireless environment and how well practical algorithms are able to compute a high performance solution.

Previous work on wireless channel access using multi-agent techniques is limited to special cases. Shirazi et al. [135] present a DEC-POMDP model for wireless relays. In [135], wireless agent actions do not affect

the state of the world and the goal is to find the best wireless relay for the current channel conditions. Multi-agent techniques such as multi-agent Q-learning [49, 84, 164] have been used in cognitive radio research. However, cognitive radio research [49, 84, 164] focuses on optimizing the allocation of frequency channels among secondary users, while minimizing interference to primary channel users. For example, the transmit queues of secondary users are omitted in [49, 84, 164] and the main goal is not the optimization of channel access between secondary users.

In general, the crucial properties of channel access are captured by a DEC-POMDP: observations are partial (sensing is imperfect and an agent does not observe the transmit queus of other agents), how the world evolves is uncertain, and the information exchange between agents is restricted. Moreover, the objective in a DEC-POMDP is to optimize co-operative behavior instead of optimizing the behavior of single agents that try to maximize their own gain. Consider the case of a wireless network topology in which one agent can block the transmissions of all other agents. A self-interested agent may choose to transmit all the time even if its transmissions prevent successful transmissions by other agents. In an efficient co-operative solution, the agents would try to avoid interfering with other agents in order to maximize the joint objective, for instance total throughput. Therefore, in principle, a good solution to channel access can be found by solving a DEC-POMDP. The next section discusses how wireless channel access can be formulated as a factored DEC-POMDP and how agent policies can be (approximately) optimized centrally to maximize the spatial and temporal reuse, and then executed decentrally.

4.2.2 Channel access as a factored DEC-POMDP

Publication V shows how wireless channel access can be formulated as a factored DEC-POMDP. Figure 4.4 illustrates the wireless model used in Publication V. The network consists of N transmitter-receiver pairs. The transmit queue of a transmitter (the terms transmitter and wireless agent are used interchangeably) is filled by a source traffic Markov process, whose states are categorized into “packet” and “idle” states. The Markov process can move from a packet state to the same packet state or to any idle state and from an idle state to the same idle state or to any packet state. A move from a packet or idle state to the same state represents an ongoing packet or idle burst and the probability of the move defines the expected burst length. The source traffic model is similar to

the Markov model in Publication IV, but without collision or listen states. The Markov model allows for bursty network traffic with varying burst lengths (for simplicity the experiments use a two-state Markov model).

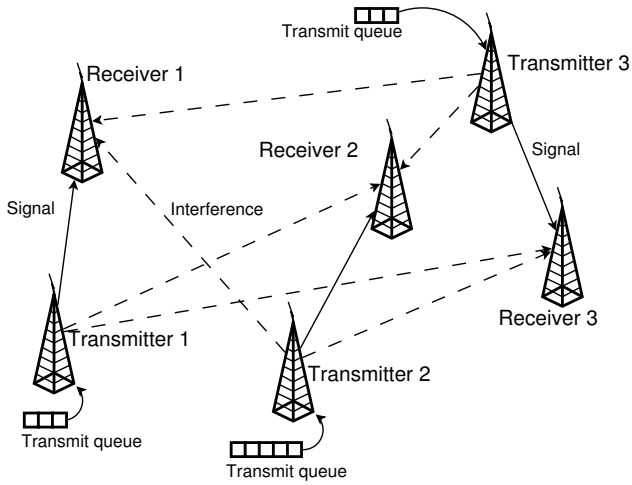


Figure 4.4. Example of a wireless network with three wireless transmitters communicating with their intended receivers. Each transmitter transmits data from a transmit queue that is filled by a stochastic process that models network traffic, for example web or VOIP traffic. The general goal is to empty the transmit queues as fast as possible, that is, to transmit data at a high rate. The channel capacity (the highest amount of data a transmitter can transmit) of transmitter i depends on the ratio of the signal power at receiver i and the interference of other transmitters $j \neq i$ at receiver i plus background noise. The interference caused by j at receiver i depends on the distance between j and the receiver i.

Intuitively, the goal is to keep transmit queues as empty as possible. A transmitter may transmit data according to the Shannon capacity, which is a theoretical upper bound on the amount of transmitted data. The Shannon capacity is measured at the receiver and depends on the signal power of the transmitter, on the interference caused by other transmitters, and on the background noise. Essentially, the spatial configuration determines capacities, because the interference power depends on the distance between the receiver and the interferer. To summarize, for determining efficient transmitter policies one has to consider the state of the transmit queues and the evolution of the transmit queues into the future, but also the spatial configuration, which determines transmission capacities.

DEC-POMDP channel access

At every time step, each transmitter observes the status of its transmit queue and an interference level, from which Shannon capacity can be estimated. Next, a transmitter either listens or transmits. Each transmitter has a policy, which decides on transmissions based on past observations. Optimal transmitter policies take into account the future evolution of transmit queues and source models in response to policy decisions and future observations. Assuming that the world is Markovian and that the optimization goal can be described with rewards, then solving the problem as a DEC-POMDP yields optimal policies. Publication V uses the factored infinite-horizon DEC-POMDP method in Publication II, with modifications, to compute stochastic finite state controller transmitter policies. This approach has several advantages. Each transmitter policy is optimized for the current network topology and source models, and policies can take sensor noise into account. The optimization objective can be selected freely, for example delay or throughput (complicated performance measures can be implemented by adding approriate states to the DEC-POMDP). Furthermore, stochastic finite state controllers allow complex behavior, for example they can implement both contention based random access behavior similar to IEEE 802.11 and deterministic access behavior similar to TDMA. Finally, for small enough finite state controllers, a human expert inspecting the controllers may gain valuable insight into what kind of policies work in what kind of spatial and temporal configurations.

To formalize the wireless network problem as a DEC-POMDP several issues must be addressed. The size of the state space of the transmit queues of the agents is exponential in the number of agents and a straightforward DEC-POMDP specification is not possible. However, because interference diminishes with distance, a transmitter's capacity depends in practice only on a limited set of other transmitters and a factored description of the problem is possible. Publication V formalizes the channel access problem as a factored DEC-POMDP and computes stochastic finite state controller policies with the help of the infinite-horizon factored DEC-POMDP method presented in Publication II. To describe the wireless network problem as a factored DEC-POMDP, Publication V uses the following modeling approximations:

- In the wireless network problem, transmit queue sizes and capacities are continuous valued, but the DEC-POMDP is discrete valued. *Ap-*

proximation: Approximate continuous values with probabilities. For example, if the continuous value is 0.5 and possible discrete values are 0 and 1, then transition to either 0 or 1 with probability 0.5.

- All transmitters interfere to some degree with other transmitters even if the interference level is low. *Approximation*: When computing the probability for the transmission capacity of a transmitter to be at a certain level, take into account largest interferers fully and compute an aggregate interference level for the smallest interferers.

Furthermore, in order to compute actual policies, Publication V proposes improvements to the factored infinite-horizon DEC-POMDP method:

- Publication V uses a non-linear reward scaling approach to speed-up convergence

- Publication V uses periodic controllers presented in Publication III to compute larger controllers

- In order to start optimization from a more likely long term probability distribution than the pre-defined initial belief (see Section 2.6 for definition of initial belief), for a specific spatial configuration, Publication V projects the initial belief many time steps forward before starting optimization from it.

Experiments. In Publication V, experiments compared the proposed DEC-POMDP approach with basic CSMA/CA and two different CSMA/CA versions with parameters tuned specifically for the spatial configuration at hand, on different uniformly randomly generated spatial configurations. For a concrete spatial configuration example, see Figure 4.5. Overall, the DEC-POMDP approach outperformed basic CSMA/CA and the two different CSMA/CA versions optimized for the spatial configuration.

