MATRIX ANALYTIC METHODS WITH MARKOV DECISION PROCESSES FOR HYDROLOGICAL APPLICATIONS

Aiden James Fisher

December 2014

Thesis submitted for the degree of
Doctor of Philosophy
in the School of Mathematical Sciences at
The University of Adelaide
برای الگوی
I would like to acknowledge the help and support of various people for all the years I was working on this thesis.

My supervisors David Green and Andrew Metcalfe for all the work they did mentoring me through out the process of researching, developing and writing this thesis. The countless hours they have spent reading my work and giving me valuable feedback, for supporting me to go to conferences, giving me paid work to support myself, and most importantly, agreeing to take me on as a PhD candidate in the first place.

The School of Mathematical Sciences for giving me a scholarship and support for many years. I would especially like to thank all the office staff, Di, Jan, Stephanie, Angela, Cheryl, Christine, Renee, Aleksandra, Sarah, Rachael and Hannah, for all their help with organising offices, workspaces, computers, casual work, payments, the postgraduate workshop, conference travel, or just for having a coffee and a chat. You all do a great job and don’t get the recognition you all deserve.

All of my fellow postgraduates. In my time as a PhD student I would have shared offices with over 20 postgraduates, but my favourite will always be the “classic line” of G23: Rongmin Lu, Brian Webby, and Jessica Kasza. I would also like to give special thanks to Nick Crouch, Kelli Francis-Staite, Alexander Hanysz, Ross McNaughtan, Susana Soto Rojo, my brotégé Stephen Wade, and my guide to postgraduate life Jason Whyte, for being my friends all these years. Brian Webby once again for helping me with, and co-authoring, my first research paper and for being there for me while I settled into postgraduate life.

Kunle Akande for his insights and advice about the Essex Transfer Scheme that helped in the writing of the paper he co-authored with me.

To my loving parents, Sue and Greg, who supported me emotionally and financially for most of the early years I was a PhD candidate. Also my brothers and sister, Damon, Dominic, and Shannon, for their support.

Most importantly I would like to thank my darling wife Elham for her love and support while writing this thesis. Without her encouragement I would I have never finished and it owes its existence to her.
# Contents

Preamble .......................................................... xxv

1 Introduction .................................................. 1

2 Discrete Time Markov Chains ............................... 5
   2.1 Introduction ............................................. 5
   2.2 Markov Chain Terminology ............................. 7
      2.2.1 Reducibility of the Markov Chain ............... 7
      2.2.2 Passage and Recurrence in the Markov Chain .... 7
      2.2.3 Types of States in a Markov Chain ............. 8
   2.3 Introductory Markov Theory ............................ 9
      2.3.1 Markov Probability Distributions ................. 9
      2.3.2 Stationary Distribution .......................... 10
      2.3.3 Ergodic Chains .................................... 14
      2.3.4 Hitting Probabilities, Passage Times, and Absorbing Chains . 16

3 Discrete Time Markov Chains with Hidden States ....... 25
   3.1 Discrete Phase-Type Distribution ...................... 25
      3.1.1 Moments of the Discrete Phase-Type Distribution .... 26
      3.1.2 Properties of the Discrete Phase-Type Family ....... 27
   3.2 Matrix Analytic Methods .............................. 28
      3.2.1 General MAM Models .............................. 28
# CONTENTS

3.3 Hidden Markov Model ........................................ 31
   3.3.1 Relation to Matrix Analytic Methods ................ 33
   3.3.2 Fitting Hidden Markov Models: The Baum-Welch Algorithm . 34
3.4 Adaptation of the Baum-Welch Algorithm ................ 40
   3.4.1 Fitting Discrete-Time Batch Markovian Arrival Processes . . 40
   3.4.2 Fitting Discrete-Time Phase-Type Distributions .......... 42

4 Markov Decision Processes ................................. 45
   4.1 Introduction ............................................ 45
   4.2 Value Iteration ....................................... 46
      4.2.1 Discounted Value Iteration ...................... 47
   4.3 Policy Iteration ...................................... 47
   4.4 Stochastic Solution Methods .......................... 50
      4.4.1 Simulated Annealing ............................ 50
      4.4.2 Genetic Algorithms ............................ 51
   4.5 Partially Observable Markov Decision Processes ........ 52

5 Literature Review ........................................... 55
   5.1 Moran Dam Model ........................................ 55
      5.1.1 Gould Probability Matrix ....................... 57
   5.2 Linear Programming Models ............................. 59
      5.2.1 Stochastic Linear Programming .................. 59
   5.3 Stochastic Dynamic Programming - Single Reservoir Models .... 60
      5.3.1 Stochastic Dynamic Programming ................. 60
      5.3.2 Seasonal Stochastic Dynamic Programming Models .... 63
      5.3.3 Serially Correlated and Forward Forecast Stochastic Dynamic Programming Models .......... 63
   5.4 Stochastic Dynamic Programming - Multi-Reservoir Models .... 65
   5.5 Hidden Markov Models .................................. 67
CONTENTS

6 Synthesis 69

6.1 Optimal Control of Multi-Reservoir Systems with Time-Dependent Markov Decision Processes 71

6.2 Managing River Flow in Arid Regions with Matrix Analytic Methods 83

6.3 Modelling of Hydrological Persistence for Hidden State Markov Decision Processes 95

6.4 First-Passage Time Criteria for the Operation of Reservoirs 107

7 Conclusion 121

7.1 Further Work 123

A Supplementary Material for Chapter 3 127

A.1 Supplementary Material for Section 3.3 127

A.1.1 Demonstration of Equation (3.7) 127

A.1.2 Demonstration of Equation (3.9) 128

Bibliography 135
Abstract

In general, a physical process is modelled in terms of how its state evolves over time. The main challenge of modelling is to describe this evolution without unnecessary computation or making unrealistic simplifying assumptions. Markov chains have found widespread applications in many fields of analytic study from engineering to biology to linguistics. One of their most notable applications in hydrological applications has been modelling the storage of reservoirs, as described in Moran’s influential monograph (Moran, 1955). One of the fundamental properties of Markov chains is that the future evolution depends only on the present state, and not on any of the previous states. This property is simply stated as the “memory-less” property or the Markov property.

In a Markov chain model the states representing the physical process are discrete, but time can be modelled as either discrete or continuous. In this thesis, time is modelled in discrete units because this is consistent with the well-established theory of Markov decision processes. The discrete states need not be a practical limitation because of continuous state variables, as in this case storage in a reservoir, can be discretised as a reasonable approximation.

There have been many advances in Markov chain modelling techniques in other fields, most notably in telecommunications with the development of matrix analytic methods. Matrix analytic methods exploit the structure of certain types of Markov chains in order to more efficiently calculate properties of the models. This thesis examines how these methods can be applied to hydrological applications with the goal of providing a framework for which more precise modelling can be achieved without extending computational times. There are many unique challenges due to the seasonal nature of hydrology as well as the tendency for persistence of hydrological conditions. This thesis explores some of these problems in four papers.

The first paper looks at the issues surrounding hydrological persistence and its incorporation into Markov decision processes using the Southern Oscillation Index as proxy.

The second paper looks at modelling using matrix analytic methods of spate flows in the Cooper Creek, which is an ephemeral river located in the South Australia.

The third paper looks at a way of modelling hydrological persistence with underlying hidden states in contrast to assumed dependence on the Southern Oscillation Index.
The final paper looks at multi-objective optimisation using first-passage time distributions with an application to a two reservoir system in South East England. The Pareto front of Pareto optimal policies is shown.
Statement of Originality

I, Aiden James Fisher, certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I acknowledge that copyright of published works contained within this thesis (as listed below in the section titled “Published Works”) resides with the copyright holders of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University’s digital research repository, the Library catalogue, the Australasian Digital Theses Program (ADTP) and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed: ___________________________ Date: 31/7/14

Published Works


STATEMENT OF AUTHORSHIP

OPTIMAL CONTROL OF MULTI-RESERVOIR SYSTEMS WITH TIME-DEPENDENT MARKOV DECISION PROCESSES


Fisher, A.J. (Candidate)

Constructed mathematical model, conducted analysis and interpreted results, wrote manuscript and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed: ___________________________ Date: 31/11/14

Webby, R.B.

Researched and developed application as decision process problem, researched and fitted rainfall - runoff model, assisted in developing model and contributed to the writing of the manuscript.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the name in the thesis.

Signed: ___________________________ Date: 28/02/2012

Green, J.A.

Supervised development of work and provided critical evaluation.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the name in the thesis.

Signed: ___________________________ Date: 31/07/2014

Metcalfe, A.V.

Supervised development of work and provided critical evaluation.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the name in the thesis.

Signed: ___________________________ Date: 31/07/2014
STATEMENT OF AUTHORSHIP

MANAGING RIVER FLOWS IN ARID REGIONS WITH MATRIX ANALYTIC METHODS


Fisher, A.J. (Candidate)

Constructed mathematical model, conducted analysis and interpreted results, wrote manuscript and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed: ___________________________ Date: 21/7/19

Green, D.A.

Supervised development of work and provided critical evaluation.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of this paper in the thesis.

Signed: ___________________________ Date: 28/02/2012

Metcalfe, A.V.

Supervised development of work and provided critical evaluation.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of this paper in the thesis.

Signed: ___________________________ Date: 21/08/2014
STATEMENT OF AUTHORSHIP

MODELLING OF HYDROLOGICAL PERSISTENCE FOR HIDDEN STATE MARKOV DECISION PROCESSES


Fisher, A.J. (Candidate)

Constructed mathematical model, conducted analysis and interpreted results, wrote manuscript and acted as corresponding author.

I hereby certify that the statement of contribution is accurate.

Signed: Date: 31/7/14

Green, D.A.

Supervised development of work and provided critical evaluation.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed: Date: 10 APR 2014

Metcalfe, A.V.

Supervised development of work and provided critical evaluation.

I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed: Date: 21/8/2014
STATEMENT OF AUTHORSHIP

FIRST-PASSAGE TIME CRITERIA FOR THE OPERATION OF RESERVOIRS

Journal of Hydrology, under review.

Fisher, A.J. (Candidate)
Constructed mathematical model, conducted analysis and interpreted results, wrote manuscript and acted as corresponding author.
I hereby certify that the statement of contribution is accurate.

Signed: 
Date: 27 May 2014

Green, D.A.
Supervised development of work and provided critical evaluation.
I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed: 
Date: 10 April 2014

Metcalfe, A.V.
Supervised development of work and provided critical evaluation.
I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed: 
Date: 10 April 2014

Akande, K.
Provided description of, and data for, the case study.
I hereby certify that the statement of contribution is accurate and I give permission for the inclusion of the paper in the thesis.

Signed: 
Date: 3 April 2014
Preamble

This thesis has been submitted to the University of Adelaide for the degree of Doctor of Philosophy. According to the University’s Specification for Thesis a Doctoral thesis may comprise,

- a combination of conventional written narrative presented as typescript and publications that have been published and/or submitted for publication and/or text in manuscripts,

and this thesis takes this form.

The thesis has been divided into seven chapters:

The first chapter is a very brief introduction to the history of water management, the problems currently facing the world’s water supplies, and the future in a changing climate.

The second chapter gives a brief introduction to discrete-time Markov processes or Markov chains. This chapter is only intended as a cursory view of the topic. A more general and in depth discussion can be found in any textbook on the topic, for example Norris (1997). A reader familiar with this material could skip this chapter.

The third chapter introduces several more advanced Markov models, including: the phase-type distribution, Markov arrival processes, and hidden Markov models.

The fourth chapter discusses solution techniques for Markov decision processes.

The fifth chapter comprises of a literature review of mathematical modelling of reservoirs and methods for determining optimal policy using Markov decision processes.

The sixth chapter comprises four published papers that form the main component of the thesis, along with an outline of each paper. The published papers are presented in the format they were printed in. All the papers have been scaled to the largest size possible that will allow the pages to fit within the University’s specifications for a thesis.

The final chapter outlines potential future research following on from this thesis.
Chapter 1

Introduction

The human need to manage water resources can be traced back at least as far as the advent of agriculture around 9,000 BC. Archaeological records shows that as early as 5,000 years ago naturally forming reservoirs in volcanic craters were being used on the Arabian peninsula for the irrigation of crops. Some time between 5,000 and 4,000BC, canals were constructed in Mesopotamia, complete with weirs and sluice gates, not only for agricultural irrigation but also transportation (Tamburrino, 2010). The oldest known dam constructed by humans was either the Jawa Dam in Jordan, circa 3,000BC, (Fahlbusch, 2009) or the Sadd-el-Kafara Dam (Dam of the Pagans) in Egypt, circa 2,650BC (Mays, 2008). However Fahlbusch (2009) contends that given the size and scale of these dams there would have likely been smaller precursor dams, citing 7,000 year-old earth-works in Jordan that serve to hold back water pressure, but do not prevent seepage. Some of the early water management systems showed signs of being extensive and well planned. For example, the Nimrud Dam built on the Tigris River 180km upstream of Baghdad diverted water over 100km via the Nahrawan Canal to supplement the flows of the Diyala River (Mays, 2008).

The development of concrete by the Romans lead to some of the most complex water systems built to date. The famous public fountains and baths marked a change from water use as essential for human survival to a luxury item (Mays, 2010a). The dam and aqueducts allowed water to be sourced and stored hundreds of miles from cities and allowed for unprecedented growth in the Roman cities. Some of these dams, such as the Cornalvo Dam in the Badajoz province of Spain, are still in use today.

Modern super-dam construction began with the construction of the Hoover Dam in Black Canyon on the border of the states of Arizona and Nevada in the United States of America. At 221.4 meters high, 379 meters long and with a volume of 2,480,000 m$^3$, Hoover Dam was at the time of completion the largest concrete structure in the world. The current largest dam in the world is the recently completed Three Gorges Dam which impounds the Yangtze River is five times the volume at 181 meters high and 2,335 meters long.

More important than the size of the dam itself is the water body (or reservoir) it
CHAPTER 1. INTRODUCTION

holds back. Kariba Dam on the border between Zambia and Zimbabwe impounds the Zambezi River forming Lake Kariba, the largest man-made reservoir in the world by capacity at 180 km$^3$. Lake Mead, formed by Hoover Dam impounding the Colorado River, was the largest man-made reservoir at construction (now 20th) with a maximum capacity of 35.2 km$^3$ (in normal operation it should never exceed 32.2 km$^3$). However current storage is down to just 12.6 km$^3$ (USBR, 2014). Whilst this in part is caused by recent droughts, the long-term future of the reservoir, along with many others around the world, is under question. Barnett and Pierce (2008) examined a series of studies that found run-off into the Colorado River is expected to decrease by 10-30% over the next 30-50 years, due to changes in the catchment basin and potential decreases in future rainfall due to climate change. They predicted that based on projections of inflow then available, there was a 10% chance that Lake Mead would reach dead storage by 2013 and a 50% chance it would reach dead storage by 2021. Longer-term predictions are even more bleak. Even if no climate change occurs, there is a 90% chance of reaching dead storage by 2055, if climate change results in a loss of inflow by 20% dead storage would be reached by 2055 with almost certainty. The impact of the reservoir reaching dead storage would be immense. Lake Mead supplies 90% of the water used by the Southern Nevada Water Authority and the power generated at the Dam for over 1.3 million people.

In less developed countries the problem is more dire. The United Nations Children’s Fund estimates that 400 million children do not have access to safe drinking water (UNICEF, 2005) and approximately 1.1 billion people in total are without potable water. The United Nations Educational, Scientific and Cultural Organization issues a report every three years on the status and future of the world’s freshwater, with particular concern to disparity in access to freshwater as a fundamental human right (UNESCO, 2003, UNESCO, 2006, UNESCO, 2009, and UNESCO, 2012). Whilst the reports have found that access to freshwater is increasing, the supply of freshwater is decreasing.

Demand increase not only comes from population increase, but economic development. The two largest users of water per head of capita are the US and Australia, who respectively use 575 and 493 litres of water per person per day for entirely domestic purposes (UNESCO, 2003). If the entire human population was to use water at the rate of Australians, 3.3% of all annual freshwater runoff would be used in domiciles alone. Based on the Department of the Environment and Heritage estimates, in order for the entire world to consume water at the rate of Australians not only domestically, but in agriculture, mining, and industry, 22.5% of all the world’s annual water runoff would need to be captured and delivered. Postel et al. (1996) examined what it termed accessible runoff. Of the 40,700 km$^3$ of renewable water that runs off into rivers globally on average each year, 7,774 km$^3$/annum is in regions too remote to be accessible and 20,426 km$^3$/annum is short-term flood water that will go uncaptured due to physical restrictions on storage sizes, leaving just 12,500 km$^3$/annum that is geographically and temporally accessible. Using this definition of accessible water, 73% of all accessible runoff would be required for every person on Earth to enjoy the same standard of living as an Australian. Postel et al. (1996) found that humans have already appropriated 54% of all the accessible renewable freshwater runoff. The problem of obtaining accessible
renewable freshwater is exacerbated by the distribution of people living on Earth. Table 1.1 shows the distribution of human population and freshwater runoff by continent. The most stark feature of the table is that the human population is centred in regions that have comparatively lower freshwater runoff, with the Americas and the Oceanic Region being the only notable exceptions.

Table 1.1: Re-creation of the table “Share of global runoff and population by continent” shown in Postel et al. (1996).

<table>
<thead>
<tr>
<th>Region</th>
<th>River runoff (km$^3$/annum)</th>
<th>Share of global river runoff (%)</th>
<th>Share of global population(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe</td>
<td>3,240</td>
<td>8.0</td>
<td>13.0</td>
</tr>
<tr>
<td>Asia</td>
<td>14,550</td>
<td>35.8</td>
<td>60.5</td>
</tr>
<tr>
<td>Africa</td>
<td>4,320</td>
<td>10.6</td>
<td>12.5</td>
</tr>
<tr>
<td>North and Central America</td>
<td>6,200</td>
<td>15.2</td>
<td>8.0</td>
</tr>
<tr>
<td>South America</td>
<td>10,420</td>
<td>25.6</td>
<td>5.5</td>
</tr>
<tr>
<td>Australia and Oceania</td>
<td>1,970</td>
<td>4.8</td>
<td>0.5</td>
</tr>
<tr>
<td>Total</td>
<td>40,700</td>
<td>100.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

With increasing demand, decreasing supply, failing water storages, and a population that is distributed unevenly in relation to the distribution of freshwater, improved and sustainable management of water resources is not only desirable, it is essential. Mays (2010b) cites at least seven civilisations, including the Mohenjo Daro of Pakistan, the Akkadian Empire, and the Maya, whose demise was the result of, or instigated through, the loss of their water supply or through the mismanagement of the supply that existed. In 1999, Ismail Serageldin, Vice-President for Environmentally and Socially Sustainable Development of the World Bank predicted “The wars of the next century will be about water.”

Whilst larger water storage and delivering facilities would go some way to alleviating the problems ahead these are both expensive and only feasible in areas where the landscape permits reservoir construction. One of the most cost effective methods is improving the management techniques of the water management facilities already in place and also the methods used in determining the appropriation and distribution of the resource.

This thesis aims to make a contribution to improving the management techniques of reservoirs, by introducing new mathematical tools to the problem of finding operating policies that preserve the long-term amenity of the storage. The earliest reservoir management model, the Moran Dam model (see Section 5.1), was developed in the 1950s and used Markov chains as the basis of its construction. Whilst the Markov chain remains one of the most important modelling tools to date, various expansions of the idea have made it a more robust tool. Most notably, matrix analytic methods, were developed in the field of telecommunication systems modelling as a means of introducing more general distribution times within the Markov chain framework. Matrix analytic methods are applied to traditional reservoir modelling and optimisation techniques to develop models and operating policies that are far more general in structure and more descriptive of the system.
CHAPTER 1. INTRODUCTION

This thesis shows that by using the special phase structure of matrix analytic methods, more information can be included into long-term planning models. This in turn allows the demands on the resources to be met whilst simultaneously allowing the manager of these resources to more accurately predict the likelihood of future shortfalls. Finally it proposes a method by which the managers of the resource can find a strategy to minimise the probability of the system reaching dead-storage and obtain the maximum economic benefit from the resource through a compromise policy of operation.
Chapter 2

Discrete Time Markov Chains

Introduced by Markov (1906), the Markov chain forms the basis of the modelling techniques used in this thesis and some of its important properties are presented in this chapter. A reader that is already familiar with Markov chains may skip this chapter.

2.1 Introduction

Consider a countable set of states, or state space, $X = \{0, 1, 2, \ldots\}$, which is either finite or infinite, where the states are numbered sequentially. If there is a finite number of states, there exists a final state denoted $N$. A discrete time process $\{J_k\}$ on this state space which describes the state of the process at time $k \in \mathbb{Z}_+$ ($\mathbb{Z}_+$ the non-negative integers) is called a Markov chain if the following property holds:

$$
\Pr[J_{k+1} = j | J_k = i, J_{k-1}, \ldots, J_0] = \Pr[J_{k+1} = j | J_k = i], \quad i, j \in X.
$$

This Markovian property for a discrete time process is that the probability of transitioning from a state $J_k = i$ at time $k$ to another state $J_{k+1} = j$ at time $k + 1$ is only dependent on the state of the process at time $k$ and not on any previous state. This is known as the “memoryless” property of the Markov chain as the conditional probability remains unchanged by the process before time $k$. The only restriction on these probabilities are that they are true probabilities,

$$
0 \leq \Pr[J_{k+1} = j | J_k = i] \leq 1,
$$

and that,

$$
\sum_{j \in X} \Pr[J_{k+1} = j | J_k = i] = 1.
$$

A Markov chain is termed time-homogeneous if,

$$
\Pr[J_{s+k+1} = j | J_{s+k} = i] = \Pr[J_{k+1} = j | J_k = i], \quad \forall s > 0.
$$
If a Markov chain is time-homogeneous it is convenient to denote
\[
\Pr[J_{k+1} = j | J_k = i] = p_{ij}.
\]
Markov chains are often visualised using state diagrams. An example of a finite state Markov chain with four states \((N = 3)\) is given in Figure 2.1.

