# Numerical analysis of communication systems through Markov reward models

Sándor Rácz

*A dissertation submitted for the degree of Doctor of Philosophy*

Department of Telecommunications and Telematics
Budapest University of Technology and Economics

# Távközlő rendszerek numerikus elemzése Markov hozam modellek alkalmazásával

Rácz Sándor

Témavezetők

Dr. Sallai Gyula
Dr. Faragó András

Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Telematikai Tanszék

# Kivonat

A disszertációban egy eljárást javasolok kommunikációs rendszerek hívás szintű elemzésére Markov hozam modellek felhasználásával. A vizsgált modell a hagyományos többsebességű veszteséges rendszerek modelljének a kibővítése, amely figyelembe vesz az állandó sávszélességigényű osztály mellett adaptív és elasztikus forgalmi osztályokat is. A forgalmi osztályok alapvető teljesítményjellemzői a hívások blokkolásának valószínűsége illetve a kapcsolat alatt elért sávszélesség. A hívás blokkolás definíciója hasonló a többsebességű veszteséges rendszerek modelljében használthoz illetve egy kapcsolat elért sávszélességét a kapcsolat alatt átvitt adat mennyiségének és a kapcsolat idejének a hányadosával definiáljuk. Az elért sávszélesség definiálása adapív és elasztikus osztályokra lehetőséget teremt olyan link megosztási politika tervezésére, amely egyidejűleg biztosítja a megfelelő blokkolási valószínűségeket az adapív és elasztikus forgalmi osztályok számára, illetve figyelembe veszi a maximális megengedett blokkolási valószínűséget az állandó sávszélességigényű osztályra és a minimálisan megkövetelt elért sávszélességet az adaptív és elasztikus kapcsolatok számára. A javasolt link megosztási politika, amely a részleges átfedésű politika alkalmazása a vizsgált rendszerben, paramétereinek meghatározásakor feltételeket szabhatunk az elért sávszélesség várható értékére vagy valamilyen kvantilisére. A elért sávszélesség várható értékére szabott feltétel esetén elegendő a rendszer Markov modelljének egyensúlyi viselkedését vizsgálni, mialatt kvantilisre vonatkozó feltétel esetén a rendszer viselkedését leíró Markov hozam modell tranziens analízise szükséges. Általában a vizsgált renszereink nagy állapotterű modelleket eredményeznek. Mialatt a Markov modellek egyensúlyi viselkedésének meghatározására számos algoritmus létezik nincsen olyan általános módszer, amely megbirkózna $10^4$ állapotnál több állapottal rendelkező Markov hozam modellek tranziens elemzésével.

Két alapvető megközelítési mód található Markov hozam modellek vizsgálatával foglalkozó publikációkban. A rendszer szempontjából vizsgálva a modellt a legfontosabb teljesítményjellemző az egy adott időintervallum alatt elvégzett munka nagysága, azaz a felhalmozott hozam nagysága. A felhasználó szemszögéből nézve rendszer úgy is elképzelhető, mint ami adott mennyiségű munkát valamennyi idő alatt el tud végezni, azaz ekkor egy adott munka teljesítési idejét vizsgáljuk. A disszertáció első fele ezt a kétfajta teljesítményjellemzőt vizsgálja olyan Markov hozam modellek esetében, ahol mind állapotokhoz rendelt ráta alapú hozamokat, mind állapot átmenetekhez rendelt azonnal jellegű hozamokat is tartalmaz a modell.

A disszertáció első fejezete egy rövid áttekintést ad a rendelkezésre álló numerikus algoritmusokról, amelyek Markov modellek egyensúlyi illetve tranziens viselkedésének meghatározására alkalmasak. Természetesen a Markov hozam modellek elemzésére alkalmas módszereket is tárgyalja. A második és a harmadik fejezet egy új módszert javasol, amely alkalmas nagy állapotterű (több mint $10^6$ állapot) Markov hozam modellek tranziens viselkedésének vizsgálatára. A javasolt algoritmus a vizsgált teljesítményjellemzők első $n$ momentumát határozza meg. A negyedik fejezet egy olyan algoritmust ismertet, amely a momentumokból kiindulva meghatároz egy alsó és egy felső korlátot a lehetséges eloszlásfüggvényekre. Az utolsó fejezet bemutatja a javasolt link megosztási politikát, illetve a hozzá kapcsolódó számítási eljárásokat. A javasolt számítási eljárások felhasználják az előző fejezetek által bemutatott új eredményeket is.

# Numerical analysis of communication systems through Markov reward models

Sándor Rácz

Supervisors

Dr. Gyula Sallai
Dr. András Faragó

Department of Telecommunications and Telematics
Budapest University of Technology and Economics

# Abstract

This dissertation proposes a method for call level analysis of a communication system through Markov reward models. The studied model is the extension of classical multi-rate loss models and considers non-adaptive stream, adaptive stream and elastic traffic classes. The main performance measures of the traffic classes are the blocking probabilities and the throughputs measures. The meaning of blocking probabilities are the same as in multi-rate loss models and the throughput of a flow is defined as the amount of transferred data divided by the transmission time. Considering throughput measures for non-adaptive stream (e.g. an adaptive video codec) and for elastic traffic (e.g. file transfer protocol) gives a possibility to design link allocation policy which is able to provide predefined blocking probability for the adaptive and for the elastic flows, while it is able to take into account the blocking probability constriants for non-adaptive stream flows and the minimum throughput constraint for the elastic and adaptive flows. Determining the parameters of the proposed link allocation policy, which is the adaptation of the partial overlap policy, we can consider constraint on the average throughput or constraint on throughput threshold. Applying constraint on the average throughput requires only the steady-state analysis of the Markovian model of the proposed policy meanwhile the transient analysis of the Markov reward model of the policy is needed to apply throughput threshold constraint. In general, our system models result in large Markovian models. Meanwhile there are several algorithms for the steady-state analysis there is no general algorithm which can deal with the transient analysis of Markov reward models with more than $10^4$ states.

Two main different point of view have been assumed in the literature when dealing with Markov reward models. In the system oriented point of view the most significant measure is the total amount of work done by the system in a finite interval (accumulated reward). In the user oriented point of view the system is regarded as a server, and the emphases of the analysis is on the ability of the system to accomplish an assigned task in due to time (completion time). The first part of the dissertation study these two measures of Markov reward models where state associated rate rewards and transition associated impulse rewards are applied.

The first chapter of the dissertation gives a brief insight into the available numerical algorithms for performing steady-state and transient analysis of Markovian models including also Markov reward models. The second and the third chapter propose new algorithms for the transient analysis of large Markov reward models. The algorithms can determine the first $n$ moments of the reward measures in large models with more than $10^6$ states. The forth chapter describes a numerical algorithm which determines upper and lower bounds of distribution function based on its moments. The last chapter presents the proposed link allocation policy with the complete link allocation procedure of a single link. The link allocation procedure uses the new Markov reward models' analysis algorithms and the proposed distribution estimation method.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

I wrote this dissertation to summarize the significant results of my three-year Ph.D. research ($1997 - 2000$). The dissertation has been prepared at the Budapest University of Technology and Economics as a partial fulfillment of the requirements for obtaining the Ph.D. degree. I have been a co-author of twelve conference papers [C1]-[C12] and seven journal papers [J1]-[J7]. I would like to express my greatest thanks to Miklós Telek and Gábor Fodor for their guidance throughout my studies and for the joint work.

## 1.1 Motivation

In recent years there have been significant advances in researching and standardizing mechanisms that are capable of providing service differentiation in the Internet. While there still seems to be a wide span of the methods which aim at providing differentiation among contending flows, it is widely accepted that there is a need for traffic engineering mechanisms which control the access of the different traffic classes to network bandwidth resources. Generally, the issue of *bandwidth sharing* should be considered in the context of *dynamically arriving and departing flows*, which naturally calls for the application of the classical multi-rate models [Ros95]. Multi-rate models have proved useful in the dimensioning and performance evaluation of circuit switched as well as ATM networks. Thus, they provided motivation for extending applicability of this modelling paradigm to the Internet context.

Unfortunately, a direct application of the multi-rate models for the traffic engineering in the Internet is non-trivial, because

- by definition, it is not possible to associate a constant bandwidth with elastic services, like the best effort without minimum rate guarantee or the "better than best effort" with minimum rate guarantee type of services and the bandwidth occupied by the elastic flows depends on the current load on the link and on the scheduling and rate control algorithms applied in the network nodes;

- the notion of blocking, when applied to elastic flows, needs to be reconsidered because an arriving elastic flow might get into service even if at the arrival instant there is no or very small bandwidth available;

- for many services, we need to take account of the fact that the actual residency time of an elastic flow depends on the throughput which the flow receives, for instance, an ftp session would last longer if its throughput decreases and real-time services' holding time,

on the other hand, is insensitive to the throughput, which is the case, for instance, with a flow associated with an adaptive video codec.

Since the reservation based multi-rate models can not directly used, the multi-service environment which is the meaningful extensions of multi-rate models is applied so as to allow the inclusion of both QoS-assured and elastic traffic into a common framework.

Markov reward models (*MRM*s) allow us to perform the *quantitative analysis* of multi-service environment. Hence, we need for an efficient *MRM* analysis algorithm which can be applied to analyze complex call level models. Nevertheless *MRM*s have been studied since a long time [How71] a significant bottleneck of the application of *MRM*s for analysis of complex systems is the lack of numerical methods for large models. While the considered telecommunication systems result in large *MRM*s. This fact moved my research direction in *MRM*s.

Consequently, my Ph.D. activity covered two different research fields. I have put research activity on the extension of the multi-rate models to the Internet context meanwhile I have put a strong effort to developing numerical algorithms to increase the applicability of *MRM*s.

## 1.2 Related works and contributions

Here I provide a review of the research results and I identify my contribution to this line of research.

### Modelling Multi-service environment

The theory of multi-rate models is covered by [Kau81, Ros95, Rob96]. Application examples of this modelling paradigm include those concentrating on routing and call admission algorithms for QoS assured traffic classes in [SVVP91, DM94] and also those that are concerned with the optimal sharing of link bandwidth resources as in [CLW95, MMR96, BM98, MRW98]. However, none of these models addresses the issue of applying this model to cases where elastic traffic is also present in the network.

The notion of call admission control for elastic traffic and fairness issues are discussed in a number of publications, see [Rob98, MR99a, MR99b, MR99c]. In fact, I feel that my publications are in line with these papers; and extend them by *proposing a computational model to arrive at specific performance measures* on the throughputs and the blocking probabilities. The blocking probability vs. throughput tradeoff is also emphasized and directly connected to the issue of charging in [Kel97, GK99].

The extension of the multi-rate model to include elastic services was proposed independently of each other in [BF96] and in [AAT97]. The application of *MRMs* to compute the mean transfer time of files with exponentially distributed sizes and the blocking probabilities for the complete sharing method and assuming two traffic classes was proposed already in [ABFT97]. Those results have been extended for the partial overlap link allocation strategy ("mixed scenario") in [QBM99], where the authors are concerned with the computation of the blocking probabilities and also of the first moment of the transfer time of a file of size $x$.

The impact of pricing on the optimal bandwidth sharing strategy, again assuming two traffic classes is considered in [AAT97] and in [FNB98].

From a more practical point of view, specifically examining the TCP traffic (which is the predominant example on the elastic traffic class in the Internet), Feng *et al.* find it beneficial to provide a minimum throughput for TCP flows, because in that case the TCP algorithm can be modified such that the "goodput" of TCP connections is much improved [FKSS99].

I summarize the contribution of my papers to this line of work in Chapter 6. We propose a *call level model* [J1, J2, C4, C8] , called multi-service environment, for analyzing call level performance measures such as the blocking probability and the throughput; and we provide *algorithms* for the numerical analysis [J3, C1, C2, C3, C9] as well.

## Markov reward models

The stochastic reward processes have been studied since a long time [How71], because the possibility of associating a reward variable to each system state increases the descriptive power and the modelling flexibility. However, only recently, stochastic reward models (*SRM*s) have received attention as a modelling tool in performance evaluation of computer and communication systems. Common assignments of the reward rates are: execution rates of tasks in computing systems (the computational capacity) [Bea78, SSM87], number of active processors (or processing power) [Bob87, GDI88], throughput [Mey82], available bandwidth [BFAT97] or average response time [LW88].

Two main different points of view have been assumed in the literature when dealing with *SRM* [KNT86]. In the *system oriented* point of view the most significant measure is the total amount of work done by the system in a finite interval. This measure is often referred to as *performability* [Mey82]. In the *user oriented* (or *task oriented*) point of view the system is regarded as a server, and the emphasis of the analysis is on the ability of the system to accomplish an assigned task in due time. Consequently, the most characterizing measure becomes the probability of accomplishing an assigned service in a given time.

A unified formulation to the system oriented and the user oriented point of view was provided in [KNT86] together with the double Laplace transform expression of the completion time for the case when the underlying stochastic process is a continuous time Markov chain (*CTMC* ). This case is referred to as Markov reward model (*MRM* ).

Various numerical techniques were proposed for the evaluation of the system and the user oriented measures of *MRM*s. Some of these methods calculate the distribution of reward measures. The distribution, in double transform domain, can be obtained by a symbolic matrix inversion. If the size of the state space allows to obtain the solution of the symbolic matrix inversion then multidimensional numerical inverse transform methods [CLW94] can provide the time domain results, but, due to the computational complexity of the symbolic inversion of matrices, this approach is not applicable for models with more than 20 states.

In time domain, reward measures can be described either by a set of equations with convolution integrals or by a set of partial differential equations, but the numerical methods compute the distribution in time domain are usually based on the evaluation of a double summation, where both of the summation parameters increase to infinity. The discrete summations are obtained by adopting the randomization technique [Ste94]. The randomization technique usually provides nice numerical properties and an overall error bound. The numerical methods based on this approach [DG91, NS96, SG98] differ in the complexity and memory requirement of one iteration step.

*MRM*s with special features allow special, effective numerical approaches. In the case when the underlying *CTMC* has an absorbing state, in which no useful work is performed, it is easy to evaluate the limiting distribution of performability [Bea78]. The numerical method in [GT87] makes use of a special structure of the underlying *CTMC*.

The numerical analysis of the distribution of reward measures is, in general, more complex than the computation of moments of those reward measures. The mean of the performability can be obtained by the transient analysis of the underlying *CTMC*. A numerical convolution

approach is proposed in [IDH86] to evaluated the $(n+1)$-th moment of the performability based on its $n$-th moment. A similar approach is followed in [TPF98] to calculate moments of the user oriented measures, but the high computational complexity of the numerical convolution does not allow to apply this approach for the analysis of *MRM*s with large ($> 100$) state spaces.

Other *direct* methods make use of a spectral - or partial fraction decomposition, which is relatively easy for acyclic *CTMC*s, since the eigenvalues of the generator matrix are available in its diagonal [STR88]. The subclass of *MRM*s where the user has an associated phase type distributed random work requirement was studied in [BT90]. In this case the completion time is phase type distributed, i.e., an "extended" *CTMC* can be defined which characterizes the distribution of the completion time.

The contribution of my papers to this line of work is as follows. We propose *algorithms for the evaluation of large Markov reward models*. The proposed analysis method [C2, C10] contains the automatic generation of the (commonly large) Markov reward model, the calculation of the moments of the reward measures and the estimation of the distribution based on the obtained moments.

## 1.3 Notation

Here I introduce that only notation required for the statement of main results in this dissertation. I give the basic rules which help to make difference among mathematical objects. The applied objects are matrices, vectors, random variables, stochastic processes, sets, functions, etc. The common notation rules make reading the dissertation easier.

I use different letter types for different objects. I indicate matrices with boldface letters (e.g. $\mathbf{Q}$). Vectors are denoted by underlying letters (e.g. $\underline{v}$). I consider a vector indicated with underlying as a column vector and I describe a row vector as the transpose of a column vector (e.g. $\underline{v}^T$ is a row vector). I use calligraphic letters to represent random variables, sets and vector spaces (e.g. $\mathcal{X}$).

I introduce some constant objects. The null vector $\underline{0}$ that all entries are null, the identity vector $\underline{h}$ that all entries equal 1, the null matrix $\mathbf{0}$ and the identity matrix $\mathbf{I}$ are used as constant vectors and matrices. Furthermore, two special objects help our with analysis work, i.e., the well-known unit step $u[x]$ and the unit impulse $\delta[x]$.

I briefly review the Laplace transform. The Laplace transforms are a powerful practical tool. The operation in transform domain could result in a simpler expression and the procession of a Laplace domain result can be treated separately making use of well-known theoretical results. The definition of Laplace (LT) and Laplace-Stieltjes (LST) transform of a function $f(t)$ are

$$\text{LT} \;:\; f^*(s) = \int_0^\infty f(t)e^{-st}\,dt \qquad \text{and} \qquad \text{LST} \;:\; f^\sim(s) = \int_0^\infty e^{-st}\,df(t) \quad .$$

I apply the Laplace and Laplace-Stieltjes transform for matrix functions as well, e.g.

$$\text{LST}\left\{e^{\mathbf{Q}t}\right\} = [s\mathbf{I} - \mathbf{Q}]^{-1}$$

where the $i,j^{th}$ entry of the right hand side matrix is the Laplace-Stieltjes transform of the $i,j^{th}$ entry of the left hand side matrix. Excellent overviews of Laplace transforms can be founded in [Fel66, pp 429-497] and in [Kle75, Appendix I].

# Chapter 2

# Numerical algorithms for Markovian models

A broad spectrum of solution techniques is available for performing steady-state and transient analysis of Markov chains and Markov reward models. This chapter gives a brief insight into the available numerical algorithms.

## 2.1 Steady-state analysis of Markov chains

This section presents algorithms which compute the *steady-state distribution* of a homogeneous Markov chain with finite states. This section based on [Ste94, BBC94, IM98].

The steady-state solution of a Markov chain can be formulated as the solution of a special linear system. We assume that the stationary solution vector $\underline{p}$ (steady-state distribution) exists and unique. Calculating the stationary solution corresponds to solving a set of steady-state equations of form

$$\mathbf{Q}^T \cdot \underline{p} = \underline{0} \qquad \text{or} \qquad \mathbf{P}^T \cdot \underline{p} = \underline{p} \tag{2.1}$$

where the first term refers to a continuous time Markov chain ($CTMC$) and the second one refers to a discrete time Markov chain ($DTMC$). The generator matrix $\mathbf{Q}$ of a $CTMC$ and the one-step transition matrix $\mathbf{P}$ of a $DTMC$ are singular and have rang $n-1$, where the considered Markov chains have $n$ states. Furthermore, if it is necessary, we can make a non-singular linear system from (2.1) replacing an arbitrary equation with the $\sum p_i = 1$ normalization equation.

Alternatively, we may interpret the computation of $\underline{p}$ as an eigenvector problem. The normalized right-hand eigenvector of $\mathbf{Q}^T$ for eigenvalue 0 is the stationary solution vector of a $CTMC$ and the eigenvector of $\mathbf{P}^T$ for eigenvalue 1 is the stationary solution vector of a $DTMC$.

There are close relation between the two obtained linear systems in (2.1) and can be transformed from each other. Starting from a $CTMC$, determined by its generator matrix $\mathbf{Q}$, and setting $\mathbf{P} = \mathbf{Q}/q + \mathbf{I}$ with $q \geq max_i|q_{ii}|$ we can obtain that $\mathbf{P}$ is a one-step transition matrix of a $DTMC$ and vice-versa.

There are several types of methods which can deal with the steady-state analysis. In most cases numerical methods which are appropriate to solve a *large linear system* $\mathbf{A} \cdot \underline{x} = \underline{b}$ will be presented. Of course, some methods highly utilize the fact that $\underline{b} = \underline{0}$ or $\underline{b} = \underline{h}$. We can consider three main types of methods.

- Direct methods

- Iterative methods
- Projection methods

The following three subsections summarize these methods.

### 2.1.1 Direct methods

Numerical methods that compute the solution of a mathematical problem in a fixed number of operations are generally referred to as *direct methods*. Direct methods have the advantage that an upper bound on the time required to obtain the solution could be determined in advance. More important, for certain class of problem, direct methods result in more accurate answers being obtained in less time [Ste82].

#### Gaussian elimination

Gaussian elimination may be viewed as a transformation from a linear system $\mathbf{A} \cdot \underline{x} = \underline{b}$ to an equivalent linear system $\mathbf{U} \cdot \underline{x} = \underline{d}$ in which the matrix $\mathbf{U}$ is upper triangular; and from it the solution $\underline{x}$ can be obtained through a simple back substitution. The procedure of obtaining the matrix $\mathbf{U}$ from the matrix $\mathbf{A}$ is called the reduction phase and it is accomplished in $(n-1)$ steps. The $i$-th step eliminates all non-zero elements below the $i$-th diagonal element by adding a suitable multiple of the $i$-th equation into each equation below the $i$-th. This algorithm requires $n^3/2 + n^2/2 - 5n/6$ additions and multiplications, and $n^2/2 + n/2$ divisions.

#### The LU decomposition

The coefficient matrix $\mathbf{A}$ of a linear system can be written as the product of a lower triangular matrix $\mathbf{L}$ and an upper triangular matrix $\mathbf{U}$, i.e.,

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U} \quad .$$

The solution of the considered linear system, $\mathbf{A} \cdot \underline{x} = \underline{b}$, is then

$$\underline{x} = \mathbf{U}^{-1} \cdot \mathbf{L}^{-1} \cdot \underline{b}$$

and can be derived from a forward and a backward substitutions. Doolittle decomposition assumes that diagonal elements of $\mathbf{L}$ equal 1 in contrast with Crout decomposition which assumes that diagonal elements of $\mathbf{U}$ equal 1. If we assume that Doolittle decomposition is being performed, then elements of matrix $\mathbf{U}$ can be determined by

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \qquad \text{for} \quad i \leq j$$

and elements of matrix $\mathbf{L}$ can be obtained by

$$l_{ij} = \left[ a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right] / u_{jj} \qquad \text{for} \quad i > j \ .$$

Gaussian elimination and Doolittle decomposition are closely related. The reduction phase of Gaussian elimination is none other than the derivation of matrix $\mathbf{U}$.

**The LDU decomposition**

The coefficient matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{D} \cdot \mathbf{U}$$

where diagonal elements of the matrix $\mathbf{L}$ and the matrix $\mathbf{U}$ equal 1 and the matrix $\mathbf{D}$ is a diagonal matrix. The solution of $\mathbf{A} \cdot \underline{x} = \underline{b}$, which is in the form

$$\underline{x} = \mathbf{U}^{-1} \cdot \mathbf{D}^{-1} \cdot \mathbf{L}^{-1} \cdot \underline{b} \ ,$$

can be calculated similarly as in the LU decomposition.

**The GTH algorithm**

The GTH (Grassman-Taksar-Heyman) algorithm [She82, GTH85] is a *special version of Gaussian elimination* that has attributes that appear to make it even more stable than the usual version. In GTH the diagonal elements are obtained by summing the off-diagonal elements rather then performing a subtraction. It is known that subtractions can sometimes lead to loss of significance in the representation of a real number. The concept evolved from probabilistic arguments, and the originally suggested implementation is a backward elimination procedure.

One may conclude that, if compact storage schemes are used then compared with the best possible implementation of Gaussian elimination GTH is likely to require either significantly more memory, significantly more time, or both. Since Gaussian elimination is known to be stable, it may be felt that the only real need for GTH occurs when the problem is very ill-conditioned.

### 2.1.2 Iterative methods

The term "iterative method" refers to a wide range of techniques that use successive approximations to obtain more and more accurate solution. Iterative methods can be expressed in a simple form

$$\underline{x}^{(k)} = \mathbf{B} \cdot \underline{x}^{(k-1)} + \underline{c}$$

where neither the matrix $\mathbf{B}$ nor the vector $\underline{c}$ depend upon the iteration count $k$. This section presents four iterative methods, i.e., the *power method*, *Jacobi method*, *Gauss-Seidel method* and the *successive overrelaxation method* (SOR).

**The power method**

The power method is applicable to the derivation of a *right-hand eigenvector corresponding a dominant eigenvalue*, $\lambda_1$, of matrix $\mathbf{A}$. Assume that $|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|$. The power method is described by the iterative procedure as follows

$$\underline{x}^{(k)} = \frac{1}{\gamma_{k-1}} \ \mathbf{A} \cdot \underline{x}^{(k-1)}$$

where $\gamma_k = \| \ \mathbf{A} \cdot \underline{x}^{(k)} \ \|_\infty$ is a normalizing factor. The rate of the convergence depends on the ratios of eigenvalues (i.e. $|\lambda_i|/|\lambda_j|$). The smaller these ratios, the quicker the method tends to corresponding eigenvector. Thus, the power method will not perform satisfactory when $|\lambda_1| \approx |\lambda_2|$.

### Jacobi method

Jacobi method is easily derived by examining each of the $n$ equations in the linear system $\mathbf{A} \cdot \underline{x} = \underline{b}$ in isolation. If the $i$-th equation

$$\sum_{j=1}^{n} a_{ij} x_j = b_j$$

is solved for the value of $x_i$ while assuming the other entries of $\underline{x}$ remain fixed then we can obtain

$$x_i = (b_i - \sum_{j \neq i} a_{ij} x_j)/a_{ii}$$

which is called as Jacobi method. In matrix terms, the definition of Jacobi method can be expressed as

$$\underline{x}^{(k)} = \mathbf{D} \cdot (\mathbf{L} + \mathbf{U}) \cdot \underline{x}^{(k+1)} + \mathbf{D}^{-1} \cdot \underline{b}$$

where the matrices $\mathbf{D}, -\mathbf{L}$ and $-\mathbf{U}$ represent the diagonal, the strictly lower-triangular and the strictly upper-triangular parts of the matrix $\mathbf{A}$, respectively. This method is easy to implement, but its convergence is slow.

### Gauss-Seidel method

Gauss-Seidel method is similar to Jacobi method, expect that it uses updated values as soon as they are available. Formally, the Gaus-Seidel iteration formula is given by

$$x_i^{(k)} = \left( b_i - \sum_{j<i} a_{ij} x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k-1)} \right) / a_{ii} \ .$$

In matrix terms, the definition of Jacobi method can be expressed as

$$\underline{x}^{(k)} = (\mathbf{D} - \mathbf{L})^{-1} \cdot (\mathbf{U} \cdot \underline{x}^{(k-1)} + \underline{b}) \ .$$

As before, $\mathbf{D}, -\mathbf{L}$ and $-\mathbf{U}$ represent the diagonal, the strictly lower-triangular and the strictly upper-triangular parts of the matrix $\mathbf{A}$, respectively. In general, if Jacobi method converges, Gauss-Seidel method will converge faster, though still relatively slowly.

### The successive overrelaxation method

The successive overrelaxation method (SOR) can be derived from Gauss-Seidel method by introduction an extrapolation parameter $\omega$. This extrapolation takes the form of a weighted average between the previous iterate and the computed Gauss-Seidel iterate successively for each component given by

$$x_i^{(k)} = \omega \hat{x}_i^{(k)} + (1 - \omega) x_i^{(k-1)}$$

where $\hat{x}$ denotes a Gauss-Seidel iterate. In matrix terms, the SOR algorithm can be written as follows.

$$\underline{x}^{(k)} = (\mathbf{D} - \omega \mathbf{L})^{-1} \cdot (\omega \mathbf{U} + (1 - \omega)\mathbf{D}) \cdot \underline{x}^{(k-1)} + \omega (\mathbf{D} - \omega \mathbf{L})^{-1} \cdot \underline{b}$$

If $\omega = 1$, the SOR method simplifies to Gauss-Seidel method. A theorem due to Kahan [Kah58] shows that the SOR method fails to converge if $\omega$ is outside the interval $(0, 2)$. The SOR may accelerate convergence of Gauss-Seidel ($\omega > 1$, over-relaxation) or may yield convergence when Gauss-Seidel fails ($0 < \omega < 1$, under-relaxation). From the optimal choice of $\omega$, the SOR method may converge faster than Gauss-Seidel by an order of magnitude. The choice of an optimal, or even a reasonable, value for $\omega$ has been the subject of much study. Some results have been obtained for certain classes of matrices. Unfortunately, little is known at present for an arbitrary linear system.

**Block iterative methods**

The concept of iterative methods can be extended to partitioned linear systems. By partitioning the matrix $\mathbf{A}$ and the vector $\underline{x}$ we can rewrite the homogeneous linear system $\mathbf{A} \cdot \underline{x} = \underline{0}$ as

$$\begin{pmatrix} \mathbf{A}_{11} & \dots & \mathbf{A}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{N1} & \dots & \mathbf{A}_{NN} \end{pmatrix} \cdot \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{pmatrix} = \underline{0}$$

and we can write the matrix $\mathbf{A}$ as $\mathbf{D}_N - (\mathbf{L}_N + \mathbf{U}_N)$ where the $\mathbf{D}_N$ is a block diagonal matrix and the $\mathbf{L}_N$ and the $\mathbf{U}_N$ are respectively strictly lower and upper block triangular matrices.

The block version of Gauss-Seidel method can be written as

$$\mathbf{D}_{ii} \cdot \underline{x}_i^{(k+1)} = \sum_{j=1}^{i-1} \mathbf{L}_{ij} \cdot \underline{x}_j^{(k+1)} + \sum_{j=i+1}^{N} \mathbf{U}_{ij} \cdot \underline{x}_j^{(k)} \quad .$$

This method implies that at each iteration we must solve $N$ linear systems separately.

In a similar manner to block Gauss-Seidel, we may also define a block Jacobi method as

$$\mathbf{D}_{ii} \cdot \underline{x}_i^{(k+1)} = \sum_{j=1}^{i-1} \mathbf{L}_{ij} \cdot \underline{x}_j^{(k)} + \sum_{j=i+1}^{N} \mathbf{U}_{ij} \cdot \underline{x}_j^{(k)} \quad .$$

Finally, a block SOR method can be defined by

$$\underline{x}_i^{(k+1)} = (1-\omega)\underline{x}_i^{(k)} + \omega \left\{ \mathbf{D}_{ii}^{-1} \left( \sum_{j=1}^{i-1} \mathbf{L}_{ij} \cdot \underline{x}_j^{(k+1)} + \sum_{j=i+1}^{N} \mathbf{U}_{ij} \cdot \underline{x}_j^{(k)} \right) \right\} \quad .$$

In general such block iterative methods require more computation per iteration, but this is offset by a faster rate of convergence.

### 2.1.3 Projection methods

The basic idea behind projection methods is quite simple. They extract an approximation of the exact solution of a linear system from *a small-dimension subspace*. In general, the dimension of this subspace more less than the dimension of the considered linear system.

The basic projection method is defined by three objects: a subspace $\mathcal{K}$ of dimension $m$ from which the approximation is selected; a subspace $\mathcal{L}$ that sets constraints necessary for extracting the unique approximation from $\mathcal{K}$. The third object is the constraint type. Consequently, we need to define two subspaces $\mathcal{K}$ and $\mathcal{L}$ and the constraint type for a projection method. Naturally, the "performance" of a projection method depends on those three objects. In general, the

constraint type comes from the problem direct (e.g. the approximated solution be orthogonal to the subspace $\mathcal{L}$).

The selection of the two subspaces may be very difficult. The majority of projection methods adopt *Krylov subspace* method. Because we hope that a "good" approximation lies in a low dimension Krylov subspace and there is an "efficient" algorithm to construct this subspace. Formally, for a general linear system , $\mathbf{A} \cdot \underline{x} = \underline{b}$, the Krylov subspace of order $m$ is defined as

$$\mathcal{K}_m(\mathbf{A}, \underline{v}) = span\{\underline{v}, \mathbf{A} \cdot \underline{v}, \ldots, \mathbf{A}^{m-1} \cdot \underline{v}\} \ .$$

One can see that the construction of a $m$ dimension Krylov subspace requires only $m-1$ matrix-vector multiplications, hence we can efficient span this subspace. But for the other requirement, i.e. a good approximation lies in a low dimension Krylov subspace, there is only a general theorem (e.g. [IM98]) which can not be apply to our context (i.e. the determination of the degree of minimal polynomial of matrix $\mathbf{A}$ is a very complex task).

**Theorem 2.1** *If the minimal polynomial of a nonsingular matrix* $\mathbf{A}$ *has degree* $m$*, then the solution to* $\mathbf{A} \cdot \underline{x} = \underline{b}$ *lies in the subspace* $\mathcal{K}_m(\mathbf{A}, \underline{b})$*.*

In a Krylov subspace $\mathcal{K} = \mathcal{K}_m(\mathbf{A}, \underline{b})$ the general idea of projection methods is to compute the approximation as

$$\underline{x}^{(m)} = c_1 \underline{y}_1 + \ldots + c_m \underline{y}_m$$

where $\underline{y}_1, \ldots, \underline{y}_m$ form a basis of Krylov subspace and the coefficients $c_1, \ldots, c_m$ are chosen according to the optimal criterion determined by the subspace $\mathcal{L}$ and the constraint type (e.g. $\mathcal{L} = \mathcal{K}$ and the residual vector $\underline{r} = \mathbf{A} \cdot \underline{x}^{(m)} - \underline{b}$ is orthogonal to the subspace $\mathcal{L}$). Gram-Schmidt algorithm (see Table 2.1) is widely used to construct an orthonormal basis for a subspace, this step is necessary, because the vectors $\{\underline{b}, \mathbf{A} \cdot \underline{b}, \ldots, \mathbf{A}^{m-1} \cdot \underline{b}\}$ span a Krylov subspace but they are not orthogonal.

| **Input** |
|---|
| **V**, a basis for the subspace $\mathcal{K}$ |
| **Output** |
| **Y**, an orthonormal basis for $\mathcal{K}$<br>**R**, the solution of $\mathbf{V} = \mathbf{Y} \cdot \mathbf{R}$ |
| $r_{11} = \parallel \underline{v}_1 \parallel_2$;<br>$\underline{y}_1 = \underline{v}_1 / r_{11}$;<br>**For** $j := 1$ **To** Dim($\mathcal{K}$) **Do**<br>**Begin**<br>**For** i:=1 **To** j-1 **Do**<br>$r_{ij} = \underline{y}_i^T \cdot \underline{v}_j$;<br>$\underline{y}_j = \underline{v}_j - \sum_{i=1}^{j-1} r_{ij} \ \underline{y}_i$;<br>$r_{jj} = \parallel \underline{y}_j \parallel_2$;<br>$\underline{y}_j = \underline{y}_j / r_{jj}$;<br>**End**; |

Table 2.1: *Algorithm : Gram-Schmidt orthogonalization, construction of an orthonormal basis*

### The basic projection process

As before, consider a linear system

$$\mathbf{A} \cdot \underline{x} = \underline{b}$$

and let the subspace $\mathcal{K}$ be spanned by a set of $m$ vectors $\mathbf{V} = [\underline{v}_1, \underline{v}_2, \ldots, \underline{v}_m]$. One can write an approximate solution as a linear combination of these vectors, $\underline{x} = \mathbf{V} \cdot \underline{y}$, where $\underline{y}$ is a vector of $m$ components. In order to extract a unique $\underline{y}$, one possibility is to require that the residual vector, $\underline{b} - \mathbf{A} \cdot \underline{x}$, be orthogonal to $m$ linearly independent vectors $\mathbf{W} = [\underline{w}_1, \underline{w}_2, \ldots, \underline{w}_m]$. The vectors of matrix $\mathbf{W}$ are a basis of the subspace $\mathcal{L}$. The above constraints mean that the residual vector is required to be orthogonal to the subspace $\mathcal{L}$, so we may rewrite constraints as

$$\mathbf{W}^T \cdot (\underline{b} - \mathbf{A} \cdot \mathbf{V} \cdot \underline{y}) = \underline{0}$$

which yields, assuming the matrix $\mathbf{W}^T \cdot \mathbf{A} \cdot \mathbf{V}$ is nonsingular,

$$\underline{y} = [\mathbf{W}^T \cdot \mathbf{A} \cdot \mathbf{V}]^{-1} \cdot \mathbf{W}^T \cdot \underline{b} \, .$$

Note that, the matrix inverse is applied to a smaller $m$ dimension matrix ($m \ll n$).

If an initial guess $\underline{x}_0$ to the solution of the linear system is known, then we may seek a correction vector $\underline{\delta}$ to the initial guess $\underline{x}_0$ such that the vector $\underline{x}_0 + \underline{\delta}$ will be a solution of the considered linear system. This approach gives the way to construct an *iterative projection method*. One starts with an initial guess $\underline{x}_0$ of the solution then seeks the correction vector $\underline{\delta}_1$ and constructs the new approximation of the solution $\underline{x}_1 = \underline{x}_0 + \underline{\delta}_1$. In the $k$-th step, we use the $(k-1)$-th approximated solution $\underline{x}_{k-1}$ as an initial approximation to determine the $k$-th approximation. Hence, each iteration step eliminates some components of the residual vector.

As a result, if we set $\underline{r}_{k-1} = \underline{b} - \mathbf{A} \cdot \underline{x}_{k-1}$, the projection step must be applied to the linear system

$$\mathbf{A} \cdot \underline{\delta}_k = \underline{r}_{k-1}$$

to compute the unknown vector $\underline{\delta}_k$ and the new approach of $\underline{x}$. Table 2.2 shows the general iterative projection method.

