Efficient Computations for a Class of Markov Chains and Related Applications

Smit, L.C.

L.C.Smit

Efficient Computations for a Class of Markov Chains and Related Applications

Master thesis, defended on August 30, 2011

Thesis advisors:

Dr. F.M.Spieksma (Leiden)

Prof. M.N.Katehakis (Rutgers)

Specialisation: Applied Mathematics

Mathematisch Instituut,
Universiteit Leiden

Rutgers Business School,
Newark and
New Brunswick, NJ

Abstract

This thesis introduces the successive lumping procedure (SLP) to compute the steady state probabilities for a class of Markov Chains with large state spaces. In this procedure we introduce one or multiple successive lumping states processes. Also, this thesis studies the classical reorder quantity, order point (Q, r) continuous review stochastic inventory model. This model has been extensively studied in the literature and its use in practice is widespread. Using the SLP efficient calculations can be done for this model when there are Bernoulli arrivals, a mixture of backorders, lost sales and a random lead time. In addition, this work extends previous research in this area by providing efficient algorithms for the computation of the optimal (Q^*, r^*) values when there are Poisson arrivals, a multi-breakpoint discount pricing structure and a fixed lead time.

Contents

1 Introduction **5**

1.1 Preface 5

1.2 Outline 6

2 Successively Lumpable Markov Chains **7**

2.1 Introduction 7

2.2 Definitions 7

2.3 The Successive Lumping Method 11

2.4 Algorithm 14

2.5 A small state space example 14

2.6 Multiple Successively Lumpable Markov Chains 19

2.7 Algorithm and Example 23

3 Application on Inventory Management **25**

3.1 Model Description 25

3.2 Approach of the Model with Successive Lumping 28

3.3 Calculations 29

3.4 Unimodality of the costfunction 34

4 Quantity Discounts **35**

4.1 Introduction 35

4.2 Assumptions, Notation and Preliminaries 35

4.3 Quantity Discounts 38

4.4 Algorithms 44

4.5 Computations 46

4.6 Appendix A 49

5 Concluding Remarks **55**

Chapter 1

Introduction

1.1 Preface

This thesis has been written as the final stage of my master Applied Mathematics in Leiden. To do research in the field of Inventory Management, Professor Michael Katehakis affiliated with the Department of Management Science and Information Systems of Rutgers University in Newark, New Jersey, agreed to guide me in this process. I am very grateful for this.
I have spent half a year in Newark, January 2011 until July 2011, and together with Professor Katehakis I have written 2 papers, one of which is currently under review and the other will be submitted shortly. The first paper is Chapter 4 of this Thesis and has been submitted to*Annals of Operations Research* with the title:

On Computing Optimal (Q, r) Replenishment Policies under Quantity Discounts.

Chapters 2 and 3 are on the edge of being submitted together to *Operations Research.* We do not have a separate title for this paper yet.

This Master thesis handles an efficient way to compute steady state probabilities of a certain class of Markov chains. Basically the main idea is to "lump" states in a consecutive way. Lumping has been done before, but as far as literature research has shown, this idea is new. The class of Markov Chains for which this successive lumping method is possible can be used in many different applications. The example that is studied thoroughly in this thesis is from Inventory Management, a specialization of Professor Katehakis.

My research direction was yet not fixed when I started in Newark. First I spent some time on analyzing the known (Q, r)-inventory model, which is the basis for Chapter 4, our first paper. The use of Quantity Discounts in the so-called EOQ model (no lead time, $r = 0$ and deterministic demand) in [7], and a formula for the optimal value of Q was derived. I generalized this to the $(Q, r)-$model.
Chapter 4 has been written first and has been the inspiration for Chapter 3. The concept of inventory position and inventory level explained in Chapter 4, for example used by [2], has been the inspiration for how to find a suitable state description for the model studied in Chapter 3. However, because of the mixture of lost sales and backorders and therefore the loss of uniformity, we had to modify the model on some points to do fast calculations to find the steady state

probabilities. After creating this model we extended our method steady state calculations to handle a larger class of Markov Chains and named this class successively lumpable Markov Chains.

1.2 Outline

Because this thesis is a combination of 2 separate papers, there is some discrepancy between Chapters 2 and 3 on one hand and Chapter 4 on the other. For example in Chapter 4, $P(X = i)$ is denoted by p_i, while in Chapters 2 and 3, because of the use of double (or triple) indices this probability is denoted with $p(i)$. Also, some definitions are explained twice or are stated slightly different, since the inventory model introduced in Chapter 3 is almost the same as the inventory model in Chapter 4. For simplicity of the exposition we will assume that the Markov Chain is irreducible and all states are positive recurrent.
Although Chapter 4 has been written before Chapters 2 and 3, we have chosen for the present order, since Chapters 3 and 4 handle almost the same model and Chapter 2 is a tool for the application in Chapter 3.

Chapter 2 introduces the concept of a successively lumpable Markov Chain. In Section 2.2 single lumping, used by [3] is explained, with different notation. In Section 2.3 a successively lumpable Markov Chain is defined and various properties for a successively lumpable Markov Chain are proven. In Section 2.4 an algorithm is introduced (a result of Section 2.3) and followed by a small example in Section 2.5. Section 2.6 generalizes the main result of [3] by showing that the successive lumping method can be used on different parts of a Markov Chain simultaneously. Such a Markov Chain will be called a multiple successively lumpable Markov Chain. Section 2.7 gives the algorithm and an example.

Chapter 3 gives the major application for the successive lumping method introduced in Chapter 2: an inventory replenishment model with random lead time and a combination of backorders and lost sales. This application has been the inspiration for the successive lumping method. Unfortunately until now it has not been possible to prove convexity or unimodality over Q and r, although Matlab graphs show that these properties are very likely to hold, as is shown on the end of this chapter. Section 3.1 explains the model and introduces the notation. Section 3.2 shows how successive lumping can be used to derive the steady state probabilities in a different way. Section 3.3 explains the different parameter regions that we need to study and how the calculations differ per area. Section 3.3 shows a Matlab graph of the model for a specific parameter choice to show the high probability of unimodality.

Chapter 4 contains the "basic" (Q, r)-model. However now the purchase price is taken into account, since buying a large quantity at once might be cheaper: so-called quantity discounts. In Section 4.2 we present the algorithm for finding the optimal Q and r, the proofs can be found in the appendix of this chapter. In Section 4.3 both all-units discount (discount on every product) and the incremental discount (discount on extra ordered products) are discussed and a method for finding the optimal strategy are constructed. Further in Section 4.3.3 a new efficient computation for finding the minimal value of a unimodal function is derived, under an extra assumption on the fixed single costs per backordered product. This chapter continues with the algorithms and an example in Sections 4.4 and 4.5 and ends with some proofs in Appendix A in Section 4.6.

This thesis ends with some concluding remarks in Chapter 5.

Chapter 2

Successively Lumpable Markov Chains

2.1 Introduction

In this Chapter we will introduce a new form of lumping, also known as state aggregation. Lumping was first introduced in [8] in 1960. State aggregation is well known and used for many applications. The process of aggregation/disaggregation is thoroughly described in [17] and [13]. In this Chapter we will show how we will lump states in a successive way and show that lumping these states does not effect the steady state probabilities of the other states. The idea is to use the mandatory state principle, which has been studied before in [3], [10] and [11].
Our method is different from the previous work for its successive approach, by creating new artificial (lumped) mandatory states, which we will call entrance states. This process will be described in Section 2.3.
In [3] a process with multiple mandatory states is described, each being the entrance state of a bigger set. We will show that this idea can be expanded to a successive process within these sets in Section 2.6. Both processes will be clarified with an example.

2.2 Definitions

Recall that we will assumed that all Markov Chains are irreducible and all states are positive recurrent. Further,in this chapter we will use the following notation: $\underline{\underline{A}}$ will denote a matrix, $\underline{a}(i)$ its i-th row vector and $a(i,j)$ its (i,j)-th element.
Let $X(t)$ be a Markov chain on a finite (or countable) state space $\mathcal{X}$. We will assume that the state space $\mathcal{X}$ can be partitioned into a (possibly infinite) sequence of mutually *exclusive* and *exhaustive* sets $\mathcal{D} = \{D_0, D_1, \ldots, D_M\}$, i.e, $M \leqslant \infty$, $\cup_{k=0}^{M} D_k = \mathcal{X}$, and $D_k \cap D_{k'} = \emptyset$, when $k \neq k'$. For notational convenience, the elements of each set D_k will be denoted (relabelled) as $\{(k,1),(k,2),\ldots,(k,\ell_k)\}$, for some fixed constants $\ell_k \leqslant \infty$. The transition matrix of $X(t)$ will be denoted by $\underline{\underline{P}} = [p(k',j' \,|\, k,j)]$, where the $((k,j)\,,(k',j'))$ element is

$$p(k',j' \,|\, k,j) = \mathbf{Pr}[X(t+1) = (k',j') \,|\, X(t) = (k,j)].$$

We assume that the stationary probabilities $\pi(k,j) = \lim_{t\to\infty} \mathbf{Pr}[X(t) = (k,j)]$ exist. We will use the notation

$$\underline{\pi} = \big(\pi(0,1),\ldots,\pi(0,\ell_0),\pi(1,1),\ldots,\pi(1,\ell_1),\ldots,\pi(M,1),\ldots,\pi(M,\ell_M)\big).$$

It is well known that $\underline{\pi}$ is the solution to the following system of equations: $\underline{\pi}\,\underline{\underline{P}} = \underline{\pi}$ and $\underline{\pi}\,\underline{1}' = 1$. Here $\underline{1}$ will denote a vector of ones of the same dimension as $\underline{\pi}$.

We state the following definition for partition $\mathcal{D}$.

Definition 2.1. *A subset D_m of $\mathcal{D}$ has an* ***entrance state*** $(m,\varepsilon(D_m)) \in D_m$ *if and only if*

a) $p(m,j\,|\,m',j') = 0$, *for all* $m' \neq m$ *with* $j \neq \varepsilon(D_m)$,

b) if the partition $\mathcal{D}$ has at least two sets, then there exists some $(m',j') \in D_{m'}$ with $m' \neq m$ such that $p(m,\varepsilon(D_m)\,|\,m',j') > 0$.

Remark 2.1. Note that because of the positive recurrent assumption the Definition 2.1b) is always true. It is been added for emphasis.

Note that an entrance state of a set D_m is the *only* state via which the set D_m can be entered by the process $X(t)$ from a state in $\mathcal{X}\backslash D_m$, where given two sets A and B, $A\backslash B$ denotes the elements of A that do not belong to B.

Given a partition $\mathcal{D}$ with an entrance state $(0,\varepsilon(D_0)) \in D_0$ we construct the following Markov chains.

a) A Markov process $Z_0(t)$ with state space D_0 and transition matrix $\underline{\underline{U}}_{D_0}$ with elements

$$u_{D_0}(0,j\,|\,0,i) = \begin{cases} p(0,\varepsilon(D_0)\,|\,0,i) + \displaystyle\sum_{(k,\ell)\notin D_0} p(k,\ell\,|\,0,i), & \text{if } j = \varepsilon(D_0),\\ p(0,j\,|\,0,i), & \text{otherwise.} \end{cases} \tag{2.1}$$

b) A Markov process $X_1(t)$ with state space $\mathcal{X}_1 = \{(1,0)\} \cup D_1 \cup \ldots \cup D_M$ and transition matrix $\underline{\underline{P}}_1$ where its $((k,j),(k',j'))$-th element is defined by Eq. (2.2) below if $(k,j) = (k'j') = (1,0)$ and by Eq. (2.3), in case (k,j) or $(k'j')$ is equal to $(1,0)$, but not simultaneously.

$$p_1(1,0\,|\,1,0) = \sum_{(0,\ell'),\,(0,\ell)\in D_0} p(0,\ell'\,|\,0,\ell)v_{D_0}(0,\ell), \tag{2.2}$$

$$p_1(k',j'\,|\,k,j) = \begin{cases} \displaystyle\sum_{(0,\ell)\in D_0} p(k',j'\,|\,0,\ell)v_{D_0}(0,\ell), & \text{if } (k,j) = (1,0),\\ \displaystyle\sum_{(0,\ell)\in D_0} p(0,\ell\,|\,k,j), & \text{if } (k',j') = (1,0),\\ p(k',j'\,|\,k,j), & \text{otherwise,} \end{cases} \tag{2.3}$$

where in Eqs. (2.2) and (2.3) above $v_{D_0}(0,\ell) = \lim_{t\to\infty} \mathbf{Pr}[Z_0(t) = (0,\ell)\,|\,Z_0(0) = (0,i)]$ are the steady state probabilities of the transition matrix $\underline{\underline{U}}_{D_0}$ the elements of which are defined by Eq. (2.1). Further,

$$\underline{\pi_1} = \big(\pi_1(1,0);\pi_1(1,1),\ldots,\pi_1(1,\ell_1),\ldots,\pi_1(M,1),\ldots,\pi_1(M,\ell_M)\big)$$

will denote the vector of the steady state probabilities of the process X_1. Note that with the above notation, state $(1,0)$ represents the lumped states of the set D_0. We have used a semicolon in the above notation to emphasize this fact.

We will use the notation $\underline{\underline{U}}_{D_0} = [\underline{u}'_{D_0}(0,1),\ldots,\underline{u}'_{D_0}(0,\ell_0)]$, where $\underline{u}'_{D_0}(0,j)$ denotes the j-th column of the matrix $\underline{\underline{U}}_{D_0}$. Similarly, $\underline{\underline{P}} = [\underline{p}'(0,1),\ldots,\underline{p}'(0,\ell_0),\ldots,\underline{p}'(M,1),\ \ldots,\underline{p}'(M,\ell_0)]$

We will next state and prove the following two propositions.

Proposition 2.1. *If D_0 has an entrance state $(0,\varepsilon(D_0))$, then the following is true for all $(0,i)\in D_0$:*

$$\upsilon_{D_0}(0,i) = \frac{\pi(0,i)}{\sum_{(0,\ell)\in D_0}\pi(0,\ell)}. \tag{2.4}$$

Proof. Let $\underline{\upsilon}_{D_0} = \big(\upsilon_{D_0}(0,1),\ldots,\upsilon_{D_0}(0,\ell_0)\big)$. It is clear that for $\underline{\upsilon}_{D_0}$, defined by Eq. (2.4), the statement $\underline{\upsilon}_{D_0}\underline{1}' = 1$ holds. Further, we will show that $\underline{\upsilon}_{D_0}$ also satisfies:

$$\underline{\upsilon}_{D_0}\underline{\underline{U}}_{D_0} = \underline{\upsilon}_{D_0}. \tag{2.5}$$

By unicity of solutions to Eq (2.5) it follows that $\underline{\upsilon}_{D_0}$ is indeed the steady state vector. We distinguish two cases: the entrance state $(0,\varepsilon(D_0))$ or any of the other states. For $(0,i) = (0,\varepsilon(D_0))$ we will use the following:

$$\begin{aligned}
\sum_{(0,j)\in D_0}\pi(0,j)p(0,\varepsilon(D_0)\,|\,0,j) =& \pi(0,\varepsilon(D_0)) - \sum_{(k,\ell)\notin D_0}\pi(k,\ell)p(0,\varepsilon(D_0)\,|\,k,\ell)\\
=& \pi(0,\varepsilon(D_0)) - \sum_{(k,\ell)\notin D_0}\bigl(1 - \sum_{(k',\ell')\notin D_0}p(k',\ell'\,|\,k,\ell)\pi(k,\ell)\\
=& \pi(0,\varepsilon(D_0)) - \sum_{(k,\ell)\notin D_0}\pi(k,\ell)\\
&+ \sum_{(k',\ell'),(k,\ell)\notin D_0}p(k',\ell'\,|\,k,\ell)\pi(k,\ell)\\
=& \pi(0,\varepsilon(D_0)) - \sum_{(k,\ell)\notin D_0}\pi(k,\ell)\\
&+ \sum_{(k',\ell')\notin D_0}\left(\pi(k',\ell') - \sum_{(0,j)\in D_0}p(k',\ell'\,|\,0,j)\pi(0,j)\right)\\
=& \pi(0,\varepsilon(D_0)) - \sum_{(0,j)\in D_0}\sum_{(k,\ell)\notin D_0}p(k,\ell\,|\,0,j)\pi(0,j).
\end{aligned}$$

Using this equality we obtain:

$$\begin{aligned}
\underline{\upsilon}_{D_0}\underline{u}'_{D_0}(0,\varepsilon(D_0)) &= \sum_{(0,j)\in D_0}\upsilon_{D_0}(0,j)u_{D_0}(0,\varepsilon(D_0)\,|\,0,j)\\
&= \frac{\sum_{(0,j)\in D_0}\pi(0,j)\left(p(0,\varepsilon(D_0)\,|\,0,j) + \sum_{(k,\ell)\notin D_0}p(k,\ell\,|\,0,i)\right)}{\sum_{(0,\ell)\in D_0}\pi(0,\ell)}
\end{aligned}$$

$$= \frac{\pi(0,\boldsymbol{\varepsilon}(D_0)) - \sum_{(0,j)\in D_0}\pi(0,j)\sum_{(k,\ell)\notin D_0}(p(k,\ell\,|\,0,j) - p(k,\ell\,|\,0,j))}{\sum_{(0,\ell)\in D_0}\pi(0,\ell)}$$
$$= \frac{\pi(0,\boldsymbol{\varepsilon}(D_0))}{\sum_{(0,\ell)\in D_0}\pi(0,\ell)}.$$

For $i \neq \boldsymbol{\varepsilon}(D_0)$:

$$\begin{aligned}
\underline{v}_{\,D_0}\,\underline{u}'_{D_0}(0,i) &= \sum_{(0,j)\in D_0} v_{D_0}(0,j)u_{D_0}(0,i\,|\,0,j)\\
&= \frac{1}{\sum_{(0,\ell)\in D_0}\pi(0,\ell)}\sum_{(0,j)\in D_0}\pi(0,j)p(0,i\,|\,0,j)\\
&= \frac{\pi(0,i)}{\sum_{(0,\ell)\in D_0}\pi(0,\ell)}\\
&= v_{D_0}(0,i)
\end{aligned}$$

so the proof is complete. □

For the process $X_1(t)$ we have the following result concerning the steady state distribution:

Proposition 2.2. *If D_0 has an entrance state $(0,\boldsymbol{\varepsilon}(D_0))$, then the following is true:*

$$\pi_1(k,j) = \begin{cases} \sum_{(0,\ell)\in D_0}\pi(0,\ell), & \textit{if } (k,j) = (1,0),\\ \pi(k,j), & \textit{otherwise.}\end{cases} \tag{2.6}$$

Proof. First, it is clear that $\underline{\pi}_1\,\underline{1}\,' = 1$. As in the previous proof we will show that $\underline{\pi}_1$ is the solution of the linear system

$$\underline{\pi}_1\,\underline{\underline{\mathrm{P}}}_{\,1} = \underline{\pi}_1.$$

For $(k,j) = (1,0)$:

$$\begin{aligned}
\underline{\pi}_1\,\underline{p}'_1(1,0) &= \sum_{(k',j')\in\mathcal{X}_1}\pi_1(k',j')p_1(1,0\,|\,k',j')\\
&= \sum_{(k',j')\in\mathcal{X}_1\setminus\{(1,0)\}}\ \sum_{(0,\ell)\in D_0} p(0,\ell\,|\,k',j')\pi(k',j')\\
&\quad + \sum_{(0,\ell),(0,\ell')\in D_0} p(0,\ell\,|\,0,\ell')v_{D_0}(0,\ell')\sum_{(0,\ell'')\in D_0}\pi(0,\ell'')\\
&= \sum_{(0,\ell)\in D_0}\ \sum_{(k,j)\in\mathcal{X}_1\setminus\{(1,0)\}} p(0,\ell\,|\,k,j)\pi(k,j) + \sum_{(0,\ell),(0,\ell')\in D_0} p(0,\ell\,|\,0,\ell')\pi(0,\ell')\\
&= \sum_{(0,\ell)\in D_0}\ \sum_{(k',j')\in\mathcal{X}} p(0,\ell\,|\,k',j')\pi(k',j')\\
&= \sum_{(0,\ell)\in D_0}\pi(0,\ell),
\end{aligned}$$

Similarly for states $(k, j) \neq (1, 0)$:

$$\begin{aligned}
\underline{\pi}_1 \underline{p}'_1(k,j) &= \sum_{(k',j')\in\mathcal{X}_1} \pi_1(k',j') p_1(k,j \,|\, k',j') \\
&= \sum_{(k',j')\in\mathcal{X}_1\setminus\{(1,0)\}} \pi(k',j') p(k,j \,|\, k',j') \\
&\quad + \sum_{(0,\ell)\in D_0} p(k,j \,|\, 0,\ell) v_{D_0}(0,\ell) \sum_{(0,\ell')\in D_0} \pi(0,\ell') \\
&= \sum_{(k',j')\in\mathcal{X}_1\setminus\{(1,0)\}} \pi(k',j') p(k,j \,|\, k',j') + \sum_{(0,\ell)\in D_0} p(k,j \,|\, 0,\ell) \pi(0,\ell) \\
&= \pi(k,j).
\end{aligned}$$

and the proof is complete. □

2.3 The Successive Lumping Method

For a process $X(t)$ with transition matrix $\underline{\underline{P}}$ and state space $\mathcal{X}$ for which there exists a partition $\mathcal{D} = \{D_0, \ldots, D_M\}$, with an entrance state $(0, \varepsilon(D_0))$ we have shown in the previous Section how to construct a lumped process $X_1(t)$ with state space $\mathcal{X}_1$ and transition matrix $\underline{\underline{P}}_1$ with the same steady state probabilities for all states not equal to a lumped artificial state $(1, 0)$. Further it was shown, c.f. Proposition 1, that the steady state probabilities of the original process $X(t)$ can be computed for all states in D_0 by proportionally distributing the steady state probability $\pi_1(1, 0)$ of the process $X_1(t)$ according to the weights of the steady state probabilities $\underline{v}_{D_0}$ of the process $Z_0(t)$. In this Section we show how to repeat this lumping procedure over the sets $D_1,\ D_2, \ldots$, successively.