![State diagram of a \(N = 3\) Markov chain.](image)

As these can become quite complex this is not always the most convenient way of describing a Markov chain. During this thesis state diagrams will generally appear without the transition probabilities marked and sometimes the “self”-transition from \(i\) to \(i\) will be left out for clarity.

All of these time-homogeneous one step transition probabilities can be expressed in the form of a stochastic matrix where the \(i\) row and \(j\) column element is the transition probability \(p_{ij}\). For example, the chain in Figure 2.1 has the one-step transition probability matrix,

\[
P = \begin{pmatrix}
0 & 1 & 2 & 3 \\
0 & p_{00} & p_{01} & p_{02} & p_{03} \\
1 & p_{10} & p_{11} & p_{12} & p_{13} \\
2 & p_{20} & p_{21} & p_{22} & p_{23} \\
3 & p_{30} & p_{31} & p_{32} & p_{33}
\end{pmatrix}.
\]

Stochastic matrices are any matrix whose rows form a probability distribution and can be used to describe more than just Markov chains, with the only restriction being that all the elements of the matrix are greater than or equal to zero and that the rows sum to one. Sometimes the matrix may have additional indexing, so in order to make this clearer the \(i\) row and \(j\) column element of the matrix \(P\) may be denoted \([P]_{ij}\) in addition to the \(p_{ij}\) notation already introduced. If the counting of the state space starts at 1 instead of 0, likewise then the numbering of the rows and columns will start at 1 instead of 0.
Due to time-homogeneity, the two step transition probability,

\[
\Pr[J_{k+2} = j | J_k = i] = \sum_{n=0}^{N} \Pr[J_{k+2} = j | J_{k+1} = n] \Pr[J_{k+1} = n | J_k = i]
\]

\[
= \sum_{n=0}^{N} \Pr[J_{k+1} = j | J_k = n] \Pr[J_{k+1} = n | J_k = i]
\]

\[
= \sum_{n=0}^{N} p_{in} p_{nj},
\]

which is the \(i\) row and \(j\) column element of the matrix \(P^2\), denoted \(p_{ij}^{(2)}\). By induction the \(n\)-step transition probability,

\[
\Pr[J_{k+n} = j | J_k = i]
\]

is given by the \(i\) row and \(j\) column element of the matrix \(P^n\), denoted \(p_{ij}^{(n)}\).

## 2.2 Markov Chain Terminology

Markov chains can be categorised into different types based on the properties of the chain and results can be obtained more readily.

### 2.2.1 Reducibility of the Markov Chain

Reducibility of the Markov chain deals with whether some states maybe reached from other states, or whether a the chain is joined at all. If states become disjoint through branching it maybe possible to reduce the state space as the chain evolves through time.

**Definition 1** The state \(j\) is said to be accessible from \(i\), denoted \(i \rightarrow j\), if there exists some integer \(k \geq 0\) such that \(p_{ij}^{(k)} > 0\). If \(j\) is not accessible from \(i\) it is denoted \(i \not\rightarrow j\).

**Definition 2** Two states \(i\) and \(j\) are said to communicate if both \(i \rightarrow j\) and \(j \rightarrow i\), denoted \(i \leftrightarrow j\).

**Definition 3** A Markov chain is said to be irreducible if \(i \leftrightarrow j\) for all \(i, j \in X\).

### 2.2.2 Passage and Recurrence in the Markov Chain

As a Markov chain is typically used to replicate a real-world system the time between states and whether the state is reached again is usually of interest.
Definition 4 The hitting time from state $i$ to state $j$ which is accessible from $i$ is the random variable defined as

$$H_{ij} = \inf\{n \geq 0 : J_k = i, J_{k+n} = j\}.$$ 

If $i \not\rightarrow j$, $H_{ij} = \infty$.

Definition 5 Let $f^{(k)}_{ij}$ be the probability that the process takes exactly $k$ steps to go from state $i$ and to state $j$ for the first time, that is,

$$f^{(k)}_{ij} = \Pr[H_{ij} = k].$$

The hitting probability is the probability of ever going from state $i$ to state $j$, that is

$$f_{ij} = \sum_{k=0}^{\infty} f^{(k)}_{ij}.$$ 

If $i \not\rightarrow j$, $f_{ij} = 0$.

Definition 6 The expected hitting time is the expected number of steps for the process to go from state $i$ to state $j$, that is

$$u_{ij} = \mathbb{E}[H_{ij}] = \sum_{k=0}^{\infty} kf^{(k)}_{ij}.$$ 

Definition 7 Let $f^{(k)}_{i}$ be the probability that there are exactly $k$ steps between leaving state $i$ and returning to state $i$, that is

$$f^{(k)}_{i} = \Pr[H_{ii} = k > 0].$$

The return probability is the probability of ever returning to state $i$, that is

$$f_{i} = \sum_{k=1}^{\infty} f^{(k)}_{i}.$$ 

Definition 8 The first passage time of state $i$ from initial distribution $\pi_0$ is the random variable defined as

$$K_i = \inf\{n \geq 1 : J_n = i|\pi_0\}.$$ 

2.2.3 Types of States in a Markov Chain

The easy identification of the type of Markov chain being dealt with allows it properties to be more readily deduced. Certain types of Markov chain states have special properties which give information about the chain as a whole.
Definition 9 A state $i$ is said to be recurrent if the probability of ever returning is one, i.e,
\[ f_i = \sum_{k=1}^{\infty} f_i^{(k)} = 1. \]

Definition 10 A recurrent state $i$ is said to be positive-recurrent if it returns in finite time, i.e,
\[ m_i = \sum_{k=1}^{\infty} k f_i^{(k)} < \infty, \]
where $m_i$ is called the mean recurrence time. It is said to be null-recurrent if $m_i \to \infty$.

Definition 11 A state $i$ is said to be transient if there exists non-zero probability of never returning, i.e,
\[ f_i = \sum_{k=1}^{\infty} f_i^{(k)} < 1. \]

Definition 12 A transient state $i$ is said to be ephemeral if once the process has left state $i$ it never returns, i.e,
\[ f_i = \sum_{k=1}^{\infty} f_i^{(k)} = 0. \]

Definition 13 A state $i$ is said to be periodic if for $k \geq 1$,
\[ d = \gcd\{k : p_i^{(k)} > 0\} > 1. \]
The state $i$ is said to have periodicity $d$. A state $i$ is said to be aperiodic if $d = 1$.

Definition 14 A state $i$ is said to be ergodic if it is both aperiodic and positive-recurrent.

Definition 15 A state $i$ is said to be absorbing if $p_{ii} = 1$.

2.3 Introductory Markov Theory

2.3.1 Markov Probability Distributions

A Markov chain can be used to describe the evolution of a distribution on $X$ over time.
Define $\pi^{(0)}$ as the distribution on $X$ at time $k = 0$, or the initial distribution, as a vector with $i^{th}$ element,
\[ \pi_i^{(0)} = \Pr[J_0 = i]. \]
The distribution on $X$ at time $k = 1$, $\pi^1$, is then given by a vector with $j^{th}$ element,
\[ \pi_j^{(1)} = \sum_{i \in X} \Pr[J_1 = j | J_0 = i] \Pr[J_0 = i] = \sum_{i \in X} \Pr[J_1 = j | J_0 = i] \pi_i^{(0)} . \]

Due to the Markov property, the distribution of \( X \) at time \( k + 1 \), \( \pi^{(k+1)} \), is given by a vector with \( j^{th} \) element,

\[ \pi_j^{(k+1)} = \sum_{i \in X} \Pr[J_{k+1} = j | J_k = i] \pi_i^{(k)} . \]

If the Markov chain is time-homogeneous the distributions are given by,

\[ \pi^{(n+1)} = \pi^{(n)} P = \pi^{(n-1)} P^2 = \cdots = \pi^{(0)} P^{n+1} . \]

### 2.3.2 Stationary Distribution

Of particular interest is the invariant or stationary distribution of the Markov chain, which represents a distribution from which the process will never move. The stationary distribution for a Markov chain, if it exists, is given by the global balance equation,

\[ \pi_j = \sum_{i \in X} \pi_i \Pr[J_{k+1} = j | J_k = i] , \tag{2.1} \]

where \( \pi_i \) is the stationary probability of state \( i \) and that,

\[ \sum_i \pi_i = 1 . \]

In the case where the Markov chain is time-homogeneous, Equation (2.1) can be expressed as,

\[ \pi_j = \sum_i \pi_i p_{ij} . \]

In the case where the state-space is finite the stationary distribution always exists, a row vector of length \( N + 1 \), \( \pi \), can be defined whose \( i^{th} \) element is the stationary probability for state \( i \), \( \pi_i \).
In the case where the Markov chain is finite and time-homogeneous the vector $\pi$ can be expressed equivalently to Equation (2.1) in the matrix equation,

$$\pi = \pi P,$$

where

$$\pi 1 = 1,$$

where $1$ is a column vector of ones of the appropriate length. In this case, finding the stationary probability vector corresponds to finding the left eigenvector of the eigenvalue equal to 1.

**Example: A Rainy Day**

The weather can be in one of two states dry or raining (Gabriel and Neumann, 1962). Assume that it is observed regularly with a fixed time-step between each observation (e.g., once every 15 minutes). If at the current observation it is dry it can be raining by the next observation in one time-step with probability $0 < \alpha < 1$. If the observation is that it is currently raining it will stop raining by the next observation in one time-step with probability $0 < \beta < 1$. This process can be modelled as a Markov process with probability transition matrix,

$$P = \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix}.$$ 

The stationary distribution $\pi$ is found by solving $\pi (I_2 - P) = 0$. This problem can either be transposed to $(I_2 - P^t) \pi^t = 0^t$ and solved using row operations, or it can be solved using column operations. Column operations reduces $(I_2 - P)$ to,

$$\begin{pmatrix} 1 & 0 \\ -\beta/\alpha & 0 \end{pmatrix}.$$ 

This gives the eigenvector solution that, $\pi = s \left( \frac{\beta}{\alpha} \ 1 \right)$ for all $s \in \mathbb{R}$. However as the probabilities need to be normalised such that $\pi_0 + \pi_1 = 1$, the final solution is found to be,

$$\pi = \begin{pmatrix} \frac{\beta}{\alpha + \beta} & \frac{\alpha}{\alpha + \beta} \end{pmatrix}.$$ 

Therefore if the process goes on long enough, \[ \frac{\beta}{\alpha + \beta} \] will be the proportion of the observations it will be dry and \[ \frac{\alpha}{\alpha + \beta} \] will be the proportion of the observations it will be raining.

**Example: A Simple Reservoir Model**

A simple reservoir model can be constructed using Markov chains, by letting $i \in X$ represent the discretised storage levels of the reservoir, with $i = 0$ representing the
storage being empty and \( i = 8 \) representing the storage being full. Let the probability that the storage increases by one unit of storage in one time-step be 0.3, the storage decreases one unit be 0.2 and the probability it is unchanged be 0.5, unless the reservoir is full or empty. If the reservoir is empty then the probability it remains empty is 0.2 + 0.5 = 0.7. Likewise, if the reservoir is full the probability that it remains full is 0.5 + 0.3 = 0.8. Therefore the transition probabilities are,

\[
p_{ij} = \begin{cases} 
0.2 & \text{if } j = i - 1, i \neq 0 \\
0.5 & \text{if } j = i, i \neq 0, 8 \\
0.3 & \text{if } j = i + 1, i \neq 8 \\
0.7 & \text{for } i = j = 0 \\
0.8 & \text{for } i = j = 8 \\
0 & \text{otherwise}
\end{cases}
\]

The reservoir can then be described by the probability transition matrix,

\[
P = \begin{pmatrix}
0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.2 & 0.5 & 0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.2 & 0.5 & 0.3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.2 & 0.5 & 0.3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.2 & 0.5 & 0.3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 \\
\end{pmatrix}
\]

The stationary distribution is once again found by solving \((I_9 - P^t)\pi^t = 0^t\). This problem can be expressed in augmented matrix form as,

\[
\begin{pmatrix}
0.3 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0.5 & 0 \\
\end{pmatrix}
\]

Due to linear dependence an infinite number of solutions will not be found. Instead the last line is replaced with the condition that \(\sum \pi_i = 1\), yielding,
2.3. INTRODUCTORY MARKOV THEORY

\[
\begin{pmatrix}
0.3 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{pmatrix}
\begin{pmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
1 \\
\end{pmatrix}
\]

Row reducing this new augmented matrix gives the solution,

\[\pi = (0.01335, 0.02003, 0.03005, 0.04507, 0.06760, 0.10140, 0.15210, 0.22816, 0.34224)\]

This tells us that over long term operation the reservoir will be full 34.2% of the time and will be empty 1.3% of the time.

**Example: M/M/1 Queue with Discrete Time-Steps**

The M/M/1 Queue is a single server queue with infinite holding capacity (\(M\) indicates that the arrival and service distributions are Markovian, later queues with a general, \(G\), arrivals and departures are discussed) (Latouche and Ramaswami, 1999). This model is similar to the reservoir model, except that it is not finite. Whilst an infinite storage reservoir is physically impossible, due to the ease with which analytic results can be obtained the Moran Dam Model in Section 5.1 makes use of such a construction and if the probability of reaching full-capacity is small it can be a good approximation.

The probability of one customer arriving in one time-step is \(\alpha > 0\), and the probability that one customer is served (given they exist) in one time-step is \(\beta > 0\), the probability there is no change in the number of customers is \(1 - \alpha - \beta \geq 0\). To make notation more intuitive, denote the state with \(i\) customers as \(i \in X\), including 0 \(\in X\), a state from which there are no customers to serve however customers arrive with probability \(\alpha\), therefore \(p_{00} = 1 - \alpha\). A system of equations can be written using Equation (2.1) as,

\[
\pi_i = \alpha \pi_{i-1} + (1 - \alpha - \beta) \pi_i + \beta \pi_{i+1} \quad \text{for } i \neq 0
\]

\[
\pi_0 = (1 - \alpha) \pi_0 + \beta \pi_1
\]

subject to \(\sum \pi_i = 1\).

Consider the boundary condition,

\[
\pi_0 = (1 - \alpha) \pi_0 + \beta \pi_1
\]

\[
\beta \pi_1 = \alpha \pi_0
\]

\[
\pi_1 = \frac{\alpha}{\beta} \pi_0
\].
Likewise,

\[
\begin{align*}
\pi_1 &= \alpha \pi_0 + (1 - \alpha - \beta) \pi_1 + \beta \pi_2 \\
\beta \pi_2 &= (\alpha + \beta) \pi_1 - \alpha \pi_0 \\
\pi_2 &= \frac{\alpha + \beta}{\beta} \pi_1 - \frac{\alpha}{\beta} \pi_0 \\
&= \frac{\alpha^2 + \alpha \beta}{\beta^2} \pi_0 - \frac{\alpha}{\beta} \pi_0 \\
&= \left(\frac{\alpha}{\beta}\right)^2 \pi_0 .
\end{align*}
\]

By induction it can be shown that,

\[
\pi_i = \left(\frac{\alpha}{\beta}\right)^i \pi_0 .
\]

Using the fact that the stationary probabilities must sum to one,

\[
\pi_0 \left[ \sum_{i=0}^{\infty} \left(\frac{\alpha}{\beta}\right)^i \right] = 1 ,
\]

it can be determined if this system has a stationary distribution. If \( \alpha < \beta \),

\[
\sum_{i=0}^{\infty} \left(\frac{\alpha}{\beta}\right)^i = \frac{1}{1 - \frac{\alpha}{\beta}} = \frac{\beta}{\beta - \alpha} ,
\]

otherwise the sum diverges. Hence,

\[
\pi_i = \frac{\beta - \alpha}{\beta} \left(\frac{\alpha}{\beta}\right)^i , \text{ if and only if } \alpha < \beta ,
\]

otherwise the stationary distribution does not exist.

### 2.3.3 Ergodic Chains

Consider an irreducible time-homogeneous Markov chain with a state \( i \) that is ergodic.

As the chain is irreducible, for any other state \( j \neq i \in X \) there exists \( r \) and \( s \) such that \( p_{ji}^{(r)} > 0 \) and \( p_{ij}^{(s)} > 0 \). As \( i \) is positive recurrent there exists a \( k \) such that \( p_{ii}^{(k)} > 0 \). As \( j \rightarrow i \rightarrow j \) represents at least one path by which the process can go from \( j \) to \( j \) in \( m = r + k + s \) time-steps then,

\[
p_{jj}^{(m=r+k+s)} \geq p_{ji}^{(r)} p_{ii}^{(k)} p_{ij}^{(s)} > 0 .
\]

It follows then that the first-return time for state \( j \), \( H_{jj} \), must be,

\[
H_{jj} \leq H_{ii} + r + s ,
\]
as $j \rightarrow i \rightarrow j$ represents at least one path by which the process may return to $j$ having left it, and as $H_{jj}$ is the number of steps on the shortest path, it maybe no longer than this. Consider the expected value,

$$m_j = \mathbb{E}[H_{jj}] \leq \mathbb{E}[H_{ii} + r + s] = m_i + r + s < \infty,$$

and so all $j \neq i \in X$ are positive-recurrent if $i \in X$ is positive recurrent and $X$ is irreducible.

For any aperiodic state it can be shown that,

$$\lim_{n \to \infty} p_{ij}^{(n)} \to 1/m_j = \pi_j,$$

where $\pi_j$ is the stationary distribution. The proof of this is very complicated involving coupling two Markov chains together, it can be found in Norris (1997, Section 1.8).

For any ergodic chain a matrix $S$ can be written such that

$$\lim_{n \to \infty} P^n \to S = 1\pi,$$

a matrix of rank 1 which has every row as the stationary distribution. It is sometimes useful to write,

$$P^n = S + T(n),$$

where

$$\lim_{n \to \infty} [T(n)]_{ij} \to 0 \quad \forall i, j \in X.$$

The limit of periodic chains do not exist in the same sense. Instead consider partitioning the state space of an irreducible Markov chain where every state has period $d$ as,

$$X = X_0 \cup X_1 \cup \cdots \cup X_{d-1}$$

where,

- $p_{ij}^{(r)} > 0$ only if $i \in X_0$ and $j \in X_r$, and
- $p_{ij}^{(nd)} > 0$ for some sufficiently large $n$ when $i, j \in X_0$.

It can be shown for all $i \in X_0$ and $j \in X_r$,

$$\lim_{n \to \infty} p_{ij}^{(dn+r)} \to d/m_j,$$

where $r = 0, 1, \ldots, d - 1$. This shows that the periodic chain exhibits long-term stable behaviour independent of its starting state even if its limit doesn’t exist in the normal sense, only being defined every $d$ steps.
2.3.4 Hitting Probabilities, Passage Times, and Absorbing Chains

The probability of hitting a state \( j \) in a finite number of steps given that the process starts in state \( i \) is denoted \( f_{ij} \) and is given by Definition 5. It can also be defined as the probability that there exists at least one non-negative integer \( n \) such that \( J_{n+k} = j \) given that \( J_k = i \) or expressed mathematically as

\[
f_{ij} = \Pr[H_{ij} < \infty] = \Pr[\inf\{n \geq 0 : J_k = i, J_{k+n} = j\} < \infty].
\]

More generally, define the random variable \( H_i^A \) of reaching a collection of states \( A \subset X \) from state \( i \) in exactly \( n \) steps as,

\[
H_i^A = \inf\{n \geq 0 : J_k = i, J_{k+n} \in A\}.
\]

Similarly the probability \( f_i^A \) of hitting a collection of states \( A \subset X \) from \( i \notin A \) is given by,

\[
f_i^A = \Pr[H_i^A < \infty] = \Pr[\inf\{n \geq 0 : J_k = i, J_{k+n} \in A\} < \infty].
\]

If \( i \in A \) then \( f_i^A = 1 \) by definition. If \( i \notin A \), then \( n \geq 1 \) and the process will have to step to another state at time \( k+1 \) that is,

\[
\Pr[\inf\{n \geq 0 : J_k = i, J_{k+1} = \ell, J_{k+n} \in A\} < \infty]
= \Pr[J_{k+1} = \ell | J_k = i] \Pr[\inf\{n \geq 1 : J_{k+1} = \ell, J_{k+n} \in A\} < \infty].
\]

From this result it follows that

\[
f_i^A = \Pr[\inf\{n \geq 0 : J_k = i, J_{k+n} \in A\} < \infty],
\]

\[
= \sum_{\ell \in X} \Pr[\inf\{n \geq 0 : J_k = i, J_{k+1} = \ell, J_{k+n} \in A\} < \infty],
\]

\[
= \sum_{\ell \in X} \Pr[J_{k+1} = \ell | J_k = i] \Pr[\inf\{n \geq 1 : J_{k+1} = \ell, J_{k+n} \in A\} < \infty] \text{ for } i \notin A.
\]

If the Markov chain is time-homogeneous then

\[
f_i^A = \sum_{\ell \in X} p_{i\ell} f_{i\ell}^A \text{ for } i \notin A.
\]

The expected number of time-steps to reach a collection of states \( A \subset X \) from state \( i \) starting at time 0 is defined as,

\[
u_i^A = E_k[k = \inf\{n : J_0 = i, J_n \in A\}].
\]
If $i \in A$, then $u_i^A = 0$, otherwise $u_i^A \geq 1$. Therefore the process has to go through one more transition state $\ell$ at time $k = 1$, that is

$$
E_k[k = \inf\{n : J_0 = i, J_1 = \ell, J_n \in A\}] = 1 + E_k[k = \inf\{n : J_1 = \ell, J_n \in A\}].
$$

From this result it follows that

$$
u_i^A = E_k[k = \inf\{n : J_0 = i, J_n \in A\}],
$$

$$
= \sum_{\ell \in X} E_k[k = \inf\{n : J_0 = i, J_1 = \ell, J_n \in A\}],
$$

$$
= \sum_{\ell \in X} \Pr[J_1 = \ell | J_0 = i] [1 + E_k[k = \inf\{n : J_1 = \ell, J_n \in A\}]],
$$

$$
v_i^A = 1 + \sum_{\ell \in X} \Pr[J_1 = \ell | J_0 = i] E_k[k = \inf\{n : J_1 = \ell, J_n \in A\}] \text{ for } i \not\in A.
$$

If the Markov chain is time-homogeneous then

$$
u_i^A = 1 + \sum_{\ell \in X} p_{i\ell} u_{\ell}^A \text{ for } i \not\in A.
$$

Both the hitting probability and the expected number of steps for a time-homogeneous Markov chain can be written a set of linear equations:

$$
f_i^A = 1 \quad \forall i \in A
$$

$$
f_i^A = \sum_{\ell \in X} p_{i\ell} f_{\ell}^A \quad \forall i \not\in A, \tag{2.2}
$$

and

$$
u_i^A = 0 \quad \forall i \in A
$$

$$
u_i^A = 1 + \sum_{\ell \in X} p_{i\ell} u_{\ell}^A \quad \forall i \not\in A. \tag{2.3}
$$

Define $f^A$ to be the vector in which $i^{th}$ element is the probability $f_i^A$. By construction it is a solution to system of linear equations given in Equation (2.2).