$\underline{x} := \underline{x}_0$;
**Repeat**
  Select a pair of subspaces $\mathcal{K}$ and $\mathcal{L}$;
  Choose bases $\mathbf{V}$ and $\mathbf{W}$ for the subspaces;
  $\underline{r} := \underline{b} - \mathbf{A} \cdot \underline{x}$;
  $\underline{y} = (\mathbf{W}^T \cdot \mathbf{A} \cdot \mathbf{V})^{-1} \cdot \mathbf{W}^T \cdot \underline{r}$;
  $\underline{x} = \underline{x} + \mathbf{V} \cdot \underline{y}$;
**Until** Convergence;

Table 2.2: *Method : Prototype of a general iterative projection method*

### Rayleigh-Ritz method

Rayleigh-Ritz method extracts eigenvalue and eigenvector approximations from a given subspace. Table 2.3 shows the algorithm. The approximation to eigenvalues and eigenvectors of

the matrix $\mathbf{A}$ are taken to be $\lambda_i$ and $\underline{y}_i$, respectively. Let $\mathbf{\Lambda} = diag\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ be a diagonal matrix of the eigenvalues of $\mathbf{B}$ and let $\mathbf{Y} = [\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_m]$. It may be shown that this approximations are optimal in the sense that $\| \mathbf{A} \cdot \mathbf{Y} - \mathbf{Y} \cdot \mathbf{\Lambda} \|_F$ is less than the norm of the residual matrix from any other set of orthonormal vectors in the same subspace ($\mathcal{K}$) with any approximate eigenvalues [Par82].

| |
|---|
| Select a subspace $\mathcal{K}$; |
| Choose an orthonormal basis $\mathbf{V}$; |
| $\mathbf{B} = \mathbf{V}^T \cdot \mathbf{A} \cdot \mathbf{V}$; |
| **Solve** ($\mathbf{B} \cdot \underline{x}_i = \lambda_i \underline{x}_i, \quad \| \underline{x}_i \|_2 = 1$); |
| $\underline{y}_i = \mathbf{V} \cdot \underline{x}_i$; |

Table 2.3: *Algorithm : Rayleigh-Ritz eigensystem approximation of matrix* $\mathbf{A}$

### Arnoldi method

Arnoldi method may be used to compute approximations to the largest eigenvalues ($\lambda_i$) and the corresponding eigenvectors ($\underline{y}_i$) of an unsymmetrix matrix $\mathbf{A}$ (Table 2.4). Unlike simultaneous iteration, these eigenvalues are not necessarily the largest in modulus, but largest in an algebraic sense. The basic idea is that we construct the $m$ dimension Krylov subspace ($\mathcal{K}_m$) of the matrix $\mathbf{A}$ and a nonzero vector $\underline{v}$ then uses the modified Gram-Schmidt ortogonalization procedure to construct an orthonormal basis for this subspace. After that the solution of the considered eigenproblem is selected from the constructed Krylov subspace, so that,

$$\underline{z} \cdot (\mathbf{A} \cdot \underline{y}_i - \lambda_i \, \underline{y}_i) = 0 \quad \forall \underline{z} \in \mathcal{K}_m \quad .$$

In other words, the residual vector between the approximated and the exact eigenvector is orthogonal to subspace $\mathcal{K}_m$.

### The generalized minimal residual method (GMRES)

A disadvantage of Arnoldi method is that it does not satisfy an optimality property. The GMRES method does not suffer from this disadvantage. The approximated solution $\underline{x}_m$ of the considered linear system is chosen from Krylov subspace in such a way as to minimize $\| \underline{b} - \mathbf{A} \cdot \underline{x}_m \|_2$. The GMRES method is described in [MSS84] in detail.

### The conjugate gradient method for the normal equation (CGNR)

The well-known conjugate gradient method has proven itself to be very effective at solving linear system whose coefficient matrix $\mathbf{A}$ is symmetric and positive-definite. When $\mathbf{A}$ is a real, nonsingular matrix, the matrix $\mathbf{A}^T \cdot \mathbf{A}$ is symmetric and positive-definite, and so conjugate gradient method may be used to compute the solution of $\mathbf{A} \cdot \underline{x} = \underline{b}$ from

$$\mathbf{A}^T \cdot \mathbf{A} \cdot \underline{x} = \mathbf{A}^T \cdot \underline{b} \quad .$$

These are called the normal equations. Alternatively, the conjugate gradient method may be applied to the linear system

$$\mathbf{A}^T \cdot \mathbf{A} \cdot \underline{z} = \underline{b}$$

and the solution to $\mathbf{A} \cdot \underline{x} = \underline{b}$ computed as $\underline{x} = \mathbf{A}^T \cdot \underline{z}$. It is referred to this method as CGNR (conjugate gradient for the normal equations). Table 2.5 shows the algorithm.

**Input**

$\mathbf{A}$, an unsymmetric matrix
$m$, the size of the Krylov subspace
$\underline{v}_1$, the initial vector for Krylov subspace

**Output**

$\lambda_i$, the approximated eigenvalues
$\underline{y}_i$, the approximated eigenvectors

$\underline{v}_1 = \underline{v}_1 / \parallel \underline{v}_1 \parallel_2$;
**For** $j := 1$ **To** $m$ **Do**
**Begin**
$\underline{w} = \mathbf{A} \cdot \underline{v}_j$;
**For** $i := 1$ **To** $j$ **Do**
**Begin**
$h_{ij} = \underline{v}_i^T \cdot \underline{w}$;
$\underline{w} = \underline{w} - h_{ij}\underline{v}_i$;
**End**;
$h_{j+1,j} = \parallel \underline{w} \parallel_2$;
$\underline{v}_{j+1} = \underline{w}/h_{j+1,j}$;
**End**;
**Solve** $(\mathbf{H} \cdot \underline{s}_i = \lambda_i \underline{s}_i)$;
$\underline{y}_i = \mathbf{V} \cdot \underline{s}_i$;

Table 2.4: *Algorithm : Arnoldi eigensystem approximation of matrix* $\mathbf{A}$

### The biconjugate gradient method (BCG)

The BCG algorithm may be viewed as being related to GMRES, for both build approximations in Krylov subspace. The BCG approximation is computed from simple three-term recurrence relations and thus requires much less work than GMRES. However, the approximation computed by BCG do not satisfy any optimality property. Table 2.6 shows the description of the BCG algorithm.

## 2.2 Transient analysis of Markov chains

This section focuses on algorithms which compute the *transient distribution* of a homogeneous Markov chain with finite states. This section based on [Ste94].

The transient behavior of a *CTMC* can be described by the following system of differential equations.

$$\frac{d\underline{P}^T(t)}{dt} = \underline{P}^T(t) \cdot \mathbf{Q} \tag{2.2}$$

The solution of (2.2) has an explicit form $\underline{P}^T(t) = \underline{P}^T(0) \cdot e^{\mathbf{Q}t}$. Consequently, we concentrate on the calculation of the solution numerically.

For performing transient analysis we can choose mainly from three types of methods. Methods from the first type are based on available *differential equation solvers*. Methods from the second type use the advantages of *randomization* and methods from the third type follow a relatively novel approach of using *Krylov subspace*.

**Input**

$\mathbf{A}$, an unsymmetric matrix
$\underline{b}$, a column vector
$\underline{x}_0$, an initial approximated solution
$\varepsilon$, the required precision

**Output**

$\underline{x}$, the approximated solution

$\underline{r}_0 := \underline{b} - \mathbf{A} \cdot \underline{x}_0$;
$\underline{v}_0 := \underline{0}$;
$\beta_0 := 0$;
$j := 0$;
**While** $\underline{r}_j^T \cdot \underline{r}_j > \varepsilon$ **Do**
**Begin**
$j := j + 1$;
$\underline{v}_j := \mathbf{A}^T \cdot \underline{r}_{j-1} + \beta_{j-1}\ \underline{v}_{j-1}$;
$\alpha_j := \underline{r}_{j-1}^T \cdot \underline{r}_{j-1} / \underline{v}_j^T \cdot \underline{v}_j$;
$\underline{x}_j := \underline{x}_{j-1} + \alpha_j\ \underline{v}_j$;
$\underline{r}_j := \underline{r}_{j-1} - \alpha_j \cdot \mathbf{A} \cdot \underline{v}_j$;
$\beta_j := \underline{r}_j^T \cdot \underline{r}_j / \underline{r}_{j-1}^T \cdot \underline{r}_{j-1}$;
**End**;

Table 2.5: *Algorithm : Conjugate gradient method for normal linear system*

### 2.2.1 Ordinary differential equation solvers

The solution of ordinary differential equations (ODEs), in our context (2.2), has been a subject of extensive research, and it is therefore appropriate for us to examine the possibilities of applying existing ODE solving techniques to the determination of transient solutions of *CTMC*s. An immediate advantage of such an approach is that numerical methods for the solution of ODEs are applicable to analyze nonhomogeneous CTCMs, i.e. *CTMC*s whose generator matrices are functions of time, i.e., $\mathbf{Q}(t)$. The available literature on the numerical solution of differential equations is vast, e.g. [HNW87, HW91].

### 2.2.2 The randomization method

The randomization method provides a stable way to calculate $\underline{P}^T(t) = \underline{P}^T(0) \cdot e^{\mathbf{Q}t}$, where the row vector $\underline{P}^T(t)$ is the distribution of the considered *CTMC*, $\mathcal{Z}(t)$, at time $t$ when the process $\mathcal{Z}(t)$ is defined with the generator matrix $\mathbf{Q}$ and the initial distribution $\underline{P}(0)$. One can introduce a new matrix $\mathbf{A}$ as $\mathbf{Q}/q + \mathbf{I}$. Applying this notation

$$\underline{P}^T(t) = \underline{P}^T(0) \cdot e^{\mathbf{Q}t} = \underline{P}^T(0) \cdot e^{(\mathbf{A}-\mathbf{I})qt} = \underline{P}^T(0) \cdot e^{\mathbf{A}qt}\ e^{-qt} \quad .$$

Then using Taylor series of the matrix exponential results in

$$\underline{P}^T(t) = \underline{P}^T(0) \cdot \sum_{i=0}^{\infty} \mathbf{A}^i\ \frac{(qt)^i}{i!}\ e^{-qt} \quad .$$

If $q \geq \max|q_{ii}|$ then the matrix $\mathbf{A}$ is a stochastic matrix. The randomization method will be used in the next chapters where it is described in detail including global error bounds and implementation aspects.

**Input**

$\mathbf{A}$, an unsymmetric matrix
$\underline{b}$, a column vector
$\underline{x}_0$, a initial approximated solution
$\varepsilon$, the required precision

**Output**

$\underline{x}$, the approximated solution

$\underline{r}_0 := \underline{b} - \mathbf{A} \cdot \underline{x}_0$;
$\underline{s}_0 := \underline{r}_0$;
$\underline{v}_0 := \underline{r}_0$;
$\underline{w}_0 := \underline{r}_0$;
$j := 0$;
**While** $\underline{r}_j^T \cdot \underline{r}_j > \varepsilon$ **Do**
**Begin**
$j := j + 1$;
$\alpha_j := \underline{s}_{j-1}^T \cdot \underline{r}_{j-1} / \underline{w}_{j-1}^T \cdot \mathbf{A} \cdot \underline{v}_{j-1}$;
$\underline{r}_j := \underline{r}_{j-1} - \alpha_j \cdot \mathbf{A} \cdot \underline{v}_{j-1}$;
$\underline{s}_j := \underline{s}_{j-1} - \alpha_j \cdot \mathbf{A}^T \cdot \underline{s}_{j-1}$;
$\underline{x}_j := \underline{x}_{j-1} - \alpha_j \underline{v}_{j-1}$;
$\beta_j := \underline{s}_j^T \cdot \underline{r}_j / \underline{s}_{j-1}^T \cdot \underline{v}_{j-1}$;
$\underline{v}_j := \underline{r}_j + \beta_j \underline{v}_{j-1}$;
$\underline{w}_j := \underline{s}_j + \beta_j \underline{w}_{j-1}$;
**End**;

Table 2.6: *Algorithm : Biconjugate gradient method for normal linear system*

### 2.2.3 Krylov subspace approach

The objective of Krylov subspace approach is the computation of an optimal approximation of a vector function. The approximation of the vector function has the form $p_{m-1}(\mathbf{A}) \cdot \underline{v}$, where $p_{m-1}(\mathbf{A})$ is a polynomial of degree $m-1$ in $\mathbf{A}$.

Note that, if the matrix $\mathbf{A}$ is $\mathbf{Q}^T t$, the row vector $\underline{v}^T$ is $\underline{P}^T(0)$ and row vector $\underline{w}^T$ equals $\underline{P}^T(t)$ then $\underline{P}(t) = \underline{P}(0) \cdot e^{\mathbf{Q}t}$ is equivalent to $\underline{w} = e^{\mathbf{A}} \cdot \underline{v}$. One can recognize that the matrix polynomial $p_{m-1}(\mathbf{A})$ is an element of Krylov subspace of $\mathbf{A}$ and $\underline{v}$, i.e. $p_{m-1}(\mathbf{A}) \in \mathcal{K}_m(\mathbf{A}, \underline{v})$. So our problem may be posed as that of finding the element of the considered Krylov subspace which best approximates the vector $e^{\mathbf{A}} \cdot \underline{v}$. If the "best" is taken to mean in the least squares sense, then the best approximation $\underline{\hat{w}}$ of $\underline{w}$ in the Krylov subspace satisfies

$$\| \underline{w} - \underline{\hat{w}} \|_2 = \min_{\underline{x} \in \mathcal{K}_m(\mathbf{A}, \underline{v})} \| \underline{x} - \underline{w} \|_2 = \min_{\underline{y} \in \mathbb{R}^m} \| \mathbf{B}_m \cdot \underline{y} - e^{\mathbf{A}} \cdot \underline{v} \|_2$$

where $\mathbf{B}_m = \{\underline{b}_1, \underline{b}_2, \ldots, \underline{b}_m\}$ is a set of orthonormal basis vectors for $\mathcal{K}_m(\mathbf{A}, \underline{v})$.

Because the vectors span Krylov subspace are not orthonormal we need to construct an orthonormal basis $\mathbf{B}_m$, we can use Gram-Schmidt method for doing that. With the matrix $\mathbf{B}_m$ in the hand we can choose $\underline{b}_1 = \underline{v} / \| \underline{v} \|_2$ and write identity matrix as $\mathbf{I} = \mathbf{B}_m \cdot \mathbf{B}_m^T$ and then

$$e^{\mathbf{A}} \cdot \underline{v} = \mathbf{B}_m \cdot \mathbf{B}_m^T \cdot e^{\mathbf{A}} \cdot \underline{v} = \mathbf{B}_m \cdot \left[ \mathbf{B}_m^T \cdot e^{\mathbf{A}} \cdot \mathbf{B}_m \right] \cdot \| \underline{v} \|_2 \cdot \underline{e}_1$$

where $\underline{e}_1$ is the unit vector $\{1, 0, 0, \ldots, 0\}$. The purpose of Krylov subspace approach, namely to project the exponential of a large matrix approximately onto a small Krylov subspace, is accomplished by approximating $\mathbf{B}_m^T \cdot e^{\mathbf{A}} \cdot \mathbf{B}_m$. This gives the approximation as

$$e^{\mathbf{A}} \cdot \underline{v} \approx \| \underline{v} \|_2 \, \mathbf{B}_m \cdot e^{\mathbf{H}_m} \cdot \underline{e}_1$$

which still involves the evaluation of the exponential of a matrix, but this time of small dimension $m$ of a particular structure, namely upper Hessenberg $\mathbf{B}_m$ and $\mathbf{H}_m$ can be computed by Arnoldi algorithm. Y. Saad [Saa92] shows that an a priori error bounds exists for the procedure

$$\| (e^{\mathbf{A}} \cdot \underline{v}) - (\| \underline{v} \|_2 \, \mathbf{B}_m \cdot e^{\mathbf{H}_m} \cdot \underline{e}_1) \|_2 \quad \leq \quad 2 \| \underline{v} \|_2 \, \frac{(\| \mathbf{A} \|_2)^m \cdot e^{\|\mathbf{A}\|_2}}{m!} \quad .$$

## 2.3 Definition of Markov reward models

Two objects determine a Markov reward model (MRM), which are *the underlying CTMC* and *the reward structure* of the model (see e.g. [TPF98]).

> The *underlying CTMC*: Let the stochastic process $\{\mathcal{Z}(t),\ t \geq 0\}$ be a continuous time Markov chain over the finite state space $\mathcal{S} = \{1, 2, \ldots, M\}$ with the generator matrix $\mathbf{Q} = [q_{ij}]$ and the initial distribution $\underline{P} = [p_i]$.
>
> The *reward structure*: A non-negative real constant $r_i$, called *rate reward*, is associated with each state of the underlying CTMC representing the reward rate in the state $i$. Let $\mathbf{R}$ be the diagonal matrix of the reward rates, i.e., $\mathbf{R} = \mathrm{diag}(r_1, r_2, \ldots, r_M)$. A non-negative real random variable $\mathcal{D}_{ij}$, called *impulse reward*, is associated with each possible state transition of the underlying *CTMC* representing the amount of reward gained at every state transition from the state $i$ to the state $j$.

I introduce Laplace transform description of the reward structure, because it simplifies the further model analysis. Let $D_{ij}(w)$ be the distribution (i.e., $D_{ij}(w) = Pr\{\mathcal{D}_{ij} \leq w\}$) and $D^{\sim}_{ij}(v)$ be the Laplace-Stieltjes transform (i.e., $D^{\sim}_{ij}(v) = \int_0^{\infty} e^{-vw} dD_{ij}(w)$) of the random variable $\mathcal{D}_{ij}$. The associated matrix is $\mathbf{D}^{\sim}(v) = \left[D^{\sim}_{ij}(v)\right]$ and the matrices of the moments of impulse rewards are

$$\mathbf{D}^{(n)} = \left[E(\mathcal{D}^n_{ij})\right] = (-1)^n \frac{\partial^n \mathbf{D}^{\sim}(v)}{\partial v^n}\bigg|_{v=0} \quad .$$

If there is no impulse reward associated with the state transition from the state $i$ to the state $j$ then $D_{ij}(w) = \mathrm{u}[w]$ and $D^{\sim}_{ij}(v) = 1$. The diagonal elements are defined similarly $D_{ii}(w) = \mathrm{u}[w]$ and $D^{\sim}_{ii}(v) = 1$.

There are two main reward measures defined in a Markov reward model, which are *the accumulated reward* and *the completion time*.

**Definition 1** *The* **accumulated reward**, $\mathcal{B}(t)$, *is the random variable which describes the accumulation of reward in time*

$$\mathcal{B}(t) = \int_0^t \left(r_{\mathcal{Z}(\tau)} + \delta_\tau \mathcal{D}_{\mathcal{Z}(\tau^-),\mathcal{Z}(\tau)}\right) d\tau \quad .$$

By this definition, $\mathcal{B}(t)$ is a stochastic process that depends on $\mathcal{Z}(u)$ for $0 \leq u \leq t$; and without any restriction we will assume $\mathcal{B}(0) = 0$. According to the Definition 1 this work restricts the attention to the class of models in which no state transition can entail to a loss of

the accumulated reward. This kind of accumulation is also referred to as preemptive resume. The distribution of the accumulated reward is defined by

$$B(t,w) = Pr\{\mathcal{B}(t) \leq w\} \qquad \text{and} \qquad B_i(t,w) = Pr\{\mathcal{B}(t) \leq w \mid \mathcal{Z}(0) = i\} \ .$$

Note that

$$B(t,w) = \sum_{i \in S} p_i \ B_i(t,w) = \underline{P}^T \cdot \underline{B}(t,w),$$

hence, in the rest of this work, we will use the initial state dependent measures; and the global measures can always be evaluated by the mean of this relation. Because algorithms which provide moments of reward measures based on their Laplace transform domain descriptions will be presented the following notations are introduced

$$\underline{B}^{\sim}(t,v) = \left[ \int_0^\infty e^{-wv} dB_i(t,w) \right] \quad \text{and} \quad \underline{m}^{(n)}(t) = \int_0^\infty w^n d\underline{B}(t,w) = (-1)^n \frac{\partial^n \underline{B}^{\sim}(t,v)}{\partial v^n}\Big|_{v=0} \ .$$

The complementary reward measure of the accumulated reward is the completion time. The completion time is defined only when all rate and impulse rewards are non-negative.

**Definition 2** *The* **completion time**, $\mathcal{C}$, *is the random variable which describes the time to accumulate the random amount of reward* $\mathcal{W}$

$$\mathcal{C} = \mathbf{min}[\ t \ : \ \mathcal{B}(t) \geq \mathcal{W}] \ .$$

The distribution of $\mathcal{C}$ is defined as

$$C(t) = Pr\{\mathcal{C} \leq t\} \ .$$

Let $\mathcal{C}(w)$ be the random variable which describes the time to accumulate $w$ (fix) amount of reward and $C(t,w)$ its distribution, i.e.,

$$\mathcal{C}(w) = \mathbf{min}[\ t \ : \ \mathcal{B}(t) \geq w] \qquad \text{and} \qquad C_i(t,w) = Pr\{\mathcal{C}(w) \leq t \mid \mathcal{Z}(0) = i\} \ .$$

Let $G(w)$ be the distribution of the random variable $\mathcal{W}$ with support on $[0,\infty)$. By the Definition 2,

$$C_i(t) = \int_0^\infty C_i(t,w) \ dG(w) \ . \tag{2.3}$$

The Laplace transform and moments of the completion time are defined as

$$\underline{C}^{\sim}(s,w) = \left[ \int_0^\infty e^{-ts} dC_i(t,w) \right] \quad \text{and} \quad \underline{s}^{(n)}(w) = \int_0^\infty t^n d\underline{C}(t,w) = (-1)^n \frac{\partial^n \underline{C}^{\sim}(s,w)}{\partial s^n}\Big|_{s=0} \ .$$

The distribution of the completion time is closely related to the distribution of the accumulated reward by the mean of the following relation (see Figure 2.1.)

$$B_i(t,w) = Pr\{\mathcal{B}(t) \leq w \mid \mathcal{Z}(0) = i\} = Pr\{\mathcal{C}(w) \geq t \mid \mathcal{Z}(0) = i\} = 1 - C_i(t,w) \ . \tag{2.4}$$

Note that this relation is valid for only those models where all rate and impulse rewards are non-negative.

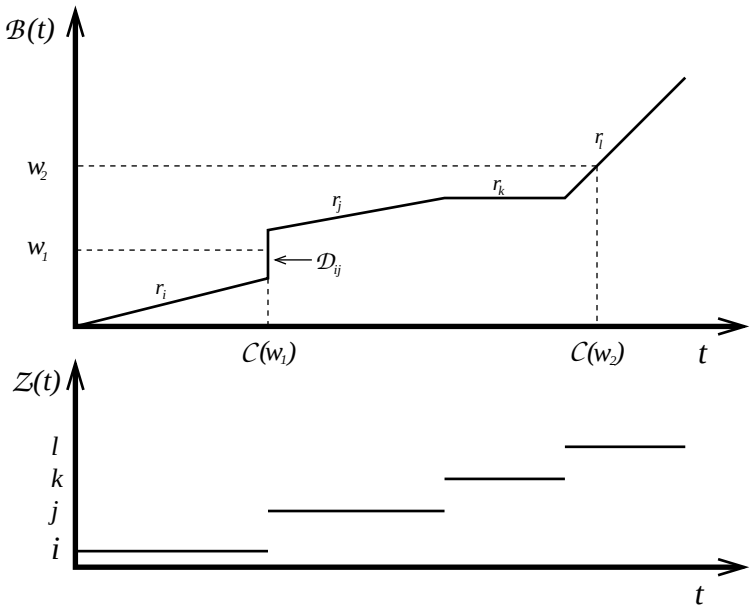



Figure 2.1: *A sample path of the underlying CTMC and the accumulated reward*

## 2.4 Steady-state analysis of Markov reward models

Basically, there are mainly two types of *MRM*s where steady-state analysis can be considered.

- *MRM*s with absorbing states where rate rewards are zeros; we can consider the accumulated reward, i.e., $\mathcal{B}(t)$;
- *MRM*s without absorbing states; we can consider the normalized accumulated reward, i.e., $\mathcal{B}(t)/t$ .

The first case requires the transient analysis of the MRM. The other, where the considered steady-state reward measure is $g = \lim_{t\to\infty} \mathcal{B}(t)/t$, does not require transient analysis. From the steady-state distribution ($\underline{P}_{SS}$) of the underlying *CTMC* we can obtain $g$ using the following relation.

$$g = \underline{P}_{SS}^T \cdot \mathbf{R} \cdot \underline{h} \quad .$$

## 2.5 Transient analysis of Markov reward models

This section provides a comprehensive review of the literature related to the transient analysis of *MRM*s.

The transient distribution of reward measures can be described with a set of partial differential equations. The transient behavior of the accumulated reward fulfills the following system of differential equations

$$\frac{\partial \underline{B}(t,w)}{\partial t} + \mathbf{R} \cdot \frac{\partial \underline{B}(t,w)}{\partial w} = \mathbf{Q} \odot \mathbf{D}(w) \cdot \underline{B}(t,w) \tag{2.5}$$

where $\odot$ denotes the piecewise matrix multiplication ( $[\mathbf{A} \odot \mathbf{B}]_{ij} = a_{ij} \cdot b_{ij}$) and the initial condition can be derived from the model assumptions as

$$\underline{B}(0,w) = u[w] \quad \text{and} \quad \underline{B}(t,0) = 0 \tag{2.6}$$

assuming that all reward rates are positive.

The double Laplace-Stieltjes transform of the considered differential equations with the given initial condition has the following form.

$$\underline{B}^{\sim\sim}(s,v) = s(s\mathbf{I} + v\mathbf{R} - \mathbf{Q} \odot \mathbf{D}^{\sim}(v))^{-1} \cdot \underline{h}$$

The lack of the explicit solution of (2.5) and (2.6) initiated strong research activity which has been put on the determination of $\underline{B}(t,w)$ numerically. Among several methods the most efficient methods are based on Laplace transform, randomization or direct probabilistic approaches.

Table 2.7 compares the available algorithms with respect to their computational efforts and memory requirements. My aim was to propose a new algorithm which has lower computational complexity. Table 2.7 also summarizes main properties of the algorithms. I used the complexity of the *CTMC* transient analysis as a reference point. In the table $t$ refers to the time point of the analysis, $M$ is the cardinality of the state space, $T$ is the number of state transitions, $K$ is the number of different reward rates and $n$ is the number of computed moments.

| Method | CPU time | memory | output |
|---|---|---|---|
| Iyer, Donatiello and Heidelberger [IDH86] | $\mathcal{O}(M^4 \cdot n^2)$ | n.a. | moments |
| Smith, Trivedi and Ramesh [STR88] | $\mathcal{O}(M^3)$ | n.a. | distr. |
| Donatiello and Grassi [DG91] | $\mathcal{O}(T \cdot K \cdot t^2)$ | $\mathcal{O}(K \cdot M \cdot t)$ | distr. |
| Nabli and Sericola [NS96] | $\mathcal{O}(T \cdot K \cdot t^2)$ | $\mathcal{O}(K \cdot M \cdot t)$ | distr. |
| Silva and Gail [SG98] | $\mathcal{O}(T \cdot t^2)$ | $\mathcal{O}(M \cdot t^2)$ | distr. |
| **Proposed new method** | $\mathcal{O}(T \cdot n \cdot t)$ | $\mathcal{O}(M \cdot n)$ | moments |
| *CTMC* transient analysis | $\mathcal{O}(T \cdot t)$ | $\mathcal{O}(M)$ | |

Table 2.7: *Complexity of numerical analysis methods of MRMs*

I provide the detailed description of the considered methods in the following subsections.

### 2.5.1 Method from Iyer, Donatiello and Heidelberger

In the paper [IDH86] authors obtained the double Laplace transform of performability (accumulated reward) for systems that may be modelled by semi-Markov processes. For the special case of Markovian models they inverted the transform in the formal parameter. To get the distribution from here, standard numerical Laplace transform inversion techniques outlined in the paper could be used. For Markovian models they obtained a recursion relating the $(n+1)$-th moment to the $n$-th moment. Further, they gave expressions for all moments and provided recursion for computing the coefficients involved in those expressions.

The first theorem of the paper provides the relation among the moments of the accumulated reward as

$$\underline{m}^{(n+1)}(t) = (n+1) \int_0^t e^{\mathbf{Q}(t-\tau)} \cdot \mathbf{R} \cdot \underline{m}^{(n)}(\tau)\, d\tau \quad .$$

Before presenting their recursive computational formula for $\underline{m}^{(n)}(t)$ they described a spectral representation method for the matrix $e^{\mathbf{Q}t}$. Let $\lambda_1 = 0, \lambda_2, \ldots, \lambda_M$ be the eigenvalues of the

generator matrix $\mathbf{Q}$. They assumed that all of the eigenvalues of $\mathbf{Q}$ are distinct. Then

$$e^{\mathbf{Q}t} = \sum_{i=1}^{M} \mathbf{Z}_i \; e^{\lambda_i t} \qquad \text{where} \qquad \mathbf{Z}_i = \frac{\prod_{l=2}^{M} (\mathbf{Q} - \lambda_l \mathbf{I})}{\prod_{k=1, k\neq i}^{M} (\lambda_i - \lambda_k)} \quad .$$

Using this spectral representation the other theorem of the paper results in the following formula

$$\underline{m}^{(n)}(t) = \sum_{i=1}^{M} \sum_{j=0}^{n} \underline{v}_n(i,j) \; e^{\lambda_i t} \; t^j$$

where the coefficient vector $\underline{v}_n(i,j)$ depends on the matrix $\mathbf{Z}_i$ and the diagonal reward matrix $\mathbf{R}$. This theorem is the main result of the paper.

Exclusive of eigenvalue and eigenvector computations, which is a computational intensive task, the time complexity to compute first $n$ moments of the accumulated reward is upper bounded by $\mathcal{O}(M^4 n^2)$.

### 2.5.2 Method from Smith, Trivedi and Ramesh

In the paper [STR88] authors described an $\mathcal{O}(M^3)$ algorithm for computing the distribution of the accumulated reward, where $M$ is the number of states in the considered Markov reward model.

The described method is based on Lapalce transform domain description of the accumulated reward that is given as

$$\underline{B}^{*\sim}(s,v) = (s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \underline{h} \quad .$$

They used Cramer's rule to formulate $\underline{B}^{*\sim}(s,v)$ as a rational function in $s$. When the underlying *CTMC* is starting from state $i$ then

$$B_i^{*\sim}(s,v) = \sum_{j=1}^{d} \sum_{k=1}^{m_j} a_{ijk}(v)(s - \lambda_j(v))^{-k} \tag{2.7}$$

where the $\lambda_j(v)\,, \quad j = 1, 2, \ldots, d$ are the $d$ distinct eigenvalues of the matrix $[\mathbf{Q} - v\mathbf{R}]$, each with algebraic multiplicity $m_j$. The QR algorithm was used to numerically determine eigenvalues of the matrix $[\mathbf{Q} - v\mathbf{R}]$ in $\mathcal{O}(M^3)$ time. Using (2.7) they inverted $B_i^{*\sim}(s,v)$ analytically with respect to $s$ and obtained

$$B_i^{\sim}(t,v) = \sum_{j=1}^{d} \sum_{k=1}^{m_j} \frac{a_{ijk}(v)}{(k-1)!} \; t^{k-1} e^{\lambda_j(v)t} \quad .$$

It remained to invert $B_i^{\sim}(t,v)$ respect to $v$. The number of methods to numerically invert Laplace transform have been developed. To avoid unnecessary notational complexity, they defined $V^{\sim}(v) = B_i^{\sim}(t,v)/v$. Then, they employed the following method to numerically obtain $V(w)$, the inverse Laplace transform of $V^{\sim}(v)$, using the well-known complex version formula

$$V(w) = \frac{e^{aw}}{\pi} \int_0^{\infty} \Re\{V^{\sim}(u) e^{iwx}\} \; dx$$

where $u = a + ix$. They used trapezoidal rule and Fourier series approximation to evaluate approximation of $V(w)$ numerically.

The time complexity to compute the approximated distribution of the accumulated reward is upper bounded by $\mathcal{O}(M^3)$.

### 2.5.3 Method from Donatiello and Grassi

In the paper [DG91] authors described an algorithm for computing the distribution of the accumulated reward. The algorithm determines the distribution at a given point $(t_1, w_1)$, i.e. determines $B_i(t_1, w_1)$.

The proposed method is based on the concept of randomization. They studied the randomized Markov reward process (i.e. $\mathcal{X}'(n)\ , \quad n = 1, 2, 3, \ldots$ a Markov chain with one-step transition matrix $\mathbf{P} = \mathbf{Q}/q + \mathbf{I}$ and a Poisson process $\mathcal{N}(t)$ with parameter $q$). They defined the following probability

$$B_i(t, w \mid k) = Pr\{\mathcal{B}_i(t) \leq w \mid \mathcal{N}(t) = k\}$$

this is the accumulated reward distribution conditioned on the occurrence of $k$ transitions starting the process from the state $i$. Consequently, we can write the unconditioned distribution as

$$B_i(t, w) = \sum_{k=0}^{\infty} B_i(t, w \mid k)\ Pr\{\mathcal{N}(t) = k\} \quad .$$

To evaluate $B_i(t, w)$ they needed determining $Pr\{\mathcal{N}(t) = k\}$ and $B_i(t, w \mid k)$. Obviously, for the former

$$Pr\{\mathcal{N}(t) = k\} = \frac{(qt)^k}{k!} e^{-qt}$$

while an explicit form for the latter is given by the following relation which is the main result of the paper (in this version of the relation it is assumed all rate rewards are different)

$$B_i(t, w \mid k) = \alpha_i^{(k)} u[w - r_i t] + \sum_{h=1}^{k} \sum_{j=1}^{M} \binom{k}{j-1}\ \beta_i^{(k)}(j, h)\ \left(\frac{w - r_j t}{t}\right)^{k-h+1} u[w - r_j t]$$

where coefficients $\alpha_i^{(k)}$ and $\beta_i^{(k)}(j, h)$ are independent of $t$ and $w$. They presented a recursive expression for these coefficients. The time complexity of the proposed algorithm is $\mathcal{O}(KTt^2)$ and the required memory is $\mathcal{O}(KMt)$.

### 2.5.4 Method from Nabli and Sericola

In the paper [NS96] authors described an algorithm for computing the distribution of the accumulated reward. The algorithm determines the distribution at a given point $(t_1, w_1)$, i.e. determines $B_i(t_1, w_1)$.

The proposed method is based on the concept of randomization. They studied the randomized Markov reward process. The main theorem of their paper has the following form

$$B_i(t, w) = 1 - \left[ \sum_{n=0}^{\infty} \frac{(qt)^n}{n!} e^{-qt} \sum_{k=0}^{n} \sum_{j=1}^{K} \binom{n}{k}\ s_j^k\ (1 - s_j)^{n-k}\ b^{(j)}(n, k) \mathbb{I}_{\{r_{j-1} t \leq w < r_j t\}} \right]$$

where $s_j = \dfrac{w - r_{j-1}t}{(r_j - r_{j-1})t}$ and the coefficient $b^{(j)}(n,k)$ is given by a special recursive expression. Since different states may have the same reward rate they denoted the $K+1$ different reward rates with $r_0 < r_1 < \ldots < r_K$. The time complexity of the proposed algorithm is $\mathcal{O}(KTt^2)$ and the required memory is $\mathcal{O}(KMt)$.

### 2.5.5 Method from Silva and Gail

In the paper [SG98] authors described an algorithm for computing the distribution of the accumulated reward. Their proposed algorithm can also deal with *MRM*s with rate and impulse reward. The algorithm determines the distribution at a given point $(t_1, w_1)$, i.e. determines $B(t_1, w_1)$.

They applied the randomization technique to the determination of the distribution of $ACIR(t) = \mathcal{B}(t)/t$ which is the distribution of the accumulated reward averaged over $t$ when both impulse and rate based rewards are present. Since different states may have the same reward rate, they denoted the $K+1$ different reward rates with $r_1 < r_2 < \ldots < r_{K+1}$. Given $n$ transitions recall that the interval $(0,t)$ is split into $n+1$ sub-intervals and they assigned a reward rate to each of them. Let $\mathbf{k} =< k_1, k_2, \ldots, k_{K+1} >$ be a vector whose $i$-th component represents the number of the intervals in $(0,t)$ associated with reward rate $r_i$ and define $\| \mathbf{k} \| = k_1 + k_2 + \ldots + k_{K+1}$. A specific vector $\mathbf{k}$ is referred to as a rate coloring. Similarly, an impulse coloring is a vector $\hat{\mathbf{k}} =< \hat{k}_1, \ldots, \hat{k}_{\hat{K}+1} >$ whose $i$-th component represents the number of transitions with impulse reward $\hat{r}_i$.

For $n = 0, 1, 2, \ldots$ let $K_n = \{\mathbf{k} \ : \ \| \mathbf{k} \| = n+1\}$ be the set of rate colorings corresponding the case of $n$ transitions and similarly let $\hat{K}_n = \{\hat{\mathbf{k}} \ : \ \| \hat{\mathbf{k}} \| = n\}$ be the corresponding set of impulse colorings. Define $L_n = \{\hat{r}(\hat{\mathbf{k}}) \ : \ \hat{\mathbf{k}} \in \hat{K}_n\}$ to be the set of impulse rewards that can be gained from $n$ transitions. Then, the distribution of the accumulated reward averaged over $t$ is given as follows

$$Pr\{ACIR(t) < w\} = \sum_{n=0}^{\infty} \frac{(qt)^n}{n!} e^{-qt} \sum_{\hat{\mathbf{r}} \in L_n} \sum_{\mathbf{k} \in K_n} \Theta(n, \mathbf{k}, \hat{r}) \ Pr\{ACIR(t) \leq w \mid n, \mathbf{k}, \hat{r}\}$$

where $\Theta(n, \mathbf{k}, \hat{r})$ is the probability, given $n$ transitions, of a rate coloring $\mathbf{k}$ and an average accumulated impulse reward of $\hat{r} = \sum_{i=1}^{\hat{K}+1} \hat{r}_i \hat{k}_i$. The time complexity of the proposed algorithm is $\mathcal{O}(Tt^2)$ and the required memory is $\mathcal{O}(Mt^2)$ without applying impulse rewards.

## 2.6 Conclusion

This chapter gave a brief insight into numerical techniques of Markovian models. Only homogeneous Markovian models with finite states was considered. Handling nonhomogeneous and/or infinity states models requires different solution approaches [Neu81, Neu89].

Section 2.1 focused on the steady-state analysis of *CTMC*s and presented three main types of methods (direct methods, iterative methods and projection methods). Section 2.2 gave a very short overview on the transient analysis of *CTMC*s and described the main properties of three solution approaches (ordinari differential equation solvers, randomization method and Krylov subspace approaches). Section 2.3 introduced the *Markov reward models* and the considered two reward measures (accumulated reward and completion time). Section 2.4 examined the possibility of steady-state analysis of *MRM*s. Finally, Section 2.5 provided a comprehensive review of the *MRM*s' transient analysis.