Summary

This section discussed how the wireless channel access problem in Publication V is formulated as a factored infinite-horizon DEC-POMDP. DEC-POMDPs are a general decision making model that take into account partial observability, uncertainty of future states, and the interplay be-

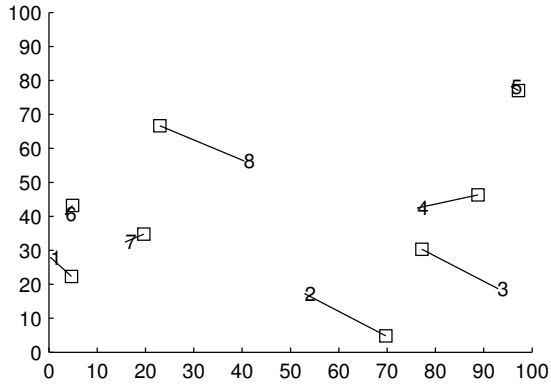


Figure 4.5. Wireless network with transmitters (numbers) and their receivers (boxes). Each transmitter interferes with receivers of other transmitters.

tween multiple co-operative agents. In DEC-POMDPs, the optimization objective is easy to specify in terms of the actual wireless network optimization objective such as minimizing the mean delay (mean delay is the average time that data has to wait in a queue before being discarded or transmitted) or maximizing the sum throughput (sum throughput is the total amount of successfully transmitted data per time step). Because of the generality of the model, computing optimal policies for even small (factored) DEC-POMDPs is intractable. Therefore, the contribution in Publication V includes showing how wireless channel access can be formulated as a factored DEC-POMDP, but also how efficient policies for the factored DEC-POMDP can be computed in practice by utilizing the factored infinite-horizon DEC-POMDP method proposed in Publication II.

5. New approach for spectrum sensing

A wireless agent makes observations about its environment. Based on these observations, it decides on its transmissions and on other activities. For example, in the common CSMA/CA protocol discussed in Section 4.2.1 wireless agents listen to the frequency channel and start transmitting only when the channel is observed to be free for a certain period of time. The performance of the protocol depends strongly on the performance of the frequency sensor. If the sensor classifies the channel as occupied, when it is not, a transmission opportunity is missed. If the sensor classifies the channel as free, when it is occupied, the wireless agent may start a new transmission that collides with ongoing transmissions of other wireless agents.

In cognitive radio research (see Section 4.1), sensing of the frequency spectrum may have even more importance than in other more traditional wireless networking scenarios, because the usual goal is to optimize the behavior of secondary users that try to avoid interfering with primary users. Preventing interference to primary users has highest priority (as in Publication IV). If sensing is inaccurate, then interfering with primary users becomes more likely. Even in cognitive radio approaches that take imperfect sensing into account (e.g., by POMDP models, see Chapter 4) sensing accuracy may heavily influence performance.

From the point of view of a mobile device that tries to take advantage of transmission opportunities over a wide range of frequencies there are two problems with frequency sensing. The first is that standard sensing techniques sense only a small portion of the frequency spectrum at a time. The second is that sensing consumes energy, which may be a big problem in low power devices such as mobile phones. Therefore, Publication VI and Publication VII discuss an analog passive *nanoscale* sensing approach that is designed to operate without any external power and to be used

over a wide range of frequencies. Because of expected nanoscale faults, the approach uses a fault tolerant *radial basis function network* for signal classification.

Next, in Section 5.1, this thesis gives some background information on nanotechnology and then, in Section 5.2, it describes the new nanoscale spectrum sensing approach.

5.1 Background

This section presents background information on the concepts involved in Publications VI and VII, namely spectrum sensing in Section 5.1.1, nanotechnology in Section 5.1.2, and fault tolerant machine learning methods in Section 5.1.3, including research in radial basis function network (RBFn) fault tolerance. Publications VI and VII use these concepts in a spectrum sensing design in which signals propagate through a nanoscale circuit. The signals are classified in Publications VI and VII by a radial basis function network trained to tolerate faults in the nanoscale circuit.

5.1.1 Spectrum sensing

Cognitive radios need to sense the spectrum to determine, whether a frequency channel is free or occupied. Although cognitive radios may receive information from sources such as the Global Positioning System (GPS) or user input, for the task of using unoccupied frequencies or communicating to other radio devices, the state of the frequency spectrum is the most important information source.

The survey [169] discusses spectrum sensing in cognitive radio networks. Spectrum sensing can be divided into co-operative sensing [6], where multiple wireless devices work together to jointly gather information about the spectrum, and individual sensing, where a device gathers information about the frequency spectrum on its own. In single device spectrum sensing, the spatial location, the sensing equipment, and computational power of the device limit the possibilities for reliable wide spectrum sensing. Combining the sensing results of multiple devices co-operatively allows one to circumvent these problems at least partly [85, 129, 170, 6]. However, compared to individual sensing, co-operative sensing requires extra communication and complicates the design of wireless devices and networks. Publications VI and VII present a sensing approach for an in-

dividual device, that is designed to sense over a wide range of frequencies with minimal power use.

Common techniques for spectrum sensing are energy detection [155], waveform-based sensing [149], matched filtering [118], and cyclostationary sensing [51]. The simplest technique of these, energy detection [155], assumes that a channel is occupied if the detected signal energy (power over time) is above a pre-defined threshold. The accuracy of energy detection depends on how well the threshold is defined and on the sensing time. Compared to more sophisticated approaches, energy detection does not take into account patterns in the signal. In waveform-based sensing [149], part of the signal pattern is assumed to be known a priori. Waveform-based sensing is applicable to detecting wireless protocols that include pre-defined sequences such as preambles. In matched filtering [118], the signal pattern is known completely a priori and if the observed signal matches the known signal, then the channel is assumed to be occupied. Cyclostationary sensing [51] assumes periodicity in the signal or its statistics. Many signals such as orthogonal frequency division multiplexed (OFDM) signals have cyclostationary components. Publications VI and VII use cyclostationary feature detection as pre-processing for signal classification.

As discussed above, many signal detection techniques recognize predefined patterns. Machine learning methods make it possible to learn signal patterns from training data (collected by sensing, simulation, or by other means) and then detect signals based on the learned patterns [50, 45, 23, 24]. In [50, 45, 24] neural networks are used for signal classification and in [50, 23] support vector machines are used. Using machine learning methods is attractive, because learning can adapt to different kinds of signal features. Furthermore, in the nanoscale spectrum sensing approach proposed in Publications VI and VII, it is imperative to learn to tolerate nanoscale faults.

5.1.2 Nanotechnology

Nanotechnology refers to the utilization of small structures or particles in the nanometer range. This section discusses nanocomputing or nanoelectronics [159], that is, computing with nanoscale components. Compared to traditional computing, nanocomputing offers not only components of smaller size, but also completely new kinds of components. A prominent component example is graphene [52], which has different kinds of phys-

ical properties compared to traditional electronic components. Research exists on nanocomponents such as a crossing mesh of nano wires [70], single carbon nanotubes [127], ultrathin films of carbon nanotubes [30], or nanoscale integrators [167]. Research also exists on devices with advanced functionality based on simple nanocomponents such as a carbon nanotube based transistor radio [68]. Also nanomaterials for harvesting energy for sensors and other devices has been studied [161]. The recent review [110] describes new kinds of components based on nanotechnology, which are not used in traditional computing devices. Furthermore, as reviewed in [109] nanoelectromechanical systems (NEMS) have several advantages over microelectromechanical systems (MEMS) for implementing low power radio devices.

Even though research on individual nanocomponents exists, complete systems, where several nanocomponents are connected and used together, are scarce. There is recent research based on simulating circuits with measured nanocomponent models. Cantley et al. [28, 29] use models of nano-crystalline silicon transistors and memristors in simulations of a neural circuit with few neurons. Lee et al. [82] use abstract simulations of a hybrid complementary metal–oxide–semiconductor (CMOS)/nanodevice circuit to evaluate the pattern recognition ability of the circuit. This thesis proposes a nano-scale approach for classifying wireless signals. The approach was investigated using simulations without detailed physical modeling of nanocomponents and their interaction, which could make simulations more accurate.