It is straightforward to show that $f^A$ is the minimal non-negative solution to Equation (2.2). Suppose $x$ is a non-negative solution to Equation (2.2), then for $i \in A$ the $i^{th}$ element of $x$, $x_i = 1$, and for $i \not\in A$, $x_i \geq 0$ and satisfies,

$$
x_i = \sum_{\ell_1 \in X} p_{i\ell_1} x_{\ell_1}
$$

$$
= \sum_{\ell_1 \in A} p_{i\ell_1} + \sum_{\ell_1 \not\in A} p_{i\ell_1} x_{\ell_1}. \tag{2.4}
$$
Substituting Equation (2.4) into itself yields,

\[ x_i = \sum_{\ell_1 \in A} p_{i\ell_1} + \sum_{\ell_1 \notin A} \left( \sum_{\ell_2 \in A} p_{\ell_1 \ell_2} + \sum_{\ell_2 \notin A} p_{\ell_1 \ell_2} x_{\ell_2} \right) \]

\[ = \sum_{\ell_1 \in A} p_{i\ell_1} + \sum_{\ell_1 \notin A} \sum_{\ell_2 \in A} p_{\ell_1 \ell_2} + \sum_{\ell_1 \notin A} \sum_{\ell_2 \notin A} p_{\ell_1 \ell_2} x_{\ell_2}. \]

The first term \( p_{ij} \) is the probability of going from \( i \) to \( j \) in one time-step. This can be expressed as,

\[ \sum_{\ell_1 \in A} p_{i\ell_1} = \Pr[J_1 \in \mathcal{A}|J_0 = i \notin \mathcal{A}] = \Pr[H_A^1 = 1]. \]

Similarly the second term can be expressed as,

\[ \sum_{\ell_1 \notin A} \sum_{\ell_2 \in A} p_{\ell_1 \ell_2} = \Pr[J_2 \in \mathcal{A}|J_0 = i \notin \mathcal{A}, J_1 \notin \mathcal{A}] = \Pr[H_A^1 = 2]. \]

By induction, repeated substitutions yields that,

\[ x_i = \Pr[H_{ij} = 1] + \cdots + \Pr[H_{ij} = n] + \sum_{\ell_1 \notin A} \cdots \sum_{\ell_n \notin A} p_{\ell_1 \cdots \ell_n} x_{\ell_n}. \quad (2.5) \]

As \( x_i \geq 0 \) for all \( i \in X \), all terms in Equation (2.5) at positive and,

\[ x_i \geq \lim_{n \to \infty} \Pr[H_A^1 \leq n] = \Pr[H_A^1 < \infty] = f^A_i. \]

Therefore \( f^A \) is the minimal non-negative solution of Equation (2.2).

Similarly let \( u^A \) be a vector with \( i^{th} \) element \( u^A_i \). By construction it is a solution to system of linear equations given in Equation (2.3).

Again it can be shown that \( u^A \) is the minimal non-negative solution to Equation (2.3). Suppose \( y \) is a non-negative solution to Equation (2.3), then for \( i \in \mathcal{A} \) the \( i^{th} \) element of \( y \), \( y_i = 0 \), and for \( i \notin \mathcal{A} \), \( y_i \geq 1 \) and satisfies,

\[ y_i = 1 + \sum_{\ell_1 \notin X} p_{i\ell_1} y_{\ell_1} \]

\[ = 1 + \sum_{\ell_1 \in A} p_{i\ell_1} y_{\ell_1} + \sum_{\ell_1 \notin A} p_{i\ell_1} y_{\ell_1} \]

\[ = 1 + \sum_{\ell_1 \notin A} p_{i\ell_1} y_{\ell_1}. \quad (2.6) \]

Substituting Equation (2.6) into itself yields,

\[ y_i = 1 + \sum_{\ell_1 \notin A} p_{i\ell_1} \left( 1 + \sum_{\ell_2 \notin A} p_{\ell_1 \ell_2} y_{\ell_2} \right) \]

\[ = 1 + \sum_{\ell_1 \notin A} p_{i\ell_1} + \sum_{\ell_1 \notin A} \sum_{\ell_2 \notin A} p_{i\ell_1} p_{\ell_1 \ell_2} y_{\ell_2}. \]
2.3. INTRODUCTORY MARKOV THEORY

The first term is the probability that the process does not start in $A$ given that the starting state is not in $A$, that is,

$$\Pr[J_0 \notin A | J_0 = i \notin A] = 1,$$

or alternatively.

$$\Pr[\inf\{n : J_n \in A, J_0 = i \notin A\} > 0] = \Pr[H_i^A \geq 1].$$

The second term is the probability the process still has not entered a state in $A$ at time-step $k = 1$,

$$\sum_{\ell_1 \notin A} p_{i\ell_1} = \Pr[H_i^A \geq 2].$$

By induction it can be shown that,

$$y_i = \Pr[H_i^A \geq 1] + \cdots + \Pr[H_i^A \geq n] + \sum_{\ell_1 \notin A} \cdots \sum_{\ell_n \notin A} p_{i\ell_1} \cdots p_{\ell_{n-1}\ell_n} y_{\ell_n}.$$

As all terms are non-negative,

$$y_i \geq \lim_{n \to \infty} \sum_{k=1}^{n} \Pr[H_i^A \geq k] = \sum_{k=1}^{\infty} \Pr[H_i^A \geq k] = \sum_{k=1}^{\infty} k \Pr[H_i^A = k] = u_i^A.$$

Therefore the vector $u^A$ is the minimal non-negative solution for Equation (2.3).

**Example: A Simple Reservoir Model**

Revisiting the simple reservoir model from Section 2.3.2, the system will now be considered a failure if it goes empty. To model this new system if the reservoir goes empty it will remain there signalling the failure of the system. The transition probabilities are now,

$$p_{ij} = \begin{cases} 
0.2 & \text{if } j = i - 1, i \neq 0 \\
0.5 & \text{if } j = i, i \neq 0, 8 \\
0.3 & \text{if } j = i + 1, i \neq 0, 8 \\
1 & \text{for } i = j = 0 \\
0.8 & \text{for } i = j = 8 \\
0 & \text{otherwise}
\end{cases}.$$

The transition probability matrix is then,

$$P = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.2 & 0.5 & 0.3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.2 & 0.5 & 0.3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.2 & 0.5 & 0.3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.2 & 0.5 & 0.3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.2 & 0.5 & 0.3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8
\end{pmatrix}. $$
which has been partitioned into four submatrices: the transition between the trans-
ient states (top-left); the transitions to the absorbing states (top-right); and the self-
transitions of the absorbing states (bottom-right), leaving a matrix of zeros in the 
bottom-left. The partitions are labelled as follows,

\[ P = \begin{pmatrix} 1 & 0 \\ A & T \end{pmatrix}, \]

so that \( T \) is a \( 8 \times 8 \) matrix of transient probabilities, \( A \) a \( 8 \times 1 \) matrix of absorbing probabilities, \( 0 \) a \( 1 \times 8 \) matrix of zeros and 1 is the absorbing state of the reservoir being empty. It can be shown that for any matrix partitioned in this way that,

\[ P^n = \begin{pmatrix} 1 & 0 \\ \sum_{i=0}^{n-1} T^i A & T^n \end{pmatrix}, \]

and that in the limit,

\[ \lim_{n \to \infty} P^n = \begin{pmatrix} 1 & 0 \\ (I - T)^{-1} A & 0 \end{pmatrix}. \]

Let \( u \) be the vector whose \( i^{th} \) element gives the expected number of steps from state \( i \notin A \) to one of the absorbing states in \( A \). Therefore

\[ u_i = 1 + \sum p_{ij} u_j, \]

or in vector form,

\[
\begin{align*}
\mathbf{u} &= \mathbf{1} + T \mathbf{u} \\
\mathbf{u} - T \mathbf{u} &= \mathbf{1} \\
(I - T) \mathbf{u} &= \mathbf{1} \\
\mathbf{u} &= (I - T)^{-1} \mathbf{1}
\end{align*}
\]

For the reservoir model the results are,

\[
\mathbf{u} = \begin{pmatrix} 246.29 \\ 407.15 \\ 511.05 \\ 576.99 \\ 617.62 \\ 641.37 \\ 653.87 \\ 658.87 \end{pmatrix}
\]

Hence, if for example the reservoir is initially full, it is expected to last 658.87 time-steps until it goes empty.

To find probability that the reservoir goes full before it goes empty from any state \( i = 1, \ldots, 7 \), consider a vectorised version of Equation (2.2),

\[ \mathbf{f} = P \mathbf{f} \]

\[ \mathbf{f} - P \mathbf{f} = \mathbf{0} \]

\[ (I_9 - P) \mathbf{f} = \mathbf{0}, \]
where $f$ is a column vector whose $i^{th}$ element is the probability of going from $i$ to 9 without reaching 0. However this equation only holds for $i \neq 9$ and $i \rightarrow 9$. To fix this only the rows $i = 1, \ldots, 8$ will be included in the augmented matrix. Instead the equations $f_9^0 = 1$ (from the definition) and $f_9^0 = 0$ (to represent the inaccessibility of 9 from 0) will be added. This yields the augmented matrix equation,

\[
P = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.2 & 0.5 & -0.3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0.5 & -0.3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 & 0.5 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\]

Row reducing now yields,

\[
f = \begin{pmatrix}
0 \\
0.34687 \\
0.57811 \\
0.73228 \\
0.83505 \\
0.90357 \\
0.94925 \\
0.97970 \\
1
\end{pmatrix}
\]

which give the probability of the reservoir being full before it goes empty starting in each state of the system.

**Example: The Gambler’s Ruin**

Similar to the $M/M/1$ server queue, however now the state $0$ is absorbing. The state $i \in X$ represents the amount of money a gambler has. Each time he bets one monetary unit and wins one additional unit with probability $\alpha > 0$, loses the bet unit with probability $\beta > 0$, or pushes (neither wins nor loses and has his bet returned) with probability $1 - \alpha - \beta \geq 0$, once the gambler runs out of money he is no longer able to place a bet and is in ruin (state $i = 0$).

If the Gambler starts the game with $C$ amount of money how long until they go broke?

There are two equations describing the amount of time until the gambler goes broke:

\[
u_C = 1 + \alpha u_{C+1} + \beta u_{C-1} + (1 - \alpha - \beta) u_C ,
\]

and the boundary condition,

\[
u_0 = 0 \Rightarrow u_1 = 1 + \alpha u_2 + (1 - \alpha - \beta) u_1 .
\]
First consider the homogeneous equation,
\[ \alpha u_{C+1} - (\alpha + \beta)u_C + \beta u_{C-1} = 0 . \]

Try the solution \( u_C = m^C \),

\[ \alpha m^{C+1} - (\alpha + \beta)m^C + \beta m^{C-1} = 0 \]
\[ \alpha m^2 - (\alpha + \beta)m + \beta = 0 \]

\[ \Rightarrow m = \frac{(\alpha + \beta) \pm \sqrt{(\alpha + \beta)^2 - 4\alpha\beta}}{2\alpha} \]
\[ = \frac{\beta}{\alpha} \text{ or } 1 . \]

So the general solution for the homogeneous case is,

\[ u_C = A_h + B_p \left( \frac{\beta}{\alpha} \right)^C . \]

Applying the boundary condition gives,

\[ u_C = A_h \left( 1 - \left( \frac{\beta}{\alpha} \right)^C \right) . \]

Try \( u_C = A_p C + B_p \) a particular solution for the non-homogeneous case. The boundary condition gives \( B_p = 0 \) and as a result,

\[ -1 = \alpha A_p(C + 1) - (\alpha + \beta)A_pC + \beta A_p(C - 1) \]
\[ = A_p(\alpha - \beta) \]
\[ \Rightarrow A_p = \frac{-1}{\alpha - \beta} \]
\[ = \frac{1}{\beta - \alpha} . \]

This results in the general solution,

\[ u_C = A_h \left( 1 - \left( \frac{\beta}{\alpha} \right)^C \right) + \frac{C}{\beta - \alpha} . \quad (2.7) \]

Now consider the case where \( \alpha = \beta \). In this case the first term of Equation (2.7) will be 0 and the second term undefined, in which case the minimal non-negative solution will be \( u_C = \infty \).

In the case where \( \alpha > \beta \), the first term is always non-negative for \( A_h \geq 0 \), however the second term is negative for \( C > 0 \). In the limit as \( C \to \infty \), the first term tends to
and the second term tends to $-\infty$, so there is no finite value for $A_h$ for which the expression will always be non-negative. As a result the minimal non-negative solution will be $u_C = \infty$.

In the case where $\alpha < \beta$, the first term will always be non-negative for all $A_h \leq 0$ and the second term is non-negative for all $C$. Therefore a non-negative solution exists for all $C \in \mathbb{Z}$. The minimal non-negative solution will occur when $A_h = 0$ and so the solution is,

$$u_C = \frac{C}{\beta - \alpha},$$

for $\alpha < \beta$.

Assume that the Gambler starts with $C > 0$, what is the probability that eventually have $C' > C$ given that $\beta > \alpha$?

There are three equations that describe the situation,

$$f_C^{C'} = \alpha f_{C+1}^{C'} + (1 - \alpha - \beta)f_C^{C'} + \beta f_{C-1}^{C'},$$

$$f_0^{C'} = 0,$$

and

$$f_C^{C'} = 1.$$

The first equation is identical to the homogeneous equation above, so,

$$f_C^{C'} = A + B \left( \frac{\beta}{\alpha} \right)^C.$$  

Applying the boundary condition that $f_0^{C'} = 0$, gives,

$$f_C^{C'} = A \left( 1 - \left( \frac{\beta}{\alpha} \right)^C \right),$$

and applying the other boundary condition gives that,

$$A \left( 1 - \left( \frac{\beta}{\alpha} \right)^{C'} \right) = 1$$

$$\Rightarrow A = \frac{1}{1 - \left( \frac{\beta}{\alpha} \right)^{C'}}.$$

As a result,

$$f_C^{C'} = \frac{\left( \frac{\beta}{\alpha} \right)^C - 1}{\left( \frac{\beta}{\alpha} \right)^{C'} - 1}.$$
Chapter 3

Discrete Time Markov Chains with Hidden States

This chapter presents more advanced Markov chain constructions: the discrete phase-type distribution, a probability distribution defined by an absorbing Markov chains; matrix analytic methods, a method of generalising Markov chains so that they have a memory processes greater than one time-step; and the hidden Markov model, a Markov chain model with similarities to matrix analytic methods. As part of the discussion on the hidden Markov model, the Baum-Welch algorithm for fitting hidden Markov models to data is discussed. A novel method for fitting discrete phase-type distributions and discrete batch Markovian arrival processes to data using an adaptation of the Baum-Welch algorithm is then presented.

3.1 Discrete Phase-Type Distribution

A non-negative discrete random variable can be modelled by the number of steps until absorption in some Markov chain with a specified initial distribution. This is a very general construction known as a discrete phase-type distribution.

Consider a Markov chain \( \{J_k\}_{k \in \mathbb{Z}_+} \) on a state space \( X = \{0, 1, 2, \ldots, m\} \), where state 0
is an absorbing state and states 1, \ldots, m, for \( m \in \mathbb{Z}_+ \), are transient states. The Markov chain can be represented by the state diagram given in Figure 3.1 and described using the following probability transition matrix

\[
P = \begin{pmatrix} 1 & 0 \\ t & T \end{pmatrix},
\]

where the \( m \times m \) matrix has entries \( [T]_{ij} \geq 0, t_i \geq 0 \) for \( i, j \in \mathcal{X} \setminus \{0\} \) such that \( t + T1 = 1 \). The number of steps to absorption \( K \) is distributed by the discrete phase-type distribution, \( \text{DPH}(\tau, T) \), where \( \tau \) is an initial probability distribution such that \( \tau_i = \Pr[J_0 = i | i \neq 0] \), \( \tau_0 = \Pr[J_0 = 0] \), and \( \tau_0 + \tau 1 = 1 \). The \( k \)-step transition matrix,

\[
P^k = \begin{pmatrix} 1 & 0 \\ 1 - T^k1 & T^k \end{pmatrix}
\]

yields,

\[
\Pr[K \leq k] = \Pr[J_k = 0] = 1 - \tau T^k 1, \quad \text{for all } k \in \mathbb{Z}_+.
\]  

(3.1)

By differencing Equation (3.1), it can be shown that,

\[
\Pr[K = k] = \Pr[J_k = 0 | J_{k-1} \neq 0] = \tau T^{k-1} t.
\]

### 3.1.1 Moments of the Discrete Phase-Type Distribution

The moment generating function of the discrete phase-type distribution is,

\[
G(z) = \sum_{k \geq 0} z^k \Pr[K = k] = \tau_0 + \sum_{k \geq 1} z^k \tau T^{k-1} t = \tau_0 + z \sum_{k \geq 1} \tau (zT)^{k-1} t = \tau_0 + z \tau (I_m - zT)^{-1} t.
\]

Differentiating with respect to \( z \), \( n \) times, and setting \( z = 1 \) yields,

\[
\mathbb{E}[K(K-1) \ldots (K-n+1)] = n! \tau (I_m - T)^{-n} T^{n-1} 1.
\]
3.1.2 Properties of the Discrete Phase-Type Family

The family of discrete phase-type distributions has several properties which allow the DPH family to easily lend itself to many applications. Similar to the geometric distribution, the sum of any two DPH distributed random variables are themselves DPH distributed, any convex mixture of two DPH distributed random variables is also DPH distributed. In addition, any discrete distribution on the non-negative integers can be arbitrarily well approximated by a discrete phase-type distribution.

There is one important operation that needs to be defined and that in the Kronecker product $\otimes$. For two matrices $A$ and $B$ the Kronecker product is defined as,

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \ldots \\ a_{21}B & a_{22}B & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Closure Properties

The following theorems give the closure properties of the DPH family. They are stated without proof as the proofs can be found in Latouche and Ramaswami (1999, Section 2.6).

**Theorem 1** Let $X \sim DPH(\tau, T)$ and $Y \sim DPH(\sigma, S)$ be two independent random variables. Then the distribution of $Z = X + Y$ is given by $DPH(\alpha, A)$, where

$$\alpha = \begin{pmatrix} \tau \\ \tau_0 \sigma \end{pmatrix}$$

and

$$A = \begin{pmatrix} T & t \cdot \sigma \\ 0 & S \end{pmatrix}.$$

**Theorem 2** Let $X \sim DPH(\tau, T)$ and $Y \sim DPH(\sigma, S)$ be two independent random variables. Then the distribution of $Z = pX + (1-p)Y$, $0 \leq p \leq 1$ is given by $DPH(\alpha, A)$, where

$$\alpha = \begin{pmatrix} p\tau \\ (1-p)\sigma \end{pmatrix}$$

and

$$A = \begin{pmatrix} T & 0 \\ 0 & S \end{pmatrix}.$$

**Theorem 3** Let $X \sim DPH(\tau, T)$ and $Y \sim DPH(\sigma, S)$ be two independent random variables. Then the distribution of $Z = \min(X,Y)$ is given by $DPH(\alpha, A)$, where

$$\alpha = \tau \otimes \sigma$$

and

$$A = T \otimes S.$$

**Theorem 4** The discrete phase-type distribution is dense in the set of all discrete probability distributions supported on the natural numbers $(\mathbb{N})$. 
3.2 Matrix Analytic Methods

Matrix analytic methods (MAM) were first formalised by Neuts (1981) as an algorithmic method of solving many stochastic modelling problems. It formalises the basic Markov chain model by considering the states of the chain in hierarchical blocks usually referred to as levels and the states within the level as the phases of the process. The simplest MAM model is one where each level has only one phase, in this case it is the standard Markov chain model.

3.2.1 General MAM Models

The general matrix analytic method model is a doubly stochastic process \{(N_k, J_{N_k}) : k \in \mathbb{Z}^+_0\}, where \(N_k\) is the level of the process at time \(k\), and \(J_{N_k}\) is the phase of process at time \(k\) that can depend on the level \(N_k\) at time \(k\). Associated with each level \(N_k\) is a collection of states known as phases. The time between changes of the level \(N_k\) has some type of DPH distribution and changes of phase within a level correspond to phase changes within that DPH distribution and hence the phase process \(J_k\) is \(N_k\) dependent.

These models can be represented in the form of probability matrices with a block structure, by allowing the rows and column of blocks to represent the level of the system and the rows and columns within each block, the phases of the Markov chain associated with each level.

Discrete Phase-type Renewal Process

A simple example of a matrix analytic model is the phase-type renewal process. A renewal process is a process in which the distribution of time between events are independent and identically distributed. The simplest example of a renewal process is the geometric process, the counting of the number of successes of a sequence of Bernoulli trial; for example, counting the number of time heads shows up in successive coin flips. Let \(p\) be the probability of a success, than the geometric process is described by the infinite Markov chain with probability transition matrix,

\[
P = \begin{pmatrix}
1 - p & p & 0 & 0 & \cdots \\
0 & 1 - p & p & 0 & \cdots \\
0 & 0 & 1 - p & p & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots 
\end{pmatrix}
\]

In a phase-type renewal process, \(N_k\) counts the number of times a phase-type variable, \(DPH(\tau, T)\), enters the absorbing state and \(J_k\) represents the phase of a phase-type
3.2. MATRIX ANALYTIC METHODS

distribution at time \(k\), \(X/\{0\}\). This process is represented by the infinite probability transition matrix,

\[
P = \begin{pmatrix}
A_0 & A_1 & 0 & 0 & \cdots \\
0 & A_0 & A_1 & 0 & \cdots \\
0 & 0 & A_0 & A_1 & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
\end{pmatrix}
\] (3.2)

where \(A_0 = T\) and \(A_1 = t\tau\), an \(m \times m\) matrix. Immediately after each renewal, the discrete phase-type distribution immediately recommences with distribution \(\tau\).