# Chapter 3

# Transient analysis of Markov reward models with only rate reward

The goal of this chapter is to describe my *results in MRMs analysis*. I have published this work in [J3, C2, C10]. I have made a study of *MRM*s which do not use the modelling power of impulse reward. Early studies focused on only this type of *MRM*s, because considering impulse reward increases the complexity of the analysis. So my aim was to develop a *stable numerical algorithm* for this type of *MRM*s. The proposed algorithm provides moments of the accumulated reward and the completion time.

## 3.1 Transform domain description of reward measures

This section presents closed-form expressions for the Laplace transform of the accumulated reward and the completion time derived from using a new approach. Detailed derivations in [IDH86, KNT86] resulted in the same expressions. I have published the following theorem with its proof in [J3].

**Theorem 3.1** *Laplace transform of the accumulated reward is as follows*

$$\underline{B}^{\sim}(t,v) = e^{(\mathbf{Q}-v\mathbf{R})t} \cdot \underline{h} \quad . \tag{3.1}$$

*Proof:* Consider an exponentially distributed work requirement, $\mathcal{W}$ with parameter $m$, which is independent from the underlying *CTMC*.

On the one hand, the completion time of the work requirement is characterized by the following distribution function

$$\begin{aligned} C_i(t) &= \int_0^\infty C_i(t,w)\, dG(w) = \int_0^\infty \Big(1 - B_i(t,w)\Big)\, dG(w) \\ &= m \int_0^\infty \Big(1 - B_i(t,x)\Big) e^{-mx}\, dx = 1 - B_i^{\sim}(t,v)\Big|_{v=m} \end{aligned}$$

which, in vector form, is

$$\underline{C}(t) = \underline{h} - \underline{B}^{\sim}(t,v)\Big|_{v=m} \; .$$

Note that, $m$ is a non-negative real variable and $v$ is a complex variable.

On the other hand, $C_i(t)$ is phase type distributed and its distribution can be obtained by the representation of the phase type distribution [BT90] as

$$\underline{C}(t) = \underline{h} - e^{(\mathbf{Q}-m\mathbf{R})t} \cdot \underline{h} \quad . \tag{3.2}$$

And since (3.1) is an analytical complex function for $\Re(v) \geq 0$ the theorem is given. □

Some further Laplace transforms of (3.1) with respect to $t \to s$ and $w \to v$ are

$$\begin{aligned} \underline{B}^{\sim\sim}(s,v) &= s(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \underline{h} \ , \\ \underline{B}^{\sim *}(s,v) &= \frac{s}{v}(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \underline{h} \ , \\ \underline{B}^{*\sim}(s,v) &= (s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \underline{h} \ , \\ \underline{B}^{**}(s,v) &= \frac{1}{v}(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \underline{h} \quad . \end{aligned} \tag{3.3}$$

Detailed derivations in [IDH86] resulted in the same expression as (3.3) for the distribution of the accumulated reward in Laplace transform domain based on a different approach.

Laplace transform of the completion time can be derived from Laplace transform of the accumulated reward. From (2.4), (3.3) and using the fact $\mathbf{Q} \cdot \underline{h} = \underline{0}$ we have

$$\begin{aligned} \underline{C}^{\sim\sim}(s,v) &= \underline{h} - \underline{B}^{\sim\sim}(s,v) \\ &= [\mathbf{I} - s(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1}] \cdot \underline{h} \\ &= [(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot (s\mathbf{I} + v\mathbf{R} - \mathbf{Q}) - s(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1}] \cdot \underline{h} \\ &= (s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot (v\mathbf{R} - \mathbf{Q}) \cdot \underline{h} \\ &= v(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \mathbf{R} \cdot \underline{h} \end{aligned} \tag{3.4}$$

which was obtained through a different way of reasoning in [KNT86].

Some further Lapalce transforms of the completion time are

$$\begin{aligned} \underline{C}^{\sim\sim}(s,v) &= v(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \mathbf{R} \cdot \underline{h} \ , \\ \underline{C}^{\sim *}(s,v) &= (s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \mathbf{R} \cdot \underline{h} \ , \\ \underline{C}^{*\sim}(s,v) &= \frac{v}{s}(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \mathbf{R} \cdot \underline{h} \ , \\ \underline{C}^{**}(s,v) &= \frac{1}{s}(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot \mathbf{R} \cdot \underline{h} \quad . \end{aligned}$$

Suppose $\mathbf{R}^{-1}$ exists, i.e. $r_i > 0 \ \ \forall i$, equation (3.4) can be inverse transformed with respect to the reward variable as follows

$$\underline{C}^{\sim\sim}(s,v) = v(s\mathbf{I} + v\mathbf{R} - \mathbf{Q})^{-1} \cdot (\mathbf{R}^{-1})^{-1} \cdot \underline{h} = v(s\mathbf{R}^{-1} + v\mathbf{I} - \mathbf{R}^{-1}\mathbf{Q})^{-1} \cdot \underline{h} \ , \tag{3.5}$$

from which

| | System A | System B |
|---|---|---|
| generator matrix | $\mathbf{Q}$ | $\mathbf{R}^{-1} \cdot \mathbf{Q}$ |
| reward matrix | $\mathbf{R}$ | $\mathbf{R}^{-1}$ |
| initial distribution | $\underline{P}$ | $\underline{P}$ |
| $\underline{B}^{\sim}(t, v)$ | $e^{(\mathbf{Q}-v\mathbf{R})t} \cdot \underline{h}$ | $e^{(\mathbf{R}^{-1}\cdot\mathbf{Q}-v\mathbf{R}^{-1})t} \cdot \underline{h}$ |
| $\underline{C}^{\sim}(s, w)$ | $e^{(\mathbf{R}^{-1}\cdot\mathbf{Q}-s\mathbf{R}^{-1})w} \cdot \underline{h}$ | $e^{(\mathbf{Q}-s\mathbf{R})w} \cdot \underline{h}$ |
| time domain | $\underline{B}_A(t, w) = \underline{h} - \underline{C}_A(t, w)$ | $\underline{B}_B(t, w) = \underline{h} - \underline{C}_B(t, w)$ |

Table 3.1: *Dual Markov reward models*

$$\underline{C}^{\sim}(s, w) = e^{(\mathbf{R}^{-1}\mathbf{Q}-s\mathbf{R}^{-1})w} \cdot \underline{h} \; . \tag{3.6}$$

Note that, we did not restrict the class of *MRM*s till (3.5), hence the results are valid for any reducible and irreducible underlying *CTMC* and any non-negative reward rates. In (3.6), the only restriction is that $\mathbf{R}$ must be invertable, i.e., strictly positive reward rates are only allowed.

## 3.2 Duality of reward measures

The duality of the accumulated reward and the completion time comes from the close relation between them in time domain (2.4) as

$$\underline{B}(t, w) = \underline{h} - \underline{C}(t, w)$$

which is valid in Laplace transform domain as well.

$$\begin{aligned} \underline{B}^{\sim}(s, w) &= \underline{h} - \underline{C}^{\sim}(s, w) \\ \underline{B}^{\sim}(t, v) &= \underline{h} - \underline{C}^{\sim}(t, v) \\ \underline{B}^{\sim\sim}(s, v) &= \underline{h} - \underline{C}^{\sim\sim}(s, v) \end{aligned}$$

As we will see later, the single Laplace transform domain description of reward measures are used during the analysis, i.e. $\underline{B}^{\sim}(t, v)$ and $\underline{C}^{\sim}(s, w)$. Unfortunately, there is no direct relation between the two considered reward measures in single Laplace transform domain. But, we can construct dual systems as Table 3.1 shows.

We can recognize that the accumulated reward in System A is equivalent to the completion time in System B when the rule of the reward ($w$ or $v$) and the time ($t$ or $s$) are interchanged. This property is valid in time domain as well. This means that $\underline{B}_A(t, w) = \underline{C}_B(w, t)$ and $\underline{C}_A(t, w) = \underline{B}_B(w, t)$. So we will refer to the System B as the dual of the System A and vice versa. Consequently, assume that we model a system with a *MRM* and the interesting performance measure is the accumulated reward then there are two ways of analysis, i.e.,

- analyzing the original *MRM* with respect to the accumulated reward;
- analyzing the dual *MRM* with respect to the completion time.

## 3.3 Moments of the accumulated reward

The column vector $\underline{m}^{(n)}(t)$, which is the $n$-th moment of the reward accumulated in $[0,t)$, can be evaluated based on its Laplace transform domain description as

$$\underline{m}^{(n)}(t) = (-1)^n \frac{\partial^n \underline{B}^{\sim}(t,v)}{\partial v^n}\bigg|_{v=0} . \tag{3.7}$$

The following theorem provides an iterative method for evaluating (3.7). I have published the theorem with its proof in [J3].

**Theorem 3.2** *The n-th moment of the accumulated reward, $\underline{m}^{(n)}(t)$, satisfies the following equation*

$$\underline{m}^{(n)}(t) = (-1)^n \sum_{i=0}^{\infty} \frac{t^i}{i!} \mathbf{N}^{(n)}(i) \cdot \underline{h}$$

*where the coefficient matrix $\mathbf{N}^{(n)}(i)$ is defined as*

$$\mathbf{N}^{(n)}(i) = \begin{cases} \mathbf{I} , & \text{if} \quad i = 0, n = 0 , \\ \mathbf{0} , & \text{if} \quad i = 0, n \geq 1 , \\ \mathbf{Q}^i , & \text{if} \quad i \geq 1, n = 0 , \\ \mathbf{Q} \cdot \mathbf{N}^{(n)}(i-1) - n\ \mathbf{R} \cdot \mathbf{N}^{(n-1)}(i-1) , & \text{if} \quad i \geq 1, n \geq 1 . \end{cases}$$

*Proof:* From (3.1) and (3.7) the $n$-th moment is

$$\begin{aligned} \underline{m}^{(n)}(t) &= (-1)^n \frac{\partial^n e^{(\mathbf{Q}-v\mathbf{R})t}}{\partial v^n}\bigg|_{v=0} \cdot \underline{h} \\ &= (-1)^n \frac{\partial^n}{\partial v^n} \sum_{i=0}^{\infty} \frac{t^i}{i!} (\mathbf{Q}-v\mathbf{R})^i \bigg|_{v=0} \cdot \underline{h} \\ &= (-1)^n \sum_{i=0}^{\infty} \frac{t^i}{i!} \ \frac{\partial^n}{\partial v^n} (\mathbf{Q}-v\mathbf{R})^i \bigg|_{v=0} \cdot \underline{h} . \end{aligned}$$

To avoid unnecessary notational complexity the following notations introduced

$$\mathbf{N}^{(n)}(i) = \frac{\partial^n}{\partial v^n} (\mathbf{Q}-v\mathbf{R})^i \bigg|_{v=0} , \qquad \text{for } \forall n, i .$$

From Leibniz rule follows an iterative method for constructing the matrix $\mathbf{N}^{(n)}(i)$ as

$$\mathbf{N}^{(n)}(i) = \mathbf{Q} \cdot \mathbf{N}^{(n)}(i-1) - n\ \mathbf{R} \cdot \mathbf{N}^{(n-1)}(i-1),$$

with the initial conditions $\mathbf{N}^{(0)}(0) = \mathbf{I}$, $\mathbf{N}^{(0)}(i) = \mathbf{Q}^i$ and $\mathbf{N}^{(n)}(0) = \mathbf{0}$. □

The iterative method to evaluate the matrix $\mathbf{N}^{(n)}(i)$ has the following properties

- it is not possible to evaluate the $n$-th moment itself, but to obtain the $n$-th moment all the previous moments, or at least the associated $\mathbf{N}^{(n)}(i)$ terms, must be computed;
- matrix-matrix multiplications are computed in each iteration step;
- numerical problems are possible due to the repeated multiplication with matrix $\mathbf{Q}$, which contains both positive and negative elements, hence Theorem 3.2 is not directly applicable to numerical analysis. Section 3.5 will relax these problems.

## 3.4 Moments of the completion time

The method presented here is the dual version of the method derived for performing the accumulated reward analysis in Section 3.3.

The column vector $\underline{s}^{(n)}(w)$, which is the $n$-th moment of the completion time, can be evaluated based on $\underline{C}^{\sim}(s,w)$ as

$$\underline{s}^{(n)}(w) = (-1)^n \frac{\partial^n \underline{C}^{\sim}(s,w)}{\partial s^n}\bigg|_{s=0} . \tag{3.8}$$

I have published the following theorem with its proof in [J3].

**Theorem 3.3** *The $n$-th moment of the completion time, $\underline{s}^{(n)}(w)$, satisfies the following equation*

$$\underline{s}^{(n)}(w) = (-1)^n \sum_{i=0}^{\infty} \frac{w^i}{i!} \mathbf{M}^{(n)}(i) \cdot \underline{h}$$

*where the coefficient matrix $\mathbf{M}^{(n)}(i)$ is defined as*

$$\mathbf{M}^{(n)}(i) = \begin{cases} \mathbf{I}, & \text{if } i = 0, n = 0 , \\ \mathbf{0}, & \text{if } i = 0, n \geq 1 , \\ (\mathbf{R}^{-1} \cdot \mathbf{Q})^i, & \text{if } i \geq 1, n = 0 , \\ \mathbf{R}^{-1} \cdot \mathbf{Q} \cdot \mathbf{M}^{(n)}(i-1) - n\ \mathbf{R}^{-1} \cdot \mathbf{M}^{(n-1)}(i-1), & \text{if } i \geq 1, n \geq 1 . \end{cases}$$

*Proof:* Using

$$\underline{s}^{(n)}(w) = (-1)^n \frac{\partial^n}{\partial s^n} e^{(\mathbf{R}^{-1} \cdot \mathbf{Q} - s\mathbf{R}^{-1})w}\bigg|_{s=0} \cdot \underline{h}$$

the proof follows the same pattern as the proof of Theorem 3.2. □

This iterative method has the same properties as the one presented in Theorem 3.2. In contrast with Theorem 3.2, the application of Theorem 3.3 is restricted to *MRM*s with strictly positive reward rates.

### 3.4.1 System with zero reward rates

In case of some of the reward rates are zero the Theorem 3.3 can not be applied for computing moments of the completion time. This section gives a method which can handle this case as well.

Let us partition the state space $\mathcal{S}$ into two disjoint sets $\mathcal{S}_+$ and $\mathcal{S}_0$. $\mathcal{S}_+$ ($\mathcal{S}_0$) contains the states with associated positive (0) reward rate, i.e., $r_i > 0; \forall i \in \mathcal{S}_+$ and $r_i = 0; \forall i \in \mathcal{S}_0$. The accumulated reward does not increase during the sojourn in $\mathcal{S}_0$. If $\mathcal{S}_0$ has an absorbing subset then the distribution of the completion time is defective. In the subsequent analysis we do not allow this case.

Without loss of generality, we can number the states in $\mathcal{S}$ such that $i < j, \forall i \in \mathcal{S}_+, \forall j \in \mathcal{S}_0$. By this partitioning of the state space the reward matrix and the generator matrix have the following sub-block structure

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \quad \text{and} \quad \mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{pmatrix} .$$

Note that the matrix $\mathbf{Q}_4$ is invertable as a consequence of the requirement that $\mathcal{S}_0$ has no absorbing subset. The partitioned form of the performance vectors are

$$\underline{C}^{\sim\sim}(s,v) = \begin{pmatrix} \underline{C}_1^{\sim\sim}(s,v) \\ \underline{C}_2^{\sim\sim}(s,v) \end{pmatrix} \quad \text{and} \quad \underline{s}^{(n)}(w) = \begin{pmatrix} \underline{s}_1^{(n)}(w) \\ \underline{s}_2^{(n)}(w) \end{pmatrix} .$$

I have published the following theorem with its proof in [J3].

**Theorem 3.4** *The n-th moment of the completion time, $\underline{s}^{(n)}(w)$, satisfies the following equations*

$$\underline{s}_1^{(n)}(w) = (-1)^n \sum_{i=0}^{\infty} \frac{w^i}{i!} \mathbf{L}^{(n)}(i) \cdot \underline{h} \quad \textit{and} \quad \underline{s}_2^{(n)}(w) = (-1)^n \sum_{i=0}^{\infty} \frac{w^i}{i!} \mathbf{H}^{(n)}(i) \cdot \underline{h}$$

*where*

$$\mathbf{L}^{(n)}(i) = \begin{cases} \mathbf{0} , & \textit{if } i = 0, n > 0 , \\ (\mathbf{R}_1^{-1} \cdot \mathbf{Q}_1 - \mathbf{R}_1^{-1} \cdot \mathbf{Q}_2 \cdot \mathbf{Q}_4^{-1} \cdot \mathbf{Q}_3)^i , & \textit{if } i \geq 0, n = 0 , \\ -\mathbf{R}_1^{-1} \cdot \mathbf{Q}_2 \cdot \mathbf{Q}_4^{-2} \cdot \mathbf{Q}_3 - \mathbf{R}_1^{-1} , & \textit{if } i = 1, n = 1 , \\ (-1)^{n+1}\ n!\ \mathbf{R}_1^{-1} \cdot \mathbf{Q}_2 \cdot \mathbf{Q}_4^{-n-1} \cdot \mathbf{Q}_3 , & \textit{if } i = 1, n \geq 2 , \\ \sum_{l=0}^{n} \binom{n}{l} \mathbf{L}^{(l)}(1) \cdot \mathbf{L}^{(n-l)}(i-1) , & \textit{if } i \geq 2, n \geq 1 , \end{cases}$$

$$\mathbf{H}^{(n)}(i) = \begin{cases} (-1)^n \mathbf{Q}_4^{-(n+1)} \cdot \mathbf{Q}_3 \ , & \text{if } \ i = 0, n \geq 0 \ , \\ \mathbf{Q}_3 \cdot \mathbf{Q}_4^{-1} \left[ \mathbf{R}_1^{-1} \left( \mathbf{Q}_1 - \mathbf{Q}_2 \cdot \mathbf{Q}_4^{-1} \cdot \mathbf{Q}_3 \right) \right]^i \ , & \text{if } \ i \geq 0, n = 0 \ , \\ \sum_{l=0}^{n} \binom{n}{l} \mathbf{H}^{(l)}(i-1) \cdot \mathbf{G}^{(n-l)} \ , & \text{if } \ i \geq 1, n \geq 1 \ , \end{cases}$$

$$\mathbf{G}^{(n)} = \begin{cases} \mathbf{R}_1^{-1} \cdot (\mathbf{Q}_1 - \mathbf{Q}_2 \cdot \mathbf{Q}_4^{-1} \cdot \mathbf{Q}_3) \ , & \text{if } \ n = 0 \ , \\ \mathbf{R}_1 \cdot \mathbf{Q}_2 \cdot \mathbf{Q}_4^{-2} \cdot \mathbf{Q}_3 - \mathbf{R}_1 \ , & \text{if } \ n = 1 \ , \\ (-1)^{n+1} \ n! \ \mathbf{R}_1 \cdot \mathbf{Q}_2 \cdot \mathbf{Q}_4^{-n-1} \cdot \mathbf{Q}_3 \ , & \text{if } \ n \geq 2 \ . \end{cases}$$

*Proof:* Substituting the vectors and matrices in (3.4) with their partitioned form and using the following form of matrix inverse

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \end{pmatrix}$$

where

$$\mathbf{A} = s\mathbf{I}_1 + v\mathbf{R}_1 - \mathbf{Q}_1 \ , \quad \mathbf{B} = -\mathbf{Q}_2 \ , \quad \mathbf{C} = -\mathbf{Q}_3 \ \text{ and } \ \mathbf{D} = s\mathbf{I}_4 - \mathbf{Q}_4$$

for the vector $\underline{C}_1^{\sim\sim}(s, v)$ we have

$$\underline{C}_1^{\sim\sim}(s, v) = v \left[ s\mathbf{I}_1 + v\mathbf{R}_1 - \mathbf{Q}_1 - \mathbf{Q}_2 \cdot (s\mathbf{I}_4 - \mathbf{Q}_4)^{-1} \cdot \mathbf{Q}_3 \right]^{-1} \cdot \mathbf{R}_1 \cdot \underline{h} \ . \tag{3.9}$$

Since the matrix ${\mathbf{R}_1}^{-1}$ exists by its definition the inverse Laplace transform of (3.9) gives

$$\underline{C}_1^{\sim}(s, w) = e^{\boldsymbol{\alpha}(s)w} \ \cdot \underline{h} = \sum_{i=0}^{\infty} \frac{\boldsymbol{\alpha}(s)^i}{i!} w^i \cdot \underline{h}$$

where

$$\boldsymbol{\alpha}(s) = \mathbf{R}_1^{-1} \cdot \mathbf{Q}_1 + \mathbf{R}_1^{-1} \cdot \mathbf{Q}_2 \cdot (s\mathbf{I}_4 - \mathbf{Q}_4)^{-1} \cdot \mathbf{Q}_3 - s\mathbf{R}_1^{-1} \ .$$

The $n$-th moment of the completion time is

$$\underline{s}_1^{(n)}(w) = (-1)^n \frac{\partial^n}{\partial s^n} \underline{C}_1^{\sim}(s, w) \bigg|_{s=0} = (-1)^n \sum_{i=0}^{\infty} \frac{w^i}{i!} \frac{\partial^n}{\partial s^n} \boldsymbol{\alpha}(s)^i \bigg|_{s=0} \cdot \underline{h}$$

where the $n$-th deviate of the matrix $\boldsymbol{\alpha}(s)^i$ can be evaluated using Leibniz rule. Now $\mathbf{L}^{(n)}(i) = \frac{\partial^n}{\partial s^n} \boldsymbol{\alpha}(s)^i \Big|_{s=0}$ this completes the proof for the performance vector $\underline{s}_1^{(n)}(w)$.

The same partitioning of (3.4) gives

$$\begin{aligned}\underline{C_2^{\sim}}(s,w) &= (s\mathbf{I}_4+\mathbf{Q}_4)^{-1}\cdot\mathbf{Q}_3\cdot\underline{C_1}(s,w)\\ &= \sum_{i=0}^{\infty}\frac{w^i}{i!}(s\mathbf{I}_4+\mathbf{Q}_4)^{-1}\cdot\mathbf{Q}_3\cdot\boldsymbol{\alpha}(s)^i\cdot\underline{h}\\ &= \sum_{i=0}^{\infty}\frac{w^i}{i!}\mathbf{Q}_3\cdot{\mathbf{Q}_4}^{-1}\left[{\mathbf{R}_1}^{-1}\left(\mathbf{Q}_1-s\mathbf{I}_1+\mathbf{Q}_2\left(s\mathbf{I}_1-\mathbf{Q}_4\right)^{-1}\mathbf{Q}_3-s\mathbf{I}_4\right)\right]^i\cdot\underline{h}\end{aligned}$$

and applying Leibniz rule as before

$$\underline{s_2^{(n)}}(x)=(-1)^n\cdot\frac{\partial^n}{\partial s^n}\underline{C_2^{\sim}}(s,x)\bigg|_{s=0}=(-1)^n\sum_{i=0}^{\infty}\frac{w^i}{i!}\mathbf{H}^{(n)}(i)\cdot\underline{h}$$

gives the theorem. □

## 3.5 Numerical algorithms based on randomization

In the previous sections iterative methods were provided to compute moments of reward measures, but due to the properties of digital computers using floating point numbers a direct application of those methods would result in numerical problems such as instabilities, "ringing" (negative probabilities), etc. The main reason of these problems is that matrices with positive and negative elements like $\mathbf{Q}$ are multiplied several times. To avoid these problems modified algorithms are proposed. The proposed algorithms utilize the advantage of the randomization. First, we construct the following two matrices as

$$\mathbf{A}=\frac{\mathbf{Q}}{q}+\mathbf{I}\quad\text{and}\quad\mathbf{S}=\frac{\mathbf{R}}{qd}$$

where $q=\max_{i,j}(|q_{ij}|)$ and $d=\max_i(r_i)/q$. By this definition $\mathbf{A}$ is a stochastic matrix ($0\le a_{i,j}\le 1,\forall i,j$ and $\sum_j a_{i,j}=1,\forall i$) and $\mathbf{S}$ is a diagonal matrix such that $0\le s_{i,i}\le 1,\forall i$. The dimension of $d$ is unit of reward. $d$ can be considered as a scaling factor of the accumulated reward. Using these matrices

$$\underline{B}^{\sim}(t,v)=e^{(\mathbf{Q}-v\mathbf{R})t}\cdot\underline{h}=e^{(\mathbf{A}-vd\mathbf{S})qt}\cdot\underline{h}\;e^{-qt}\;. \tag{3.10}$$

I have published the following theorem with its proof in [J3].

**Theorem 3.5** *The n-th moment of the accumulated reward can be computed using only matrix-vector multiplications and saving only vectors of size $\#\mathcal{S}$ as follows*

$$\underline{m}^{(n)}(t)=n!\;d^n\;\sum_{i=0}^{\infty}\underline{U}^{(n)}(i)\frac{(qt)^i}{i!}e^{-qt} \tag{3.11}$$

*where the coefficient vector* $\underline{U}^{(n)}(i)$ *is defined as*

$$\underline{U}^{(n)}(i) = \begin{cases} \underline{0}\ , & \text{if} \quad i = 0, n \geq 1\ , \\ \underline{h}\ , & \text{if} \quad i \geq 0, n = 0\ , \\ \mathbf{A} \cdot \underline{U}^{(n)}(i-1) + \mathbf{S} \cdot \underline{U}^{(n-1)}(i-1)\ , & \text{if} \quad i \geq 1, n \geq 1\ . \end{cases} \tag{3.12}$$

*Proof:* Starting from (3.10) the proof of Theorem 3.5 follows the same pattern as the proof of Theorem 3.2. □

To demonstrate the iterative algorithm of computing the coefficient vector $\underline{U}^{(n)}(i)$ the first elements of $\underline{U}^{(n)}(i)$ evaluated based on (3.12) are provided in Table 3.2.

| $\underline{U}^{(n)}(i)$ | i=0 | i=1 | i=2 | i=3 |
|---|---|---|---|---|
| n=0 | $\underline{h}$ | $\underline{h}$ | $\underline{h}$ | $\underline{h}$ |
| n=1 | $\underline{0}$ | $\mathbf{S}\underline{h}$ | $\mathbf{AS}\underline{h} + \mathbf{S}\underline{h}$ | $\mathbf{AAS}\underline{h} + \mathbf{AS}\underline{h} + \mathbf{S}\underline{h}$ |
| n=2 | $\underline{0}$ | $\underline{0}$ | $\mathbf{SS}\underline{h}$ | $\mathbf{ASS}\underline{h} + \mathbf{SAS}\underline{h} + \mathbf{SS}\underline{h}$ |
| n=3 | $\underline{0}$ | $\underline{0}$ | $\underline{0}$ | $\mathbf{SSS}\underline{h}$ |

Table 3.2: *First elements of the coefficient vector (only rate reward)*

Suppose we are interested in the first 3 moments of the accumulated reward. To perform the computation 3 vectors of size $\#\mathcal{S}$ need to store $\underline{U}^{(n)}(i), n = 1, 2, 3$. In each iteration step $i = 1, 2, 3, \ldots$ matrix-vector multiplications and vector summations must be performed according to (3.12) using the vectors of the previous iteration step and the constant matrices **A** and **S**. Figure 3.1. shows the dependency structure of the computation. We can recognize that only the $(i-1)$-th column (iteration) of $\underline{U}$ is used for calculating the $i$-th column of $\underline{U}$. Note that **S** is a diagonal matrix and the matrix **A** is as sparse as the matrix **Q** is. Further 3 vectors of the same size need to store the "actual value" of $\underline{m}^{(n)}(t), n = 1, 2, 3$ according to (3.11).

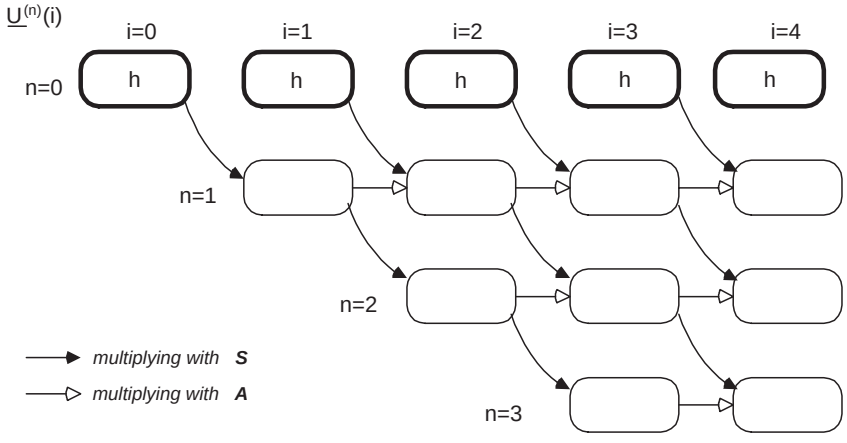



Figure 3.1: *The dependency structure of iteration steps (only rate reward)*

The following theorem provides a global error bound for the algorithm. I have published the theorem with its proof in [J3].

**Theorem 3.6** *The n-th moment of the accumulated reward can be calculated as a finite sum and an error part, where the maximum allowed error is* $\varepsilon$

$$\underline{m}^{(n)}(t) = n!\ d^n\ \sum_{i=0}^{G-1} \underline{U}^{(n)}(i) \frac{(qt)^i}{i!} e^{-qt} + \underline{\xi}(G)$$

*where the value of* $G$ *can be determined by*

$$G = \min_{g \in \mathbb{N}} \left( g \ \Big| \sum_{i=0}^{g-2} \frac{(qt)^i}{i!} e^{-qt} > 1 - \frac{\varepsilon}{(qt)\ n!\ d^n} \right)$$

*and the* $\underline{0} \le \underline{\xi}(G) \le \underline{h}\ \varepsilon$ *inequality holds for all the elements of the vectors.*

*Proof:* By the definition of $\mathbf{S}$ and $\mathbf{A}$

$$\underline{0} \le \mathbf{S} \cdot \underline{h}\ \le\ \underline{h} \quad \text{and} \quad \underline{0} \le \mathbf{A} \cdot \mathbf{S} \cdot \underline{h}\ \le\ \underline{h}$$

hold piece-wise (as all the subsequent vector inequalities), hence $\underline{U}^{(n)}(i)$ is bounded by

$$\underline{0} \le \underline{U}^{(n)}(i) \le i\ \underline{h}\ . \tag{3.13}$$

The error $\underline{\xi}(G)$ incurred when eliminating the tail of the infinite sum is also bounded by

$$\underline{\xi}(G) = n! d^n \sum_{i=G}^{\infty} \underline{U}^{(n)}(i)\ \frac{(qt)^i}{i!} e^{-qt} \le n! d^n \sum_{i=G}^{\infty} \underline{h}\ i\ \frac{(qt)^i}{i!} e^{-qt} \le qtn! d^n \sum_{i=G-1}^{\infty} \underline{h}\ \frac{(qt)^i}{i!} e^{-qt} \tag{3.14}$$

which gives the theorem. □

The same approach can be applied for the analysis of the completion time, when all the reward rates are positive, i.e., $\mathbf{R}^{-1}$ exists. Introduce the following two matrices as

$$\mathbf{B} = \frac{\mathbf{R}^{-1} \cdot \mathbf{Q}}{z} + \mathbf{I} \quad \text{and} \quad \mathbf{T} = \frac{\mathbf{R}^{-1}}{zf} \tag{3.15}$$

where $z = \max_{i,j}(|q_{ij}/r_i|)$ and $f = \max_i(1/r_i)/z$. By this definition $\mathbf{B}$ is a stochastic matrix and $\mathbf{T}$ is a diagonal matrix such that $0 \le t_{i,i} \le 1, \forall i$. $f$ is a number with no dimension. Using these matrices

$$\underline{C}^{\sim}(s,w) = e^{(\mathbf{R}^{-1}\mathbf{Q} - s\mathbf{R}^{-1})w} \cdot \underline{h}\ = e^{(\mathbf{B} - sf\mathbf{T})zw} \cdot \underline{h}\ e^{-zw}\ .$$

I have published the following theorem with its proof in [J3].

**Theorem 3.7** *The n-th moment of the completion time can be computed using only matrix-vector multiplications and saving only vectors of size $\#\mathcal{S}$ as follows*

$$\underline{s}^{(n)}(w) = n!\ f^n\ \sum_{i=0}^{\infty} \underline{V}^{(n)}(i) \frac{(zw)^i}{i!} e^{-zw}$$

*where the coefficient vector $\underline{V}^{(n)}(i)$ is defined as*

$$\underline{V}^{(n)}(i) = \begin{cases} \underline{0}\ , & \text{if} \quad i = 0, n \geq 1\ , \\ \underline{h}\ , & \text{if} \quad i \geq 0, n = 0\ , \\ \mathbf{B} \cdot \underline{V}^{(n)}(i-1) + \mathbf{T} \cdot \underline{V}^{(n-1)}(i-1)\ , & \text{if} \quad i \geq 1, n \geq 1\ . \end{cases}$$

*Proof:* Starting from (3.15) the proof of Theorem 3.7 follows the same pattern as the proof of Theorem 3.3. □

**Theorem 3.8** *The n-th moment of the completion time can be calculated as a finite sum and an error part, where the maximum allowed error is $\varepsilon$*

$$\underline{s}^{(n)}(w) = n!\ f^n\ \sum_{i=0}^{G-1} \underline{V}^{(n)}(i) \frac{(zw)^i}{i!} e^{-zw} + \underline{\xi}(G)$$

*where the value of $G$ can be determined by*

$$G = \min_{g \in \mathbb{N}} \left( g \ \Big| \sum_{i=0}^{g-2} \frac{(zw)^i}{i!} e^{-zw} > 1 - \frac{\varepsilon}{(zw)\ n!\ f^n} \right)$$

*and the $\underline{0} \leq \underline{\xi}(G) \leq \underline{h}\ \varepsilon$ inequality holds for all the elements of the vectors.*

*Proof:* The proof of Theorem 3.8 follows the same pattern as the proof of Theorem 3.6. □.

The numerical analysis of completion time in large models when states with zero reward rate are present is more complicated. A numerical algorithm similar to the one in the Theorem 3.8 can be obtained as well, but on the one hand it is very complicated and on the other hand its applicability is strongly limited by the cardinality of $\mathcal{S}_0$. The $\mathbf{Q}_4$ matrix of cardinality $\#\mathcal{S}_0$ has to be inverted in this case. In general, the complexity of inverting a matrix of cardinality $10^4$ has higher computational complexity and memory requirement than the proposed numerical algorithm with $10^6$ states.

### 3.5.1 Algorithm description

Table 3.3 shows a Pascal like description of the algorithm described by the Theorem 3.6. The description makes the implementation of the algorithm easy. This algorithm calculates moments of the accumulated reward, but using the duality property the same algorithm can be used calculating moments of the completion time, i.e. substituting the first three input parameters as $\mathbf{Q} \to \mathbf{Q} \cdot \mathbf{R}^{-1}$, $\mathbf{R} \to \mathbf{R}^{-1}$ and $t \to w$ results in the algorithm described by the Theorem 3.8.

**Input**

$\mathbf{Q}$, generator matrix of the underlying *CTMC*
$\mathbf{R}$, diagonal matrix of the rate rewards
$t$, the time of accumulation
$n$, number of required moments
$\varepsilon$, required precision

**Output**

$\underline{m}^{(1)}(t)$, the first moment of the accumulated reward
$\underline{m}^{(2)}(t)$, the second moment of the accumulated reward
$\vdots$
$\underline{m}^{(n)}(t)$, the $n$-th moment of the accumulated reward

**Memory requirement** (disregarding the input data)

$2\ n+1$ [Vectors of size $\dim(\mathbf{Q})$]

**Required operations**

$2\ (G-1)\ (n+1)$ [Matrix-vector multiplications]
$(G-1)\ (2\ n+1)$ [Vector-vector additions]
$2\ (G-1)$ [Scalar-vector multiplications]

$q := \max_{i,j}(|q_{ij}|)$; $d := \max_i(r_i)/q$;
$\mathbf{A} := \mathbf{Q}/q + \mathbf{I}$; $\mathbf{S} := \mathbf{R}/(q\ d)$;
$i := 0$;
$C := q\ t\ n!\ d^n$;
$x := \text{Poisson}(0; qt)$;
**While** $x < (1 - \varepsilon/C)$ **Do**
  **Begin**
    $i := i + 1$;
    $x := x + \text{Poisson}(i; qt)$;
  **End**;
$G := i + 2$;
$\underline{U}^{(0)} = \underline{h}$;
**For** $i := 1$ **To** $n$ **Do** $\underline{U}^{(i)} := \underline{0}$;
**For** $i := 1$ **To** $n$ **Do** $\underline{m}^{(i)}(t) := \underline{0}$;

**For** $i := 1$ **To** $G-1$ **Do**
  **Begin**
    **For** $j := n$ **DownTo** $0$ **Do**
      $\underline{U}^{(j)} := \mathbf{S} \cdot \underline{U}^{(j-1)} + \mathbf{A} \cdot \underline{U}^{(j)}$;
    **For** $k := 1$ **To** $n$ **Do**
      $\underline{m}^{(k)}(t) := \underline{m}^{(k)}(t) + \underline{U}^{(k)} \cdot \text{Poisson}(i; qt)$;
  **End**;
**For** $i := 1$ **To** $n$ **Do**
  $\underline{m}^{(i)}(t) := i!\ d^i \cdot \underline{m}^{(i)}(t)$;

Table 3.3: *Algorithm : First moments of the accumulated reward (only rate reward)*

## 3.6 Numerical example

In this example, a performance parameter of a Carnegie-Mellon multiprocessor system is evaluated by the proposed algorithm. The system is similar to the one presented in [STR88]. The

system consists of $P$ processors, $M$ memories and an interconnection network composed by switches which allows any processor to access any memory. The failure rates per hour for the system are set to be $\mu_P = 0.5$, $\mu_M = 0.05$ and $\mu_S = 0.02$ for the processors, memories and switches respectively.

| | |
|---|---|
| **Input** | |
| $P = 16$ | number of processors |
| $M = 16$ | number of memories |
| $S = 8$ | number of switches |
| $\mu_P = 0.5$ | failure rate of a processor |
| $\mu_M = 0.05$ | failure rate of a memory |
| $\mu_S = 0.02$ | failure rate of a switch |
| $\mu_G = 0.1$ | preventive maintenance rate |
| $\lambda_P = 2$ | repair rate of processors |
| $\lambda_M = 1$ | repair rate of memories |
| $\lambda_S = 0.5$ | repair rate of switches |
| $\lambda_G = 1$ | preventive repair rate |
| $GR =$ **True** | enable/disable preventive maintenance |
| **State space** | |
| $a$ : 0 **To** $P$ | number of active processors |
| $b$ : 0 **To** $M$ | number of active memories |
| $c$ : 0 **To** $S$ | number of active switches |
| **Extra state** $= \{ F \}$ | extra state |
| **Generator matrix** | |
| $(a,b,c) \rightarrow (a+1,b,c) = \lambda_P$ | processor repair |
| $(a,b,c) \rightarrow (a,b+1,c) = \lambda_M$ | memory repair |
| $(a,b,c) \rightarrow (a,b,c+1) = \lambda_S$ | switch repair |
| $GR$ : $(F) \rightarrow (P,M,S) = \lambda_G$ | global repair |
| $(a,b,c) \rightarrow (a-1,b,c) = a \cdot \mu_P$ | processor failure |
| $(a,b,c) \rightarrow (a,b-1,c) = b \cdot \mu_M$ | memory failure |
| $(a,b,c) \rightarrow (a,b,c-1) = c \cdot \mu_S$ | switch failure |
| $GR$ : $(a,b,c) \rightarrow (F) = \mu_G$ | global failure |
| **Diagonal reward matrix** | |
| $(a,b,c) =$ **Min**$(a,b,c)$ | processing power in state (a,b,c) |
| **Initial distribution** | |
| $(P,M,S) = 1$ | starting form the perfect state |

Table 3.4: *High-level description of the MRM of a Carnegie-Mellon multiprocessor system*

Viewing the interconnecting network as $S$ switches and modelling the system at the processor-memory-switch level, the system performance depends on the minimum of the number of operating processors, memories and switches. Each state is thus specified by a triple $(a, b, c)$ indicating the number of operating processors, memories and switches, respectively. We augment the preventive maintenance with state $F$. Events that decrease the number of operational units are the failures and events that increase the number of operational elements are the repairs. When a component fails, a recovery action must be taken (e.g., shutting down the failed processor, etc.).