5.1.3 Fault tolerance

Fault tolerance of machine learning methods, especially neural networks and radial basis function networks (RBFn), has been an active research area for several decades. The structure of both neural and RBF networks works well in fault tolerance, because both contain parallel computing elements. One motivation for fault tolerance research is analog circuit implementation. Especially for smaller scale analog circuits, such as nanocomponent circuits, thermal noise and structural faults influence circuit operation heavily. Previously studied neural network and RBFn fault types include noise on network parameters [99, 108, 43] and structural faults that fix neurons or parameters to certain values [130, 37, 128, 173, 83].

One important result of previous work is that a neural network can be trained to tolerate faults during both training and evaluation [130,

37, 128]. Furthermore, training for fault tolerance may even help the network to generalize better [111]. For more background information on fault tolerant neural networks, the reader is referred to Section 2.3 of Publication VII.

5.2 Nanoscale spectrum sensing based on fault tolerant RBFn

This section discusses the nanoscale spectrum sensing design presented in Publications VI and VII. For classification, the design uses an RBFn, which is trained to tolerate nanoscale faults. The system is partially abstract, but as discussed in Section 3.5 of Publication VII research exists on many of the nanocomponents that the system requires. For instance, the carbon nanotube based transistor radio in [68] illustrates how radio signals can be captured by new kinds of nanocomponents.

The main idea in the new spectrum sensing approach is to use nanosensors to capture incoming radio signals. The power in the incoming signals drives the classification system. In the classification system, cyclostationary feature extractation turns the incoming radio signal into features that are classified by a radial basis function network (RBFn) as either "channel free" or "channel occupied". These different parts of the system have been selected so that they can be implemented in the future as a physical nanoscale circuit. The RBFn is not only a good generic classifier, but the RBFn can also be trained to tolerate faults. Fault tolerance is necessary, because thermal noise and structural faults such as broken or displaced wires are expected to be common in the nanoscale implementation. Taking displaced wire faults into account is an additional novel contribution of our work. Furthermore, because signal power decreases as the signal propagates through the nanoscale circuit, the effect of thermal noise increases farther away from the incoming signal. Figure 5.1 shows an overview of the approach. Next, this thesis discusses in more detail cyclostationary feature extraction, RBFn classification, and training of the RBFn to tolerate faults.

Cyclostationary feature extraction. Cyclostationary [51] feature extraction extracts periodic correlations in a time series, where the correlations can be, for example, statistical means or variances. Many common radio signals such as OFDM signals have cyclostationary features. Furthermore, cyclostationary feature extraction separates additive white Gaus-

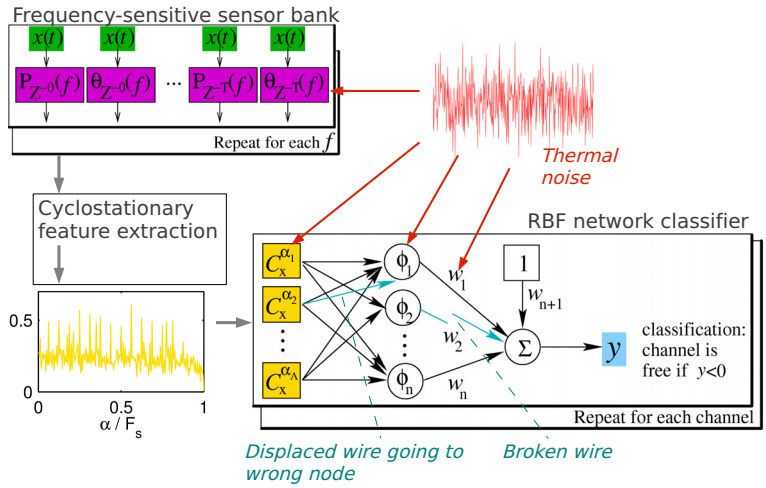


Figure 5.1. Overview of analog passive nanoscale spectrum sensing circuit design proposed in Publication VI. A nanoscale sensor bank captures signals on a frequency channel with a wide bandwidth. The system inputs the amplitude and phase of captured signals for varying delays into a cyclostationary feature extraction circuit. A radial basis function network (RBFn) classifies the frequency channel as either free or occupied based on the cyclostationary features. Thermal noise perturbs signals temporarily and displaced and broken wires cause permanent faults. The RBFn is trained to take all these possible faults into account, when performing the classification. Note that the signal strength deteriorates as the signal propogates through the nanoscale circuit and therefore the relative thermal noise strength becomes stronger farther away from the sensor bank that drives the circuit.

sian noise (AWGN) very well from signals, because AWGN does not contain any cyclostationary frequencies. Therefore, a system that uses cyclostationarity to detect signals finds signals even when the signal-to-noise ratio (SNR) is very low. The system in Publication VI extracts cyclostationary features from spectral correlations. Spectral correlation can be defined (see [51] for details) as the average correlation of two frequency components, centered at f and separated by α, over time. Publication VI computes final features for a selected set of α values. Each α-feature is the maximum over absolute normalized spectral correlations. The spectral correlation computations in Publication VI require enough frequency sensors (and possibly other components) to cover a sufficient range of frequencies, which should not be a problem, because sensors are in the nanoscale.

Radial basis function network. The nanoscale system uses an RBFn to classify the cyclostationary features into “channel free” and “channel occupied”. A RBFn is essentially a weighted sum of non-linear functions. In

Publication VI the RBFn is a weighted sum of Gaussian functions with weights w_i and a bias term with weight w_{n+1}:

$$y = \sum_{i=1}^{n} w_i e^{-||\vec{x}-\vec{c}_i||^2/(2\sigma_i^2)} + w_{n+1} \ , \tag{5.1}$$

where each Gaussian has a width parameter σ_i and a center parameter $\vec{c}_i$, which correspond to the standard deviation and mean for Gaussian probability distributions.

Fault tolerance. The nanoscale system described above is evaluated in simulations. The simulations include a fault model that models thermal noise and structural faults, which are expected to occur in the nanoscale implementation (the expected implementation is discussed in more detail in Publications VI and VII with the help of existing nanoscale component examples). In the simulations, the fault model adds noise to the frequency sensors, the RBFn spreads, centers, Gaussian function outputs, and weights. Because the nanoscale system is designed to operate without external power, the wireless signal attenuates farther away from the frequency sensors. Moreover, because of this signal attenuation the relative noise power increases farther away from the frequency sensors. In addition to noise, the fault model also models displaced and broken wires in the RBFn.

As discussed in Section 5.1.3 inserting faults during training improves the fault tolerance of an RBFn. In Publication VI faults, similar to the ones expected to occur during evaluation, are inserted during the training. A gradient ascent method is used to improve RBFn weight, spread, and center parameters.

Experimental results. In the experiments, faults were divided into three categories: feature extraction noise, RBFn noise, and RBFn structural faults. The fault level of each fault type was varied while other fault types were held at a default level. The experiments showed that when the structural fault level is increased to a very high value, it decreases performance heavily. At the expected default fault level the system performed well.

5.2.1 Improvements to the spectrum sensing approach

Publication VII improves on the work in Publication VI. Publication VII discusses the possible nanoscale implementation in more detail and makes

several crucial changes to the proposed nanoscale architecture that improve classification accuracy. Publication VII improves also the fault model and experimental setup and includes several new comparisons.

In more detail, Publication VII changes the nanoscale architecture. It removes normalization from cyclostationary feature extraction, which increases performance significantly. Publication VII also removes the bias term from the RBFn, because the bias term is very sensitive to faults. In addition to the faults in Publication VI the fault model in Publication VII includes noise and structural faults inside the feature extraction circuit. The experiments included new structural variability evaluations and a comparison of the RBFn classifier to a support vector machine (SVM) [38] classifier. Experiments showed that even without faults the RBFn yields equal performance compared with an SVM classifier.

5.2.2 Summary

This section discussed the nanoscale fault tolerant spectrum sensing approach in Publications VI and VII. Publications VI and VII give as much details of the expected nanoscale implementation as is possible, but, at the same time, stress that more research is needed on nanoscale components for an actual implementation. The proposed spectrum sensing system proposes a solution to the problems of nanoscale fault tolerance and signal sensing in different SNR regimes under the expected conditions, and is an important step in making actual low power cognitive radios operating over a wide range of frequencies a reality.