This matrix contains sub-matrices which describe the DPH distribution for the times between renewals. A renewal will occur when there is a transition in the submatrix \(t\tau\), which will be referred to as \textit{marked} transitions. When a marked transition occurs, \(N_k\) will go from \(i \in \mathbb{N}\) to \(i + 1\). The transitions within the sub-matrix \(T\) that do not correspond to a renewal, will be referred to as \textit{unmarked} transitions. This allows the renewal times to have a distribution other than the geometric distribution. Note that a geometric distribution is the simplest form of a DPH distribution, \(DPH(1, 1 - p)\).

**Discrete Markovian Arrival Process**

In a more general setting, the marked transition do not have to correspond to renewal points and are applicable when arrivals are not renewals but just special transitions. This can be modelled using the discrete Markovian arrival process, where \(N_k\) counts the number of arrivals and \(J_k\) tracks the state of an underlying Markov chain. The discrete time Markov chain described by a finite state probability transition matrix \(A\) which can be filtered into two matrices \(A_0\) and \(A_1\) such that \(A = A_0 + A_1\). The matrix \(A_1\) are transitions in the Markov chain that correspond to an arrival, and the matrix \(A_0\) are the other transitions of the chain. The only restriction on the form of the matrices \(A_0\) and \(A_1\) is that \(0 \leq [A_n]_{ij} \leq 1\) for \(n \in \{0, 1\}\), such that \((A_0 + A_1) 1 = 1\). The matrix \(A_1\) is not in general rank 1 or of the form \(t\tau\), except in the phase-type renewal case, however its rows give the starting distribution of the phase-type distribution that describes the next inter-arrival time. Here the time between arrivals will not necessarily be independent or identically distributed, but will still have some sort of phase-type distribution from the family of phase-type distributions described by the matrix \(A_0\) and the state of the underlying Markov chain in which the last arrival time occurred. The arrival process can be expressed in the same form as the infinite probability transition matrix shown in Equation (3.2), except \(A_1\) is no longer restricted to be of rank 1.

**Discrete Batch Markovian Arrival Process**

Discrete Markovian arrival process (DMAP) is a special case of the discrete batch Markovian process (DBMAP) where the stochastic process \(N_k\) is allowed to increase
CHAPTER 3. DTMCS WITH HIDDEN STATES

by any finite positive integer. The process $J_k$ is described by the finite state probability transition matrix $A = \sum_{\ell \in \mathbb{Z}_+} A_\ell$, where $A_{\ell > 0}$ are the transitions of marking an arrival of size $\ell > 0$ and $A_0$ is the same as before. The special case where $\max\{\ell\} = 1$ is the Markovian arrival process. The restrictions on the elements of $A_\ell$ are that, $0 \leq [A_\ell]_{ij} \leq 1$, such that $A_1 1 = 1$. The arrival process can be expressed in terms of the infinite probability transition matrix,

$$P = \begin{pmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & \cdots \\ 0 & A_0 & A_1 & A_2 & A_3 & \cdots \\ 0 & 0 & A_0 & A_1 & A_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \quad (3.3)$$

Discrete Quasi-Birth and Death Process

Given that the level $N_k$ can increase as discussed, it can also be allowed to decrease. A discrete quasi-birth and death process is another special case of a matrix analytic model where $N_k$ is only allowed to move skip-free up or down one level. That is, any transition will cause $N_k$ to, increase by one, decrease by one, or remain unchanged. The process $\{J_{N_k}\}$ is described by the probability transition matrix $D_1 = B_1 + A_0 + A_1$ when $N_k > 0$ and $D_0 = B_0 + A_1$ when $N_k = 0$, with the restrictions that $0 \leq [A_n]_{ij}, [B_n]_{ij} \leq 1$ for $n \in \{0, 1\}$, such that $D_0 1 = D_1 1 = 1$. The homogeneous quasi-birth and death process can be expressed in terms of the infinite block tri-diagonal probability transition matrix,

$$P = \begin{pmatrix} B_0 & A_1 & 0 & 0 & \cdots \\ B_1 & A_0 & A_1 & 0 & \cdots \\ 0 & B_1 & A_0 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Latouche and Ramaswami (1999, Chap. 13) showed that single server queues with general inter-arrival time distribution with a Markovian distributed server, $GI/M/1$ queues, or Markovian inter-arrival time distribution with a generally distributed server, $M/G/1$ queue, can be encompassed into the QBD structure. All of the previous structures can be allowed to have level dependence and therefore can be described as inhomogeneous. For example the homogeneous quasi-birth and death process,

$$P = \begin{pmatrix} B_0 & A_{01} & 0 & 0 & \cdots \\ B_{11} & A_{10} & A_{11} & 0 & \cdots \\ 0 & B_{21} & A_{20} & A_{21} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$
3.3 Hidden Markov Model

A hidden Markov model is a doubly stochastic process \( \{(Y_k, J_k) : k \in \mathbb{Z}_+\} \), where \( Y_k \) is an observable process whose distribution is dependent on the state of some underlying Markov process \( J_k \). However only the transitions of \( Y_k \) are observed, which are referred to as emissions or observations in the HMM literature. The states of \( J_k \) are hidden or unobserved and represent the transitions of some underlying Markov chain at time \( k \).

A hidden Markov model can be described by five quantities:

- \( X \) the state space of the hidden Markov process \( J_k \),
- \( O \) the set of observations for the observable process \( Y_k \),
- \( P \) the Markov probability transition matrix \( P : X \to X \),
- \( Q \) the probability transition matrix \( Q : X \to O \),
- \( \pi \) the initial distribution of \( J_0 \) on \( X \).

In this thesis \( X \) and \( O \) will both be discrete and finite, as will time \( k \), it should be noted that they do not have to be. For a more formal and general discussion of hidden Markov models see Cappé et al. (2005).

Definitions, Notation, and Assumptions

\( \{J_k : k \in \mathbb{Z}_+\} \) is a hidden Markov process on the states \( X = \{1, \ldots, N\} \), where \( N \) is the number of elements in \( X \). \( \{Y_k : k \in \mathbb{Z}_+\} \) is an observed process on states \( O = \{1, \ldots, M\} \), where \( M \) is the number of possible observations.

Let \( Y_0, Y_1, \ldots, Y_K \), where \( Y_k \in O \), be the sequence of observations from time \( k = 0 \) up to time \( k = K \), and let \( J_0, J_1, \ldots, J_K \), where \( J_k \in X \), be a sequence of possible hidden states for the observation \( Y_k \).

Note: As \( J_k \) is unobserved it is by no means unique for each \( Y_k \), but that some sequences of \( J_0, J_1, \ldots, J_K \) are more likely than others.

To prevent odious notation, let,

\[
Y = \{Y_0, Y_1, \ldots, Y_K\}
\]

be an observed sequence and assume that,

\[
J = \{J_0, J_1, \ldots, J_K\}
\]

is the hidden sequence corresponding to \( Y \). The partial sequences of observation will be denoted by,

\[
\bar{Y}_k = \{Y_0, Y_1, \ldots, Y_k\}
\]
and
\[ \vec{Y}_k = \{Y_k, Y_{k+1}, \ldots, Y_K\}. \]
Likewise the partial sequence of hidden transition will be denoted by,
\[ \vec{J}_k = \{J_0, J_1, \ldots, J_k\} \]
and
\[ \vec{J}_k = \{J_k, J_{k+1}, \ldots, J_K\}. \]

There are two independence assumptions for hidden Markov processes. It is assumed that the \( t^{th} \) hidden random variable, given the \( (t-1)^{th} \) hidden random variable, is independent of all other previous hidden random variables,
\[
\Pr(J_k | \vec{J}_{k-1}) = \Pr(J_k | J_{k-1}),
\]
and hence the process is Markovian. For the observations, it is assumed that the \( t^{th} \) observation, given the \( t^{th} \) hidden random variable, is independent all other observation and hidden random variables, ie,
\[
\Pr(Y_k | J, \vec{Y}_{k-1}, \vec{Y}_{k+1}) = \Pr(Y_k | J_k).
\]
It is also assumed that the hidden Markov process is time-homogeneous, such that the \( N \times N \) stochastic matrix \( P \) describes the transition of the process \( \{J_k : k \in \mathbb{Z}_+\} \) which has elements,
\[
[P]_{ij} = \Pr[J_k = j | J_{k-1} = i] = p_{ij} \quad \forall i, j \in X, \forall k \in \mathbb{Z}_+.
\]
Let \( \pi \) be the initial distribution across \( X \), a vector of length \( N \) with elements,
\[
\pi_i = \Pr[J_0 = i], \quad \forall i \in X.
\]
It is also assumed that the observation probabilities are time-homogeneous for each state. As a result a \( N \times M \) stochastic probability matrix \( Q \) describes the probability of each observation in \( O \) given a state in \( X \), which has elements,
\[
[Q]_{ij} = \Pr[Y_k = j | J_k = i] = q_{ij} \quad \forall i \in X, \forall j \in O, \forall k \in \mathbb{Z}_+.
\]

**Example**

For example, consider a Markov chain with two hidden states; clement, inclement: \( X = \{\ominus, \odot\} \); and three possible observations; sunny, overcast, rain : \( O = \{\odbl, \dodbl, \ominus\} \). The transition probabilities between the states are given in Table 3.1 and the observation probabilities are given in Table 3.2.
### Table 3.1: Transition probability matrix $P$ for the transitions between hidden states.

<table>
<thead>
<tr>
<th></th>
<th>☁️</th>
<th>☀️</th>
</tr>
</thead>
<tbody>
<tr>
<td>☁️</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>☀️</td>
<td>0.4</td>
<td>0.6</td>
</tr>
</tbody>
</table>

### Table 3.2: Observation probability matrix $Q$ for the observations given the hidden states.

<table>
<thead>
<tr>
<th></th>
<th>☁️</th>
<th>☀️</th>
<th>☂️</th>
</tr>
</thead>
<tbody>
<tr>
<td>☁️</td>
<td>0.70</td>
<td>0.25</td>
<td>0.05</td>
</tr>
<tr>
<td>☀️</td>
<td>0.15</td>
<td>0.20</td>
<td>0.65</td>
</tr>
</tbody>
</table>

The process can be represented in diagrammatic form such as in Figure 3.2, with the colored line representing the observations from each state and the black lines the transitions between the hidden states.

A HMM also requires a starting distribution for the hidden process, in this example let $\pi = (0.3, 0.7)$.

![Figure 3.2: A diagram of a hidden Markov model](image)

An example of a sample path of the sequence $\{Y_k\}$ (simulated from this model) is:

☁️, ☁️, ☀️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️, ☁️.

An observer sees the sequence of sunshine, overcast, and rain, not the underlying hidden states, represented here by the weather’s colouring. In general, at time $k$ the observations $Y_0, Y_1, \ldots, Y_k \in O$ are known, however the hidden states that produced these observations, $J_0, J_1, \ldots, J_k \in X$, are unknown.

### 3.3.1 Relation to Matrix Analytic Methods

Hidden Markov models (HMM), in the fully discrete case, can be viewed a special case of matrix analytic models described in Section 3.2, although they were developed
independently. A hidden Markov model is a more structured batch Markovian arrival process if the observations are \( Y_k \in \mathbb{Z}_+ \) and the process \( J_k \) is stationary. This gives a matrix analytic model that has a reduced number of parameters due to the phases being homogeneous in both level and arrival size. The counting process \( N_k \), which counts the number of arrivals, is then described by the probability transition matrix shown in Equation (3.3), where,

\[
A_n = \begin{pmatrix}
  p_{11}q_{1n} & \cdots & p_{1N}q_{1n} \\
  \vdots & \ddots & \vdots \\
  p_{N1}q_{Nn} & \cdots & p_{NN}q_{Nn}
\end{pmatrix}.
\]

### 3.3.2 Fitting Hidden Markov Models: The Baum-Welch Algorithm

Rabiner (1989) considers the problem of fitting hidden Markov models as a three step problem, given the clarity with which the author explained the topic, the same general structure will be followed here.

**Given \( Y \) and model parameters \( P, Q, \pi \), calculate \( \Pr[Y|P,Q,\pi] \)**

Denote the probability \( q_{J_kY_k} \) as the probability that the \( k^{th} \) observation of \( Y \) was \( Y_k \in O \) given that the \( k^{th} \) hidden state of \( J \) was \( J_k \in X \). Given \( P, Q \) and \( \pi \), the probability that \( Y \) was observed given that the sequence \( J \) occurred is

\[
\Pr[Y|J,P,Q,\pi] = \prod_{k=0}^{K} \Pr[Y_k|J_k,P,Q,\pi],
\]

by the assumption of independence of observations. The probability that the sequence \( J \) occurred given \( P, Q \), and \( \pi \) is

\[
\Pr[J|P,Q,\pi] = \prod_{k=1}^{K} \Pr[J_k|J_{k-1},P,Q,\pi] = \pi_{J_0}p_{J_0J_1}\cdots p_{J_{K-1}J_K}.
\]

By conditional probability, the probability that both \( Y \) and \( J \) occurred given \( P, Q \), and \( \pi \) is then,

\[
\Pr[Y,J|P,Q,\pi] = \Pr[Y|J,P,Q,\pi] \Pr[J|P,Q,\pi]. \tag{3.4}
\]

Summing Equation (3.4) over all possible sequences \( J \) will yield \( \Pr[Y|P,Q,\pi] \). However, there are \( N^{K+1} \) possible sequences for \( J \) and as a result such a calculation would be impractical.

Leonard Baum developed two procedures for calculating \( \Pr[Y|P,Q,\pi] \) without the need for extensive computation (Baum and Eagon 1967 and Baum and Sell 1968) known as the Forward Procedure and the Backward Procedure.
3.3. HIDDEN MARKOV MODEL

Forward Procedure: Define,
\[ \alpha_k(i) = \Pr[\tilde{Y}_k, J_k = i|P, Q, \pi] , \]
the probability of being in hidden state \( i \in X \) at time \( k \) and observing the sequence \( \tilde{Y}_k \) given \( P, Q, \) and \( \pi \).

The probability that the hidden process is in state \( i \) at time 0 is \( \pi_i \), therefore the probability that observation \( Y_0 \) occurred given that the first state was \( i \) is given by
\[ \alpha_0(i) = \pi_i q_{Y_0} . \]

The probability the hidden state is \( j \) at time \( k \), given the observations \( \tilde{Y}_{k-1} \), is given by the probability the hidden process is in state \( i \) at time \( k - 1 \), given the observations \( \tilde{Y}_{k-1} \), multiplied by the probability of transition from state \( i \) to state \( j \), summed over all the possible states \( i \in X \), expressed mathematically as,
\[ \Pr[\tilde{Y}_{k-1}, J_k = j|P, Q, \pi] = \sum_{i=1}^{N} \alpha_{k-1}(i)p_{ij} . \]

Therefore the probability that \( \tilde{Y}_k \) is observed and the process is in the hidden state \( j \) at time \( k \) is given by
\[ \alpha_k(j) = \left[ \sum_{i=1}^{N} \alpha_{k-1}(i)p_{ij} \right] q_{jY_k} , \]
for all \( j = 1, \ldots, N \) and all \( k = 1, \ldots, K \), for some final time \( K \) given by the time of the last observation.

The desired result can be found by summing over all \( i \in X \) at final time \( K \),
\[ \Pr[Y|P, Q, \pi] = \sum_{i=1}^{N} \alpha_K(i) . \]

Backward Procedure: Define,
\[ \beta_k(i) = \Pr[\tilde{Y}_{k+1}|J_k = i, P, Q, \pi] , \]
the probability of observing the sequence \( \tilde{Y}_{k+1} \) given the hidden process being in the hidden state \( i \in X \) at time \( k \) and \( P, Q, \pi \).

If then,
\[ \beta_K(j) = 1, \]
then for all \( i = 1, \ldots, N, \)
\[ \beta_{k-1}(i) = \sum_{j=1}^{N} p_{ij} q_{jY_k} \beta_k(j) , \]
for all \( k = K, K - 1, \ldots, 1. \)
The desired result can be found by,

\[
\Pr[Y|P, Q, \pi] = \sum_{i=1}^{N} \pi_i q_{Y_0}{\beta}_0(i).
\]

Note: It can also be written that,

\[
\Pr[Y|P, Q, \pi] = \sum_{i=1}^{N} \Pr[Y, J_k = i|P, Q, \pi] \Pr[Y_{k+1}|J_k = i, P, Q, \pi] = \sum_{i=1}^{N} \alpha_k(i)\beta_k(i),
\]

for all \(k = 0, 1, \ldots, K\).

Given \(Y\) and model parameters \(P, Q, \pi\) find the “most likely” \(J\)

Finding the “most likely” state \(J_k\) a time \(k\) given \(Y, P, Q,\) and \(\pi\) is relatively straightforward,

\[
\Pr[J_k = i|Y, P, Q, \pi] = \frac{\Pr[Y, J_k = i|P, Q, \pi]}{\Pr[Y|P, Q, \pi]} = \frac{\Pr[Y_k, J_k = i|P, Q, \pi] \Pr[Y_{k+1}|J_k = i, P, Q, \pi]}{\Pr[Y|P, Q, \pi]} = \frac{\alpha_k(i)\beta_k(i)}{\sum_{i=1}^{N} \alpha_k(i)\beta_k(i)}. \tag{3.5}
\]

The most likely state \(J_k\) at time \(k\) is then,

\[
\arg\max_{i \in X} \left\{ \frac{\alpha_k(i)\beta_k(i)}{\sum_{i=1}^{N} \alpha_k(i)\beta_k(i)} \right\}.
\]

This however will not yield the most likely sequence \(J\). Consider for example if that the most likely state at time \(k - 1\) was \(i\) and the most likely state at time \(k\) was \(j\), but the probability \(p_{ij} = 0\), then such a sequence could not occur.

Viterbi (1967) instead found the “mostly likely” sequence \(J\) based on the observations \(Y\) given \(P, Q,\) and \(\pi\).
3.3. **HIDDEN MARKOV MODEL**

Define the highest probability of a path to state \( i \) at time \( k \) based on observations \( Y_0, Y_1, \ldots, Y_k \) given \( P, Q, \) and \( \pi \) as,

\[
\gamma_k(i) = \max_{J_{k-1}} \left\{ \Pr[Y_k, J_{k-1}, J_k = i | P, Q, \pi] \right\}.
\]

(3.6)

The probability of starting in state \( i \) with observation \( Y_0 \) is given by

\[
\gamma_0(i) = \pi_i q_{iY_0}.
\]

It can be shown (Appendix A.1.1) that an iterative equation for \( \gamma_k(j) \) is

\[
\gamma_k(j) = \left[ \max_i \{ \gamma_{k-1}(i)p_{ij} \} \right] q_{jY_k}.
\]

(3.7)

Let \( \psi_k(j) \) be the hidden state \( i \in X \) at time \( k - 1 \) that maximises the probability of being in hidden state \( j \in X \) at time \( k \), given the observations \( Y_{k-1} \),

\[
\psi_k(j) = \arg\max_{i \in X} \{ \gamma_{k-1}(i)p_{ij} \}.
\]

The Viterbi state, \( J_{k-1}^* \), is the state of \( J \) at time \( k - 1 \) in the most likely sequence. The Viterbi state at final observation time \( K \) is,

\[
J_K^* = \arg\max_{i \in X} \{ \gamma_K(i) \}
\]

and the most likely sequence can be found recursively for all \( k = 1, \ldots, k-1 \) by,

\[
J_{k-1}^* = \psi_k(J_k^*).
\]

This is the most likely or Viterbi sequence, given \( Y, P, Q, \) and \( \pi \).

**Find \( P, Q, \pi \) to maximise \( \Pr[Y|P, Q, \pi] \)**

The most important question is that given a set of observations, can estimates for \( P, Q, \) and \( \pi \) be found that best describe that data given that \( P, Q, \) and \( \pi \) are non-unique? The answer is that whilst they can, these estimates are in terms of the parameters \( P, Q, \) and \( \pi \) themselves.

The probability that a transition from state \( i \) to state \( j \) occurs at time \( k + 1 \) is denoted,

\[
\xi_k(i, j) = \Pr[J_k = i, J_{k+1} = j | Y, P, Q, \pi].
\]

(3.8)

It can be shown that (Appendix A.1.2) Equation (3.8) can be written as,

\[
\xi_k(i, j) = \frac{\alpha_k(i)p_{ij}q_{jY_{k+1}}\beta_{k+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_k(i)p_{ij}q_{jY_{k+1}}\beta_{k+1}(j)}.
\]

(3.9)
The probability that the process is in state $i$ at time $k$ was shown earlier in Equation (3.5) to be,

$$\eta_k(i) = \frac{\alpha_k(i) \beta_k(i)}{\sum_i^N \alpha_k(i) \beta_k(i)}.$$ 

The sum of $\xi_k(i,j)$ over all $k = 0, \ldots, K - 1$ is the expected number of transitions from state $i$ to state $j$ given $Y$, $P$, $Q$, and $\pi$. Similarly, the sum of $\eta_k(i)$ over all $k = 0, \ldots, K - 1$ is the expected number of transitions from state $i$ given $Y$, $P$, $Q$, and $\pi$.