Two kinds of repair actions are possible, preventive maintenance is initiated with rate $\mu_G = 0.1$ per hour which restores the system to state $(N, M, S)$ with rate $\lambda_G = 1.0$ per hour from state $F$ and local repair which can be thought of as a repair person beginning to fix a component of the system as soon as a component failure occurs. We assume that there is only

one repair person for each component type. Let the local repair rates be $\lambda_P = 2.0$, $\lambda_M = 1.0$ and $\lambda_S = 0.5$ for processors, memories and switches, respectively.

The system starts from the perfect state $(P, M, S)$. The studied system has 16 processors, 16 memories and 8 switches, thus the state space consists of $2,602$ states. The performance of the system is proportional to the number of cooperating processors and memories, whose cooperation is provided by one switch. The reward rate is defined as the minimum number of the operational processors, memories and switches. The minimal operational configuration is supposed to have one processor, one memory and one interconnection switch. We consider the *processing power* of the system averaged over a given time interval. Therefore we will study the accumulate reward averaged over the time which gives the processing power. In this example the processing power is always between 0 and 8 processors because the system has 8 switches.

Table3.4 defines the considered *MRM* using a high-level description. We developed a numerical tool called MRMSolve [C2, C10] which determines moments of the accumulated reward starting from this type of high-level *MRM* description.

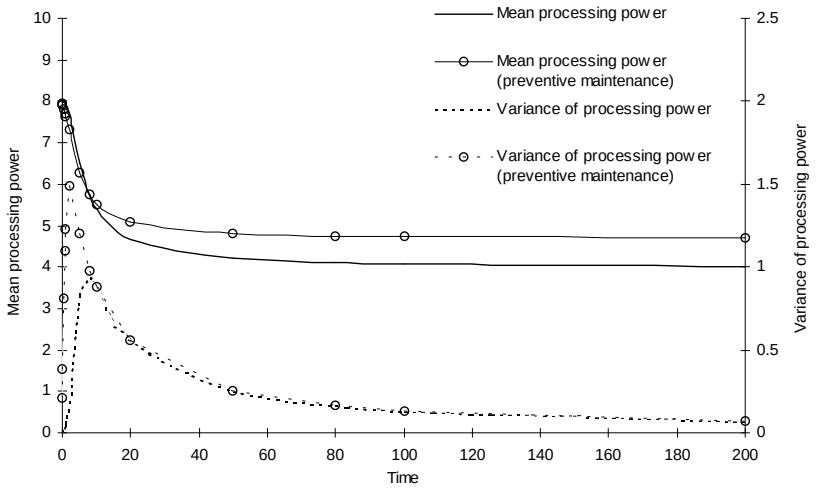



Figure 3.2: *Mean and variance of the processing power*

The mean and the variance of the processing power of the two cases, enabling or disabling preventive maintenance, are compared in Figure 3.2. As it was expected, the mean processing power of the case without preventive maintenance is less. The variance curves show that the preventive maintenance causes higher variance in the processing power. Consequently, applying preventive maintenance results in a higher mean processing power.

## 3.7 Conclusion

This chapter described an algorithm which calculates the moments of the accumulated reward and the completion time. The algorithm can deal with *MRM*s with more than $10^6$ states.

Section 3.1 gave a new approach to derive the Laplace transform domain description of the reward measures. Section 3.2 studied the duality of the accumulated reward and the completion time. Section 3.3 and Section 3.4 presented the basic method which derived moments of a reward measure from its Laplace transform domain description. Section 3.5 provided the stable numerical algorithm for moments calculation. Finally, in Section 3.6 a Carnegie-Mellon multiprocessor system was evaluated by the proposed algorithm.

# Chapter 4

# Transient analysis of Markov reward models with rate and impulse reward

I address this chapter for presenting my *results in the analysis of MRMs with rate and impulse reward*. I have published this work in [C1, C3]. Compare to the literature of *MRM*s with only rate reward there are very few results available for the analysis of *MRM*s with rate and impulse reward. So my aim was to develop a *stable numerical algorithm* which can deal with this type of *MRM*s. The proposed algorithm provides moments of the accumulated reward and the completion time.

The duality of reward measures plays an important role of *MRM*s analysis. The duality of the completion time and the accumulated reward of a *MRM* with only rate reward was discussed in Section 3.2. This chapter extends the duality concept for *MRM*s with rate and impulse reward.

## 4.1 Transform domain description of the accumulated reward

This section gives the distribution of the accumulated reward in single Laplace transform domain applying a new analysis approach that utilizes the duality of the considered reward measures.

The description of the accumulated reward is given by the following theorem. I have published the theorem with its proof in [C3].

**Theorem 4.1** *Laplace transform of the accumulated reward is as follows*

$$\underline{B}^{\sim}(t,v) = e^{[\mathbf{Q}\odot\mathbf{D}^{\sim}(v)-v\mathbf{R}]t}\cdot\underline{h}$$

*where $\odot$ denotes the piecewise matrix multiplication ( $[\mathbf{A}\odot\mathbf{B}]_{ij} = a_{ij}\cdot b_{ij}$).*

The proof of the theorem is readable from the following two lemmas.

**Lemma 4.1** *Let $\underline{C}_m(t)$ be the column vector of the distribution of the completion time when the work requirement is exponentially distributed with parameter $m$, i.e. $\underline{C}_m(t) = [C_i(t) \mid G(w) = 1-e^{-mw}]$, and $\hat{\underline{C}}_m(t)$ is the analytical continuation of $\underline{C}_m(t)$. The $\underline{B}^{\sim}(t,v)$ satisfies*

$$\underline{B}^{\sim}(t,v) = \underline{h} - \hat{\underline{C}}_m(t)\Big|_{m=v}.$$

*Proof of Lemma 4.1* From (2.3) and (2.4) we have

$$C_i(t) = \int_0^\infty \Big(1 - B_i(t,w)\Big)\ dG(w) = m \int_0^\infty \Big(1 - B_i(t,x)\Big)\ e^{-mx}\ dx\ . \tag{4.1}$$

Equation (4.1) can be rewritten using Laplace transform of the accumulated rewards

$$C_i(t) = 1 - vB_i^*(t,v)\Big|_{v=m} = 1 - B_i^\sim(t,v)\Big|_{v=m}$$

which, in vector form, is

$$\underline{C}_m(t) = \underline{h} - \underline{B}^\sim(t,v)\Big|_{v=m}\ .$$

Since $\underline{B}^\sim(t,v)$ is an analytical complex function for $\Re(v) \geq 0$ the lemma is given. □

**Lemma 4.2** *The completion time of an exponentially distributed work requirement with parameter $m$ is a phase type distributed random variable, even with generally distributed impulse rewards, and its distribution function can be evaluated as*

$$\underline{C}_m(t) = \underline{h} - e^{[\mathbf{Q} \odot \mathbf{D}^\sim(m) - m\mathbf{R}]t} \cdot \underline{h} \tag{4.2}$$

*where the matrix $[\mathbf{Q} \odot \mathbf{D}^\sim(m) - m\mathbf{R}]$ is the generator of a phase type distribution.*

*Proof of Lemma 4.2* Due to the memoryless property of the exponentially distributed work requirement the remaining work to complete is exponentially distributed with the same parameter at any instance of time before completion. At a state transition from the state $i$ to the state $j$ the completion occurs if the impulse reward $\mathcal{D}_{ij}$ is not less than the remaining work to complete, $\mathcal{W}_r$, i.e., the completion occurs with the following probability

$$\begin{aligned} &Pr\{\text{completion}\} = Pr\{\mathcal{D}_{ij} > \mathcal{W}_r\} = \int_0^\infty Pr\{\mathcal{D}_{ij} > w\}\ dG(w) = \\ &1 - \int_0^\infty Pr\{\mathcal{D}_{ij} \leq w\}\ dG(w) = 1 - \int_0^\infty D_{ij}(w)\ dG(w) = \\ &1 - m \int_0^\infty D_{ij}(w)\ e^{-mw}\ dw = 1 - m\ D_{ij}^*(m) = 1 - D_{ij}^\sim(m)\ \ . \end{aligned} \tag{4.3}$$

Assuming the process stays in the state $i$ at time $t$ before completion the following cases can occur in the interval $(t, t+dt)$

- *no state transition and no completion* occurs with probability $\quad 1 + (q_{ii} - r_i m)dt + \sigma(dt)$,
- *no state transition and completion* occurs with probability $\quad r_i m dt + \sigma(dt)$,

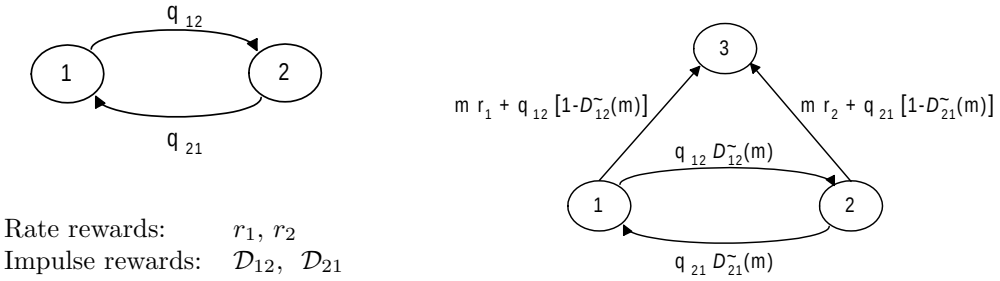



Figure 4.1: *A simple two-state MRM (impulse and rate rewards)*

- *state transition to $j$ and no completion* occurs with probability $D_{ij}^{\sim}(m)q_{ij}dt + \sigma(dt)$,
- *state transition to $j$ and completion* occurs with probability $(1 - D_{ij}^{\sim}(m))q_{ij}dt + \sigma(dt)$,
- *any other cases* occur with probability $\sigma(dt)$.

Based on this behavior a new *CTMC* can be defined by adding an absorbing state, $M+1$, to the state space of the underlying *CTMC*, defining state transitions from $\forall i \in \mathcal{S}$ to $M+1$ with the rate $mr_i + \sum_{j,j\neq i} q_{ij}(1 - D_{ij}^{\sim}(m))$ and setting the transition rates between the states in $\mathcal{S}$ according to the above described behavior. The absorbing state represents the completion of the exponentially distributed work requirement. The new *CTMC* defines a phase type distribution of order $\#\mathcal{S}$. Its $\#\mathcal{S} \times \#\mathcal{S}$ generator is $[\mathbf{Q} \odot \mathbf{D}^{\sim}(m) - m\mathbf{R}]$, and its time to absorption is given by (4.2). □

Lemma 4.2 is demonstrated through a simple example of a two-state system shown in Figure 4.1. The considered *MRM* is defined by the generator matrix of the underlying *CTMC*, the rate reward matrix and the impulse reward matrix.

$$\mathbf{Q} = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} \quad \text{and} \quad \mathbf{D}^{\sim}(v) = \begin{pmatrix} 1 & D_{12}^{\sim}(v) \\ D_{21}^{\sim}(v) & 1 \end{pmatrix} .$$

The generator of the *CTMC* with the additional absorbing state, which describes the phase type distribution, is

$$\begin{pmatrix} q_{11} - mr_1 & q_{12}D_{12}^{\sim}(m) & mr_1 + q_{12}(1 - D_{12}^{\sim}(m)) \\ q_{21}D_{21}^{\sim}(m) & q_{22} - mr_2 & mr_2 + q_{12}(1 - D_{21}^{\sim}(m)) \\ 0 & 0 & 0 \end{pmatrix}$$

whose $\#\mathcal{S} \times \#\mathcal{S}$ $(2 \times 2)$ upper left submatrix has the form $[\mathbf{Q} \odot \mathbf{D}^{\sim}(m) - m\mathbf{R}]$.

## 4.2 Transform domain description of the completion time

This section provides the distribution of the completion time in single Laplace transform domain applying a new analysis approach which utilizes the duality of the accumulated reward and the completion time.

The main theorem is provided for *MRM*s with strictly positive rate and phase type distributed impulse rewards, but comments are also made on the analysis of *MRM*s which have states with zero reward rates and generally distributed holding times.

### 4.2.1 MRM with positive rate and phase type distributed impulse reward

**Theorem 4.2** *Laplace transform of the completion time of a MRM with positive rate and phase type distributed impulse rewards can be computed by the following expression*

$$\underline{C}^{\sim}(s,w) = \mathbf{\Gamma} \cdot \left(e^{(\mathbf{T}-s\mathbf{F})w} \cdot \underline{h}\right) \tag{4.4}$$

*where*

- $\mathbf{T}$ *is a generator matrix of a modified Markov chain over an enlarged state space* $\mathcal{G}$ *(*$\mathcal{S} \subset \mathcal{G}$*), that is composed by the state space of the original CTMC and the phase type structures of the impulse rewards. The matrix elements representing state transitions between states in* $\mathcal{S}$ *are maintained if there is no impulse reward associated with them and the rate of these state transitions are set to* $T_{ij} = q_{ij}/r_i$*. If there is an impulse reward associated with the state transition from* $i \in \mathcal{S}$ *to* $j \in \mathcal{S}$ *the* $T_{ij}$ *matrix element is* $0$ *and there is a state transition with rate* $q_{ij}/r_i$ *to the phase type structure of the associated impulse reward. The exit of this phase type structure is directed to state* $j$*.*

  *(This way the enlarged CTMC is such that the rate of the state transitions in* $\mathcal{S}$ *are rescaled and the state transition from state* $i$ *to* $j$ *goes through the phase type structure of the associated impulse reward, if any, i.e., a phase type distributed time, equal the associated impulse reward, is added to the rescaled time of state transition.)*

  *The structure of* $\mathbf{T}$ *is shown in Figure 4.2.*

- $\mathbf{F}$ *is a diagonal matrix of cardinality* $\#\mathcal{G}$ *whose diagonal element associated with* $i \in \mathcal{S}$ *equals* $1/r_i$ *and with* $i \in \mathcal{G} \setminus \mathcal{S}$ *equals* $0$*.*

  *(* $\mathbf{F}$ *can be viewed as the reward matrix of the enlarged Markov chain.)*

- $\mathbf{\Gamma} = [\ \mathbf{I} \mid \mathbf{0}\ ]$ *is a filter matrix of size* $\#\mathcal{S} \times \#\mathcal{G}$ *composed by a unity matrix of cardinality* $\#\mathcal{S}$ *and a* $\#\mathcal{S} \times (\#\mathcal{G} - \#\mathcal{S})$ *matrix of zeros.*

  *(The only role of the* $\mathbf{\Gamma}$ *matrix is that the multiplication with* $\mathbf{\Gamma}$ *eliminates the* $\#\mathcal{G} - \#\mathcal{S}$ *extra vector elements.)*

I have published this theorem with its proof in [C1]. The proof of the theorem is provided through the following two lemmas. First, the counterpart of Lemma 4.1 is provided.

**Lemma 4.3** *The column vector of the distribution of the completion time in Laplace transform domain satisfies*

$$\underline{C}^{\sim}(s,w) = \underline{h} - \underline{\hat{B}}_u(w)\Big|_{u=s}$$

*where* $\underline{B}_u(w)$ *is the distribution of reward accumulated during an interval which is exponentially distributed with parameter* $u$*, i.e.,* $\underline{B}_u(w) = [B_i(\mathcal{T}, w) \mid \mathcal{T}$ *is exp. with parameteru*$]$*, and* $\underline{\hat{B}}_u(w)$ *is the analytical continuation of* $\underline{B}_u(w)$*.*

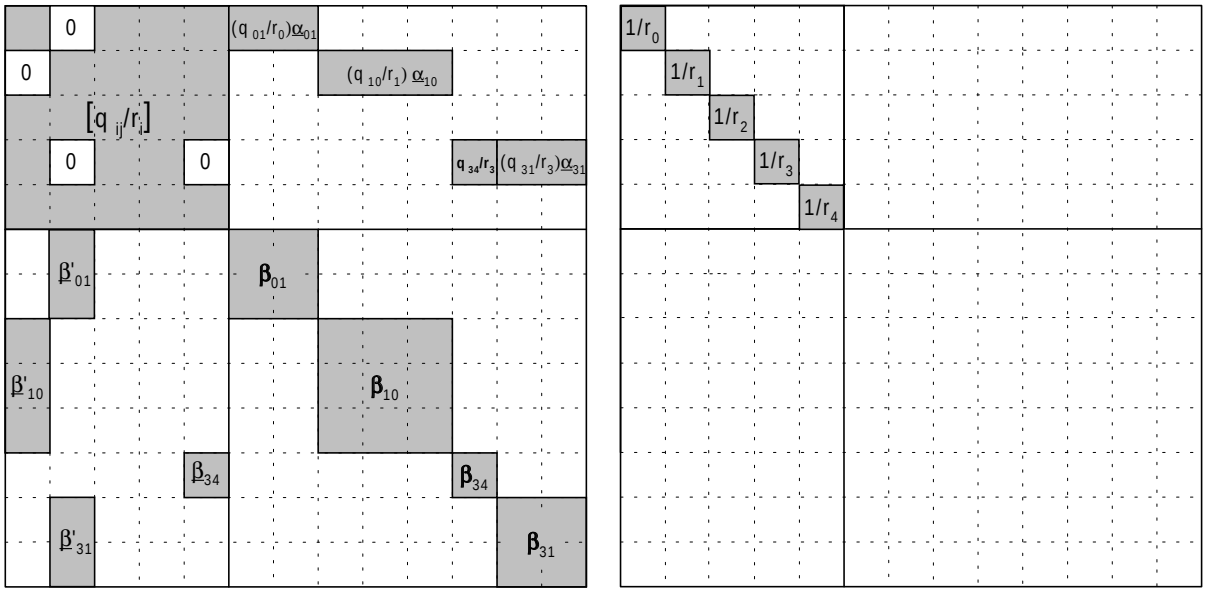



Figure 4.2: *The structure of the generator and reward matrices of the enlarged Markov chain*

*Proof of Lemma 4.3* Using the properties of Laplace transform and equation (2.4) $\underline{C}^{\sim}(s,w)$ can be written as

$$\underline{C}^{\sim}(s,w) = s\,\underline{C}^{*}(s,w) = s\int_0^\infty \underline{C}(t,w)e^{-st}dt = \underline{h} - s\int_0^\infty \underline{B}(t,w)e^{-st}dt\ .$$

When $s$ takes the positive real value $u$ we have

$$\underline{C}^{\sim}(u,w) = \underline{h} - \int_0^\infty \underline{B}(t,w)ue^{-ut}dt = \underline{h} - \underline{B}_u(w)$$

and since $\underline{C}^{\sim}(s,w)$ is an analytical complex function for $\Re(s) \geq 0$ the lemma is given. □

Note that the role of the accumulated reward and the completion time are interchanged in the Lemma 4.1 and Lemma 4.3.

Based on Lemma 4.3 the remaining task to obtain Theorem 4.2 is to determine the amount of reward accumulated by a *MRM* with positive rate and phase type distributed impulse reward during an exponentially distributed period of time.

**Lemma 4.4** *The amount of reward accumulated by a MRM with positive rate and phase type distributed impulse reward during an exponentially distributed period of time is phase type distributed and the generator matrix of this phase type distribution is* $\mathbf{T} - u\mathbf{F}$*, i.e.,*

$$\underline{B}_u(w) = \mathbf{\Gamma}\cdot\left(\underline{h} - e^{(\mathbf{T}-u\mathbf{F})w}\cdot\underline{h}\right)$$

*where* $u$ *is the parameter of the exponential distribution of the accumulation period.*

*Proof of Lemma 4.4* The proof is based on the fact that the accumulation period is exponentially distributed, hence at any instant of time of the accumulation the remaining time till the end of the accumulation period is exponentially distributed with the same parameter.

According to (2.3) $\mathcal{B}_i(\mathcal{T})$ denotes the (random) amount of reward accumulated during the exponentially distributed period $\mathcal{T}$ when $\mathcal{Z}(0) = i$. Let $\tau_i$ be the first sojourn time in state $i$. To evaluate the reward accumulated during a $\mathcal{T}$ long period starting from the state $i$ the following cases have to be considered

- $\mathcal{T} \leq \tau_i$:
  the probability of this case is

  $$Pr\{\mathcal{T} \leq \tau_i\} = \frac{u}{u + q_i}$$

  where $q_i = -q_{ii}$. Under this condition the accumulation period is exponentially distributed with parameter $(u + q_i)$, because

  $$Pr\{\mathcal{T} \leq t | \mathcal{T} \leq \tau_i\} = 1 - e^{-(u+q_i)t} \ .$$

  The amount of reward accumulated in this case is also exponentially distributed with parameter $(u + q_i)/r_i$.

- $\mathcal{T} > \tau_i$:
  the probability of this case is

  $$Pr\{\mathcal{T} > \tau_i\} = \frac{q_i}{u + q_i} \ .$$

  Under this condition the sojourn time is exponentially distributed with parameter $(u+q_i)$, since

  $$Pr\{\tau_i \leq t | \mathcal{T} > \tau_i\} = 1 - e^{-(u+q_i)t} \ .$$

  The amount of reward accumulated when $\mathcal{T} > \tau_i$ is composed by three parts: the reward accumulated in the state $i$, the phase type distributed impulse reward associated with the state transition from the state $i$ to the state $j$, and the reward accumulated starting from the state $j$, $\mathcal{B}_j(\mathcal{T})$, since the remaining time $\mathcal{T} - \tau_i$ is also exponentially distributed with the same parameter. Assuming $\mathcal{T} > \tau_i$ the reward accumulated in state $i$ is exponentially distributed with parameter $(u+q_i)/r_i$, as before, since $\tau_i$ is exponentially distributed with parameter $(u + q_i)$.

Considering these cases Laplace-Stieltjes transform of $\mathcal{B}_i(\mathcal{T})$, $B_i^{\sim}(\mathcal{T}, v) = E(e^{-v\mathcal{B}_i(\mathcal{T})})$, satisfies

$$\begin{aligned} B_i^{\sim}(\mathcal{T}, v) &= \frac{u}{u + q_i} \frac{\frac{u+q_i}{r_i}}{v + \frac{u+q_i}{r_i}} + \frac{q_i}{u + q_i} \left( \frac{\frac{u+q_i}{r_i}}{v + \frac{u+q_i}{r_i}} \sum_{j \in \mathcal{S}, j \neq i} \frac{q_{ij}}{q_i} D_{ij}^{\sim}(v) \ B_j^{\sim}(\mathcal{T}, v) \right) \\ &= \frac{u}{r_i v + u + q_i} + \sum_{j \in \mathcal{S}, j \neq i} \frac{q_{ij}}{r_i v + u + q_i} D_{ij}^{\sim}(v) \ B_j^{\sim}(\mathcal{T}, v) \ . \end{aligned} \tag{4.5}$$

Now, we can evaluate the initial state dependent time to absorption of the phase type distribution characterized by the matrix $\mathbf{T} - u\mathbf{F}$. The random variable $\mathcal{K}_i$ denotes the time

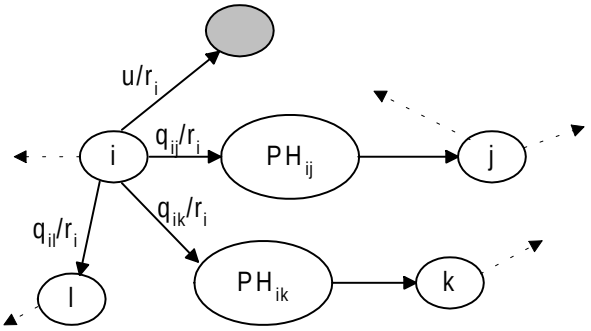
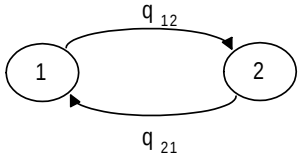



Figure 4.3: *The structure of the phase type distribution characterized by the matrix* $\mathbf{T} - u\mathbf{F}$




Rate rewards: $r_1$, $r_2$
Impulse reward: $\mathcal{D}_{12} \sim \underline{\alpha}_{12}, \boldsymbol{\beta}_{12}$

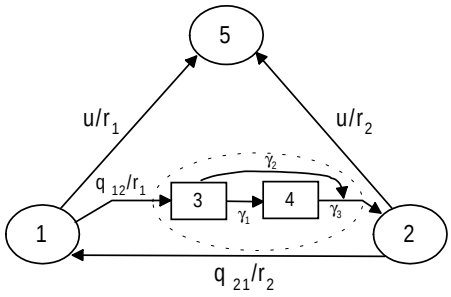



Figure 4.4: *A simple two-state MRM with a phase type distributed impulse reward*

to absorption starting from the state $i$. One can utilize that due to the given structure of the matrix $\mathbf{T}$ and matrix $\mathbf{F}$ a phase type distributed time $\mathcal{D}_{ij}$ is spent in $\mathcal{G} \setminus \mathcal{S}$ if $j$ is the next state in $\mathcal{S}$ that is visited after a sojourn in the state $i \in \mathcal{S}$. Further more, the row sum of the matrix $\mathbf{T} - u\mathbf{F}$ equals $\underline{0}$ for the states in $\mathcal{G} \setminus \mathcal{S}$, i.e., there is no direct transition to the absorbing state from $\mathcal{G} \setminus \mathcal{S}$. Figure 4.3 shows the schematic structure of the phase type distribution characterized by the matrix $\mathbf{T} - u\mathbf{F}$. Based on this structure Laplace transform of $\mathcal{K}_i$, $K_i^{\sim}(v)$, can be provided as a function of Laplace transform of $\mathcal{K}_j; j \in \mathcal{S}$ as follows

$$K_i^{\sim}(v) = \frac{u/r_i}{v + (u + q_i)/r_i} + \sum_{j \in \mathcal{S}, j \neq i} \frac{q_{ij}/r_i}{v + (u + q_i)/r_i} D_{ij}^{\sim}(v) \; K_j^{\sim}(v) \quad . \tag{4.6}$$

By the equavivalence of (4.5) and (4.6) the lemma is given. □

Lemma 4.4 is demonstrated through the following example of a two-state *MRM* (see Figure 4.4.), where a phase type distributed impulse reward (characterized by the vector $\underline{\alpha}_{12}$ and the matrix $\boldsymbol{\beta}_{12}$) is associated with the state transition from the state 1 to the state 2.

$$\mathbf{Q} = \begin{pmatrix} -q_{12} & q_{12} \\ q_{21} & -q_{21} \end{pmatrix}, \quad \underline{\alpha}_{12} = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \boldsymbol{\beta}_{12} = \begin{pmatrix} -(\gamma_1 + \gamma_2) & \gamma_1 \\ 0 & -\gamma_3 \end{pmatrix} \quad \text{and} \quad \underline{\beta}'_{12} = \begin{pmatrix} \gamma_2 \\ \gamma_3 \end{pmatrix}.$$

By the Theorem 4.2 the $\mathbf{T}$ and $\mathbf{F}$ matrices are as follows

$$\mathbf{T} = \begin{pmatrix} -q_{12}/r_1 & 0 & q_{12}/r_1 & 0 \\ q_{21}/r_2 & -q_{21}/r_2 & 0 & 0 \\ 0 & \gamma_2 & -(\gamma_1 + \gamma_2) & \gamma_1 \\ 0 & \gamma_3 & 0 & -\gamma_3 \end{pmatrix} \quad \text{and} \quad \mathbf{F} = \begin{pmatrix} 1/r_1 & 0 & 0 & 0 \\ 0 & 1/r_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad .$$

For easier understanding of the phase type characteristic of the matrix $\mathbf{T} - u\mathbf{F}$ an absorbing state (state 5) is also depicted in Figure 4.4. Note that the case of exponentially distributed impulse reward can be captured as a special case, as a phase type distribution of order 1.

**Potential state space reduction**

In some special cases the number of additional states, resulted by the phase type structure of the impulse rewards, can be reduced. These special cases can be classified as follows

- **common impulse reward when entering state** $i$: if $\boldsymbol{\beta}_{ji} = \boldsymbol{\beta}_{ki}$ it is enough to represent $\boldsymbol{\beta}_{ji} = \boldsymbol{\beta}_{ki}$ once in $\mathbf{T}$ and the row vectors $q_{ji}/r_j \ \underline{\alpha}_{ji}$ and $q_{ki}/r_k \ \underline{\alpha}_{ki}$ must be saved at the same positions of the $j$-th and $k$-th rows of $\mathbf{T}$, respectively. $\underline{\beta}'_{ji} = \underline{\beta}'_{ki}$ is saved once in column $i$ (left side of Figure 4.5). Note that $\underline{\alpha}_{ji}$ and $\underline{\alpha}_{ki}$ can be different in this case.
- **common impulse reward when leaving state** $i$: if $\boldsymbol{\beta}_{ij} = \boldsymbol{\beta}_{ik}$ and $\underline{\alpha}_{ij} = \underline{\alpha}_{ik}$ then it is enough to represent $\boldsymbol{\beta}_{ji} = \boldsymbol{\beta}_{ki}$ once in $\mathbf{T}$ and the row vector describing the enter of this phase type structure has to be $(q_{ij} + q_{ik})/r_i \ \underline{\alpha}_{ij}$. The column vector describing the exit of this phase type structure must be save twice. In column $j$ it must be saved as $q_{ij}/(q_{ij} + q_{ik}) \ \underline{\beta}'_{ij}$ and in column $j$ as $q_{ik}/(q_{ij} + q_{ik}) \ \underline{\beta}'_{ik}$ (right side of Figure 4.5).

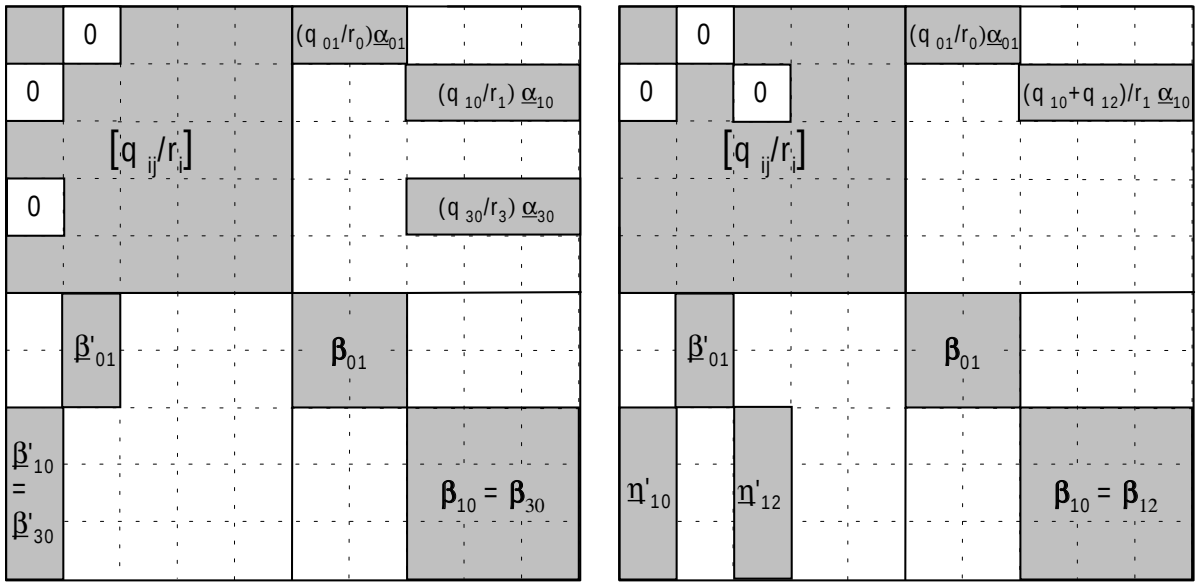



Figure 4.5: *The structures of the generator of the enlarged Markov chain*

### 4.2.2 MRM with non-negative rate and phase type distributed impulse reward

The case of zero reward rates can be viewed as the limiting case when the reward rate tends to zero. Assuming that the reward rate of the state $i$ ($i \in \mathcal{S}$) decreases to zero the associated transition rate of the matrix $\mathbf{T}$ increases to infinity. The limiting case when $r_i = 0$ can be handled based on the concept of Generalized Stochastic Petri Nets [ABC84] that has two kinds of "states" (referred to as marking)

- regular state – that is visited for an exponentially distributed period of time (referred to as tangible marking),
- immediate state – that is visited instantaneously (referred to as vanishing marking).

If $r_i = 0$ ($i \in \mathcal{S}$) the state $i$ became an immediate state, which performs only a random switching, i.e., the process stays in the state $i$ for zero time and after it visits

- the absorbing state – with probability $\frac{u}{u+q_i}$,
- the state $j$, if there is no impulse reward associated with the $i$ to $j$ state transition – with probability $\frac{q_{ij}}{u+q_i}$,
- the phase type structure of the $i$ to $k$ state transition (if there is an impulse reward associated with the $i$ to $k$ state transition) – with probability $\frac{q_{ik}}{u+q_i}$.

Note that similar argument can be used to support states with zero rate rewards and generally distributed holding times. The obtained Laplace transform domain description of the completion time has the following form

$$\underline{C}^{\sim}(s,w) = \mathbf{\Gamma} \cdot \left(e^{(\mathbf{T} \odot \mathbf{H}^{\sim}(s) - s\mathbf{F})w} \cdot \underline{h}\right)$$

where the matrix $\mathbf{H}^{\sim}(s)$ describes the generally distributed holding times.

As it is shown in [ABC84] the obtained stochastic process is a *CTMC*, hence the amount of reward accumulated by a *MRM* with non-negative rate and phase type distributed impulse reward during an exponentially distributed period of time is phase type distributed as well.

## 4.3 Numerical analysis of MRMs with impulse and rate reward

In this section numerical methods are introduced which provide moments of the accumulated reward and the completion time based on their Laplace transform domain descriptions.

### 4.3.1 Moments of the accumulated reward

The column vector $\underline{m}^{(n)}(t)$ can be evaluated based on $\underline{B}^{\sim}(t,v)$ as

$$\underline{m}^{(n)}(t) = (-1)^n \frac{\partial^n \underline{B}^{\sim}(t,v)}{\partial v^n}\bigg|_{v=0}. \tag{4.7}$$

I have published the following theorem with its proof in [C3].

**Theorem 4.3** *The n-th moment of the accumulated reward, $\underline{m}^{(n)}(t)$, satisfies the following equation*

$$\underline{m}^{(n)}(t) = \sum_{i=0}^{\infty} \frac{t^i}{i!} \cdot \mathbf{N}^{(n)}(i) \cdot \underline{h}$$

*where the coefficient matrix* $\mathbf{N}^{(n)}(i)$ *is defined as*

$$\mathbf{N}^{(n)}(i) = \begin{cases} \mathbf{Q}^i \ , & if \quad i \geq 0, n = 0 \ , \\ \mathbf{0} \ , & if \quad i = 0, n \geq 1 \ , \\ \mathbf{Q} \odot \mathbf{D}^{(1)} + \mathbf{R} \ , & if \quad i = 1, n = 1 \ , \\ \mathbf{Q} \odot \mathbf{D}^{(n)} \ , & if \quad i = 1, n \geq 2 \ , \\ \sum_{l=0}^{n} \binom{n}{l} \mathbf{N}^{(l)}(1) \cdot \mathbf{N}^{(n-l)}(i-1) \ , & if \quad i \geq 2, n \geq 1 \ . \end{cases} \tag{4.8}$$

*Proof:* From (4.7) and Theorem 4.1 the $n$-th moment is

$$\begin{aligned} \underline{m}^{(n)}(t) &= (-1)^n \left. \frac{\partial^n}{\partial v^n} e^{[\mathbf{Q} \odot \mathbf{D}^{\sim}(v) - v\mathbf{R}]t} \right|_{v=0} \cdot \underline{h} \\ &= (-1)^n \left. \frac{\partial^n}{\partial v^n} \sum_{i=0}^{\infty} \frac{t^i}{i!} \left[\mathbf{Q} \odot \mathbf{D}^{\sim}(v) - v\mathbf{R}\right]^i \right|_{v=0} \cdot \underline{h} \\ &= (-1)^n \left. \sum_{i=0}^{\infty} \frac{t^i}{i!} \frac{\partial^n}{\partial v^n} \left[\mathbf{Q} \odot \mathbf{D}^{\sim}(v) - v\mathbf{R}\right]^i \right|_{v=0} \cdot \underline{h} \ . \end{aligned}$$

We introduce the following notation

$$\mathbf{N}^{(n)}(i) = \left. \frac{\partial^n}{\partial v^n} \left[\mathbf{Q} \odot \mathbf{D}^{\sim}(v) - v\mathbf{R}\right]^i \right|_{v=0} \qquad \text{for } \forall n, i \ .$$

From Leibniz rule it follows

$$\mathbf{N}^{(n)}(i) = [\mathbf{N}^{(0)}(1) \cdot \mathbf{N}^{(0)}(i-1)]^{(n)} = \sum_{l=0}^{n} \binom{n}{l} \mathbf{N}^{(l)}(1) \cdot \mathbf{N}^{(n-l)}(i-1) \quad \text{if } \ i \geq 2, n \geq 1$$

with the initial conditions

$$\begin{aligned} &\mathbf{N}^{(0)}(0) = \mathbf{I} \ , \\ &\mathbf{N}^{(0)}(i) = \mathbf{Q}^i \ , && \text{if } \ i > 0 \ , \\ &\mathbf{N}^{(n)}(0) = \mathbf{0} \ , && \text{if } \ n \geq 1 \ , \\ &\mathbf{N}^{(1)}(1) = \mathbf{Q} \odot \mathbf{D}^{(1)} + \mathbf{R} \ , \\ &\mathbf{N}^{(n)}(1) = \mathbf{Q} \odot \mathbf{D}^{(n)} \ , && \text{if } \ n \geq 2 \ . \end{aligned}$$

This completes the proof. □

Based on (4.8) the $n$-th moment of the accumulated reward is finite if all the moments of the impulse rewards from the 1st to the $n$-th one are finite (independent of the higher moments); and the $n$-th moment of the accumulated reward can become infinite if at least one moment of the impulse reward is infinite.