We will need the following notation and definitions. Given a partition $\mathcal{D} = \{D_0, \ldots, D_M\}$, we define $\Delta_0 = D_0$, $\Delta_1 = \{(1, 0)\} \cup D_1$, $\Delta_m = \{(m, 0)\} \cup D_m$, where states $(m, 0)$ are artificial states that are representing sets $(\bigcup_{k=0}^{m-1} D_k)$ of lumped states. We further define the partition $\mathcal{D}_m = \{\Delta_m, D_{m+1}, \ldots, D_M\}$, for $m = 0, \ldots, M$. Also for notational consistency, we will use the notation: $X_0(t) = X(t)$, $\mathcal{X}_0 = \mathcal{X}$, $\mathcal{D}_0 = \mathcal{D}$, $\underline{\underline{P}}_0 = \underline{\underline{P}}$, and $\underline{\pi}_0 = \underline{\pi}$.

Given such a partition $\mathcal{D}_m$ we successively construct the following Markov chains.

a) A Markov process $Z_m(t)$ with state space Δ_m and transition matrix $\underline{\underline{U}}_{\Delta_m}$ with

$$u_{\Delta_m}(m,j \,|\, m,i) = \begin{cases} p_m(m,j \,|\, m,i) + \sum\limits_{(k,\ell)\notin\Delta_m} p(k,\ell \,|\, m,i), & if (m,j) = (m,0), \\ p_m(m,j \,|\, m,i), & otherwise. \end{cases} \tag{2.7}$$

b) A Markov process $X_{m+1}(t)$ with state space $\mathcal{X}_{m+1} = \{(m+1, 0)\} \cup D_{m+1} \cup \ldots \cup D_M$ and transition matrix $\underline{\underline{P}}_{m+1}$, with elements $((k, j), (k', j'))$ defined by Eq. (2.8) below if $(k, j) = (k', j') = (m + 1, 0)$ and by Eq. (2.9) otherwise.

$$p_{m+1}(m+1,0 \,|\, m+1,0) = \sum_{(m,\ell'),\,(m,\ell)\in\Delta_m} p_m(m,\ell' \,|\, m,\ell) v_{\Delta_m}(m,\ell), \tag{2.8}$$

$$p_{m+1}(k',j' \,|\, k,j) = \begin{cases} \displaystyle\sum_{(m,\ell)\in\Delta_m} p_m(k',j' \,|\, m,\ell) v_{\Delta_m}(m,\ell), & \text{if } (k,j) = (m+1,0), \\ \displaystyle\sum_{(m,\ell)\in\Delta_m} p_m(m,\ell \,|\, k,j), & \text{if } (k',j') = (m+1,0), \\ p_m(k',j' \,|\, k,j), & \text{otherwise,} \end{cases} \tag{2.9}$$

where in Eqs. (2.8), (2.9) above $v_{\Delta_m}(m,\ell) = \lim_{t\to\infty} \mathbf{Pr}[Z_m(t) = (m,\ell)]$ are the steady state probabilities of the transition matrix $\underline{\underline{U}}_{\Delta_m}$ defined in Eq. (2.7). Note again, that state $(m+1,0)$ above is the artificial state that represents all states in the set $\bigcup_{k=0}^{m} D_k$.

Further note that in order to compute $p_{m+1}(\cdot\,|\,\cdot)$ we first need to compute $v_{\Delta_m}(\cdot)$. The vector of the steady state probabilities of the process X_{m+1} will be denoted by:

$$\underline{\pi}_{m+1} = \big(\pi_{m+1}(m,0); \pi_{m+1}(m,1), \ldots, \pi_{m+1}(m,\ell_m), \ldots, \pi_{m+1}(M,1), \ldots, \pi_{m+1}(M,\ell_M)\big).$$

We will use the notation $\underline{\underline{U}}_{\Delta_m} = [\underline{u}'_{\Delta_m}(m,1), \ldots, \underline{u}'_{\Delta_m}(m,\ell_m)]$, and

$$\underline{\underline{P}}_m = [\underline{p}'_m(m,0); \underline{p}'_m(m,1), \ldots, \underline{p}'_m(m,\ell_m), \ldots, \underline{p}'_m(M,1), \ldots, \underline{p}'_m(M,\ell_M)].$$

We next state the following definition.

Definition 2.2. *A Markov Chain $X(t)$ is called **successively lumpable** with respect to partition $\mathcal{D} = \{D_0, \ldots, D_M\}$ if and only if a) D_0 has an entrance state $(0, \varepsilon(D_0))$ and b) $p(k',j' \,|\, k,j) = 0$ for all (k',j'), (k,j) with $k' < k$ and $(k',j') \neq (0,\varepsilon(D_0))$.*

The above condition means that a state in D_k can not be entered from a state in $D_{k'}$ when $k' > k > 0$. Note also that the definition implies that transitions out of states in D_k can only lead to states in $D_{k'}$ with $k' \geqslant k \geqslant 0$ or to the entrance state of the set Δ_0.

Remark 2.2. Every Markov Chain is successively lumpable with respect to a partition $\mathcal{D} = \{D_0, D_1\}$ where $D_0 = \{(0,\varepsilon(D_0))\}$, any single state and D_1 contains the remaining states.

Proposition 2.3. *If the process $X_0(t)$ with transition matrix $\underline{\underline{P}}_0$ is successively lumpable with respect to partition $\mathcal{D}_0$, then $X_m(t)$ with transition matrix $\underline{\underline{P}}_m$ is successively lumpable with respect to partition $\mathcal{D}_m$, for all $m = 1, \ldots, M$.*

Proof. To complete an induction proof we need to show that if $X_m(t)$ with transition matrix $\underline{\underline{P}}_m$ is successively lumpable with respect to partition $\mathcal{D}_m$, then $X_{m+1}(t)$ with transition matrix $\underline{\underline{P}}_{m+1}$ is successively lumpable with respect to partition $\mathcal{D}_{m+1}$.

For $m = 0$, conditions (a) and (b) of Definition (2.2) hold by assumption on $\underline{\underline{P}}_0 = \underline{\underline{P}}$. We assume the induction holds for $k = 0, \ldots, m$ we show it holds for $m+1$.

Next, we have defined $\underline{\underline{P}}_{m+1}$ (c.f., Eq. (2.9)) . To show that $(m+1,0)$ is the entrance state of Δ_{m+1}, it suffices to consider any other state $(m+1,j)$ of Δ_{m+1}, and show that $p_{m+1}(m+1, j \,|\, k,\ell) = 0$ when $(k,\ell) \notin \Delta_{m+1}$. Now by Eq. (2.9) we have

$$p_{m+1}(k',j' \,|\, k,j) = p_m(k',j' \,|\, k,j) = 0,$$

where the last claim is the induction hypothesis (b) . For condition (b) the proof is similar. □

We next state the following.

Corollary 2.1. *Under the assumption of Proposition 2.3 the following are true:*

$$\upsilon_{\Delta_m}(m,i) = \frac{\pi_m(m,i)}{\sum_{(m,\ell)\in\Delta_m} \pi_m(m,\ell)}. \tag{2.10}$$

$$\pi_{m+1}(k,j) = \begin{cases} \sum\limits_{(m,\ell)\in\Delta_m} \pi_m(m,\ell), & if (k,j) = (m+1,0), \\ \pi_m(k,j), & otherwise. \end{cases} \tag{2.11}$$

Proof. The proof is easy to complete by induction using a similar derivation as in Proposition 2.1 and 2.2, combined with the induction result of Propositions 2.3. □

Using a combination of the previous results, the following theorem holds.

Theorem 2.1. *If $X(t)$ is successively lumpable, with the number of partitions $M < \infty$, the following is true:*

$$\pi_0(m,j) = \upsilon_{\Delta_m}(m,j) \prod_{k=m+1}^{M} \upsilon_{\Delta_k}(k,0), \ \forall (m,j) \in \mathcal{X}_0$$

Proof. The proof follows by induction on decreasing values of $n = M, M-1\dots,0$ for fixed M. For $n = M$, we need to show that

$$\pi_0(M,j) = \upsilon_{\Delta_M}(M,j), \text{ for all } (M,j) \in D_M.$$

Indeed, by Corollary 2.1, we have $\upsilon_{\Delta_M}(M,j) = \pi_M(M,j)/1$, where the denominator is 1 because Δ_M contains all states of $\mathcal{X}_M$. Since $j \neq 0$, (i.e., (M,j) has never been lumped by our lumping procedure) using Corollary 2.1 repeatedly we obtain $\pi_M(M,j) = \pi_{M-1}(M,j) = \ldots = \pi_0(M,j)$, and the proof is complete for $n = M$.

We next show that the claim is true for $n = M-1$, i.e.,

$$\pi_0(M-1,j) = \upsilon_{\Delta_{M-1}}(M-1,j) \prod_{k=M}^{M} \upsilon_{\Delta_k}(k,0).$$

Indeed, the right hand side of the above is

$$\begin{aligned} \upsilon_{\Delta_M-1}(M-1,j)\upsilon_{\Delta_M}(M,0) &= \frac{\pi_{M-1}(M-1,j)}{\sum_{(M-1,\ell)\in\Delta_{M-1}} \pi_{M-1}(M-1,\ell)} \upsilon_{\Delta_M}(M,0) \\ &= \frac{\pi_{M-1}(M-1,j)}{\sum_{(M-1,\ell)\in\Delta_{M-1}} \pi_{M-1}(M-1,\ell)} \frac{\pi_M(M,0)}{\sum_{(M,\ell)\in\Delta_M} \pi_M(M,\ell)} \\ &= \pi_{M-1}(M-1,j), \end{aligned}$$

where the first two equalities follow from Corollary 2.1, Eq. (2.10) and the last one from Eq. (2.11) and the fact that $\sum_{(M,\ell)\in\Delta_M} \pi_M(M,\ell) = 1$, as before. The proof for $n = M-1$ is complete when we observe that $\pi_{M-1}(M-1,j) = \pi_{M-2}(M-1,j) = \ldots = \pi_0(M-1,j)$ since $j \neq 0$, as in the case $n = M$.

The induction on n is easy to complete using similar arguments. □

2.4 Algorithm

Using the method of the previous Section, we can state an algorithm for a successively lumpable Markov chain with respect to partition $\mathcal{D}$. This algorithm does not require a proof, since it uses Theorem 2.1 directly.

```
Algorithm SL
1     Construct $\underline{\underline{U}}_{D_0}$, c.f. Eq. (2.1).
2     Calculate $\underline{v}_{D_0}$.
3     Lump $D_0$ to $(1,0)$ and let $\Delta_1 = \{(1,0)\} \cup D_1$.
      Set $m = 1$.
      While $m \leqslant M$
4.1       Construct $\underline{\underline{U}}_{\Delta_m}$ c.f. Eq. (2.7).
4.2       Calculate $\underline{u'}_{\Delta_m}$.
4.3       Lump $\Delta_m$ to $(m+1,0)$ and let $\Delta_{m+1} = (m+1,0) \cup D_m$.
          $m = m + 1$
      End
5     Calculate $\underline{\pi}$ , c.f. Theorem 2.1.
```

2.5 A small state space example

We will clarify the previous results with a small example. Suppose we have a Markov chain $X(t)$. For clearance we will number the state space directly according to the notation introduced in Section 2.3, so: $\mathcal{X} = \{(0,1),(0,2),(1,1),(1,2),(2,1),(2,2),(3,1),(3,2),(3,3)\}$ and this transition diagram $\underline{\underline{P}}$.

$$\underline{\underline{P}} = \left[\begin{array}{c|ccccccccc}
 & (0,1) & (0,2) & (1,1) & (1,2) & (2,1) & (2,2) & (3,1) & (3,2) & (3,3) \\
\hline
(0,1) & 0 & 1/3 & 5/9 & 0 & 0 & 0 & 0 & 1/9 & 0 \\
(0,2) & 0 & 0 & 1/3 & 2/3 & 0 & 0 & 0 & 0 & 0 \\
(1,1) & 0 & 0 & 0 & 1/6 & 2/3 & 0 & 1/6 & 0 & 0 \\
(1,2) & 0 & 0 & 0 & 0 & 1/6 & 3/4 & 0 & 0 & 1/12 \\
(2,1) & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
(2,2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
(3,1) & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 \\
(3,2) & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\
(3,3) & 1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0
\end{array}\right]$$

We define a partition $\mathcal{D} = \{D_0, D_1, D_2, D_3\}$ with $D_0 = \{(0,1),(0,2)\}, D_1 = \{(1,1),(1,2)\}$, $D_2 = \{(2,1),(2,2)\}$ and $D_3 = \{(3,1),(3,2),(3,3)\}$. A graphical representation of $X(t)$ in Figure 2.1 directly shows that $X(t)$ is successively lumpable with respect to partition $\mathcal{D}$.

The first steps of the algorithm are:

1. $\underline{\underline{U}}_{\Delta_0} = \left[\begin{array}{cc} 2/3 & 1/3 \\ 1 & 0 \end{array}\right]$.

2. $\underline{v}_{\Delta_0} = [3/4, 1/4]$.

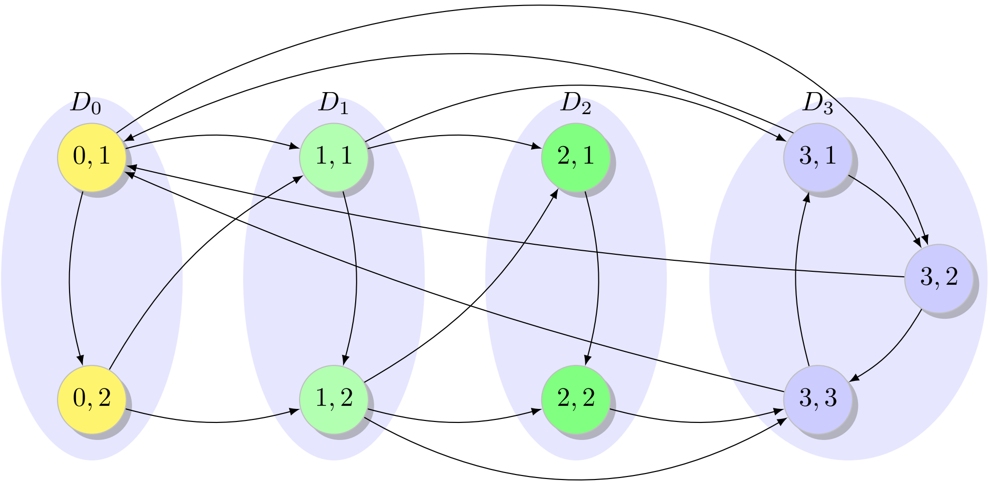


Figure 2.1: Representation of possible transitions under $\underline{\underline{P}}$.

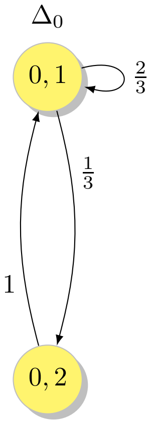


Figure 2.2: Graphical representation of $\underline{\underline{U}}_{\Delta_0}$.

Figure 2.2 illustrates this process. Now we continue with D_1, already distinguished in the picture and the numbering. We choose $D_1 \neq \emptyset$ in such a manner that for every $p(k', j' \mid k, j)$ with $(k', j') \in D_1$ we have $(k, j) \in D_0 \cup D_1$.

3 Lump $\{(0,1),(0,2)\}$ to $(1,0)$ and let $\Delta_1 = \{(1,0),(1,1),(1,2)\}$

4.1 $\underline{\underline{U}}_{\Delta_1} = \begin{bmatrix} 1/3 & 1/2 & 1/6 \\ 5/6 & 0 & 1/6 \\ 1 & 0 & 0 \end{bmatrix}$

4.2 $\underline{\upsilon}_{\Delta_1} = [4/7, 2/7, 1/7]$

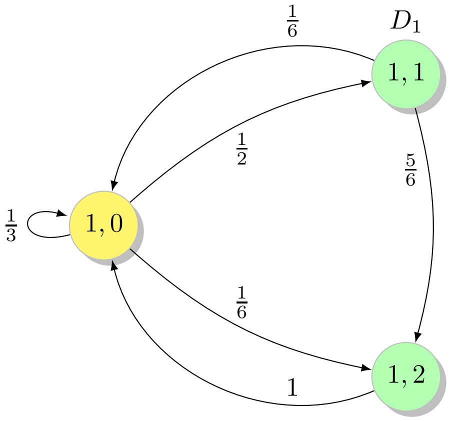


Figure 2.3: Graphical representation of $\underline{\underline{U}}_{\Delta_1}$.

Figure 2.3 illustrates this process.
Now, since we know $\underline{\upsilon}_{\Delta_1}$ we can construct transition probabilities of a set D_2 without knowledge of $\underline{\pi}$ with the use of the previous states $\{(1,0),(1,1),(1,2)\}$.

4.3 Lump $\{(1,0),(1,1),(1,2)\}$ to $(2,0)$ and let $\Delta_2 = \{(2,0),(2,1),(2,2)\}$

4.1 $\underline{\underline{U}}_{\Delta_2} = \begin{bmatrix} 19/28 & 3/14 & 3/28 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

4.2 $\underline{\upsilon}_{\Delta_2} = [28/43, 6/43, 9/43]$

Next we look at subset D_3 and repeat the previous.