6. Summary and future work

To summarize, this thesis presented new models and techniques for cognitive radio and wireless channel access that yield significant improvements over previous approaches, a new low power wide bandwidth nanoscale spectrum sensing approach, and new methods for partially observable Markov decision processes (POMDPs) and decentralized POMDPs (DEC-POMDPs) that in experiments solve larger problems and compute larger policies more efficiently than previous methods, in both wireless network applications and in benchmark problems.

In addition to new generic methods for POMDPs and DEC-POMDPs, this thesis presented solutions for wireless networking problems. In particular, in Publications IV and V the behavior of single or several wireless agents that transmit data on shared channels is optimized. Similarly to other real-world problems, in these wireless networking problems it is uncertain how the world evolves in response to agent actions. Furthermore, as in most real-world problems agents make noisy observations restricted to only parts of the operating environment. For solving these kind of problems, the thesis discussed the general decision making frameworks of POMDPs for single agents and DEC-POMDPs for multiple co-operative agents. In the past POMDPs and DEC-POMDPs have been used successfully for solving problems in many diverse application domains, such as wireless networking, robotics, elder care, tiger conservation, and manufacturing. However, solving POMDPs and DEC-POMDPs is computationally very challenging even for small problems and especially for large complicated problems such as the wireless networking problems studied in Publications IV and V.

All POMDP and DEC-POMDP methods are limited in practice by problem size in some way or another. For instance, in wireless applications the problem size can grow exponentially with the number of frequencies and

agents. Many POMDP problems can be described compactly in a factored form that fits into computer memory, but the factored specification does not alone solve the problem of efficient policy computation. Publication I presented a new factored POMDP method that can efficiently compute policies for large problems, such as the cognitive radio problem in Publication IV. In comparisons with state-of-the-art POMDP solvers it yielded good results in smaller problems and could compute policies for larger problems than the comparison methods.

In contrast to the single agent cognitive radio problem in Publication IV, in the wireless channel access problem in Publication V the behavior of multiple wireless agents is optimized. Wireless agents can be assumed to operate for indefinitely long periods of time and thus an infinite-horizon approach is required. Because there were no previous suitable infinite-horizon methods, Publication II introduced the first general factored infinite-horizon DEC-POMDP method. In experimental comparisons with non-factored DEC-POMDP methods in Publication II, the new method yielded equal performance in small benchmark problems. The new method computed policies for benchmark problems with larger state spaces and more agents than the comparison methods could. Moreover, the wireless approach in Publication V utilized the new method for optimizing wireless channel access policies yielding better results than wireless comparison methods, which were optimized for the same radio environment.

In addition to problem specification size, in problems with a long optimization horizon the size of agent policies restricts performance. As a solution, Publication III introduces periodic finite state controllers (FSCs) that allow optimization of much larger controllers and new kinds of policy optimization methods. In experiments the new methods performed better than state-of-the-art DEC-POMDP methods and better than restricted policy size POMDP methods. Furthermore, because the factored infinite-horizon DEC-POMDP method in Publication II is based on expectation maximization (EM), it was straightforward to use periodic EM introduced in Publication III for optimizing periodic controllers for wireless agents in Publication V.

In the context of wireless networking, the thesis discussed both wireless channel access and spectrum sensing. One central problem in wireless channel access is that transmissions of a wireless agent interfere with transmissions of other wireless agents. The thesis discussed how one can

try to avoid interference in the temporal, frequency, or spatial dimensions. A widely researched cognitive radio problem is that of finding a frequency channel that a primary user is not currently transmitting on so that a secondary user can temporarily transmit on the channel. Publication IV presented a more complete novel POMDP model than previous work, that takes into account varying length idle and packet bursts and also primary user reactions, on the frequency channels. In the experiments in Publication IV, the approach successfully outperformed comparison methods.

Research on wireless channel access often relies on tuning parameters of existing protocols, which have been designed by wireless experts relying on certain assumptions about the operating environment. Finding high performance policies automatically is computationally difficult: because of uncertain traffic patterns and agents making individual observations and actions, a computationally complex model for planning such as a DEC-POMDP is needed to take the crucial properties of channel access into account. However, transmissions of a wireless agent influence only wireless agents spatially close enough allowing a factored problem specification. Therefore, Publication V formulates channel access as a factored infinite-horizon DEC-POMDP. The solution to the DEC-POMDP yields a different channel access policy for each wireless agent optimized for the current operating environment. In order to optimize wireless agent policies in practice, Publication V utilized the factored infinite-horizon DEC-POMDP method presented in Publication II and periodic policies introduced in Publication III. In experiments, the approach outperformed optimized wireless protocols, indicating that a DEC-POMDP based approach for channel access is a viable solution.

For the purpose of good decision making, sensing is of paramount importance. For instance, if a wireless agent senses a channel as idle, when it is occupied, it may transmit and transmissions collide. If the agent thinks the channel is occupied when actually it is not, a transmission opportunity is lost. In addition to sensing accuracy, power usage and the range of frequencies are critical properties of wireless sensors. Publications VI and VII propose a new passive nanoscale spectrum sensing approach for cognitive radio that is intended to solve the low power and wide frequency range problems together with good sensing accuracy. However, a major problem in nanoscale designs are nanoscale faults. As a solution, the new approach used an RBFn to classify signals with good accuracy, while at the same time the RBFn was trained to tolerate nanoscale faults. The

proposed system had good performance even with a high level of faults.

To recapitulate, the thesis presented new efficient methods for computing policies for very large POMDP problems and for DEC-POMDP problems with large state spaces and many agents. The factored POMDP method in Publication I yielded in comparisons with state-of-the-art POMDP solvers good results in smaller problems and could solve larger problems than the comparison methods. The first general factored infinite-horizon DEC-POMDP method in Publication II yielded in experiments equal performance in small problems and could solve much larger problems than non-factored DEC-POMDP comparison methods. Publication III discussed a new periodic controller approach that enables larger controllers and new kinds of algorithms for POMDPs and DEC-POMDPs. In experiments, periodic controllers performed better than state-of-the-art DEC-POMDP methods or restricted policy size POMDP methods. In addition to new POMDP and DEC-POMDP methods, the thesis presented solutions for wireless networking problems. In Publication IV, a novel comprehensive POMDP model for cognitive radios performed in experiments significantly better than simpler comparison methods. Publication V showed how to formulate wireless channel access as a DEC-POMDP and how to adapt the methods from Publications II and III for computing wireless DEC-POMDP policies efficiently. In experiments, the DEC-POMDP approach outperformed optimized wireless channel access protocols. Lastly, Publications VI and VII discussed a new nanoscale spectrum classifier design, which uses a RBFn that is trained to tolerate nanoscale faults. In experiments, the system classified signals with good accuracy even under high fault levels.

6.1 Future work

There is ample room for further work based on the contributions presented in this thesis. For the nanoscale spectrum sensing approach, the obvious next step is implementation of necessary nanocomponents and construction of a working prototype. The first step towards this goal is to perform circuit simulations using detailed physical models of nanocomponents.

Regarding wireless experiments, both the cognitive radio POMDP approach in Publication IV and the wireless factored DEC-POMDP approach in Publication V were evaluated using computer simulations. The simu-

lations could be made more detailed by utilizing additional information, such as mobility models [67] or other kinds of topology information [91]. However, the ultimate test for these approaches is to implement them in actual wireless devices. Another line of potential new work would be to make the wireless factored DEC-POMDP approach in Publication V operate on multiple frequency channels. This would allow optimization over all the three dimensions of time, frequency, and space.

The cognitive radio application in Publication IV requires the primary user models to be estimated before a POMDP policy for the cognitive radio can be computed. Therefore a POMDP method which would learn primary user models during online operation would be beneficial. There exists research about active learning in POMDPs [41, 117, 125], but existing approaches do not seem to be able to solve very large problems, such as the cognitive radio problem. The wireless factored DEC-POMDP channel access application in Publication V would similarly benefit from active learning. The learning approach should take traffic pattern changes, topology changes, and moving wireless devices into account.