### 4.3.2 Moments of the completion time

The column vector $\underline{s}^{(n)}(w)$ can be evaluated based on $\underline{C}^{\sim}(s,w)$ as

$$\underline{s}^{(n)}(w) = (-1)^n \frac{\partial^n \underline{C}^{\sim}(s,w)}{\partial s^n}\bigg|_{s=0} \quad . \tag{4.9}$$

I have published the following theorem with its proof in [C1].

**Theorem 4.4** *The n-th moment of the completion time, $\underline{s}^{(n)}(w)$, satisfies the following equation*

$$\underline{s}^{(n)}(w) = (-1)^n \ \mathbf{\Gamma} \cdot \left( \sum_{i=0}^{\infty} \frac{t^i}{i!} \cdot \mathbf{M}^{(n)}(i) \cdot \underline{h} \right)$$

*where the coefficient matrix $\mathbf{M}^{(n)}(i)$ is defined as*

$$\mathbf{M}^{(n)}(i) = \begin{cases} \mathbf{I} \ , & if \quad i = 0, n = 0 \ , \\ \mathbf{0} \ , & if \quad i = 0, n \geq 1 \ , \\ \mathbf{T}^i \ , & if \quad i \geq 1, n = 0 \ , \\ \mathbf{T} \cdot \mathbf{M}^{(n)}(i-1) - n \ \mathbf{F} \cdot \mathbf{M}^{(n-1)}(i-1) \ , & if \quad i \geq 1, n \geq 1 \ . \end{cases}$$

*Proof:* From (4.9) and (4.4) the $n$-th moment is

$$\begin{aligned} \underline{s}^{(n)}(w) &= (-1)^n \ \mathbf{\Gamma} \cdot \left( \frac{\partial^n e^{(\mathbf{T}-s\mathbf{F})w}}{\partial s^n}\bigg|_{s=0} \cdot \underline{h} \right) \\ &= (-1)^n \ \mathbf{\Gamma} \cdot \left( \frac{\partial^n}{\partial s^n} \ \sum_{i=0}^{\infty} \frac{w^i}{i!} \ (\mathbf{T} - s\mathbf{F})^i \bigg|_{s=0} \cdot \underline{h} \right) \\ &= (-1)^n \ \mathbf{\Gamma} \cdot \left( \sum_{i=0}^{\infty} \frac{w^i}{i!} \ \frac{\partial^n}{\partial s^n} \ (\mathbf{T} - s\mathbf{F})^i \bigg|_{s=0} \cdot \underline{h} \right) \quad . \end{aligned}$$

We introduce the following notation

$$\mathbf{M}^{(n)}(i) = \frac{\partial^n}{\partial s^n} \ (\mathbf{T} - s\mathbf{F})^i \bigg|_{s=0} \qquad \text{for } \forall n, i \ .$$

From Leibniz rule it follows

$$\mathbf{M}^{(n)}(i) \quad = \quad [\mathbf{M}^{(0)}(1) \cdot \mathbf{M}^{(0)}(i-1)]^{(n)} = \sum_{l=0}^{n} \binom{n}{l} \mathbf{M}^{(l)}(1) \cdot \mathbf{M}^{(n-l)}(i-1) \quad \text{if } \ i \geq 2, n \geq 1$$

with the initial conditions $\mathbf{M}^{(0)}(0) = \mathbf{I}$, $\mathbf{M}^{(0)}(i) = \mathbf{T}^i$ and $\mathbf{M}^{(n)}(0) = \mathbf{0}$ $n \geq 1$. This completes the proof. □

## 4.4 Numerical algorithms based on randomization

The iterative methods presented above are not tuned to have nice numerical properties. To avoid numerical problems like instability, "ringing" (negative probabilities), etc., modified algorithm is proposed.

### 4.4.1 Moments of the accumulated reward

The concept of randomization is adopted for avoiding numerical instabilities. We introduce two matrices as follows

$$\mathbf{A} = \frac{\mathbf{Q}}{q} + \mathbf{I} \quad \text{and} \quad \mathbf{S} = \frac{\mathbf{R}}{q}$$

where $q = \max_{i,j}(|q_{ij}|)$. By this definition $\mathbf{A}$ is a stochastic matrix. Using these matrices we have

$$\underline{B}^{\sim}(t,v) = e^{[\mathbf{Q}\odot\mathbf{D}^{\sim}(v)-v\mathbf{R}]t}\cdot\underline{h} = e^{[(\mathbf{A}-\mathbf{I})\odot\mathbf{D}^{\sim}(v)-v\mathbf{S}]qt}\cdot\underline{h} = e^{[\mathbf{A}\odot\mathbf{D}^{\sim}(v)-v\mathbf{S}]qt}\cdot\underline{h}\cdot e^{-qt}. \quad (4.10)$$

I have published the following theorem with its proof in [C3].

**Theorem 4.5** *The n-th moment of the accumulated reward can be computed using only matrix-vector multiplications and saving only vectors of size $\#\mathcal{S}$ as follows*

$$\underline{m}^{(n)}(t) = \sum_{i=0}^{\infty}\underline{U}^{(n)}(i)\cdot\frac{(qt)^i}{i!}e^{-qt} \quad (4.11)$$

*where the coefficient vector $\underline{U}^{(n)}(i)$ is defined as*

$$\underline{U}^{(n)}(i) = \begin{cases} \underline{h}\ , & if \quad i \geq 0, n = 0\ , \\ \underline{0}\ , & if \quad i = 0, n \geq 1\ , \\ \sum_{k=0}^{n}\binom{n}{k}\mathbf{V}^{(k)}\cdot\underline{U}^{(n-k)}(i-1)\ , & if \quad i \geq 1, n \geq 1\ , \end{cases} \quad (4.12)$$

*and the matrix $\mathbf{V}^{(n)}$ is defined as*

$$\mathbf{V}^{(n)} = \begin{cases} \mathbf{A} & if \quad n = 0\ , \\ \mathbf{A}\odot\mathbf{D}^{(1)} + \mathbf{S} & if \quad n = 1\ , \\ \mathbf{A}\odot\mathbf{D}^{(n)} & if \quad n \geq 2\ , \end{cases}$$

*where $\mathbf{D}^{(n)}$ is the matrix of the n-th moment of impulse reward.*

*Proof:* Starting from (4.10) the proof of the Theorem 4.5 follows the same pattern as the proof of the Theorem 4.3. □

To demonstrate the iterative algorithm of computing $\underline{U}^{(n)}(i)$ first elements of the iteration are provided in Table 4.1 and the dependence of the consecutive term is depicted in Figure 4.6.

| | $\mathbf{V}^{(n)}$ | $\underline{U}^{(n)}(0)$ | $\underline{U}^{(n)}(1)$ | $\underline{U}^{(n)}(2)$ | $\underline{U}^{(n)}(3)$ |
|---|---|---|---|---|---|
| $n=0$ | $\mathbf{A}$ | $\underline{h}$ | $\underline{h}$ | $\underline{h}$ | $\underline{h}$ |
| $n=1$ | $\mathbf{A}\odot\mathbf{D}^{(1)}+\mathbf{S}$ | $\underline{0}$ | $\mathbf{V}^{(1)}\cdot\underline{h}$ | $\mathbf{V}^{(0)}\cdot\underline{U}^{(1)}(1)+$ $\mathbf{V}^{(1)}\cdot\underline{U}^{(0)}(1)$ | $\mathbf{V}^{(0)}\cdot\underline{U}^{(1)}(2)+$ $\mathbf{V}^{(1)}\cdot\underline{U}^{(0)}(2)$ |
| $n=2$ | $\mathbf{A}\odot\mathbf{D}^{(2)}$ | $\underline{0}$ | $\mathbf{V}^{(2)}\cdot\underline{h}$ | $\mathbf{V}^{(0)}\cdot\underline{U}^{(2)}(1)+$ $2\cdot\mathbf{V}^{(1)}\cdot\underline{U}^{(1)}(1)+$ $\mathbf{V}^{(2)}\cdot\underline{U}^{(0)}(1)$ | $\mathbf{V}^{(0)}\cdot\underline{U}^{(2)}(2)+$ $2\cdot\mathbf{V}^{(1)}\cdot\underline{U}^{(1)}(2)+$ $\mathbf{V}^{(2)}\cdot\underline{U}^{(0)}(2)$ |
| $n=3$ | $\mathbf{A}\odot\mathbf{D}^{(3)}$ | $\underline{0}$ | $\mathbf{V}^{(3)}\cdot\underline{h}$ | $\mathbf{V}^{(0)}\cdot\underline{U}^{(3)}(1)+$ $3\cdot\mathbf{V}^{(1)}\cdot\underline{U}^{(2)}(1)+$ $3\cdot\mathbf{V}^{(2)}\cdot\underline{U}^{(1)}(1)+$ $\mathbf{V}^{(3)}\cdot\underline{U}^{(0)}(1)$ | $\mathbf{V}^{(0)}\cdot\underline{U}^{(3)}(2)+$ $3\cdot\mathbf{V}^{(1)}\cdot\underline{U}^{(2)}(2)+$ $3\cdot\mathbf{V}^{(2)}\cdot\underline{U}^{(1)}(2)+$ $\mathbf{V}^{(3)}\cdot\underline{U}^{(0)}(2)$ |

Table 4.1: *First elements of the coefficient vector (rate and impulse reward)*

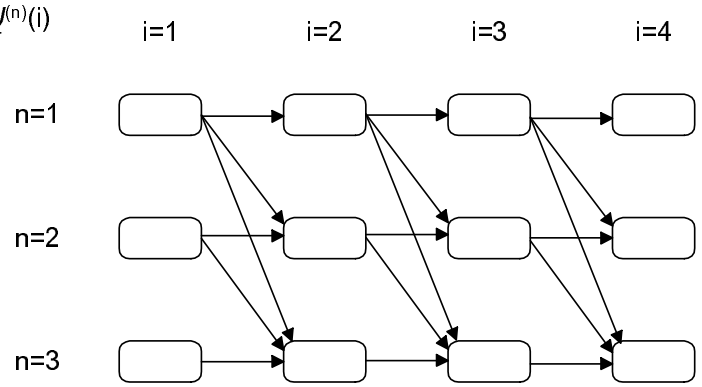



Figure 4.6: *The dependency structure of iteration steps (rate and impulse reward)*

An accuracy control is proposed to eliminate the infinity sum from (4.11). To evaluate the error incurred by applying a finite summation instead of (4.11) the following vector norms are introduced

$$d_1=\max_j\ [(\mathbf{A}\odot\mathbf{D}^{(1)}+\mathbf{S})\cdot\underline{h}]_j \quad \text{and} \quad d_n=\max_j\ [(\mathbf{A}\odot\mathbf{D}^{(n)})\cdot\underline{h}]_j\ ,\quad n\geq 2\ .$$

The norm of $\underline{U}^{(n)}(i)$ is upper bounded by $u^{(n)}(i)\geq\max_j\ [\underline{U}^{(n)}(i)]_j$, which can be calculated iteratively in a similar manner like $\underline{U}^{(n)}(i)$

$$u^{(n)}(i)=\begin{cases}1\ , & if\ \ i\geq 0, n=0\ ,\\ 0\ , & if\ \ i=0, n\geq 1\ ,\\ \displaystyle\sum_{k=0}^{n}\binom{n}{k}d_k\cdot u^{(n-k)}(i-1)\ , & if\ \ i\geq 1, n\geq 1\ .\end{cases} \tag{4.13}$$

Let $a_n=\max\limits_{\ell\in\{1,\ldots,n\}}\ d_\ell$ be the largest of the norms $d_1,\ldots,d_n$. From (4.13) it can be seen that

$$\max_j\ [\underline{U}^{(n)}(i)]_j\leq u^{(n)}(i)\leq a_n(a_n 2^n)^{i-1}\ . \tag{4.14}$$

Theorem 4.6 gives a procedure of accuracy control based on (4.11) and (4.14). I have published the theorem with its proof in [C3].

**Theorem 4.6** *The n-th moment of the accumulated reward can be calculated as a finite sum and an error part, where the maximum allowed error is* $\varepsilon$

$$\underline{m}^{(n)}(t) = \sum_{i=0}^{G-1} \underline{U}^{(n)}(i) \cdot \frac{(qt)^i}{i!} e^{-qt} + \underline{\xi}(G)$$

*where the value of* $G$ *can be determined by*

$$G = \min_{g \in \mathbb{N}} \left( g \;\Big|\; \sum_{i=0}^{g-2} \frac{(qt(1-a_n 2^n))^i}{i!} e^{-qt(1-a_n 2^n)} > 1 - \frac{\varepsilon}{2^{-n} e^{-qt(1-a_n 2^n)}} \right)$$

*and the* $\underline{0} \le \underline{\xi}(G) \le \varepsilon \cdot \underline{h}$ *inequality holds for all the elements of the vectors.*

*Proof:* The error vector $\underline{\xi}(G)$ piecewise satisfies the following inequality

$$\begin{aligned} \underline{\xi}(G) &= \sum_{i=G+1}^{\infty} \underline{U}^{(n)}(i) \cdot \frac{(qt)^i}{i!} e^{-qt} \le \sum_{i=G+1}^{\infty} a_n (a_n 2^n)^{i-1} \frac{(qt)^i}{i!} e^{-qt} \cdot \underline{h} \\ &= 2^{-n} e^{-qt(1-a_n 2^n)} \sum_{i=G+1}^{\infty} \frac{(qt a_n 2^n)^i}{i!} e^{-qt a_n 2^n} \cdot \underline{h} \; . \end{aligned}$$

From which the theorem comes. □

### 4.4.2 Moments of the completion time

The concept of randomization is used to avoiding numerical instabilities. We introduce the following notations

$$\mathbf{H} = \frac{\mathbf{T}}{z} + \mathbf{I} \quad \text{and} \quad \mathbf{L} = \frac{\mathbf{F}}{z\ f}$$

where $z = \max_{i,j \in \mathcal{G}} (|t_{ij}|)$ and $f = \max_{i \in \mathcal{G}}(r_i)/z$. By this definition $\mathbf{H}$ is a stochastic matrix and $\mathbf{S}$ is a diagonal matrix such that $0 \le s_{i,i} \le 1, \forall i \in \mathcal{S}$. Using these matrices

$$\underline{C}^{\sim}(s,w) = \mathbf{\Gamma} \cdot e^{(\mathbf{T} - s\mathbf{F})w} \cdot \underline{h} = \mathbf{\Gamma} \cdot e^{(\mathbf{H} - sf\mathbf{S})zw} \ \cdot \ \underline{h}\ e^{-zw} \; . \tag{4.15}$$

I have published the following theorem with its proof in [C1].

**Theorem 4.7** *The n-th moment of the completion time can be computed using only matrix-vector multiplications and saving only vectors of size* $\#\mathcal{G}$ *as follows*

$$\underline{s}^{(n)}(w) = n!\ f^n \cdot \mathbf{\Gamma} \cdot \ \sum_{i=0}^{\infty} \underline{U}^{(n)}(i) \frac{(zw)^i}{i!} e^{-zw}$$

*where the coefficient vector* $\underline{U}^{(n)}(i)$ *is defined as*

$$\underline{U}^{(n)}(i) = \begin{cases} \underline{0} \ , & \text{if} \quad i = 0, n \geq 1 \ , \\ \underline{h} \ , & \text{if} \quad i \geq 0, n = 0 \ , \\ \mathbf{H} \cdot \underline{U}^{(n)}(i-1) + \mathbf{L} \cdot \underline{U}^{(n-1)}(i-1) \ , & \text{if} \quad i \geq 1, n \geq 1 \ . \end{cases}$$

*Proof:* Starting from (4.15) the proof of Theorem 4.7 follows the same pattern as the proof of Theorem 4.4. □

The following theorem provides a global error bound of the procedure. I have published the theorem with its proof in [C1].

**Theorem 4.8** *The n-th moment of the completion time can be calculated as a finite sum and an error part, where the maximum allowed error is* $\varepsilon$

$$\underline{s}^{(n)}(w) = n! \ f^n \cdot \mathbf{\Gamma} \cdot \ \sum_{i=0}^{G-1} \underline{U}^{(n)}(i) \frac{(zw)^i}{i!} e^{-zw} + \underline{\xi}(G)$$

*where the value of* $G$ *can be determined by*

$$G = \min_{g \in \mathbb{N}} \ \left( g \ \Big| \sum_{i=0}^{g-2} \frac{(zw)^i}{i!} e^{-zw} > 1 - \frac{\varepsilon}{(zw) \ n! \ f^n} \right)$$

*and the* $\underline{0} \leq \underline{\xi}(G) \leq \underline{h} \ \varepsilon$ *inequality holds for all the elements of the vectors.*

*Proof:* By the definition of the matrix $\mathbf{H}$ and the matrix $\mathbf{L}$

$$\underline{0} \leq \mathbf{L} \cdot \underline{h} \ \leq \ \underline{h} \quad \text{and} \quad \underline{0} \leq \mathbf{H} \cdot \mathbf{L} \cdot \underline{h} \ \leq \ \underline{h}$$

hold piece-wise (as all the subsequent vector inequalities), hence $\underline{U}^{(n)}(i)$ is bounded by

$$\underline{0} \leq \underline{U}^{(n)}(i) \leq i \ \underline{h} \ .$$

The error $\underline{\xi}(G)$ incurred when eliminating the tail of the infinite sum is also bounded by

$$\underline{\xi}(G) = n! \ f^n \ \sum_{i=G}^{\infty} \underline{U}^{(n)}(i) \ \frac{(zw)^i}{i!} e^{-zw} \leq n! \ f^n \ \sum_{i=G}^{\infty} \underline{h} \ i \ \frac{(zw)^i}{i!} e^{-zw} = (zw) \ n! \ f^n \ \sum_{i=G-1}^{\infty} \underline{h} \ \frac{(zw)^i}{i!} e^{-zw}$$

which gives the theorem. □

### 4.4.3 Algorithm description

Numerical algorithms are given in the previous sections, with their formal derivations, which are appropriate to analyze *MRM*s with rate and impulse reward. Table 4.2 shows a Pascal like description of the proposed algorithm. The algorithm which calculates moments of the accumulated reward in *MRM* with rate and impulse rewards is presented, because the completion time analysis could be done as an accumulated reward analysis after the model transformation described in previous sections.

**Input**

$\mathbf{Q}$, generator matrix of the underlying *CTMC*
$\mathbf{R}$, diagonal matrix of the rate rewards
$\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(n)}$, moments of impulse rewards
$t$, the time of accumulation
$n$, number of required moments
$\varepsilon$, required precision

**Output**

$\underline{m}^{(1)}(t)$, the first moment of the accumulated reward
$\underline{m}^{(2)}(t)$, the second moment of the accumulated reward
$\vdots$
$\underline{m}^{(n)}(t)$, the $n$-th moment of the accumulated reward

**Memory requirement** (disregarding the input data)

$2n+1$ [Vectors of size $\dim(\mathbf{Q})$]

**Required operations**

$(G-2)\dfrac{n(n+3)}{2}$ [Matrix-vector multiplications]

$(G-2)\dfrac{n(n+3)}{2} + (G-2)n$ [Vector-vector additions]

$G-2$ [Scalar-vector multiplications]

$q := \max_{i,j}(|q_{ij}|)$;
$\mathbf{A} := \mathbf{Q}/q + \mathbf{I}$; $\mathbf{S} := \mathbf{R}/q$;
$d_1 := \max_j[(\mathbf{A} \odot \mathbf{D}^{(1)} + \mathbf{S}) \cdot \underline{h}]_j$;
**For** $k := 2$ **To** $n$ **Do**
  $d_k := \max_j[(\mathbf{A} \odot \mathbf{D}^{(k)}) \cdot \underline{h}]_j$;
$a_n := \max_{l \in \{1,\ldots n\}} d_l$; $C := 2^{-n} e^{-qt(1-a_n 2^n)}$;
$i := 0$; $x := \text{Poisson}\,(0; qt(1-a_n 2^n))$;
**While** $x < (1-\varepsilon/C)$ **Do**
  **Begin**
    $i := i+1$;
    $x := x + \text{Poisson}(i; qt(1-a_n 2^n))$;
  **End**;
$G := i+2$;
$\mathbf{V}^{(0)} := \mathbf{A}$; $\mathbf{V}^{(1)} := \mathbf{A} \odot \mathbf{D}^{(1)} + \mathbf{S}$;
**For** $i := 2$ **To** $n$ **Do**
  $\mathbf{V}^{(i)} := \mathbf{A} \odot \mathbf{D}^{(n)}$;
**For** $j := 0$ **To** $n$ **Do**
  **Begin**
    $\underline{U}^{(j)} := \mathbf{V}^{(j)} \cdot \underline{h}$; $\underline{m}^{(j)} := \underline{U}^{(j)} \text{Poisson}(1; qt)$;
  **End**;

**For** $i := 2$ **To** $G-1$ **Do**
  **For** $j := n$ **DownTo** $0$ **Do**
    **Begin**
      $\underline{U}^{(j)} := \text{Binomial}(j, 0)\ \underline{U}^{(j)}$;
      **For** $k := 1$ **To** $j$ **Do**
        $\underline{U}^{(j)} := \underline{U}^{(j)} + \text{Binomial}(j, k)\ \mathbf{V}^{(k)} \cdot \underline{U}^{(j-k)}$;
      $\underline{m}^{(j)}(t) := \underline{m}^{(j)}(t) + \underline{U}^{(j)}\ \text{Poisson}(i; qt)$;
    **End**;

Table 4.2: *Algorithm : First moments of the accumulated reward (rate and impulse reward)*

## 4.5 Numerical example

In this example, a performance parameter of a dependable equipment is evaluated by the proposed algorithm.

| |
|---|
| **Input** |
| $S = P$ |
| **State space** |
| **States** $= \{\ LS1, LS2, P, G, A, S1, S2\ \}$ |
| **Generator matrix** |
| $(LS1) \rightarrow (P) = 3$ |
| $(P) \rightarrow (LS1) = 0.1$ |
| $(LS2) \rightarrow (G) = 3$ |
| $(G) \rightarrow (LS2) = 0.1$ |
| $(P) \rightarrow (G) = 0.05$ |
| $(P) \rightarrow (S1) = 0.01$ |
| $(S1) \rightarrow (P) = 1$ |
| $(G) \rightarrow (A) = 0.05$ |
| $(G) \rightarrow (S1) = 0.01$ |
| $(A) \rightarrow (S1) = 0.11$ |
| $(S1) \rightarrow (S2) = 0.5$ |
| $(S2) \rightarrow (P) = 0.5$ |
| **Diagonal reward matrix** |
| $(LS1) = 1$ |
| $(LS2) = 1.5$ |
| $(G) = 0.1$ |
| $(A) = 0.2$ |
| $(S1) = 10$ |
| $(S2) = 20$ |
| **Impulse reward matrix** **(mean value)** |
| $(P) \rightarrow (LS1) = 0.2$ |
| $(P) \rightarrow (S1) = 10$ |
| $(G) \rightarrow (LS2) = 0.2$ |
| $(G) \rightarrow (S1) = 10$ |
| $(A) \rightarrow (S1) = 10$ |
| $(S1) \rightarrow (S2) = 5$ |
| **Initial distribution** |
| $(S) = 1$ |

Table 4.3: *High-level description of the MRM of the considered dependable equipment*

A dependable equipment (a computer or a machine producing goods) is operated according to the following rules. Three operational conditions (states) are distinguished, namely `perfect, good` and `adequate`. The system degradation (transition form the state `perfect` to the state `good`, and from the state `good` to the state `adequate`) occurs at a constant rate. The equipment is periodically stopped for a preventive maintenance. If the system state is `adequate` a complete repair is initiated instead of a preventive maintenance. System failure

| rate reward | impulse reward |
|---|---|
| $r_{LS1} = 1$ | $d_{LS1} = 0.2$ |
| $r_{LS2} = 1.5$ | $d_{LS2} = 0.2$ |
| $r_{S1} = 10$ | $d_{S1} = 10$ |
| $r_{S2} = 20$ | $d_{S2} = 5$ |
| $r_G = 0.1$ | |
| $r_A = 0.2$ | |

Table 4.4: *Reward structure of the considered dependable equipment*

can occur in any operational states at the same constant failure rate. A system failure results in a complete repair as well, i.e, at the end of the repair the system is restored to the "as good as new" condition represented by the state `perfect`.

The cost of preventive maintenance has a fix and a time dependent component which can depend on the system state as well. The cost of complete repair also has a fix and a time dependent component, but there is a correlation between the repair time and the associated fix cost. An additional fix cost is assigned to the longer repair periods (e.g., in some cases the complete repair requires the renewal of some special parts that is expensive and time consuming). Some cost can be associated with the system performance degradation in the state `good` and in the state `adequate`.

Assuming all the mentioned state transitions occur at a constant rate the system behavior can be described by an *MRM* with impulse and rate reward as it is shown in Figure 4.7; and the high-level description of the $MRM$ of the considered equipment is given in Table 4.3.

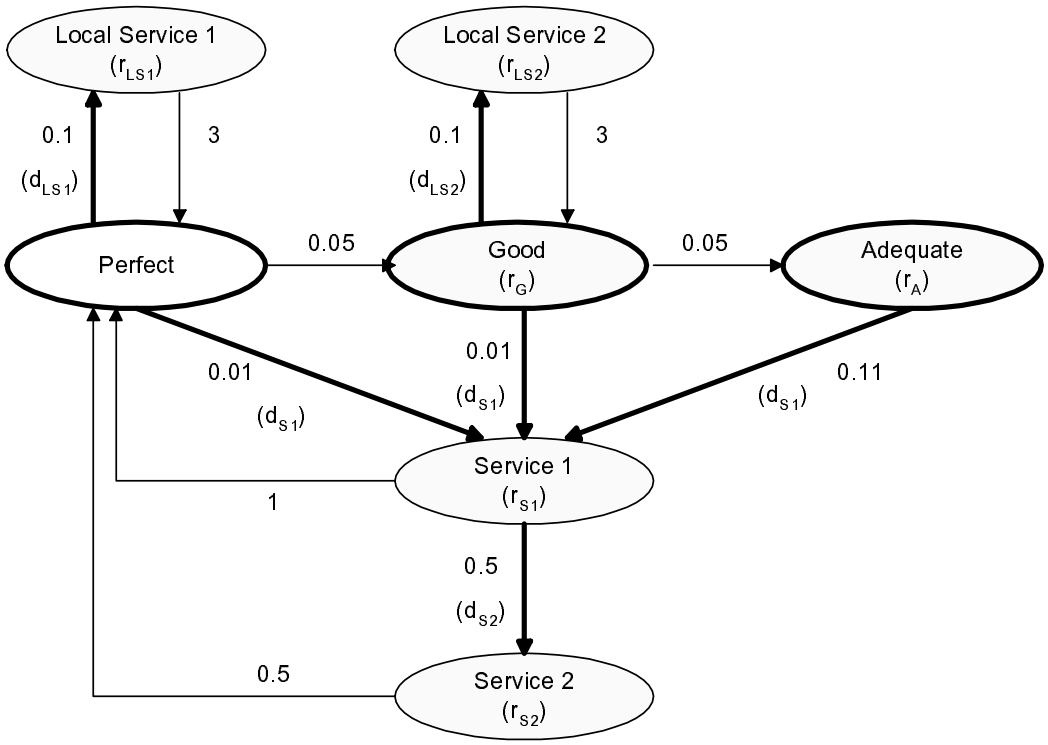



Figure 4.7: *The state structure of the considered dependable equipment*

| | $E\{\mathcal{B}(t)\}$ | $E\{\mathcal{B}(t)^2\}$ | $E\{\mathcal{B}(t)^3\}$ | $E\{\mathcal{B}(t)^4)\}$ |
|---|---|---|---|---|
| $t = 1$ | 0.1918 | 2.6511 | 56.22 | 1404 |
| | 0.3298 | 3.1877 | 65.65 | 1608 |
| | 1.7694 | 28.56 | 613.50 | 15382 |
| $t = 10$ | 3.9198 | 164.29 | 12890 | 1341083 |
| | 7.9019 | 325.25 | 24579 | 2462323 |
| | 21.50 | 1149 | 98501 | 10945246 |
| $t = 100$ | 83.04 | 10271 | 1720854 | 362927485 |
| | 94.95 | 12627 | 2213071 | 482081341 |
| | 106.91 | 15440 | 2869241 | 653702868 |
| $t = 1000$ | 913.24 | 869967 | 863458021 | $8.91 \cdot 10^{11}$ |
| | 925.15 | 892099 | 895609345 | $9.35 \cdot 10^{11}$ |
| | 937.11 | 914768 | 929167155 | $9.81 \cdot 10^{11}$ |
| $t = 10000$ | 9215 | 85283872 | $7.92 \cdot 10^{11}$ | $7.39 \cdot 10^{15}$ |
| | 9227 | 85503760 | $7.95 \cdot 10^{11}$ | $7.43 \cdot 10^{15}$ |
| | 9239 | 85724994 | $7.98 \cdot 10^{11}$ | $7.47 \cdot 10^{15}$ |

Table 4.5: *Example : First four moments of the accumulated reward with different initial state*

Based on the *MRM* of the system the following performance parameters can be evaluated : the interval availability, the operational cost during a time interval and the number of different failures during a time interval. Among these performance parameters the analysis of the operational cost requires the use of impulse and rate rewards at the same time. I have evaluated the first four moments of the operational cost assuming the state transition rates shown in Figure 4.7. All states except the `perfect` have an associated rate reward and the transitions drawn as thick arrows have an associated impulse reward. In each case the impulse reward is assumed to be deterministic. The impulse and rate reward values were used is shown in Table 4.4.
Table 4.5 contains the moments of the operational cost for different time intervals and initial states. The three data of each boxes are calculated assuming the initial state is the `perfect`, the `good`, and the `adequate` state, respectively.

## 4.6 Conclusion

This chapter described an algorithm which determines the moments of reward measures in *MRM*s with rate and impulse reward. The algorithm can deal with large *MRM*s with more than $10^6$ states.

Section 4.1 an Section 4.2 gave a novel approach to derive the Laplace transform domain description of the reward measures. Section 4.3 presented the basic method which derived moments of a reward measure from its Laplace transform domain description. Section 4.4 provided the stable numerical algorithm for moments calculation. Finally, in Section 4.5 a performance parameter of a dependable equipment was evaluated by the proposed algorithm.

# Chapter 5

# Moment based distribution estimation

I address this chapter to describe a *distribution estimation algorithm* which I can use in *MRM*s analysis. My aim was to develop an algorithm which starts from the moments of a reward measure and then provides *lower* as well as *upper bounds* on their distribution. Here I focus on only the algorithm description. I applied my achieved results in [J2, C4, C8, C10].

The problem of inversely characterizing distribution from their moments has been studied for over 100 years. Stieltjes, [Sti1894a, Sti1894b], established necessary and sufficient conditions for the existence of a *real valued*, *bounded* and *non-decreasing* function, for example a distribution function, on the interval $[0, \infty)$ such that its moments match given values. An excellent overview of the moment problem and some variations can be founded in [AK62]. My contribution of this line of work is that I have put strong effort on developing an *applicable numerical algorithm*.

The research activities on this field cover several areas of the application of distribution estimation. Numerous attempts have been made to obtain continuous or discrete distribution from their moments [AK62, Lie, Tagl00, SKR00]. Before the deeper discussion we need to classify the distribution estimation methods. An estimation method can provide mainly two different results.

- A *reconstructed distribution function*. It means that the mission of the estimation method is to make a choice among the distribution functions which match the available information (e.g. if the only information is the mean value then the estimation method has to choose a distribution function which has the given mean value). The maximum entropy principle is widely used in this context [FT97, Tagl00]. The idea of maximum entropy is simply to choose the distribution function from the feasible distribution functions which maximizes some measure of entropy. In [SKR00] the authors derive the maximum and the minimum entropy distribution functions. In contrast to maximum entropy distribution the minimum entropy distribution represents the most biased and least uniform distribution consistent with the available information. In [MH00] the authors used Chebyshev polynomials to obtain an approximation for the investigated probability density function based on its moments. (This work does not study this case.)

- The *domain which contains all feasible distribution functions*. It means practically that the method gives upper and lower bounds on the feasible distribution functions (e.g. if the only information from a non-negative random variable is its mean value ($\mu_1$) then the ideal estimator gives the domain $1 \geq F(x) \geq 1 - \frac{\mu_1}{x}$ based on the Markov-inequality). We can realize that this type of estimation methods give the best and the worst cases for the examined measure.

In the selected *MRM*s analysis approach, first calculating the moments of the examined reward measure and then determining upper and lower bounds on its distribution, the analysis requires an estimation method from the second type, which provides the feasible domain for the considered distribution. It is necessary that a distribution estimation algorithm uses the information coded in the moments totally because calculating new moments results in additive computational effort and memory requirement. For example, the calculation of the first 4 moments instead of the first 2 moments requires approximately 2 times more memory and *CPU* time. So the best way to reduce analysis complexity is to improve the estimation method as far as possible.

## 5.1 The moment problem and its solution

The solution of the moment problem [AK62] provides the basic component of the distribution estimation algorithm which will be presented in the next sections.

*The moment problem* may be stated as follows. Given a sequence of numbers $\{\mu_n\}_{n=0}^{\infty}$, under what conditions is it possible to determine a positive bounded non-decreasing function $F(x)$ in the interval $[a, b]$ such that

$$\mu_n = \int_a^b x^n dF(x) \quad , \qquad \text{for} \quad n = 0, 1, 2, \ldots \quad .$$

Such a sequence is called a moment sequence, and Hausdorff (1921) was the first to obtain necessary and sufficient conditions for a sequence to be a moment sequence [Hau21a, Hau21b].

By varying the definition interval of $F(x)$ we can consider three types of the classical moment problem as

- Hamburger moment problem $\quad x \in (-\infty, \infty)$
- Stieltjes moment problem $\quad x \in [0, \infty)$
- Hausdorff moment problem $\quad x \in [0, 1]$

Before presenting the solution of the moment problem we can introduce Hankel determinant of real numbers as the equation shows below.

$$\alpha(a_a, a_1, \ldots, a_{2n}) = \mathbf{Det} \begin{pmatrix} a_0 & \ldots & a_n \\ \vdots & \ddots & \vdots \\ a_n & \ldots & a_{2n} \end{pmatrix}$$

In the case of Hamburger moment problem the sequence $\{\mu_n\}_{n=0}^{\infty}$ is a moment sequence if and only if

$$\alpha(\mu_0, \mu_1, \ldots, \mu_{2k}) \geq 0 \quad , \qquad \text{for} \quad k = 0, 1, 2, 3, \ldots \quad .$$

In order for a solution exist whose spectrum is not reducible to a finite set of points is necessary and sufficient that

$$\alpha(\mu_0, \mu_1, \ldots, \mu_{2k}) > 0 \quad , \qquad \text{for} \quad k = 0, 1, 2, 3, \ldots \quad .$$

In order for a solution exist whose spectrum consists of precisely $n$ distinct points is necessary and sufficient that

$$\alpha(\mu_0, \mu_1, \ldots, \mu_{2k}) > 0 \quad , \qquad \text{for} \quad k = 0, 1, \ldots, n-1$$

and

$$\alpha(\mu_0, \mu_1, \ldots, \mu_{2k}) = 0 \quad , \qquad \text{for} \quad k = n, n+1, n+2, \ldots \quad .$$

Figure 5.1 shows the feasible value of moments in a particular case. The moments given by a point of the surface determine a discrete distribution. The moments given by the points under the surface can not be derived from any distribution function and the point above the surface can be derived from a distribution function.

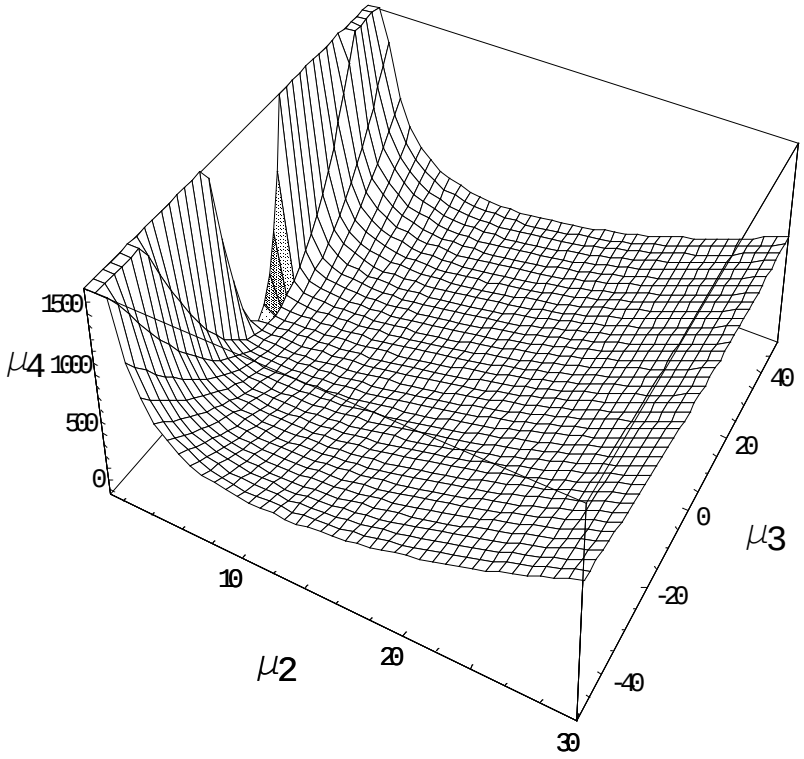



Figure 5.1: *Feasible second, third and fourth moments assuming* $\mu_0 = 1$ *and* $\mu_1 = 1$

Similar manner in the case of Stieltjes moment problem the sequence $\{\mu_n\}_{n=0}^{\infty}$ is a moment sequence if and only if

$$\begin{aligned} \alpha(\mu_0, \mu_1, \ldots, \mu_{2k}) &\geq 0 \quad , \qquad \text{for} \quad k = 0, 1, 2, 3, \ldots \\ \alpha(\mu_1, \mu_1, \ldots, \mu_{2k+1}) &\geq 0 \quad , \qquad \text{for} \quad k = 0, 1, 2, 3, \ldots \quad . \end{aligned}$$

The most complicated case is the Hausdorff moment problem. So in the case of Hausdorff moment problem the sequence $\{\mu_n\}_{n=0}^{\infty}$ is a moment sequence if and only if

$$\begin{aligned} \alpha(\mu_0, \mu_1, \ldots, \mu_{2k}) &\geq 0 \quad , \qquad \text{for} \quad k = 0, 1, 2, 3, \ldots \\ \alpha(\mu_1, \mu_2, \ldots, \mu_{2k+1}) &\geq 0 \quad , \qquad \text{for} \quad k = 0, 1, 2, 3, \ldots \\ \alpha(\mu_0 - \mu_1, \mu_1 - \mu_2, \ldots, \mu_{2k} - \mu_{2k+1}) &\geq 0 \quad , \qquad \text{for} \quad k = 0, 1, 2, 3, \ldots \\ \alpha(\mu_1 - \mu_2, \mu_2 - \mu_3, \ldots, \mu_{2k-1} - \mu_{2k}) &\geq 0 \quad , \qquad \text{for} \quad k = 1, 2, 3, \ldots \quad . \end{aligned}$$

## 5.2 Theoretical background

In fact, in general a distribution is not uniquely determined by its moments. An interesting example is due to C. C. Heyde (e.g. [Fel66, pp 227]). We can construct a class of density functions with parameter $-1 \leq a \leq 1$ as

$$f_a(x) = \frac{1}{x\sqrt{2\pi}} e^{-\frac{1}{2}(\ln x)^2} \left(1 + a \sin(2\pi \ln(x))\right) \quad .$$

Setting $a = 0$ results in the density function of the log-normal distributed random variable.

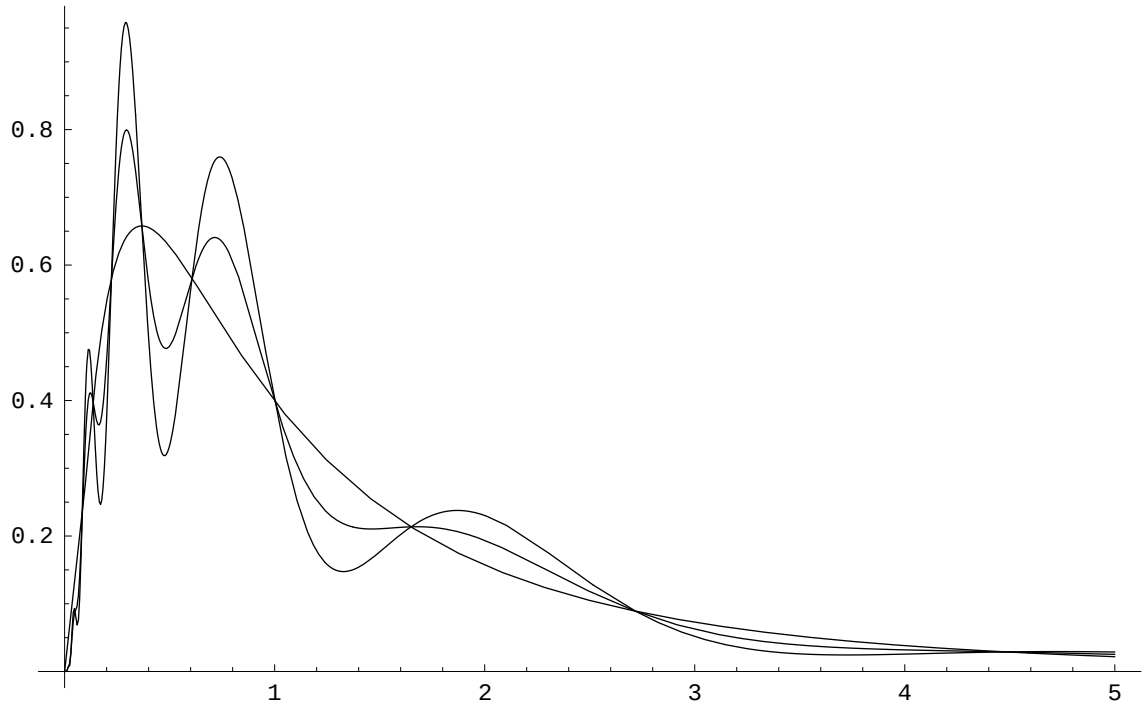



Figure 5.2: *Three density functions with the same moments*

It can be proved that $f_a(x)$ is a probability density and its moments do not depend on the parameter $a$. The first six moments equal $\{\sqrt{e}, e^2, e^{9/2}, e^8, e^{25/2}, e^{18}\}$. Figure 5.2 shows the density function $f_{-1/2}(x)$, $f_{-1/4}(x)$ and the log-normal density function $f_0(x)$. Consequently, we have defined different distributions with the same moment sequence.

There are several methods to check whether the infinity sequence of moments determines an unique distribution function. The best results is a theorem of Carleman to the effect that a distribution on $(-\infty, \infty)$ is uniquely determined by its moments if

$$\sum_{n=1}^{\infty} \frac{1}{\sqrt[2n]{\mu_{2n}}} = \infty$$

that is, if the series on the left diverges. A weaker condition is that a distribution is uniquely determined by its moments whenever the power series

$$\sum_{n=0}^{\infty} \frac{\mu_{2n} t^n}{(2n)!}$$

converges in some interval.

Now, we introduce some notations which we will use to formulate the investigated problem. First, I introduce the set of distribution functions which have the same first moments.

**Definition 3** *The set of distribution functions which have the same moments from $\mu_0$ to $\mu_n$ and their definition interval is $[a,b]$ is denoted by $\mathcal{M}^{[a,b]}_{\{\mu_0,\mu_1,\dots,\mu_n\}}$.*

$$\mathcal{M}^{[a,b]}_{\{\mu_0,\mu_1,\dots,\mu_n\}} = \left\{ F(x) \ : \ \int_a^b x^k dF(x) = \mu_k \ , \quad k = 0,1,\dots,n \right\}$$

Second, the extreme points of $\mathcal{M}^{[a,b]}_{\{\mu_0,\mu_1,\dots,\mu_n\}}$ are defined as follows.

**Definition 4** *The upper and the lower bound of the set $\mathcal{M}^{[a,b]}_{\{\mu_0,\dots,\mu_n\}}$ is defined as*

$$L^{[a,b]}(C;\mu_0,\dots,\mu_n) = \inf_{F\in\mathcal{M}^{[a,b]}_{\{\mu_0,\dots,\mu_n\}}} F(C)$$

$$U^{[a,b]}(C;\mu_0,\dots,\mu_n) = \sup_{F\in\mathcal{M}^{[a,b]}_{\{\mu_0,\dots,\mu_n\}}} F(C)$$

The following section gives a numerical algorithm to determine the considered upper and lower bound.

## 5.3 Algorithm for determining the upper and lower bounds

This section proposes a numerical algorithm to construct $U^{[a,b]}(C;\mu_0,\dots,\mu_n)$ and $L^{[a,b]}(C;\mu_0,\dots,\mu_n)$. The algorithm which determines the value of $L$ and $U$ numerically is depicted in Table 5.1 and Table 5.2 and the required other functions are summarized in Table 5.3-5.7. Broadly speaking calculating a point of the function $U$ and $L$, if we have the first $n$ moments, requires to find all roots of a polynomial of degree about $n/2$ and some algebraic manipulations.

The construction algorithm has two main steps. First, it determines the maximal mass at $C$. Second, the impact of the determined mass will be eliminated from the given moment sequence and using the modified moment sequence a discrete distribution construction is applied. Both steps are based on the solution of the moment problem.

## 5.4 Conclusion

This chapter gave a very short insight into distribution estimation. We identified two types of estimation methods (providing a reconstructed distribution function or providing a domain which contains all feasible distribution functions). This chapter focused on the second type methods. Section 5.1 presented *the moment problem* and its solution. Section 5.2 contains some theoretical results and definitions. Finally, Section 5.3 proposed a distribution estimation algorithm which provides upper and lower bounds on distribution functions which have the same first $n$ moments.

I think this field of my research requires further activity. First of all the theoretical background of the algorithm should be improved.

| **Function name** : UL_Normal |
|---|
| **Input**<br>$\{\mu_0, \mu_1, \ldots, \mu_n\}$ # $n = 2, 4, 6, \ldots$<br>$C$ |
| **Output**<br>$U,\ L$ |
| $\{\mu_0, \mu_1, \ldots, \mu_n\} :=$ Move$(\mu_0, \mu_1, \ldots, \mu_n, 1, -C)$;<br>$\{\mu'_0, \mu'_1, \ldots, \mu'_{n-1}, p\} :=$ Mass_Normal$(\mu_0, \mu_1, \ldots, \mu_n)$;<br>$m := n/2$;<br>$\{x_1, p_1, \ldots, x_m, p_m\} :=$ Discrete_Construction$(\mu'_0, \mu'_1, \ldots, \mu'_{n-1})$;<br>$L := 0; \quad U := 0;$<br>**For** $I := 1$ **To** $m$ **Do**<br>**If** $x_I < 0$ **Then** $L := L + p_I$;<br>$U := L + p$; |

Table 5.1: *Algorithm : Determining values of functions $L$ and $U$ at $C \in (-\infty, \infty)$*

| **Function name** : UL_Positive |
|---|
| **Input**<br>$\{\mu_0, \mu_1, \ldots, \mu_n\}$ # $n = 1, 3, 5, \ldots$<br>$a$<br>$C$ |
| **Output**<br>$U,\ L$ |
| $\{\mu_0, \mu_1, \ldots, \mu_n\} :=$ Move$(\mu_0, \mu_1, \ldots, \mu_n, 1/(C-a), -a/(C-a))$;<br>$\{\mu'_0, \mu'_1, \ldots, \mu'_{n-2}, p, p_A\} :=$ Mass_Positive$(\mu_0, \mu_1, \ldots, \mu_n)$;<br>$m := (n-1)/2$;<br>$\{x_1, p_1, \ldots, x_m, p_m\} :=$ Discrete_Construction$(\mu'_0, \mu'_1, \ldots, \mu'_{n-2})$;<br>**If** $p > 0$ **And** $p_A > 0$ **Then**<br>**Begin**<br>$L := p_A; \quad U := 0;$<br>**For** $I := 1$ **To** $m$ **Do**<br>**If** $x_I < 0$ **Then** $L := L + p_I$;<br>$U := L + p$;<br>**End** |

Table 5.2: *Algorithm : Determining values of functions $L$ and $U$ at $C \in [a, \infty)$*

| **Function name** : Discrete_Construction |
|---|
| **Input**<br>$\{\mu_0, \mu_1, \ldots, \mu_{2n-1}\}$ |
| **Output**<br>$\{x_1, x_2, \ldots, x_n\}$<br>$\{p_1, p_2, \ldots, p_n\}$ |
| **Complexity**<br>Finding the roots of a $n$-th degree polynomial (distinct real roots) |
| $P(\lambda) := \mathbf{Det}\begin{pmatrix} \mu_0 & \mu_1 & \ldots & \mu_n \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n-1} & \mu_n & \ldots & \mu_{2n-1} \\ 1 & \lambda & \ldots & \lambda^n \end{pmatrix};$<br>$\{x_1, x_2, \ldots, x_n\} := \mathbf{Roots}(P(\lambda));$<br>$\{p_1, p_2, \ldots, p_n\} := \mathbf{Solve}\left(\left\{\sum_{i=1}^{n} x_i^k \ p_i = \mu_k \ ; \quad k = 0, 1, \ldots, n-1\right\}, \{p_1, p_2, \ldots, p_n\}\right);$ |

Table 5.3: *Algorithm : Discrete distribution construction*

| **Function name** : Move |
|---|
| **Input**<br>$\{\mu_0, \mu_1, \ldots, \mu_n\}$<br>$A$<br>$B$ |
| **Output**<br>$\{\mu'_0, \mu'_1, \ldots, \mu'_n\}$ |
| **For** $I := 0$ **To** $n$ **Do**<br>$\mu_I := \mu_I \ A^I;$<br>**For** $I := 0$ **To** $n$ **Do**<br>**For** $j := 0$ **To** $I$ **Do**<br>$\mu'_I := \mu'_I + \text{Binomial}(I, j) \ \mu_j \ B^{I-j};$ |

Table 5.4: *Algorithm : Distribution moving*

| **Function name** : Alpha |
|---|
| **Input**<br>$\{a_0, a_1, \ldots, a_{2n}\}$ |
| **Output**<br>$x$ |
| $x := \mathbf{Det}\begin{pmatrix} a_0 & \ldots & a_n \\ \vdots & \ddots & \vdots \\ a_n & \ldots & a_{2n} \end{pmatrix};$ |

Table 5.5: *Algorithm : Hankel determinant*

| **Function name** : Mass_Normal |
|---|
| **Input** |
| $\{\mu_0, \mu_1, \ldots, \mu_{2n}\}$ |
| **Output** |
| $\{\mu'_0, \mu'_1, \ldots, \mu'_{2n-1}\}$ |
| $p$ |
| $p := \text{Alpha}(\mu_0, \mu_1, \ldots, \mu_{2n})/\text{Alpha}(\mu_2, \mu_3, \ldots, \mu_{2n});$ |
| $\mu'_0 := \mu_0 - p;$ |
| **For** $I := 1$ **To** $2n$ **Do** |
| $\mu'_I := \mu_I;$ |

Table 5.6: *Algorithm : Determining maximal mass at* 0

| **Function name** : Mass_Positive |
|---|
| **Input** |
| $\{\mu_0, \mu_1, \ldots, \mu_{2n+1}\}$ |
| **Output** |
| $\{\mu'_0, \mu'_1, \ldots, \mu'_{2n-1}\}$ |
| $p$ |
| $p_A$ |
| $X := \textbf{Det}\begin{pmatrix} \mu_1 & \mu_2 & \ldots & \mu_{n+1} \\ \mu_2 - \mu_1 & \mu_3 - \mu_2 & \ldots & \mu_{n+2} - \mu_{n+1} \\ \vdots & \ddots & \ddots & \vdots \\ \mu_{n+1} - \mu_n & \mu_{n+2} - \mu_{n+1} & \ldots & \mu_{2n+1} - \mu_{2n} \end{pmatrix};$ |
| $Y := \textbf{Det}\begin{pmatrix} 1 & 1 & \ldots & 1 \\ \mu_2 - \mu_1 & \mu_3 - \mu_2 & \ldots & \mu_{n+2} - \mu_{n+1} \\ \vdots & \ddots & \ddots & \vdots \\ \mu_{n+1} - \mu_n & \mu_{n+2} - \mu_{n+1} & \ldots & \mu_{2n+1} - \mu_{2n} \end{pmatrix};$ |
| $p := X/Y;$ |
| **For** $I := 0$ **To** $2n + 1$ **Do** |
| $\mu'_I := \mu_I - p;$ |
| $p_A := \text{Alpha}(\mu'_0, \mu'_1, \ldots, \mu'_{2n})/\text{Alpha}(\mu'_2, \mu'_3, \ldots, \mu'_{2n});$ |
| $\mu'_0 = \mu'_0 - p_A;$ |

Table 5.7: *Algorithm : Determining maximal mass at* 1

# Chapter 6

# Modelling Multi-service Environment

This chapter summarizes the main ideas published in my telecommunication related papers [J2, J4, J5, J6, J7, C4, C6, C8]. The general aim of these papers is to provide a method to perform *quantitative analysis* of call level models which support *stream* as well as *elastic* traffic.

## 6.1 Introduction

In recent years there have been significant advances in researching and standardizing mechanisms that are capable of providing service differentiation in the Internet. While there still seems to be a wide span of the methods which aim at providing QoS differentiation between contending flows, it is widely accepted that there is a need for traffic engineering mechanisms which control the access of the different traffic classes to network bandwidth resources. In particular, there is a growing interest in devising *bandwidth sharing* algorithms which can cope with a high utilization in the network and at the same time take into account the different traffic classes' throughput and blocking probability requirements. Recent research results indicate that it is meaningful to exercise call admission control (CAC) even for elastic traffic, because CAC algorithms (and consequently the *blocking* of some arriving flows) provide a means to prevent e.g. TCP sessions from excessive throughput degradation [MR99b, MR99c]. From this perspective it is important to develop models and computational techniques that make analytical studies of the behavior of such future types of networks possible.

Generally the issue of bandwidth sharing should be considered in the context of dynamically arriving and departing flows, which naturally calls for the application of the classical multi-rate loss models. These models have proved useful in the dimensioning and performance evaluation of circuit switched as well as ATM networks. Thus, they provide motivation for extending the applicability of this modelling paradigm to Internet context. Unfortunately, a direct application of the multi-rate models for traffic engineering in the Internet is non-trivial, because:

- By definition, it is not possible to associate a constant bandwidth with elastic services, like the best effort (without minimum rate guarantee) or the "better than best effort" (with minimum rate guarantee) type of services. The bandwidth occupied by the elastic flows depend on the current load on the link and on the scheduling and rate control algorithms applied in the network nodes.

- The notion of blocking, when applied to elastic flows, needs to be reconsidered because an arriving elastic flow might get into service even if at the arrival instant there is no (or very small) bandwidth available.

- For many services, we need to take account of the fact that the actual residency time of the elastic flows depend on the throughput which the flow receives. For instance, an ftp session would last longer if its throughput decreases. (Real-time services' holding time, on the other hand, is insensitive to the throughput, which is the case, for instance, with a flow associated with an adaptive video codec. As specified later, we will refer to these services as adaptive stream services.)

Since we cannot directly use the reservation based multi-rate models, we seek the meaningful extensions so as to allow the inclusion of both QoS-assured and elastic traffic into a common framework. After putting our work into context, in Section 6.3 we consider a single link where flows belonging to three service classes arrive. *Non-adaptive stream* (also called *rigid*) calls require peak bandwidth allocation. *Adaptive* stream calls and *elastic* calls are modelled as being associated with both a peak- and a minimum bandwidth requirement, and they are allowed into service as long as their respective minimum bandwidth requirements are fulfilled. Furthermore, we allow the bandwidth given to these flows to fluctuate in time, depending on the instantaneous available capacity of the link. While the actual residency time of adaptive calls does not depend on the acquired throughput, the elastic calls' holding time is determined by the actual throughput that the flow receives. Specifically, in the case of Poisson arrivals, these assumptions lead naturally to the application of Markov reward models (MRM). We argue that the *completion time* [BT90] of the MRM corresponds to the flow residency times of elastic flows that depend on the throughput. Next, in Section 6.4 we adopt the well known *partial overlap* (POL) link bandwidth sharing policy [SVVP91] to our model. Also in Section 6.4 we introduce the notion of the throughput threshold constraint, which is a constraint on the probability that the user-perceived throughput during the transfer of a file of size $x$ drops below a certain level. Also in this section, we consider a simple yet efficient link capacity sharing method, which allows for the tuning of the blocking probability vs. throughput trade-off for each traffic class.

In Sections 6.5 and 6.6 we are concerned with the computation of blocking probabilities and throughputs, exploring the limitations of our modelling method in terms of the size of the state space. A recent result [J1] allows us to study large state spaces. We find that in order for the dimensioning to take into account the throughput threshold constraint, the steady state analysis of the associated MRM is not sufficient, and therefore we seek methods for finding the higher moments of the completion time. Section 6.7 gives an example of the application of the model.

We conclude by outlining some possibilities for further applications of our model in Internet-related questions.

## 6.2 Related Works

The theory of multi-rate loss models is covered by for instance [Kau81], [Rob96] and [Ros95]. Application examples of this modelling paradigm include those concentrating on routing and call admission algorithms for QoS assured traffic classes in [DM94] and [SVVP91] and also those that are concerned with the optimal sharing of link bandwidth resources as in [CLW95] and in [BM98, MRW98, MMR96]. However, none of these models addresses the issue of applying this model to cases where elastic traffic is also present in the network, as detailed by the three bullet items in the Introduction.

The notion of call admission control for elastic traffic and fairness issues are discussed in a number of publications, see [MR99a], [MR99b], [MR99c] and [Rob98]. In fact, we feel that our present work here is in line with these papers, and extend them by proposing a

computational model to arrive at specific performance measures on the throughputs and the blocking probabilities. The blocking probability vs. throughput tradeoff is also emphasized and directly connected to the issue of charging in [GK99, Kel97].

The extension of the multi-rate model to include elastic services was proposed independently of each other in [BF96] and in [AAT97]. The application of the MRM to compute the mean transfer time of files with exponentially distributed sizes and the blocking probabilities for the complete sharing method and assuming two traffic classes was proposed already in [ABFT97]. Those results have been extended for the partial overlap link allocation strategy ("mixed scenario") in [QBM99], where the authors are concerned with the computation of the blocking probabilities and also of first moment of the transfer time of a file of size $x$.

The impact of pricing on the optimal bandwidth sharing strategies, again assuming two traffic classes is considered in [AAT97] and in [C8].

From a more practical point of view, specifically examining the TCP traffic (which is the predominant example on the elastic traffic class in the Internet), Feng *et al.* find it beneficial to provide a minimum throughput for TCP connections, because in that case the TCP algorithm can be modified such that the "goodput" of TCP connections is much improved [FKSS99].

Our contribution to this line of works is twofold. First, by applying the multi-rate loss framework with three traffic classes and assuming the partial overlap link allocation technique we formulate the trade-off between the blocking probabilities and the throughput as an optimization task. We will find that the we need the higher moments of the file transmission time for any file size $x$ as well. Second, we propose an efficient computational technique to derive numerical results on the single link level with large state spaces. The proposed numerical approach allows to consider models with $\sim 10^6$ states.

## 6.3 The Multi-class Model of a Single Link : Assumptions and Notations

In this section we formulate the Markovian model of a single transmission link serving peak-bandwidth assured (rigid), adaptive stream and elastic traffic classes. In the presentation we restrict ourselves to these three traffic classes, noting that the model can be extended to more general cases.

Similarly to [AAT97] and [QBM99], we will assume that calls of all three classes arrive at the link according to independent Poisson processes. That is, we assume that the arrival process of requests for document transfer on a given network route is Poisson. As pointed out in [MR99b], this process results naturally when a large population of users emits requests independently, each at a relatively low intensity. Poisson statistics at the call (flow) level have been confirmed in observations of Web traffic in [AW96]. We note, however, that refinements of this assumption are the topic of current research, see for instance Section 3.2 of [MR99b].

The system under consideration consists of a transmission link of capacity $C$. Calls arriving at the link belong to one of the following three traffic classes:

- *Non-adaptive stream* or *rigid* traffic class flows are characterized by their peak bandwidth requirement $b_1$, flow arrival rate $\lambda_1$ and departure rate $\mu_1$;

- *Adaptive* stream class flows are characterized by their peak bandwidth requirement $b_2$, minimum bandwidth requirement $b_2^{min}$, flow arrival rate $\lambda_2$ and departure rate $\mu_2$. Although the bandwidth occupied by adaptive flows may fluctuate as a function of the link load, their actual holding time is not influenced by the received throughput throughout

their residency in the system. This is the case for instance with an adaptive video codec, which, in case of throughput degradation decreases the quality of the video images and thereby occupies less bandwidth.

- *Elastic* class flows are characterized by their peak bandwidth requirement $b_3$, minimum bandwidth requirement $b_3^{min}$, flow arrival rate $\lambda_3$, and their *ideal* departure rate $\mu_3$. The ideal departure rate is experienced when the peak bandwidth is available. The actual instantaneous departure rate is proportional to the bandwidth of the flows. Note that this class can be further classified into two subclasses. If the minimum accepted bandwidth is 0, then this class is the model of the best effort traffic class. If the minimum accepted bandwidth is greater than zero, then this class corresponds to the "better-than-best-effort" traffic class. A typical example of this class is the file transfer protocol (ftp).

We denote the actual bandwidth associated with a flow of class-2 and class-3 in a given system state with $b_2^r$ and $b_3^r$, both of which vary in time as flows arrive and depart. We will also use the quantity $r_{min} := b_{min}/b$ associated with elastic flows with minimum bandwidth requirements. All three types of flows arrive according to independent Poisson processes, and the (ideal) holding time for the rigid, adaptive and elastic flows are exponentially distributed. As we will see, the moments of the actual holding time of the elastic flows can be determined using the theory of Markov reward processes.

To ensure a given QoS of the different flows (that, in general, differ in their peak and minimum bandwidth, i.e. $b_2 \neq b_3$, $b_2^{min} \neq b_3^{min}$) we need to establish some policy which governs the bandwidth sharing among the adaptive stream and elastic classes. For this reason, we define the following bandwidth sharing rules between these two classes.

- If there is enough bandwidth for all flows to get their respective peak bandwidth demands, then class-2 and class-3 flows occupy $b_2$ and $b_3$ bandwidth units respectively.

- If there is a need for bandwidth compression, i.e. $n_1 \cdot b_1 + n_2 \cdot b_2 + n_3 \cdot b_3 > C$, then the bandwidth compression of the flows is such that $r_2 = r_3$, where $r_2 = b_2^r/b_2$ and $r_3 = b_3^r/b_3$, as long as the minimum rate constraint is met for both classes (i.e. $b_2^{min}/b_2 \leq r_2 \leq 1$ and $b_3^{min}/b_3 \leq r_3 \leq 1$).

- If there is still need for further bandwidth compression, but either one of the two classes does not tolerate further bandwidth decrease, (i.e. $r_i$ is already $b_i^{min}/b_i$ for either $i = 2$ or $i = 3$) at the time of the arrival of a new flow, then the service class which tolerates further compression decreases equally the bandwidth occupied by its flows, as long as the minimum bandwidth constraint is kept for this traffic class.

Three underlying assumptions of the above model are noteworthy. First, we assume that both the adaptive and the elastic flows are greedy, in the sense that they always occupy the maximum possible bandwidth on the link, which is the smaller of their peak bandwidth requirement ($b_2$ and $b_3$ respectively) and the equal share (in the above sense) of the bandwidth left for them by the rigid flows (which will depend on the link allocation policy). Second, we assume that all adaptive and elastic flows in progress share proportionally equally the available bandwidth among themselves, i.e. the newly arrived flow and the in-progress flows will be squeezed to the same $r_i$ value. (This assumption actually corresponds to a weighted max-min fair allocation the weights being determined by the peak rates of the flows. By associating a minimum and maximum bandwidth requirements with the flows we in this paper focus on the throughput and blocking probability performance measures.) If a newly arriving flow decreased the flow

bandwidth below $b_2^{min}$ and $b_3^{min}$ (i.e. both the adaptive and the elastic classes were compressed to their respective minima), that flow is not admitted into the system, but it is blocked and lost. Note that all arriving flows are allowed to "compress" the in-service adaptive and elastic flows, as long as the minimum bandwidth constraints are kept. Third, the model assumes that the rate control of the adaptive and elastic flows in progress is ideal, in the sense that an infinitesimal amount of time after any system state change (i.e. flow arrival and departure) these sources readjust their current bandwidth on the link. We realize that the connection between packet level mechanisms and the call level model we consider here needs further research, but it is not the topic of this paper.

It is intuitively clear that the residency time of the elastic flows in this system depends not only on the amount of data they want to transmit (which is a random variable), but also on the bandwidth they receive during their holding times. Similarly, the amount of data transmitted through an adaptive elastic flow depends on the received bandwidth. In order to specify this relationship we define the following quantities:

- $\theta_2(t)$ and $\theta_3(t)$ defines the instantaneous *throughput* of adaptive and elastic flows of at time $t$, respectively, (e.g., if there are $n_1, n_2, n_3$ rigid, adaptive, and elastic flows in the system at time $t$, respectively, the instantaneous throughput are $min(b_2, (C - n_1 \cdot b_1 - n_3 \cdot r_3(n_1, n_2, n_3) \cdot b_3)/n_2)$ and $min(b_3, (C - n_1 \cdot b_1 - n_2 \cdot r_2(n_1, n_2, n_3) \cdot b_2)/n_3))$ for adaptive and elastic flows, respectively. Note that $\theta_2(t)$, and $\theta_3(t)$ are discrete random variables (r.v.) for any $t \geq 0$.

- $\tilde{\theta}_t = \frac{1}{t} \int_0^t \theta_2(\tau) d\tau$ defines the *throughput* of the adaptive flow whose holding time is $t$.

- $\tilde{\theta} = \int_0^\infty \tilde{\theta}_\tau \; dF(\tau) = \mu_2 \int_0^\infty \tilde{\theta}_\tau \; e^{-\mu_2 \tau} \; d\tau$ (r.v.) defines the *throughput* of the adaptive flow, where $F(t)$ is the exponentially distributed flow holding time.

- $T_x = \inf\{t \mid \int_0^t \theta_3(\tau) d\tau \geq x\}$ (r.v.) gives the time it takes for the system to transmit $x$ amount of data through an elastic flow,

- $\hat{\theta}_x = x/T_x$ defines the *throughput* of the elastic flow during the transmission of $x$ data unit. Note that $\theta_x$ is a continuous r.v.

- $\hat{\theta} = \int_0^\infty \hat{\theta}_x \; dG(x) = \mu_3/b_3 \int_0^\infty \hat{\theta}_x \; e^{-x \; \mu_3/b_3} \; dx$ (r.v.) defines the *throughput* of the elastic flow, where the amount of transmitted data is exponentially distributed with parameter $\mu_3/b_3$.

In addition, we associate the maximum accepted blocking probability with all three traffic classes i.e., $B_1^{max}$, $B_2^{max}$ and $B_3^{max}$, respectively, and the minimum accepted throughput $\tilde{\theta}^{min}$, $\hat{\theta}^{min}$ with the adaptive and elastic classes respectively. (The meaning of the minimum accepted throughput and their relation with the random variables, $\tilde{\theta}$, $\hat{\theta}$ are discussed later.)

We refer to the set of the arrival $(\lambda_1, \lambda_2, \lambda_3)$ and departure rates $(\mu_1, \mu_2, \mu_3)$[1], the bandwidths $(b_1, b_2, b_3)$ and minimum bandwidth demands ($b_2^{min}$, $b_3^{min}$), the blocking probabilities ($B_1^{max}$,$B_2^{max}$,$B_3^{max}$) and throughput constraints ($\tilde{\theta}^{min}$, $\hat{\theta}^{min}$) as the *input parameters* of the system.

[1] $\mu_3$ is the maximum departure rate of the elastic class assuming that the bandwidth of the elastic flow equals to $b_3$.

## 6.4 The Partial Overlap Link Allocation Strategy

### 6.4.1 System Description

The system under investigation (with the above assumptions regarding the arrival processes and holding times/transmission requirements) is a continuous time Markov chain (CTMC) whose state is uniquely characterized by the triple $(n_1, n_2, n_3)$, where $n_1$ is the number of rigid flows, $n_2$ and $n_3$ are the number of adaptive and elastic flows in the system, respectively.

It is clear that in order to obtain the performance measure of this system we need to determine the CTMC's generator matrix $\mathbf{Q}$ and its steady state solution, $\underline{P} = \{P_{(n_1,n_2,n_3)}\}$.

We would like to define the link allocation policy such that it is able to provide predefined call blocking probability for the adaptive and for the elastic flows, while it is able to take into account the GoS (blocking probability) constraints for the rigid flows and the minimum throughput constraint for the adaptive and elastic flows. Because of its flexibility (in that it is able to take into account the above constraints) and simplicity (in that the performance measures of interest can be determined even for large systems) we in this paper adopt the *partial overlap, POL* link allocation policy from the multi-rate circuit switched modelling paradigm [SVVP91].

According to the POL policy, the link capacity $C$ is divided into two parts, the $C_{COM}$ common part and the $C_{ELA}$ part, which is reserved for the adaptive and elastic flows only, such that $C = C_{COM} + C_{ELA}$. Under the considered POL policy the number of flows in progress on the link is subject to the following constraints:

$$\begin{aligned} n_1 \cdot b_1 &\leq C_{COM} && (6.1)\\ N_2 \cdot b_2^{min} + N_3 \cdot b_3^{min} &\leq C_{ELA} && (6.2)\\ n_2 &\leq N_2 && (6.3)\\ n_3 &\leq N_3 && (6.4) \end{aligned}$$

where $N_2$ and $N_3$ stand for the maximum number of adaptive and elastic flows in the system (sometimes referred to as the *cut-off* parameter [SM88]) and will be determined later. Note that this policy has only three free parameters, ($C_{COM}$, $N_2$ and $N_3$) which allows for the easy dimensioning of a system with blocking and throughput constraints.

The set of such $(n_1, n_2, n_3)$ triples that satisfies these constraints constitutes the set of *feasible states* of the system which we denote by $\mathcal{S}$. The cardinality of the state space can be determined as:

$$\#\mathcal{S} \leq \left(\frac{C_{COM}}{b_1} + 1\right) \cdot (N_2 + 1) \cdot (N_3 + 1) \qquad (6.5)$$

In (6.1) the adaptive and elastic flows are protected from rigid flows. In (6.2-6.4) the maximum number of adaptive and elastic flows is limited by three constraints. Eq. (6.2) protects the rigid flows, while (6.3-6.4) protect the in-progress adaptive and elastic flows from the arriving new flows, because if too many such flows were admitted into the system then either $\tilde{\theta}$ or $\hat{\theta}$ could decrease below $\tilde{\theta}^{min}$ or $\hat{\theta}^{min}$ respectively. Clearly, the $\theta$ of the $i$-th class can be modified by changing the value of the $N_{ELi}$'s.

The generator matrix, $\mathbf{Q}$, possesses a nice structure, because only transitions between "neighboring states" are allowed in the following sense. Let $q(n_1, n_2, n_3 \rightarrow n_1', n_2', n_3')$ denote the transition rate from state $(n_1, n_2, n_3)$ to state $(n_1', n_2', n_3')$. Then, taking into account the above constraints associated with the proposed POL policy, the non-zero transition rates between the

feasible states are:

$$
\begin{aligned}
q(n_1, n_2, n_3 \to n_1 + 1, n_2, n_3) &= \lambda_1 \\
q(n_1, n_2, n_3 \to n_1, n_2 + 1, n_3) &= \lambda_2 \\
q(n_1, n_2, n_3 \to n_1, n_2, n_3 + 1) &= \lambda_3 \\
q(n_1, n_2, n_3 \to n_1 - 1, n_2, n_3) &= n_1 \cdot \mu_1 \\
q(n_1, n_2, n_3 \to n_1, n_2 - 1, n_3) &= n_2 \cdot \mu_2 \\
q(n_1, n_2, n_3 \to n_1, n_2, n_3 - 1) &= n_3 \cdot r_3(n_1, n_2, n_3) \cdot \mu_3
\end{aligned}
$$

The first three equations represent the state transitions due to call arrivals, while the second three equations represent the transitions due to call departures. The $n_3 \cdot r_3(n_1, n_2, n_3) \cdot b_3$ quantity denotes the total bandwidth of the elastic flows when the system is in state $(n_1, n_2, n_3)$. The generator matrix of the CTMC is constructed automatically based on the above set of equations using the MRMSolve tool [C2, C10].

The POL policy as described above is fully determined by specifying the following parameters: the capacity of the common part, $C_{COM}$, and the maximum number of adaptive and elastic flows, $N_2$ and the $N_3$. We refer to the $C_{COM}$, the $N_2$ and the $N_3$ parameters of the POL policy as the *output parameters* of the system.

For illustration purposes we consider a small system with a link of capacity $C = 7$ and for ease of presentation $n_1 = 1$ is kept fixed, i.e. the available bandwidth for the adaptive and the elastic flows is 6 bandwidth unit. Further more $b_2 = 3$ and $b_3 = 2$. The adaptive and the elastic flows are further characterized by their *minimum* accepted bandwidth, which we set to $b_2^{min} = 1.8$ and $b_3^{min} = 0.8$. The *cut-off* parameters are $N_2 = 2$ and $N_3 = 3$. This setting gives rise to 12 feasible states, out of which there are 5 (gray) states where at least one of the flows is compressed below the peak bandwidth specified by $b_2$ and $b_3$. The Markov chain that describes the system behavior is depicted in Figure 6.1. The states are identified by the number of active flows $(n_1, n_2, n_3)$. The number below the state identifier indicates the bandwidth compression of the adaptive and elastic traffic $(r_2, r_3)$. The last state $(1, 2, 3)$ is the only one where the bandwidth compression of the adaptive and elastic class differs due to the different minimum bandwidth requirement ($r_2^{min} = 0.6$, $r_3^{min} = 0.4$).

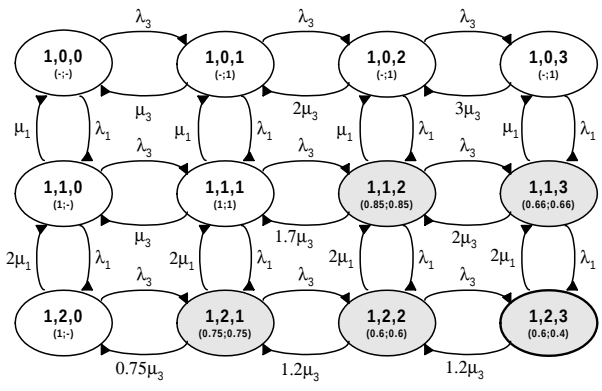



Figure 6.1: *Part of the state space where* $n_1 = 1$ *is kept fixed*

### 6.4.2 Constraints for Determining the Output Parameters of the POL Policy

The POL policy is easy to dimension, since its performance can be tuned by its three output parameters. At the same time it guarantees call level GoS for rigid, adaptive and elastic flows and throughput level for adaptive and elastic services. The GoS of rigid flows is guaranteed by the proper setting of $C_{COM}$. In the case of a change in the adaptive and/or elastic traffic load (i.e. the call arrival intensity or the lengths of the flows), the $N_2$ and $N_3$ parameters have to be adjusted to keep the required throughput and blocking probabilities. We divide the problem of determining the output parameters of the POL policy into two steps. In the first step we determine the minimum required capacity ($C_{COM}$) for rigid flows, that guarantees the required blocking probability:

$$\mathbf{min}\Big\{C_{COM} \ : \ B_1 \leq B_1^{max}\Big\} \tag{6.6}$$

where $B_1$ is the blocking probability of the rigid flows. The Erlang-B formula can be applied for solving this problem. In the second step we determine the maximum number of adaptive and elastic flows ($N_2$,$N_3$) simultaneously present in the system.

In fact, we determine the pairs of maximum number of adaptive/elastic flows (i.e. ($N_2$;$N_3$)) where the system can provide the required throughput and blocking probabilities. It is intuitively clear that if we increase the maximum number of adaptive flows ($N_2$) the blocking probability of adaptive flows ($B_2$) decreases and its throughput decreases. Moreover, unfortunately, changing $N_2$ might affect both the blocking probability ($B_3$) and the throughput of the elastic flows and vice-versa.
The following two constraints are considered:

- constraint on the average throughput:

  The ($N_2$;$N_3$) pair fulfills the blocking probability and the throughput constraints if

  $$B_2 \leq B_2^{max}, \ B_3 \leq B_3^{max}, \ E(\tilde{\theta}) \geq \tilde{\theta}^{min}, \ E(\hat{\theta}) \geq \hat{\theta}^{min}$$

  To make a plausible interpretation of this constraint let us assume that the distribution of $\theta$ is fairly symmetric around $E(\theta)$, i.e. the median of $\theta$ is close to $E(\theta)$. In this case the probability that an adaptive or elastic flow obtains less bandwidth than $\theta^{min}$ is around 50%. Users (even with adaptive or elastic traffic) often prefer more informative throughput constraints like the next one.

- constraint on throughput threshold:

  The ($N_2$;$N_3$) pair fulfills the blocking probability and the throughput constraints if

  $$B_2 \leq B_2^{max}, \quad B_3 \leq B_3^{max}$$
  $$Pr(\tilde{\theta}_t \geq \tilde{\theta}^{min}) \geq \varepsilon_2, \ \forall t; \ Pr(\hat{\theta}_x \geq \hat{\theta}^{min}) \geq \varepsilon_3, \ \forall x;$$

  This throughput threshold constraint requires that the throughput of adaptive and elastic flows be greater than $\tilde{\theta}^{min}$ and $\hat{\theta}^{min}$ with predefined probabilities $\varepsilon_2$ and $\varepsilon_3$ independent of the associated service requirements ($x$) or holding times ($t$). Hence, if the (input) parameter $\theta^{min}$ is much less than $E(\theta)$ then this second constraint is much more informative for the user about the expected minimum level of the elastic flow throughput.

In the case of applying the throughput threshold constraint $\varepsilon_2$ and $\varepsilon_3$ are also input parameters of the model[2].

In general, $N_2$ and $N_3$ have to increase to fulfill the blocking probability constraints and $N_2$ and $N_3$ have to decrease to fulfill the throughput constraints. Depending on the model parameters and the bounds it can occur that the constraints cannot be satisfied at the same time, which means that the link is overloaded with respect to the GoS and QoS requirements.

## 6.5 Analysis of Call Blocking Probabilities

The call blocking probabilities are obtained from the steady state distribution ($\underline{P}$) of the CTMC specified by its generator matrix $\mathbf{Q}$. Considering the model size of practically interesting cases iterative analysis methods are applicable for steady state analysis [Ste94]. Iterative methods begin from an initial guess and produce a sequence of intermediate results, which converge to the solution. The number of required iteration steps to achieve a given precision depends on the model properties, the applied iterative scheme and the initial guess. We applied the Gauss-Seidel algorithm for the iteration and an initial guess that is fairly close to the solution. The initial guess is computed utilizing the fact that the system with only non-adaptive and adaptive traffic classes can be closely approximated by a product form solution [J2].

Based on the steady state distribution of the CTMC, the call blocking probabilities of the different classes are obtained as the sum of the steady state probability of blocking states[3].

## 6.6 Analysis of Throughput Measures of Elastic Flows

### 6.6.1 Average Throughput Constraint

The calculation of the average throughput of the adaptive and the elastic flows is straightforward based on the steady state distribution of the CTMC, since

$$E(\tilde{\theta}) = b_2 \cdot \sum_{(n_1,n_2,n_3)\in\mathcal{S}} P_2(n_1,n_2,n_3)\cdot r_2(n_1,n_2,n_3)$$

where the probability that an adaptive flow is compressed to $r_2(n_1,n_2,n_3)$ is

$$P_2(n_1,n_2,n_3) = \frac{n_2 \cdot p_{(n_1,n_2,n_3)}}{\sum_{(n_1',n_2',n_3')\in\mathcal{S}} n_2' \cdot p_{(n_1',n_2',n_3')}}$$

Similarly,

$$E(\hat{\theta}) = \frac{\sum_{(n_1,n_2,n_3)\in\mathcal{S}} n_3 \cdot p_{(n_1,n_2,n_3)} \cdot b_3 \cdot r_3(n_1,n_2,n_3)}{\sum_{(n_1',n_2',n_3')\in\mathcal{S}} n_3 \cdot p_{(n_1',n_2',n_3')}} .$$

[2]In practical cases, the value of $\varepsilon_2$ ($\varepsilon_3$) is between 50% and 100%. Setting $\varepsilon_2$ ($\varepsilon_3$) to 50% provides approximately the average throughput constraint. Higher value of $\varepsilon_2$ ($\varepsilon_3$) gives tighter throughput guarantee. The case when $\varepsilon_2 = 100\%$ ($\varepsilon_3 = 100\%$) is equivalent with setting $b_2^{min}$ equal to $\tilde{\theta}^{min}$ ($b_3^{min}$ to $\hat{\theta}^{min}$).

[3]Blocking states of a given traffic class are those states in which a new arrival of that class would result in an infeasible state.

### 6.6.2 Throughput Threshold Constraint

Unfortunately, it is harder to check the throughput threshold constraint, since neither the distribution nor the higher moments of $\tilde{\theta}_t$ and $\hat{\theta}_x$ can be analyzed based on the steady state distribution of the above studied Markov chain. Hence, a new analysis approach is applied to analyze the system with the throughput threshold constraint.
The throughput threshold constraint on adaptive flows can be check based on the distribution of $\tilde{\theta}_t$ and on the elastic flows based on the distribution of $T_x$, because:

$$Pr\left(\hat{\theta}_x \geq \hat{\theta}^{min}\right) = Pr\left(\frac{x}{T_x} \geq \hat{\theta}^{min}\right) = Pr\left(T_x \leq \frac{x}{\hat{\theta}^{min}}\right) .$$

Since it is computationally too hard to evaluate the distribution of $T_x$ and $\tilde{\theta}_t$ for realistic models, but there are effective numerical methods to obtain their moments, as discussed later, we check the throughput threshold constraint applying moment based distribution estimation methods. The applied estimation method uses the first three moments of $\tilde{\theta}_t$ and $T_x$ and provides upper and lower bounds of their distribution.

### 6.6.3 Customer Tagging and System Behavior During Adaptive/Elastic Traffic Service

The method we follow to evaluate the moments of $\tilde{\theta}_t$ and $T_x$ is based on tagging an adaptive or an elastic flow arriving to the system, and carefully examining the possible transitions from the moment this tagged call enters the system until it leaves the system. The system behavior during the service of the tagged flow can be described by a slightly modified Markov chain. To analyze $\tilde{\theta}_t$ a tagged adaptive flow is considered while to analyze $T_x$ a tagged elastic flow is used.

Here we detail the analysis of a tagged adaptive flow and at the end of this section we consider the analysis of a tagged elastic flow. The system introduced in Section 6.3, is specified by a CTMC over the state space $\mathcal{S}$ with generator matrix $\mathbf{Q}$. The *modified system* used to evaluate $\tilde{\theta}_t$ has the following properties:

- Since we assume that at least the tagged adaptive flow is in the system we exclude states where $n_2 = 0$.

- With each state of the state space there is an associated entrance probability, which is the probability of the event that the modified CTMC starts from that state. When the tagged adaptive flow finds the system in state $(n_1, n_2, n_3)$ it will bring the system into state $(n_1, n_2 + 1, n_3)$ unless $(n_1, n_2, n_3)$ happens to be a blocking state of the tagged adaptive flow.

Let $\{\mathcal{Z}^{2+}(t),\ t \geq 0\}$ be the modified CTMC assuming that the tagged adaptive flow never leaves the system over the finite state space $\mathcal{S}^{2+}$ with generator $\mathbf{Q}^{2+}$. The state space $\mathcal{S}^{2+}$ can be defined as:

$$\begin{aligned} 0 \leq\ & n_1 \cdot b_1 && \leq C_{COM} && (6.7)\\ 1 \leq\ & n_2 && \leq N_2 && (6.8)\\ 0 \leq\ & n_3 && \leq N_3\ . && (6.9) \end{aligned}$$

Indeed, $\mathcal{S}^{2+} = \mathcal{S} \backslash \mathcal{S}_0^{2+}$ where $\mathcal{S}_0^{2+}$ is the states in $\mathcal{S}$ where $n_2 = 0$. The transition rates in $\mathbf{Q}^{2+}$ are closely related to the appropriate rates in $\mathbf{Q}$ and they differ only in (6.10):

$$\begin{aligned}
q^{2+}(n_1, n_2, n_3 \to n_1 + 1, n_2, n_3) &= \lambda_1 \\
q^{2+}(n_1, n_2, n_3 \to n_1, n_2 + 1, n_3) &= \lambda_2 \\
q^{2+}(n_1, n_2, n_3 \to n_1, n_2, n_3 + 1) &= \lambda_3 \\
q^{2+}(n_1, n_2, n_3 \to n_1 - 1, n_2, n_3) &= n_1 \cdot \mu_1 \\
q^{2+}(n_1, n_2, n_3 \to n_1, n_2 - 1, n_3) &= (n_2 - 1) \cdot \mu_2 \\
q^{2+}(n_1, n_2, n_3 \to n_1, n_2, n_3 - 1) &= n_3 \cdot r_3(n_1, n_2, n_3) \cdot \mu_3
\end{aligned} \tag{6.10}$$

The initial probability of the modified Markov chain, $p^{2+}_{(n_1,n_2,n_3)}$, is obtained by considering the system state immediately after the tagged adaptive flow joints the system in steady state. i.e. the probability that the system is in state $(n_1, n_2, n_3)$ after the tagged adaptive flow arrival is proportional to the steady state probability of state $(n_1, n_2 - 1, n_3)$. Hence

$$p^{2+}_{(n_1,n_2,n_3)} = \frac{p_{(n_1,n_2-1,n_3)}}{\sum_{(n'_1,n'_2,n'_3) \in \mathcal{S}^{2+}} p_{(n'_1,n'_2,n'_3)}}$$

Figure 6.2 depicts the modified Markov chain that describes the system behavior during the service of a tagged adaptive flow assuming the same system as in Figure 6.1. The numbers in brackets under the state identifier indicate the bandwidth of the tagged adaptive flow in the given state. The initial probabilities of the states are evaluated based on the steady state probability of the "original" states that are related with the states with dashed arrows.

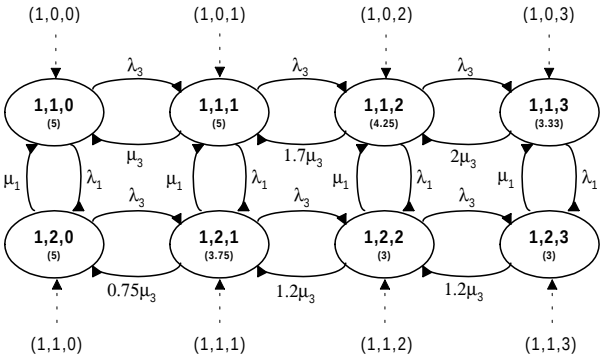



Figure 6.2: *Tagging an adaptive flow*

To obtain the moments of $\tilde{\theta}_t$ a Markov reward model [J1] is defined over $\{\mathcal{Z}^{2+}(t),\ t \geq 0\}$. $\tilde{\theta}_t$ is a random variable which depends on the (random) arrival and departure of the rigid, adaptive and elastic flows as described by $\mathbf{Q}^{2+}$. The reward rate associated with the states of the modified Markov chain represents the bandwidth of the tagged adaptive flow in that state. Let $r^{2+}(n_1, n_2, n_3)$ be the reward rate (the bandwidth of the tagged adaptive flow) in state $(n_1, n_2, n_3)$ and $\mathbf{R}^{2+}$ the diagonal matrix composed by the $r^{2+}(n_1, n_2, n_3)$ entries. $r^{2+}(n_1, n_2, n_3) = r_2(n_1, n_2, n_3) \cdot b_2$, where $r_2(n_1, n_2, n_3)$ is the bandwidth compression in state $(n_1, n_2, n_3)$. This way the dynamics of the number of flows in the system during the service of the tagged adaptive flow is described by the Modified Markov chain and the instantaneous bandwidth of the tagged flow by the instantaneous reward rate. If there are more flows in the

system the bandwidth of the tagged adaptive flow decreases toward $b_2^{min}$ and if there are less flows it increases to $b_2$. The generator matrix $\mathbf{Q}^{2+}$ and the reward matrix $\mathbf{R}^{2+}$ define the Markov reward model that accumulates $t \cdot \tilde{\theta}_t$ amount of reward in the $(0,t)$ interval. It can be interpreted as the reward accumulated in the $(0,t)$ interval represents the amount of data transmitted through the tagged adaptive flow in this interval, i.e. $\tilde{\theta}_t$ = amount of transferred data / $t$.

The tagging of an elastic flow follows the same pattern as the tagging of an adaptive one. The appropriate measures are denoted by $p^{3+}_{(n_1,n_2,n_3)}$, $\mathbf{Q}^{3+}$ and $\mathbf{R}^{3+}$. $T_x$ is the (random) amount of time it takes to transmit $x$ unit of data through the tagged elastic flow. Defining a Markov reward model as before the reward accumulated in $(0,t)$ represents the (random) amount of data transmitted through the tagged flow, hence $T_x$ is the (random) time the Markov reward model takes to accumulate $x$ amount of reward. This measure is commonly referred to as *completion time*.

Having the initial probability distributions $p^{2+}_{(n_1,n_2,n_3)}$, and $p^{3+}_{(n_1,n_2,n_3)}$, the generator matrices $\mathbf{Q}^{2+}$, and $\mathbf{Q}^{3+}$ and the reward matrices $\mathbf{R}^{2+}$ and $\mathbf{R}^{3+}$, we applied the numerical analysis method proposed in [J1] to evaluate the moments of $\tilde{\theta}_t$ and $T_x$, which is applicable for Markov reward models with large state spaces ($\sim 10^6$ states).

### 6.6.4 The Complete Link Allocation Procedure

Finally, the steps of the link allocation procedure is summarized (Figure 6.3). Here we discuss the procedure which maximize the throughput of adaptive and elastic flows.

1) $C_{COM}$: $C_{COM}$ is calculated using Erlang's loss formulae from $\lambda_1, \mu_1, b_1, B_1^{max}$ such that the blocking probability of the rigid flows, $B_1$, is less than $B_1^{max}$. If the obtained $C_{COM}$ is larger than the link capacity, $C$, the link is overloaded by the rigid flows and the requirements can not be satisfied.

2) Initial value of $N_2$: The number of adaptive flows in the system is independent of the other flows as long as the system stays in the non-blocking region, i.e., the bandwidth of the adaptive flows is greater than $b_2^{min}$ such that a new adaptive flow can enter the system. Assuming this independent property is dominant, we calculate the initial value of $N_2$ independent of $N_3$. Actually, we calculate the initial value of $N_2$, when $N_3 = 0$. In this case a reversible, "two dimensional" Markov chain characterize the blocking probabilities, hence it is fast and easy to calculate $N_2$ [J2]. Due to the above mentioned independence of the adaptive and elastic load, $N_2$ hardly changes during the consecutive iterative procedure when $B_2$ and $B_3$ take practically interesting values ($< 15\%$).

3) Initial value of $N_3$: We apply the following heuristic relation for the initial value of $N_3$:

$$N_3 = \frac{C - C_{COM} - N_2 \, \tilde{\theta}^{min}}{\hat{\theta}^{min}} \tag{6.11}$$

4) Iterative refinement of $N_2$ and $N_3$:

   4A) *Calculation of $B_2$ and $B_3$*: The calculation of $B_2$ and $B_3$ with the given $N_2, N_3$ pair ($N_2 > 0, N_3 > 0$) is rather expensive, since it requires the analysis of a (non-reversible) "3 dimensional" Markov chain (Section 6.5).

   4B) *Optimization of $N_3$*: The blocking probability of elastic flows, $B_3$, is a monotone function of $N_3$ for a given fixed $N_2$. In this step we search for the minimal $N_3$ which results in a blocking probability $B_3$ less than $B_3^{max}$.

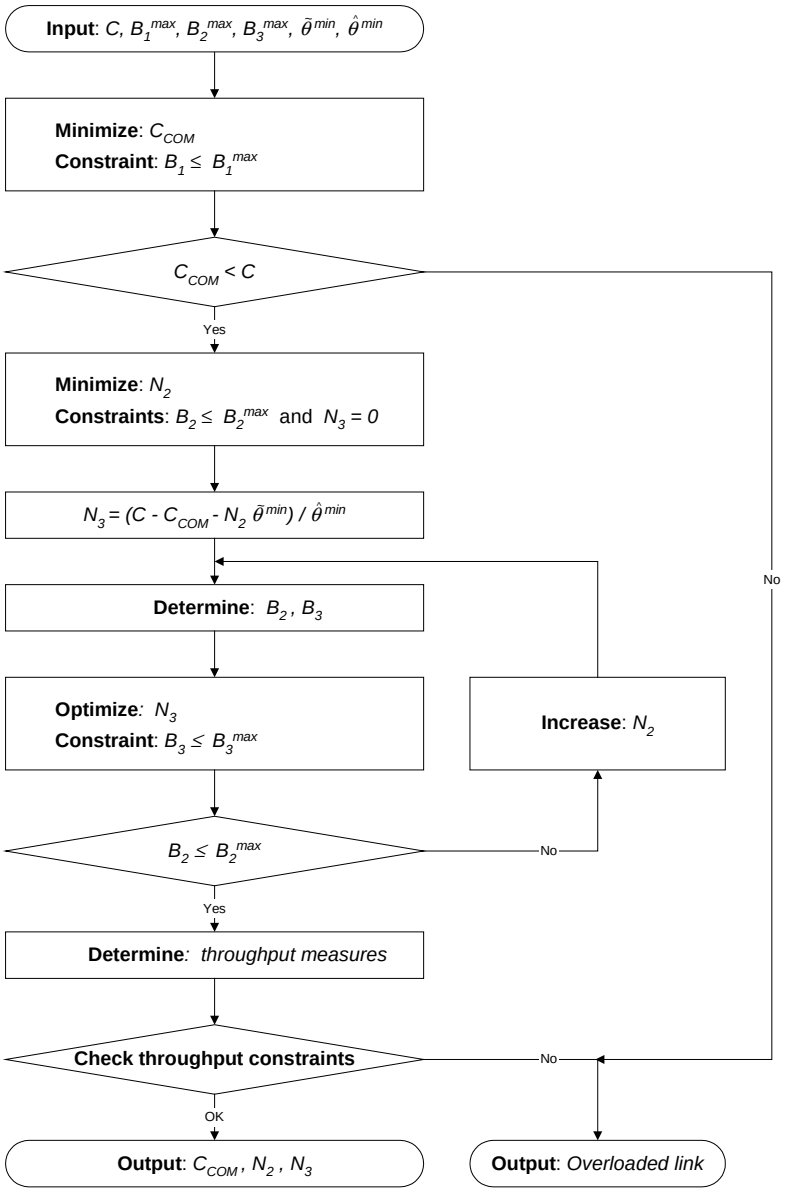



Figure 6.3: *The block diagram of the link allocation procedure*

4C) *Refinement of* $N_2$: If $B_2 > B_2^{max}$ with the given $N_2, N_3$ pair then we increase $N_2$ and go back to step A).

5) Throughput check: The above iterative procedure obtains the minimal $N_2, N_3$ pair which fulfills the blocking constraints. This minimal $N_2, N_3$ pair results in the maximal throughput for the adaptive and elastic flows with the given blocking constraints. Hence a final check of the throughput constraints (Sec. 6.6) decides if the obtained set of output parameters fulfills all requirements or the link is overloaded and the requirements cannot be fulfilled.

### 6.6.5 The Computational Complexity of the Link Allocation Procedure

The only computationally intensive step of the link allocation procedure is the analysis of the "3 dimensional" Markov chain. Unfortunately, this computationally intensive step is repeated in the iterative loop for the analysis of the blocking probabilities as long as a proper $N_2$, $N_3$ pair is obtained. Using the notation $N_{RIG} = C_{COM}/b_1$, in each cycle of the iteration a Markov chain of size $N_{RIG} \times N_2 \times N_3$ has to be analyzed. The blocking probabilities and the average throughput of adaptive and elastic flows are evaluated based on the steady state behavior of this Markov chain. Instead, in case of applying the throughput threshold constraint the reward analysis provided in [J1] has to be completed. Calculating the first $n$ moments of reward measures, the complexity of this procedure (regarding both, the number of matrix-vector multiplication and the memory requirement) is $n$ times more than the transient analysis of the underlying Markov chain. Fortunately, the throughput analysis step is performed only once when we optimize for the throughput of adaptive and elastic flows, as it is discussed in section 6.6.4.

## 6.7 Numerical Examples on the POL Link Allocation Policy

In this Section we present and discuss a numerical example which demonstrates the use of the average throughput- and the throughput threshold constraints.

### 6.7.1 Input Parameters

In the following we will refer to the quantity $S_i = b_i \cdot \lambda_i/\mu_1$ as the class-$i$ offered load to the system, $i = 1 \ldots 3$. The class-1 traffic may represent a voice or a fax over IP application that requires peak allocation. We choose this bandwidth requirement to be our bandwidth unit [BU]. Class-2 traffic may correspond to an adaptive video codec (requiring three times the bandwidth of the voice application). We assume that the application tolerates a temporary throughput degradation down to 20% of the peak bandwidth requirement ($r_2^{min} = 0.2$). Finally, class-3 represents a wide band (three times the bandwidth of the voice application) file transfer (ftp) application where a minimum bandwidth demand is associated with the application, which, also in this case is 20% of the peak data rate demand ($r_3^{min} = 0.2$). Note that for the sake of this example we have chosen the parameters such that the class-wise offered traffic is the same for all three classes ($\lambda_i/\mu_i = 15, i = 1 \cdots 3$), which makes an intuitive interpretation of the results more straightforward. For our time unit [TU], we have chosen the equal holding times of the three traffic classes. Note that we set the throughput constraints of the adaptive and the elastic traffic classes to 2.5 [BU], which is somewhat lower than the maximum required 3 [BU], but higher than the minimum required 0.6 [BU]. Recall that the minimum required bandwidth is the one, which must be ensured for all in-progress class flows at all times, whereas the interpretation of the $\tilde{\theta}^{min}$, $\hat{\theta}^{min}$, $\epsilon_2$ and $\epsilon_3$ values is throughput definition dependent, as discussed in Section 6.6. We will return to these definitions in the subsequent subsections.

### 6.7.2 Determining the $C_{COM}$ Parameter

Recall from Section 6.6 and from the Figure 6.3 that both under the average throughput constraint and under the throughput threshold constraint the first step is to determine the

| System Input Parameter | Interpretation | Value | Unit |
|---|---|---|---|
| $C$ | Link capacity | 100 | [BU] |
| $b_1$ | Rigid traffic class bandwidth demand | 1 | [BU] |
| $b_2$ | Adaptive traffic class maximum bandwidth demand | 3 | [BU] |
| $b_2^{min}$ | Adaptive traffic class minimum bandwidth demand | 0.6 | [BU] |
| $b_3$ | Elastic traffic class maximum bandwidth demand | 3 | [BU] |
| $b_3^{min}$ | Elastic traffic class minimum bandwidth demand | 0.6 | [BU] |
| $\lambda_1 = \lambda_2 = \lambda_3$ | Rigid/Adaptive/Elastic traffic flows' arrival intensity | 15 | 1/[TU] |
| $1/\mu_1 = 1/\mu_2 = 1/\mu_3$ | Rigid/Adaptive/Elastic (ideal) mean holding time | 1 | [TU] |
| $B_1$ | Rigid traffic class blocking probability | 2 | % |
| $B_2$ | Adaptive traffic class blocking probability | 10 (10) | % |
| $B_3$ | Elastic traffic class blocking probability | 10 (10) | % |
| $\tilde{\theta}^{min}$ | Adaptive traffic class throughput constraint | 2.5 (2.0) | [BU] |
| $\hat{\theta}^{min}$ | Elastic traffic class throughput constraint | 2.5 (2.0) | [BU] |
| $\epsilon_2$ | Adaptive traffic class throughput threshold constraint | (90) | % |
| $\epsilon_3$ | Elastic traffic class throughput threshold constraint | (90) | % |

Table 6.1: *The input parameters of the example system, consisting of a single transmission link and three traffic classes (rigid, adaptive and elastic). The link capacity is specified in bandwidth units [BU], the time unit [TU] is explicitly chosen to s.*

common part of the link, $C_{COM}$. From the offered traffic load of the rigid traffic class, applying the Erlang-B formula, it directly follows that $C_{COM}$ must at least be 23 [BU] in order to meet the $B_1 \leq 2\%$ blocking probability constraint. ($B_1 = 1.35\%$ when $C_{COM} = 23$ and $B_1 = 2.1\%$ when $C_{COM} = 22$).

### 6.7.3 Determining the $N_2$ and $N_3$ Parameters with Average Throughput Constraints

The feasible $N_2, N_3$ pairs (cut-off parameters) are limited by the following constraints:

- $B_2 \leq B_2^{max}$, $B_3 \leq B_3^{max}$
- throughput constraints for adaptive and elastic flows
- Eq. (6.2)

The blocking probability constraints define the lower bounds of the feasible $N_2$ and $N_3$ values. An upper bound of the feasible region is defined by Eq. (6.2). If the throughput constraints are too loose, e.g., $\tilde{\theta}^{min} \leq b_2^{min}$ and $\hat{\theta}^{min} \leq b_3^{min}$, Eq. (6.2) limits the feasible $N_2, N_3$ pairs. (In our example, Eq. (6.2) yields that $N_2 + N_3 \leq 128$.) In case of meaningful throughput constraints the feasible $N_2, N_3$ region is further restricted.

The impact of the average throughput constraint and of the maximum blocking probabilities of the adaptive and the elastic classes on the feasible cut-off parameters is shown in Figures 6.4 - 6.7.

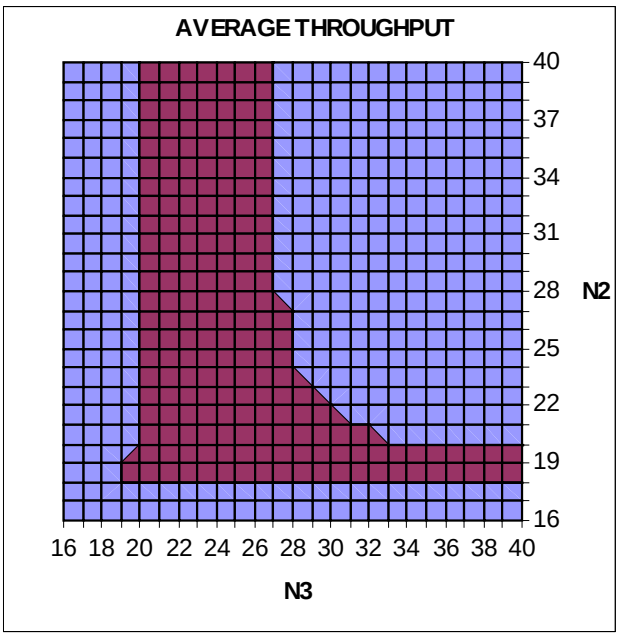



Figure 6.4: *Feasible $N_2, N_3$ region with average throughput constraints ($B_2^{max} = B_3^{max} = 10\%$, $\tilde{\theta}^{min} = \hat{\theta}^{min} = 2.5$)*

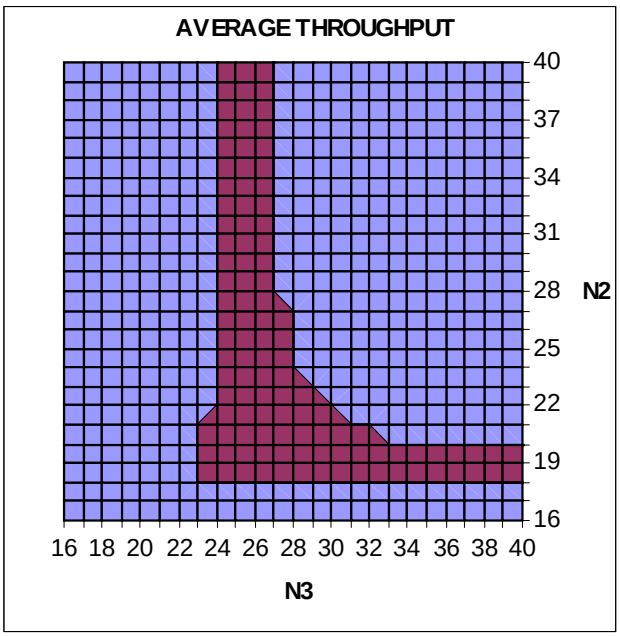



Figure 6.5: *Feasible $N_2, N_3$ region with average throughput constraints ($B_2^{max} = 10\%, B_3^{max} = 5\%$, $\tilde{\theta}^{min} = \hat{\theta}^{min} = 2.5$)*

In Figure 6.4 the minimum average throughput for the adaptive and elastic classes is set as in Table 6.1, $E(\tilde{\theta}) \geq 2.5$ and $E(\hat{\theta}) \geq 2.5$. The set of $N_2$ and $N_3$ pairs that fulfill this throughput constraint and at the same time meet the blocking probability constraints ($B_2 \leq 10\%$ and $B_3 \leq 10\%$) is shown as the "interior" (dark part) of the framed area.

For instance, the pairs ($N_2 = 18$, $N_3 = 19$) and ($N_2 = 28$, $N_3 = 26$) both meet the throughput constraint, but (not surprisingly), the blocking probabilities are minimized under the highest possible cut-off parameters (i.e. under the second cut-off parameter pair). On the other hand, when the blocking probability constraints ($B_2 \leq 10\%$ and $B_3 \leq 10\%$ are "just kept", the throughput values are maximized when keeping the admitted calls to the minimum, in this example under the $N_2 = 18$ and $N_3 = 19$ cut-off parameter pair.

Figure 6.5 and 6.6 investigate the effect of required minimal blocking probabilities. In Figure 6.5 the same area is depicted when reducing the maximum blocking probabilities of the elastic class to $B_3^{max} = 5\%$ and leaving $B_2^{max}$ and the minimum average throughput, $\tilde{\theta}^{min} = \hat{\theta}^{min} = 2.5$, unchanged. Figure 6.6 shows the case when the blocking probability of the adaptive class is reduced to $B_2^{max} = 5\%$ and all the other parameters are as in Figure 6.4. It can be seen that the independent behavior of the lower bounds of $N_2$ and $N_3$ (which we utilize in our link allocation procedure in section 6.6.4) is verified in this example. The maximum blocking probability of the adaptive (elastic) class affects only the lower bound of $N_2$ ($N_3$) and leaves the other boundaries

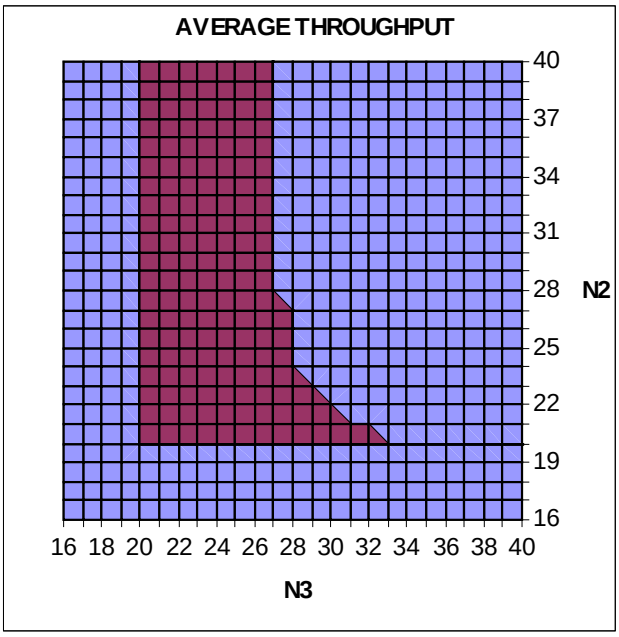



Figure 6.6: *Feasible $N_2, N_3$ region with average throughput constraints ($B_2^{max} = 5\%, B_3^{max} = 10\%$, $\tilde{\theta}^{min} = \hat{\theta}^{min} = 2.5$)*

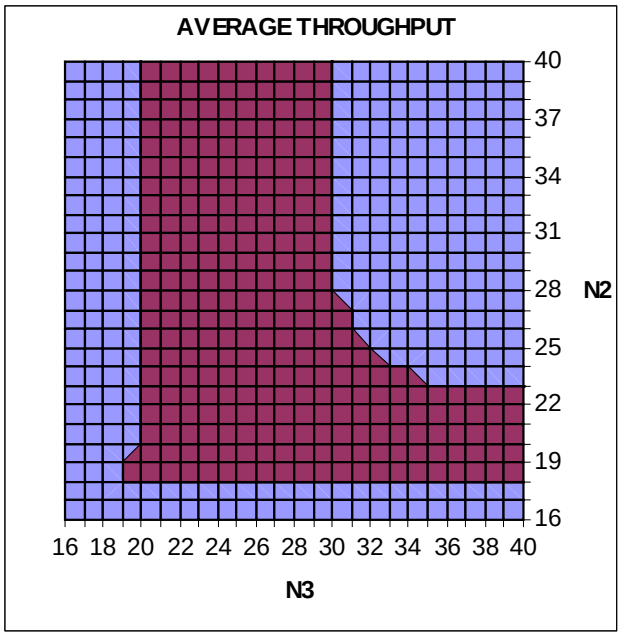



Figure 6.7: *Feasible $N_2, N_3$ region with average throughput constraints ($B_2^{max} = B_3^{max} = 10\%$, $\tilde{\theta}^{min} = 2.5, \hat{\theta}^{min} = 0.6$)*

of the feasible cut-off parameter set unchanged.

Figure 6.7 illustrates the effect of throughput constraints. Starting from the parameters of Figure 6.4, we relaxed the throughput constraint on the adaptive class first. We obtained the same feasible region as in Figure 6.4 even for meaningless low throughput constraint (i.e., $\tilde{\theta}^{min} = b_2^{min}$). It means that, in this example, the upper bound of the feasible cut-off parameter region is determined by the throughput constraint of the elastic class. Starting again from the parameters of Figure 6.4, we relaxed the throughput constraint on the elastic class ($\hat{\theta}^{min} = b_3^{min}$) second. The obtained enlarged feasible region is depicted in Figure 6.7. Thus, the upper boundary in Figure 6.4 comes from the $E(\hat{\theta}) \geq \hat{\theta}^{min} = 2.5$ throughput constraints (independent of the $E(\tilde{\theta}) \geq \tilde{\theta}^{min}$ constraint as long as $\tilde{\theta}^{min} \leq 2.5$), and the higher upper boundary in Figure 6.7 comes from the $E(\tilde{\theta}) \geq \tilde{\theta}^{min} = 2.5$ throughput constraints. The lower one of these two upper boundaries limits the feasible cut-off parameter region, as it is in Figure 6.4.

Our algorithm is capable of determining the "framed" area, that is the set of feasible ($N_2$;$N_3$) pairs including the case when the set is empty. Once this finite set is determined, it is straightforward to select the one pair which is desirable (i.e. maximizing the throughput or minimizing the blocking probabilities). In subsection 6.7.8 we consider an example when we are interested in maximizing the average throughput of adaptive and elastic flows under the blocking probability constraints.