4.3 Lump $\{(2,0),(2,1),(2,2)\}$ to $(3,0)$ and let $\Delta_3 = \{(3,0),(3,1),(3,2),(3,3)\}$

4.1 $\underline{\underline{U}}_{\Delta_3} = \begin{bmatrix} 93/129 & 4/129 & 4/129 & 4/129 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 0 & 0 \end{bmatrix}$

4.2 $\underline{\upsilon}_{\Delta_3} = [43/67, 142/1407, 104/1407, 248/1407]$

We advance to step 5 and calculate $\underline{\pi}$:

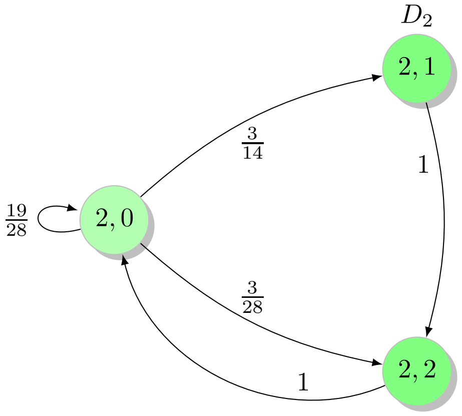


Figure 2.4: Graphical representation of $\underline{\underline{\mathrm{U}}}_{\Delta_2}$.

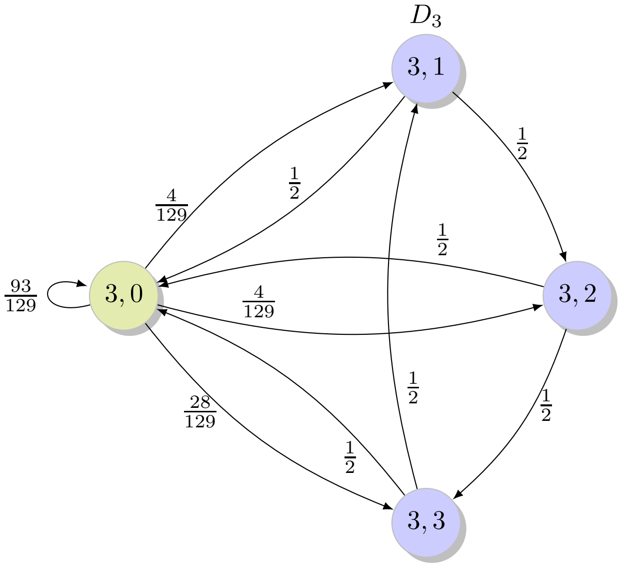


Figure 2.5: Graphical representation of $\underline{\underline{\mathrm{U}}}_{\Delta_3}$.

$$\begin{array}{rclcccc}
\pi(0,1) & = & \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,0)\upsilon_{\Delta_0}(0,1) & = & \frac{43}{67}\frac{28}{43}\frac{4}{7}\frac{3}{4} & = & \frac{12}{67} \\
\pi(0,2) & = & \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,0)\upsilon_{\Delta_0}(0,2) & = & \frac{43}{67}\frac{28}{43}\frac{4}{7}\frac{1}{4} & = & \frac{4}{67} \\
\pi(1,1) & = & \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,1) & = & \frac{43}{67}\frac{28}{43}\frac{2}{7} & = & \frac{8}{67} \\
\pi(1,2) & = & \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,0)\upsilon_{\Delta_1}(1,2) & = & \frac{43}{67}\frac{28}{43}\frac{1}{7} & = & \frac{4}{67}
\end{array}$$

$$\begin{aligned}
\pi(2,1) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,1) &&= \frac{43}{67}\frac{6}{43} &&= \frac{6}{67}\\
\pi(2,2) &= \upsilon_{\Delta_3}(3,0)\upsilon_{\Delta_2}(2,2) &&= \frac{43}{67}\frac{9}{43} &&= \frac{9}{67}\\
\pi(3,1) &= \upsilon_{\Delta_3}(3,1) &&= \frac{142}{1407}\\
\pi(3,2) &= \upsilon_{\Delta_3}(3,2) &&= \frac{104}{1407}\\
\pi(3,3) &= \upsilon_{\Delta_3}(3,3) &&= \frac{248}{1407}
\end{aligned}$$

To illustrate that $X(t)$ can be successively lumpable with respect to different partitions Figure 2.6 shows Markov Chain $X(t)$ with partition $\mathcal{D}$. It is easy to see that $X(t)$ is also successively lumpable with respect to this partition.

Further in Figure 2.7 the original partition $\mathcal{D}$ is shown, but with transition matrix $\underline{\underline{P}}_{\max}$. In this matrix the following holds for all m, i, m', j': $p_{\max}(m', i' \mid m, j) = 0$ if and only if $X(t)$ is not be successively lumpable with respect to partition $\mathcal{D}$ when $p_{\max}(m', i' \mid m, j) > 0$. For graphical reasons loops are not displayed, but are all nonzero.

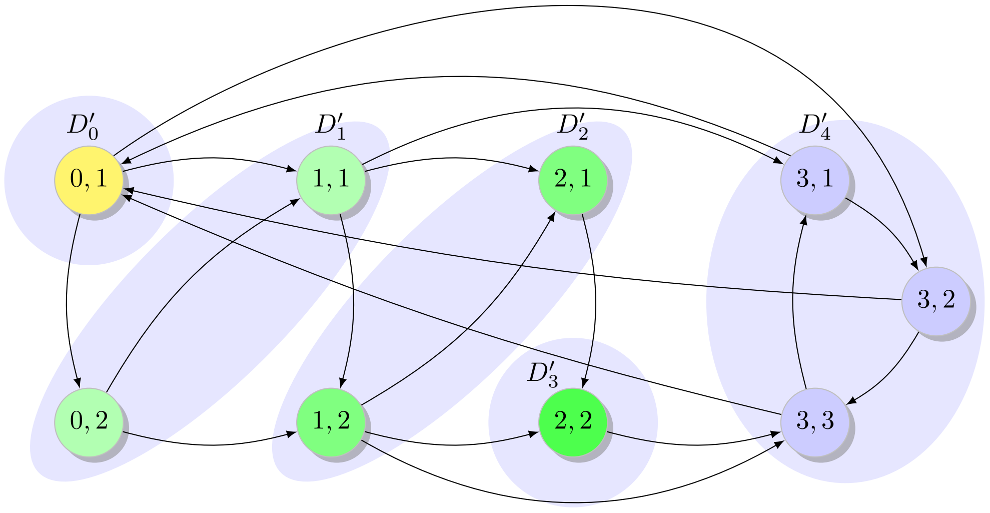


Figure 2.6: Representation of possible transitions under $\underline{\underline{P}}$ with partition $\mathcal{D}' = \{D'_0, D'_1, D'_2, D'_3, D'_4\}$

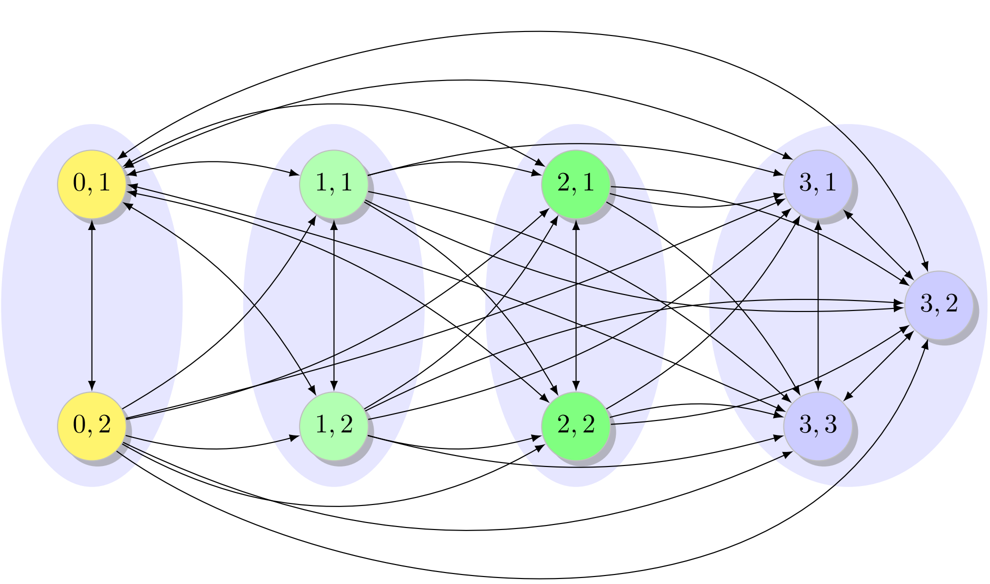


Figure 2.7: Representation of possible transitions under $\underline{\underline{P}}_{\text{max}}$.

2.6 Multiple Successively Lumpable Markov Chains

The main result of the previous Section is that we can calculate the steady state vector in a successively lumpable way, if the Markov Chain is successively lumpable. In this Section we will show that it is possible to have multiple lumpable processes in one Markov Chain and that in this case it is possible to explicitly calculate the steady state vector as well. We will conclude with an example.

Let $X(t)$ be a Markov chain on a finite (or countable) state space $\mathcal{X}$ with transition matrix $\underline{\underline{P}}$. We will assume that the state space $\mathcal{X}$ is composed of N mutually exclusive and exhaustive sets, i.e.,

$$\mathcal{X} = \bigcup_{n=1}^{N} \mathcal{X}^n,$$

where each subset $\mathcal{X}^n$ can be partitioned into a (possibly infinite) sequence of

$$\mathcal{D}^n = \{D_0^n, \ldots, D_{M_n}^n\}.$$

Alternatively the partition

$$\mathcal{D} = \{D_0^1, \ldots, D_{M_1}^1, \ldots, D_0^N, \ldots, D_{M_N}^N\}$$

is a sequence of $N \leqslant \infty$ subpartitions of $\mathcal{X}$. For notational convenience, the elements of each set D_m^n will be relabelled by a triple-notation as $\{(n, m, 1), (n, m, 2), \ldots, (n, m, \ell_{(n,m)})\}$, for given constants $\ell_{(n,m)} \leqslant \infty$. After this state relabeling, the transition matrix of $X(t)$ will be denoted by $\underline{\underline{P}} = [p(n', m', j' \,|\, n, m, j)]$, where the $((n, m, j)\, , (n', m', j'))$ element is given by

$$p(n', m', j' \,|\, n, m, j) = \mathbf{Pr}[X(t+1) = (n', m', j') \,|\, X(t) = (n, m, j)].$$

The definition of an **entrance state** in this triple index notation is as follows.

Definition 2.3. *A subset D^n_m of $\mathcal{D}$ has an **entrance state** $(n, m, \varepsilon(D^n_m)) \in D^n_m$ if and only if*

a) $p(n, m, j \,|\, n', m', j') = 0$, for all $m' \neq m$, $n' \neq n$ and $j \neq \varepsilon(D^n_m)$,

b) when the partition $\mathcal{D}$ has at least two sets, then there exists some $(n', m', j') \in D^{n'}_{m'}$ with $m' \neq m$ or $n' \neq n$ such that $p(n, m, \varepsilon(D^n_m) \,|\, n', m', j') > 0$.

We also define the following:

Definition 2.4. *A Markov Chain $X(t)$ is* multiple successively lumpable *with respect to a partition $\mathcal{D} = \{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$ if and only if*

a) D^n_0 has an entrance state $(n, 0, \varepsilon(D^n_0))$, for all $n = 1, \ldots, N$,

b) $p(n, m', j' \,|\, n, m, j) = 0$, for all (n, m', j'), (n, m, j) with $m' < m$ and $(n, m', j') \neq (n, 0, \varepsilon(D^n_0))$, and

c) $p(n', m', j' \,|\, n, m, j) = 0$, for all (n, m, j) and (n', m', j'), with $n \neq n'$ and $(n', m', j') \neq (n', 0, \varepsilon(D^{n'}_0))$.

The above conditions b) and c) assert that a state in D^n_m can not be entered from a state in $D^{n'}_{m'}$ when $n = n' \,\wedge\, m' > m > 0$ or when $n' \neq n \,\wedge\, m > 0$. Note also that the definition implies that transitions out of states in D^n_m can only lead to states in $D^n_{m'}$ with $m' \geqslant m$ or to the entrance state of $D^{n'}_0$, i.e, to state $(n', 0, \varepsilon(D^{n'}_0))$, for any n'.

We now give the following definition.

Definition 2.5. *Assume that $X(t)$ is successively lumpable. Let $X^n(t)$ be a Markov chain with state space $\mathcal{X}^n$ and transition matrix $\underline{\underline{P}}^n$, where its $(n, m', j' \,|\, n, m, j)$ element is defined as follows:*

a) if $(n, m', j') = (n, 0, \varepsilon(D^n_0))$, then

$$p^n(n, m', j' \,|\, n, m, j) = p(n, m', j' \,|\, n, m, j) + \sum_{(n', m'', j'') \notin \mathcal{X}^n} p(n', m'', j'' \,|\, n, m, j),$$

b) otherwise

$$p^n(n, m', j' \,|\, n, m, j) = p(n, m', j' \,|\, n, m, j).$$

,

Lemma 2.1. *When $X(t)$ is multiple successively lumpable with respect to $\mathcal{D}$, $X^n(t)$, defined above, is successively lumpable with respect to $\mathcal{D}^n$ for all $n \leqslant N$.*

Proof. The transition probabilities $p^n(n, m', j' \,|\, n, m, j)$ can be shown to satisfy the conditions of Definition 2.2 using Definition 2.4 a) and b) and their construction in Definition 2.5. This becomes easier to see when $p^n(n, m', j' \,|\, n, m, j)$ is abbreviated to $p^n(m', j' \,|\, m, j)$, since n is fixed. □

Given a partition $\mathcal{D} = \{D_0^1, \ldots, D_{M_1}^1, \ldots, D_0^N, \ldots, D_{M_N}^N\}$, we define $\Delta_0^n = D_0^n$, $\Delta_1^n = \{(n,1,0)\} \cup D_1^n$, $\Delta_m^n = \{(n,m,0)\} \cup D_m^n$, where states $(n,m,0)$ are lumped states representing $\bigcup_{k=0}^{m-1} D_k^n$. We also define the partition $\mathcal{D}_m^n = \{\Delta_m^n, D_{m+1}, \ldots, D_M\}$ for $m = 0, \ldots, M$. Also for notational consistency we will use the notation: $X_0(t) = X(t)$, $\mathcal{X}_0 = \mathcal{X}$, $\mathcal{D}_0^n = \mathcal{D}^n$, and $\underline{\underline{P}}_0 = \underline{\underline{P}}$. Analogously to the chains $Z_m(t)$ defined in Section 2.3, we define Markov Chains $\underline{\underline{Z}}_m^n(t)$ with state space Δ_m^n and transition matrix $\underline{\underline{U}}_{\Delta_m}^n$. For notational simplicity we will abbreviate its elements $u_{\Delta_m^n}^n(n,m,j \,|\, n,m,i)$ to $u_{\Delta_m^n}^n(j \,|\, i)$. To be precise:

$$u_{\Delta_m^n}^n(j \,|\, i) = \begin{cases} p_m^n(n,m,j \,|\, n,m,i) + \sum\limits_{(n,k,\ell) \notin \Delta_m^n} p^n(n,k,\ell \,|\, n,m,i), & \text{if } (n,m,j) = (n,m,0), \\ \\ p_m(m,j \,|\, m,i), & \text{otherwise.} \end{cases} \tag{2.12}$$

Generalizing the notation of the previous Section, let $\pi(n,m,j) = \lim_{t\to\infty} \mathbf{Pr}[X(t) = (n,m,j)]$, $\pi^n(n,m,j) = \lim_{t\to\infty} \mathbf{Pr}[X^n(t) = (n,m,j)]$, $\upsilon_{\Delta_m^n}^n(j) = \lim_{t\to\infty} \mathbf{Pr}[Z_m^n(t) = (n,m,j)]$ and let $\underline{\pi}$, $\underline{\pi}^n$, $\underline{\upsilon}_{\Delta_m^n}^n$ be the corresponding vectors, with dimensions $\prod_{n=1}^N \prod_{m=0}^{M_n} \ell_m^n$, $\prod_{m=0}^{M_n} \ell_m^n$, $\ell_m^n + \delta(m)$, respectively, where the term "$\delta(m)$" has been added due to the presence, $m > 0$, or absence, $m = 0$, of an artificial state in Δ_m^n .

Finally, we define a process $Y(t)$ with state space $\mathcal{E} = \{1, \ldots, N\}$ and transition matrix $\underline{\underline{Q}}$ with its (n', n) element being equal to:

$$q(n' \,|\, n) = \sum_{(n',m',j') \in \mathcal{X}^{n'}} \; \sum_{(n,m,j) \in \mathcal{X}^n} \pi^n(n,m,j) p(n',m',j' \,|\, n,m,j) \tag{2.13}$$

We will use the notation $\sigma(n)$ for the steady state probabilities of the above process, i.e., $\sigma(n) = \lim_{t\to\infty} \mathbf{Pr}[Y(t) = n]$. Note that the process $Y(t)$ can be viewed as a process between the different "lumped" successively lumpable processes.

We will first show the following:

Lemma 2.2. *Assuming that $X(t)$ is a multiple successively lumpable Markov Chain with $X^n(t)$ defined as above, the following is true:*

$$\pi^n(n,m,j) = \frac{\pi(n,m,j)}{\sum_{(n,m',j') \in \mathcal{X}^n} \pi(n,m',j')}.$$

Proof. It is clear that $\underline{\pi}^n \underline{1}' = 1$. Now from Definition 2.4 c) we see that $\mathcal{X}^n$ has an entrance state $(n, 0, \varepsilon(D_0^n))$ and therefore we can use a similar derivation as is used in Proposition 2.1 to complete the proof. □

Lemma 2.3. *Assuming that $X(t)$ is a multiple successively lumpable Markov Chain with $Y(t)$ defined as above the following is true:*

$$\sigma(n) = \sum_{(n,m,j) \in \mathcal{X}^n} \pi(n,m,j).$$

Proof. It is clear that $\underline{\sigma}\,\underline{1}' = 1$. It suffices to prove that σ is the solution of

$$\sigma(n') = \sum_{n=1}^N \sigma(n) q(n'|n) \quad \text{for } n' = 1, 2, \ldots, N.$$

Indeed:

$$
\begin{aligned}
\sum_{n=1}^{N}\sigma(n)q(n'|n) &= \sum_{n=1}^{N}\left(\sum_{(n,m,j)\in\mathcal{X}^n}\pi(n,m,j)\sum_{(n',m',j')\in\mathcal{X}^{n'}}\sum_{(n,m,j)\in\mathcal{X}^n}\pi^n(n,m,j)p(n',m',j'\,|\,n,m,j)\right)\\
&= \sum_{n=1}^{N}\left(\sum_{(n,m,j)\in\mathcal{X}^n}\pi(n,m,j)\sum_{(n',m',j')\in\mathcal{X}^{n'}}\sum_{(n,m,j)\in\mathcal{X}^n}\frac{\pi^n(n,m,j)p(n',m',j'\,|\,n,m,j)}{\sum_{(n,m',j')\in\mathcal{X}^n}\pi(n,m',j')}\right)\\
&= \sum_{n=1}^{N}\sum_{(n',m',j')\in\mathcal{X}^{n'}}\sum_{(n,m,j)\in\mathcal{X}^n}\pi(n,m,j)p(n',m',j'\,|\,n,m,j) \qquad (2.14)\\
&= \sum_{(n',m',j')\in\mathcal{X}^{n'}}\sum_{(n,m,j)\in\mathcal{X}}\pi(n,m,j)p(n',m',j'\,|\,n,m,j) \qquad (2.15)\\
&= \sum_{(n',m',j')\in\mathcal{X}^{n'}}\pi(n',m',j')\\
&= \sigma(n').
\end{aligned}
$$

The second equality above follows from Lemma 2.2. It is clear that the somations in Eqs. (2.14) and (2.15) can be switched since they are independent. □

The main result of this Section is the next theorem.