In addition to research on active learning in POMDP and DEC-POMDP methods, improvements to the actual optimization goal should be considered. The optimization goal in real-world wireless networking problems is usually average throughput or average delay. The discounted reward optimization goal in POMDP and DEC-POMDP methods eases algorithm design, because it emphasizes rewards near the starting probability distribution more, but it also does not fully correspond to the actual optimization criterion. In future work, the average reward optimization criterion in (factored) POMDP and DEC-POMDP methods should be investigated.

POMDPs and DEC-POMDPs can be used to solve many real world problems, such as problems in wireless networking or robotics. However, because of their large size, practical problems often require the use of approximations. Further work should investigate improvements to approximation techniques in POMDPs and DEC-POMDPs. In particular, even more efficient approximation methods with analytic error bounds would be useful.

Finally, the developed POMDP and DEC-POMDP methods were applied in this thesis on benchmark problems and on the discussed wireless networking problems. In the future, the methods should be tested in other application fields such as robotics or the computer games domain.

Bibliography

[1] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. URL http://standards.ieee.org/getieee802/download/802.11-2007.pdf. 2007 revision.

[2] IEEE 802.16 Task Group d, . URL http://www.ieee802.org/16/tgd/. Retrieved 2008-03-12.

[3] IEEE 802.16e Task Group (Mobile WirelessMAN), . URL http://www.ieee802.org/16/tge/. Retrieved 2008-03-12.

[4] S. H. A. Ahmad, M. Liu, T. Javidi, Q. Zhao, and B. Krishnamachari. Optimality of myopic sensing in multichannel opportunistic access. *IEEE Transactions on Information Theory*, 55(9):4040–4050, 2009.

[5] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50:2127–2159, 2006.

[6] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication*, 2010.

[7] B. Alawieh, Y. Zhang, C. Assi, and H. Mouftah. Improving Spatial Reuse in Multihop Wireless Networks-A Survey. *IEEE Communications Surveys & Tutorials*, 11(3):71–91, 2009.

[8] M. Allen and S. Zilberstein. Complexity of decentralized control: Special cases. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 19–27. Curran Associates Inc., 2010.

[9] C. Amato, D. S. Bernstein, and S. Zilberstein. Optimizing Memory-Bounded Controllers for Decentralized POMDPs. In *Proceedings of the 23rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1–8. AUAI Press, 2007.

[10] C. Amato, J. S. Dibangoye, and S. Zilberstein. Incremental policy generation for finite-horizon DEC-POMDPs. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 2–9. AAAI Press, 2009.

[11] C. Amato, D. S. Bernstein, and S. Zilberstein. Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *Autonomous Agents and Multi-Agent Systems*, 21(3):293–320, 2010.

[12] C. Amato, B. Bonet, and S. Zilberstein. Finite-state controllers based on Mealy machines for centralized and decentralized POMDPs. In *Proceedings of the Twenty-Fourth National Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2010.

[13] A. Arapostathis, V. S. Borkar, E. Fernández-Gaucherand, M. K. Ghosh, and S. I. Marcus. Discrete-time controlled Markov processes with average cost criterion: a survey. *SIAM Journal on Control and Optimization*, 31 (2):282–344, 1993.

[14] H. Bai, D. Hsu, W. Lee, and V. Ngo. Monte Carlo value iteration for continuous-state POMDPs. *Algorithmic Foundations of Robotics IX*, pages 175–191, 2011.

[15] R. Becker, S. Zilberstein, and C. V. Goldman. Solving transition independent decentralized Markov decision processes. In *Journal of Artificial Intelligence Research*, volume 22, pages 423–455. AAAI Press, 2004.

[16] R. Bellman. A Markovian Decision Process. *Journal of Mathematics and Mechanics*, 6:679–684, 1957.

[17] D. S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of Markov decision processes. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 32–37. Morgan Kaufmann, 2000.

[18] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research*, pages 819–840, 2002.

[19] D. S. Bernstein, E. A. Hansen, and S. Zilberstein. Bounded policy iteration for decentralized POMDPs. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1287–1292. International Joint Conferences on Artificial Intelligence, 2005.

[20] D. S. Bernstein, C. Amato, E. A. Hansen, and S. Zilberstein. Policy iteration for decentralized control of Markov decision processes. *Journal of Artificial Intelligence Research*, 34(1):89–132, 2009.

[21] D. P Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, 1995.

[22] J. C. Bicket. Bit-rate selection in wireless networks, 2005.

[23] L. Bixio, G. Oliveri, M. Ottonello, and C. S. Regazzoni. OFDM recognition based on cyclostationary analysis in an Open Spectrum scenario. In *Proceedings of 69th IEEE Vehicular Technology Conference (VTC)*. IEEE, 2009.

[24] L. Bixio, M. Ottonello, H. Sallam, M. Raffetto, and C. S. Regazzoni. Signal Classification based on Spectral Redundancy and Neural Network Ensembles. In *Proceedings of 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*. IEEE, 2009.

[25] B. Bonet and H. Geffner. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1641–1646. AAAI Press, 2009.

[26] C. Boutilier and D. Poole. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*, pages 1168–1175. AAAI Press, 1996.

[27] X. Boyen and D. Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Inttelligence (UAI)*, pages 33–42. Morgan Kaufmann, 1998.

[28] K. D. Cantley, A. Subramaniam, H. J. Stiegler, R. A. Chapman, and E. M. Vogel. Hebbian Learning in Spiking Neural Networks With Nanocrystalline Silicon TFTs and Memristive Synapses. *IEEE Transactions on Nanotechnology*, 10(5):1066–1073, 2011.

[29] K. D. Cantley, A. Subramaniam, H. J. Stiegler, R. A. Chapman, and E. M. Vogel. Neural Learning Circuits Utilizing Nano-Crystalline Silicon Transistors and Memristors. *IEEE Transactions on Neural Networks and Learning Systems*, 23(4):565–573, 2012.

[30] Q. Cao and J. A. Rogers. Ultrathin Films of Single-Walled Carbon Nanotubes for Electronics and Sensors: A Review of Fundamental and Applied Aspects. *Advanced Materials*, 21(1), 2009.

[31] A. R. Cassandra. A Survey of POMDP Applications. Technical report, Austin, USA, 1998. Presented at the AAAI Fall Symposium 1998.

[32] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 963–972. IEEE, 1996.

[33] A. R. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 54–61. Morgan Kaufmann, 1997.

[34] I. Chadès, E. McDonald-Madden, M. A. McCarthy, B. Wintle, M. Linkie, and H. P. Possingham. When to stop managing or surveying cryptic threatened species. *Proceedings of the National Academy of Sciences (PNAS)*, 105 (37):13936–13940, 2008.

[35] Y. Chen, Q. Zhao, and A. Swami. Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors. *IEEE Transactions on Information Theory*, 54(5):2053–2071, 2008.

[36] J. Choi, J. Yoo, S. Choi, and C. Kim. EBA: an enhancement of the IEEE 802.11 DCF via distributed reservation. *IEEE Transactions on Mobile Computing*, 4(4):378–390, 2005.

[37] R. D. Clay and C. H. Sequin. Fault tolerance training improves generalization and robustness. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, volume 1, pages 769–774. IEEE, 1992.

[38] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20 (3):273–297, 1995.

[39] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[40] D. J. Deng and R. S. Chang. A Priority Scheme for IEEE 802.11 DCF Access Method. *IEICE transactions on communications*, 82(1):96–102, 1999.

[41] F. Doshi, J. Pineau, and N. Roy. Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs. In *Proceedings of the 25th International Conference on Machine learning (ICML)*, pages 256–263. ACM, 2008.

[42] P. Doshi, Y. Zeng, and Q. Chen. Graphical models for interactive POMDPs: Representations and solutions. *Autonomous agents and multi-agent systems*, 18(3):376–416, 2009.