### 6.7.4 Determining the $N_2$ and $N_3$ Parameters with Throughput Threshold Constraints

In our example the adaptive and elastic throughput fluctuate between 0.6 and 3 [BU]. The mean throughput parameter indicates only the average behavior of this fluctuation. For example, the average throughput is 2.5 when the adaptive and elastic flows always receive 2.5 [BU] throughput, and also when they receive the maximum throughput with probability ∼0.8 and

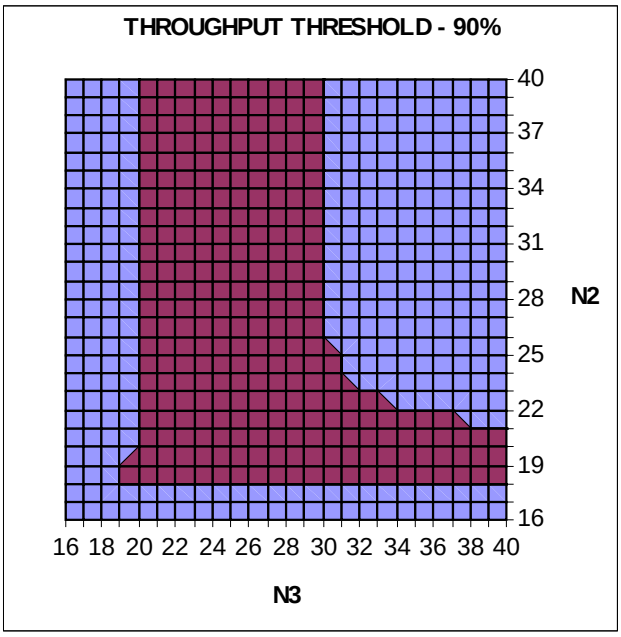



Figure 6.8: *Feasible $N_2, N_3$ region with throughput threshold constraints ($B_2^{max} = B_3^{max} = 10\%$, $\tilde{\theta}^{min} = \hat{\theta}^{min} = 2$, $\epsilon_2 = \epsilon_3 = 90\%$)*

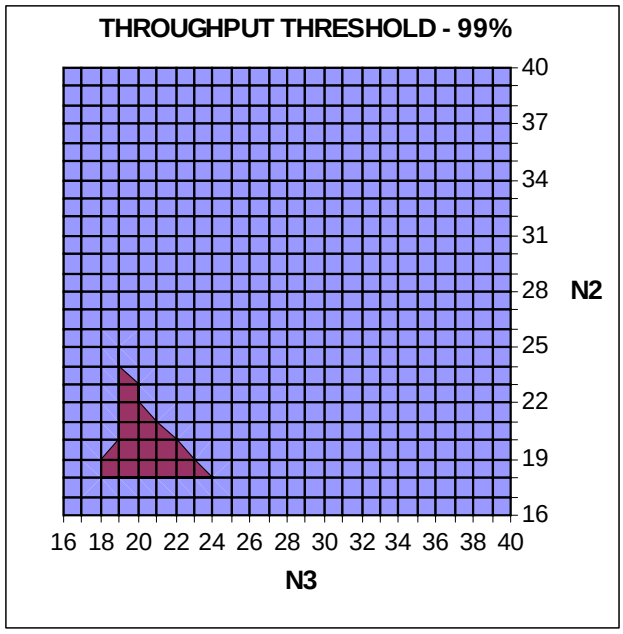



Figure 6.9: *Feasible $N_2, N_3$ region with throughput threshold constraints ($B_2^{max} = B_3^{max} = 10\%$, $\tilde{\theta}^{min} = \hat{\theta}^{min} = 2$, $\epsilon_2 = \epsilon_3 = 99\%$)*

the minimum throughput with probability $\sim$0.2. The real throughput fluctuation is always between these two extreme cases. For some applications it could be important to limit the probability that the flow receives "low" throughput. For example, we may want to require that the throughput of adaptive and elastic flows are higher than 2 [BU] with probability 0.9.

Figure 6.8 specifies the set of $(N_2; N_3)$ pairs which satisfy the same blocking probability constraints for $B_2$ and $B_3$ as above ($B_2 \leq 10\%$, $B_3 \leq 10\%$) and the "90%-threshold constraints" ($Pr(\tilde{\theta} \geq 2) \geq 0.9$ and $Pr(\hat{\theta} \geq 2) \geq 0.9$). With the cut-off parameters inside the framed area the throughput for the adaptive and elastic connections are higher than 2 [BU] with probability 0.9 and the blocking probabilities are less than 10%. Figure 6.9 shows the same $(N_2; N_3)$ set under the "99%-threshold constraint", where we observe that the set of such cut-off parameter pairs that satisfy this constraint "shrinks" significantly as compared to the "90%-threshold" constraint.

### 6.7.5 Blocking Probabilities

In this subsection we study the dependency of the class-wise blocking probabilities on the system output parameters $N_2, N_3$. Figure 6.10 shows how the adaptive traffic class blocking probability depends on the cut-off parameters. As expected, as the maximum number of simultaneously admitted adaptive calls increases, the blocking probability decreases, at the expense of decreasing this class' throughput (Figure 6.12). For instance, at $N_2 = 28; N_3 = 26$, the average throughput constraints are kept (both average throughput values are above 2.5), and the blocking probabilities are minimized ($B_2 < 0.29\%$ and $B_3 < 2\%$).

Two observations are noteworthy. First, we note that $B_2$ is strongly dependent on $N_2$, but basically independent from $N_3$ (Figure 6.10). On the other hand, $B_3$ does depend on both cut-

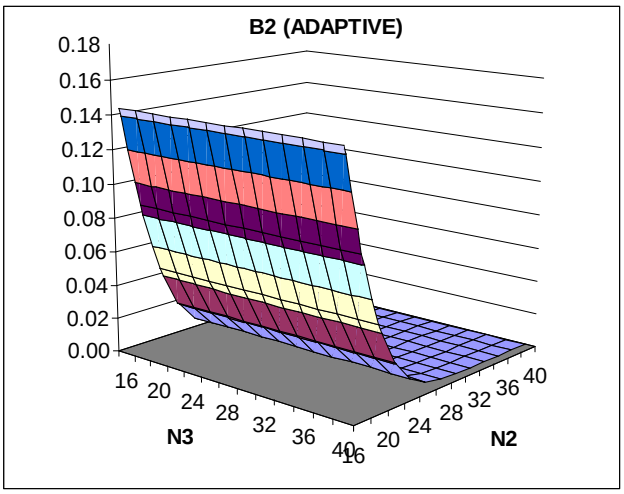



Figure 6.10: *Adaptive class blocking probability as the function of the cut-off parameters*

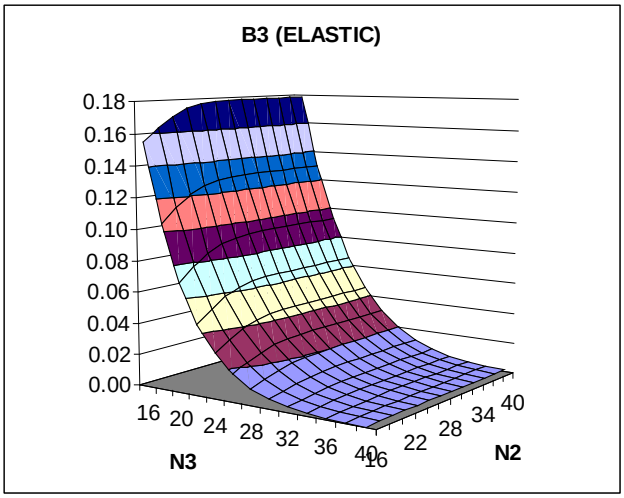



Figure 6.11: *Elastic class blocking probability as the function of the cut-off parameters*

off parameters when $N_2 \leq 22$), as it is in Figure 6.11. This observation verifies the assumptions used in the proposed link allocation procedure for this example.

Secondly, under the given input parameters (i.e., $b_2^{min} = b_3^{min}$) it is clear that *without* the cut-off parameters, $B_2$ would be equal $B_3$, since the blocking states of the underlying Markov chain for the two class are the same. However, as we observe, with the cut-off parameters, the elastic class blocking probability is significantly higher. This significant difference of the blocking probabilities comes from the different nature of the adaptive and elastic flows. The adaptive flows "reduce" their load, i.e., they transmit less data, in case of link saturation and depart from the system at the same rate as the link is not saturated. In contrast, the elastic flows transmit the same amount of data independently of the actual link load (since they tend to increase their holding time if the received throughput decreases). Thus, in the case of link saturation elastic the flows stay much longer in the system, which results in a higher blocking probability for the incoming flows.

### 6.7.6 Average Throughput

With respect to the proposed link allocation procedure the most important feature is the *monotonicity* of the average throughput of adaptive and elastic flows as a function of the cut-off parameters. In figure 6.12 and 6.13 we can see that both average throughput surfaces are monotone with respect to both cut-off parameters. The mean throughput of adaptive and elastic flows are very close, because the same throughput is assigned with the adaptive and elastic flows in each state of the Markov process. The slice difference is due to the different distribution of the number of adaptive and elastic flows in the system.

### 6.7.7 Throughput Thresholds

Figures 6.14 depicts the 99%-threshold plane of adaptive flows' throughput, i.e., the probability that the throughput of adaptive flows, $\tilde{\theta}$, is greater than the given surface point is 99%. Similarly, the 90%-threshold plane of adaptive flows' throughput is shown in Figures 6.15. The significant difference of the two surfaces indicates that a small reduction of the GoS requirements results in significant gain with respect to acceptable system load. The nature of the

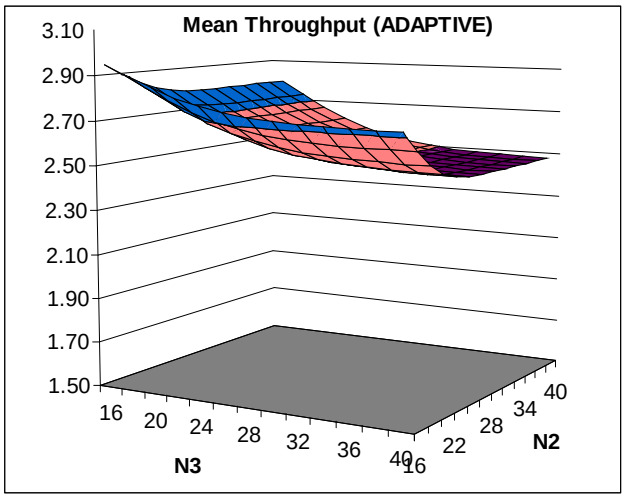



Figure 6.12: *Average throughput of the adaptive class*

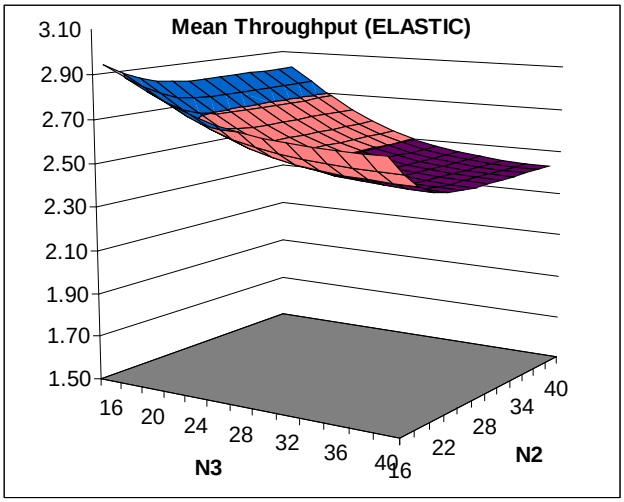



Figure 6.13: *Average throughput of the elastic class*

same threshold planes of elastic flows' throughput is the same, as it is in Figure 6.16 and 6.17. Similarly to the mean throughput surfaces (Figure 6.12 and 6.13) the threshold planes show monotone behavior.

### 6.7.8 The Link Allocation Procedure

We applied the proposed link allocation procedure (section 6.6.4) for the considered example. The first step provided $C_{COM} = 23$ (as it is already mentioned above). In the second step, the analysis of the "two dimensional" Markov chain (with $N_3 = 0$) resulted $N_2 = 18$ and $B_2 = 8.61\%$. Using (6.11), 13 is the initial value of $N_3$. The steps of the iterative refinement of $N_2$ and $N_3$ are provided in Table 6.2. We applied the interval bisection method to find the minimal $N_3$ parameter which still fulfills the blocking constraint. Due to the independence of $B_2$ and $N_3$ the $N_2$ parameter remained unchanged during the iterative refinement. The optimal cut-off parameter pair is $N_2 = 18$ and $N_3 = 19$. The very last step of the iterative analysis checks if $N_3$ can be reduced below 19. Finally, the mean throughput of adaptive and elastic flows with the optimal cut-off parameters are $E(\tilde{\theta}) = 2.82$ and $E(\hat{\theta}) = 2.82$, respectively. The evaluation of this complete link allocation procedure with 6 iteration steps (i.e., analysis of 6 "3 dimensional" Markov chains) required 6 min running time on a 500 Mhz Pentium PC.

| $(N_2, N_3)$ | $(B_2, B_3)$ |
|---|---|
| $(18, 13)$ | $(8.61\%; 26.86\%)$ |
| $(18, 26)$ | $(8.61\%; 2.18\%)$ |
| $(18, 20)$ | $(8.61\%; 7.78\%)$ |
| $(18, 17)$ | $(8.61\%; 13.65\%)$ |
| $\boxed{(18, 19)}$ | $\boxed{(8.61\%; 9.43\%)}$ |
| $(18, 18)$ | $(8.61\%; 11.37\%)$ |

Table 6.2: *The steps of the iterative refinement of the cut-off parameters*

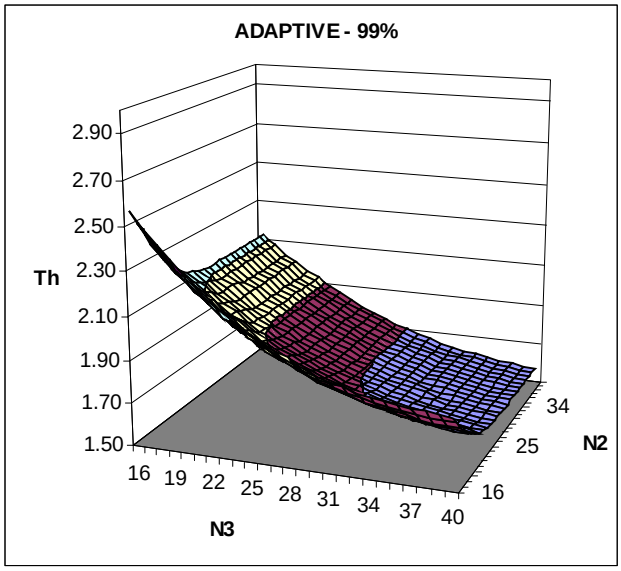



Figure 6.14: *The 99%-threshold plane of the adaptive class*

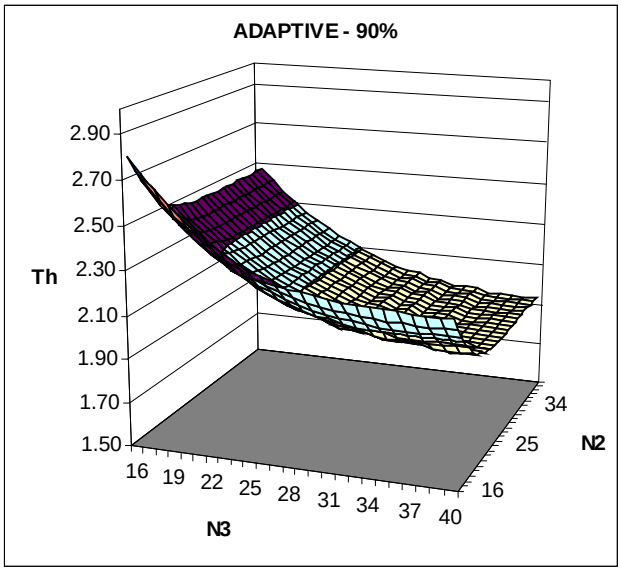



Figure 6.15: *The 90%-threshold plane of the adaptive class*

## 6.8 Conclusion

In this chapter we argued that providing QoS in the Internet necessitates the use of models that allow us to quantitatively study the impact of admission control on flow throughput and blocking. Indeed, there seems to be a growing consensus regarding the necessity of traffic engineering [IETF] and the use of analytical models for the quantifying the relationship between demand, capacity and performance for both streaming and elastic flows [Rob01]. Therefore, we proposed the extension of the classical loss model - successfully applied for the dimensioning of circuit and ATM networks - such that it takes into account the properties of peak allocated stream (*rigid*), adaptive *stream* and *elastic* flows. Our flow differentiation at this abstraction level attempts to capture the fundamental distinguishing characteristics of three broad categories of applications. Under not too limiting assumptions at the call level, we have also shown that a relatively simple link *partial overlap* allocation scheme used in concert with appropriate *cut-off parameters* can be very flexible in the sense that by determining a few parameters (link division and the two cut-off parameters), the system can meet different performance objectives. Specifically, this link capacity division method can take into account blocking probability constraints and maximize the stream and elastic throughputs. Alternatively, it can meet the stream and elastic throughput constraints and minimize the blocking probabilities. During the construction of the model and analyzing the numerical results of a specific example we also found that it is important to realize the (often overlooked) differences between the streaming and the elastic flows in terms of how their actual carried traffic is impacted by the system load. Streaming flows will suffer from quality degradation (but their holding time remains unchanged), while elastic flows will remain for a longer time in the system in case of overload. This distinguishing characteristics of these two class of flows in turn leads to different blocking probability behavior when admission control (by enforcing cut-off parameters) is exercised in the system. (As discussed in details in Section 6.7.5.) Furthermore, we have also argued and shown that the throughput threshold constraint can be a more informative performance measure than the simple average throughput. From the user perspective, it provides more information on the expected quality of e.g. a video session, while from the network engineer it may result in different admission control

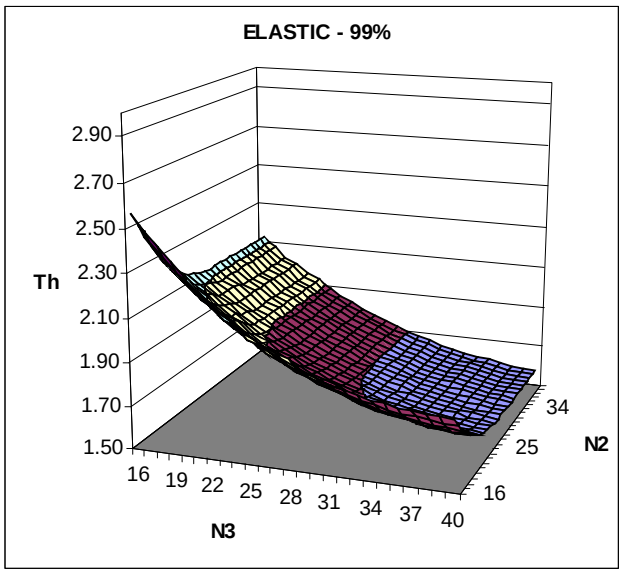



Figure 6.16: *The 99%-threshold plane of the elastic class*

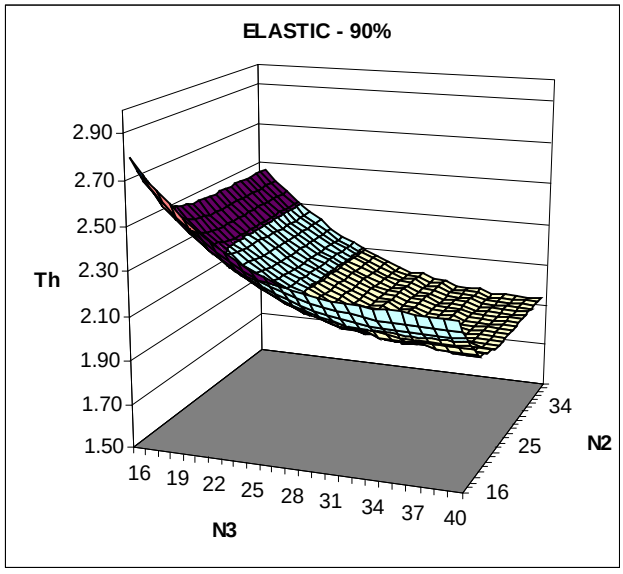



Figure 6.17: *The 90%-threshold plane of the elastic class*

parameters. Thus, we believe that our extension of the multi-rate model and the associated capacity sharing algorithm together with the numerical examples provide some insights and provide arguments for applying traffic engineering methods in IP networks. It is an interesting future research topic how the proposed high level classification can be used for various future applications, including adaptive applications. Also, simulation could be used at the network level to evaluate the behavior of this link sharing method at the network level. Finally, the impact of the fact that real rate control methods (e.g. TCP) and scheduling algorithms can only provide an approximation of our ideal model requires further studies.

# Chapter 7

# Publications


[J1] Gábor Fodor, Sándor Rácz and Miklós Telek. "On Providing Blocking Probability- and Throughput Guarantees in a Multi-service Environment", *International Journal of Communications*, accepted, 2001.

[J2] Sándor Rácz, Miklós Telek and Gábor Fodor, "Link Capacity Sharing Between Guaranteed- and Best Effort Service on an ATM Transmission Link under GoS Constraints", *Telecommunication Systems*, Vol. 17:1,2, 2001.

[J3] Miklós Telek and Sándor Rácz, "Numerical analysis of large Markov reward models", *Performance Evaluation*, Vol. 36&37, pp 95 – 114, August 1999.

[J4] László Ast, Tibor Cinkler, Gábor Fodor, Sándor Rácz and Soren Blaabjerg, "Blocking Probability Approximations and Revenue Optimization in Multi-Rate Loss Networks", *Simulation*, Special Issue on Modeling and Simulation of Computer System and Networks, Journal of SCS, USA, Vol. 68, No. 1, pp 56 – 65, January 1997.

[J5] L. Vajda, J. Mikus, S. Rácz, Cs. Simon, Á Marquetant and Cs. Lukovszki, "Szolgáltatásminőség biztosítása az interneten", *Magyar Távközlés*, 5. szám, pp 23 – 27, 2000.

[J6] Sándor Rácz, Miklós Telek and Gábor Fodor, "A harmadik generációs mobil gerinchálózatok hívásszintű teljesítményelemzśe", *Híradástechnika*, pp 11-17, Vol. LVI, 2. szám, 2001.

[J7] Szabolcs Malomsoky, Sándor Rácz and Szilveszter Nádas, "Connection Admission Control in UMTS Radio Access Networks", *Computer Communications Journal*, submitted, 2001.

[C1] Sándor Rácz and Miklós Telek, "Mission Time Analysis of Large Dependable Systems", *IPDS '00*, In proc., IEEE 4-th International Computer Performance and Dependability Symposium, Chicago, USA, pp 13 – 22, March 2000.

[C2] Sándor Rácz, Béla P. Tóth and Miklós Telek, "MRMSolve: A Tool for Transient Analysis of Large Markov Reward Models", In *Tools 2000*, pp 337 – 340, Springer, LNCS 1786, 2000.

[C3] Sándor Rácz and Miklós Telek, "Performability analysis of Markov reward models with rate and impulse reward", *NSMC '99*, In proc., International Conference on Numerical solution of Markov chains, In M. Silva, B. Plateau and W. Stewart editor, pp 169 – 187, Zaragoza, Spain, September 1999.

[C4] Sándor Rácz, Miklós Telek and Gábor Fodor, "Link Capacity Sharing Between Guaranteed and Best Effort Service on an ATM Transmission Link under GoS Constraints", In *System performance evaluation: Methodologies and applications* (E. Gelenbe, ed), Chapter 5, pp $69-79$, CRC Press, 2000.

[C5] A. Faragó, T. Cinkler, Sándor Rácz, G. Gordos, Á. Horvath, P. Laborczi and Á Magi, "Virtual Path Layout Design", *Networks '98*, In proc., International Telecommunication Network Planing Symposium, Sorrento, Italy, October 1998.

[C6] Tibor Cinkler, Gábor Fodor, Sándor Rácz and Tamás Henk, "Simulative Analysis of End-to-End Call Blocking Evaluation Techniques in Multi-Rate Loss Networks", *ESM '96*, In proc., $10^{th}$ European Simulation Multiconfeence, Budapest, Hungary, June 1996.

[C7] László Ast, Gábor Fodor, Tibor Cinkler and Sándor Rácz, "End-to-End Blocking Probability Approximations for Resource Management in Multi-Rate Loss Networks", *ITC Seminar*, Bangkok, Thailand, November 1995.

[C8] Sándor Rácz, Miklós Telek and Gábor Fodor, "Call Level Performance Analysis of $3^{rd}$ Generation Mobil Core Networks", *ICC 2001*, Helsinki, 2001.

[C9] Sándor Rácz and Miklós Telek, "Analysis of partial loss reward models and its application", *11th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems*, Aachen, 2001.

[C10] Sándor Rácz Árpád Tari and Miklós Telek, "MRMSolve: Distribution Estimation of Large Markov Reward Models", *Tools 2002*, submitted, 2001.

[C11] Szabolcs Malomsoky, Szilveszter Nádas, Sándor Rácz, Attila Báder and András Ivánka, "Performance analysis of switching alternatives in UTRAN", *World Telecommunications Congress 2002*, submitted, 2002.

[C12] Szilveszter Nádas, Sándor Rácz and Szabolcs Malomsoky "On Quality of Service differentiation in UMTS Radio Access Networks", *Globecom 2002*, submitted, 2002.

# Bibliography


[ABC84] M. Ajmone, G. Balbo and G. Conte, "A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems", *ACM Transactions on Computer System*, Vol. 2, pp 93 – 122, 1984.

[AK62] N. I. Akhiezer and M. G. Krein, "Some questions in the theory of moments", *Translations of mathematical monographs*, 2, American Mathematical Society, Providence, R.I., 1962.

[AAT97] E. Altman, D. Artiges and K. Traore, "On the Integration of Best-Effort and Guaranteed Performance Services", *INRIA Research Report No. 3222*, July 1997.

[ABFT97] A. T. Andersen, S. Blaabjerg, G. Fodor and M. Telek, "A Partially Blocking-Queueing System with CBR/VBR and ABR/UBR Arrival Streams", $5^{th}$ *IFIP International Conference on Telecommunications Systems*, Nashville, TN, USA, March 1997.

[AW96] M. F. Arlitt and C. Williamson, "Web Server Workload Characterization: The Search for Invariants", *ACM Sigmetrics*, 1996.

[BBC94] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst, "Template for the solution of linear systems: Building blocks for iterative methods", *SIAM*, 1984.

[Bea78] M. D. Beaudry, "Performance-related reliability measures for computing systems", *IEEE Transactions on Computers*, C-27, pp 540 – 547, 1978.

[BF96] S. Blaabjerg and G. Fodor, "An Extension of the Multi-rate Circuit Switched Loss Model to Model ABR Services in ATM Networks", *IEEE International Conference on Communication Systems*, Singapore, 1996.

[BFAT97] S. Blaabjerg, G. Fodor, A. T. Andersen and M. Telek, "A partially blocking-queueing system with CBR/VBR and ABR/UBR arrival streems", $5^{th}$ *IFIP International Conference on Telecommunications Systems*, Nashville, TN, USA, March 1997.

[BT90] A. Bobbio and K. S. Trivedi, "Computation of the distribution of the completion time when the work requirement is a PH random variable", *Stochastic Models*, Vol. 6, pp 133 – 149, 1990.

[Bob87] A. Bobbio, "The effect of an imperfect coverage on the optimum degree of redundancy of a degradable multiprocessor system", *Reliability'87*, Paper 5B/3, Birmingham, 1987.

[BM98] S. C. Borst and D. Mitra, "Virtual Partitioning for Robust Resource Sharing: Computational Techniques for Heterogeneous Traffic", *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 5, pp 668 – 678, June 1998.

[CLW95] G. Choudhury, K. K. Leung and W. Whitt, "Efficiently Providing Multiple Grade of Service with Protection Against Overloads in Shared Resources", *AT&T Technical Journal*, July/August, pp 50 – 63, 1995.

[CLW94] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Multi dimensional transform inversion with applications to the transient M/G/1 queue", *Annual Applied Probability*, Vol. 4, pp 719 – 740, 1994.

[DG91] L. Donatiello and V. Grassi, "On evaluating the cumulative performance distribution of fault-tolerant computer systems", *IEEE Transactions on Computers*, Vol. 40, No. 11, pp 1301 – 1307, November 1991.

[DM94] Z. Dziong and L. G. Mason, "Call Admission and Routing in Multi-Service Loss Networks", *IEEE Transactions on Communications*, Vol. 42, No. 2, 1994.

[Fel66] W. Feller, "An itroduction to probability theory and its applications", *Wiley*, 1966.

[FKSS99] W. Feng, D. D. Kandlur, D. Saha and K. G. Shin, "Understanding and Improving TCP Performance Over Networks with Minimum Rate Guarantees", *IEEE/ACM Transactions on Networking*, Vol. 7, No. 2, pp 173 – 187, April 1999.

[FNB98] G. Fodor, E. Nordström and S. Blaabjerg, "Revenue Optimization and Fairness Control of Priced Guaranteed and Best Effort Services on an ATM Transmission Link", *IEEE International Conference on Communications, ICC '98*, Vol. 3, pp 1696 – 1705, Atlanta, GA, USA, June 1998.

[FT97] M. Frontini and A. Tagliani, "Entropy-convergence in Stieltjes and Hamburger moment problem".

[GK99] R. J. Gibbens and F. P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network", $16^{th}$ *International Teletraffic Congress*, Edinburgh, UK, June 1999.

[GT87] A. Goyal and A. N. Tantawi, "Evaluation of performability for degradable computer systems", *IEEE Transactions on Computers*, C-36, pp 738 – 744, 1987.

[GDI88] V. Grassi, L. Donatiello and G. Iazeolla, "Performability evaluation of multicomponent fault-tolerant systems", *IEEE Transactions on Reliability*, R-37, pp 216 – 222, 1988.

[GTH85] W. K Grassmann, M. I. Taskar and D. P. Heyman, "Regenerative analysis and steady state distributions for Markov chains", *Operations Research*, 33(5), pp 1107 – 1116 , 1985.

[HNW87] E. Hairer, S. P. Norsett and G. Wanner, "Solving Ordinary Differential Equations I : Nonstiff Problems", *Series in Computational Mathematics*, Springer-Verlag, 1987.

[HW91] E. Hairer and G. Wanner, "Solving Ordinary Differential Equations II : Stiff and Differential-Algebraic Problems", *Series in Computational Mathematics*, Springer-Verlag, 1991.

[Hau21a] F. Hausdorff, "Summationsmethoden und Momentfolgen I", *Math. Z.*, Vol. 9, pp 74 – 109, 1921.

[Hau21b] F. Hausdorff, "Summationsmethoden und Momentfolgen II", *Math. Z.*, Vol. 9, pp 280 – 299, 1921.

[How71] R. A. Howard, "Dynamic probabilistic systems, II. Semi-Markov and decision processes", *Wiley*, 1971.

[IM98] I. C. F. Ipsen and C. D. Meyer, "The idea behind Krylov methods", *Department of Mathematics North Carolina State University Raleigh*, NC 27695 – 8205, USA, 1998.

[IDH86] B. R. Iyer, L. Donatiello and P. Heidelberger, "Analysis of performability for stochastic models of fault-tolerant systems", *IEEE Transactions on Computers*, C-35, pp 902 – 907, 1986.

[IETF] Internet Engineering Task Force, Traffic Engineering Working Group, "Traffic Engineering WG Charter", http://www.ietf.org

[Kah58] W. Kahan, "Gauss-Seidel methods for solving large systems of linear equations", *PhD thesis*, University of Toronto, 1958.

[Kau81] J. Kaufman, "Blocking in a Shared Resource Environment", *IEEE Transactions on Communication*, pp 1474 – 1481, 1981.

[Kel97] F. Kelly, "Charging and Rate Control for Elastic Traffic", *European Transaction on Telecommunications*, Vol. 8, pp 33 – 37, 1997.

[Kle75] L. Kleinrock, "Queuing systems : Theory", *Wiley*, 1975.

[KNT86] V. G. Kulkarni, V. F. Nicola and K. Trivedi, "On modeling the performance and reliability of multi-mode computer systems", *The Journal of Systems and Software*, Vol. 6, pp 175 – 183, 1986.

[LW88] Y. Levy and P. E. Wirth, "A unifying approach to performance and reliability objectives", $12^{th}$ *International Teletraffic Congress*, Torino, 1988.

[Lie] A. Van De Liefvoort, "The moment problem for continuous distributions".

[MR99a] L. Massoulie and J. Roberts, "Bandwidth Sharing : Objectives and Algorithms", *IEEE Infocom '99*, New York, March 1999.

[MR99b] L. Massoulie and J. Roberts, "Bandwidth Sharing and Admission Control for Elastic Traffic", 1999.

[MR99c] L. Massoulie and J. Roberts, "Arguments in Favour of Admission Control for TCP Flows", 1999.

[MH00] I. T. Monroy and G. Hooghiemstra, "On a recursive formula for the moments of phase noise", *IEEE Transactions on Communications*, Vol. 48, No. 6, pp 917 – 920, June 2000.

[MSS84] D. F. McAllister, G. W. Stewart anf W. J. Stewart, "On a Rayleigh-Ritz refinement technique for nearly-uncoupled stochastic matrices", *Linear Algebra and Its Applications*, Vol. 60, pp 1 – 25 , 1984.

[Mey82] J. F. Meyer, "Closed form solution of performability", *IEEE Transactions on Computers*, C-31, pp 648 – 657, 1982.

[MRW98] D. Mitra, M. I. Reiman and J. Wang, "Robust Dynamic Admission Control for Unified Cell and Call QoS in Statistical Multiplexers", *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 5, pp 692 – 707, June 1998.

[MMR96] D. Mitra, J. A. Morrison and K. G. Ramakrishnan, "ATM Network Design and Optimization: A Multirate Loss Network Framework", *IEEE/ACM Transactions on Networking*, Vol. 4, No. 4, pp 531 – 543, August 1996.

[NS96] H. Nabli and B. Sericola, "Performability analysis: A new algorithm", *IEEE Transactions on Computers*, C-45(4), pp 491 – 494, 1996.

[Neu81] M.F. Neuts, "Matrix Geometric Solutions in Stochastic Models", Johns Hopkins University Press, Baltimore, 1981.

[Neu89] M.F. Neuts, "Structured Stochastic Matrices og M/G/1 Type and Their Applications", MARCEL DEKKER, New York, 1989.

[Par82] B. N. Parlett, "A recurrence among the elements of functions of triangular matrices", *Linear Algebra and Its Applications*, Vol. 14, pp 117 – 121 , 1976.

[QBM99] R. N. Queija, J. L. van den Berg and M. R. H. Mandjes, "Performance Evaluation of Strategies for Integration of Elastic and Stream Traffic", *International Teletraffic Congress*, UK, 1999.

[RST89] A. Reibman, R. Smith and K. S. Trivedi, "Markov and Markov reward model transient analysis: an overview of numerical approaches", *European Journal of Operational Research*, Vol. 40, pp 257 – 267, 1989.

[Rob98] J. W. Roberts, "Realizing Quality of Service Guarantees in Multi-service Networks", in *Performance Management of Complex Communication Networks*, eds.: T. Hasegawa, H. Takagi, Y. Takahashi, IFIP., Chapman & Hall, pp 277 – 293, 1998.

[Rob96] J. W. Roberts (ed), "Methods for the Performance Evaluation and Design of Broadband Multi-service Networks", *Published by the Commission of the European Communities, Information Technologies and Sciences, COST 242 Final Report*, 1996.

[Rob01] J. W. Roberts, "Traffic Theory and the Internet", *IEEE Communications Magazine*, Vol. 39, No. 1, pp. 94-99.

[Ros95] K. W. Ross, "Multi-service Loss Models for Broadband Telecommunication Networks", *Springer Verlag London Limited*, ISBN 3 – 540 – 19918 – 7, 1995.

[Saa92] Y. Saad, "Analysis of some Krylov subspace approximations to the matrix exponential operator", *SIAM Journal on Numerical Analysis*, 29(1), pp 209 – 228, 1992

[SM88] Y. D. Serres and L. G. Mason, "A Multi-server Queue with Narrow- and Wide-band Customers and Wide-band Restricted Access", *IEEE Transactions on Communications*, Vol. 36, pp 675 – 684, 1988.

[She82] T. J. Sheskin, "A Markov chain partitioning algorithm for computing steady state probabilities" , *Operations Research*, 33(1), pp 228 – 235 , 1985.

[STR88] R. Smith, K. Trivedi and A.V. Ramesh, "Performability analysis: Measures, an algorithm and a case study", *IEEE Transactions on Computers*, C-37, pp 406 – 417, 1988.

[SG98] E. De Souza e Silva and H.R. Gail, "An algorithm to calculate transient distribution of cumulative rate and impulse based reward", *Stochastic Models*, 14(3), pp 509 – 536, 1998.

[SKR00] M. Srikanth, H. K. Kesavan and P. H. Roe, "Probability density function estimation using the MinMax measure", *IEEE Transactions on System, Man, and Cybernetics*, Vol. 30, No. 1, pp 1 – 7, February 2000.

[Ste94] W. J. Stewart, "Introduction to the Numerical Solution of Markov Chains", *Princeton University Press*, Princeton, New Jersey, ISBN 0 – 691 – 03699 – 3, 1994.

[Ste82] W. J. Stewart, "On computing the stationary probability vector of a network of two Coxian servers", *Applied Probability - Computer Sciencd : The interface*, Vol. 1, pp 275 – 296, 1982.

[Sti1894a] T. Stieltjes, "Reserches sur les fractions continues", *Ann. Fac. Sci. Univ. Toulouse*, Vol. 2, pp J 1 – 122 , 1894.

[Sti1894b] T. Stieltjes, "Reserches sur les fractions continues", *Ann. Fac. Sci. Univ. Toulouse*, Vol. 2, pp A 1 – 47 , 1894.

[SSM87] U. Sumita, J. G. Shanthikumar, and Y. Masuda, "Analysis of fault tolerant computer systems", *Microelectronics and Reliability*, Vol. 27, pp 65 – 78, 1987.

[SVVP91] E. D. Sykas, K. M. Vlakos, I. S. Venieris and E. N. Protonotarios, "Simulative Analysis of Optimal Resource Allocation and Routing in IBCN's", *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 3, 1991.

[Tagl00] Aldo Tagliani, "Existence and stability of a discrete probability distribution by maximum entropy approach", *Applied Mathematics and Computation*, No. 110, pp 105 – 114, 2000.

[TPF98] M. Telek, A. Pfenning and G. Fodor, "An Effective Numerical Method to Compute the Moments of the Completion Time of Markov Reward Models", *Computers and Mathematics with Applications*, Vol. 36, No. 8, pp 59 – 65, 1998.