Theorem 2.2. *The following is true for a multiple successively lumpable $X(t)$ with respect to partition $\mathcal{D}$:*

$$\pi(n,m,j) = \sigma(n)\upsilon^n_{\Delta^n_m}(j)\prod_{k=m+1}^{M}\upsilon^n_{\Delta^n_k}(0) \quad \textit{for all } (n,m,j)\in\mathcal{X}.$$

Proof. Since by lemma 2.1, $X^n(t)$ is a successively lumpable Markov chain with respect to partition $\mathcal{D}^n$ we know by Theorem 2.1 that for all n

$$\pi^n(n,m,j) = \upsilon^n_{\Delta^n_m}(j)\prod_{k=m+1}^{M}\upsilon^n_{\Delta^n_k}(0).$$

The proof is easy to complete using Lemmata 2.2 and 2.3. □

Remark 2.3. When $M_n = 1$ for all n, then Theorem 2.2 has been proved by different methods in [3].

Remark 2.4. For a multiple successively lumpable Markov Chain we can solve $\prod_{n=1}^{N} M_n$ of sizes $\ell^n_{m_n} + \delta(m_n)$ each, instead of one big system of size $\prod_{n=1}^{N}\prod_{m=0}^{M_n}\ell^n_m$. For example, if $N = 10^4$, $M_n = 10^4$ for all n and $l^n_{m_n} = 10^2$ for all n, m, we need to solve 10^8 systems of size 10^2 instead of 1 of size 10^{10}.

2.7 Algorithm and Example

Similar to algorithm SL for a successively lumpable Markov Chain presented in Section 2.4, we state an algorithm for a Markov Chain that is multiple successively lumpable with respect to a partition $\mathcal{D} = \{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$. Again, this algorithm does not require a proof, it is a direct result of Theorem 2.2.

```
Algorithm MSL
        For  $n = 1, \ldots N$
1.1     Construct $X^n$ with Def. 2.5
1.2     Call Algorithm SL and solve $X^n$
        End
  2     Construct $Q$, c.f., Eq. (2.13)
  3     Calculate $\sigma$ with Lemma 2.3
  4     Calculate $\underline{\pi}$ , c.f., Theorem 2.2
```

To clarify the algorithm, Figure 2.8 shows a multiple successively lumpable Markov Chain, with $N = 2, M_1 = 2, M_2 = 2, \ell^1_{1_1} = 2, \ell^2_{1_2} = 3, \ell^1_{2_1} = 2, \ell^2_{2_2} = 3$, and the transition matrix $\underline{\underline{P}}$ is given below.

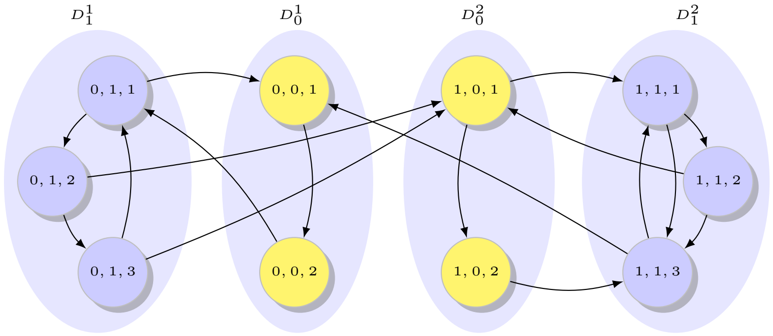


Figure 2.8: Representation of possible transitions under $\underline{\underline{P}}$.

Where $\underline{\underline{P}}$ is as follows.

$$\left[\begin{array}{c|cccccccccc}
 & (0{,}0{,}1) & (0{,}0{,}2) & (0{,}1{,}1) & (0{,}1{,}2) & (0{,}1{,}3) & (1{,}0{,}1) & (1{,}0{,}2) & (1{,}1{,}1) & (1{,}1{,}2) & (1{,}1{,}3) \\
\hline
(0,0,1) & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(0,0,2) & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(0,1,1) & 0.5 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
(0,1,2) & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 0 & 0 & 0 & 0 \\
(0,1,3) & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
(1,0,1) & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.4 & 0 & 0 \\
(1,0,2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
(1,1,1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 \\
(1,1,2) & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 \\
(1,1,3) & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 0
\end{array}\right]$$

After constructing X^1 and X^2, $\underline{\pi}^1$ and $\underline{\pi}^2$ are calculated with Algorithm SL as shown below.

For X^1:

$$\underline{\underline{U}}^1_{D^1_0} = \left[\begin{array}{c|cc} & (1,0,1) & (1,0,2) \\ \hline (1,0,1) & 0 & 1 \\ (1,0,2) & 1 & 0 \end{array}\right]$$

$$\underline{\upsilon}^1_{D^1_0} = [1/2, 1/2]$$

$$\underline{\underline{U}}^1_{\Delta^1_1} = \left[\begin{array}{c|cccc} & (1,0,0) & (1,1,1) & (1,1,2) & (1,1,3) \\ \hline (1,0,0) & 0.5 & 0.5 & 0 & 0 \\ (1,1,1) & 0.5 & 0 & 0.5 & 0 \\ (1,1,2) & 0.8 & 0 & 0 & 0.2 \\ (1,1,3) & 1 & 0 & 0 & 0 \end{array}\right]$$

$$\underline{\upsilon}^1_{\Delta^1_1} = [20/36, 10/36, 5/36, 1/36].$$

And now:

$$\underline{\pi}^1 = [10/36, 10/36, 10/36, 5/36, 1/36].$$

For X^2 :

$$\underline{\underline{U}}^2_{D^2_0} = \left[\begin{array}{c|cc} & (2,0,1) & (2,0,2) \\ \hline (2,0,1) & 0.4 & 0.6 \\ (2,0,2) & 1 & 0 \end{array}\right]$$

$$\underline{\upsilon}^2_{D^2_0} = [5/8, 3/8]$$

$$\underline{\underline{U}}^2_{\Delta^2_1} = \left[\begin{array}{c|cccc} & (2,0,0) & (2,1,1) & (2,1,2) & (2,1,3) \\ \hline (2,0,0) & 0.375 & 0.25 & 0 & 0.375 \\ (2,1,1) & 0 & 0 & 0.2 & 0.8 \\ (2,1,2) & 0.5 & 0.5 & 0 & 0 \\ (2,1,3) & 0.2 & 0.8 & 0 & 0 \end{array}\right]$$

$$\underline{\upsilon}^2_{\Delta^2_1} = [104/583, 220/583, 44/583, 215/583].$$

And thus:

$$\underline{\pi}^2 = [65/583, 39/583, 220/583, 44/583, 215/583].$$

Now, step 2 of Algorithm MSL gives Q:

$$\underline{\underline{Q}} = \left[\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 31/36 & 5/36 \\ 1 & 43/583 & 540/583 \end{array}\right].$$

Solving gives:

$$\sigma = [1548/4463, 2915/4463].$$

And finally step 5 of MSL gives $\underline{\pi}$:

$$\begin{aligned}\underline{\pi} = [&430/4463, 430/4463, 430/4463, 215/4463, 43/4463,\\ &325/4463, 195/4463, 1100/4463, 220/4463, 1075/4463].\end{aligned}$$

Chapter 3

Application on Inventory Management

3.1 Model Description

The basic *order quantity - reorder point* model studied herein was first introduced by [6] and it is described in [7] (pages 181-194) as follows. The system under consideration consists of a single installation that uses a transactions based inventory management ("reporting") system for a single item. There are no interactions between other items that the system may handle.

In each time interval there can be either an arrival of a single customer with probability α or no customer arrival with probability $1-\alpha$. This type of arrival process is the discrete time analog of a Poisson arrival process. The transaction system monitors in discrete time the inventory on hand and an order of size Q is placed whenever the inventory on hand reaches the reorder point r with Q and r integers.

In our model we will assume that orders are fulfilled after a random nonnegative procurement lead time τ. As far as we know this model has not been studied before. The lead time τ takes the values $0, 1, \ldots, T$ with known probabilities $f(t) = \mathbf{Pr}[\tau = t]$, $\sum_{t=0}^{T} f(t) = 1$. We also assume that the value of the lead time becomes known and is announced to the waiting customers as soon as an order is placed. Thus, each arriving customer that finds the system with no inventory left is told the value of the remaining waiting time (remaining lead time) j and upon this information he either chooses to wait with probability $s(j)$, or he leaves the system with probability $1 - s(j)$. Often $s(j)$ a decreasing function of j: the probability that customers will wait increases if the remaining lead time decreases. The model including a mixture of lost sales and backorders has been studied in [9], with a fixed probability of lost sales ($s(j) = s$ for all j), a positive r and a fixed lead time. When r is negative, an arriving customer may find the system with negative inventory level greater than r, in which case he is informed that no order is pending, but he cannot be served. It will be assumed that such a customer decides upon arrival to either stay in the system until an order arrives (not directly placed) or to leave the system with respective probabilities $s(0)$ and $1 - s(0)$.

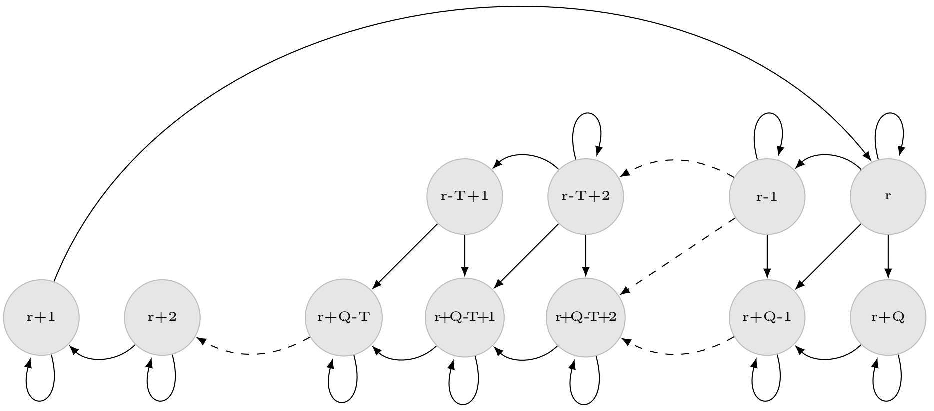
To avoid pathological cases, we assume that there is at most one outstanding order. Because the arrivals occur according to a Bernoulli process this assumption implies that $T < Q$, otherwise there is a positive chance that there are more than Q arrivals of customers during the lead time.

Further, it is easy to see that if $r < -Q$ it is not possible to ever have positive inventory and if $r > T$ the inventory is never negative.

If an order is outstanding with inventory j such that $j + Q \leqslant 0$, we assume that $s(j) = s(0)$. This is a logical assumption since arriving customers will not be served with the next order to arrive, and have to wait an *unknown* amount of time for the next order to be placed. Therefore it makes sense to tread them as customers that arrive with an empty system without any order outstanding.

Let c_h, c_p and c_o denote respectively the holding cost per inventory unit, the penalty costs per backordered unit and the costs per replenishment order. Further, let c_L denote the cost per lost sale, i.e. the cost (missed income) for each customer that leaves the system if there is no inventory on hand.

To obtain a discrete time Markov Chain model for the state evolution of the system, we first make the following assumptions on the order of events and the way costs are incurred.
Formally, let $X(t)$ denote the state of the system at time t. It would be neat and easy to work with if we could only use the inventory level as indicator for the different states. Then $X(t)$ would have the form displayed in Figure 3.1.



However, it is not directly possible to construct a correct Markov Chain on this state space, since the remaining lead time influences the customer decision, and therefore needs to be incorporated in the state space description.

Therefore, at any point in time the system state can be summarized by the inventory level *and* the remaining lead time. A negative inventory level represents the number of waiting customers. The remaining lead time is taken to be zero if there is no order outstanding.

At any point in time the following events occur. a) There is either an arrival or no arrival of a customer. b) If there is a customer arrival the inventory decreases by 1 or increases by $Q - 1$, depending on whether there is an arrival of an outstanding order. If there is no customer arrival the inventory stays the same or increases by Q, depending on whether there is an arrival of an outstanding order. c) When the inventory level becomes equal to r immediately after a customer arrival, an order is placed and the lead time value becomes known for this order. If the lead time is zero, the placed order arrives immediately and the inventory increases with Q. d) Inventory holding cost is computed using this final inventory. Also, at any point in time the remaining lead time decreases by 1 if it is positive (if an order is outstanding).

Including the remaining lead time in the state space to distinguish the states. So under the above assumptions $X(t)$ is a Markov chain on state space $\mathcal{X}$ defined below, where state (i, j) represent a state with i inventory on hand an j time periods remaining lead time.

$$\mathcal{X} = \{(i, 0) \ : \ i = r+1, \ldots, r+Q\} \cup \bigcup_{j=1}^{T} \{(i, j) \ : \ i = r-T+j, \ldots, r\}$$

The transition matrix of $X(t)$ will be denoted by $\underline{\underline{P}} = [p(i', j' \,|\, i, j)]$, where its $((i, j), (i', j'))$-element is defined as follows, for all $(i, j) \in \mathcal{X}$.

For notational convenience we introduce $\beta(i, j)$ for state (i, j) :

$$\beta(i, j) = \begin{cases} \alpha & \text{if } i > 0, \\ s(j)\alpha & \text{if } i \leqslant 0. \end{cases}$$

a) If $j = 0$, then

$$\begin{aligned} p(i-1, 0 \,|\, i, 0) &= \beta(i, 0) && \text{for } r+1 < i \leqslant r+Q, \\ p(i, 0 \,|\, i, 0) &= 1 - \beta(i, 0) && \text{for } r+1 \leqslant i \leqslant r+Q, \\ p(r, j \,|\, r+1, 0) &= \beta(r+1, 0) f(j) && \text{for } j = 0, \ldots, T, \\ p(i', j' \,|\, i, 0) &= 0 && \text{otherwise.} \end{aligned}$$

b) If $j = 1$, then

$$\begin{aligned} p(i+Q-1, 0 \,|\, i, 1) &= \beta(i, 1), \\ p(i+Q, 0 \,|\, i, 1) &= 1 - \beta(i, 1), \\ p(i', j' \,|\, i, 1) &= 0 && \text{otherwise.} \end{aligned}$$

b) If $j > 1$, then

$$\begin{aligned} p(i-1, j-1 \,|\, i, j) &= \beta(i, j), \\ p(i, j-1 \,|\, i, j) &= 1 - \beta(i, j), \\ p(i', j' \,|\, i, j) &= 0 && \text{otherwise.} \end{aligned}$$

Figure 3.1 show a graphical representation of this model. The gray background means that an order is outstanding in states with this background. For graphical simplicity we do not label all arrows with their associated transition probabilities. Loops and arrows downwards leaving state (i, j) have probability $1 - \beta(i, j)$, left and diagonal arrows leaving state (i, j) have probability $\beta(i, j)$.

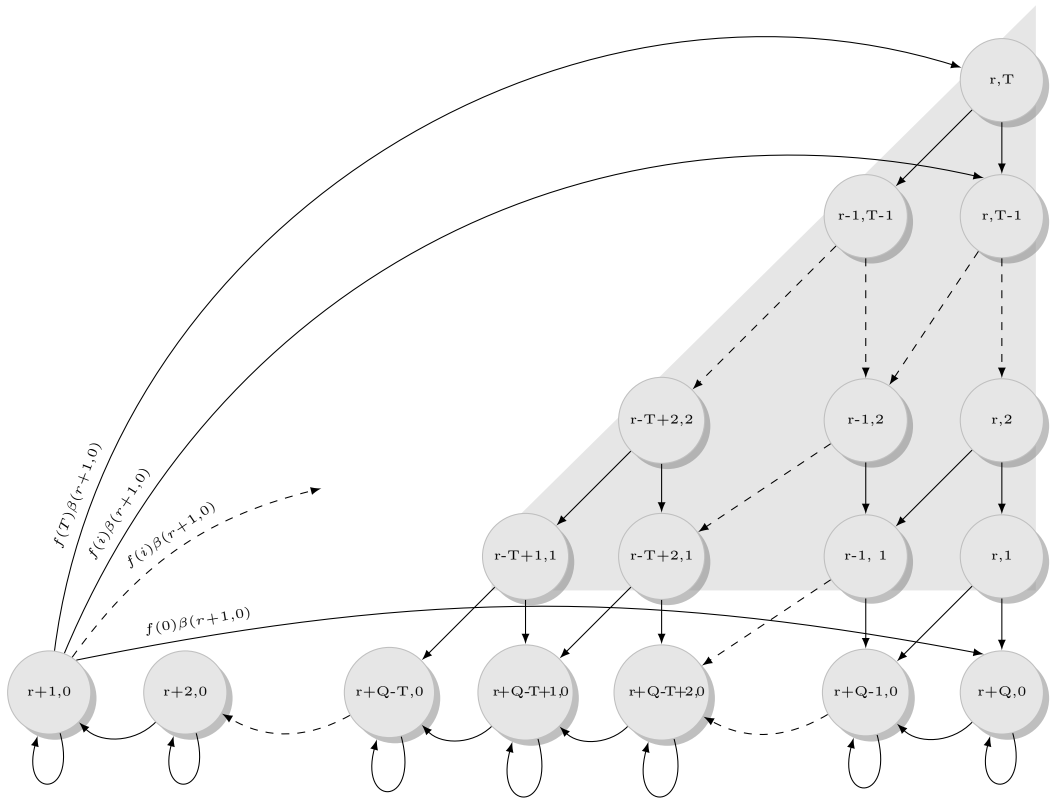


Figure 3.1: General representation of the inventory model

3.2 Approach of the Model with Successive Lumping

Our first objective is to calculate the steady state probabilities of the previously defined Markov Chain. We define partition $\mathcal{D} = \{D_0, \dots D_{T+1}\}$ of $\mathcal{X}$ as follows: $D_0 = \{(r+1,0)\}$, $D_i = \{(r+1-i,1)\dots,(r+1-i,T+1-i),(r+Q+1-i,0)\}$ for $i = 1,\dots,T$ and $D_{T+1} = \{(r+2,0),\dots,(r+Q-T)\}$. In this model there is a natural two dimensional state description and to avoid unnecessary complications we will not relabel the states (i.e., we do not need to refer to the elements of set D_m as (m,i).)

We can now state and prove the following lemma.

Lemma 3.1. *$X(t)$ is successively lumpable with respect to $\mathcal{D}$.*

Proof. It is clear that D_0 has an entrance state, since it contains only one state. Also, from the above we have that $p(i',j'\,|\,i,j) = 0$ if $(i',j') \in D_{m'}$ and $(i,j) \in D_m$, for $m > m' > 0$. Therefore both conditions of the definition of successively lumpable are satisfied. □

We will refer to the lumped state in Δ_i as $(r+i,-1)$. In the notation of the previous chapter this lumped state would be denoted by $(i-1,0)$, but as we already use this for the state with inventory level $i-1$ and remaining lead time 0, we switch to $(r+i,-1)$.

3.3 Calculations

We distinguish 5 intervals for possible values of r and Q, graphically shown in Figure 3.2. The different areas differ by the choice of r for given Q. The structure of the transition probabilities is the same on an area in the parameter space. Hence we study the following possibilities:

1. $r \leqslant -Q$ (inventory can never be positive);
2. $-Q < r < T-Q$ (inventory can cycle through negative states);
3. $T-Q \leqslant r < 0$ (inventory will pass a nonnegative state in every possible cycle);
4. $0 \leqslant r < T-1$ (inventory can become negative during lead time);
5. $T-1 \leqslant r$ (inventory is never negative).

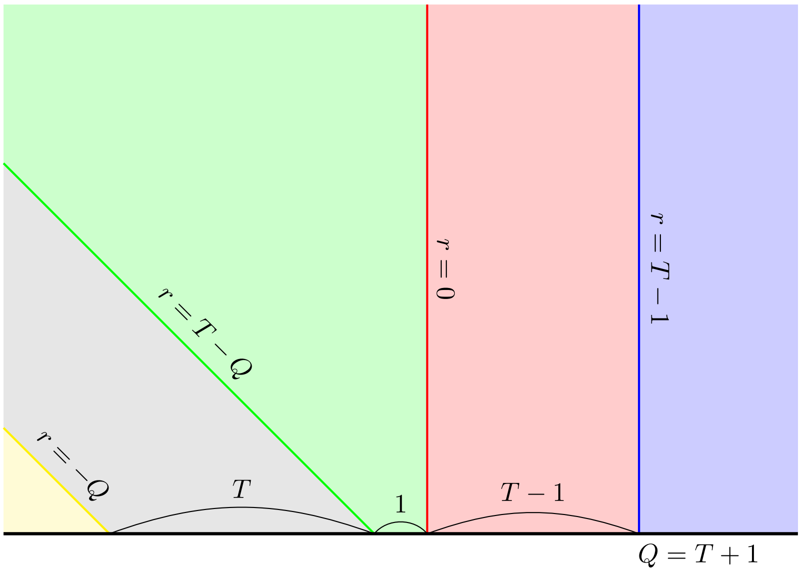


Figure 3.2: The 5 different areas.

For every area costs are calculated in a different way. Using the successive lumping method introduced before, calculating steady states probabilities can be done in an iterative way. We want to find expressions for holding costs C_h, penalty costs C_p, order costs C_o and lost sales costs C_l.

First we look at $\Delta_1 = \{(r+1,0),(r,0),\ldots,(r,T),(r+Q,0)\}$. This corresponding process $X_1(T)$ is graphically shown in Figure 3.3.

Solving gives:

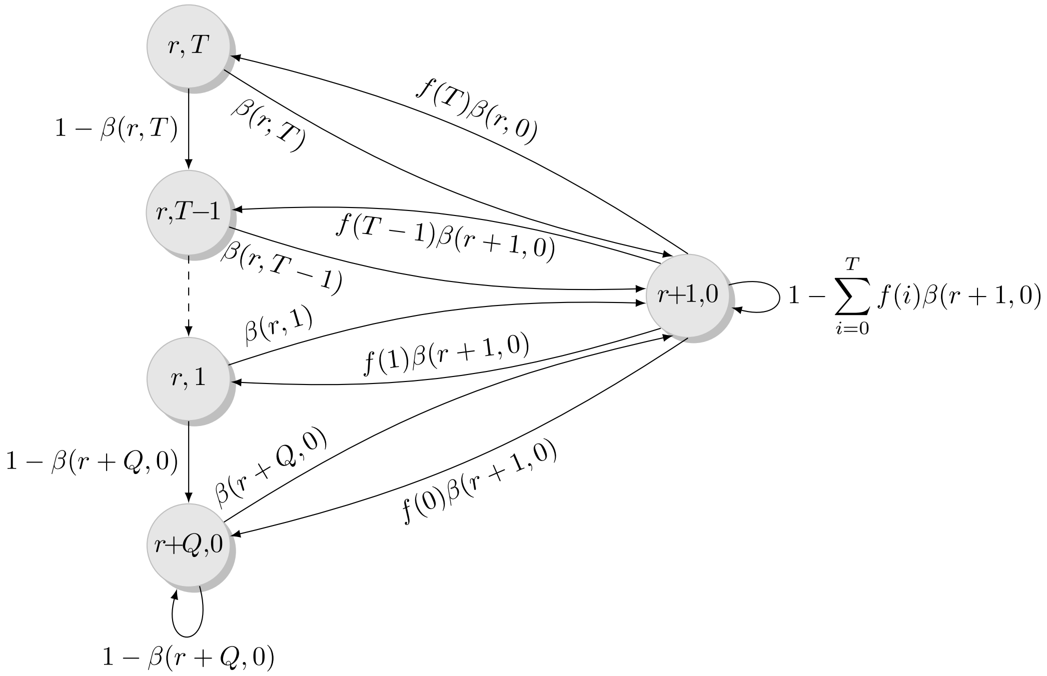


Figure 3.3: $\underline{\underline{\mathrm{U}}}_{\Delta_1}$

$$\upsilon_{\Delta_1}(r,T) = f(T)\beta(r+1,0)\upsilon_{\Delta_1}(r+1,0)$$

$$\upsilon_{\Delta_1}(r,k) = \sum_{i=k}^{T}\left(f(i)\prod_{j=k+1}^{i}(1-\beta(r,j))\right)\beta(r+1,0)\upsilon_{\Delta_1}(r+1,0) \tag{3.1}$$

$$\begin{aligned}\upsilon_{\Delta_1}(r+Q,0) &= \sum_{i=0}^{T}\left(f(i)\prod_{j=1}^{i}(1-\beta(r,j))\right)\beta(r+1,0)\upsilon_{\Delta_1}(r+1,0)\\ &\quad+(1-\beta(r+Q,0))\upsilon_{\Delta_1}(r+Q,0)\\ &= \sum_{i=0}^{T}\left(f(i)\prod_{j=1}^{i}(1-\beta(r,j))\right)\frac{\beta(r+1,0)}{\beta(r+Q,0)}\upsilon_{\Delta_1}(r+1,0)\ .\end{aligned} \tag{3.2}$$

The calculation of $\underline{\upsilon}_{\Delta_m}$ follows similar lines, i.e., the process has the same structure, only $f(i)\beta(r+1,0)$ needs to be replaced by $\upsilon_{\Delta_{m-1}}(r-m+1,i+1)\beta(r-m+1,i+1)$ and the transition leaving $(r+Q-m,0)$ to $(r+Q-m+1,0)$ needs to be taken in account. So:

$$\begin{aligned}&\upsilon_{\Delta_m}(r-m+1,k) =\\ &\sum_{i=k}^{T-m+1}\left(\upsilon_{\Delta_{m-1}}(r-m+2,i+1)\beta(r-m+2,i+1)\prod_{j=k+1}^{i}(1-\beta(r-m+1,j))\right)\upsilon_{\Delta_m}(r-m+2,-1)\end{aligned}$$

and

$$\begin{aligned}
&\upsilon_{\Delta_m}(r+Q-m+1,0) = \\
&\left(\sum_{i=0}^{T-m+1}\left(\upsilon_{\Delta_{m-1}}(r-m+2,i+1)\beta(r-m+2,i+1)\prod_{j=1}^{i}(1-\beta(r-m+1,j))\right)\right. \\
&\qquad \left. + \upsilon_{\Delta_{m-1}}(r+Q-m+2,0)\beta(r+Q-m+2,0)\right)\frac{\upsilon_{\Delta_m}(r-m+2,-1)}{\beta(r+Q-m+1,0)}.
\end{aligned}$$

In the sequel we use shorthand notations:

$$\pi(r-m) = \sum_{i=1}^{T-m}\pi(r-m,i) \qquad \text{for } m = 1,\dots,T \tag{3.3}$$

$$\pi(i) = \pi(i,0) \qquad \text{for } i = r+1,\dots,r+Q. \tag{3.4}$$

3.3.1 $T-1 \leqslant r$

In this case $\beta(i,j)=\alpha$ for every state (i,j). Furthermore, since there are never backorders it is clear that $C_p = C_l = 0$.
Equations (3.1) and (3.2), now simplify to:

$$\begin{aligned}
\upsilon_{\Delta_1}(r,k) &= \sum_{i=k}^{T}\left(f(i)(1-\alpha)^{i-k}\right)\alpha\upsilon_{\Delta_1}(r+1,-1) \\
\upsilon_{\Delta_1}(r+Q,0) &= \sum_{i=0}^{T}\left(f(i)(1-\alpha)^{i}\right)\upsilon_{\Delta_1}(r+1,-1)
\end{aligned}$$

and for $\underline{\upsilon}_{\Delta_m}$:

$$\begin{aligned}
\upsilon_{\Delta_m}(r-m+1,k) &= \sum_{i=k}^{T-m+1}\left(\upsilon_{\Delta_{m-1}}(r-m+2,i+1)\alpha(1-\alpha)^{i-k}\right)\upsilon_{\Delta_m}(r-m+2,-1) \\
\upsilon_{\Delta_m}(r+Q-m+1,0) &= \left(\sum_{i=0}^{T-m+1}\left(\upsilon_{\Delta_{m-1}}(r-m+2,i+1)(1-\alpha)^{i}\right)\right. \\
&\qquad \left. + \upsilon_{\Delta_{m-1}}(r+Q-m+2,0)\right)\upsilon_{\Delta_m}(r-m+2,-1)
\end{aligned}$$

We can now state and prove the following lemma.

Lemma 3.2. *For $m = 1,2,\dots,T$ the following holds:*

$$\upsilon_{\Delta_m} = m/(m+1)$$

Proof. First we proof for $m=1$:

$$\sum_{i=1}^{T}\upsilon_{\Delta_1}(r,i) + \upsilon_{\Delta_1}(r+Q,0) = \upsilon_{\Delta_1}(r+1,-1)\ .$$

So $\upsilon_{\Delta_1}(r+1,-1) = 1/2$. Now, suppose the lemma is true for $m-1$. Then for m:

$$\sum_{i=1}^{T-m+1} \upsilon_{\Delta_m}(r-m+1,i) + \upsilon_{\Delta_m}(r+Q-m+1,0) =$$

$$\left(\sum_{i=1}^{T-m+2} \upsilon_{\Delta_{m-1}}(r-m+2,i) + \upsilon_{\Delta_{m-1}}(r+Q-m+2,0)\right) \upsilon_{\Delta_m}(r-m+2,-1) =$$

$$\frac{\upsilon_{\Delta_m}(r-m+2,-1)}{m}$$

thus $\upsilon_{\Delta_m}(r-m+2,-1) = m/(m+1)$. □

Further, in $\upsilon_{\Delta_{T+2}}$, it is clear that $\upsilon_{\Delta_{T+2}}(r+i,0) = \upsilon_{\Delta_{T+2}}(r+i+1,0)$ for $i = 2, \dots Q-T-2$ and

$\upsilon_{\Delta_{T+2}}(r+Q-T-1,0) = \upsilon_{\Delta_{T+1}}(r+Q-T,0)\upsilon_{\Delta_{T+2}}(r+Q-T,0) = \upsilon_{\Delta_{T+2}}(r+Q-T,0)/(T+2)$.

Thus $\upsilon_{\Delta_{T+2}}(r+Q-T,0) = (T+2)/Q$ and $\upsilon_{\Delta_{T+2}}(r+i,0) = 1/Q$ for $i = 2, \dots Q-T-1$.

Now using the main result of a successively lumpable Markov Chain, and the abbreviation of equations (3.3) and (3.4) we get that for $k = 0, \dots T-1$

$$\pi(r-k) + \pi(r+Q-k) = \frac{1}{Q}\ .$$

and for $k = 1, \dots Q-T$:

$$\pi(r+k) = \frac{1}{Q}\ .$$

$\pi(r-k)$ and $\pi(r+Q-k)$ separately can be calculated with the successively lumpable result, but these expressions are not very neat. Therefore we will just refer to these probabilities as $\pi(r-k)$ and $\pi(r+Q-k)$.

The cost calculation will be as follows.

$$\begin{aligned}
C_h &= c_h\left(\sum_{i=r-T+1}^{r+Q} i\pi(i)\right) \\
&= c_h\left(\sum_{i=r+1}^{r+Q} i/Q - Q\sum_{i=r-T+1}^{r} \pi(i)\right) \\
&= c_h\left(\frac{Q(Q+2r+1)}{2Q} - Q\sum_{i=r-T+1}^{r} \pi(i)\right) \\
&= c_h\left(\frac{Q+2r+1}{2} - \sum_{i=r-T+1}^{r} \pi(i)\right) \\
C_o &= c_o\alpha\pi(r+1) \\
&= \frac{c_o\alpha}{Q}
\end{aligned}$$

It is immediately clear that increasing r will lead to higher costs. Therefore, the minimum value in this area is at the boundary, where $r = T-1$.

3.3.2 $r \leqslant -Q$

In this case, the inventory will never reach a positive value. Therefore $C_h = 0$. Further, $s(i) = s(0)$ for all i by definition, since $i + Q \leqslant 0$ for all i.
The derivation of the model is the same as when $T - 1 \leqslant r$, only α is substituted by $s(0)\alpha$. Further, we have lost sales. The probability of a lost sale in state i is:$(1 - s(0))$.
We calculate C_o, C_p and C_l in the same way as in previous:

$$
\begin{aligned}
C_p &= c_p \left(\sum_{i=r-T+1}^{r+Q} -i\pi(i) \right) \qquad\qquad = c_p \left(\frac{-(Q+2r+1)}{2} + Q \sum_{i=r-T+1}^{r} \pi(i) \right) \\
C_l &= c_l(1 - s(0))\alpha \sum_{i=r-T+1}^{r+Q} \pi(i) \\
&= c_l(1 - s(0))\alpha \\
C_o &= \frac{c_o s(0)\alpha}{Q}
\end{aligned}
$$

Again, it is clear that now *decreasing* r will increase the costs, thus the minimum value is at the boundary where $r = Q$.

3.3.3 $-Q < r < T - Q$ or $T - Q \leqslant r < 0$ or $0 \leqslant r < T - 1$

Calculations for these three areas follow similar lines. There is a mixture of penalty costs and holding costs. Since the transitions differ from α to $\alpha s(0)$ we do not get the term $1/Q$. However, the successively lumpable procedure still gives a (fast) way to calculate $\pi(i)$ for all i.

$$
\begin{aligned}
C_h &= c_h \sum_{i=1}^{r+Q} i\pi(i) \\
C_p &= c_p \sum_{i=r-T+1}^{-1} -i\pi(i) \\
C_l &= c_l \left(\sum_{i=r-T+1}^{\min(r,0)} \sum_{j=1}^{i+T-r} (1 - s(j))\alpha\pi(i,j) + \sum_{i=r+1}^{0} (1 - s(0))\pi(i) \right) \\
C_o &= c_o \beta(r+1, 0)\pi(r+1)
\end{aligned}
$$

The distinction between the different areas is important for the following.
When $-Q \leqslant r < T - Q$ it is clear by using the successively lumpable method that

$$
\begin{aligned}
\pi(i) &= \pi(r+1) && \text{for } i = r+2, \ldots, r+Q-T \text{ and} \\
\pi(i) + \pi(i+Q) &= \pi(r+1) && \text{for } i = r-T+1, \ldots -Q.
\end{aligned}
$$

Similar for $T - Q \leqslant r < 0$:

$$
\pi(i) = \pi(r+1) \qquad \text{for } i = r+2, \ldots 0
$$

and for $0 \leqslant r < T - 1$:

$$
\begin{aligned}
\pi(i) &= \pi(r+1) && \text{for } i = 2, \ldots, r+Q-T \text{ and} \\
\pi(i) + \pi(i+Q) &= \pi(r+1) && \text{for } i = 1, \ldots r.
\end{aligned}
$$

3.4 Unimodality of the costfunction

The previous expressions have been used the cost as a function of Q and r. However, it is still an open problem whether unimodality is true for the cost function over Q and r. Matlab graphs show that this is very likely, for example shown in Figure 3.4 with the parameter choices given in Table 3.4. Also, $r = -10, \ldots, 9$ and $Q = 6, \ldots, 25$ and we assume that $s(i) = s$ for all i.

T	c_o	c_h	c_p	c_l	α	s
5	100	1	5	10	0.8	0.3

Table 3.1: Model parameters.

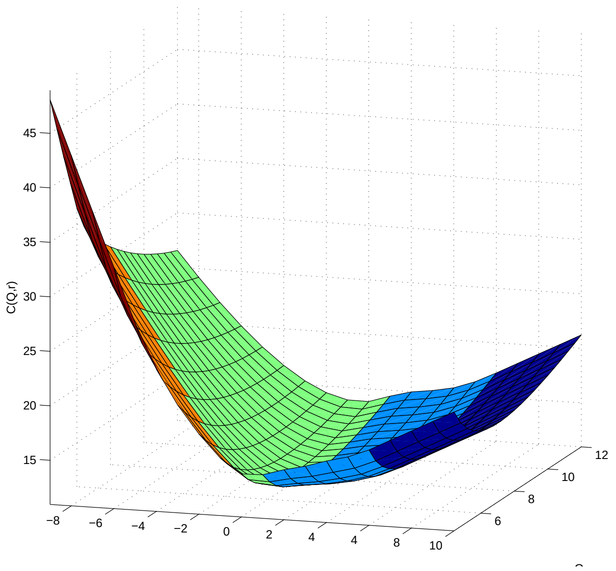


Figure 3.4: The costfunction over Q and r.

Chapter 4

Quantity Discounts

4.1 Introduction

This Chapter treats the classical order quantity, reorder point (Q, r) continuous review stochastic inventory model with Poisson arrivals and a fixed lead time under quantity discount pricing, c.f., [7]. Procedures for both all-units and incremental quantity discount schedules are provided. The term all-units refers to the discount schedule where a discount is given for all ordered products, once you order more than a given breakpoint, multiple breakpoints are possible. Incremental refers to the discount schedule where a discount is only given for the number of products ordered above a certain ordering level.

For the case of a single ordering price, an efficient algorithm for this problem has been described in [2]. This article generalizes this previous work by constructing algorithms for both all-units and incremental quantity discount schedules. The quantity discounts studied herein are discussed in the context of the "EOQ" model, in [7], pp. 62 to 68.

For other literature related to inventory management with quantity discounts, we refer to [15], [16] and [14]. For a review of this general area we refer to [1] and [5].

This chapter is organized as follows. In Section 4.2 we review the work of [2] and [7] and give some of the proofs, others are given in appendix A in Section 4.6. Section 4.3 contains the main results of this paper regarding algorithms and complexity. Further, in this section it is shown that a computational improvement of the order of $O(r^*)$ is possible when one uses a normal approximation to the Poisson distribution. Section 4.4 contains the algorithms and Section 4.5 provides computational and graphical illustrations.

4.2 Assumptions, Notation and Preliminaries

The basic *order quantity* - *reorder point* model studied herein was first introduced by [6] and it is described in [7] (pages 181-194) as follows. A system under consideration consists of a single installation that uses a transactions based inventory management ("reporting") system for a single item. There are no interactions between other items that the system may handle.

Let $N_{\tau\, t}$ denote the number of arrivals (which is the cumulative demand in the time interval

$(t-\tau,t]$, arrivals occur according to a Poisson Process. We denote: $p_k = P(N_{\tau\, t} = k) = e^{-\lambda\tau}\,(\lambda\tau)^k/k!$ and $P_j = \sum_{k=0}^{j} p_k = P(N_{\tau\, t} \leqslant j)$. Further, there is a fixed, positive, procurement lead time τ. A stochastic lead time was discussed in the previous Chapters, without quantity discounts. The transaction system monitors continuously the inventory position, defined as the inventory on hand plus the quantity on order - if any. The policy employed is determined by two integer values Q (the order quantity) and r (the reorder level) such that an order of size $Q > 0$ is placed when the inventory position reaches the reorder level r. In this model the demand variable is discrete, the order quantity Q, the reorder point r, and all the inventory levels will also be treated as discrete.

Let c_h and c_p denote respectively the holding cost per inventory unit and the penalty costs per backordered unit per time unit. Let $c_{\tilde{p}}$, c_K denote respectively the fixed single costs per backordered unit and the costs per replenishment order.

Finally let c_i denote the unit ordering cost (unit price) per product when the quantity ordered is in the i^{th} price interval: $[b_i, b_{i+1})$, for $i = 1,\ldots,M$, where M is the fixed number of pricing intervals, with $b_1 = 0$ and $b_{M+1} = \infty$.

The time unit is taken to be a year and the optimal values of Q and r are those that minimize the annual expected cost. To avoid trivialities: c_h, c_K and c_i are assumed to be positive as well as that at least one of the costs: c_p, $c_{\tilde{p}}$ is positive. Further, without loss of generality we will assume that

$$c_i > c_{i+1} \text{ for all } i = 0,1,\ldots,M.$$

Also, for notational convenience we will rename

$$c_i = c_i\lambda,\ c_K = c_K\lambda \text{ and } c_{\tilde{p}} = c_{\tilde{p}}\lambda. \tag{4.1}$$

Definitions.
a) A sequence of real numbers $\{d_1, d_2, \ldots\}$ is said to be *unimodal* if there exists a finite index k^* called the *mode* such d_j *decreases* as j increases up to $j = k^*$ and then increases from then on, i.e., $d_1 \geqslant d_2 \geqslant \ldots \geqslant d_{k^*} < d_{k^*+1} \leqslant d_{k^*+2} \leqslant \ldots$

b) We call a real function g on the integers *unimodal* if there exists a minimizing point x^* such that $\ldots \geqslant g(x^*-1) \geqslant g(x^*) < g(x^*+1) \leqslant g(x^*+2) \leqslant \ldots$.

c) For a umimodal function g we define the $x-mode$ to be the point x^g:

$$x^g = \min\{x\ :\ g(x-1) \geqslant g(x) < g(x+1)\}.$$

d) For a fixed unimodal function g and for any $k \geqslant 1$ define the set of points $\mathcal{L}_k^g = \{x_1,\ldots,x_k\}$ as follows:

$$\begin{aligned}
x_1 &= x^g,\\
x_2 &= argmin_x\{g(x),\ x \notin \{x_1\}\},\\
\vdots &\quad \vdots\\
x_k &= argmin_x\{g(x),\ x \notin \{x_1,\ldots,x_{k-1}\}\}.
\end{aligned}$$

Remark 4.1. From the definition of unimodality it follows easily that each set $\mathcal{L}_k^g$ contains k *adjacent* points, by expanding the set to the left or to the right in each iteration.

Theorem 4.1 and Lemma 4.1 below are due to [2] (see also, [16]). Theorem 4.1 provides an expression of the annual cost $C(Q,r)$, without quantity discounts, in terms of a unimodal function G. Further, it is shown that for this G, its $x-mode$ point x^G exists (i.e. the "min" is well defined above) and it is the maximal minimizing point of G. Lemma 4.1 provides a simple characterization for $r^*(Q)$ that minimizes $C(Q,r)$, with respect to r for a fixed $Q>0$. Also, Lemma 4.1 gives the minimum value $C(Q^*,r^*(Q^*))$ of $C(Q,r)$, with respect to Q and r, in terms of G. Further it provides an implicit solution for Q^* (and thus $r^*=r^*(Q^*)$) in terms of a condition on G and the function of Q defined as follows: $C^*(Q):=\min_r C(Q,r)$. A proof of Theorem 4.1 is included in the appendix of this Chapter. We repeat both Theorem 4.1 and Lemma 4.1 for clarification and the proves are partly along different lines.

Theorem 4.1. *The following are true:*
a)

$$C(Q,r)=c_K/Q+\sum_{x=r+1}^{r+Q} G(x)/Q, \tag{4.2}$$

where,

$$G(x)=(c_h+c_p)\sum_{i=0}^{x-1} P_i + c_p(\lambda\tau - x) + c_{\tilde{p}}(1-P_{x-1}).. \tag{4.3}$$

b) The function $G(x)$ is unimodal and its $x-mode$ point x^G is its maximal minimizing point.

Remark 4.2. If $c_p>0$, simple algebra implies: $G(x+1)-G(x)=-c_p<0$ (for all $x<0$.) It then follows that x^G is the unique minimizing point of $G(x)$.

In the sequel x^G will always refer to the $x-mode$ of G, and $\mathcal{L}_Q^G\{x_1,\dots,x_Q\}$ are the sets constructed as in Definition d) above for G.

Lemma 4.1. *a) For any fixed positive integer Q, $C(Q,r)$, is minimized with respect to r at $r^*=\min\mathcal{L}_Q^G-1$, and $C^*(Q):=\min_r\ C(Q,r)$ is given by:*

$$C^*(Q)=\frac{c_K+\sum_{i=1}^{Q} G(x_i)}{Q}. \tag{4.4}$$

b) The value $Q^*=\min\{Q:\ G(x_{Q+1})<C^*(Q)\}$ is the optimal order quantity i.e., $\min_{Q,r}\{C(Q,r\}=C(Q^*,r^*(Q^*)))$.

Proof. For a) note that the unimodality of G and Remark 4.1 imply that $\mathcal{L}_Q^G$ contains Q adjacent integers and $G(x_1),\dots,G(x_Q)$ are the smallest values of $G(x)$. Then $\sum_{i=1}^{Q} G(x_i)$ is the summation of the Q smallest values of $G(x)$, corresponding to $C(Q,r)$ attaining a minimum at $r=r^*=\min\mathcal{L}_Q^G-1$.

For part b) we first show that the following inequalities are equivalent.

$$C^*(Q+1)\leqslant C^*(Q) \tag{4.5}$$

$$G(x_{Q+1})<C^*(Q). \tag{4.6}$$

The proof of the equivalence is by noticing that $C^*(Q+1)-C^*(Q)=(QC^*(Q)+G(x_{Q+1})/(Q+1)-C^*(Q)=(G(x_{Q+1})-C^*(Q)/(Q+1))$, and the equivalence follows directly.

Now, note that if $Q > Q^*$ we have:

$$\begin{aligned} C^*(Q) - C^*(Q^*) &= \frac{c_K + \sum_{i=1}^{Q} G(x_i)}{Q} - C^*(Q^*) \\ &= \frac{c_K + \sum_{i=1}^{Q^*} G(x_i)}{Q} + \frac{\sum_{i=Q^*+1}^{Q} G(x_i)}{Q} - C^*(Q^*) \\ &= \frac{1}{Q}\left(Q^* C^*(Q^*) + \sum_{i=Q^*+1}^{Q} G(x_i) - QC^*(Q^*) \right) \\ &\geqslant \frac{Q - Q^*}{Q} \left(G(x_{Q^*+1}) - C^*(Q^*) \right) \\ &\geqslant 0, \end{aligned}$$

where the first inequality above follows using $\sum_{i=Q^*+1}^{Q} G(x_i) \geqslant \sum_{i=Q^*+1}^{Q} G(x_{Q^*+1}) = (Q - Q^*)G(x_{Q^*+1})$. □

Remark 4.3. Note that Q^* is *not* the x-mode: Q^* is the size of set $\mathcal{L}_{Q^*}^G$) and has the property that adding an extra point will lead to a higher value of $C^*(Q^* + 1)$.

Now, using the previous lemma, a process for finding (Q^*, r^*) works as follows: first find x^G, by comparing $G(x + 1)$ with $G(x)$, starting at $x = 0$, and stop when $G(x + 1) > G(x)$.

After finding $x_1 = x^G$, we continue to the second stage with initializing $Q = 1$, $C^*(Q) = c_K + G(x^G)$ and also $\mathcal{L}_1^G = x^G$.

Next, we compare $G(x_{Q+1})$ with $C^*(Q)$, with x_{Q+1} defined as in the previous. If $G(x_{Q+1}) > C^*(Q)$ we stop, and $C^* = C^*(Q)$, $Q^* = Q$ and $r^* = \min(\mathcal{L}_Q^G) - 1$.
If not, $C^*(Q + 1) = (QC^*(Q) + G(x_{Q+1}))/(Q + 1)$, $Q = Q + 1$, and we repeat the process.

4.3 Quantity Discounts

In this Section we consider the case when the unit ordering cost (unit price) depends on the quantity ordered Q and it is c_i per unit of the product when Q is in the i^{th} price interval: $[b_i, b_{i+1})$, for $i = 1, \dots, M$, where $M + 1 \geqslant 2$, is a fixed number of pricing intervals, with $b_1 = 0$ and $b_{M+1} = \infty$. Both all-units and incremental discounts are considered. Note that in the previous Section, the cost function $C(Q, r)$ did not contain unit ordering costs, since they were all equal and therefore they did not influence the values of Q^* and r^*.

Let $C_D(Q, r)$ denote the expected annual cost function, including the average unit price. If $D = A$, $C_D(Q, r)$ will refer to the all units case and if $D = I$ it will refer to the incremental case. For $C_D(Q, r)$ we state the following lemma (with $G(x)$ as in the previous Section):

Lemma 4.2.

$$C_D(Q, r) = \frac{(c_K + R_{i(Q)}) + \sum_{x=r+1}^{r+Q} G(x)}{Q} + c_{i(Q)}, \tag{4.7}$$

where $i(Q)$ is the unique i for which $Q \in [b_i, b_{i+1})$. and

$$R_i = \begin{cases} \sum_{j=1}^{i} b_j (c_{j-1} - c_j), & \text{if D=I,} \\ 0, & \text{if D=A.} \end{cases}$$

Proof. Let $c_D^{av}(Q)$ be the average unit ordering cost, for $D = A, I$. Then:

$$\begin{aligned}
c_A^{av}((Q) &= (Qc_i(Q))/Q \\
&= c_{i(Q)}, \text{ for the all-units case.} && (4.8)\\
c_I^{av}(Q) &= \left(c_{i(Q)}(Q - b_{i(Q)}) + c_{i(Q)-1}(b_{i(Q)} - b_{i(Q)-1}) + \ldots + c_0(b_1 - 0)\right)/Q \\
&= \left(Qc_{i(Q)} + b_{i(Q)}(c_{i(Q)-1} - c_{i(Q)}) + \ldots + b_1(c_0 - c_1)\right)/Q \\
&= \sum_{j=1}^{i(Q)} b_j(c_{j-1} - c_j)/Q + c_{i(Q)}, \text{ for the incremental case.} && (4.9)
\end{aligned}$$

Furthermore, $C_D(Q, r)$ is by definition:

$$C_D(Q, r) = c_K\frac{1}{Q} + c_h\sum_{x=0}^{r+Q} x\pi_x - c_p\sum_{x=-\infty}^{0} x\pi_x + c_{\tilde{p}}\sum_{x=-\infty}^{0}\pi_x + c_D^{av}(Q)$$

The proof can be now completed using Lemma 4.9 and Eq. (4.8), (4.9). □

4.3.1 The All Units Discount Case

In this case we have:

$$C_A(Q, r) = C(Q, r) + c_{i(Q)},$$

where $C(Q, r)$ is the cost function of Section 2.
Furthermore, define

$$C_A^*(i) = \min_{Q,r}\{C_A(Q, r) : \ Q \in [b_i, b_{i+1})\},$$

and

$$C_A^* = \min_i \ C_A^*(i).$$

In Section 2 unimodality of G was used for creating a stopping criterion for finding Q^* that minimizes $C^*(Q) = \min_r C(Q, r)$, which was in turn used to minimize $C(Q, r)$.
However in the present case the unimodality of G does not suffice directly to construct stopping criteria for an algorithm. This is achieved by the following two lemmata, Lemma 4.3 describes a useful property of $C^*(Q)$.

Lemma 4.3. *For any integers Q' and Q'' with $Q'' > Q' > Q^*$ the following holds: $C^*(Q'') > C^*(Q')$.*

Proof. First we will look at the difference between $C^*(Q'')$ and $C^*(Q')$. By the same arguments as in Lemma 4.1 we get that:

$$C^*(Q'') - C^*(Q') \geqslant \frac{Q'' - Q'}{Q''}\left(G(x_{Q'+1}) - C^*(Q')\right). \tag{4.10}$$

Now, we will prove the lemma by contradiction. If $C^*(Q'') < C^*(Q')$, then by (4.10) we get that $G(x_{Q'+1}) < C^*(Q')$. If this is true, then:

$$G(x_{Q'+1}) < \frac{c_K + \sum_{i=0}^{Q'} G(x_i)}{Q'}.$$

It is clear that this can only hold if $c_K = Q'G(x_{Q'+1}) - \sum_{i=0}^{Q'} G(x_i) + \alpha$, for some $\alpha > 0$.
We look at $C^*(Q^*)$. In the first inequality we use that $G(x_{Q'+1}) \geqslant G(x_i)$ for all $i \leqslant Q'$.

$$\begin{aligned}
C^*(Q^*) &= \frac{Q'G(x_{Q'+1}) - \sum_{i=0}^{Q'} G(x_i) + \alpha + \sum_{i=0}^{Q^*} G(x_i)}{Q^*} \\
&= \frac{Q^*G(x_{Q'+1}) + \alpha + (Q' - Q^*)G(x_{Q'+1}) - \sum_{i=Q^*+1}^{Q'} G(x_i)}{Q^*} \\
&\geqslant G(x_{Q'+1}) + \frac{\alpha}{Q^*} \\
&> G(x_{Q'+1}) + \frac{\alpha}{Q'} \\
&> C^*(Q').
\end{aligned}$$

The above implies $C^*(Q') < C^*(Q^*)$ and this is a contradiction to the definition of Q^*. Thus, $G(x_{Q'+1}) > C^*(Q')$ and by (4.10) we have $C^*(Q'') > C^*(Q')$. □

Recall, $C_A(Q, r)$ equals $C(Q, r) + c_{i(Q)}$ if we get all-units discounts. The results from Lemma 4.1 (b) and Lemma 4.3 will form the basis for the next lemma, with Q^* the previously defined optimal order quantity of $C(Q, r)$ and $r^*(Q) = \mathcal{L}_Q^G - 1$.

Lemma 4.4. *For the expected all-units discount cost function $C_A(Q, r)$, the following hold:*

a) $C_A^*(i) > C_A^*(i(Q^*))$ for all $i < i(Q^*)$,

b) $C_A^*(i) = C_A(b_i, r^*(b_i))$ for all $i > i(Q^*)$.

Proof. Note that for every $Q < b_{i(Q^*)}$ by definition $c_{i(Q^*)} < c_{i(Q)}$. This implies that, for every $Q < Q^*$:

$$\begin{aligned}
C_A(Q, r) &= C(Q, r) + c_{i(Q)}, \\
&> C(Q, r) + c_{i(Q^*)} \\
&\geqslant C^*(Q^*) + c_{i(Q^*)}. \\
&\geqslant C_A^*(i(Q^*)).
\end{aligned}$$

The proof of part a) is complete by noting that the validity of the above inequality for every $Q < Q^*$ implies its validity for $C_A^*(i)$, with $i < i(Q^*)$.
For the proof of part b) note that Lemma 4.3 implies:

$$\begin{aligned}
C_A(b_i, r^*(b_i)) &= C^*(b_i) + c_i \\
&\leqslant C^*(Q) + c_i \text{ for all } Q \in [b_i, b_{i+1}).
\end{aligned}$$

□

Corollary 4.1. *In the all units discount case the following are true,*

$$\begin{aligned}
C_A^* &= min_{i>i(Q^*)}\{C^* + c_{i(Q^*)},\ C^*(b_i) + c_i\}. \\
Q_A^* &= argmin_Q\ (C_A(Q, r_A^*(Q)),\ with\ Q \in \{Q^*, b_i,\ for\ i > i(Q^*)\}). \\
r_A^* &= r_A(Q_A^*).
\end{aligned}$$

Proof. It follows immediately from Lemma 4.4. □

Corollary 4.1 suggests a procedure for determining the minimal value of C_A^*:

We start with finding C^*, Q^* and r^*, according to the process described for the case without quantity discounts. Then $C_A^*(i(Q^*)) = C^* + c_{i(Q^*)}$. Next, determine $r^*(b_{i(Q^*)+j})$ and compute $C_A^*(i(Q^*)+j) = C^*(b_{i(Q)+j}) + c_{i(Q^*)+j}$ for $1 \leqslant j \leqslant M$. Compute the minimum of all these values. The overall minimum is C_A^*, with (Q_A^*, r_A^*) the corresponding order quantity and reorder level.

4.3.2 The Incremental Discount Case

In this case we have:

$$C_I(Q,r) \quad = \quad C_{i(Q)}(Q,r) + c_{i(Q)}, \tag{4.11}$$

where $C_{i(Q)}(Q,r)$ is the general cost function for the case without quantity discounts, where, instead of c_K, we use $c_K + R_{i(Q)}$ as the order costs. As before $i(Q)$ is the unique i for which $Q \in [b_i, b_{i+1})$ and

$$R_i = \sum_{j=1}^{i} b_j(c_{j-1} - c_j).$$

Again, the unimodality of G does not suffice to construct stopping criteria for an algorithm to determine the minimal costs. As in the all units discounts case, we will derive a different procedure for finding (Q_I^*, r_I^*) to minimize $C_I(Q,r)$.

Towards this end we introduce the functions:

$$\widehat{C}_I(Q,r,i) = C_i(Q,r) + c_i, \tag{4.12}$$

where $C_i(Q,r)$ is the general cost function for the case without quantity discounts, where, instead of c_K, we use $c_K + R_i$ as the order costs, *independent* of Q. The idea behind these functions, is that these functions can be treated as the general cost function for the case without quantity discounts plus a constant. In this way the results for the case without quantity discounts can be used directly.
We also define the following:

$$\widehat{C}_I^*(i) \quad = \quad \min_{Q,r}\{\widehat{C}_I(Q,r,i)\}.$$

A minimizing point for the unconstrained problem $\widehat{C}_I^*(i) = \min_{Q,r}\{\widehat{C}_I(Q,r,i)\}$ will be denoted by (Q_i^*, r_i^*). This minimizing point will be called *achievable* if $Q_i^* \in [b_i, b_{i+1})$.

Let $\mathcal{A}$ be the set of i such that (Q_i^*, r_i^*) is achievable. For such an achievable point we have by definition $\widehat{C}_I^*(Q_i^*, r_i^*, i) = C_I(Q_i^*, r_i^*)$.

Lemma 4.5. *For the "cost" function $\widehat{C}_I(Q,r,i)$ there is a price domain $[b_{i_o}, b_{i_o+1})$ for which $i_o \in \mathcal{A}$.*

Proof. Note that the assumption $0 \leqslant c_{i+1} < c_i$, for all $i < M$ implies that the set up "costs" $c_K + R_i$ are increasing in $i : R_{i+1} > R_i$. Because $G(x)$ is the same for $C_i(Q, r)$ and $C_{i+1}(Q, r)$ we have:

$$C^*_{i+1}(Q_{i+1})) = \frac{C^*_i(Q) + (R_{i+1} - R_i)}{Q}.$$

Then it is easy to see that:

$$Q^*_{i+1} \geqslant Q^*_i. \tag{4.13}$$

Now, suppose that $i \notin \mathcal{A}$ for every i. Because Q^*_i is increasing over i and the intervals are positioned in an increasing way as well, there are four cases possible if the assumption is true. It is easy to see that the cases below are the only possible cases. We will show that each case leads to a contradiction.

Case 1: $\exists i_1, i_2$, with $i_1 < i_2$, such that $Q^*_i > b_{i+1}$, for all $i \leqslant i_1$ and $Q^*_i \leqslant b_i$, for all $i \geqslant i_2$.

In this case, there is a j with $i_1 \leqslant j < i_2$ for which $Q^*_{j-1} > b_j$ and $Q^*_{j+1} \leqslant b_{j+1}$, by the increasing property of Q^*_i. This means that Q^*_j is achievable, a contradiction to our assumption.

Case 2: $\exists j$ *such that* $Q^*_i < b_i, \forall i \leqslant j$ *and* $Q^*_{i+1} \geqslant b_{i+1}, \forall i > j$.

Recall that $b_1 = 0$ and $b_{M+1} = \infty$. In this case the only solution can be a $Q^*_0 < 0$ However, for all i Theorem 4.1 implies $0 < Q^*_i < \infty$ which is a contradiction.

Case 3: For all $i : \; Q^*_i < b_i$.

A contradiction can be obtained as in case 2.

Case 4: For all $i : \; Q^*_i \geqslant b_{i+1}$.

In this case $Q^*_M > \infty$ a contradiction.

All the cases lead to a (trivial) contradiction and therefore there is an index i_o for which $i_o \in \mathcal{A}$, i.e. $\mathcal{A}$ is not the empty set. □

Now we know, $\widehat{C}^*_I(Q^*_i, r^*_i, i) = C_I(Q^*_i, r^*_i)$ for some i, next we will show that $C^*_I = C_I(Q^*_i, r^*_i)$ for some i.

Lemma 4.6. *The solution to* $C^*_I = \min_{Q,r}\{C_I(Q, r)\}$, *is the minimum of* $\widehat{C}^*_I(i)$ *for which* $i \in \mathcal{A}$.

Proof. In the this proof we will denote price interval $[b_i, b_{i+1})$ by price interval i.
Let Q^*_I be the optimal order quantity. As said before, Q^*_I can not be $b_1 = 0$. If Q^*_I is not Q^*_i for all $i \in \mathcal{A}$, then using the unimodality of $C_i(Q, r)$ this is the global minimum of $C^*_i(Q)$. Q^*_I occurs at a boundary point b_i for some i or at $b_i - 1$ for some i.

If $Q = b_i$ $(i > 0)$ is a local minimum of $C_I(Q, r^*(Q))$ then $C_I(Q, r^*(Q))$ is increasing for Q in the entire price interval i and strictly decreasing for Q in price interval $i - 1$. This is clear from Lemma 4.3.

Then by Theorem 4.1 we get that $Q^*_i < b_i$ and $Q^*_{i-1} \geqslant b_i$, and therefore $Q^*_i < Q^*_{i-1}$. which is not possible, by virtue of Lemma 4.5. So $Q = b_i$ can not be the global minimum of $C_I(Q, r^*(Q))$, and thus $Q^* \in (b_i, b_{i+1})$ for some b_i and then C^*_I is the minimum of $\widehat{C}^*_I(i)$ for which $i \in \mathcal{A}$. A similar argument shows that $Q = b_i - 1$ can not be the global minimum if Q is not the minimum of $C^*_i(Q, r^*(Q))$. □

Corollary 4.2. *In the incremental case, C_I^* is the minimum of all achievable solutions of $\widehat{C}_I^*(i)$ which exists for at least one i.*

Proof. It is a direct consequence of Lemmata 4.5 and 4.6. □

Now, a process of computing (Q_I^*, r_I^*) works by finding all the (Q_i^*, r_i^*) (the minimizers of $C_i(Q, r)$, found as in the no discount case), checking if they are achievable and comparing the corresponding $C_I(Q_i^*, r_i^*)$. The minimum of these values is C_I^*.

4.3.3 Efficient computation of the $x - mode$ of G when $c_{\tilde{p}} = 0$

Next we point out that when $c_{\tilde{p}} = 0$, the computation of the $x - mode$ point x^G, c.f., Definition (c), can be done more efficiently using the observation of the lemma below, where P^{-1} is the inverse cumulative distribution function of the Poisson distribution P.

Lemma 4.7. *If $c_{\tilde{p}} = 0$ then $x^G = P^{-1}(c_p/(c_p + c_h))$.*

Proof. Since $c_{\tilde{p}} = 0$, simple algebra using Eq. (4.3) shows that

$$G(x+1) - G(x) = (c_h + c_p)P_x - c_p. \tag{4.14}$$

The above and the definition of x^G imply the following:

$$\begin{aligned} P_{x^G-1} &\leqslant c_p/(c_h + c_p), \\ P_{x^G} &> c_p/(c_h + c_p). \end{aligned}$$

The proof is easy to complete. □

Note that in the above proof the argument for the difference $G(x+1) - G(x)$ is analogous to the one used in the context of the newsvendor model, c.f. [7], p. 297.

Further, one can use the Normal approximation to P_{x^G} to obtain the easily computable expression for $x^G = P^{-1}(c_p/(c_p + c_h)) \approx F^{-1}(c_p/(c_p + c_h))$, where F is the normal cumulative distribution with mean $\mu = \lambda\tau$ and variance $\sigma^2 = \lambda\tau$, c.f., [4].

Indeed, as Table 4.1 displays the exact value of x^G and the corresponding value of $F^{-1}(c_p/(c_p + c_h)$, we see that in all cases when $F^{-1}(c_p/(c_p + c_h)$ is rounded to its closest integer we obtain the exact value for x^G. In Table 4.1 below ρ denotes the fraction $c_p/(c_p + c_h)$.

	$\lambda\tau = 10$		$\lambda\tau = 50$		$\lambda\tau = 100$		$\lambda\tau = 250$	
ρ	x^G	$F^{-1}(\rho)$	x^G	$F^{-1}(\rho)$	x^G	$F^{-1}(\rho))$	x^G	$F^{-1}(\rho)$
0.1	6	5.9	41	40.9	87	87.2	230	229.7
0.3	8	8.3	46	46.3	95	94.8	242	241.7
0.5	10	10.0	50	50.0	100	100.0	250	250.0
0.7	12	11.7	54	53.7	105	105.2	258	258.3
0.9	14	14.1	59	59.1	113	112.8	270	270.3

Table 4.1: Display of x^G with corresponding values of $F^{-1}(c_p/(c_p + c_h))$.

4.4 Algorithms

The procedures described in the previous Section can be presented as statement algorithms. The NoDiscounts-algorithm is meant for the computation of a global optimum when there are no quantity discounts. The AllUnits and Incremental algorithms work according to the procedures described in Corollary 4.1 and 4.2.

AllUnits first calls NoDiscounts to find the value of C^*, then finds the corresponding interval and adds $G(x_1-1)$ or $G(x_Q+1)$ until $\mathcal{L}_Q^G$ contains b_i points, so $Q=b_i$. Then it compares the current lowest value with the corresponding costs. This is done for all b_i, with $i>i(Q^*)$.

Incremental calls NoDiscounts once for each price interval with the corresponding value R_i, $i=0,\ldots,M$ and it computes the minimum of $\widehat{C}_I(i)$. It checks whether $i\in\mathcal{A}$, and if $i\in\mathcal{A}$ it compares C^*+c_i with the up to then minimum value of C^*+c_j for $j\in\mathcal{A}$ and $j\leqslant i-1$ defined in the algorithm as C_I. When $i=M$, $C_I^*=C_I$.
For notational convenience, all parameters in the algorithms are not defined explicitly as inputs but we assume they are global. Also, the results returned by NoDiscounts are used with the same notation in AllUnits and Incremental. Finally computations for $G(x)$ are done by using Eq. (4.15) below.

$$\begin{aligned} G(x) &= (c_h+c_p)\sum_{i=0}^{x-1}P_i + c_p(\lambda\tau - x) + c_{\tilde{p}}(1-P_{x-1}) \\ &= (c_h+c_p)(xP_{x-1}-\lambda\tau P_{x-2}) + c_p(\lambda\tau - x) + c_{\tilde{p}}(1-P_{x-1}). \end{aligned} \tag{4.15}$$

```
NoDiscounts($K$)
    If $c_{\tilde{p}} > 0$
        $i = 0$
        While $G(i+1) < G(i)$
            $i = i + 1$
        End
        $x_1 = i$
    Else
        $x_1 = F^{-1}(c_p/(c_h+c_p))$
        $G(x_1) = (c_h+c_p)(x_1 P_{x_1-1} - \lambda\tau P_{x_1-2}) + c_p(\lambda\tau - x_1)$
    End
    $SumG = G(x_1)$; $C = K + G(x_1)$; $Q = 1$; $\underline{r} = x_1 - 1$; $\overline{r} = x_1 + 1$
    While $C > G(\underline{r})$ and $C > G(\overline{r})$
        If $G(\underline{r}) < G(\overline{r})$
            $SumG = SumG + G(\underline{r})$; $\underline{r} = \underline{r} - 1$
        Else
            $SumG = SumG + G(\overline{r})$; $\overline{r} = \overline{r} + 1$
        End
        $Q = Q + 1$; $C = (K + SumG)/Q$
    End
    $C^* = C$; $Q^* = Q$; $r^* = \underline{r}$;
Return $C^*$; $Q^*$; $r^*$
```

AllUnits

```
    Call NoDiscounts($c_K$)
    $i = 0$
    While $b_{i+1} \leqslant Q^*$
        $i = i + 1$
    End
    $C_A = C^* + c_i$; $Q_A = Q^*$; $\underline{r_A} = r^*$; $\overline{r_A} = r^* + Q^* + 1$; $b_i = Q^*$
    For $j = i : M - 1$
        For $k = 1 : b_{j+1} - b_j$
            If  $G(\underline{r_A}) < G(\overline{r_A})$
                    $SumG = SumG + G(\underline{r_A})$; $\underline{r_A} = \underline{r_A} - 1$
            Else
                    $SumG = SumG + G(\overline{r_A})$; $\overline{r_A} = \overline{r_A} + 1$
            End
        End
        If $(c_K + SumG)/b_{j+1} + c_{j+1} < C_A$
            $C_A = (c_K + SumG)/b_{j+1} + c_{j+1}$; $Q_A = b_j$; $r_A = \underline{r_A}$;
        End
    End
    $C_A^* = C_A$; $Q_A^* = Q_A$; $r_A^* = r_A$;
Return $C_A^*$; $Q_A^*$; $r_A^*$
```

```
Incremental
    $C_I = \infty$
    For $i = 0 : M$
        Call NoDiscounts($c_K + R_i$)
        If $b_i \leqslant Q^* < b_{i+1}$ and $C^* + c_i < C_I$
            $C_I = C^* + c_i$; $Q_I = Q^*$; $r_I = r^*$;
        End
    End
    $C_I^* = C_I$; $Q_I^* = Q_I$; $r_I^* = r_I$;
Return $C_I^*$; $Q_I^*$; $r_I^*$
```

In the next theorem we assume that the computation of P_x is can be done in $O(1)$.

Theorem 4.2. *Algorithms* AllUnits *and* Incremental *have complexity $O(Q^* + |r^*| + b_m M)$ and $O((Q^* + |r^*|)M)$ respectively. If $c_{\tilde{p}} = 0$, the complexities are $O(Q^* + b_m M)$ and $O(Q^* M)$.*

Proof. The complexity of NoDiscounts is $O(Q^* + |r^*|)$: it consists of two while loops, where every loop has length $O(1)$. The first while-loop runs in at most $|r^*| + i$, iterations, with $i < Q^*$ (since $x^G \leqslant |r^*| + Q^*$), and the second loop in Q^* steps. Together both loops take at most $Q^* + |r^*| + i$ iterations with complexity $O(Q^* + |r^*|)$.

AllUnits runs NoDiscounts once and afterwards a loop of b_i iterations and a double loop with at most $(M - b_i)b_m$ iterations. So AllUnits has complexity $O(Q^* + |r^*| + b_m M)$. Furthermore, AllUnits is correct since Corollary 4.1.

Incremental calls NoDiscounts and makes 2 single computations each loops. Therefore, NoDiscounts has complexity $O((Q^* + r^*)M)$. □

Remark 4.4. a) If $c_{\tilde{p}} = 0$, one can use the Normal Approximation to replace the first loop of NoDiscounts, see proof of theorem 4.2 above, by a single computation. Then, the complexities of the AllUnits and the Incremental algorithms become respectively $O(Q^* + b_m M)$ and $O(Q^* M)$.

b) It is preferable to express the complexity in terms of exogenous parameters c_h, c_p, $c_{\tilde{p}}$, M, b, c, λ, τ. However, in this model this is not possible, because there are too many parameters and they have correlated effects.

4.5 Computations

To test the algorithms and demonstrate some of the issues discussed above, some simple computations have been done, both for all-units and incremental discounts. The data used for both cases is summarized in Table 4.2 below.

λ	τ	c_h	c_p	c_K	$c_{\tilde{p}}$	M
1	15	2	5	100	5	3

Table 4.2: Parameter values

Tables 4.3 and 4.4 below summarize the results. Note that an entry of the form: "-" means that the corresponding value need not to be computed because it cannot be optimal (for all-units) or it is not achievable (for incremental). The boldface entry is the corresponding minimal value of C_A^* and C_I^* respectively. The values of $\vec{c}$ are chosen differently for the all-units and the incremental discounts cases for emphasizing different issues regarding both cases.

In Table 4.3, C^* is computed, and the lower bounds of the above price intervals are compared since these values can be optimal as well, exactly as the algorithm has described. The minimal value can be any value of these.

	$\vec{b} = [0, 10, 20, 30]$, $\vec{c} = [10, 7, 6, 1.5]$			
τ	$(C_A^*(0), Q_0^*, r_0^*)$	$(C_A^*(1), Q_1^*, r_1^*)$	$(C_A^*(2), Q_2^*, r_2^*)$	$(C_A^*(3), Q_3^*, r_3^*)$
5	-	**(25.00, 12, 0)**	(25.95, 20, -2)	(26.70, 30, -5)
15	-	(27.63, 14, 11)	**(27.84, 20, 9)**	(27.98, 30, 6)
25	-	(29.58, 15, 21)	(29.43, 20, 19)	**(28.63, 30, 16)**
	$\vec{b} = [0, 20, 40, 50]$, $\vec{c} = [10, 7, 6, 1.5]$			
5	(28.00, 12, 0)	**(26.95, 20, -2)**	(37.40, 40, -8)	(40.48, 50, -11)
15	(30.63, 14, 11)	**(28.84, 20, 9)**	(38.36, 40, 3)	(40.25, 50, 0)
25	(32.58, 15, 21)	**(30.43, 20, 19)**	(39.24, 40, 13)	(40.95, 50, 10)

Table 4.3: All Units discounts

Table 4.4, for the incremental case, shows that the differences between the optimal values can be very small, even for large discounts, as in this case. However, we still need to compute every value since all the values can be feasible, especially with an uneven distribution of boundary points or discount prices.

	$\vec{b} = [0, 10, 20, 30]$, $\vec{c} = [60, 50, 40, 30]$			
τ	$(\widehat{C}_I^*(0), Q_0^*, r_0^*)$	$(\widehat{C}_I^*(1), Q_1^*, r_1^*)$	$(\widehat{C}_I^*(2), Q_2^*, r_2^*)$	$(\widehat{C}_I^*(3), Q_3^*, r_3^*)$
3	(-, 12, -1)	(74.50, 17, 3)	**(74.23, 24, -4)**	(75.05, 32, -7)
10	(-, 13, 6)	(75.88, 18, 8)	**(75.28, 25, 2)**	(75.81, 32, 0)
15	(-, 14, 11)	(76.77, 19, 9)	**(75.94, 25, 7)**	(76.35, 33, 5)
	$\vec{b} = [0, 20, 40, 50]$ $\vec{c} = [60, 50, 40, 30]$			
3	**(77.71, 12, -1)**	(79.78, 29, -6)	(-, 29, -6)	(-, 43, -10)
10	**(79.52, 13, 6)**	(80.63, 22, 3)	(-, 30, 1)	(-, 43, -3)
15	**(80.63, 14, 11)**	(81.69, 22, 8)	(-, 33, 5)	(-, 44, 2)

Table 4.4: Incremental discounts

Below are graphical representations of both cases. They show C_A and C_I respectively as function of Q and r.

The figures show the characteristic described in the lemmata. In the all-units case, local minima can be found at the “no discounts” minimum in the second interval and on the lower bounds of the third and fourth interval.
In the incremental case, Figure 4.2 shows that the minimum can be either the “achievable” minimum in the second, third or fourth interval.

To emphasize the forms of the surfaces the values of the parameters $\vec{c}$ and $\vec{b}$ are chosen as in Table 4.5 below, all other parameters are chosen as in Table 4.2. It is hard to compare the all-units discounts case and the incremental discounts case. The only thing that is clear when comparing is that when the same parameter values are used, the all-units leads to more discount. Then the discount is for all articles, and therefore the optimal value Q^* is usually higher.

$\vec{c}$	$[30, 20, 10, 0]$ for all-units
$\vec{c}$	$[75, 50, 25, 0]$ for incremental
$\vec{b}$	$[0, 20, 40, 60]$ for both

Table 4.5: Parameter values

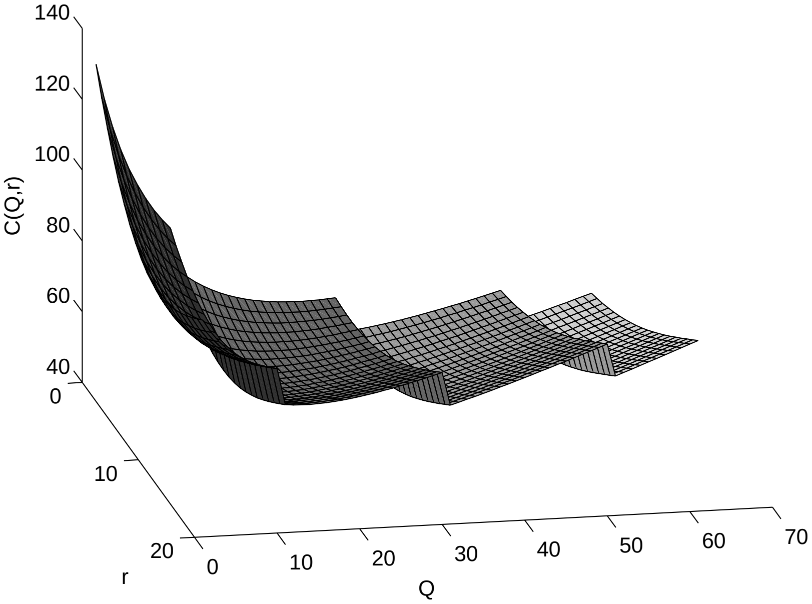


Figure 4.1: The Expected cost function with all units discount pricing.

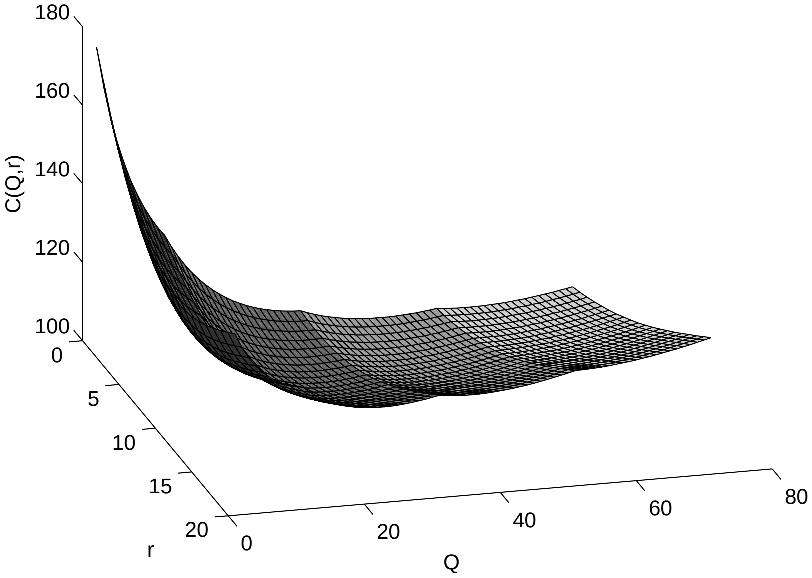


Figure 4.2: The Expected cost function with incremental discount pricing.