Given the data $Y$ and parameters $P$, $Q$, and $\pi$, the best estimates for the parameters are given by $\hat{P}$, $\hat{Q}$, and $\hat{\pi}$ respectively,

$$\hat{\pi}_i = \text{Pr}[J_0 = 0|Y, P, Q, \pi] = \eta_0(i),$$

$$\hat{p}_{i,j} = \frac{\sum_{k=0}^{K-1} \text{Pr}[J_k = i, J_{k+1} = j|Y, P, Q, \pi]}{\sum_{k=0}^{K-1} \text{Pr}[J_k = i|Y, P, Q, \pi]} = \frac{\sum_{k=0}^{K-1} \xi_k(i,j)}{\sum_{k=0}^{K-1} \eta_k(i)},$$

$$\hat{q}_{i,j} = \frac{\sum_{k=0}^{K} \text{Pr}[J_k = i|Y_k = j, Y, P, Q, \pi]}{\sum_{k=0}^{K} \text{Pr}[J_k = i|Y, P, Q, \pi]} = \frac{\sum_{k=0}^{K} \eta_k(i) \delta(Y_k - j)}{\sum_{k=0}^{K} \eta_k(i)},$$

where $\delta(n)$ is the Kronecker delta given by,

$$\delta(n) := \begin{cases} 
1 & \text{if } n = 0, \\
0 & \text{if } n \neq 0.
\end{cases}$$

**The Baum-Welch Algorithm**

The Baum-Welch algorithm, proposed by Lloyd Welch (Welch, 2003) and developed over a number of years by Leonard Baum (Baum and Petrie 1966, Baum and Eagon 1967, Baum and Sell 1968, Baum et al. 1970, and Baum 1972), provides a means for finding a locally optimal solution to,

$$\arg\max_{P, Q, \pi} \{\text{Pr}[Y|P, Q, \pi]\}.$$ 

The algorithm works iteratively by updating estimates for $P$, $Q$, and $\pi$. The $\nu^{th}$ estimate for $P$, $Q$, and $\pi$ will be denoted $\hat{P}^\nu$, $\hat{Q}^\nu$, and $\hat{\pi}^\nu$ respectively. The first step is the expectation step. Using the current estimates $\hat{P}^\nu$, $\hat{Q}^\nu$, and $\hat{\pi}^\nu$, calculate the expected value of the likelihood function with respect to the conditional distribution of $J$ given the data and parameter estimates, and set,

$$B(P, Q, \pi|\hat{P}^\nu, \hat{Q}^\nu, \hat{\pi}^\nu) = E_{J|Y, P^\nu, Q^\nu, \pi^\nu} [\text{Pr}[Y|P, Q, \pi]]$$
3.3. HIDDEN MARKOV MODEL

\[
= E_{J \mid Y, \hat{P}^{\nu}, \hat{Q}^{\nu}, \hat{\pi}^{\nu}} \left[ \sum_{J \in X} \Pr[Y, J \mid P, Q, \pi] \right].
\]

Whilst this equation looks unwieldy, it simply gives the most likely sequence \( J \). The next step is the maximisation step. The problem is to find new estimates,

\[
\hat{P}^{\nu+1}, \hat{Q}^{\nu+1}, \hat{\pi}^{\nu+1} = \arg\max_{P, Q, \pi} \left\{ B(P, Q, \pi \mid \hat{P}^{\nu}, \hat{Q}^{\nu}, \hat{\pi}^{\nu}) \right\},
\]

where these new estimates are used to find the expectation once again and then the process is repeated. Baum showed that \( \Pr[Y \mid \hat{P}^{\nu+1}, \hat{Q}^{\nu+1}, \hat{\pi}^{\nu+1}] \geq \Pr[Y \mid \hat{P}^{\nu}, \hat{Q}^{\nu}, \hat{\pi}^{\nu}] \) and that the algorithm converges to a locally optimal solution. A pseudo-code of the Baum-Welch algorithm for fitting hidden Markov models to data is shown in Algorithm 3.1.

Algorithm 3.1: Pseudo-code for the Baum-Welch algorithm for fitting HMMs.

**The Baum-Welch Algorithm:**

1. Set \( \nu = 0 \) with initial estimates of \( \hat{P}^{0}, \hat{Q}^{0}, \hat{\pi}^{0} \).
2. Calculate:
   \[
   \xi^{\nu}_{k}(i, j) = \Pr[J_{k} = i, J_{k+1} = j \mid Y, \hat{P}^{\nu}, \hat{Q}^{\nu}, \hat{\pi}^{\nu}]
   \]
   \[
   \eta^{\nu}_{k}(i) = \Pr[J_{k} = i \mid Y, P^{\nu}, Q^{\nu}, \pi^{\nu}]
   \]
3. Update parameters to:
   \[
   \hat{\pi}^{\nu+1} = \eta^{\nu}_{0}(i)
   \]
   \[
   \hat{p}^{\nu+1}_{ij} = \frac{\sum_{k=0}^{K-1} \xi^{\nu}_{k}(i, j)}{\sum_{k=0}^{K-1} \eta^{\nu}_{k}(i)}
   \]
   \[
   \hat{q}^{\nu+1}_{ij} = \frac{\sum_{k=0}^{K} \eta^{\nu}_{k}(i) I(Y_{k} = j)}{\sum_{k=0}^{K} \eta^{\nu}_{k}(i)}
   \]
   \[
   \nu \leftarrow \nu + 1
   \]
4. Repeat steps 2 and 3 until either \( \nu \) reaches a desired value or until the change in the likelihood function falls below a set threshold.

**Notes on fitting the data with the Baum-Welch algorithm**

1. The likelihood is only locally optimal. Different starting parameters may yield different estimates after the algorithm converges. These different estimates may have different likelihoods.
2. The estimates are non-unique. Different estimated parameters may have the same likelihood.

3. If an initial parameter, $\hat{p}^0_{ij}$, $\hat{q}^0_{ij}$, or $\hat{\pi}_i^0$ is set to zero, for every estimate the parameter will remain zero.

4. The number of states $N$ must be chosen in advance, it is usually best to choose a number less than the maximal observation $M = \max\{Y_k\}$.

### 3.4 Adaptation of the Baum-Welch Algorithm for the Fitting of Matrix Analytic Models

The Baum-Welch algorithm is readily adaptable to the fitting of discrete-time matrix analytic models. Despite this, a literature search turns up no published papers where this has been done. However, continuous-time examples do exist using the more general expectation-maximisation (EM) algorithm of which the Baum-Welch was an early example (Dempster et al., 1977). Breuer (2002) and Klemm et al. (2003) have apparently independently arrived at the same method for fitting continuous-time BMAPs using the EM algorithm. Asmussen et al. (1996) found a method for fitting continuous-time phase-type distributions also using the EM algorithm and Olsson (1996) discussed EM fitting for continuous-time phase-type distributions with censored data. The free software EMpht (Olsson, 1998) will fit continuous-time phase-type distributions to both censored and uncensored data. In this section methods for fitting discrete-time BMAPs and PH distributions using adapted Baum-Welch algorithms are presented for completeness.

#### 3.4.1 Fitting Discrete-Time Batch Markovian Arrival Processes

Let $Y = \{Y_0, Y_1, \ldots, Y_K\}$ be a set of $K + 1$ observed arrivals (or observations) each of batch size $Y_k \in \mathbb{Z}_+$ and let the maximal batch size $M = \max\{Y_k\}$ the maximum batch size from the observations. It is important to note that no arrival at time $k$ is an observation, that is an arrival of batch size $Y_k = 0$. For an initial distribution $\pi$ and DBMAP process $A_0, A_1, \ldots, A_M$ the probability of observing such a sequence is,

$$\Pr[Y | \pi, A_0, A_1, \ldots, A_M] = \pi A_{Y_0} A_{Y_1} \ldots A_{Y_k} \mathbf{1}.$$ 

It is assumed that process is stationary, so that

$$\Pr[Y | \pi, A_0, A_1, \ldots, A_M] = \Pr[Y | A_0, A_1, \ldots, A_M]$$

and $\sum_{t=0}^{M} \pi A_t = \pi \sum_{t=0}^{M} A_t = \pi$. 

3.4. ADAPTATION OF THE BAUM-WELCH ALGORITHM

The forward probabilities are then defined as,

\[
\alpha_0(j) = \sum_i \pi_i [A_{Y_0}]_{i,j}, \quad \forall j \in X,
\]

\[
\alpha_k(j) = \sum_i \alpha_{k-1}(i) [A_{Y_k}]_{i,j}, \quad \forall j \in X, k = 1, \ldots, K.
\]

Similarly the backward probabilities are defined as,

\[
\beta_{K+1}(i) = 1, \quad \forall i \in X,
\]

\[
\beta_{k-1}(j) = \sum_i [A_{Y_k}]_{i,j} \beta_k(j), \quad \forall j \in X, k = 1, \ldots, K.
\]

Similarly,

\[
\xi_k(i, j) = \frac{\alpha_k(i) [A_{Y_{k+1}}]_{i,j} \beta_{k+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_k(i) [A_{Y_{k+1}}]_{i,j} \beta_{k+1}(j)} \quad (3.10)
\]

\[
\eta_k(i) = \frac{\alpha_k(i) \beta_k(i)}{\sum_{i=1}^N \alpha_k(i) \beta_k(i)} \quad (3.11)
\]

The fitting procedure then follows the same format as the Baum-Welch algorithm as shown in Algorithm 3.2. The notes about fitting the Baum-Welch algorithm for HMMs are also relevant for the fitting of DBMAPS.

Algorithm 3.2: Pseudo-code for a modified Baum-Welch algorithm for fitting DBMAPs

<table>
<thead>
<tr>
<th>Modified Baum-Welch Algorithm for discrete batch Markovian arrival processes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set ( \nu = 0 ) with initial estimates ( \hat{A}_0^0, \hat{A}_1^0, \ldots, \hat{A}_M^0 ), and .</td>
</tr>
<tr>
<td>2. Use ( \hat{A}_0^\nu, \hat{A}_1^\nu, \ldots, \hat{A}_M^\nu ) to evaluate Equation (3.10) and Equation (3.11) for all ( i, j = 1, \ldots, N ) and ( k = 0, \ldots, K ).</td>
</tr>
</tbody>
</table>
| 3. Update parameters using:

\[
[\hat{A}_l^\nu]_{ij} = \frac{\sum_{k=0}^K \xi_k(i, j) \delta(Y_k - l)}{\sum_{k=0}^K \eta_k(i)}.
\]

4. Repeat steps 2 and 3 until either \( \nu \) reaches a desired value or until the change likelihood function falls below a set threshold.
3.4.2 Fitting Discrete-Time Phase-Type Distributions

Let \( Y = \{ Y_1, Y_2, \ldots, Y_M \} \) be a set of \( M \) observed times until absorption of a discrete time phase-type Markov chain, \( DPH(\tau, T) \), where the times \( Y_i \in \mathbb{Z}_+ \). The probability of observing the random variable \( Y_i \) given \( \tau \) and \( T \) is,

\[
\Pr[Y_i|\tau, T] = \tau T^{Y_i - 1} t.
\]

Using the previous notation for the phase-type distribution given in Section 3.1 the forward and backward probabilities can be defined in terms of the transient states \( 1, \ldots, m \).

The forward probabilities are,

\[
\alpha_k^l(j) = \tau_i, \quad \forall j \in X,
\]

\[
\alpha_k^l(j) = \sum_i \alpha_{k-1}(i) T_{i,j}, \quad \forall j \in X, k = 1, \ldots, Y_l - 1.
\]

The backward probabilities (steps backwards from absorption) are,

\[
\beta_{Y_l}^l(i) = t_i, \quad \forall i \in X,
\]

\[
\beta_k^l(j) = \sum_i T_{i,j} \beta_{k+1}(j), \quad \forall j \in X, k = 1, \ldots, Y_l - 1.
\]

Having obtained the forward and backward probabilities,

\[
\Pr[Y_l|\tau, T] = \sum_{i=1}^m \alpha_{Y_l-1}^l(i) t_i = \sum_{i=1}^m \tau_i \beta_1^l(i).
\]

The probability of transitioning from state \( i \in X \setminus \{0\} \) to state \( j \in X \setminus \{0\} \) and the probability of being in state \( i \in X \setminus \{0\} \) at time \( k \) are given by

\[
\xi_k^l(i, j) = \frac{\alpha_k(i) T_{i,j} \beta_{k+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_k(i) T_{i,j} \beta_{k+1}(j)}, \quad \forall i, j \in X \setminus \{0\},
\]

\[
\eta_k^l(i) = \frac{\alpha_k(i) \beta_k(i)}{\sum_{i=1}^N \alpha_k(i) \beta_k(i)}, \quad \forall i \in X \setminus \{0\}.
\]

The probability of starting is state \( i \) given \( \tau, T, \) and \( Y_l \) is given by,

\[
\varpi^l(i) = \frac{\tau_i \beta_1^l(i)}{\sum_{i=1}^m \tau_i \beta_1^l(i)}, \quad \forall i \in X \setminus \{0\}.
\]
3.4. ADAPTATION OF THE BAUM-WELCH ALGORITHM

The probability of being absorbed at time $Y_l$ from state $i$ given $\tau$, $T$, and that absorption has not occurred up until time $Y_l-1$ is given by,

$$\omega^l(i) = \frac{\alpha^l_{Y_l-1}(i)T_i}{\sum_{i=1}^{m} \alpha^l_{Y_l-1}(i)T_i}, \quad \forall i \in X \setminus \{0\}.$$ 

The total expected number of transitions from $i$ to $j$ given $\tau$, $T$, and $Y$ is

$$\xi(i, j) = \sum_{l=1}^{M} \sum_{k=1}^{Y_l-1} \xi^l_k(i, j).$$ \hspace{1cm} (3.12)

The total expected number of times the process is in $i$ given $\tau$, $T$, and $Y$ is

$$\eta(i) = \sum_{l=1}^{M} \sum_{k=1}^{Y_l} \eta^l_k(i).$$ \hspace{1cm} (3.13)

The total expected number of times the process begins in $i$ given $\tau$, $T$, and $Y$ is

$$\varpi(i) = \sum_{l=1}^{M} \varpi^l(i).$$ \hspace{1cm} (3.14)

The total expected number of times the process is absorbed from $i$ given $\tau$, $T$, and $Y$ is

$$\omega(i) = \sum_{l=1}^{M} \omega^l(i).$$ \hspace{1cm} (3.15)

The Baum-Welch algorithm is then modified for the discrete phase-type distribution as shown in Algorithm 3.3. The notes about fitting the Baum-Welch algorithm for HMMs are also relevant to the fitting of discrete phase-type distributions.

Algorithm 3.3: Pseudo-code for a modified Baum-Welch algorithm for fitting DPHs.

\begin{center}
\begin{tabular}{|l|}
\hline
\textit{Modified Baum-Welch Algorithm for discrete phase-type distributions:} \\
\hline
1. Set $\nu = 0$ with initial estimates $\hat{\tau}^0, \hat{T}^0, \hat{t}^0$. \\
2. Use $\hat{\tau}^\nu, \hat{T}^\nu, \hat{t}^\nu$ to evaluate Equations (3.12), (3.13), (3.14), and (3.15) for all $i, j = 1, \ldots, m$, and $k = 1, \ldots, Y_l$. \\
3. Update parameters to:

$$\hat{\tau}_{i}^{\nu+1} = \varpi(i)/M$$

$$[\hat{T}^{\nu+1}]_{ij} = \xi(i, j)/\eta(i)$$

$$\hat{t}_{i}^{\nu+1} = \omega(i)/\eta(i)$$

4. Repeat steps 2 and 3 until either $\nu$ reaches a desired value or until the change likelihood function falls below a set threshold. \\
\hline
\end{tabular}
\end{center}
Chapter 4

Markov Decision Processes

This chapter will present the Markov decision process which uses many of the Markov chain constructions seen to this point and will be the main focus of study of the original research presented in Chapter 6. The Markov decision process is based on two concepts, a reward(s) for making transitions in a Markov chain, and a decision(s) which can change the transition probabilities and/or the reward(s). Fundamental to the Markov decision processes is that the system of interest be modelled by a Markov chain. In Chapter 5, previous applications of Markov decision processes to modelling and optimising release from reservoirs will be discussed and these will be further extended in Chapter 6.

4.1 Introduction

A decision process $M$ is a stochastic process \( \{J_k : k \in \mathbb{Z}_+\} \) with a reward associated with being in each state of the process and a set of decisions the effects how the process evolves. The decision process is defined by 4 quantities $M = (X, D, P, R)$:

- $X$ a countable state space representing the states of the system,
- $D$ a compact set of decisions,
- $P$ a transition function that maps state/decision pairs to a probability distribution over $X$, $\Pr[J_{k+1} = j | J_k = i, d \in D]$,
- $R$ a reward function $R : X \times D \rightarrow \mathbb{R}$.

A Markov decision process (MDP) is one where the transitions between states are Markovian. That is, the states in $X$ are the states of a Markov chain.

A probability transition between the states $i, j \in X$ for a given action $d \in D$ is written $p_{ij}(d) = \Pr[J_{k+1} = j | J_k = i, d \in D]$, which is the $i^{th}$ row and $j^{th}$ column element of a
probability transition matrix $P(d)$. Likewise, a reward, $R(d)$, is defined for a decision $d$, where $r_j(d)$ is the reward associated with ending in state $j$ with decision $d$.

A Bellman equation (Bellman, 1957) describing the value $v_k(i)$ of being in state $i$ at time $k$ is given by

$$v_k(i) = \sum_j p_{ij}(d) [r_j(d) + v_{k+1}(j)]$$

$$= \lambda_i(d) + \sum_j p_{ij}(d)v_{k+1}(j),$$

where $\lambda_i(d) = \sum_j p_{ij}(d)r_j(d)$.

An alternative formulation of an MDP can have $R(d)$ being dependent on the state before and after the decision $d$ is taken. That is, a matrix giving a reward associated with the transition from one state of $X$ to another state. Here $r_{ij}(d)$ would denote the reward associated with the transition from state $i$ to state $j$ in $X$ under decision $d$. Consequently $\lambda_i(d) = \sum_j p_{ij}(d)r_{ij}(d)$, otherwise there is no practical difference between the two formulations. Both are used in this thesis, where appropriate.

The expected monetary value (EMV) until some future time $K$ of being in state $i \in X$ at time $k$ is denoted $v_k(i)$. A backwards recursive Bellman equation is used to determine the maximum EMV, $v^*_k(i)$, at time $k$ given the maximum value at time $k + 1$ of all states of $X$, by choosing the decision $d \in D$ such that,

$$v^*_k(i) = \max_{d \in D} \left\{ \lambda_i(d) + \sum_j p_{ij}(d)v^*_{k+1}(j) \right\}. \quad (4.1)$$

A policy, denoted $u$, is a set of decisions to be made at each state. That is, $u$ is a vector of length $|X|$, the number of states in $X$, where the $i^{th}$ element of $u$ is $u_i \in D$.

### 4.2 Value Iteration

Howard (1960, chap. 3) proposes a method for finding an optimal set of decisions that will create a policy to give the maximal long term reward. Let $n$ be the number of time steps until the end time of the process. Then Equation (4.1) is rewritten as,

$$v^*_n(i) = \max_{d \in D} \left\{ \lambda_i(d) + \sum_j p_{ij}(d)v^*_{n+1}(j) \right\}. \quad (4.2)$$

By choosing/setting a final value, $v_0(i)$ for all states $i \in X$, solving Equation (4.2) will give a policy $u^n$ at each time step $n$ that maximises future expected EMV.

If the final time is allowed to tend to infinity, in which case Bellman’s equation is solved recursively from an arbitrary final value, $v_0(i)$, at some unspecified future time working
backwards until a steady state solution is found. Bellman (1957) showed that this will converge, although not necessarily in a finite number of steps. The convergence on an optimal policy: \( u^n \rightarrow u^* \) will only occur as \( n \rightarrow \infty \). In practice the algorithm stops when \( u^{n+1} = u^n \), as the process has become stationary and further iterations will not result in a different policy. Different initial values for \( v_0(i) \) are used to check that this policy is globally optimal, as there may be locally optimal solutions, although in practice this is rare.

### 4.2.1 Discounted Value Iteration

Howard (1960, chap. 7) also considers that future values of a system will have a lower utility now than the current value and proposes that future values be discounted by a factor of \( \beta \), \( 0 \leq \beta \leq 1 \). This results in a geometric utility function were the value of one dollar at \( n \) time steps into the future is worth \( \beta^n \) dollars now. It has a Bellman equation of the form,

\[
 v^*_n(i) = \max_{d \in D} \left\{ \lambda_i(d) + \beta \sum_j p_{ij}(d) v^*_{n-1}(j) \right\}. \tag{4.3}
\]

Equation (4.3) is identical to Equation (4.2) when \( \beta = 1 \). When \( \beta = 0 \) the decision is made to maximize the current reward only with no value placed on future rewards. This has stronger convergence than (4.2) and will converge quicker for smaller \( \beta \).

### 4.3 Policy Iteration

Howard (1960, chap. 4) also proposes an alternative method for solving Markov decision processes. This method converges on an optimal policy \( u^* \) in a finite number of iterations so long as the number of states of \( X \) is a finite number \( m \). Policy iteration is performed in two-steps, a value-determination procedure, followed by a policy-improvement procedure.

**Value-Determination Procedure**

Consider Equation (4.2), if the system is operating with policy \( u \) at all times \( n \) then,

\[
 v_n(i) = \lambda_i + \sum_j p_{ij} v_{n-1}(j). \tag{4.4}
\]

If \( P \) is the probability transition matrix for the system operating under this policy, then if the vector \( v_n \) is the state vector for the values at time \( n \) and the vector \( \lambda \) has
elements $\lambda_i$ then Equation (4.4) can be rewritten in matrix form as,

$$v_{n+1} = \lambda + P v_n.$$  

(4.5)

The $z$-transform of Equation (4.5) is,

$$V_z = \frac{z}{1-z}(I_m - zP)^{-1}\lambda + (I_m - zP)^{-1}V_0 ,$$

(4.6)

where $V_z$ is the $z$-transform of $v_n$.

If $P$ is ergodic then $P^n$ can be expressed as $S + T(n)$, where $S$ is the long-term stable behaviour of the matrix $P$ where every row is equal to $\pi = \pi P$, and $\pi 1 = 1$. $T(n)$ is the asymptotic behaviour and all the entries tend to zero as $n$ becomes very large. As a result,

$$(I_m - zP)^{-1} = \frac{1}{1-z}S + T(z) ,$$

(4.7)

where $T(z)$ is the $z$-transform of $T(n)$. Substituting (4.7) into (4.6) yields,

$$V_z = \frac{z}{(1-z)^2}S\lambda + \frac{z}{1-z}T(z)\lambda + \frac{1}{1-z}SV_0 + T(z)V_0 ,$$

and taking the inverse $z$-transform yields,

$$v_n = nS\lambda + \sum_{k=1}^{n} T(k)\lambda + Sv_0 + T(n)v_0 .$$

As $n \to \infty$, $T(n)v_0 \to 0$ and $\sum_{k=1}^{n} T(k)\lambda + Sv_0 \to v$ a vector of constants associated with the value of each state. As the rows of $S$ are $\pi$, define $g = S\lambda$ where the elements $g_i = g = \sum \pi_j \lambda_j$. This yields the asymptotic form,

$$v_n = ng + v ,$$

or for each state $i \in X$,

$$v_n(i) = ng + v(i) .$$

(4.8)

Substituting Equation (4.8) into Equation (4.4) yields the equation describing the value increase with each time step for very large $n$,

$$ng + v(i) = \lambda_i + \sum_j p_{ij}((n-1)g + v(j))$$

$$\Rightarrow g + v(i) = \lambda_i + \sum_j p_{ij}v(j) .$$

(4.9)

It is worth noting that $v(i), \forall i \in X$ are relative values (substitute $r_j = ar_j' + b$ into Equation (4.9) to check). Therefore fixing $v(N) = 0$, a series of linear equations can be solved for Equation (4.9). This will determine the relative values for each state given a fixed policy when $n$ is large.
4.3. POLICY ITERATION

Policy-Improvement Procedure

Consider Equation (4.1), which states,

\[ v^*_n(i) = \max_{d \in D} \left\{ \lambda_i(d) + \sum_j p_{ij}(d)v^*_{n-1}(j) \right\}. \]

As only the long-term stable solution is of interest, consider the asymptotic values,

\[ ng + v(i) = \max_{d \in D} \left\{ \lambda_i(d) + \sum_j p_{ij}(d)[(n - 1)g + v(j)] \right\}, \]
\[ g + v(i) = \max_{d \in D} \left\{ \lambda_i(d) + \sum_j p_{ij}(d)v(j) \right\}. \]

As \( g \) is a constant, the policy that maximises \( v(u) \) is the policy that maximises,

\[ \lambda_i(d) + \sum_j p_{ij}(d)v(j) \quad (4.10) \]

for all \( i \in X \).

Policy Iteration is performed by alternately solving Equation (4.9) and maximising Equation (4.10) as shown in Algorithm 4.1.

Algorithm 4.1: Pseudo-code for the Policy Iteration procedure.

<table>
<thead>
<tr>
<th>Policy Iteration:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set ( \nu = 0 ) with initial estimate ( u^0 ).</td>
</tr>
<tr>
<td>2. For ( u^\nu ) solve Equation (4.9) with ( v(N) = 0 ) to find ( v^{\nu+1} ).</td>
</tr>
<tr>
<td>3. Using ( v^{\nu+1} ) maximise (4.10) to find ( u^{\nu+1} ).</td>
</tr>
<tr>
<td>4. If ( u^{\nu+1} = u^\nu ) stop, otherwise set ( \nu \leftarrow \nu + 1 ) and repeat steps 2 and 3.</td>
</tr>
</tbody>
</table>

Puterman and Brumelle (1978 and 1979) showed that policy iteration is equivalent to the Newton-Kantorovich root-finding procedure, such that there exists an optimal policy \( u^* \), and that \( u^\nu \) converges to \( u^* \). For problems where \( X \) has finite cardinality and \( D \) is a discrete finite set (such that will be considered in this thesis), policy iteration converges \( O(N^2) \).
4.4 Stochastic Solution Methods

In situations where the number of states, $|X|$, and the number of decisions, $|D|$, are very large it is not always practical to solve using policy iteration or value iteration. In recent years a number of stochastic solution methods have been proposed to find the solution to Markov Decision Processes without having to calculate every state/decision pair for each iteration step. Instead, policies are generated randomly in a systematic way to find policies that are optimal for the reward function. These have also proven popular when solving optimisation problems with more than one reward function, such as in multi-objective optimisation. Two widely used methods are simulated annealing and genetic algorithms.

For a fixed policy $u$ there is a matrix $P(u)$ whose elements are given by $p_{ij}(u_i)$, where $u_i \in D$, is the decision made in state $i$ given by policy $u$. Define $\pi(u)$ as the stationary (row) vector of the matrix $P(u)$.

Now define a column vector $\lambda(u)$ which has as its $i^{th}$ element the expected reward $\lambda(u_i)$ for decision $u_i \in D$ given by policy $u$.

The expected monetary value of the system operating under a policy $u$ is the given by,

$$f(u) = \pi(u)\lambda(u). \quad (4.11)$$

Stochastic solution methods focus on finding the policy which will maximise Equation (4.11). These are adaptations of well established solution methods. However it should be noted that there is no one ideal one way of implementing these methods and they are subject to modification on a case by case basis.

4.4.1 Simulated Annealing

Simulated annealing is based on the idea of annealing in metallurgy where metal is heated and then cooled in a controlled way to achieve a desired crystal structure. To do this a temperature $T$ is introduced which will control the process.

Starting with an initial policy $u^0$ a number of the elements of the vector, usually the number is conditioned on $T$, are mutated at random. This is done by taking a chosen element, $u_i \in D$ and selecting another decision in $D$, $u_i^* \in D$, to replace its position in the vector. For example,

$$u^0 = [u_1, u_2, \ldots, u_{c-1}, u_c, u_{c+1}, \ldots, u_n] \rightarrow u^1 = [u_1, u_2^*, \ldots, u_{c-1}, u_c^*, u_{c+1}, \ldots, u_n^1].$$

If the new policy performs better, then it always replaces the old policy. However if it performs worse, it replaces the old policy with a probability dependent on $T$. After a predetermined number of iterations $M$, the temperature is lowered. This process is then repeated a predetermined number of times $N$. All the while the best policy,
4.4. STOCHASTIC SOLUTION METHODS

defined as the one that gives the highest EMV, is recorded and is used to restart
the mutation process each time. The best policy found through this process is used as the
maximum. A summary of the simulated annealing procedure is shown in Algorithm
4.2.

Algorithm 4.2: Pseudo-code for Simulated Annealing procedure applied to MDPs.

\begin{algorithm}
\caption{Simulated Annealing:}
\begin{algorithmic}
\STATE 1. Set $\nu = 0$ with initial policy $u^0$, temperature $T$, and $0 < \alpha < 1$.
\STATE 2. Begin search at temperature $T$:
\begin{itemize}
\item[(a)] Probabilistically mutate $u^\nu$ conditional on $T$ to create $u'$.
\item[(b)] Calculate $g = \min\{1, \exp[-(f(u) - f(u'))/T]\}$.
\item[(c)] Set $u^{\nu+1} \leftarrow u'$ with probability $g$, otherwise set $u^{\nu+1} \leftarrow u^\nu$.
\item[(d)] Set $\nu \leftarrow \nu + 1$.
\item[(e)] Repeat Steps (a)–(d) $M$ times while keeping note of $u^* = \arg\max_u \{f(u^0), \ldots, f(u^M)\}$.
\end{itemize}
\STATE 3. Lower temperature $T \leftarrow \alpha T$, then set $\nu = 0$ and $u^0 \leftarrow u^*$
\STATE 4. Repeat Steps 2 & 3, $N$ times.
\end{algorithmic}
\end{algorithm}

4.4.2 Genetic Algorithms

Genetic algorithms are based on the idea of Mendelian inheritance in evolutionary
theory. Here a population $Q^0$ of $N$ policies is generated. The policies are then ranked
by how well they perform on the EMV measure, this is called their “fitness”. These
rankings are used to determine “breeding” pairs. Breeding pairs are selected randomly
where the higher ranked polices are favoured, usually by a weighting method. When a
breeding pair is selected, a point along the vectors is chosen at random (via a probability
$p_c$) for a crossover. Each of the “offspring” vectors will inherit one portion of the vector
from each of their “parent” policy vectors (see Figure 4.1), making two new vectors.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.1.pdf}
\caption{The crossover in two policy vectors from the parents to the offspring.}
\end{figure}

After a population $Q^1$ of $N$ new policies is generated, random elements (chosen typically
with a small probability $p_m$) of the new policies vectors are mutated in the same way as in the simulated annealing example. This process is repeated for $M$ generations, at the end of which the “fittest” policy found via this procedure is used as the maximum. A summary of a genetic algorithm for MDPs is shown in Algorithm 4.3.

Algorithm 4.3: Pseudo-code for a Genetic Algorithm applied to MDPs.

<table>
<thead>
<tr>
<th>Genetic Algorithms:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Generate a population of $N$ policies $Q^0 = {u^1, \ldots, u^N}$ and set $0 &lt; p_c, p_m &lt; 1$ and $\nu = 0$.</td>
</tr>
<tr>
<td>2. Evaluate the EMV of the population $f(Q^\nu) = {f(u^1), \ldots, f(u^N)}$.</td>
</tr>
<tr>
<td>3. Breeding pairs are chosen from $Q$ based on their value in $f(Q)$, preference given to better performers.</td>
</tr>
<tr>
<td>4. For a breeding pair a point is chosen along the vector with probability $p_c$ and the two vectors are cut at this point, swapped and recombined, to create two new policies.</td>
</tr>
<tr>
<td>5. Repeat Step 3 and 4 until a new population $Q^{\nu+1}$ of $N$ policies is created.</td>
</tr>
<tr>
<td>6. Mutate individual element in the policies in $Q^{\nu+1}$ with probability $p_m$.</td>
</tr>
<tr>
<td>7. Set $\nu \leftarrow \nu + 1$ and repeat step 2–6 $M$ times.</td>
</tr>
</tbody>
</table>

### 4.5 Partially Observable Markov Decision Processes

Partially observable Markov decision processes (POMDPs) are based on similar ideas to hidden Markov models (Section 3.3) in that only a sequence of observations $\{Y_k\}$ is observed, not the process $\{J_k\}$. Formulations can vary, but Littman (2009) defines POMDPs as follows:

- $X$, the state space of the Markov chain,
- $D$, the set of decisions,
- $Y$, the set of observations,
- $P$, the probability transition matrix dependent on the decision,
- $R$, the reward function dependent on the state of the Markov chain and the decision,
- $Q$, the observation function which relates the states and the decisions to the observations,
4.5. PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES

- \( b \), a vector of probabilities describing the belief of the current state in \( X \), such that \( \sum_i b_i = 1 \).

The constructions \( X, P, D \) and \( R \) are defined in the same way as before. As \( Q \) is some probability function that maps \( X, D \) and \( R \) to \([0, 1]\) then the likelihood of being in the unobserved states \( X \) based on observations \( Y \) is tracked with the belief vector \( b \). The \( i^{th} \) element \( b_i \) of \( b \) is the probability of the process being in state \( i \) and \( \sum_i b_i = 1 \). What the observations \( Y \) are is dependent on the function \( Q \), they could be rewards, subsets of \( X \), or another indicator variable entirely. Given that rewards are a function of state and action, \( Q \) is a map of \( X \) and \( D \) to \([0, 1]\) and once again can be represented as a set of matrices that vary with \( d \in D \). The function \( Q \) is used to update \( b \) based on the observations \( Y \) as they occur.

Dealing with POMDPs

The probability of observing \( Y_k \) based on the belief about the of state \( J_{k-1}, b^{k-1} \), and the decision \( d \in D \),

\[
p_y = \Pr[Y_k|d, b^{k-1}] = \sum_i b_i^{k-1} \sum_j \Pr[J_k = j|d, J_{k-1} = i] \Pr[Y_k|d, J_k = j] = \sum_i b_i^{k-1} \sum_j p_{ij}(d) q_{jY_k}(d) .
\]

The belief vector is updated at time \( k \) based on the decision at time \( k - 1 \) and observation \( Y_k \) at time \( k \),

\[
b_j^{k} = \Pr[J_k = j|Y_k, d, b^{k-1}] = \frac{\sum_i b_i^{k-1} p_{ij}(d) q_{jY_k}(d)}{\sum_i b_i^{k-1} \sum_j p_{ij}(d) q_{jY_k}(d)} .
\]

If the observations are not the rewards, using the belief vector \( b^k \) at time \( k \), the expected reward received for making decision \( d \) is,

\[
\lambda(d, b^k) = \sum_i b_i^k \sum_j r_{ij}(d) .
\]

Using the expected reward, a Bellman equation can be written that when solved iteratively will give a policy that maximises the expected return until the end time,

\[
v_{k-1}^*(b^{k-1}) = \max_{d \in D} \left\{ \lambda(d, b^{k-1}) + \sum_y p_y v_{k}^*(b^k) \right\} .
\]

Similar to Section 4.2.1 a discount factor \( 0 \leq \beta \leq 1 \) can be applied.

\[
v_{k-1}^*(b^{k-1}) = \max_{d \in D} \left\{ \lambda(d, b^{k-1}) + \beta \sum_y p_y v_{k}^*(b^k) \right\} .
\]
It should be noted that unlike standard MDPs, in many cases POMDPs cannot be solved to find stable solutions without the use of simulations.
Chapter 5

Literature Review

5.1 Moran Dam Model

First proposed by Moran (1954), the Moran Dam Model is a simple, discrete-time stochastic model for a reservoir of maximal capacity $C$. The available storage $Z_{k+1}$ of the reservoir at time $k + 1$ is given by the recursive relation,

$$Z_{k+1} = \min[Z_k + Y_{k+1}, C] - \min[Z_k + Y_{k+1}, D],$$

where $Y_{k+1}$ is the net inflow (inflow minus evaporation) in the time step $(k, k + 1]$ and $D \in (0, C)$ is the target amount of water released from the reservoir during one time step.

Typically it is assumed that $Y_1, Y_2, \ldots$ are identically and independently distributed (iid), such as Moran (1959). Another assumption implicit in the model is that the water is released from the dam after it flows in, as a result the storage will never be above $C - D$ at the end of the time step. Both of these assumptions are valid for the modelling of a reservoir created by a dam which is used to hold back an annual inflow of water that arrives in a relatively short period of time, before being slowly released during the year. If the main inflows are close to one year apart the assumption of independence between inflows should be valid.

The Moran Dam Model under these assumption is a Markov chain and is analytically identical to the $M/G/1$ or $G/M/1$ server queues (Stadje, 1993). In Section 3.2.1 it was discussed that $M/G/1$ or $G/M/1$ server queues can be represented in the MAM framework as quasi-birth and death processes, so that it can be inferred that the Moran Dam Model is also a quasi-birth and death process.

If $Z_k$ and $Y_k$ take only discrete values, $Z_k = 0, 1, \ldots, C - D$ and $Y_k \in \mathbb{Z}_+$, then the Moran Dam Model is a finite state Markov chain, with the $K + 1$ states of the chain representing levels of storage in the reservoir.

The probability of inflow $Y_{k+1} = i$ is denoted $p_i$ for all $i = 0, 1, \ldots$ and the probability of the storage being $Z_k = i$ at time $k$ is denoted $\pi_k^i$. Consider a release policy of
$C - D > D$ (this is done for convenience, an alternative set of equations can be written for $C - D \leq D$), the probabilities $\pi_{i}^{k+1}$ are then:

$$
\begin{align*}
\pi_{0}^{k+1} &= \pi_{0}^{k}(p_{0} + \cdots + p_{D}) + \pi_{1}^{k}(p_{0} + \cdots + p_{D-1}) + \cdots + \pi_{D}^{k}p_{0} \\
\pi_{1}^{k+1} &= \pi_{0}^{k}p_{D+1} + \pi_{1}^{k}p_{D} + \cdots + \pi_{D+1}^{k}p_{0} \\
& \vdots \\
\pi_{C-2D}^{k+1} &= \pi_{0}^{k}p_{C-D} + \pi_{1}^{k}p_{C-D-1} + \cdots + \pi_{C-D}^{k}p_{0} \\
& \vdots \\
\pi_{C-D-1}^{k+1} &= \pi_{0}^{k}p_{C-1} + \pi_{1}^{k}p_{C-2} + \cdots + \pi_{C-D-1}^{k}p_{D-1} \\
\pi_{C-D}^{k+1} &= \pi_{0}^{k}(p_{C} + \cdots) + \pi_{1}^{k}(p_{C-1} + \cdots) + \cdots + \pi_{C-D}^{k}(p_{D} + \cdots). 
\end{align*}
$$

(5.1)

As $k \to \infty$, $\pi_{i}^{k} \to \pi_{i}$, the stationary probability of being in state $i$ at the end of each time-step. By setting $\pi_{i}^{k} = \pi_{i}^{k+1} = \pi_{i}$ for all $i = 0, 1, \ldots, C - D$ in the set of linear equations (5.1) and using the fact that $\sum \pi_{i} = 1$, the system of linear equations can be solved to find the stationary probability distribution of the water storage in the reservoir.

Using the Moran model it is possible to study two other stochastic processes, $\{R_{k}\}$, the sequence of actual releases and, $\{W_{k}\}$, the sequence of overflows defined by,

$$
R_{k+1} = \min\left[Z_{k} + Y_{k+1}, D\right] \quad \text{and} \quad W_{k+1} = \max\left[Z_{k} + Y_{k+1} - C, 0\right].
$$

Define the process $U_{k+1} = Z_{k} + Y_{k+1}$, this would model a Moran Dam of infinite capacity and no release. Let $\Pr(U_{k} = i) = \rho_{i}^{k}$ and let the stationary probability distribution be $\lim_{k \to \infty} \rho_{i}^{k} = \rho_{i}$. It can be shown that,

$$
\begin{align*}
\rho_{0} &= \pi_{0}p_{0} \\
\rho_{1} &= \pi_{0}p_{1} + \pi_{1}p_{0} \\
& \vdots \\
\rho_{C-D} &= \pi_{0}p_{C-D} + \cdots + \pi_{C-D}p_{0} \\
\rho_{C-D+s} &= \pi_{0}p_{C-D+s} + \cdots + \pi_{C-D+s}p_{s}, \quad s > 0.
\end{align*}
$$

Then the stationary probability distribution of the process $\{R_{k}\}$ is,

$$
\begin{align*}
\Pr[R_{\infty} = 0] &= \rho_{0} \\
\Pr[R_{\infty} = 1] &= \rho_{1} \\
& \vdots \\
\Pr[R_{\infty} = D - 1] &= \rho_{D-1} \\
\Pr[R_{\infty} = D] &= \rho_{D} + \rho_{D+1} + \cdots \\
& = 1 - \rho_{0} - \rho_{1} - \cdots - \rho_{D-1}
\end{align*}
$$

The stationary probability distribution of the process $\{W_{k}\}$ is,

$$
\begin{align*}
\Pr[W_{\infty} = 0] &= \rho_{0} + \rho_{1} + \cdots + \rho_{C}
\end{align*}
$$
5.1. MORAN DAM MODEL

\[ Pr[W_\infty = s] = \rho_{C+s}, \quad s > 0. \]

Reznicek and Cheng (1991) commented that it is the ease with which these stationary probability distributions can be calculated that makes the Moran Dam Model desirable. However if the assumptions are not valid the theory cannot be applied in practice.

5.1.1 Gould Probability Matrix

First proposed by Gould (1961), the Gould probability matrix uses a modified Moran model to determine the capacity \( C \) at which a reservoir should be built by setting an acceptable probability of failure. Described by McMahon and Mein (1978), the method is as follows:

1. Determine the monthly releases from the reservoir \( D_k \) for \( k = 1, \ldots, 12 \).

2. Using the historical inflows to the reservoir, find all the monthly inflows \( Y_k \). Let \( N \) denote the total number of years for which the monthly inflows are known.

3. Determine the monthly evaporation \( E_Z \) for the storage level \( Z \in [0, C] \).

4. For all initial levels of \( Z_0 \), calculate \( Z_1 = Z_0 + Y_1 - D_1 \). That is, for each initial level \( Z_0 \) of the reservoir, find what level \( Z_1 \) the reservoir would be after one month.

5. Calculate iteratively \( Z_{k+1} = \min[Z_k + Y_{k+1}, C] - \min[Z_k + Y_{k+1}, D_k] \) until \( Z_{12} \) is obtained.

6. If \( Z_0 = i \) and \( Z_{12} = j \), add 1 to \( i^{th} \) row and \( j^{th} \) column element of a table, and perform this calculation for all rows \( i \). That is calculate what the reservoir level would be at the end of the year for the reservoir starting the year at each initial level \( i \).

7. Repeat Steps 4 to 6 for all \( N \) years in the data set and divide each row of the table by \( N \) to form a matrix \( P \).

8. \( P \) is now the annual transition matrix for the storage of the reservoir with releases \( D_k \) for each month \( k = 1, \ldots, 12 \) during the year.

This new model improves upon the Moran model in that it allows for seasonal variation in the inflows, as well as allowing the releases and inflows to be spaced over the year. It still assumes that the annual inflows are independent and identically distributed. The capacity \( C \) for which the dam should be constructed is found as follows:

1. Partition a reservoir into \( C \) levels and a level zero (0) representing the dead storage of the reservoir.
2. Calculate the probability transition matrix $P$ as described above.

3. Whilst calculating $P$, count the number of times the storage reaches zero during the year (not just at the end). Create the column vector $f$, the elements of which are the probability that the storage reaches zero, given it is at level $i$ at the start of the year.

4. Calculate the steady state vector $\pi = \pi P$, where $\pi 1 = 1$.

5. Calculate the overall risk $\pi f$ of the reservoir reaching dead storage.

6. If the overall risk is greater than expected probability of failure, increase the maximum capacity $C$ and repeat Steps 2 to 5 until the risk is below the desired value.

Srikanthan and McMahon (1985a) considered that given $P$ is the annual probability transition matrix calculated using the inflows and releases on a monthly basis, what should be used as the starting month? Analysis showed that by starting the annual time step at different months, that the calculated capacity (Gould storage) could vary by as much as 20\%. They concluded that the starting month should be the one with the minimum mean monthly inflow as this was the month that produced the largest Gould storage and would therefore provide the largest guard against failure. Srikanthan and McMahon (1985b) considered the problem of adding temporal, or serial, correlation to the inflows as the Gould matrix model is unrealistic when dealing with such scenarios. They concluded that a correction factor had to be applied to the Gould storage if the correlation coefficient of the lag-1 inflows $|r| \geq 0.2$ and that if the correction coefficient was greater than 1.5 it indicated that the Gould matrix method was unsuitable for such applications. Otieno and Ndiritu (1997) determined that for annual serial correlated inflows, the starting month with the lowest annual serial correlation coefficient should be used as the starting month.