[43] R. Eickhoff and U. Rückert. Robustness of radial basis functions. *Neurocomputing*, 70(16–18):2758–2767, 2007.

[44] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 1, pages 136–143. ACM, 2004.

[45] A. Fehske, J. Gaeddert, and J. H. Reed. A new approach to signal classification using spectral correlation and neural networks. In *Proceedings of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 144–150. IEEE, 2005.

[46] S. Filippi, O. Cappé, F. Clérot, and E. Moulines. A near optimal policy for channel allocation in cognitive radio. *Recent Advances in Reinforcement Learning*, pages 69–81, 2008.

[47] J. A. Fuemmeler, N. H. Vaidya, and V. V. Veeravalli. Selecting Transmit Powers and Carrier Sense Thresholds for CSMA Protocols. Technical report, University of Illinois at Urbana-Champaign, 2004.

[48] J. A. Fuemmeler, N. H. Vaidya, and V. V. Veeravalli. Selecting transmit powers and carrier sense thresholds in csma protocols for wireless ad hoc networks. In *Proceedings of the 2nd Annual International Workshop on Wireless Internet (WICON)*, WICON '06. ACM, 2006.

[49] A. Galindo-Serrano and L. Giupponi. Distributed Q-learning for aggregated interference control in cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 59(4):1823–1834, 2010.

[50] M. Gandetto, M. Guainazzo, and C. S. Regazzoni. Use of time-frequency analysis and neural networks for mode identification in a wireless software-defined radio approach. *EURASIP Journal on Applied Signal Processing*, 2004:1778–1790, 2004.

[51] W. A. Gardner, A. Napolitano, and L. Paura. Cyclostationarity: half a century of research. *Signal Processing*, 86(4):639–697, 2006.

[52] A. K. Geim and K. S. Novoselov. The rise of graphene. *Nature Materials*, 6:183–191, 2007.

[53] S. Geirhofer, L. Tong, and B. M. Sadler. Cognitive medium access: constraining interference based on experimental models. *IEEE Journal on Selected Areas of Communication*, 26(1):95–105, 2008.

[54] M. Ghallab, D. S. Nau, and P. Traverso. *Automated Planning: theory and practice*. Morgan Kaufmann, 2004.

[55] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of Artificial Intelligence Research*, 24(1):49–79, 2005.

[56] C. V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 137–144. ACM, 2003.

[57] C. Guestrin, D. Koller, and R. Parr. Solving factored POMDPs with linear value functions. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence Workshop on Planning under Uncertainty and Incomplete Information*, pages 67–75, 2001.

[58] E. A. Hansen. An improved policy iteration algorithm for partially observable MDPs. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10 (NIPS)*. MIT Press, 1998.

[59] E. A. Hansen and Z. Feng. Dynamic programming for POMDPs using a factored state representation. In *Proceedings of the Fifth International Conference on AI Planning Systems*, pages 130–139. AAAI Press, 2000.

[60] E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 709–715. AAAI Press, 2004.

[61] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communication*, 23:201–220, 2005.

[62] J. Hoey, R. St-Aubin, A. Hu, and C. Boutilier. Spudd: Stochastic planning using decision diagrams. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 279–288. Morgan Kaufmann, 1999.

[63] J. Hoey, A. Von Bertoldi, P. Poupart, and A. Mihailidis. Assisting persons with dementia during handwashing using a partially observable Markov decision process. In *Proceedings of the 5th International Conference on Vision Systems (IVCS)*. Bielefeld University Library, 2007.

[64] R. A. Howard. *Dynamic programming and Markov processes*. MIT press, 1960.

[65] D. Hsu, W. S. Lee, and N. Rong. What makes some POMDP problems easy to approximate? In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 689–696. Curran Associates Inc., 2008.

[66] G. Hwang and S. Roy. Design and analysis of optimal random access policies in cognitive radio networks. *IEEE Transactions on Communications*, (99):1–11, 2012.

[67] E. Hyytiä, P. Lassila, and J. Virtamo. Spatial node distribution of the random waypoint mobility model with applications. *IEEE Transactions on Mobile Computing*, 5(6):680–694, 2006.

[68] K. Jensen, J. Weldon, H. Garcia, and A. Zettl. Nanotube Radio. *Nano Letters*, 7:3508–3511, 2007.

[69] S. Ji, R. Parr, H. Li, X. Liao, and L. Carin. Point-based policy iteration. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI)*, volume 22, pages 1243–1249. AAAI Press, 2007.

[70] G.-Y. Jung, E. Johnston-Halperin, W. Wu, Z. Yu, S.-Y. Wang, W. M. Tong, Z. Li, J. E. Green, B. A. Sheriff, A. Boukai, Y. Bunimovich, J. R. Heath, and R. S. Williams. Circuit fabrication at 17 nm half-pitch by nanoimprint lithography. *Nano Letters*, 6:351–354, 2006.

[71] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[72] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2): 99–134, 1998.

[73] M. Kaynia, N. Jindal, and G. E. Oien. Improving the Performance of Wireless Ad Hoc Networks Through MAC Layer Design. *IEEE Transactions on Wireless Communications*, 10(1):240–252, 2011.

[74] T. S. Kim, H. Lim, and J. C. Hou. Understanding and Improving the Spatial Reuse in Multihop Wireless Networks. *IEEE Transactions on Mobile Computing*, pages 1200–1212, 2008.

[75] L. Kleinrock and J. Silvester. Spatial Reuse in Multihop Packet Radio Networks. *Proceedings of the IEEE*, 75(1):156–167, 1987.

[76] Y. B. Ko, J. M. Choi, and N. H. Vaidya. MAC protocols using directional antennas in IEEE 802.11 based ad hoc networks. *Wireless Communications and Mobile Computing*, 8(6):783–795, 2008.

[77] T. Korakis, G. Jakllari, and L. Tassiulas. A MAC protocol for full exploitation of Directional Antennas in Ad-hoc Wireless Networks. In *Proceedings of ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, pages 98–107. ACM, 2003.

[78] A. Kumar and S. Zilberstein. Point-based backup for decentralized POMDPs: Complexity and new algorithms. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1315–1322. IFAAMAS, 2010.

[79] A. Kumar and S. Zilberstein. Anytime planning for decentralized POMDPs using Expectation Maximization. In *Proceedings of the Twenty-Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 294–301. AUAI Press, 2010.

[80] A. Kumar, S. Zilberstein, and M. Toussaint. Scalable multiagent planning using probabilistic inference. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, pages 2140–2146. AAAI Press, 2011.

[81] H. Kurniawati, D. Hsu, and W. S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proceedings of Robotics: Science and Systems IV*, pages 65–72. MIT Press, 2008.

[82] J. H. Lee and K. K. Likharev. Defect-tolerant nanoelectronic pattern classifiers. *International Journal of Circuit Theory and Applications*, 35:239–264, 2007.

[83] C. S. Leung and J. P. F. Sum. A fault-tolerant regularizer for RBF networks. *IEEE Transactions on Neural Networks*, 19(3):493–507, 2008.

[84] H. Li. Multiagent Q-learning for aloha-like spectrum access in cognitive radio systems. *EURASIP Journal on Wireless Communications and Networking*, 2010:56, 2010.

[85] H. Li, M. Junfei, X. Fangmin, L. ShuRong, and Z. Zheng. Optimization of Collaborative Spectrum Sensing for Cognitive Radio. In *Proceedings of IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 1730–1733. IEEE, 2008.

[86] X. Li, W. Cheung, J. Liu, and Z. Wu. A novel orthogonal NMF-based belief compression for POMDPs. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 537–544. ACM, 2007.

[87] Z. W. Lim, D. Hsu, and L. Sun. Monte Carlo Value Iteration with Macro-Actions. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 1287–1295. Curran Associates Inc., 2012.

[88] M. L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances In Neural Information Processing Systems 14 (NIPS)*, pages 1555–1561. MIT Press, 2002.

[89] K. Liu and Q. Zhao. Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11):5547–5567, 2010.