4.6 Appendix A

This appendix summarizes some of the previous work in this area. The proofs are sometimes along different lines.

In this model, the *inventory position* (defined as the inventory at hand plus outstanding orders) provides a suitable state description variable. This is not the case with the *inventory level* (defined as inventory at hand or net inventory). Indeed, when there is heavy demand during some cycle resulting in a large number of backorders, then the arrival of outstanding orders might never bring the on hand inventory back up to the reorder point again, and hence another order would never be placed under a (Q, r) system that is based on the inventory at hand. However, when a (Q, r) system is based on the inventory position, the holding costs cannot be computed directly. If during some cycle there is a considerable number of backorders, then a large number of orders will be placed, for the reorder point in terms of the inventory position will be crossed a large number of times. If r is the reorder point in terms of the inventory position, then immediately after an order is placed the inventory position is $Q + r$. Using the Poisson demand arrival assumption we see that the time evolution of the inventory position can be described by a continuous time Markov Chain with state space $\mathcal{S} = \{r + 1, \ldots, r + Q\}$ and transition diagram given by Figure 4.3 below.

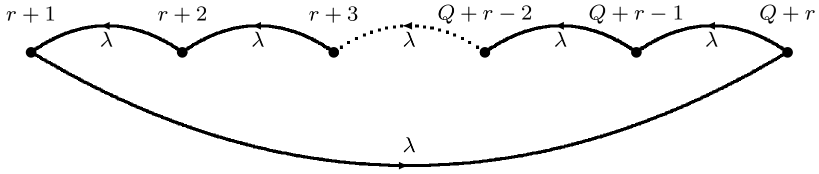


Figure 4.3: inventory position

Since all rates of this Markov chain are equal it follows that the steady state probabilities of the inventory position, $\pi_{ip}(x) = lim_{t\to\infty} P(X(t) = x) = 1/Q$, for all $x = r + 1, \ldots, r + Q$, i.e., in equilibrium the inventory position is uniformly distributed over $r + 1, \ldots, r + Q$.

Repeat from the previous that $N_{\tau\, t}$ denotes the number of arrivals (which is the cumulative demand in the time interval $(t - \tau, t]$ and that $p_k = P(N_{\tau\, t} = k) = e^{-\lambda\tau} (\lambda\tau)^k / k!$. Further, let $P_j = \sum_{k=0}^{j} p_k = P(N_{\tau\, t} \leqslant j)$. Also, let $X(t)$, $Y(t)$ denote respectively the *inventory position* nd the *inventory level* at time t. Note that $X(t) = Y(t) + O_t Q$, all $t \geqslant 0$, where O_t denotes the number of outstanding orders at time t. Since orders placed after $t - \tau$ have not arrived by time t, the following equation holds:

$$Y(t) = Y(t - \tau) + O_{t-\tau} Q - N_{\tau\, t}. \tag{4.16}$$

Note also that $X(t - \tau) = Y(t - \tau) + O_{t-\tau} Q$, hence

$$Y(t) = X(t - \tau) - N_{\tau\, t}. \tag{4.17}$$

From Eq. (4.17) it follows that $Y(t)$ is also a continuous time Markov chain. Its state space is the set $\{\ldots, -2, -1, 0, 1, 2, \ldots, Q + r\}$ and transition diagram given by Figure 4.4 below.

Even though, the transition rates and diagram of the *inventory level* process $Y(t)$ are more complex, Eq. (4.17) above allows the computation of the steady state probabilities $\pi(x) =$

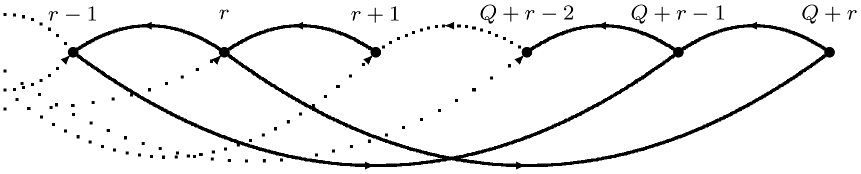


Figure 4.4: inventory level

$\lim_{t\to\infty} P(Y(t) = x)$ in terms of those of the probabilities $\pi_{ip}(x) = 1/Q$ and $P_j = P(N_{\tau\, t} \leqslant j)$ as follows.

Lemma 4.8. *For any integer inventory level x the following are true:*

$$\pi(x) = \begin{cases} (P_{r+Q-x} - P_{r-x})/Q, & \textit{for } -\infty < x \leqslant r, \\ P_{r+Q-x}/Q, & \textit{for } r+1 \leqslant x \leqslant r+Q. \end{cases}$$

Proof. Since it is easy to see that both $X(t)$ and $Y(t)$ Markov chains are ergodic, we can assume that for $t = \infty$, $P(X(t) = k) = \pi_{ip}(k) = 1/Q$ and $P(Y(t) = x) = \pi(x)$. For $-\infty < x \leqslant r$ we have:

$$\begin{aligned} \pi(x) = P(Y(t) = x) &= \sum_{j=r+1}^{Q+r} P(Y(t) = x \,|\, X(t-\tau) = j)\, P(X(t-\tau) = j) \\ &= \sum_{j=r+1}^{Q+r} P(N_{\tau\, t} = j - x \,|\, X(t-\tau) = j) P(X(t-\tau) = j) \\ &= \sum_{j=r+1}^{Q+r} P(N_{\tau\, t} = j - x)\pi_{ip}(j) \\ &= \sum_{k=r+1-x}^{r+Q-x} p_k/Q, \end{aligned}$$

where in the above we have used the independence of $X(t-\tau)$ and $N_{\tau\, t}$ as well as the observation that conditional on $X(t-\tau) = j$, $Y(t) = x$ if and only if $N_{\tau\, t} = j - x$. Similarly, for $r+1 \leqslant x \leqslant r+Q$ we have:

$$\begin{aligned} \pi(x) = P(Y(t) = x) &= \sum_{j=r+1}^{Q+r} P(Y(t) = x \,|\, X(t-\tau) = j)\, PX(t-\tau) = j) \\ &= \sum_{j=x}^{Q+r} P(Y(t) = x \,|\, X(t-\tau) = j)\, PX(t-\tau) = j) \\ &= \sum_{j=x}^{Q+r} P(N_{\tau\, t} = j - x \,|\, X(t-\tau) = j) P(X(t-\tau) = j) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j=x}^{Q+r} p_{j-x}\pi_{ip}(j) \\
&= \sum_{y=0}^{r+Q-x} p_y/Q.
\end{aligned}
$$

where the first equality above follows from the observation that $P(Y(t) = x \,|\, X(t-\tau) = j) = 0$ if $j < x$, since it is not possible for the *inventory level* at time t to be $Y(t) = x \geqslant r+1$, if the *inventory position* at time $t-\tau$ is smaller then x. The other equalities follow as those of part i). □

Next, assuming for the moment for simplicity that there are no quantity discounts, the expected annual cost is a function $C(Q, r)$, can be written as follows.

$$
C(Q,r) = c_K \frac{1}{Q} + c_h \sum_{x=0}^{r+Q} x\pi(x) - c_p \sum_{x=-\infty}^{0} x\pi(x) + c_{\tilde{p}} \sum_{x=-\infty}^{0} \pi(x) \tag{4.18}
$$

One can simplify $C(Q, r)$ using the lemma below.

Lemma 4.9. *The following are true:*

i) $\sum_{x=0}^{r+Q} x\pi(x) = \sum_{x=r+1}^{r+Q} \sum_{i=0}^{x-1} P_i/Q.$

ii) $-\sum_{x=-\infty}^{-1} x\pi(x) = \left(\sum_{x=r+1}^{r+Q} \left(\sum_{i=0}^{x-1} P_i + \lambda\tau - x\right)\right)/Q.$

iii) $\sum_{x=-\infty}^{0} \lambda\pi(x) = \left(\sum_{x=r+1}^{r+Q} \lambda(1 - P_{x-1})\right)/Q.$

Proof. For i) we have:

$$
\begin{aligned}
\sum_{x=0}^{r+Q} x\pi(x) &= \left(\sum_{x=0}^{r} x(P_{r+Q-x} - P_{r-x}) + \sum_{x=r+1}^{r+Q} xP_{r+Q-x}\right)/Q \\
&= \left(\sum_{x=0}^{r+Q} (r+Q-x)P_x - \sum_{x=0}^{r} (r-x)P_x\right)/Q \\
&= \left(\sum_{x=r+1}^{r+Q} (r+Q-x)P_x + Q\sum_{x=0}^{r} P_x + \sum_{x=0}^{r} (r-x)P_x - \sum_{x=0}^{r} (r-x)P_x\right)/Q \\
&= \sum_{x=r+1}^{r+Q} \left(\sum_{i=r+1}^{x-1} P_i + \sum_{i=0}^{r} P_i\right)/Q \\
&= \sum_{x=r+1}^{r+Q} \sum_{i=0}^{x-1} P_i/Q.
\end{aligned}
$$

Similarly, for ii) using the following property of a Poisson distribution, $xp_x = \lambda\tau p_{x-1}$ for any $x = 1, 2, \ldots$ we obtain:

$$\begin{aligned}
-\sum_{x=-\infty}^{0} x\pi_x &= -\sum_{x=-\infty}^{-1} x(P_{r+Q-x} - P_{r-x})/Q \\
&= \sum_{x=1}^{\infty} x((1-P_{r+x}) - (1-P_{r+Q+x}))/Q \\
&= \left(\sum_{x=r+Q+1}^{\infty} (r+Q-x)(1-P_x) - \sum_{x=r+1}^{\infty} (r-x)(1-P_x) \right) /Q \\
&= \left(Q\sum_{x=0}^{\infty} xp_x - \sum_{x=r+1}^{r+Q} \left((r+Q-x)(1-P_x) + \sum_{i=0}^{r} (1-P_i) \right) \right) /Q \\
&= \left(\sum_{x=r+1}^{r+Q} \left(\sum_{i=0}^{x-1} P_i + \lambda\tau - x \right) \right) /Q.
\end{aligned}$$

Finally, for iii) we have

$$\begin{aligned}
\sum_{x=-\infty}^{0} \pi(x) &= \sum_{x=-\infty}^{0} (P_{r+Q-x} - P_{r-x})/Q \\
&= \left(\sum_{x=r}^{\infty} (1-P_x) - \sum_{x=r+Q}^{\infty} (1-P_x) \right) /Q \\
&= \left(\sum_{x=r+1}^{r+Q} (1-P_{x-1}) \right) /Q.
\end{aligned}$$

□

Proof of Theorem 4.1:
It is important to remember the renaming in equation 4.1 so the factor λ is not repeated separately in the calculations below. a) From Lemma 4.9 we have:

$$\begin{aligned}
C(Q,r) &= c_K \frac{1}{Q} + c_h \sum_{x=0}^{r+Q} x\pi_x - c_p \sum_{x=-\infty}^{0} x\pi_x + c_{\tilde{p}} \sum_{x=-\infty}^{0} \pi_x \\
&= \frac{c_K + \sum_{x=r+1}^{r+Q} \left(c_h \sum_{i=0}^{x-1} P_i + c_p \left(\sum_{i=0}^{x-1} P_i + \lambda\tau - x \right) + c_{\tilde{p}}(1-P_{x-1}) \right)}{Q} \\
&= \frac{c_K + \sum_{x=r+1}^{r+Q} \left((c_h + c_p) \sum_{i=0}^{x-1} P_i + c_p(\lambda\tau - x) + c_{\tilde{p}}(1-P_{x-1}) \right)}{Q}
\end{aligned}$$

b) First, we will establish that x^G exists. Then the uniqueness of x^G will be established by showing that $G(x)$ is strictly increasing for $x \geqslant x^G + 1$.

Simple algebra shows that

$$G(x+1) - G(x) = (c_h + c_p)P_x - c_{\tilde{p}}p_x - c_p. \tag{4.19}$$

The above implies the following

$$G(x+1) - G(x) = \begin{cases} \leqslant 0, & \text{iff } (c_h + c_p)P_x - c_{\tilde{p}}p_x \leqslant c_p \\ > 0, & \text{iff } (c_h + c_p)P_x - c_{\tilde{p}}p_x > c_p \end{cases}$$

We notice that for every $x < 0$, $G(x+1) - G(x) = -c_p \leqslant 0$. Thus, x^G is non-negative if it exists. Since $G(x+1) - G(x) > 0$, (because $G(x+1) - G(x) \approx c_h > 0$) for large x, x^G exists.

For the uniqueness, when $c_{\tilde{p}}$ is positive, we show that $G(x)$ is strictly increasing for $x \geqslant x^G + 1$, by first finding a lower bound for x^G, using its definition and Eq. (4.19), as follows.

$$\begin{aligned} G(x^G) - G(x^G - 1) &\leqslant 0 < G(x^G + 1) - G(x^G) \\ G(x^G) - G(x^G - 1) &< G(x^G + 1) - G(x^G) \\ -c_{\tilde{p}}p_{x^G - 1} &< (c_h + c_p - c_{\tilde{p}})p_{x^G} \\ p_{x^G - 1}/p_{x^G} &> (c_{\tilde{p}} - c_h - c_p)/c_{\tilde{p}} \end{aligned}$$

i.e.,

$$x^G > \frac{\tau(c_{\tilde{p}} - c_h - c_p)}{c_{\tilde{p}}}.$$

Using this bound, we obtain for $G(x^G + 2) - G(x^G + 1)$:

$$\begin{aligned} G(x^G + 2) - G(x^G + 1) &= (c_h + c_p)P_{x^G+1} - c_{\tilde{p}}p_{x^G+1} - c_p \\ &= (c_h + c_p)P_{x^G} - c_{\tilde{p}}p_{x^G} \\ &\quad +(c_h + c_p - c_{\tilde{p}})p_{x^G+1} + c_{\tilde{p}}p_{x^G} - c_p \\ &> (c_{\tilde{p}} - (c_{\tilde{p}} - c_h - c_p)\lambda\tau/(x^G + 1))p_{x^G} \\ &> c_{\tilde{p}}(1 - \tau(c_h + c_p - c_{\tilde{p}})/(\tau(c_{\tilde{p}} - c_h - c_p) + c_{\tilde{p}})) \\ &> 0. \end{aligned}$$

Thus, $G(x^G + 2) > G(x^G + 1)$ and by an induction argument on $x > x^G$ we have $G(x+2) > G(x+1 > 0)$ and we see that x^G is the maximal minimizing point.

If $c_{\tilde{p}} = 0$ then $G(x+1) - G(x) = (c_h + c_p)P_x - c_p$. Thus, for x^G we have:

$$\begin{aligned} P_{x^G - 1} &\leqslant c_p/(c_h + c_p), \\ P_{x^G} &> c_p/(c_h + c_p). \end{aligned}$$

Since P_x is increasing in x, we have that $P_x \leqslant P_{x^G - 1}$ for $x < x^G$ and $P_x > P_{x^G}$ for $x > x^G$, thus it follows that x^G is the maximum minimizing point of the function $G(x)$, i.e, $G(x)$ is unimodal.

□

Chapter 5

Concluding Remarks

Infinitely many extensions are possible for further research. For example, a combination of Chapter 3 and 4: a model with quantity discounts, random lead time, and a mixture of backorders and lost sales. Also it is interesting to include perishable products instead of products that do not lose their value or quality. Also it is still a challenge to find a way to prove unimodality for the model described in Chapter 3, similar to the first part of Chapter 4. We conjecture that it is very likely that unimodality can be proven for this model, but until now we have not been able to do find this. Matlab graphs confirm this conjecture. If unimodality can be proven, an efficient algorithm can be constructed for finding the optimal value of Q and r for this model.

Another interesting aspect to investigate is whether another model is wether there are other strategies more efficient for one of the pricing and customer behaviour models above: a dynamic reorder point or a changing value of Q might be some ideas.

Furthermore, with respect to Chapter 2 there are a lot of new research directions. Maybe more complex structures can be handled analogously and the class of successively lumpable Markov Chains can be expanded. Maybe there is a neat way of approximating a complete new class of Markov Chains with a successively lumpable Markov Chain. In the paper concerning the successively lumpable Markov Chains we will show that a direct extension to a Semi-Markov chain is possible.

Finally, I would like to thank Michael Katehakis and Floske Spieksma for their incredible amount of time and support they have given me during the process. Michael especially for the patience and for being both my supervisor and mentor in New Jersey. Floske for helping to find a place abroad to do my master thesis work, various comments on the first article, and a lot of comments during the process of completing this thesis.

Bibliography

[1] Benton, W.C., Seungwook Park, 1996, "A classification of literature on determining the lot size under quantity discounts", *European Journal of Operational Research*, 92, 219-238.

[2] Federgruen, A., Zheng, Y.S. 1992, "An efficient algorithm for finding an optimal (r,Q) policy in continuous review stochastic inventory systems", *Operations Research*, 40, 808-813.

[3] Feinberg, B.N., Chui, S.S., 1987, "A method to calculate steady state distributions of large Markov chains", *Operations Research*, Vol. 35, No 2, 282-290.

[4] Feller W. 1968, *An introduction to probability theory and its applications Vol. 1*, Wiley, New York.

[5] Gallego G. and Katircioglu K. 2007, "Inventory management under highly uncertain demand", *Oper. Res. Letters*, 281-289.

[6] Galliher, H., P. Morse and M. Simmond. 1959, "Dynamics of two Classes of Continuous-Review Inventory Systems", *Operations Research*, 7, 362-384.

[7] Hadley, G., Whitin, T.M. 1963, *Analysis of Inventory Systems*, Prentice Hall International, Englewood Cliffs, N.J.

[8] Kemeny, J., G., Snell, J., L. 1960, *Finite Markov Chains,* D. van Nostrand Company, Inc., Princeton, N.J..

[9] Kim, D.H., Park, K.S., 1985, "(Q, r) Inventory Model with a Mixture of Lost Sales and Time-Weighted Backorders", *The Journal of the Operational Research Society*, 36, 231-238.

[10] Kim, D.S., Smith, R.L., 1989, "An Exact Aggregation Algorithm for a Special Class of Markov Chains", *Technical Report*, 89-2.

[11] Kim, D.S., Smith, R.L., 1990, "An Exact Aggregation-Disaggregation Algortihm for Mandatory Set Decomposable Markov Chains", *Numerical Solution of Markov Chains*, 89-104.

[12] Meyer, C.D., 1989, Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems, *SIAM Rev.*, Vol. 31, No 2, 240-272.

[13] Miranker, W. L., Pan, V. Ya., 1980, "Methods of aggregation", *Linear Algebra and its Applications*, Vol. 29, 231-257.

[14] Munson, CL and Hu, J., 2010, “Incorporating quantity discounts and their inventory impacts into the centralized purchasing decision”, *European Journal of Operational Research*, 201, 581-592.

[15] Papachristos, S. and Skouri, K., 2003, “An inventory model with deteriorating items, quantity discount, pricing and time-dependent partial backlogging”, *International Journal of Production Economics*, 83, 247-256.

[16] Rubalskiy, G., 1972, “Calculation of optimum parameters in an inventory control problem”, *Journal of Computer and Systems Sciences International*, 10, 182-187.

[17] Schweitzer, P.J., Puterman, M.L., Kindle, W.L, 1984, “Iterative Aggregation-Disaggregation Procedures for Discounted Semi-Markov Reward Processes”, *Operations Research*, Vol. 33, No 3, 589-605.