The Gould matrix method has also been used to determine reservoir operation policy when the capacity and releases are fixed. Price (2008) described how a modified Gould matrix was used to determine when water should be transferred via a pipeline to supplement the water supply in neighbouring counties in the South East of England. The counties Essex and Suffolk have two reservoirs which can be supplemented by buying and transferring water from a river called the Ely Ouse which is located in a third county to the North. A modified Gould probability transition matrix was used to model the total capacity of the two reservoirs, treating them as a single storage. For each month and each level of storage the probability that the system would be full at the start of May was calculated (similar to the probability that the reservoir reaches dead storage) and if it was less than 0.9, water would be transferred from the Ely Ouse. This produced a control curve on which to operate the transfer scheme.
5.2 Linear Programming Models

A linear programming (LP) model is a model in which both the objective function to optimise and the inequality constraints on the optimisation are linear in the variables. A linear programming model is expressed in canonical form as:

\[
\min_x Z = C^T x,
\]

subject to,

\[
Ax \leq b
\]

\[x_i \geq 0 \text{ for all } i = 1, \ldots, n,
\]

where \(C\) is an \(n \times n\) matrix of coefficients of the linear objective function, \(x\) is an \(n \times 1\) vector of the variables, \(A\) is an \(m \times n\) matrix of coefficient of the linear constraints and \(b\) is an \(m \times 1\) vector representing the right hand side of the linear constraints.

Whilst LP models are deterministic they have been used in the planning of reservoirs such as finding the minimal capacity required to meet expected inflows, Yeh (1985) has a comprehensive list of such applications. LP models have found wide use in conjunction with dynamic programming (DP) models, such as in Webby et al. (2009) and Vedula and Nagesh Kumar (1996) amongst may others. Dynamic programming models were introduced by Bellman (1957) and have been used to model the non-linear and dynamic components of reservoirs.

5.2.1 Stochastic Linear Programming

Proposed by Loucks (1968) stochastic linear programming models combine the efficient calculations of linear programming with the stochastic nature of the system being modelled. Consider the point probability \(X_{\nu,i,d,k}\) of having initial storage volume \(\nu\) at time \(k\), net inflow \(i\) and release \(d\) during the time \([k, k + 1)\). Let \(R_{\nu,i,d,k}\) be the reward for such states. Correlated serial flows can be modelled using a Markov chain. Denote the inflow probabilities \(P_{j,k} = \Pr[Y_{k+1} = j|Y_k = i]\), where \(Y_{k+1}\) is the net inflow in the time period \([k, k + 1)\). The problem is then expressed in canonical form as:

\[
\max_{X_{\nu,i,d,k}} R_{\nu,i,d,k} X_{\nu,i,d,k},
\]
subject to,

$$\sum_{\nu,i,d} X_{\nu,i,d,t} = 1 \text{ for all } k,$$

$$\sum_{e=\bar{\nu}+j-C}^{\bar{\nu}+j} X_{\nu,j,e,k+1} = \sum_k \sum_{\nu} \sum_{d=\nu+i-C}^{\nu+i} X_{\nu,i,d,k} P_{i,j,k} \text{ for all } \bar{\nu}, j, k$$

$$0 < X_{\nu,i,d,k} < 1 \text{ for all } \nu, j, d, k,$$

where $e$ and $Z_{k+1} = \bar{\nu}$ are the release and storage volume at time $k + 1$ respectively.

This method can be used to calculate probability based operating policies using

$$\Pr[d|\nu,i,k] = \frac{X_{\nu,i,d,k}}{\sum_d X_{\nu,i,d,k}}.$$

The major drawback of this method is that the number of constraints can easily reach into the thousands (Yeh, 1985).

### 5.3 Stochastic Dynamic Programming - Single Reservoir Models

Stochastic dynamic programming (SDP), in the hydrological literature (Reznicek and Cheng, 1991), refers to the value iteration method of solving Markov decision processes as presented in Section 4.2, this modelling technique allows for the solution of both non-linear and stochastic elements for reservoir management. Most of the time these models can be solved using the policy iteration technique (Section 4.3) as well.

#### 5.3.1 Stochastic Dynamic Programming

If $v_k(Z_k)$ is the expected value at time $k$ of the reservoir having storage $Z_k$, the problem is to solve,

$$v_k(Z_k) = \max_{d_k \in D} \left\{ \sum_{Y_{k+1}} \Pr[Y_{k+1}] \left[ R(d_k, Y_{k+1}, Z_k) + v_{k+1}(Z_{k+1}) \right] \right\},$$

subject to,

$$0 \leq Z_{k+1} = Z_k + Y_{k+1} + \Delta(d_k) \leq C.$$
where $Y_{k+1}$ is the inflow from $(k, k+1]$, $\Delta(d_k)$ is the net effect of the decision $d_k$ decided at time $k$ (typically negative unless the reservoir is a pump-storage), and $C$ is the capacity of the reservoir.

Alternatively some authors let

$$\lambda(d_k, Z_k) = E_{Y_{k+1}}[R(d_k, Y_{k+1}, Z_k)]$$

and then solve,

$$v_k(Z_k) = \max_{d_k \in D} \left\{ \lambda(d_k, Z_k) + \sum_{Y_{k+1}} \Pr[Y_{k+1}] [v_{k+1}(Z_{k+1})] \right\},$$

subject to,

$$0 \leq Z_{k+1} = Z_k + Y_{k+1} + \Delta(d_k) \leq C.$$

Both of these problems can be solved using the formulation presented in Chapter 4. The basic model for SDP with a single reservoir system is as follow:

- $X$ represents the level of storage in the reservoir,
- $D$ represents the set of decisions that can be applied to the system (typically releases),
- $P$ gives the probability of the net change in storage (inflow minus release minus evaporation) after one time step,
- and $R$ represents the rewards for the storage/decision pairs.

Here the states of the Markov chain $X$ will represent discretised partitions of the reservoir storage, so $Z_k \in X = \{0, 1, 2, \ldots, C-1, C\}$.

The probability $p_{i,j}(d)$ shall denote the time-homogeneous probability of going from $Z_k = i \in X$ to $Z_{k+1} = j \in X$ under release decision $d_k \in D$.

When $0 < Z_{k+1} < C$, the probability of the change in storage is given by,

$$\Pr[Z_{k+1}|Z_k] = \Pr[Z_k + Y_{k+1} + \Delta(d_k)|Z_k] = \Pr[Y_{k+1}] .$$

As a result the probability $p_{i,j}(d_k)$ for $0 < j < C$ is given by,

$$p_{i,j}(d_k) = \Pr[Y_{k+1} = j - i - \Delta(d_k)] .$$

When $Z_{k+1} \geq C$ then,

$$p_{i,C}(d_k) = \sum_{Z_{k+1} \geq C} \Pr[Z_{k+1}|Z_k] = \sum_{Y_{k+1} \geq C-i-\Delta(d_k)} \Pr[Y_{k+1}] .$$
Similarly when \( Z_{k+1} \leq 0 \),

\[
p_{i,0}(d_k) = \sum_{Z_{k+1} \leq 0} \Pr[Z_{k+1} | Z_k] = \sum_{Y_{k+1} \leq -i - \Delta(d_k)} \Pr[Y_{k+1}].
\]

Further constraints are typically applied using \( R \) and \( D \). States that are possible, but undesirable are given a large negative reward or penalty. States that are never possible to enter are constrained by making decisions that would cause them not available.

Solving to find an optimal policy \( u^* \), selecting the rows of \( P \) that correspond to that decision being implemented the matrix \( P(u^*) \) can constructed that describes the transition probabilities of the system being operated under this optimal policy. The steady state vector for operating under this policy can be found by solving \( \pi P(u^*) = \pi \) and \( \pi 1 = 1 \). If column vector \( r(u^*) \) is constructed by putting \( R(u^*_i) \), the reward for being in state \( i \) under optimal decision policy, as the \( i^{th} \) element, the expected gain \( g \) for the system can be found by \( g = \pi r(u^*) \).

Figure 5.1: Reconstruction of the results of Goulter and Tai (1985) on the effect on the expected gain by storage space discretisation.

Goulter and Tai (1985) looked at the effect of the discretisation of the storage on the optimal policy. They found that a smaller number of storage states caused an optimal solution with a lower expected gain (see Figure 5.1) and that the stationary distribution \( \pi \) would be skewed, with more probability mass at the higher storage states. This artificial skewing was found to be greatly reduced when at least nine storage levels was used.
5.3. Seasonal Stochastic Dynamic Programming Models

Butcher (1971) suggests a method for modelling reservoirs using SDP which have a seasonal component to them. In their proposed model, a decision process is expanded to

\[ M = (X, D, P_0, \ldots, P_{s-1}, R_0, \ldots, R_{s-1}), \]

where each \( P_l \) is the probability transition function at time \( l = (k \mod s) \) for a transition over the time interval \((k - 1, k]\) and \( R_l \) is the reward function at time \( l = (k \mod s) \), where \( s \) is the number of seasons in the cycle.

As the matrices are now time dependent Equation (4.1) becomes,

\[
v^*_k(i) = \max_d \left\{ \lambda_{l,i}(d) + \sum_j p_{l,ij}(d)v^*_{k+1}(d) \right\}. \tag{5.2}
\]

As the probability transition matrices and rewards are now dependent on time, this is also sometimes referred to as a time-dependent Markov decision process.

When solving the Equation (5.2) by backwards induction, in general, \( u^{n+1} \neq u^n \), as \( n \) and \( n + 1 \) are in different seasons. The new stopping rule becomes stop iterations when \( u^{n+m} = u^{n+m+s} \) for all \( m = 0, 1, \ldots, s - 1 \), that is when the policy is the same for the equivalent season in two years.

Matrix analytic methods gives us a structured framework that allows us to see that for every seasonal MDP there exists an equivalent cross-seasonal MDP. By letting

\[
P^* = \begin{bmatrix}
0 & P_1 & 0 & \ldots & 0 & 0 \\
0 & 0 & P_2 & \ddots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & \ddots & \ddots & P_{s-2} & 0 \\
0 & 0 & 0 & \ddots & 0 & P_{s-1} \\
P_0 & 0 & 0 & \ldots & 0 & 0
\end{bmatrix}
\]

and a seasonal construction for \( R^* \); by expanding the state space \( X^* = [X_0, X_1, \ldots, X_{s-1}] \), such that \( X_k \) is a collection of states in \( X \) for season \( k \) of the seasonal MDP,

\[ M = (X, D, P_0, \ldots, P_{s-1}, R_0, \ldots, R_{s-1}) \equiv (X^*, D, P^*, R^*). \]

This Markov chain is periodic with period \( s \) and as a result is not ergodic, therefore the policy iteration method cannot be used to solve seasonal problem.

5.3.3 Serially Correlated and Forward Forecast Stochastic Dynamic Programming Models

As described in Yeh (1985) a serially correlated SDP can be described as follows:
\[ v_k(Z_k, Y_k) = \max_{d_k \in D} \left\{ \lambda(d_k, Z_k) + \sum_{Y_{k+1}} \Pr[Y_{k+1}|Y_k] [v_{k+1}(Z_{k+1}, Y_{k+1})] \right\} , \tag{5.3} \]

subject to
\[ 0 \leq Z_{k+1} = Z_k + Y_{k+1} + \Delta(d_k) \leq C , \]

where \( q(d_k, Z_k) \) is the reward for the release and storage, and \( v_k(Z_k, Y_k) \) is the expected return of the system at time with \( Z_k \) units of storage and net inflow of \( Y_k \) over the time period \((k - 1, k)\). This equation can solved using value iteration from a time horizon that is \( n \) time steps away, where \( v_0(Z_0, Y_0) \) is the value of having \( Z_0 \) units of storage at end of the system’s life and \( Y_0 \) units of inflow at the preceding time interval, by substituting \( k = n, k + 1 = n - 1, \) and \( k - 1 = n + 1 \) into Equation (5.3).

Loucks and Falkson (1970) did a comparison of a reservoir operation model with serially correlated inflows using stochastic linear, dynamic and policy iteration methods to find the optimal operating policy. They all achieved the same optimal policy, but the stochastic dynamic programming approach was the fastest.

Stedinger et al. (1984) made an alteration to Equation (5.3) to produce,
\[ v_k(Z_k, Y_k) = \max_{d_k \in D} \left\{ \lambda(d_k, Z_k) + \sum_{Y_{k+1}} \Pr[Y_{k+1}|Y_k, x_k] [v_{k+1}(Z_{k+1}, Y_{k+1})] \right\} , \tag{5.4} \]

subject to
\[ 0 \leq Z_{k+1} = Z_k + Y_{k+1} + \Delta(d_k) \leq C , \]

where \( x_k \) is all other information available at time \( k \). This model is a forward forecasting model where any other hydrological or meteorological information can be incorporated into this model and Stedinger et al. (1984) found that in simulations this method out performed normal serially correlated inflow models. One major drawback to this method is that it can not be used to determine steady-state operating policies that are needed to manage a resource in the long-term, however it has found widespread use in short-term planning. Røtting and Gjelsvik (1992) for example, discuss the operation of Norway’s 38 connected hydroelectric-dam system where forward forecasting of two weeks is done using simulations to determine the best short-term operating policy within the constraints of a longer-term management plan.

Both of these methods have achieved an increase in operating performance by expanding the states space from the number of capacity states \( C + 1 \) (the usable capacity \( C \) plus a dead storage level 0) to \((C + 1) \times \max\{Y_k\}\), assuming that \( Y_k \) has a finite maximal value that can be known in advance.

Turgeon (2005) proposed a method of further increasing the correlation without expanding the state space too greatly. First consider the \( p \)-lag auto-correlation of inflows,
\[ Y_k = \phi_{0,k} + \phi_{1,k}Y_{k-1} + \phi_{2,k}Y_{k-2} + \cdots + \phi_{p,k}Y_{k-p} + \varepsilon_k \]
where \(\varepsilon_k\) is the error term. The term \(H_k\) is defined as,
\[
H_k = \phi_{1,k} Y_{k-1} + \phi_{2,k} Y_{k-2} + \cdots + \phi_{p,k} Y_{k-p},
\]
so that,
\[
Y_k = \phi_{0,k} + H_k + \varepsilon_k.
\]
Define,
\[
\hat{H}_{k+1} = \phi_{2,k+1} Y_{k-2} + \cdots + \phi_{p,k+1} Y_{k-p+1},
\]
so that,
\[
H_{k+1} = \phi_{1,k+1} Y_k + \hat{H}_{k+1}.
\]
It can then be shown that,
\[
E(\hat{H}_{k+1}|H_k) = \hat{m}_{k+1} + \rho_{k,k+1} \hat{\nu}_{k+1} \nu_k (H_k - m_k),
\]
where:
\[
m_k = E(H_k),
\]
\[
\hat{m}_{k+1} = E(\hat{H}_{k+1}),
\]
\[
\nu_k = \text{Var}(H_k),
\]
\[
\hat{\nu}_{k+1} = \text{Var}(\hat{H}_{k+1}),
\]
\[
\rho_{k,k+1} = \text{Cor}(H_k, \hat{H}_{k+1}).
\]

In the formulation given in Turgeon (2005), Equation (5.4) becomes,
\[
v_k(Z_k, H_k) = \max_{d_k \in D} \left\{ \lambda(d_k, Z_k) + \sum_{Y_{k+1}} \Pr[Y_{k+1}|Y_k, H_k] \left[ v_{k+1}(Z_{k+1}, H_{k+1}) \right] \right\},
\]
where \(H_{k+1}\) is given by,
\[
H_{k+1} = \phi_{1,k+1} Y_k + \hat{m}_{k+1} + \rho_{k,k+1} \hat{\nu}_{k+1} \nu_k (H_k - m_k).
\]

The state space is now expanded by \(\max\{H_{k+1}\}\) which is of the same order as \(\max\{Y_{k+1}\}\).

### 5.4 Stochastic Dynamic Programming - Multi-Reservoir Models

Extending stochastic dynamic programming to a system of more than one reservoir would initially appear straightforward. Let the system consist of \(N\) reservoirs is labelled \(n = 1, \ldots, N\), where typically they will be labelled from the most upstream down. Denote each of the reservoir properties at time \(k\) using, \(Z^n_k\) for storage, \(Y^n_{k+1}\) for future stochastic inflow, \(d^n_k\) for decision, and \(C^n\) for capacity for each \(n \in \{1, \ldots, N\}\).

Define:
• $\hat{Z}_k = (Z^1_k, \ldots, Z^N_k)$, the $N$-tuplet of all the storages at time $k$,
• $\hat{Y}_k = (Y^1_k, \ldots, Y^N_k)$, the $N$-tuplet of all the inflows at time $k$,
• $\hat{d}_k = (d^1_k, \ldots, d^N_k)$, the $N$-tuplet of all the decisions at time $k$
• $V_k(\hat{Z}_k) = \sum_n v(Z^n_k)$, the expected future reward at time $k$ for all $N$ storages,
• $\Lambda_k(\hat{d}_k, \hat{Z}_k) = \sum_n \lambda(d^n_k, Z^n_k)$, the total expected reward at time $k$ for the decision.

Also define $\delta(d^n_k)$ as the release at reservoir $n$ due to decision $d^n_k$ and $\delta(\hat{d}^n_k)$ as the releases from all the reservoirs directly upstream of $n$, i.e, only reservoirs whose release will first flow into $n$ before any other.

Optimizing the system as a whole is then just a matter of recursively solving,

$$\begin{align*}
V_k(\hat{Z}_k) &= \max_{\hat{d}_k} \left\{ \Lambda(\hat{d}_k, \hat{Z}_k) + \sum_{\hat{Y}_{k+1}} \Pr[\hat{Y}_{k+1}] \left[ V_{k+1}(\hat{Z}_{K+1}) \right] \right\},
\end{align*}$$

subject to,

$$0 \leq Z^n_{k+1} = Z^n_k + Y^n_{k+1} + \delta(\hat{d}^n_k) - \delta(d^n_k) \leq C^n.$$ 

Whilst this may seem straight forward, this problem is exceptionally large. The new state space is of size $C^1 \times C^2 \times \cdots \times C^N$, and with seasons it will be even larger again. The state space becomes larger again if any lag-correlation is included. Similarly the number of decisions will be the Cartesian product of the number of decisions that can be enacted on each reservoir. Similarly the inflow distribution has gone from a 1-dimensional distribution that is to be summed over to an $N$-dimensional distribution where all possible discretised $N$-tuplets are to be summed over, again the Cartesian product of all the maximal inflows to all $N$ storages. Opinions vary on the number of storages that can be handled with this direct method. Turgeon (1981) puts it at four storages, whereas Lamontagne and Stedinger (2013) says three. Labadie (2004) gives an overview of methods for dealing with multi-reservoir modelling.

Turgeon (1980), Turgeon (1981), Archibald et al. (1997), and Archibald et al. (2006) have developed an effective method for dealing with multi-reservoir systems, aggregation and decomposition. In this method reservoirs are optimised one-at-a-time to determine their target release, with other reservoirs being aggregated to form composite reservoirs. Archibald et al. (2006) suggests a 4-group method, with the target reservoir as a single state, all reservoirs upstream composited into a single state, all reservoirs parallel to the target composited as a single state, and all reservoirs downstream stream composited as a single state. After the target release policy is determined for the target reservoir, a new target reservoirs is selected, and the previous target is composited into the appropriate composite reservoir. This process is repeated until it converges on a policy that is feasible. These aggregated methods are efficient in that they consider no more than 4 storages at one time, however it does have the draw back of finding a policy that has an expected monetary value slightly below the maximum (Archibald et al., 1997).
5.5 Hidden Markov Models

Early modelling of rainfall with Markov chains used basic observed state models, for example Gabriel and Neumann (1962) and Stern and Coe (1984). Here the probability of rain is conditional on whether it had rained or not the previous, one, two, or three days depending on the number of states included in the model. Whilst models with increased state space are useful, they have an associated the risk of over-fitting and therefore of not having any predictive power as they only explain the observed data. Another drawback of this type of modelling is that for each additional time-step of lag requires a doubling of a state space. Hidden Markov models are advantageous in that they have a similar structure without prescribing any physical meaning to the states, thus allowing for a more general structure.

Hidden Markov models appear to have been introduced to the hydrological literature by Zucchini and Guttorp (1991) then further expanded in MacDonald and Zucchini (1997) and have found wide-spread application in the hydrological literature, for example Thyer and Kuczera (2003a, 2003b), Lambert et al. (2003), and Akintuğ and Rasmussen (2005). However these have typically focussed on a continuous observation distribution as rainfall is normally treated as a continuous variable. Akintuğ and Rasmussen (2005) give a fairly typical example of this type of modelling. Letting $Y_k$ be the observed rainfall modelled by,

$$Y_k = \mu J_k + \sigma J_k \varepsilon, \quad \varepsilon \sim N(0,1),$$

where $J_k$ is an underlying hidden Markov process on a state space $X$, where each state $i \in X$ has its own mean rainfall, $\mu_i$, and standard deviation, $\sigma_i$. These models are not restricted to the normal distribution. Other distributions, such as the log-normal distribution (Lambert et al., 2003), can be used where each state of the Markov chain defines its own rainfall distribution dependent on that state.

Hidden Markov models are often used to model climatic states and their effects on rainfall, such as the El-Niño/La-Niña phenomena, which dominates the climatic conditions in the southern Pacific region and influences the rainfall on eastern Australia as shown extensively in Whiting (2003, 2004, and 2005).

However, as previously mentioned, all these models assume that the rainfall or riverflows are a continuous variables and to be incorporated in to a stochastic dynamic programming model, such as those laid-out in Sections 5.3 and 5.4, requires discretised observations to apply to the hidden Markov, such as those shown in Section 3.3.
Chapter 6

Synthesis

Having done a review of the historical and state-of-the-art hydrological literature in Chapter 5 this thesis will demonstrate how some of the innovative Markov modelling techniques discussed in Chapter 3 can be combined with the Markov decision processes outlined in Chapter 4 to great benefit.

The first aspect that will be explored is whether the phase-structure makes a noticeable difference in the solution to Markov decision processes and if so how much? The inclusion of phases into the stochastic dynamic programming models shown in Sections 5.3 and 5.4 should improve fitting and predictive power. However as previously mentioned, as with all fitting procedure, they can lead to over-fitting which results in a model which describes too much of the already realised data. To prevent over-fitting, initially the number of phases is limited to a number of physically measurable states. The Southern Oscillation Index (SOI), which measures the normalised difference in air-pressure between Tahiti and Darwin, is used as a phase description for the purpose of phase-type construction. This inclusion of phases demonstrates that it is beneficial and the results of this are shown in the paper Fisher et al. (2008) in Section 6.1.

Having established that phases have a beneficial effect when incorporated into a model, the thesis then investigates the benefits that matrix analytic models provide to hydrology and water resource management. Hydrology is already a well established field with many techniques and models of its own, however the usefulness these models from the matrix analytic literature can extended the field further. For example Cigizoglu et al. (2002) proposed a complicated model for rivers with spate flows which would be unsuitable for inclusion in a Markov decision processes. A Markov chain based model would be of great benefit. Matrix analytic methods provide such constructions and in this case a Markov modulated arrival processes as described in Latouche and Ramaswami (1999) is appropriate and tools already exist for their fitting (Rydén, 2000). The results from these considerations are shown in the paper Fisher et al. (2010) in Section 6.2.

The inclusion of hidden phases into the models appears essential if the full computational capabilities of matrix analytic methods are to be realised in hydrology. However,
what effects does having hidden phases have on the modelling and implementation of a Markov decision process? Whilst hidden Markov models have been used in hydrology before (see Section 5.5), they have been based on continuous variables and have not been used in conjunction with MDPs. The thesis shows directly how a system operates under a model with hidden information and how it compares to an equivalent known phase model. The results of this research is shown in the paper Fisher et al. (2012) in Section 6.3.

The final question that has been explored in this thesis is: can the first-passage time distribution be used as an operating criteria and if so how would this impact the operating policies found using this criteria? The motivation for this question comes from Young and McColl (2005) who advocate that water resources be managed with the objective of maintaining the long-term viability of the storage system first and then proportioning the releases based on the economic values placed on the water after basic human needs had been met. The objective criteria is then to maximise the number of steps until any undesirable state is reached or to minimise the number of steps until a target state is reached. This obviously has an impact on the economic out-put of the system as the operating policy will not be chosen to maximise expected monetary value as has been the focus of all previous research shown in Sections 5.3 and 5.4. The results of this research is shown in the paper Fisher et al. (2014) in Section 6.4.

The results as indicated above are now presented as a collection of papers as published in the literature.
6.1 Optimal Control of Multi-Reservoir Systems with Time-Dependent Markov Decision Processes

Paper highlights:

- A seasonal (time-dependent) Markov decision process conditioned on the southern oscillation index is applied to the Wivenhoe Reservoir, a flood-control and water-storage reservoir in Southern Queensland.
- The method is extended to Wivenhoe and Somerset as a 2-reservoir system.

The paper *Optimal Control of Multi-Reservoir Systems with Time-Dependent Markov Decision Processes* (Fisher et al., 2008) establishes the use of a phase-structure in a Markov decision processes where transition probabilities change with time. Recall from Section 3.2.1 that a matrix analytic model describes a doubly stochastic process \{\(N_k, J_k\) : \(k \in \mathbb{Z}_+\)}.

The first model, a single reservoir storage with stochastic inflows from a river along with a water recycling transfer. Here the levels of the reservoir storage are represented by the level process of the MAM model \(N_k\). The transition probabilities are conditioned on an underlying hydrological state, the Southern Oscillation Index (SOI), which is used as a proxy measure of the El-Niño/La-Niña phenomenon that effects the prevailing weather condition on the Eastern seaboard of Australia. The SOI is included as a phase process within each level of the reservoir model.

The second model incorporates the upstream reservoir, Somerset that flows into the lower reservoir. Here an approximation using a triply stochastic system \{(\(N^2_{N^1_k}, N^1_k, J_k\) : \(k \in \mathbb{Z}_+\)} is used to model the system, where \(N^2_{N^1_k}\), which is dependent on \(N^1_k\), the level of the upstream dam, and \(J_k\) the independent hydrological state determined by the SOI.

The matrix analytic model leads to a better performance due to its ability to incorporate additional information of the SOI in the structure.
*In: Water Down Under.*

NOTE:
This publication is included on pages 73-82 in the print copy of the thesis held in the University of Adelaide Library.
6.2 Managing River Flow in Arid Regions with Matrix Analytic Methods

Paper highlights:

- Matrix analytic methods used for modelling arid flow in Cooper Creek conditioning on both hidden states and known states.
- Discussion of phase-type distribution in the context of Markov decision processes.

The paper *Managing River Flow in Arid Regions with Matrix Analytic Methods* (Fisher et al., 2010) introduces further concepts of matrix analytic methods to the hydrological literature.

In the first part of the paper a matrix analytic model for an ephemeral river flow is considered, with the levels \( N_k \) of the process representing the current flow in the river. The river had long sojourn times between spates of flow arrivals which are modelled using a discrete phase-type distribution. The flows are then modelled as a batch arrival process using historical count data conditioned on the previous arrival. The spates have two phases, described by the process \( J_{N_k} \), which determines whether the last flow was an increase or decrease on the previous flow. This phase process induces a correlation structure on the size of the flows.

In the second part of the paper, a model is used to demonstrate the phase structure, where counting processes were used to fit the model parameters that give a direct physical meaning to the phases. The use of general fitting with expectation-maximisation algorithms is further developed and shown in Section 3.4.

NOTE:
This publication is included on pages 85-94 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

6.3 Modelling of Hydrological Persistence for Hidden State Markov Decision Processes

Paper highlights:

- Comparison of a decision process based on a hidden Markov model with a decision process based on the climatic indicator given by the Southern Oscillation Index for the Wivenhoe Reservoir in Southern Queensland.

The paper *Modelling of Hydrological Persistence for Hidden State Markov Decision Processes* (Fisher *et al.*, 2012) is a comparative study of known vs hidden phases in Markov decision processes, in particular where it relates to hydrological applications. The use of state variables are common in hydrological applications but in some situations such a variable is not always available.

In this paper, a hidden state Markov decision processes (HSMDP) is compared to a model constructed using an assumed dependence on the Southern Oscillation Index. The two models are compared in the context of decision processes for releases from the Wivenhoe, Somerset system of reservoirs in Southern Queensland.

The paper finds an almost identical result for the HSMDP as the one informed by the SOI information, however in one instance the HSMDP was surprisingly less conservative. In a practical application caution would be needed in implementing a planning based primarily on hidden phase information as the hidden phase can only be estimated as a probability and if the policy is less conservative it could lead to more failures of the system.
6.4 First-Passage Time Criteria for the Operation of Reservoirs

Paper highlights:

- The Abberton and Hanningfield two-reservoir system is studied using seasonal Markov decision processes.
- First-passage time criteria are introduced into Markov decision processes for hydrological applications.
- Pareto optimisation is used to compare first-passage time criteria to expected monetary value.

The paper *First-Passage Time Criteria for the Operation of Reservoirs* (Fisher et al., 2014) takes the work of Jianyong and Huang (2001) and looks at it in both a matrix analytic methods and hydrological setting. Jianyong and Huang (2001) proposed using the first-passage time until a system reaches failure as criteria for optimising its operating policy. However they did not appear to notice the link between first passage time of a Markov chain and the standard Bellman equation used in calculating the reward.

Consider:

\[
v_k(Z_k) = \max_{d_k \in D} \left\{ \lambda(d_k, Z_k) + \sum_{Y_{k+1} \in D} \Pr[Y_{k+1}] \left[ v_{k+1}(Z_{k+1}) \right] \right\},
\]

where if for the states where \( Z_k \) is a failure of the reservoir system, \( v_k(Z_k) = 0 \) for all \( k \), and \( \lambda(d_k, Z_k) = 1 \), a system of equations that calculates the first-passage times for the system is obtained. This can be used iteratively to find a policy, which maximises the first-passage times. These first-passage times can also be calculated with the phase-type distribution as is discussed in the paper.

Preliminary research focussed on maintaining the reservoir amenity outside failure either by overflow or the reservoir reaching dead storage. This however made for predictable results. Instead, the applications as part of multi-objective programming were considered for the paper.

In multi-objective programming there are competing reward structures that are considered. Typically policy decisions that are optimal for one will not be optimal for another reward structure. Instead “compromise” policies will be needed. Pareto optimality is instead the criteria used to determine whether a policy should be considered. This means that policies that are sub-optimal for both criteria will need to be found. This has typically been done using genetic algorithms following on from Huang and Guo (2009) or simulated annealing Teegavarapu and Simonovic (2002). However in the paper a simpler method based on the Metropolis-Hastings algorithm is used instead. The result of using the Metropolis-Hastings algorithm causes the sampling to take place.
where the decision space is densest and as a result it is seen that perturbations do not have a large effect on the rewards, with only minimal losses.
*Journal of Hydrology, v. 519(Part B), pp. 1836-1847*

**NOTE:**
This publication is included on pages 109-120 in the print copy of the thesis held in the University of Adelaide Library.

It is also available online to authorised users at:

Chapter 7

Conclusion

In Chapter 6 five questions were proposed:

1. What benefits can matrix analytic models provide to hydrology and water resource management?

2. Whether the phase-structure makes a noticeable difference in the solution to Markov decision processes and if so how much?

3. What effects does having hidden phases have on the modelling and implementation of a Markov decision process?

4. How would a system operating under a model with hidden information perform and would it be severally detrimental compared to an equivalent known phase model?

5. Can the first-passage time distribution itself be an operating criteria and if so how would this impact the operating policies found using this criteria?

The four published papers (Fisher et al. 2008, 2010, 2012, and 2014) addressed each of these questions.

Benefits of Matrix Analytic Methods

Matrix analytic methods provide more generalised Markov chain modelling techniques that improve upon the standard single state per level chain with multi-phases per level incorporating additional information into the modelling processes. For example lag-one correlations (Fisher et al., 2010), climatic states (Fisher et al., 2010), or structural states that had no physical meaning but allowed for different flow probabilities (Fisher et al., 2012). Fisher et al. (2012) dealt with this question in the most detail, as it was the first paper to propose that hydrology could be done in a matrix analytic methods framework. The paper put forward three ideas in which matrix analytic methods could benefit hydrology.
The first was the use of the phase-type distribution to measure inter-arrival times. The phase-type distribution is advantageous in this respect as the discrete-time form can be made to fit very general distributions and is easily incorporated into a Markov decision process if desired. Its main disadvantage is that it can often require a large number of phases to fit data with any degree of accuracy. For example Faddy (1998) required 397 phases for a sufficiently good fit of inter-arrival times of eruptions of the Old-Faithful geyser in Yellowstone Park, something that can be satisfactorily fit with a mixture of two normal distributions for a total of five parameters (two mean, two standard deviation and the mixture probability). However, many of the parameters in Faddy’s model were restricted to either zero or being equal leaving just four parameters to estimate meaning that it was smaller in that respect even though it was larger in analytic expression and computer storage terms.

The second proposed idea was the use of phases with a direct physical interpretation to model a lag-one correlation of whether the flow had increased or decreased previously. The incorporation of this information improved the fit of the model, whilst keeping a Markov chain structure, but at the expense of doubling of the state space. The results of fitting hidden phases shown in Akintuğ and Rasmussen (2005) imply that a three-phase model is sufficiently good for hydrological applications, although this is possibly context dependent and warrants further investigation. Fitting models also presents a challenge as there is no readily available packages to do this, although Section 3.4 proposes methods that are adaptable enough to do this fitting.

The final idea presented in Fisher et al. (2010) was the recasting of seasonal models proposed by Butcher (1971) as a periodic Markov chain. The computational advantage for this conceptualisation was dealt with in Fisher et al. (2014) where algorithms exploiting this structure were presented.

**Incorporation of Known and Hidden Phases into Markov Decision Processes**

Questions two through to four relate to the incorporation of phases into Markov decision processes and were dealt with primarily in Fisher et al. (2008) and Fisher et al. (2012). The first question regarding the inclusion of known phases in the Markov decision process model gave the intuitive result that the model performed better with more information. In the same way, the Stedinger et al. (1984) forward-forecasting model discussed in Section 5.3 gives better management decision by including meteorological predictions. However the model in Fisher et al. (2008) has the benefit of a smaller state space expansion and due to the El Niño/La Niña’s long-term persistence, allows for longer time-steps in the model.

McMahon (2008) demonstrates that hidden phases included into a Markov decision process can potentially lead to an overall suboptimal operation of the system and that the current phase needs to be estimated in order for it to be implemented. In Fisher et al. (2012) it is shown that by using hidden Markov models, the current hidden phase can be estimated with the Viterbi algorithm and implemented with only a 0.2% loss in expected monetary value in that particular example.
Incorporation of First-Passage Time Distributions into Markov Decision Processes

The final question deals with the use of first-passage time as an optimisation criterion. As previously mentioned in Chapter 6, the motivation comes from the fact that every Markov chain’s first-passage times between states can be used to either minimise the expected number of time-steps to reach a target state (as done in Fisher et al. 2014) or used to maximise the number of time-steps to reach an undesirable state. As mentioned in Fisher et al. (2014), when used as the only optimisation criteria, it performs best when used to keep the system within an operating window between two states. However, when used as part of a multi-objective optimisation in conjunction with expected monetary value it can used to find a policy that is a compromise between the two using a Pareto optimisation where only minimal difference below the optimal value found with either criteria in isolation.

7.1 Further Work

In all the applications considered in this thesis the fundamental property of interest after establishing an appropriate model is the stationary or invariant distribution of the Markov chain that describes the system operating under a fixed policy. From this distribution, the long-term reward for the system can be calculated for a single or multi-objective optimisation. Calculating this distribution as efficiently as possible leads to the ability to construct more realistic models. In Fisher et al. (2014) (Section 6.4) this was addressed with methods for reducing the calculation time by one order of magnitude for monthly seasonal models. Extending this, another efficient method may have its beginnings somewhere in the earliest parts of the literature. The Moran Dam Model was designed primarily to find the capacity a reservoir would need to have in order to hold back an annual inundation and then release some fixed amount. The model’s elegance lies in the fact that it is analytically similar to simple Markov models, most noticeably those of M/G/1 and G/M/1 type, this allowed the stationary distribution to be quickly calculated with just the knowledge of the inflow distribution. The Moran Dam Model uses a fixed release for all states, however the system of equations can be amended to incorporate different release decisions for each state. Even with this alteration the Moran Dam Model fits into the matrix analytic framework as it is still a skip-free process and can be represented as a quasi birth and death process.

Expanding the Moran Dam Model for multi-reservoir systems can be achieved by increasing the state space. The most natural way of doing this is by embedding one reservoir into the next downstream dam using a Kronecker product (this was done in both Sections 6.1 and 6.4). Without any state aggregation, this results in a matrix for a multi-reservoir system that is dimensionally very large. Goulter and Tai (1985) suggest that somewhere between 15 and 20 states is needed to describe a reservoir without compromising the effectiveness of the model. This means that the Markov chain for modelling $n$ reservoirs would need about $20^n$ states. In addition, any auto-correlation
that needs to be included will further increase the state-space by a factor equal to the
dimension associated with the flow.

However, expressed as a $QBD$, the Moran Dam Model has a tri-diagonal block struc-
ture. The expanded embedded model also has a tri-diagonal block structure, with the
actual transition matrix being very sparse. It is here that matrix analytic methods can
be used to implement the policy iteration method 4.3 or the stochastic methods 4.4.
Neuts (1981) and Latouche and Ramaswami (1999) provide a number of algorithms for
calculating stationary distributions for matrix analytic models. Exploiting these well-
defined structures to calculate the stationary distributions should allow for specialised
Markov decision process algorithms that can go beyond the currently accepted four
reservoir limitation of MDPs (as discussed in Section 5.4). Given that the Moran Dam
Model can be put into the matrix analytic framework, this is a promising approach
into the future.

Another critical issue in modelling hydrological systems is seasonality, which results in
the Markov chain being non-ergodic, as it demonstrates periodic behaviour. Strictly
speaking a periodic Markov chain has no limiting distribution, however, a stationary
regime can be found, which has allowed policy iteration to be used to solve seasonal
Markov decision processes in the literature. It is this seasonality that prevents the
easy application of many established algorithms that rely on the ergodic property.
For example, Piantadosi et al. (2010) only considered non-seasonal behavior, whereas
in Fisher et al. (2014) the algorithms used actually exploited this cyclic structure of
the seasonal model in order to ease the calculations. Finding enhanced methods for
exploiting both the cyclic pattern and embedded tri-diagonal block structures in order
to further ease the calculation of the stationary distribution is a task for the future.

The inclusion of more elaborate costings that include for example hydroelectricity
generation, spot-costs, and renewable energy storage, also will present new challenges
especially in the framework of multi-reservoir systems.
Appendix A

Supplementary Material for Chapter 3

A.1 Supplementary Material for Section 3.3

A.1.1 Demonstration of Equation (3.7)

Let $J_{k-1} = i$ for $i \in X$. Equation (3.6) can be rewritten as,

$$\gamma_k(j) = \max_{\tilde{J}_{k-2},i} \left\{ \Pr[\tilde{J}_{k-2}, J_{k-1} = i, J_k = j, \bar{Y}_k | P, Q, \pi] \right\}$$

$$= \max_{\tilde{J}_{k-2},i} \left\{ \Pr[J_k = j, Y_k | \tilde{J}_{k-2}, J_{k-1} = i, \bar{Y}_{k-1}, P, Q, \pi] \times \Pr[\tilde{J}_{k-2}, J_{k-1} = i, \bar{Y}_{k-1} | P, Q, \pi] \right\}$$

$$= \max_{\tilde{J}_{k-2},i} \left\{ \Pr[Y_k | J_k = j, \tilde{J}_{k-2}, J_{k-1} = i, \bar{Y}_{k-1}, P, Q, \pi] \times \Pr[J_k = j | \tilde{J}_{k-2}, J_{k-1} = i, \bar{Y}_{k-1}, P, Q, \pi] \times \Pr[\tilde{J}_{k-2}, J_{k-1} = i, \bar{Y}_{k-1} | P, Q, \pi] \right\}$$
Now consider that the process $J_k$ is Markovian and that $Y_k$ is dependent only on $J_k$.

$$
\gamma_k(j) = \max_{J_{k-2},i} \left\{ \Pr[Y_k|J_k = j, P, Q, \pi] \Pr[J_k = j|J_{k-1} = i, P, Q, \pi] \right. \\
\left. \quad \times \Pr[J_{k-2}, J_{k-1} = i, \bar{Y}_{k-1}|P, Q, \pi] \right\} \\
= \max_{J_{k-2},i} \left\{ q_{jY_k}p_{ij} \Pr[J_{k-2}, J_{k-1} = i, \bar{Y}_{k-1}|P, Q, \pi] \right\} \\
= \max_i \left\{ q_{jY_k}p_{ij}\gamma_{k-1}(i) \right\} \\
= \left[ \max_i \{p_{ij}\gamma_{k-1}(i)\} \right] q_{jY_k}
$$

Hence Equation (3.7) is obtained.

### A.1.2 Demonstration of Equation (3.9)

First rewrite Equation (3.8) as,

$$
\xi_k(i, j) = \Pr[J_k = i, J_{k+1} = j|Y, P, Q, \pi] \\
= \frac{\Pr[J_k = i, J_{k+1} = j|Y, P, Q, \pi]}{\Pr[Y|P, Q, \pi]}
$$

Concentrating on the enumerator,

$$
\Pr[J_k = i, J_{k+1} = j, Y|P, Q, \pi] \\
= \Pr[J_k = i, \bar{Y}_k|P, Q, \pi] \times \Pr[J_{k+1} = j, \bar{Y}_{k+1}|J_k = i, P, Q, \pi] \\
= \Pr[J_k = i, \bar{Y}_k|P, Q, \pi] \times \Pr[J_{k+1} = j|J_k = i, P, Q, \pi] \\
\times \Pr[\bar{Y}_{k+1}|J_{k+1} = j, P, Q, \pi] \\
= \Pr[J_k = i, \bar{Y}_k|P, Q, \pi] \times \Pr[J_{k+1} = j|J_k = i, P, Q, \pi] \\
\times \Pr[Y_{k+1}|J_{k+1} = j, P, Q, \pi] \times \Pr[\bar{Y}_{k+2}|J_{k+1} = j, P, Q, \pi] \\
= \alpha_k(i)p_{ij}q_{jY_{k+1}}\beta_{k+1}(j).
$$
Likewise for the denominator,

\[
\Pr[Y|P,Q,\pi] = \sum_{i=1}^{N} \sum_{j=1}^{N} \Pr[J_{k} = i, J_{k+1} = j, Y|P,Q,\pi] \\
= \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_k(i)p_{ij}q_{j,Y_{k+1}}\beta_{k+1}(j).
\]

Which gives the result,

\[
\xi_k(i,j) = \frac{\Pr[J_{k} = i, J_{k+1} = j, Y|P,Q,\pi]}{\Pr[Y|P,Q,\pi]} = \frac{\alpha_k(i)p_{ij}q_{j,Y_{k+1}}\beta_{k+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_k(i)p_{ij}q_{j,Y_{k+1}}\beta_{k+1}(j)}.
\]

Hence Equation (3.9) is obtained.
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>State diagram of a $N = 3$ Markov chain.</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>State diagram of the discrete phase-type distribution.</td>
<td>25</td>
</tr>
<tr>
<td>3.2</td>
<td>A diagram of a hidden Markov model.</td>
<td>33</td>
</tr>
<tr>
<td>4.1</td>
<td>The crossover in two policy vectors from the parents to the offspring.</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>Reconstruction of the results of Goulter and Tai (1985) on the effect on</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>the expected gain by storage space discretisation.</td>
<td></td>
</tr>
</tbody>
</table>
List of Tables

1.1 Re-creation of the table “Share of global runoff and population by continent” shown in Postel et al. (1996). ........................................... 3

3.1 Transition probability matrix $P$ for the transitions between hidden states. 33

3.2 Observation probability matrix $Q$ for the observations given the hidden states. ................................................................. 33
Bibliography