[90] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI)*, pages 541–548. AAAI Press, 1999.

[91] P. Mähönen, M. Petrova, and J. Riihijarvi. Applications of Topology Information for Cognitive Radios and Networks. In *Proceedings of 2nd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 103–114. IEEE, 2007.

[92] D. McAllester and S. Singh. Approximate planning for factored POMDPs using belief state simplification. In *Proceedings of the Fifteenth Annual*

Conference on Uncertainty in Artificial Inttelligence (UAI), pages 409–417. Morgan Kaufmann, 1999.

[93] S. McCanne and S. Floyd. ns network simulator. URL `http://www.isi.edu/nsnam/ns/`.

[94] K. Medepalli and F. A. Tobagi. On Optimization of CSMA/CA based Wireless LANs: Part I - Impact of Exponential Backoff. In *Proceedings of IEEE International Conference on Communications (ICC)*, volume 5, pages 2089–2094. IEEE, 2006.

[95] J. V. Messias, M. T. J. Spaan, and P. U. Lima. Efficient offline communication policies for factored multiagent POMDPs. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 1917–1925. Curran Associates Inc., 2012.

[96] N. Meuleau, L. Peshkin, K. E. Kim, and L. P. Kaelbling. Learning finite-state controllers for partially observable environments. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 427–436. Morgan Kaufmann, 1999.

[97] J. Mitola III and G. Q. Maguire Jr. Cognitive radio: making software radios more personal. *Personal Communications, IEEE*, 6(4):13–18, 1999.

[98] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, 2002.

[99] A. F. Murray and P. J. Edwards. Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training. *IEEE Transactions on Neural Networks*, 5(5):792–802, 1994.

[100] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: a synthesis of distributed constraint optimization and POMDPs. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)*, volume 1, pages 133–139. AAAI Press, 2005.

[101] P. Nicopolitidis, G. I. Papadimitriou, and A. S. Pomportsis. Learning automata-based polling protocols for wireless LANs. *IEEE Transactions on Communications*, 51(3):453–463, 2003.

[102] F. A. Oliehoek. *Value-Based Planning for Teams of Agents in Stochastic Partially Observable Environments*. PhD thesis, Informatics Institute, University of Amsterdam, Feb 2010.

[103] F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored DEC-POMDPs. In *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 1, pages 517–524. IFAAMAS, 2008.

[104] F. A. Oliehoek, S. Whiteson, and M. T. J. Spaan. Lossless clustering of histories in decentralized POMDPs. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 1, pages 577–584. IFAAMAS, 2009.

[105] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of operations research*, pages 441–450, 1987.

[106] S. Paquet, L. Tobin, and B. Chaib-draa. An online POMDP algorithm for complex multiagent environments. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent systems (AAMAS)*, pages 970–977. ACM, 2005.

[107] J. S. Park, A. Nandan, M. Gerla, and H. Lee. SPACE-MAC: Enabling spatial reuse using MIMO channel-aware MAC. In *Proceedings of IEEE International Conference on Communications (ICC)*, volume 5, pages 3642–3646. IEEE, 2005.

[108] X. Parra and A. Català. Learning fault-tolerance in radial basis function networks. In *Proceedings of 9th European Symposium on Artificial Neural Networks (ESANN)*, pages 341–346, 2001. URL `http://www.dice.ucl.ac.be/esann/proceedings/electronicproceedings.htm`.

[109] A. Pärssinen, R. Kaunisto, and A. Kärkkäinen. *Nanotechnologies for Future Mobile Devices*, chapter Future of Radio and Communication. Cambridge University Press, 2010.

[110] P. Pasanen, M. A. Uusitalo, V. Ermolov, J. Kivioja, and C. Gamrat. *Nanotechnologies for Future Mobile Devices*, chapter Computing and Information Storage Solutions. Cambridge University Press, 2010.

[111] D. S. Phatak. Relationship between fault tolerance, generalization and the Vapnik-Chervonenkis (VC) dimension of feedforward ANNs. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, volume 1. IEEE, 1999.

[112] J. Pineau, G. Gordon, and S. Thrun. PBVI: An anytime algorithm for POMDPs. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025–1032. International Joint Conferences on Artificial Intelligence, 2003.

[113] P. Poupart. *Exploiting structure to efficiently solve large scale partially observable Markov decision processes*. PhD thesis, Univ. of Toronto, Toronto, Canada, 2005.

[114] P. Poupart and C. Boutilier. Value-directed compression of POMDPs. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15 (NIPS)*, pages 1547–1554. MIT Press, 2003.

[115] P. Poupart and C. Boutilier. Bounded finite state controllers. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 (NIPS)*, pages 823–830. MIT Press, 2004.

[116] P. Poupart and C. Boutilier. VDCBPI: an approximate scalable algorithm for large POMDPs. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17 (NIPS)*, pages 1081–1088. MIT Press, 2005.

[117] P. Poupart and N. Vlassis. Model-based Bayesian reinforcement learning in partially observable domains. In *Proceedings of the Tenth International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2008.

[118] J. G. Proakis. *Digital communications*, volume 1221. McGraw-hill, 1987.

[119] D. V. Pynadath and M. Tambe. Multiagent teamwork: Analyzing the optimality and complexity of key theories and models. In *Proceedings of the first International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 2, pages 873–880. ACM, 2002.

[120] D. V. Pynadath and M. Tambe. The communicative multiagent team decision problem: analyzing teamwork theories and models. *Journal of Artificial Intelligence Research*, 16(1):389–423, 2002.

[121] L. Rabiner and B. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[122] S. Ross and B. Chaib-draa. AEMS: An anytime online search algorithm for approximate policy refinement in large POMDPs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2592—2598. International Joint Conferences on Artificial Intelligence, 2007.

[123] S. Ross, J. Pineau, and B. Chaib-draa. Theoretical analysis of heuristic search methods for online POMDPs. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 1233–1240. Curran Associates Inc., 2008.

[124] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32(1): 663–704, 2008.

[125] S. Ross, J. Pineau, B. Chaib-Draa, and P. Kreitmann. A Bayesian approach for learning and planning in partially observable Markov decision processes. *Journal of Machine Learning Research*, 12(May):1729–1770, 2011.

[126] N. Roy, G. J. Gordon, and S. Thrun. Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research*, 23: 1–40, 2005.

[127] V. Sazonova, Y. Yaish, H. Üstüunel, D. Roundy, T. A. Arias, and P. L. McEuen. A Tunable Carbon Nanotube Electromechanical Oscillator. *Nature*, 431:284–287, 2004.

[128] B. E. Segee and M. J. Carter. Comparative fault tolerance of parallel distributed processing networks. *IEEE Transactions on Computers*, 43(11): 1323–1329, 1994.

[129] Y. Selén, H. Tullberg, and J. Kronander. Sensor Selection for Cooperative Spectrum Sensing. In *Proceedings of 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 1–11, 2008.

[130] C. H. Sequin and R. D. Clay. Fault-tolerance in artificial neural networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, volume 1, pages 703–708. IEEE, 1990.

[131] S. Seuken and S. Zilberstein. Memory-bounded dynamic programming for DEC-POMDPs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2009–2016. International Joint Conferences on Artificial Intelligence, 2007.

[132] S. Seuken and S. Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Autonomous Agents and Multi-Agent Systems*, 17(2):190–250, 2008.

[133] G. Shani, R. Brafman, and S. Shimony. Adaptation for changing stochastic environments through online POMDP policy learning. In *Proceedings of the Workshop W9 on Reinforcement Learning in Nonstationary Environments, in conjunction with 16th ECML and 9th PKDD*, pages 61–70. Vrije Universiteit Brussel, 2005.

[134] G. Shani, R. I. Brafman, and S. E. Shimony. Forward search value iteration for POMDPs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2619–2624. International Joint Conferences on Artificial Intelligence, 2007.

[135] G. N. Shirazi, P.-Y. Kong, and C.-K. Tham. A Cooperative Retransmission Scheme in Wireless Networks with Imperfect Channel State Information. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2009.

[136] D. Silver and J. Veness. Monte-Carlo Planning in Large POMDPs. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2164–2172. Curran Associates Inc., 2011.

[137] H. S. Sim, K. E. Kim, J. H. Kim, D. S. Chang, and M. W. Koo. Symbolic heuristic search value iteration for factored POMDPs. In *Proceedings of the Twenty-Third National Conference on Artificial Intelligence (AAAI)*, pages 1088–1093. AAAI Press, 2008.

[138] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, pages 1071–1088, 1973.

[139] T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the Twentieth Annual Conference on Uncertainty in artificial intelligence (UAI)*, pages 520–527. AUAI Press, 2004.

[140] T. Smith and R. Simmons. Point-Based POMDP Algorithms: Improved Analysis and Implementation. In *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 542–549. AUAI Press, 2005.

[141] E. J. Sondik. *The optimal control of partially observable Markov processes*. PhD thesis, Stanford University, 1971.

[142] E. J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research*, 26(2): 282–304, 1978.

[143] M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[144] M. T. J. Spaan, F. A. Oliehoek, and C. Amato. Scaling up optimal heuristic search in DEC-POMDPs via incremental expansion. In *Proceedings of*

the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI). AAAI Press, 2011.

[145] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge University Press, 1998.

[146] D. Szer and F. Charpillet. An optimal best-first search algorithm for solving infinite horizon DEC-POMDPs. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML)*, pages 389–399. Springer, 2005.

[147] D. Szer and F. Charpillet. Point-based dynamic programming for DEC-POMDPs. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1233–1238. AAAI Press, 2006.

[148] D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A heuristic search algorithm for solving decentralized POMDPs. In *Proceedings of the Twenty-First Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 576–583. AUAI Press, 2005.

[149] H. Tang. Some physical layer issues of wide-band cognitive radio systems. In *Proceedings of IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 151–159. IEEE, 2005.

[150] C. Tekin and M. Liu. Online learning in opportunistic spectrum access: A restless bandit approach. In *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, pages 2462–2470. IEEE, 2011.

[151] S. Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[152] M. Toussaint, S. Harmeling, and A. Storkey. Probabilistic inference for solving (PO)MDPs. Technical report, University of Edinburgh, 2006.

[153] M. Toussaint, L. Charlin, and P. Poupart. Hierarchical POMDP Controller Optimization by Likelihood Maximization. In *Proceedings of the Twenty-Fourth Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 562–570. AUAI Press, 2008.

[154] J. Unnikrishnan and V. V. Veeravalli. Algorithms for dynamic spectrum access with learning for cognitive radio. *IEEE Transactions on Signal Processing*, 58(2):750–760, 2010.

[155] H. Urkowitz. Energy detection of unknown deterministic signals. *Proceedings of the IEEE*, 55(4):523–531, 1967.

[156] R. Van Nee, V. K. Jones, G. Awater, A. Van Zelst, J. Gardner, and G. Steele. The 802.11n MIMO-OFDM standard for wireless LAN and beyond. *Wireless Personal Communications*, 37(3):445–453, 2006.

[157] B. Wang and K. J. R. Liu. Advances in cognitive radio networks: A survey. *IEEE Journal of Selected Topics in Signal Processing*, 5(1):5–23, 2011.

[158] C. Wang, B. Li, and L. Li. A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF. *IEEE Transactions on Vehicular Technology*, 53(4):1235–1246, 2004.

[159] R. Waser (Ed.). *Nanoelectronics and Information Technology: Advanced Electronic Materials and Novel Devices*. Wiley, 2005.

[160] R. Washington. BI-POMDP: Bounded, incremental partially-observable Markov-model planning. In S. Steel and R. Alami, editors, *Recent Advances in AI Planning*, volume 1348 of *Lecture Notes in Computer Science*, pages 440–451. Springer, 1997.

[161] B. E. White. Energy-harvesting devices: Beyond the battery. *Nature Nanotechnology*, 3:71–72, 2008.

[162] S. J. Witwicki and E. H. Durfee. Influence-based policy abstraction for weakly-coupled DEC-POMDPs. In *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press, 2010.

[163] S. J. Witwicki, F. A. Oliehoek, and L. P. Kaelbling. Heuristic search of multiagent influence space. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 2, pages 973–980. IFAAMAS, 2012.

[164] C. Wu, K. Chowdhury, M. Di Felice, and W. Meleis. Spectrum Management of Cognitive Radio Using Multi-agent Reinforcement Learning. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS): Industry track*, pages 1705–1712. IFAAMAS, 2010.

[165] F. Wu, S. Zilberstein, and X. Chen. Point-based policy generation for decentralized POMDPs. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1307–1314. IFAAMAS, 2010.

[166] F. Wu, S. Zilberstein, and X. Chen. Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175(2):487–511, 2011.

[167] H. Ye, Z. Gu, T. Yu, and D. H. Gracias. Integrating nanowires with substrates using directed assembly and nanoscale soldering. *IEEE Transactions on Nanotechnology*, 5(1):62–66, 2006.

[168] H. Yu and D. P. Bertsekas. On near optimality of the set of finite-state controllers for average cost POMDP. *Mathematics of Operations Research*, 33(1):1–11, 2008.

[169] T. Yucek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys & Tutorials*, 11 (1):116–130, 2009.

[170] S. Zarrin and T. J. Lim. Belief Propagation on Factor Graphs for Cooperative Spectrum Sensing in Cognitive Radio. In *Proceedings of 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 1–9. IEEE, 2008.

[171] Q. Zhao and B. M. Sadler. A survey of dynamic spectrum access. *IEEE Signal Processing Magazine*, 24(3):79–89, 2007.

[172] Q. Zhao, L. Tong, A. Swami, and Y. Chen. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. *IEEE Journal on Selected Areas in Communication*, 25(3):589–600, Apr 2007.

[173] Z. H. Zhou, S. F. Chen, and Z. Q. Chen. Improving tolerance of neural networks against multi-node open fault. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 1687–1692. IEEE, 2001.

[174] J. Zhu, X. Guo, L. L. Yang, W. S. Conner, S. Roy, and M. M. Hazra. Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks. *Wireless Communications and Mobile Computing*, 4(8):933–946, 2004.

Errata

Publication II

In Figure 2 there should be arrows from a_2 to o_1 and from a_1 to o_2.

Publication III

Line 14 of Algorithm 1 is missing the sum over s', $\vec{a}$, $\vec{o}$, and $\vec{q}'$ after the equal sign "=".

DISSERTATIONS IN INFORMATION AND COMPUTER SCIENCE

Aalto-DD33/2012 Caldas, José
Graphical Models for Biclustering and Information Retrieval in Gene Expression Data. 2012.

Aalto-DD45/2012 Viitaniemi, Ville
Visual Category Detection: an Experimental Perspective. 2012.

Aalto-DD51/2012 Hanhijärvi, Sami
Multiple Hypothesis Testing in Data Mining. 2012.

Aalto-DD56/2012 Ramkumar, Pavan
Advances in Modeling and Characterization of Human Neuromagnetic Oscillations. 2012.

Aalto-DD97/2012 Turunen, Ville T.
Morph-Based Speech Retrieval: Indexing Methods and Evaluations of Unsupervised Morphological Analysis. 2012.

Aalto-DD115/2012 Vierinen, Juha
On statistical theory of radar measurements. 2012.

Aalto-DD117/2012 Huopaniemi, Ilkka
Multivariate Multi-Way Modelling of Multiple High-Dimensional Data Sources. 2012.

Aalto-DD137/2012 Paukkeri, Mari-Sanna
Language- and domain-independent text mining. 2012.

Aalto-DD133/2012 Ahlroth, Lauri
Online Algorithms in Resource Management and Constraint Satisfaction. 2012.

Aalto-DD158/2012 Virpioja, Sami
Learning Constructions of Natural Language: Statistical Models and Evaluations 2012.

Aalto-DD 20/2013

ISBN 978-952-60-4998-4
ISBN 978-952-60-4999-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Information and Computer Science
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS