# Efficient heuristics for simulating rare events in queuing networks

**Tatiana S. Zaburnenko**

PhD dissertation committee

Chairman: Prof. dr. Kees Hoede
Promotor: Prof. dr. ir. Boudewijn R.H.M. Haverkort
Assistant promotor: Dr. ir. Pieter-Tjerk de Boer

Members: Prof. dr. Hans L. van den Berg
Prof. dr. Richard J. Boucherie
Dr. ing. Poul E. Heegaard
Prof. dr. Michel R.H. Mandjes
Dr. Ad A.N. Ridder
Dr. Gerardo Rubino

ISBN 978-90-365-2622-7
ISSN 1381-3617, CTIT PhD thesis series no. 08-111

Center for Telematics and Information Technology,
P.O. Box 217, 7500 AE Enschede, The Netherlands

Copyright © 2008, Tatiana S. Zaburnenko, Enschede, The Netherlands.
All rights reserved. No part of this publication may be reproduced, stored in the retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written approval of the copyright owner. No part of this publication may be adapted in whole or in part without the prior written permission of the author.

# EFFICIENT HEURISTICS FOR SIMULATING RARE EVENTS IN QUEUING NETWORKS

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. W.H.M. Zijm,
on account of the decision of the graduation committee,
to be publicly defended
on Friday January, 25 2008 at 15.00

by

Tatiana Sergeevna Zaburnenko
born on January, 25 1978
in Yaroslavl, Soviet Union

This dissertation has been approved by

prof. dr. ir. Boudewijn R.H.M. Haverkort (promotor)
dr. ir. Pieter-Tjerk de Boer (assistant promotor)

# Abstract / Resumé / Резюме

In this thesis we propose state-dependent importance sampling heuristics to estimate the probability of population overflow in queuing networks. These heuristics capture state-dependence along the boundaries (when one or more queues are almost empty) which is crucial for the asymptotic efficiency of the change of measure. The approach does not require difficult (and often intractable) mathematical analysis or costly optimization involved in adaptive importance sampling methodologies. Experimental results on tandem, parallel, feed-forward, and a 2-node feedback Jackson queuing networks as well as a 2-node tandem non-Markovian network suggest that the proposed heuristics yield asymptotically efficient estimators, sometimes with bounded relative error. For these queuing networks no state-*in*dependent importance sampling techniques are known to be efficient.

In dit proefschrift introduceren we toestandsafhankelijke importance sampling heuristieken om de waarschijnlijkheid van overflow van de totale populatie in netwerken van wachtrijsystemen te schatten. Deze heuristieken hebben de juiste toestandsafhankelijkheid langs de grenzen (als een of meer wachtrijen bijna leeg zijn). Dit is cruciaal voor de asymptotische efficiëntie van de kansmaatverandering. De aanpak vereist geen moeilijke (en vaak ingewikkelde) wiskundige analyse of kostbare optimalisatie zoals in adaptieve importance sampling methodieken. Experimentele resultaten voor tandem, parallel, feed-forward en 2-node feedback Jackson wachtrijnetwerken alsook voor een 2-node tandem non-Markov netwerk suggereren dat de voorgestelde heuristieken asymptotisch efficiënte schatters opleveren, soms met begrensde relatieve fout. Voor deze wachtrijnetwerken zijn geen efficiënte toestands*on*afhankelijke importance sampling technieken bekend.

В данной работе предлагаются эвристические методы получения оценки вероятности переполнения сети с помощью зависимой от состояния сети выборки по важности. Методы отражают зависимость вдоль границ пространства состояний сети (в случае, когда одна или более очереди почти пусты), имеющую решающее значение для асимптотической эффективности замены меры. Подход не требует сложного (и часто неразрешимого) математического анализа и оптимизационных затрат, участвующих в адаптативных методиках нахождения выборки по важности. Результаты экспериментов на тандемных, параллельных, сетях с прямой связью, и 2-х узелных Джексон сетях с обратной связью, а также 2-х узелных тандемных не-Джексон сетях, подтверждают, что предлагаемый эвристический подход дает асимптотически эффективные оценки, в некоторых случаях даже

с ограниченной относительной погрешностью. Для такого рода сетей существование эффективных *не*зависимых от состояния сети выборок по важности до настоящего момента было неизвестно.

# Contents

**1 Introduction** **1**
1.1 Motivation of the work . . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.2 Outline of the thesis . . . . . . . . . . . . . . . . . . . . . . . . . . 2

**2 Importance Sampling** **5**
2.1 Monte Carlo simulation . . . . . . . . . . . . . . . . . . . . . . . . 5
2.2 Techniques for rare event simulation . . . . . . . . . . . . . . . . 6
2.3 Basics of Importance Sampling . . . . . . . . . . . . . . . . . . . 7
2.3.1 Notation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
2.3.2 The optimal change of measure . . . . . . . . . . . . . . . 8
2.3.3 Asymptotic efficiency . . . . . . . . . . . . . . . . . . . . 9
2.3.4 State dependency . . . . . . . . . . . . . . . . . . . . . . 9
2.4 Analytical and heuristic IS techniques . . . . . . . . . . . . . . . 10
2.4.1 Large deviation theory . . . . . . . . . . . . . . . . . . . 10
2.4.2 State-independent heuristics . . . . . . . . . . . . . . . . 12
2.4.3 State-dependent heuristics . . . . . . . . . . . . . . . . . 13
2.5 Adaptive IS techniques . . . . . . . . . . . . . . . . . . . . . . . . 14
2.5.1 Basics of cross-entropy method . . . . . . . . . . . . . . 14
2.5.2 Algorithmic description . . . . . . . . . . . . . . . . . . . 15
2.5.3 State-dependency . . . . . . . . . . . . . . . . . . . . . . 16
2.6 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

**3 State-dependent heuristics for tandem networks** **19**
3.1 Model and notation . . . . . . . . . . . . . . . . . . . . . . . . . . 19
3.2 Two-node tandem networks . . . . . . . . . . . . . . . . . . . . . 20
3.2.1 Motivation of the heuristic . . . . . . . . . . . . . . . . . 20
3.2.2 Time reversal argument . . . . . . . . . . . . . . . . . . . 21
3.2.3 State-dependent heuristic (SDH) . . . . . . . . . . . . . . 23
3.2.4 Improved heuristic (SDHI) . . . . . . . . . . . . . . . . . 25
3.3 Multiple-node tandem networks . . . . . . . . . . . . . . . . . . . 26
3.3.1 State-dependent heuristic (SDH) . . . . . . . . . . . . . . 27
3.3.2 Improved heuristic (SDHI) . . . . . . . . . . . . . . . . . 28
3.4 Performance comparison . . . . . . . . . . . . . . . . . . . . . . . 30
3.4.1 Simulation . . . . . . . . . . . . . . . . . . . . . . . . . . 30
3.4.2 Methods used for performance comparison . . . . . . . . . 32
3.5 Experimental results . . . . . . . . . . . . . . . . . . . . . . . . . 33

3.5.1 Performance for 2 queues in tandem . . . . . . . . . . . . . . . 33
3.5.2 Performance for 3 and 4 queues in tandem . . . . . . . . . . . . 39
3.6 Extensive experimental results . . . . . . . . . . . . . . . . . . . . . 46
3.6.1 Validation for 2 queues in tandem . . . . . . . . . . . . . . . . 46
3.6.2 Validation of SDHI for 3 and 4 queues in tandem . . . . . . . . 62
3.7 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

**4 State-dependent heuristic for queues in parallel 71**
4.1 Model and notation . . . . . . . . . . . . . . . . . . . . . . . . . . 71
4.2 Preliminary discussion . . . . . . . . . . . . . . . . . . . . . . . . . 72
4.2.1 Motivation of the heuristic . . . . . . . . . . . . . . . . . . . . 72
4.2.2 Time reversal argument . . . . . . . . . . . . . . . . . . . . . . 72
4.3 State-dependent heuristic . . . . . . . . . . . . . . . . . . . . . . . 73
4.4 Experimental results . . . . . . . . . . . . . . . . . . . . . . . . . . 74
4.4.1 Performance . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
4.4.2 Sensitivity with respect to $b$ . . . . . . . . . . . . . . . . . . . . 79
4.4.3 Validation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
4.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88

**5 State-dependent heuristics for Jackson networks 89**
5.1 Model and notation . . . . . . . . . . . . . . . . . . . . . . . . . . 89
5.2 Buffer overflow at an arbitrary node . . . . . . . . . . . . . . . . . 90
5.3 State-dependent heuristics . . . . . . . . . . . . . . . . . . . . . . . 92
5.3.1 SDH for a feed-forward network . . . . . . . . . . . . . . . . . 92
5.3.2 SDH for a feedback network . . . . . . . . . . . . . . . . . . . 95
5.3.3 Possible generalization . . . . . . . . . . . . . . . . . . . . . . 97
5.4 Experimental results . . . . . . . . . . . . . . . . . . . . . . . . . . 98
5.4.1 Performance for a feed-forward network . . . . . . . . . . . . . 98
5.4.2 Performance for a feedback network . . . . . . . . . . . . . . . 101
5.5 Extensive experimental results . . . . . . . . . . . . . . . . . . . . . 104
5.5.1 Sensitivity with respect to $b_i$ . . . . . . . . . . . . . . . . . . . 104
5.5.2 Behavior of relative error . . . . . . . . . . . . . . . . . . . . . 108
5.5.3 Dependence of $b_{opt}$ on the overflow level . . . . . . . . . . . . 108
5.5.4 Guideline for finding $b_{opt}$ . . . . . . . . . . . . . . . . . . . . . 112
5.6 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112

**6 Exploring further 113**
6.1 A proof of asymptotic efficiency? . . . . . . . . . . . . . . . . . . . 113
6.1.1 Related work . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
6.1.2 First approach for a proof . . . . . . . . . . . . . . . . . . . . . 114
6.1.3 Second approach for a proof . . . . . . . . . . . . . . . . . . . . 115
6.1.4 Final remarks . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
6.2 Non-Markovian networks . . . . . . . . . . . . . . . . . . . . . . . . 122
6.2.1 Non-Markovian models . . . . . . . . . . . . . . . . . . . . . . 123
6.2.2 Optimal change of measure for $GI/GI/1$ queue . . . . . . . . . 124
6.2.3 Simulating non-Markovian networks . . . . . . . . . . . . . . . 125
6.2.4 Extension for tandem networks. General case . . . . . . . . . 126

6.2.5 Exact calculation of COM for Model 1: M/Bim/1 → ·/Bim/1 . . . . . 128
6.2.6 Exact calculation of COM for Model 2: $H_2$/Bim/1 → ·/Bim/1 . . . . . 130
6.2.7 Experimental results for Model 1 . . . . . 131
6.2.8 Experimental results for Model 2 . . . . . 137
6.2.9 Final remarks . . . . . 142

**7 Conclusions 143**
7.1 Contributions of the thesis . . . . . 143
7.2 Future work . . . . . 144

**A Fully state-dependent heuristic for tandem networks 145**

**B Proof of Observation 4 (Section 6.1.3) 151**

**Acknowledgments 159**

**Curriculum Vitae 161**

**Bibliography 163**

**Index 169**

# Chapter 1

# Introduction

In this thesis we study the problem of estimating the probability of population overflow in queuing networks. This introductory chapter provides the motivation of this work and an outline of the thesis.

## 1.1 Motivation of the work

Rare event probabilities have been an interesting topic of research for many years. Despite their rareness, these probabilities have a huge, and, sometimes, very crucial importance, like, for example, in predicting the crash of a space craft or an atomic factory explosion. They play a very important role in forecasting, e.g., in estimating the probabilities of hurricanes and earth quakes.

For telecommunication networks these probabilities are not life-critical, but still are much of the interest. Nowadays, with increased networks capabilities and huge progress in the network applications, a lot of data is transfered via the Internet: text files for e-mails, books, program files (eg., updates for web-browsers, e-mail programs), multi-media data, etc. It is very important that all the information reaches the addressee. Some high-level protocols (like TCP) take care, via retransmissions, that all the sent data is delivered. However, it is still important to prevent data loss as much as possible to decrease unnecessary retransmissions. For that, first, the network capacity needs to be large enough to support the traffic. Second, no, or a very small amount of packets may be lost during the transmission. With the recent advances in optical networking the first requirement is not a problem anymore. The second one is up to the designer to develop. While going from one end system to another, the traffic needs to pass several routers on its way. Since every link has a limited capacity, packets that find it busy, are stored in buffers of intermediate routers. When a newly arriving packet finds the buffer full, it has to be dropped. For a small buffer size and high network traffic this probability can become unacceptably large, thus, the system needs to be designed in such a way that the dropping, or, *overflow* is a rare event.

Two kinds of probabilities are usually studied, an overflow of an individual buffer and the total network population overflow. The exact calculation of these probabilities is only available for simple networks of small size. For more realistic and, thus, complicated systems, typically simulation is being used to obtain insight in these probabilities. Some special techniques are involved to speed up the simulation, since the events of interest are rare, and, thus, a long simulation run would be needed to estimate them with a high confidence. The most popular technique involved is *Importance Sampling* (IS) (e.g., [1], [2], [3], [4]). With IS, the simulation is done under a new distribution, called *a change of measure*, and the final estimator is weighted by the corresponding factors to compensate for this change. There are, however, no general guidelines on how to choose the new distribution functions under which the system is simulated. Theoretically, there exists the best IS distribution, which gives a zero-variance estimator. However, it is unpractical, since it depends on the probability to be estimated. Thus, other ways have to be employed to find IS distribution functions.

A change of measure to be used in IS can be either, *state-dependent*, i.e., different for each network state, or, *state-independent*, i.e., the same for each state. There are several approaches to find a change of measure to be used in IS. Some theoretical results exist but they are only available for small queuing networks of a specific type (e.g., [5], [6], [7]). For complicated systems either adaptive techniques (e.g., [8], [9], [10], [11], [12], [13], [14]), or, heuristic guesses (e.g., [15], [16], [17], etc.) are used. The advantage of adaptive algorithms is their general applicability. The disadvantage is their convergence, which can not be guaranteed. The problem with heuristic approaches is that there is no general rule which works for all types of networks and all possible network parameters.

Some asymptotically efficient results based on a heuristic approach have been obtained to estimate the overflow probability of individual buffers (e.g., [18], [19], [20], [21], [22]), but no such results are known to exist for the estimation of the total population overflow probability. The first work was done in [15] and was continued in [16], [17], [23], but proven later in [24] and [25] to be useful only for restricted network parameter settings. Thus, up to now, there have been no heuristics to estimate the total network overflow probability that can be applied to any type of network for all possible network parameters.

In this thesis, heuristics to estimate the total network overflow probability for different types of networks are developed. The networks that are considered include Jackson networks of nodes in tandem, in parallel, feed-forward networks and networks with feedback, as well as a 2-node non-Markovian tandem queuing network.

## 1.2 Outline of the thesis

**Chapter 2** provides necessary background information on existing techniques to simulate rare-event probabilities. The Importance Sampling technique, which is used in this thesis, is considered in more detail. We discuss a well-known heuristic to estimate total population overflow, as well as its applicability to different network parameters and topologies. We also talk about an adaptive method which we use later for comparison with the performance of the new heuristics developed in Chapters 3–5.

In **Chapter 3** we describe two heuristics for simulating rare events in tandem queuing networks. We give some motivation behind the approach based on a time-reversal argument. In the experimental section we compare the performance of our heuristics with the adaptive algorithm (described in Chapter 2) and show that they perform similar or better. We also validate the heuristics by extensive experimentation for networks of up to four nodes.

In **Chapter 4** we develop a heuristic for simulating rare events in queuing networks of nodes in parallel. The heuristic, theoretically, can be applied for any number of nodes. A large variety of experiments is done for networks of up to four nodes. The heuristic is validated and also shows good performance. The performance is, again, compared with the adaptive algorithm.

In **Chapter 5** the heuristics are developed for other Jackson network topologies, namely, feed-forward and feedback. Again, feed-forward network topologies of up to four nodes are considered and a heuristic for multiple-node network of a specific topology is developed. The comparison with the performance of the adaptive algorithm is presented and extensive experimental results are performed to demonstrate the validity of the heuristic. A heuristic for a small network with feedback is also developed. Several experimental results are given to compare the performance of the new heuristic with the adaptive algorithm.

**Chapter 6** describes further research ideas, in two parts. In the first part we discuss several approaches to analytically prove asymptotic efficiency of the heuristics developed in Chapter 3. In the second part we discuss how the heuristic methods developed in Chapter 3 can be generalized to non-Markovian networks. We give a couple of examples which show very good performance.

We conclude in **Chapter 7** and discuss some possibilities for future work.

# Chapter 2

# Importance Sampling

This chapter aims to provide background information on rare event simulation techniques and especially emphasizes the Importance Sampling (IS) method. It also discusses the applicability of some well known IS heuristics as well as adaptive algorithms for calculating rare event probabilities in queuing networks. The chapter is organized as follows: in Section 2.1 we discuss why ordinary Monte Carlo simulation is not applicable for estimating rare event probabilities; in Section 2.2 we present two techniques for rare event simulation; Section 2.3 describes the IS method in more detail; Section 2.4 discusses a well known IS heuristic which is applicable only for restricted types of networks and only for some network parameters, thus, showing a need for a new approach; Section 2.5 represents some adaptive algorithms and discusses their limitations.

## 2.1 Monte Carlo simulation

Consider an open queuing network of $d$ nodes. Let $X = (X_t, t \geq 0)$ be a stochastic process with the state space $S$, describing the network state at time $t$, i.e., $X_t = (x_{t,1}, ...x_{t,d})$, where $x_{t,i}$ is the number of customers at node $i$ at time $t$. We assume that $X_t$ is a Markov process. Let $A$ be a rare event set ($A \subset S$); suppose we are interested in estimating probability $\gamma = \Pr(A)$, i.e., the probability that the rare event $A$ occurs.

For example, if we want to estimate the probability that in a queuing system a queue reaches its maximum size before it gets empty (which is a rare event for a large buffer size) then $\gamma$ could be expressed as $\Pr\{T_A < T_0\}$, where $T_A$ is the first time the process enters the rare event set $A$ (the queue reaches its maximum size) and $T_0$ is the first time the queue gets empty.

The Monte Carlo (MC) simulation method for estimating $\gamma$ means collecting, say, $n$ samples $(\tilde{X}^{(1)}, ..., \tilde{X}^{(n)})$ of $X_t$, where $\tilde{X}^{(i)}$ represents the state of the system at the end of the $i$-th simulation run, i.e., $\tilde{X}^{(i)}$ is a state $(x^{(i)}_{T_i,1}, ...x^{(i)}_{T_i,d})$, where $T_i$ is the time of the $i$-th simulation run, and calculating the fraction of those samples that ended in $A$. Formally, let $I_{\{.\}}$ be an indicator function and $I_i = I_{\{\tilde{X}^{(i)} \in A\}}$, i.e., $I_i$ is equal to one if the rare event was observed during the simulation and is equal to zero, otherwise, then, the estimate $\tilde{\gamma}$ for $\gamma$ is equal to

$$\tilde{\gamma} \; = \; \frac{\sum_{i=1}^{n} I_i}{n}. \tag{2.1}$$

Note that $\tilde{\gamma}$ is an unbiased estimate of $\gamma$, i.e., $\mathbb{E}\tilde{\gamma} = \gamma$. The variance of the estimator $\gamma$ is given by

$$Var(\tilde{\gamma}) \; = \; \frac{\gamma(1-\gamma)}{n}, \tag{2.2}$$

and the relative error, defined as the ratio of the standard deviation of the estimator over its expectation, is equal to

$$RE(\tilde{\gamma}) \; = \; \frac{\sqrt{Var(\tilde{\gamma})}}{\gamma} \; = \; \sqrt{\frac{1-\gamma}{n\gamma}} \; \approx \; \frac{1}{\sqrt{n\gamma}}. \tag{2.3}$$

Thus, for a fixed number of samples $n$ the relative error $RE \rightarrow \infty$ as $\gamma \rightarrow 0$. In other words, for a fixed $RE = r$ we need at least $n = r^{-2}(1-\gamma)/\gamma \approx r^{-2}/\gamma$ samples, which means that $n \rightarrow \infty$ as $\gamma \rightarrow 0$. This fact makes the standard MC method inapplicable for estimating rare event probabilities.

## 2.2 Techniques for rare event simulation

To overcome the problems with estimating rare events, two main techniques have been developed: the splitting method and the Importance Sampling method. The main idea of both techniques is (in different ways) to make the rare event happen more often, and, hence, the MC method efficient again.

The **splitting method**, mostly known as **RESTART** (the REpetitive Simulation Trials After Reaching Thresholds) creates many hits of the rare event by repeating (splitting) the most promising paths, *a path* is a sequence of states visited during simulation) i.e., paths that have more chance to reach the rare event. The idea of splitting is based on the assumption that there exist some intermediate states that are visited much more often than the target (rare event) states. Those states behave as “gateways” to reach the target states. If, for example, the target states represent the full queue in a queuing system then states corresponding to the case when the queue is, say, half full can be regarded as intermediate states.

The splitting method starts by dividing the system state space into several intermediate subsets (or levels) called *restart levels.* Each time a path reaches the next restart level it is split into several trajectories. When one of the trajectories hits the next restart level the splitting repeats. The rare event probability is then calculated as a combination of many non-rare event probabilities that can be estimated by the MC method. Calculating the variance of the estimator is, however, not always straightforward and depends on the details of the splitting method. For example, when the path reaches the next restart level one could either split only this path and do not consider all other paths that were split on the previous level, or one could consider all paths that reached the next level from the previous one and split all of them; for more details see [27].

The efficiency of the splitting method depends on the proper choice of restart levels (how many of them to use and how to choose them) and the number of splits per level.

This choice is relatively simple for small models, but becomes more complicated and crucial for multi-node network models, where choosing the wrong parameters could lead to inefficient simulation.

In this thesis the splitting method will not be considered further; the interested reader can find more details in [27], [28], [29], [30], [31] and references therein.

Another way to increase the frequency of a rare event is to use the **Importance Sampling** (IS) method (e.g., [1], [2], [3], [4]). The idea of IS is to modify (bias) the underlying probability distribution such that the rare events occur much more frequently, i.e., important events are sampled more often, hence the name. To correct for this modification, the results are weighted in a way that yields a statistically unbiased estimator. The main problem of IS is to determine which parameter(s) of the system to bias (the technique), and how much to bias each of them. When the new parameters are defined correctly, the IS method allows to speed up the simulation considerably and to obtain a significant increase in estimator precision.

## 2.3 Basics of Importance Sampling

In this thesis we use the IS method, therefore below we discuss it in more detail. In Section 2.3.1 we introduce the notation, in Section 2.3.2 we describe the best change of measure that can be achieved, in Sections 2.3.3 and 2.3.4 we talk, respectively, about asymptotic efficiency and state-dependency properties.

### 2.3.1 Notation

IS involves simulating the model under a different underlying probability distribution so as to increase the frequency of typical sample paths leading to the rare event. Formally, let $X = (X_t, t \geq 0)$ be a stochastic process and $\gamma(\epsilon)$ be a sequence of rare event probabilities indexed by a *rarity parameter* $\epsilon$ ($\epsilon > 0$) so that $\gamma(\epsilon) \to 0$ as $\epsilon \to 0$. For example, in a buffer sizing problem, we could let $\epsilon = 1/b$ and $\gamma(\epsilon) = \Pr(q > b)$ where $b$ is a buffer size and $q$ is the random variable describing the steady-state queue length distribution.

Denote by $f$ and $\tilde{f}$ the original and new probability measures, respectively. Let $A$ be a rare event set, $I_{\{.\}}$ be an indicator function and $\omega$ be a sample path over the interval $[0, t]$. Then the rare event probability of interest can be expressed as follows

$$\gamma \,=\, \Pr(A) = \,\mathbb{E}\, I_{\{A\}} = \,\mathbb{E}\, I_{\{A\}} \frac{f(\omega)}{\tilde{f}(\omega)} \tilde{f}(\omega) \,=\, \tilde{\mathbb{E}}\, L_t(\omega)\, I_{\{A\}}\,, \tag{2.4}$$

where $\tilde{\mathbb{E}}$ is the expectation under the new measure $\tilde{f}$ and $L_t(\omega)$ is the *likelihood ratio* associated with path $\omega$, i.e.,

$$L_t(\omega) = \frac{f(\omega)}{\tilde{f}(\omega)}. \tag{2.5}$$

Thus, $\gamma$ can be estimated by simulating a random variable with a new probability density function $\tilde{f}$ and then unbiasing the output by multiplying it with the likelihood ratio. Sampling with a different density is called a *change of measure* and the density

$\tilde{f}$ is called the *Importance Sampling (IS) density.* The only condition on $\tilde{f}$ required to obtain an unbiased estimator is that

$$\tilde{f}(\omega) > 0, \forall \omega \in A \text{ such that } f(\omega) > 0. \tag{2.6}$$

Denote by $\tilde{\gamma}$ the estimator of $\gamma$ under the new measure $\tilde{f}$, i.e.,

$$\tilde{\gamma} \;=\; \tilde{\mathbb{E}}\, L_t(\omega)\, I_{\{A\}}. \tag{2.7}$$

Since $\mathbb{E}\tilde{\gamma} = \gamma$ (which also means that $\tilde{\gamma}$ is an unbiased estimator of $\gamma$) the variance of $\tilde{\gamma}$ is given by

$$Var(\tilde{\gamma}) \;=\; \tilde{\mathbb{E}}\, {L_t}^2(\omega)\, I_{\{A\}} \;-\; \gamma^2 \tag{2.8}$$

and a variance reduction is obtained if $\tilde{f}$ is chosen such that

$$\tilde{\mathbb{E}}\, {L_t}^2(\omega)\, I_{\{A\}} \;<\; \mathbb{E}\, I_{\{A\}}\,. \tag{2.9}$$

### 2.3.2 The optimal change of measure

Essentially any change of measure $\tilde{f}$ satisfying the condition (2.6) can be used. Then the natural question arises, what is the optimal change of measure, i.e., what is the density that minimizes the variance of $\tilde{\gamma}$?

Since $Var(\tilde{\gamma}) \geq 0$, the minimum is achieved when $\tilde{f}$ is chosen such that $Var(\tilde{\gamma}) = 0$. To show that this is possible, consider $\tilde{f} = f^*$ where

$$f^*(x) = \frac{f(x) I_{\{A\}}}{\gamma}. \tag{2.10}$$

Then $L_t(\omega) I_{\{A\}} \;=\; \gamma$ with probability one and $Var(\tilde{\gamma}) \;=\; 0$, since the variance of a constant is equal to zero. Thus, $f^*$ is the optimal change of measure, which is simply the original distribution *conditioned* on the occurrence of the rare event of interest. However, this knowledge can not be used in practice, since it requires a priori knowledge of $\gamma$, the probability we are trying to estimate.

How, then, can one find a good IS change of measure, i.e., a change of measure that reduces the variance of the estimator $\tilde{\gamma}$ and that satisfies Equation (2.9)? Since $\mathbb{E}\tilde{\gamma} = \gamma$ for any density $\tilde{f}$ that satisfies (2.6), reducing the variance is equivalent to reducing the second moment:

$$\tilde{\mathbb{E}}\, {L_t}^2(\omega)\, I_{\{A\}} \;=\; \tilde{\mathbb{E}}\, \frac{f(\omega)}{\tilde{f}(\omega)}\, I_{\{A\}} \frac{f(\omega)}{\tilde{f}(\omega)} \;=\; \mathbb{E}\, \frac{f(\omega)}{\tilde{f}(\omega)}\, I_{\{A\}}) \;=\; \mathbb{E}\, L_t(\omega) I_{\{A\}}\,, \tag{2.11}$$

i.e., making the likelihood ratio $L_t(\omega) \;=\; f(\omega)/\tilde{f}(\omega)$ small on the set $A$. Note that, outside $A$, Equation (2.11) is equal to zero due to the $I_{\{A\}}$ multiplicand. Since $f$ is already small on A (a rare event), the problem is to find a new measure $\tilde{f}$ that is large on $A$, i.e., the event of interest is likely to occur more often under density $\tilde{f}$. Under the (unknown) density $f^*$ the event of interest occurs with probability one.

### 2.3.3 Asymptotic efficiency

To measure the effectiveness of the IS density $\tilde{f}$, the asymptotic behavior of the estimator is studied, i.e., how the relative error of the estimator $\gamma(\epsilon)$, defined as

$$RE(\tilde{\gamma}(\epsilon)) \;=\; \frac{\sqrt{Var(\tilde{\gamma}(\epsilon))}}{\gamma(\epsilon)}, \tag{2.12}$$

changes when the rarity parameter $\epsilon \to 0$. The estimator $\tilde{\gamma}(\epsilon)$ is asymptotically efficient if its relative error grows at sub-exponential (e.g., polynomial) rate as $\epsilon \to 0$. Formally, let $\lim_{\epsilon \to 0} \epsilon \log \gamma(\epsilon) = -\theta$ with $\theta > 0$; that is, $\theta$ is the asymptotic decay rate of $\gamma(\epsilon)$ as $\epsilon \to 0$. Then, an estimator is said to be *asymptotically efficient* if

$$\lim_{\epsilon \longrightarrow 0} \epsilon \log \tilde{\mathbb{E}}\, {L_t}^2(\omega)\, I_{\{A\}} = -2\theta. \tag{2.13}$$

The property of asymptotic efficiency is very beneficial since it guarantees that the number of simulation samples needed to achieve a given relative error grows less than exponentially fast when the rare event probability is (exponentially) decaying. One can easily see this by rewriting Equation (2.12) (first, taking the logarithm, then multiplying by $\epsilon$ and, then, taking the $\lim_{\epsilon \to 0}$) in the equivalent form:

$$\lim_{\epsilon \longrightarrow 0} \epsilon \log RE(\gamma(\epsilon)) = \lim_{\epsilon \longrightarrow 0} \epsilon \frac{1}{2} \log Var(\tilde{\gamma}(\epsilon)) - \lim_{\epsilon \longrightarrow 0} \epsilon \log \gamma(\epsilon). \tag{2.14}$$

Taking into account that

$$Var(\tilde{\gamma}(\epsilon)) \;\leq\; \tilde{\mathbb{E}}\, {L_t}^2(\omega)\, I_{\{A\}} \tag{2.15}$$

we obtain

$$\lim_{\epsilon \longrightarrow 0} \epsilon \log RE(\gamma(\epsilon)) \leq -\frac{1}{2}(-2\theta) + \theta = 0, \tag{2.16}$$

which means that $RE$ grows less than exponentially fast with $\gamma(\epsilon)$ decaying exponentially.

Even better than asymptotic efficiency is the *bounded relative error* property. It means that the relative error remains bounded as the estimator goes to zero, i.e., $RE(\tilde{\gamma}(\epsilon)) \leq C$ for all $\gamma(\epsilon)$ as $\epsilon \to 0$. This is the most desirable characteristic that can be achieved in practice; it implies that one needs only a fixed (bounded) number of samples $n$ to estimate $\gamma(\epsilon)$ within a certain relative precision, *independent of how small the probability of interest is.* Note that bounded relative error implies asymptotic efficiency.

### 2.3.4 State dependency

In general, a change of measure may depend on the system state, even if the original underlying distributions are state-independent. For example, in a Markovian queuing network, the new arrival and service rates to be used in importance sampling may depend on the state of the network, i.e., the buffer content at each node. Recent theoretical and empirical studies (e.g., [6], [32], [13]) reveal that state-dependent changes of measure are generally more effective and can be applied when no effective state-independent change of measure exists.

## 2.4 Analytical and heuristic IS techniques

There exists no general rule for choosing a change of measure $\tilde{f}$. The main techniques used in the literature are those based on large deviation theory (discussed in Section 2.4.1), or heuristic approaches (Sections 2.4.2–2.4.3), or iterative methods, like the cross-entropy method (discussed in Section 2.5). They will be considered in more detail below together with their advantages and disadvantages.

### 2.4.1 Large deviation theory

As mentioned before, the problem of IS is to find the optimal change of measure. An analytical way of doing this is based on Large Deviation Theory (LDT) (see, for example, [33], [34], [2], [35], [36]). Loosely speaking, LDT can be viewed as an extension of the traditional limit theorems of probability theory. The (Weak) Law of Large Numbers basically states that certain probabilities converge to zero, while LDT is concerned with the rate of convergence, as explained below.

Formally, let $X_1$, $X_2$,... be a sequence of i.i.d. random variables with mean $\mu$ and variance $\sigma^2$ taking values in $\mathbb{R}^d$ and $S_n = X_1 + ... + X_n$. Let

$$M(\theta) = \mathbb{E}e^{\langle\theta, X_1\rangle},\ \theta \in \mathbb{R}^d, \tag{2.17}$$

be the moment generating function associated with $X_i$. The sequence $S_n/n$ is said to *satisfy a large deviation principle* ([36]) if for all closed subsets $C \subset \mathbb{R}^d$

$$\limsup_{n\to\infty} \frac{1}{n} \log \Pr(S_n \in C) \leq -\inf_{x\in C} h(x), \tag{2.18}$$

and for all open subsets $F \subset \mathbb{R}^d$

$$\liminf_{n\to\infty} \frac{1}{n} \log \Pr(S_n \in F) \geq -\inf_{x\in F} h(x), \tag{2.19}$$

where the function $h(x)$, called *Cramér* or *Legendre transform*[1] (or, *the large deviation rate function*), is defined as

$$h(x) = \sup_{\theta\in\mathbb{R}^d} [\langle\theta, x\rangle - \log M(\theta)]. \tag{2.20}$$

The inequality (2.18) is usually referred to as the large deviation upper bound and (2.19) as the large deviation lower bound, and both of them are called as *Cramér's theorem.* This theorem gives the rate of convergence for the Weak Law of Large Numbers (WLLN). This can be seen from an equivalent statement of (2.18) and (2.19) for $\mathbb{R}^1$. The WLLN states that

$$\lim_{n\to\infty} \Pr(S_n/n > y) = 0,\ \text{for } y > \mu. \tag{2.21}$$

For $y > \mu$ Cramér's theorem gives

$$\lim_{n\to\infty} \frac{1}{n} \log \Pr(S_n/n > y) = -h(y), \tag{2.22}$$

---

[1] In other literature it is, sometimes, called *convex dual* or *Fenchel-Legendre transform.*

i.e., roughly speaking, it means that $\Pr(S_n/n \approx y) \approx e^{-nh(y)}$.

Now, how can LDT be applied to speed up simulation? The answer and the theory behind it can be found in [2]. Since the correct mathematical description of the theoretical results include a lot of new notation that will not be used later in this thesis, we present only the general idea. In [2] Markov chains with discrete time, making "small" jumps over $\mathbb{R}^d$ are considered. The set of continuously piecewise differentiable functions $\phi$: $[0,T] \rightarrow \mathbb{R}^d$ called *paths* is introduced. An *action integral* along path $\phi$ is defined as

$$I(\phi) = \int_0^T h_{\phi(t)}(\phi'(t))dt. \tag{2.23}$$

It is shown that for a given set $A$ the probability of interest can be approximated by

$$\Pr(S_n \in A) \approx e^{-n \inf_{\phi \in A} I(\phi)}. \tag{2.24}$$

The path $\phi^*$ for which $\inf_{\phi \in A} I(\phi)$ is achieved is the most likely path to reach the set $A$, i.e., *finding the optimal path means minimizing the action integral.*

Practically, this means that if $A$ is the target rare set than it is most likely to be reached by following the path $\phi^*$. Thus, simulating the system under the change of measure that favors path $\phi^*$ is the quickest way to reach the set $A$. In [2] it is shown that the change of measure (to be used in IS) that satisfies this property is *the exponential change of measure* (also called exponential twist) defined as

$$dF^*(x) = \frac{e^{\langle \theta, x \rangle} dF(x)}{M(\theta)}, \text{ with } \theta \in \mathbb{R}^d, \tag{2.25}$$

where $F(x)$ is the original distribution. It was proven in [2] that among all exponential changes of measure the distribution with $\theta = \theta^*$, with $\theta^*$ being the one on which the supremum (2.20) is achieved, is the optimal change of measure in the sense that it minimizes Equation (2.11). In other words, the exponential twisting with parameter $\theta^*$ is the best exponential change of measure to be used in IS.

Below we will present the LDT results for the simple case of the M/M/1 queue.

### M/M/1 queue

Consider the example (as in [15]) of a one server queue with exponentially distributed inter arrival (with mean $1/\lambda$) and service (with mean $1/\mu$) times. Let $f_B(x)$ and $f_D(x)$ be the corresponding probability density functions and $M_B$ and $M_D$ be the corresponding moment generating functions.

Suppose we are interested in the probability that the queue exceeds its capacity $N$ (*overflow level*) before becoming empty. For a high overflow level this probability is rare, thus, IS needs to be employed. The change of measure to be used in IS is the exponential twisting with parameter $\theta^*$ (see discussion above) which follows the most likely path to reach level $N$.

This path was found in [15] and a system of equations was derived to find the parameter $\theta^*$ on which the supremum (2.20) is achieved. As discussed above the change of measure to be used in IS is the exponential twist (cf. (2.25)) with the parameter $\theta^*$.

It was proven in [15] that $\theta^*$ is a solution of equation $M_B(-\theta) \cdot M_D(\theta) = 1$, i.e.,

$$\left(\frac{\lambda}{\lambda+\theta}\right)\left(\frac{\mu}{\mu-\theta}\right) = 1 \tag{2.26}$$

with $\theta^* = \mu - \lambda$. Substituting $\theta^*$ in Equation (2.25), taking into account that $M_B(-\theta^*) = \lambda/(\lambda+\theta^*) = \lambda/\mu$ and $M_D(\theta^*) = \mu/(\mu-\theta^*) = \mu/\lambda$, gives the new density functions defined as

$$f_B^*(x) = \mu e^{-\mu x}, \tag{2.27}$$

$$f_D^*(x) = \lambda e^{\lambda x}. \tag{2.28}$$

Thus, the exponential change of measure with parameter $\theta^*$ for an M/M/1 queue means that the new arrival and service rates are twisted, i.e., the new arrival rate is equal to the original service rate and the new service rate is equal to the original arrival rate.

#### Difficulties with applying LDT to other (networks of) queues

As we have just seen, LDT can be successfully applied to find the optimal change of measure in case of an M/M/1 queue. The bad news, however, is that the extension of this approach to general Jackson queuing networks is not possible (unless some new results of large deviation theory for Markov processes with discontinuous kernels will be available [15]). The main problem lies in the fact that for finding the solution of Equation (2.20) one needs to solve the variational problem with some restrictions on the transition rate functions (see [15]). Those restrictions are violated for more general queuing networks. In particular, in queuing network models, transition rates are not smooth functions of the state space; there is a discontinuity on the boundary when a server changes from busy to idle. For a single queue there is only one boundary at 0, but since the overflow probability can be estimated considering the behavior of the queue during a busy period, this boundary plays no essential role. In contrast, the boundaries in queuing networks significantly affect the form of the likelihood ratio associated with a change of measure, and make it much more difficult to identify effective IS distributions.

### 2.4.2 State-independent heuristics

To overcome the difficulties with applying LDT to general queuing networks, researchers started to look for heuristic approaches. The breakthrough in this direction was made in 1989 by Parekh and Walrand ([15]). Based on a heuristic application of LDT techniques, they proposed state-independent importance sampling estimators (referred in the sequel as *PW heuristic*) for overflow probabilities in various Jackson networks. In particular, they were interested in a probability of total network population overflow; namely, the probability that the total number of customers in the network reaches some high level $N$ before becoming empty. For queues in parallel their estimator interchanges the arrival and the service rates of the queue with the largest traffic intensity. For tandem networks, the PW heuristic interchanges the arrival rate and the slowest service rate, thus generalizing the M/M/1 estimator described in the previous section.

For other types of Jackson networks and networks of GI/GI/1 queues the PW heuristic was described as a solution of a variational problem, which was solved later in [17] for Jackson networks and in [37] for tandem networks of GI/GI/1 queues.

However, it was shown in [24] and later investigated in [25] that for two or more queues in tandem the PW heuristic does not always give an asymptotically efficient simulation, depending on the values of arrival and service rates. In particular, asymptotic efficiency fails when the two service rates are nearly equal. This can be intuitively explained. In general, a good change of measure assigns large probabilities to the most likely paths to the rare event ([4]). When service rates are significantly different, the overflow in the network is most likely to occur because of a buildup in the bottleneck queue (see [38] where it was proven asymptotically, i.e., when the population level $N \to \infty$). IS based on interchanging the arrival rate with the bottleneck service rate mimics this behavior. On the other hand, if the service rates are close, there are many ways for a large network population to accumulate. In [38] it was proven that for two node tandem networks with equal service rates asymptotically the hitting probability is uniformly distributed over the hitting line (the line $n_1+n_2 = N$). Thus, IS based on the interchanging rule is not effective for estimating the probability of a total network population, though, it is still effective for estimating the single buffer overflow probability (see [22]). Those two probabilities (total network population overflow and a single node overflow) are mostly the ones of the researchers' interest.

In this thesis we are particularly interested in the probability of the total network population overflow starting from an empty network.

The computation of overflow probabilities of a single node was studied in many other papers. For example, in [18] the asymptotically optimal state-independent heuristic based on the theory of effective bandwidths and Markov additive processes was developed for a single queue and for in-tree networks. The exponential change of measure was studied in detail in [1].

### 2.4.3 State-dependent heuristics

The above mentioned heuristics are state-independent, i.e., the change of measure does not depend on the state of the network; this, of course, keeps the heuristics simple. On the other hand, by allowing dependence on the system state (typically, the content of each of the queues), more efficient IS schemes may be obtained. In [6] the overflow probability of the second queue in a two-node tandem network is estimated using an exponential change of measure that depends on the content of the first buffer. The approach is based on a Markov additive process representation of the system and yields asymptotically efficient simulation when the first buffer is finite; otherwise, the relative error is bounded only if the second server is the bottleneck. A state-dependent change of measure is also used in [39] for simulating link overloads in telecommunication networks; again the functional dependence of the IS rates on the system state is derived using a heuristic and rather specific mathematical models.

In this thesis we propose very effective state-dependent heuristics for various types of Jackson networks (Chapters 3–5).

## 2.5 Adaptive IS techniques

As an alternative to analytical or heuristic approaches to find a good change of measure several adaptive techniques have been proposed. The key idea of an adaptive method is an iterative (adaptive) procedure, which at every step recalculates (adapts) a change of measure using the results from the previous step, until it converges to the "optimal" change of measure. One class of adaptive techniques does that by iteratively minimizing the variance of the estimator (i.e., doing IS directly, see, e.g., [40], [41], [42]); another class does that indirectly by minimizing some distance (namely, the cross-entropy) to the (generally unachievable) zero-variance change of measure (e.g., [43],[9], [32], [10], [11]).

Since the cross-entropy method will be later used for comparison with the heuristics developed in this thesis we present it in more detail here.

### 2.5.1 Basics of cross-entropy method

Assume that the change of measure is parametrized by some vector $u$. Define $\omega$ as the sample path of one replication of the simulation and by $I(\omega)$ the indicator function of the occurrence of the rare event in $\omega$. Denote by $f(\omega, u)$ the probability (or, for continuous systems, probability density) of the sample path $\omega$ under $u$, with $u = 0$ corresponding to the original system. The likelihood ratio, associated with the sample path $\omega$ and a parameter $u$ is given by

$$L(\omega, u) = \frac{f(\omega, 0)}{f(\omega, u)}, \tag{2.29}$$

and the expectation under $u$ is denoted by $\tilde{\mathbb{E}}_u$.

*The Kullback-Leibler cross-entropy* between two probability distributions $f(x)$ and $g(x)$ is defined as follows:

$$CE = \int f(x) \ln \frac{f(x)}{g(x)} dx \tag{2.30}$$

Note, that this distance measure is not symmetric; also, if $f(x)$ and $g(x)$ are identical, $CE = 0$.

The Kullback-Leibler cross-entropy is applied to measure the distance between the distribution to be used for simulation and the optimal (ideal) distribution. If we substitute $g(x) = f(x, u)$ (the distribution to be optimized by changing $u$) and $f(x) = \rho_0 h(x) f(x, 0)$ with $\rho_0 = (\int h(x) f(x, 0) dx)^{-1}$ (the "ideal" distribution, i.e., the original distribution ($= f(x, 0)$) conditioned on the occurrence of the rare event (see Section 2.3.2)), then, minimizing $CE$ means finding a vector $u^*$ such that

$$\begin{aligned} u^* &= \arg\min_u \int \rho_0 h(x) f(x, 0) \ln \frac{\rho_0 h(x) f(x, 0)}{f(x, u)} dx \\ &= \arg\max_u \int h(x) f(x, 0) \ln f(x, u) dx \\ &= \arg\max_u \mathbb{E}_0 h(x) \ln f(x, u), \end{aligned} \tag{2.31}$$

which is equivalent to (see Equation (2.4)):

$$u^* = \arg\max_u \mathbb{E}_{u_j} h(x) L(x, u_j) \ln f(x, u), \tag{2.32}$$

where $u_j$ is a parameter vector yet to be found. Since the expectation can be approximated by the sum over $n$ samples of simulation performed with parameter $u_j$, the above equation can be rewritten as follows:

$$u_{j+1} = \arg\max_u \sum_{i=1}^{n} I(\omega_i) L(\omega_i, u_j) \ln f(\omega_i, u), \tag{2.33}$$

where $u_{j+1}$ is an approximation of $u^*$. This equation forms the base of the iterative procedure described below.

**Algorithm 1** Adaptive algorithm for finding a change of measure $u^*$

```
1: choose the initial parameter $u_0$ (see discussion below)
2: j := 1
3: repeat
4:     $u_j := u_{j-1}$
5:     simulate $n$ sample paths with parameter $u_j$, yielding $\omega_1, ..., \omega_n$
6:     use Equation (2.33) to find the new parameter $u_{j+1}$
7:     j := j+1;
8: until $u_j$ has converged, i.e., $u_j \approx u_{j-1}$
```

### 2.5.2 Algorithmic description

To start applying the algorithm, first, some initial parameter $u_0$ at step 1 needs to be chosen. Theoretically, it could be any value; for example $u_0 = 0$, which corresponds to the original distribution. However, it is not practical since under the original distribution the event of interest is rare and, hence, will typically not be observed, which makes (2.33) unusable. To overcome this, the parameter $u_0$ should be chosen such that the rare event is, somehow, made less rare. In [43] this is done by introducing an additional step in the algorithm in which the rare event is temporarily modified (under the same probability distribution) into a less rare event (for example, the size of a buffer in a buffer sizing problem is made smaller). Another approach is to use as the initial vector $u_0$ a heuristic change of measure (like, for example, the PW heuristic of interchanging the arrival rate with the service rate of the most loaded queue, [10]), or, as another version of this, use as $u_0$ the results of a (state-independent) adaptive procedure (see [32]).

**Remark 2.5.1.** Note that the convergence of the above algorithm has not been theoretically proven. As a matter of fact, it can happen that it does not converge at all if the number of replications for simulation is not large enough or the initial parameter is not good.

The state-dependence property of a change of measure, obtained by the above algorithm, is hidden in parameter $u_j$. When $u_j$ is chosen to be the same for each system state, we obtain the state-independent version of the algorithm. As discussed in Section 2.4.2, state-independent changes of measure are not always efficient, thus, allowing dependence on the state makes the algorithm less restrictive and, hence, broader applicable.

### 2.5.3 State-dependency

Although allowing state-dependency does seem to be a good idea, it has its own pitfalls when applying the adaptive procedure to networks of queues. The main problem lies in the fact that the number of states grows very quickly with the number of nodes in the network, or, for some problems, can be even infinite. For example, if we are interested in the probability of an individual buffer overflow in a network where all other queues have infinite buffers, we obtain an infinite state space (hence, can not store the information for all states in a computer memory), which makes the above described algorithm in its present form completely inapplicable.

To overcome the problem of large state spaces several techniques can be used, like, for example, local average, boundary layers and splines (e.g., [32], [10], [13]). Each of these deals with a specific part of a large state space problem.

*Local averaging* helps to overcome the problem with rarely visited states. When the state space is large some states during the simulation may not be visited, or are not visited often enough, which leads to very high relative errors for this kind of states. By averaging the statistic over neighboring states this problem can be overcome; it also helps to reduce relative errors of the estimated quantities (in case of Markovian models those quantities are the new rates or transition probabilities).

*Boundary layers* are used to reduce the state space directly by combining the states with a large number of customers at some queue in one state (i.e., in some sense, truncating the state space). It is based on the assumption that dependence on the content of the buffer diminishes when the number of customers increases. For example, in a two node network with finite buffers of size, say, $N = 100$ and the probability of interest being the first time one of the buffers gets full, i.e., reaches level $N$, the size of the state space is equal to $(N + 1) \cdot (N + 1) = 10201$. If we choose, say, 4 boundary layers, the state space can be reduced to 25 states, which is more than 400 times smaller! (In that case the state space consists of the states (0,0),...,(0,4),(1,0),..,(1,4),...,(4,0),...,(4,4) where states $(4,j)$ and $(i,4)$ represent all states with at least 4 customers at the first or the second queue, respectively.) The number of boundary layers $b$ is usually chosen by trial and error. If $b$ is too small, the resulting estimate may not converge since the change of measure is close to a state-independent, which does not always work, or will have high relative error. For too large $b$ the convergence can be slow since there are too many states to be processed.

*Smoothing using spline fitting* is another technique. In general the idea of smoothing is to reduce the “noise” in the resulting rate functions by approximating them with smooth functions. It can be done, for example, by dividing the state space domain into segments and, then, choosing some approximating (smooth) function on each segment. When these functions are chosen to be polynomials (splines), the smoothing technique is called smoothing using spline fitting. The smoothing technique has several advantages. First, it allows to keep the state-dependence property. Second, it gives a significant convergence speed-up in cases of very “noisy” approximations of rate functions. On the other hand, if the transition rates are already smooth enough, doing spline fitting can lead to a higher variance of the resulting estimates, i.e., worsen the accuracy.

## 2.6 Summary

In this chapter we have given an overview of rare event simulation techniques. We have seen that Monte Carlo simulation can not be used when one aims to find the probability of an event which occurs only rarely, so techniques like splitting method or Importance Sampling (IS) method have to be applied. The IS method aims to find a new probability distribution (a change of measure) that makes the event occur more frequently. Several ways of doing this were discussed: an analytical approach, which is limited to only simple and small networks; an adaptive method, which aims to find a change of measure by an iterative algorithm either minimizing the variance of the estimator, or some distance (like cross entropy) to the "optimal" (zero-variance) distribution (the original distribution conditioned on the occurrence of the rare event). Though adaptive method can be applied for different types of Jackson networks, it is more applicable for small networks of two or three nodes. Starting with four node networks it might experience slow convergence, or, sometimes, might not converge at all. Finally, heuristic methods have also been discussed and it was shown that the existing heuristics are applicable only for restricted network parameters. Thus, the necessity for a new approach is clear.

In this thesis we will propose new heuristics applicable for various types of Jackson networks (tandem, parallel, feed-forward and feed-back) and all network parameters.

# Chapter 3

# State-dependent heuristics for tandem networks

In this chapter we discuss state-dependent heuristics for simulating population overflow in tandem networks. Section 3.1 provides the formal model and notation. Sections 3.2 and 3.3 discuss the heuristics for two and $d$-node tandem networks, respectively. Section 3.4 describes how we compare the performance of different methods. Sections 3.5–3.6 include simulation results and discuss performance gain in comparison with other methods (Section 3.5) and validation of the proposed heuristics (Section 3.6).

## 3.1 Model and notation

Consider a Jackson network consisting of $d$ nodes (queues) in tandem. Customers arrive at the first node according to a Poisson process with rate $\lambda$. The service time of a customer at node $i$ is exponentially distributed with rate $\mu_i$. Customers that leave node $i$ join node $i+1$ (if $i < d$) or leave the network (if $i = d$). Each node has its own buffer, which is assumed to be infinite. We also assume that the queuing network is stable, i.e.,

$$\lambda < \min_i \{\mu_i\}, \tag{3.1}$$

and the rates are normalized, i.e.,

$$\lambda + \sum_{i=1}^{d} \mu_i = 1. \tag{3.2}$$

Let $X_{i,t}$ $(1 \leq i \leq d)$ denote the number of customers at node $i$ at time $t \geq 0$ (including those in service). Then the vector $\mathbf{X}_t = (x_{1,t}, x_{2,t}, ..., x_{d,t})$ is a Markov process representing the state of the network at time $t$. Denote by $S_t$ the total number of customers in the network (network population) at time $t$, i.e., $S_t = \sum_{i=1}^{d} x_{i,t}$.

Assuming that the initial network state is $\mathbf{X}_0 = (1, 0, ..., 0)$, i.e., upon an arrival of the first customer to an empty network (the probability of arrival to an empty network is equal to one), we are interested in the probability that the network population reaches some high level $N \in \mathbb{N}$ before returning to 0. We denote this probability by $\gamma(N)$ and refer to it as *the population overflow probability*, starting from an empty network.

## 3.2 Two-node tandem networks

As discussed in Section 2.4.2 even for the simplest (2-node) tandem network, there is no state-independent change of measure which is asymptotically efficient over the entire range of feasible network parameters (arrival and service rates) (e.g., [24], [25]). Only state-dependent change of measures, carefully developed through analysis (e.g., [6]) or determined using adaptive optimization methods (e.g., [12], [13]), have shown to be efficient in cases where no state-independent change of measure is known to work. Unfortunately, recently proposed methods (e.g., [6], [12], [13]) to determine state-dependent change of measures have some drawbacks. It is not clear whether the analysis in [6] can be easily extended to larger and more general networks. Similarly, computational demands and large state-space limit the effectiveness of adaptive methods (e.g., [12], [13]).

In this section we propose a new approach and use it to determine a state-dependent change of measure to estimate the probability of population overflow in 2-node tandem networks. Although no proofs of asymptotic efficiency are provided, the heuristics are motivated by arguments based on “time-reversal” of large deviation paths [38] and are empirically shown to yield estimates with bounded (or linear in $N$) relative error.

### 3.2.1 Motivation of the heuristic

The change of measure proposed in this section is inspired by theoretical and empirical results in [6] and [13]. These results indicate that the “optimal” change of measure depends on the state of the network, i.e., the number of customers at the network nodes. Furthermore, this dependence is strong along the boundaries of the state-space (i.e., when one or more buffers are empty) and eventually (often quickly) disappears in the interior of the state-space (i.e., when the contents of all nodes in the network are sufficiently large).

The above observation suggests that if we know the “optimal” change of measure along the boundaries and in the interior of the state-space, then we might be able to “construct” a change of measure that approximates the “optimal” one over the entire state-space. If the approximation is sufficiently good, then the change of measure may yield asymptotically efficient estimators. Empirical results and comparisons in Section 3.6 indeed confirm that changes of measure constructed in that way produce asymptotically efficient estimators with a bounded relative error for all feasible parameters of the 2-node tandem network.

To realize the above idea we need to determine the “optimal” change of measure in the interior and along the boundaries of the state-space. To do that, we use heuristics based on combining known large deviations results and “time-reversal” arguments, as explained in the following section.

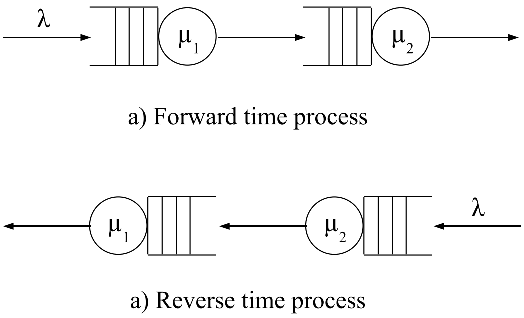


Figure 3.1: Time reversal of the 2-node tandem network

### 3.2.2 Time reversal argument

In this section we apply time reversal arguments [44] to give an insight to the change of measure we are going to introduce in Section 3.2.3; this, by no means, is a formal proof of asymptotic efficiency.

The reverse time process is also a 2-node tandem network (see Figure 3.1); however, arrivals (rate $\lambda$) enter the network at node 2 (service rate $\mu_2$) and exit from node 1 (service rate $\mu_1$). Roughly speaking, according to [38], in the limit as $N \to \infty$, the most likely path to the rare set (i.e., population overflow) in the forward time process is the same path by which the reverse time process evolves, given that the latter starts from the rare set. Since both node 1 and node 2 may be non-empty upon entry into the rare set, the hitting state $(x_1, x_2)$, is somewhere along the line $x_1 + x_2 = N$.

Let $\mu_2 \leq \mu_1$, and the reverse time process starts at $(n_1, n_2)$ such that $n_1 + n_2 = N$. Node 2 has arrival rate $\lambda$ and initially (if $n_2 > 0$) its departure rate is $\mu_2$, thus it empties at rate $(\mu_2 - \lambda)$. In the meantime, node 1 has arrival rate $\mu_2$ and (if $n_1 > 0$) departure rate $\mu_1$, thus it empties at rate $(\mu_1 - \mu_2)$. If $\mu_1 = \mu_2$, then node 1 is "critical" and does not empty; this corresponds to Path III in Figure 3.2. If and when node 2 empties first, its arrival and departure rates are equal to $\lambda$. At that time, node 1 has arrival rate $\lambda$ and departure rate $\mu_1$, thus it empties at rate $(\mu_1 - \lambda)$. This corresponds to Path III and Path II (to the right) in Figure 3.2. If and when node 1 empties first, its arrival and departure rates are equal to $\mu_2$. At that time, node 2 has arrival rate $\lambda$ and departure rate $\mu_2$, thus it empties at rate $(\mu_2 - \lambda)$. This corresponds to Path I and Path II (to the left) in Figure 3.2.

Note that departures (resp. arrivals) in reverse time correspond to arrivals (resp. departures) in forward time. It follows that along the most likely path from an empty network to population overflow (in forward-time), there are two possible scenarios depending on the entry state $(n_1, n_2)$ into the rare set, which in turn depends on the arrival and service rates: One scenario corresponds to Path II (to the right), which is more likely when $\mu_2$ is less than, but sufficiently close to, $\mu_1$. In this scenario, node 1 builds up first while node 2 is stable (i.e., $\tilde{\lambda} = \mu_1, \tilde{\mu}_1 = \lambda, \tilde{\mu}_2 = \mu_2$). At some point, also node 2 starts to build up until the rare set is hit (i.e., $\tilde{\lambda} = \mu_1, \tilde{\mu}_1 = \mu_2, \tilde{\mu}_2 = \lambda$). Path III is simply the limit of Path II when $\mu_1 = \mu_2$. A second scenario corresponds to Path II (to the left), which is more likely when $\mu_2$ is less than, but not sufficiently

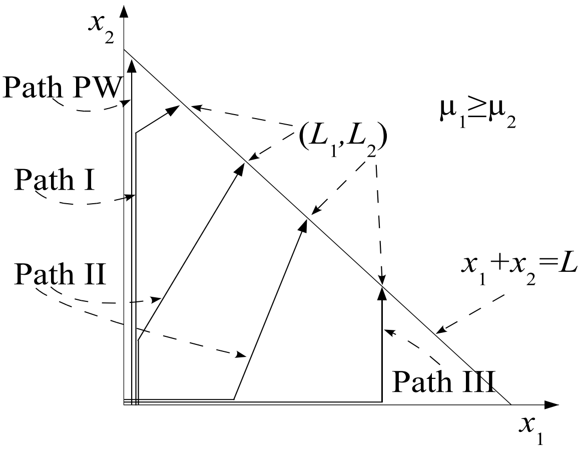


Figure 3.2: Most likely path to population overflow in a 2-node tandem network ($\mu_1 \geq \mu_2$)

close to, $\mu_1$. In this scenario, node 2 builds up first while node 1 is stable (i.e., $\tilde{\lambda} = \mu_2, \tilde{\mu}_1 = \mu_1, \tilde{\mu}_2 = \lambda$). At some point, also node 1 starts to build up until the rare set is hit (i.e., $\tilde{\lambda} = \mu_1, \tilde{\mu}_1 = \mu_2, \tilde{\mu}_2 = \lambda$). Path II tends to Path I when $\mu_2 \ll \mu_1$.

Now, if $\mu_2 \leq \mu_1$, then the heuristic in [15] exchanges $\lambda$ and $\mu_2$ leaving $\mu_1$ unchanged; i.e., node 1 is stable, and node 2 builds up all the way until the rare set is hit. This corresponds to the Path PW in Figure 3.2. It is interesting to note that for $\mu_2 \ll \mu_1$ Path I is the most likely and it gets closer to Path PW, which explains the effectiveness of the heuristic in [15] for sufficiently small $\mu_2$. For larger $\mu_2$ (closer to $\mu_1$) the most likely path gets closer to Path II and deviates further from Path PW, which explains why the heuristic in [15] becomes ineffective in this case.

Similar discussion can be applied for the case $\mu_1 < \mu_2$ with the only difference that in case when both nodes are non-empty node 1 builds up with rate $\mu_2 - \mu_1$ (instead of emptying). Another difference is that the case when node 2 is non-empty and node 1 is empty quickly goes to the case when both nodes are non-empty since output rate from node 2 is larger than the output rate from node 1. Thus, in case $\mu_1 < \mu_2$ Path III and Path II (to the right) are more probable, i.e., node 1 builds up first, and then, node 2.

Thus, in both cases ($\mu_1 \geq \mu_2$ and $\mu_1 < \mu_2$) there are three possibilities: either node 1 or node 2 builds up (i.e., $\tilde{\lambda} = \mu_1$, $\tilde{\mu}_1 = \lambda$, $\tilde{\mu}_2 = \mu_2$, or $\tilde{\lambda} = \mu_2$, $\tilde{\mu}_1 = \mu_1$, $\tilde{\mu}_2 = \lambda$), or both of them simultaneously (i.e., $\tilde{\lambda} = \mu_1$, $\tilde{\mu}_1 = \mu_2$, $\tilde{\mu}_2 = \lambda$).

Below we propose the heuristic (and its improved version), which is a combination of two over three possibilities and which can (roughly) capture the most likely path to overflow (i.e., Path I, Path II or Path III, depending on the network parameters). This clarifies the apparent robustness and effectiveness of this heuristic over the entire feasible parameter range (as evidenced from experimental results in Sections 3.5–3.6).

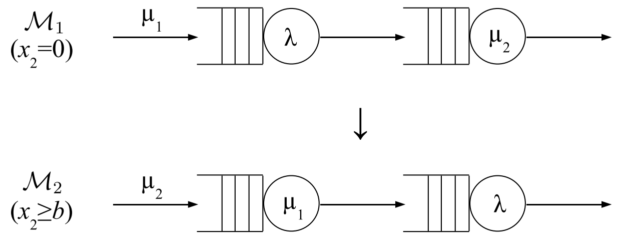


Figure 3.3: the SDH change of measure

### 3.2.3 State-dependent heuristic (SDH)

Let $\mathbf{x} = (x_1, x_2)$ be the state of the network at some time $t$. Define as $\tilde{\lambda}$, $\tilde{\mu}_1$, $\tilde{\mu}_2$ the arrival and the service rates corresponding to the importance sampling change of measure. Note that $\tilde{\lambda}$, $\tilde{\mu}_1$, $\tilde{\mu}_2$ may, in general, depend on the state of the network and, thus, are functions of the buffer contents $x_1$ and $x_2$. As before (see Equation (3.2)), we assume that $\tilde{\lambda} + \tilde{\mu}_1 + \tilde{\mu}_2 = 1$.

Define $[a]^+ = \max(a, 0)$ and $[a]^1 = \min(a, 1)$, and let $0 \le b \le N$ be a fixed integer. The following equations describe the proposed change of measure (SDH) for the 2-node tandem network:

$$\tilde{\lambda}(x_2) = \left[\frac{b - x_2}{b}\right]^+ \cdot \mu_1 + \left[\frac{x_2}{b}\right]^1 \cdot \mu_2, \tag{3.3}$$

$$\tilde{\mu}_1(x_2) = \left[\frac{b - x_2}{b}\right]^+ \cdot \lambda + \left[\frac{x_2}{b}\right]^1 \cdot \mu_1, \tag{3.4}$$

$$\tilde{\mu}_2(x_2) = \left[\frac{b - x_2}{b}\right]^+ \cdot \mu_2 + \left[\frac{x_2}{b}\right]^1 \cdot \lambda, \tag{3.5}$$

$$\tilde{\mu}_2(0,1) = 0. \tag{3.6}$$

Note that, except for Equation (3.6), the new arrival and service rates depend on the state of the network only through $x_2$, the buffer content at the second node, and as long as $x_2 < b$. When $x_2$ exceeds value $b$ SDH no longer changes.

Note also that since $\tilde{\lambda} + \tilde{\mu}_1 + \tilde{\mu}_2 = 1$, Equation (3.6) implies $\tilde{\lambda}(0,1) = 1$ (since $\tilde{\mu}_1(0,1) = 0$ and $\tilde{\mu}_2(0,1) = 0$) and guarantees that during the simulation all cycles hit the rare set (the overflow level $N$).

The above heuristic (SDH) is a combination of two changes of measure ($\mathcal{M}_1$) and ($\mathcal{M}_2$) (as indicated schematically in Figure 3.3). Along the boundary, $x_2 = 0$, the change of measure is $\mathcal{M}_1$ given by:

$$\mathcal{M}_1 : \begin{cases} \tilde{\lambda} = \mu_1, \\ \tilde{\mu}_1 = \lambda, \\ \tilde{\mu}_2 = \mu_2. \end{cases}$$

When $x_2 \ge b$, the change of measure is $\mathcal{M}_2$ given by:

$$\mathcal{M}_2 : \begin{cases} \tilde{\lambda} = \mu_2, \\ \tilde{\mu}_1 = \mu_1, \\ \tilde{\mu}_2 = \lambda. \end{cases}$$

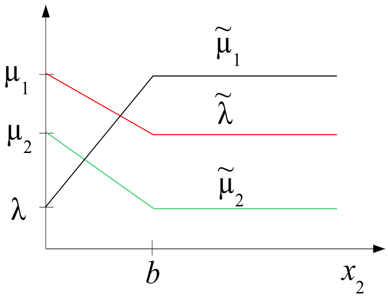


Figure 3.4: Change of rates in SDH

Roughly speaking, $\mathcal{M}_1$ is SDH at $b = \infty$; $\mathcal{M}_2$ is SDH at $b = 0$. In the interim, $0 < x_2 < b$, the new rates are simply linear functions of $x_2$, i.e., linear interpolation from their values at $x_2 = 0$ to their values at $x_2 = b$ (see Figure 3.4). The proposed change of measure indeed can (roughly) follow the most likely path to population overflow, discussed in Section 3.2.2. Let us follow a sample path starting from an arrival to an empty network. SDH implies the following: initially, and while $x_2 = 0$, exchange the arrival rate ($\lambda$) with the service rate at node 1 ($\mu_1$), i.e., start with the first node being unstable and the second node is stable. As the buffer content in the second buffer increases in the range ($0 < x_2 < b$), gradually and simultaneously reduce the "load" on the first node while increasing the "load" on the second node. When the buffer content at the second node reaches (and exceeds) level $b$, exchange the arrival rate ($\lambda$) with the service rate at node 2 ($\mu_2$). That is, as long as $x_2 \geq b$ the second node is unstable and the first node is stable (if $\mu_1 > \mu_2$) or unstable (if $\mu_1 < \mu_2$), and the new rates do not depend on the network state.

**Remark 3.2.1.** Note that the only variable parameter in the above heuristic is a number $b$, called *the boundary level*, for which the change of measure depends on the network state. Proper selection of $b$ is crucial for asymptotic efficiency of the proposed heuristic. In Section 3.6.1 we discuss the algorithm for finding the optimal value $b$ (the $b$ which yields estimates with the lowest variance). According to experimental results (Sections 3.6) the best value of $b$ depends on the network parameters and, in some cases, also on the overflow level $N$.

**Remark 3.2.2.** Note that without Equation (3.6) SDH is equal, on its extremes (i.e. when $b = 0$ or $b = \infty$) to PW; $\mathcal{M}_1$ is PW for $\mu_1 < \mu_2$ and $\mathcal{M}_2$ is PW for $\mu_1 > \mu_2$ (by definition, $\mathcal{M}_2$ is applied when $x_2 \geq b$, i.e., always in case $b = 0$ (in that case SDH=$\mathcal{M}_2$) and never in case $b = \infty$ (in that case SDH=$\mathcal{M}_1$)).

**Remark 3.2.3.** The above heuristic is a combination of two possible scenarios discussed in Section 3.2.2 (i.e., it is either, node 1 is overloaded ("pushed"), or node 2). We also experimented with other combinations, i.e., tried to include the scenario when both nodes are overloaded ("pushed") simultaneously, but it was not successful. Apparently, "pushing" both nodes together only degrades the performance and even "pushing" one node too much can already do that (as evidenced from the Section below).

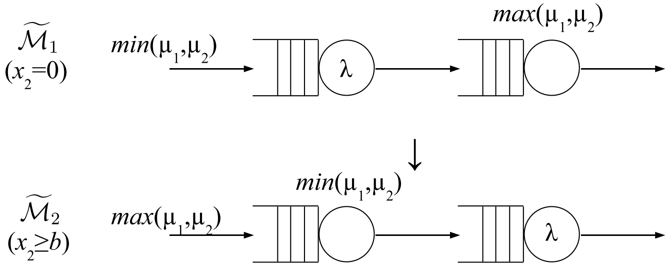


Figure 3.5: the SDHI change of measure

### 3.2.4 Improved heuristic (SDHI)

It is important to note that the most likely path to the rare set (as predicted from time reversal) may not necessarily correspond to the actual (or "optimal") one, particularly along the boundaries (i.e., when one of the nodes is empty). Thus, a proper adjustment to the proposed change of measure along the boundaries may result in significant efficiency improvement. Indeed, it turns out (as we empirically observed, see Sections 3.5.1–3.5.2) that the proposed change of measure (SDH) described in Section 3.2.3 tends to "over-bias" unstable nodes along and close to the boundaries, i.e., $x_1$ and/or $x_2$ close to 0. Let us consider, for example, the case $x_1 = 0$ and $x_2 \geq b$ (i.e., far from the influence of the border $x_2 = 0$). Then SDH $= \mathcal{M}_2$ and for $\mu_1 < \mu_2$ both nodes will be unstable. On the boundary $x_2 = 0$ the change of measure SDH $= \mathcal{M}_1$; the first queue is unstable and in case $\mu_1 > \mu_2$ it may lead to "over pushing" the first node, and, hence, make a heuristic less effective.

This observation prompted the modified change of measure, described as follows (in the sequel, we refer to it as SDHI):

$$SDHI : \begin{cases} \tilde{\lambda}(x_2) &= \min(\mu_1, \mu_2), \\ \tilde{\mu}_1(x_2) &= \left[\dfrac{b - x_2}{b}\right]^+ \cdot \lambda \,+\, \left[\dfrac{x_2}{b}\right]^1 \cdot \max(\mu_1, \mu_2), \\ \tilde{\mu}_2(x_2) &= \left[\dfrac{b - x_2}{b}\right]^+ \cdot \max(\mu_1, \mu_2) \,+\, \left[\dfrac{x_2}{b}\right]^1 \cdot \lambda, \\ \tilde{\mu}_2(0,1) &= 0. \end{cases}$$

where $[a]^+ = \max(a, 0)$ and $[a]^1 = \min(a, 1)$, and $b$ is a fixed integer between 0 and $N$ (i.e. $0 \leq b \leq N$).

Note that, unlike the SDH change of measure, the arrival rate in SDHI is independent of $x_2$. The modified heuristic (SDHI) suggests two changes of measures ($\widetilde{\mathcal{M}}_1$) and ($\widetilde{\mathcal{M}}_2$) (as indicated schematically in Figure 3.5).

Along the boundary, $x_2 = 0$, the change of measure ($\widetilde{\mathcal{M}}_1$) is given by:

$$\widetilde{\mathcal{M}}_1 : \begin{cases} \tilde{\lambda} &= \min(\mu_1, \mu_2), \\ \tilde{\mu}_1 &= \lambda, \\ \tilde{\mu}_2 &= \max(\mu_1, \mu_2). \end{cases}$$

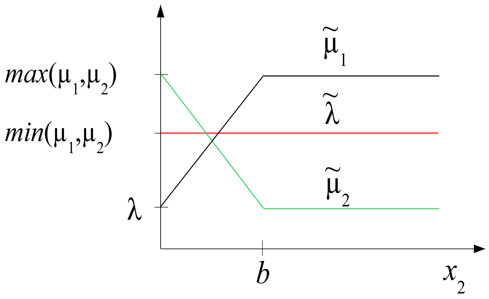


Figure 3.6: Change of rates in SDHI

When $x_2 \geq b$, the change of measure $(\widetilde{\mathcal{M}}_2)$ is given by:

$$\widetilde{\mathcal{M}}_2 : \begin{cases} \tilde{\lambda} & = & \min(\mu_1, \mu_2), \\ \tilde{\mu}_1 & = & \max(\mu_1, \mu_2), \\ \tilde{\mu}_2 & = & \lambda. \end{cases}$$

Note that, similar to the SDH change of measure, the new rates for SDHI are also linear functions of $x_2$, but with the new arrival rate being a constant (see Figure 3.6).

Let us follow a sample path starting from an arrival to an empty network. The proposed change of measure implies the following: the arrival rate is fixed at the minimum service rate; initially, and while $x_2 = 0$, node 1 is unstable with service rate $\lambda$, and node 2 is stable with service rate $\mu_2$ if $\mu_2 > \mu_1$ and the service rate $\mu_1$ if $\mu_1 > \mu_2$. When the buffer content at node 2 reaches (and exceeds) level $b$, node 2 is unstable with service rate $\lambda$, and node 1 is stable (if $\mu_1 \neq \mu_2$) or "critical" (if $\mu_1 = \mu_2$).

**Remark 3.2.4.** Note that in case $\mu_1 = \mu_2$ SDH and SDHI are equivalent (i.e., SDH≡SDHI; henceforward, by sign "≡" we mean "equivalent").

**Remark 3.2.5.** Note also that $\widetilde{\mathcal{M}}_1 = \mathcal{M}_1$ for $\mu_1 < \mu_2$ and $\widetilde{\mathcal{M}}_2 = \mathcal{M}_2$ for $\mu_1 > \mu_2$, i.e., SDH and SDHI behave the same way in case $\mu_1 < \mu_2$ on the border $x_2 = 0$ and in case $\mu_1 > \mu_2$ when $x_2 \geq b$; or, this can be reformulated as: SDHI differs from SDH in case $\mu_1 > \mu_2$ on the border $x_2 = 0$ and in case $\mu_1 < \mu_2$ for $x_2 \geq b$ (and, consequently, also on the border $x_1 = 0$).

For the 2-node tandem network, the above modified heuristic (SDHI) performs at least as good as SDH (see experimental results in Sections 3.5.1) and in most of the cases even better (cf. Section 3.6.1).

## 3.3 Multiple-node tandem networks

In this section we discuss extension of the proposed heuristics SDH and SDHI for networks with more than two nodes in tandem.

### 3.3.1 State-dependent heuristic (SDH)

As before, let $\lambda$ and $\mu_i$ $(i = 1, \ldots, d)$ be the arrival rate at the first node and the service rate at the $i^{th}$ node, respectively. Without loss of generality we assume that $\lambda + \sum_{i=1}^{d} \mu_i = 1$. Denote by $\tilde{\lambda}, \tilde{\mu}_i$ the corresponding rates under the new change of measure, and by $\mathbf{SDH}_d$ the $(d+1) \times (d+1)$ SDH transformation matrix for the $d$-node tandem network. Thus, $\mathbf{SDH}_d$ is a linear operator transforming the original rates into the new rates. In Section 3.2.3 we used different representation of SDH to give a reader the idea of how it works. In this section we aim to extend the heuristic for more queues and, thus, want it to be more compact. For $d = 2$, the change of measure in Section 3.2.3 can now be expressed as follows

$$\begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu}_1 \\ \tilde{\mu}_2 \end{bmatrix} = \mathbf{SDH}_2 \cdot \begin{bmatrix} \lambda \\ \mu_1 \\ \mu_2 \end{bmatrix}, \tag{3.7}$$

$$\tilde{\mu}_2(0,1) = 0, \tag{3.8}$$

where

$$\mathbf{SDH}_2 = \left[\frac{b - x_2}{b}\right]^+ \cdot \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \left[\frac{x_2}{b}\right]^1 \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3.9}$$

The first matrix is the identity matrix with the first and the second rows interchanged; this corresponds to interchanging the arrival rate $\lambda$ with the service rate $\mu_1$. The second matrix is the identity matrix with the first and the third rows interchanged; this corresponds to interchanging the arrival rate $\lambda$ with the service rate $\mu_2$.

The above heuristic can be generalized for a $d$-node tandem network. Define the transformation matrix $\mathbf{SDH}_d$ recursively as follows:

$$\mathbf{SDH}_k = \left[\frac{b_k - x_k}{b_k}\right]^+ \cdot \mathbf{SDH}_{k-1} + \left[\frac{x_k}{b_k}\right]^1 \cdot \mathbf{I}_k, \quad k = 2, \ldots, d, \tag{3.10}$$

with

$$\mathbf{SDH}_1 = \mathbf{I}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 \end{bmatrix}, \tag{3.11}$$

$\mathbf{I}_k$ is the identity matrix of dimension $(d+1)$ with the first and the $(k+1)$-st rows interchanged. Then, SDH for an $d$-node tandem network is given by

$$\begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu}_1 \\ \vdots \\ \tilde{\mu}_d \end{bmatrix} = \mathbf{SDH}_d \cdot \begin{bmatrix} \lambda \\ \mu_1 \\ \vdots \\ \mu_d \end{bmatrix}, \tag{3.12}$$

$$\tilde{\mu}_d(0, ..., 0, 1) = 0. \tag{3.13}$$

**Remark 3.3.1.** Note that the parameter $b$ in the above heuristic became $b_k$, i.e., the number of boundary layers for which SDH depends on the number of customers at queue $k$ may now be different from one queue to another (for 2-node tandem queues we omitted the index 2, i.e., there $b = b_2$). In Section 3.6.2 we will discuss this issue in more detail and will give a guideline for finding the $b_{k(opt)}$ for each $k$.

Note also that for $d = 1$ (a single queue), $\mathbf{SDH}_d$ corresponds to the PW heuristic of interchanging the arrival rate $\lambda$ and the service rate $\mu$ [15]. For $d = 3$, the transformation matrix $\mathbf{SDH}_3$ is as follows:

$$\mathbf{SDH}_3 = \left[\frac{b_3 - x_3}{b_3}\right]^+ \cdot \left(\left[\frac{b_3 - x_2}{b_3}\right]^+ \cdot \mathbf{I}_1 + \cdot \left[\frac{x_2}{b_2}\right]^1 \cdot \mathbf{I}_2\right) + \left[\frac{x_3}{b_3}\right]^1 \cdot \mathbf{I}_3, \tag{3.14}$$

where

$$\mathbf{I}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{I}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{I}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{3.15}$$

where the first matrix ($\mathbf{I}_1$) corresponds to interchanging $\lambda$ and $\mu_1$, the second matrix ($\mathbf{I}_2$) corresponds to interchanging $\lambda$ and $\mu_2$ and the third matrix ($\mathbf{I}_3$) corresponds to interchanging $\lambda$ and $\mu_3$. Initially, the network is empty and we start by interchanging the arrival rate $\lambda$ and $\mu_1$, i.e., biasing the first node. As soon as a customer arrives at node 2, we also start biasing the second node by gradually increasing the weight of matrix $\mathbf{I}_2$ and reducing the weight of matrix $\mathbf{I}_1$. When the number of customers at node 2 is sufficiently large ($x_2 > b_2$), the weight of matrix $\mathbf{I}_1$ becomes 0. In the meantime, as soon as a customer arrives at node 3, we start biasing the third node by gradually increasing the weight of matrix $\mathbf{I}_3$ and reducing the weights of matrices $\mathbf{I}_1$ and $\mathbf{I}_2$. When the number of customers at node 3 is sufficiently large ($x_3 > b_3$), the weights of matrices $\mathbf{I}_1$ and $\mathbf{I}_2$ become 0.

### 3.3.2 Improved heuristic (SDHI)

As discussed in Section 3.2.4 for the 2-node tandem case, the SDH change of measure can be improved. This improved version (SDHI) can also be extended for more than 2 queues in tandem. From now on we consider only the case of *ordered queues*, i.e., $\mu_i > \mu_{i+1}$ for $i = 1, ..., d-1$, which is enough according to the interchangeability argument given in [45]. Let us, first, reformulate the SDHI change of measure for a 2-node tandem network ($\mu_1 \geq \mu_2$). More formally it can be described as follows:

$$\begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu}_1 \\ \tilde{\mu}_2 \end{bmatrix} = \mathbf{SDHI}_2 \cdot \begin{bmatrix} \lambda \\ \mu_1 \\ \mu_2 \end{bmatrix}, \tag{3.16}$$

$$\tilde{\mu}_2(0,1) = 0, \tag{3.17}$$

where

$$\mathbf{SDHI}_2 = \left[\frac{b - x_2}{b}\right]^+ \cdot \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} + \cdot \left[\frac{x_2}{b}\right]^1 \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3.18}$$

The second matrix corresponds to interchanging the arrival rate $\lambda$ and the service rate at the second node ($\mu_2$) (and is equal to the identity matrix with the first and the last rows interchanged). The first matrix corresponds to the change of measure: $\tilde{\lambda} = \mu_2$, $\tilde{\mu}_1 = \lambda$ and $\tilde{\mu}_2 = \mu_1$, i.e., the last row of the identity matrix becomes the first row, the first row becomes the second and the second becomes the third (i.e. first two rows are shifted down).

The generalization of the alternative heuristic (SDHI) for $d$ nodes in tandem can be formalized as follows:

$$\mathbf{SDHI}_k \;=\; \left[\frac{b_k - x_k}{b_k}\right]^+ \cdot \mathbf{SDHI}_{k-1} + \left[\frac{x_k}{b_k}\right]^1 \cdot \mathbf{J}_k, \quad k = 2, \ldots, d\,, \tag{3.19}$$

with

$$\mathbf{SDHI}_1 \;=\; \mathbf{J}_1 \;=\; \begin{bmatrix} 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix}. \tag{3.20}$$

Then SDHI for an $d$-node tandem network is given by

$$\begin{bmatrix} \tilde{\lambda} \\ \tilde{\mu}_1 \\ \vdots \\ \tilde{\mu}_d \end{bmatrix} \;=\; \mathbf{SDHI}_d \cdot \begin{bmatrix} \lambda \\ \mu_1 \\ \vdots \\ \mu_d \end{bmatrix}, \tag{3.21}$$

$$\tilde{\mu}_d(0, ..., 0, 1) \;=\; 0. \tag{3.22}$$

$\mathbf{J}_k$ is constructed as follows: in the identity matrix of dimension $(d+1)$ all rows $i \geq k+1$ are pushed down (by one row); the first row goes to the $(k+1)$-st row (which is now empty) and the last row (which is pushed out) goes to the first row. This corresponds to $\tilde{\mu}_k = \lambda$, $\tilde{\mu}_i = \mu_{i-1}$ for $i = k+1, \ldots, d$, $\tilde{\lambda} = \mu_d$ and means the following: we "push" node $k$ by decreasing its service rate, but, unlike $\mathbf{SDH}_d$ (where it was done by interchanging the arrival rate $\lambda$ with the service rate at the $k^{th}$ node ($\mu_k$)), $\mathbf{SDHI}_d$ makes the new arrival rate always equal to $\mu_d$ (i.e. "pushes" node $k$ less since $\mu_d$=min($\mu_i$)); all the service rates up and including to node $k-1$ stay unchanged, i.e., all queues are stable (if $\mu_i > \mu_{i+1}$ for all $i$) or some of them become unstable (if for some $i$ $\mu_i = \mu_{i+1}$); the service rate at node $k$ is equal to $\lambda$ (i.e. queue $k$ is unstable) and all the service rates from node $k+1$ get the "available" (we consider only permutations of rates) values of the service rates in the descending order, i.e. the service rate at node $k+1$ becomes equal to $\mu_k$, the service rate at node $k+2$ becomes equal to $\mu_{k+1}$ and so on. In case of a 3-node tandem network we have the following:

$$\mathbf{SDHI}_3 = \left[\frac{b_3 - x_3}{b_3}\right]^+ \cdot \left( \left[\frac{b_2 - x_2}{b_2}\right]^+ \cdot \mathbf{J}_1 + \left[\frac{x_2}{b_2}\right]^1 \cdot \mathbf{J}_2 \right) + \left[\frac{x_3}{b_3}\right]^1 \cdot \mathbf{J}_3, \tag{3.23}$$

where

$$\mathbf{J}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{J}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \mathbf{J}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.24}$$

$\mathbf{J}_3$ corresponds to "pushing" node 3 by interchanging the arrival rate $\lambda$ with the service rate $\mu_3$ ($\tilde{\lambda} = \mu_3$, $\tilde{\mu}_1 = \mu_1$, $\tilde{\mu}_2 = \mu_2$, $\tilde{\mu}_3 = \lambda$); $\mathbf{J}_2$ corresponds to "mild pushing" of node 2, i.e., by making $\tilde{\lambda} = \mu_3$, $\tilde{\mu}_1 = \mu_1$, $\tilde{\mu}_2 = \lambda$, $\tilde{\mu}_3 = \mu_2$ and $\mathbf{J}_1$ corresponds to "mild pushing" of node 1, i.e. $\tilde{\lambda} = \mu_3$, $\tilde{\mu}_1 = \lambda$, $\tilde{\mu}_2 = \mu_1$, $\tilde{\mu}_3 = \mu_2$.

Note that both heuristics SDH and SDHI depend on number of customers at all nodes except the first. However, unlike SDH, in SDHI the new arrival rate is independent of $x_i$, $1 \le i \le d$ and is equal to $\min(\mu_1, \ldots, \mu_d)$. Empirical results in Sections 3.5.1–3.5.2 confirm that the modified heuristic (SDHI) performs slightly better than SDH.

## 3.4 Performance comparison

In this section we describes general issues concerning simulation (Section 3.4.1) and discuss methods used for performance comparison together with restrictions used while gathering the experimental results (Section 3.4.2).

### 3.4.1 Simulation

Define a *busy cycle* as the period starting with an empty system and ending at the instant the system, for the first time, either reaches level $N$ or becomes empty again. Starting a cycle at time 0, let $T_N$ define the first time the network population reaches level $N$ and $T_0$ the first time the network population returns to 0 again. Importance sampling simulation to estimate the probability of population overflow $\gamma(N)$ involves generating, say, $n$ of those cycles with the new probability distribution, e.g., SDH or SDHI, chosen in the beginning and used during the entire simulation. The indicator function $I_i(T_N < T_0)$ takes the value 1 if cycle $i$ ended at level $N$ and the value 0, otherwise. Then an unbiased estimator $\tilde{\gamma}(N)$ of $\gamma(N)$ is given by

$$\tilde{\gamma}(N) = \frac{1}{n} \sum_{i=1}^{n} I_i \, L_i \, , \tag{3.25}$$

where $L_i$ is the likelihood ratio associated with cycle $i$. The second moment ($\sigma^2$) of the random variable $I \cdot L$ (the indicator function of the rare event multiplied by the likelihood ratio) is estimated by

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} I_i \, {L_i}^2 \, . \tag{3.26}$$

The variance $Var(\tilde{\gamma}(N))$ and the relative error $RE(\tilde{\gamma}(N))$ of the importance sampling estimator $\tilde{\gamma}(N)$ are given by:

$$Var(\tilde{\gamma}(N)) = \frac{1}{n-1} \left( \sigma^2 - (\tilde{\gamma}(N))^2 \right), \tag{3.27}$$

$$RE(\tilde{\gamma}(N)) \;=\; \frac{\sqrt{Var(\tilde{\gamma}(N))}}{\tilde{\gamma}(N)}\;. \tag{3.28}$$

*RE* can be used to check the performance of the importance sampling estimator (an estimator with bounded *RE* would give the same *RE* for any value of level $N$).

It can also be used to compare the efficiency of different estimators (estimators obtained with different changes of measure), but only for those that work equally fast, i.e., require approximately the same amount of simulation time per number of simulation cycles. In that case, the change of measure with smaller *RE* is the best of these two. In general, however, for the same number of (simulation) cycles, different changes of measure may need different amount of (simulation) time. Since *RE* does not include simulation time, it can not be applied for a fair comparison of such measures. In that case *the relative time variance* ($RTV$) product is employed, which is defined as the simulation time $T$ (in seconds) multiplied by the squared relative error of the estimator:

$$RTV \;=\; T \cdot RE^2. \tag{3.29}$$

$RTV$ is a good measure of comparison between two changes of measures since it reflects not only the $RE$ of the estimator but also takes into account the effort (simulation time) needed to get this $RE$. Thus, the change of measure with the smallest (over some set of changes of measure) $RTV$ is *optimal* in the sense that it takes the smallest amount of time to get the same $RE$. In other words, if $RTV_1$ for estimator 1 is smaller than $RTV_2$ for estimator 2, then it will take estimator 1 a shorter simulation time to reach the same accuracy. Note, that for the above described case (when two estimators have the same simulation time $T = T_1 = T_2$) comparing $RTV$'s indeed reduces to comparing $RE$'s since $RTV_k = T \cdot {RE_k}^2$ and $RTV_i > RTV_j \Leftrightarrow RE_i > RE_j$. For a large number of samples $n$ the simulation time $T$ is proportional[1] to $n$. We call the estimate *stable* if it has a bounded variance (note, that "being stable" is a "normal" characteristic of a direct estimator and a "well behaving" IS estimator). The sample variance of a stable estimator is inversely proportional to $n$, $RE$ is inversely proportional to $\sqrt{n}$ and $RTV$ tends to a constant value, which is smaller for a more efficient estimator.

We introduce the efficiency gain of using estimator 1 over estimator 2 as *the variance reduction ratio* (e.g., [21]):

$$VRR = \frac{RTV_2}{RTV_1}. \tag{3.30}$$

Namely, $VRR > 1$ if the estimator 1 is more efficient than estimator 2, and $VRR \leq 1$ otherwise. We use this ratio for comparing the efficiency of different importance sampling heuristics in the next section.

---

[1] $T = \sum_{i=1}^{n} t_i$ where $t_i$ is time needed for simulating cycle $i$; since $t_i$'s are independent, for large $n$ the sum $\sum_{i=1}^{n} t_i/n$ tends to a constant value $\bar{t}$ and $T = n \cdot \bar{t}$

### 3.4.2 Methods used for performance comparison

We run simulation experiments with three different methods:

1) the PW heuristic to show where it does, or does not work;
2) the SDA method, the state-dependent change of measure determined using the adaptive methodology defined in [32] and discussed in Section 2.5. In our comparison we used two out of three techniques described in [32], namely, local average and boundary layers, and did not use spline fitting;
3) the SDH and SDHI (state-dependent) heuristics described in Section 3.2.3 and Section 3.2.4, respectively (typed SDHs, when referring to both of them).

We also included numerical results (using the algorithm outlined in [32], [27]) to verify the correctness of the simulation estimates.

**Remark 3.4.1.** The numerical algorithm in [32], [27] involves inverting $O(N)$ number of matrices of size $O(N^{d-1})$ where $N$ is the overflow level and $d$ is the number of queues in the network. Thus, already for networks with more than two nodes, the size of the matrices grows very quickly with the overflow level, which restricts the applicability of the method to only small networks. For three queues we were able to calculate the exact probabilities for levels up to $N = 50$; for four queues we could to do that only for levels not larger than $N = 25$.

The state-dependent heuristics (SDA and both SDHs) assume dependence on the network state, but only for some small number $b$ of boundary layers: for SDA dependence is on states $(x_1, x_2)$, where $0 \leq x_1, x_2 \leq b$ and for SDHs on states $(., x_2)$, where $0 \leq x_2 \leq b$, i.e., in SDHs there is no dependence on $x_1$, the number of customers at the first queue. In both SDHs or SDA, the best (i.e., the optimal) $b$ can be determined by repeating the simulation for increasing values of $b$, starting with $b = 0$, i.e., no state-dependence. For the SDA algorithm we started with $b = 1$ since it is the minimum possible value of $b$ that can be used (without specifically defining what SDA would mean in case $b = 0$). The best $b$ is the one that yields the maximum efficiency (or minimum $RTV$).

**Remark 3.4.2.** It is important to note that it is very difficult (if possible at all) to ensure fair comparison between the SDA and SDHs methods since they are fundamentally different. The main difference lies in the fact that SDA, in principle, has *quadratic convergence*[2], i.e., the $RE$ of a stable estimator decreases proportionally to the number of cycles $n$ (not $\sqrt{n}$ as for SDHs), and, hence, $RTV$ *decreases* with increasing $n$ (i.e., *it is not a constant* value as an $RTV$ of a stable SDHs estimate).

Thus, if we run all methods long enough, i.e., enough to ensure that SDA converges (in case it does) and gets small $RTV$, and, the same amount of time for SDHs, then SDA will "win" (if converged), i.e., will give the smallest $RTV$ and, at the same time, the smallest $RE$. The problem, however, is that the simulation time required to be able to achieve that is sometimes prohibitively long (several hours), although

---

[2] this is the characteristic of "unlimited" SDA, i.e., SDA without any restrictions (like boundary layers, local average or splines); applying one or more of those techniques may lead to the degradation of this feature

in such cases we might get $RE$ for SDA less than 0.001%. In practice, one is usually satisfied with $RE$ of 1% especially if it means short simulation time. One can reduce the amount of simulation time by reducing the number of simulation cycles $n$, but for SDA this is not always possible, since having less cycles may not be enough for convergence. Note also, that there exist situations where for a given number of replications SDA does not converge. In such cases SDHs always win. Later in this section (for the case of 3 and 4 queues in tandem, cf. Remark 3.5.2) we will discuss this issue in more detail.

**Restrictions used while gathering the results**

To ensure somewhat fair comparison, we made several restrictions:

1) SDA was run at $10^5$ cycles until $RE$ became less than 1% and then two iterations at $10^6$ cycles were performed. Due to quadratic convergence the second iteration usually decreased $RE$ of the SDA estimator by $\sqrt{10}$ and, hence, increased the performance by a factor of 10. Further iterations did not give such an improvement. For comparison, only the second iteration (at $10^6$ cycles) was considered;

2) in SDHs the estimates were calculated for $10^6$ replications;

3) in both SDHs and SDA we assumed that the optimal parameter $b$ was predetermined and used to run the simulation.

   The time needed to get these optimal values was not included since in all cases (SDA and both SDHs) we needed to find the $b_{opt}$ by trial and error. The amount of time spent on this was impossible to predict since it depends on how close the initial guess was to the optimal value. There was also another reason for not doing this: in Section 3.6.1 the full investigation of $b_{opt}$ on the network parameters would be done with the clear guideline which one to use for SDHI (Section 3.6.1, Proposition 3). Note that there would be no guideline how to choose the $b_{opt}$ for SDA or SDH; the only observation was that the $b_{opt}$ for SDH was very often near the $b_{opt}$ for SDHI.

## 3.5 Experimental results

In this section we consider tandem networks and present experimental results obtained using the SDH and SDHI heuristics proposed in Section 3.3.1 and Section 3.3.2, respectively. Sections 3.5.1–3.5.2 show the performance of the proposed changes of measure in comparison with other methods.

Since there is more background information available for a 2-node tandem networks in comparison with 3- and 4-node networks, we separate the discussion for the 2-node case in a distinct subsection.

### 3.5.1 Performance for 2 queues in tandem

As discussed in Section 2.4.2, the PW heuristic yields asymptotically efficient estimates with bounded relative error only for some values of feasible network parameters. In [25] it was shown that they can be classified in three different regions (see Figure 3.7):

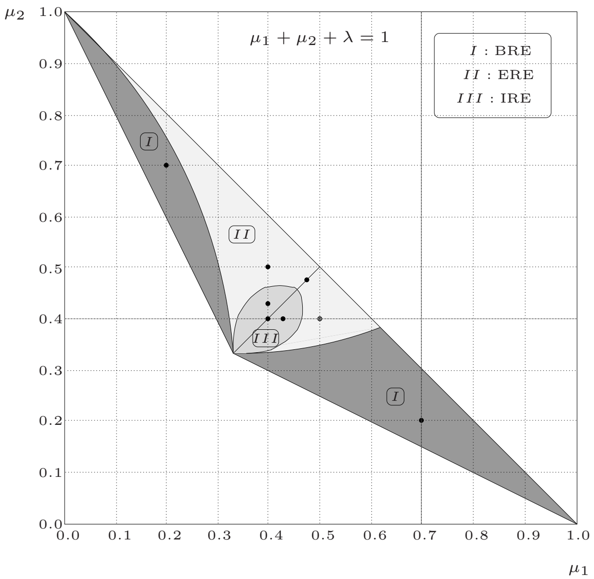


Figure 3.7: Asymptotic efficiency of PW in the feasible parameter space (as shown in [25])

BRE region: PW yields estimates with bounded relative error;
ERE region: PW yields estimates with exponentially growing relative error;
IRE region: PW yields estimates with infinite variance/relative error.

Thus, for the network parameters belonging to the BRE region, an asymptotically efficient state-independent change of measure with bounded relative error is already known (PW). For the ERE and IRE regions, only state-dependent change of measure, determined adaptively or using some heuristic guess, can be asymptotically efficient (see [25]). The goal of this section is to show that the SDH and SDHI heuristics have better performance than the other methods.

For this purpose we chose 8 different parameter points covering all regions in Figure 3.7 (these points are representative and the results are consistent with extensive simulations discussed in Section 3.6.1), namely, 2 points in each of the 3 regions (BRE, ERE, and IRE) and 2 points along the line $\mu_1 = \mu_2$ (networks with equal service rates appeared to be more difficult for simulation).

## Performance

Tables 3.1–3.8 (one table for each of the points indicated in Figure 3.7) include estimates of three overflow probability levels ($N$ = 25, 50 and 100) and their relative error (in percentage) obtained with PW, the SDHs and SDA changes of measure. The relative error is a good enough indicator for the performance comparison of the PW and SDHs estimators (since they “work” equivalently fast), but not for comparing

SDHs and SDA. SDA needs some time to converge and it is unclear how many iterations and how many repetitions per iteration should be performed to get a specified accuracy. Therefore, restrictions mentioned above were applied and another measure of comparison ($VRR$, as discussed in Section 3.4.1) was used. Hence, $VRR = 1$ for SDA, and a $VRR > 1$ implies efficiency gain over SDA.

A careful inspection of Table 3.1 through Table 3.8 leads to the following observations:

**BRE region** (Tables 3.1 and 3.2):
All heuristics yield very accurate estimates with bounded relative error. Note, however, that for Point I-1, SDH and SDHI have an "optimal" $b$ equal to 0. For this point, SDA outperforms the other heuristics ($VRR < 1$). For Point I-2, SDH and SDHI have an "optimal" $b = \infty$ and yield efficiency gains over SDA ($VRR > 1$) except for the level 100 ($VRR = 0.92$). As noted in Remark 3.2.1, the heuristics SDH and SDHI reduce to PW for $b = 0$ (if $\mu_1 > \mu_2$) and for $b = \infty$ (if $\mu_1 < \mu_2$) with the only difference that the transition ending a busy cycle in the "empty network" state is not allowed, i.e., SDH($\equiv$ SDHI) is just Equation (3.6). Since all cycles during the simulation end in the rare set, SDH and SDHI give the performance gain over PW.

**ERE region** (Tables 3.3 and 3.4):
Except for PW, all heuristics yield stable estimates. The '*' next to PW in the tables is to indicate that its estimates are not stable. For Point II-1, SDA, SDH, and SDHI give bounded relative error, and SDA outperforms SDH and SDHI ($VRR < 0.1$). It is not clear why SDA yields much lower relative error for this point than it does for any other point. For Point II-2, the relative errors seem to grow slowly and linearly with $N$, and SDH and SDHI yield efficiency gains over SDA ($2 < VRR < 7$).

**IRE region** (Tables 3.5 and 3.6):
Except for PW, all heuristics yield stable estimates. For Point III-1, SDA, SDH, and SDHI give bounded relative error, with SDH and SDHI being more efficient than SDA ($1 < VRR < 3$). For Point III-2, the relative errors seem to grow linearly with $N$, and SDH and SDHI yield efficiency gains over SDA ($VRR > 3$).

**ERE/IRE line** $\mu_1 = \mu_2$ (Tables 3.7 and 3.8):
Note that for Points IV-1 and IV-2, SDH and SDHI are the same. Except for PW, all heuristics yield stable estimates with bounded relative error. SDH and SDHI yield efficiency gains over SDA ($1 < VRR < 6$).

| N | Numerical | PW | SDA | | SDH ≡ SDHI | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm \boldsymbol{RE}\%$ | $b$ | $\tilde{\gamma}(N) \pm \boldsymbol{RE}\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 4.1723-08 | 4.1813e-08 ± 0.09 | 6 | 4.17217e-08 ± 4e-3 | 0 | 4.1757e-08 ± 0.05 | 0.13 |
| 50 | 1.2435-15 | 1.2438e-15 ± 0.09 | 5 | 1.24327e-15 ± 9e-3 | 0 | 1.2435e-15 ± 0.05 | 0.25 |
| 100 | 1.1044-30 | 1.1057e-30 ± 0.09 | 5 | 1.10430e-30 ± 5e-3 | 0 | 1.1046e-30 ± 0.05 | 0.05 |

Table 3.1: 2-node tandem network - BRE region. Point I-1 ($\lambda = 0.1,\ \mu_1 = 0.7,\ \mu_2 = 0.2$) ($\rho_1 = 0.143,\ \rho_2 = 0.5$)

| N | Numerical | PW | SDA | | SDH | | | SDHI | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 4.1723e-08 | 4.1614e-08 ± 0.12 | 3 | 4.1696e-08 ± 0.07 | $\infty$ | 4.1698e-08 ± 0.10 | 3.21 | $\infty$ | 4.1784e-08 ± 0.11 | 2.25 |
| 50 | 1.2435e-15 | 1.2445e-15 ± 0.12 | 3 | 1.2431e-15 ± 0.07 | $\infty$ | 1.2430e-15 ± 0.10 | 1.60 | $\infty$ | 1.2430e-15 ± 0.11 | 1.36 |
| 100 | 1.1044e-30 | 1.1038e-30 ± 0.12 | 3 | 1.0826e-30 ± 0.06 | $\infty$ | 1.1033e-30 ± 0.10 | 1.01 | $\infty$ | 1.1047e-30 ± 0.11 | 0.92 |

Table 3.2: 2-node tandem network - BRE region. Point I-2 ($\lambda = 0.1,\ \mu_1 = 0.2,\ \mu_2 = 0.7$) ($\rho_1 = 0.5,\ \rho_2 = 0.143$)

| N | Numerical | PW * | SDA | | SDH | | | SDHI | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 1.3269e-14 | 1.3897e-14 ± 5.02 | 3 | 1.32693e-14 ± 8e-3 | 3 | 1.3251e-14 ± 0.13 | 0.05 | 2 | 1.3266e-14 ± 0.10 | 0.08 |
| 50 | 1.1833e-29 | 1.1645e-29 ± 0.75 | 5 | 1.18324e-29 ± 1e-3 | 3 | 1.1822e-29 ± 0.13 | 2e-3 | 2 | 1.1840e-29 ± 0.09 | 5e-3 |
| 100 | 9.3345e-60 | 9.2549e-60 ± 0.97 | 4 | 9.33365e-60 ± 5e-3 | 3 | 9.3341e-60 ± 0.13 | 0.01 | 2 | 9.3085e-60 ± 0.09 | 0.02 |

Table 3.3: 2-node tandem network - ERE region. Point II-1 ($\lambda = 0.1,\ \mu_1 = 0.5,\ \mu_2 = 0.4$) ($\rho_1 = 0.2,\ \rho_2 = 0.25$)

| N | Numerical | PW * | SDA | | SDH | | | SDHI | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 1.3269e-14 | 1.1333e-14 ± 4.95 | 3 | 1.3245e-14 ± 0.14 | 7 | 1.3251e-14 ± 0.20 | 5.86 | 7 | 1.3228e-14 ± 0.19 | 6.03 |
| 50 | 1.1833e-29 | 1.1576e-29 ± 10.5 | 3 | 1.1877e-29 ± 0.23 | 7 | 1.1822e-29 ± 0.26 | 6.21 | 7 | 1.1835e-29 ± 0.27 | 5.07 |
| 100 | 9.3345e-60 | 8.8379e-60 ± 11.2 | 3 | 9.3462e-60 ± 0.38 | 7 | 9.3341e-60 ± 0.41 | 4.64 | 7 | 9.3346e-60 ± 0.49 | 2.77 |

Table 3.4: 2-node tandem network - ERE region. Point II-2 ($\lambda = 0.1,\ \mu_1 = 0.4,\ \mu_2 = 0.5$) ($\rho_1 = 0.25,\ \rho_2 = 0.2$)

| **N** | **Numerical** | **PW** $*$ | **SDA** | | **SDH** | | | **SDHI** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 3.8066e-08 | 4.2325e-08 ± 8.53 | 5 | 3.8046e-08 ± 0.04 | 6 | 3.8089e-08 ± 0.10 | 1.91 | 5 | 3.8056e-08 ± 0.08 | 2.98 |
| 50 | 1.0684e-16 | 1.1551e-16 ± 13.8 | 7 | 1.0681e-16 ± 0.02 | 6 | 1.0687e-16 ± 0.08 | 1.79 | 5 | 1.0688e-16 ± 0.07 | 2.49 |
| 100 | 5.3355e-34 | 3.8945e-34 ± 3.00 | 6 | 5.3357e-34 ± 0.03 | 6 | 5.3390e-34 ± 0.07 | 2.51 | 5 | 5.3372e-34 ± 0.07 | 2.97 |

Table 3.5: 2-node tandem network - IRE region. Point III-1 ($\lambda = 0.18,\ \mu_1 = 0.42,\ \mu_2 = 0.4$) ($\rho_1 = 0.429,\ \rho_2 = 0.45$)

| **N** | **Numerical** | **PW** $*$ | **SDA** | | **SDH** | | | **SDHI** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 3.8066e-08 | 3.1401e-08 ± 6.77 | 4 | 3.8044e-08 ± 0.08 | 9 | 3.8075e-08 ± 0.13 | 3.62 | 9 | 3.7993e-08 ± 0.12 | 3.95 |
| 50 | 1.0684e-16 | 5.7496e-17 ± 3.54 | 5 | 1.0667e-16 ± 0.16 | 9 | 1.0683e-16 ± 0.14 | 18.2 | 9 | 1.0678e-16 ± 0.14 | 17.9 |
| 100 | 5.3355e-34 | 2.9055e-34 ± 4.52 | 5 | 5.3364e-34 ± 0.27 | 9 | 5.3187e-34 ± 0.26 | 5.46 | 9 | 5.3556e-34 ± 0.28 | 4.57 |

Table 3.6: 2-node tandem network - IRE region. Point III-2 ($\lambda = 0.18,\ \mu_1 = 0.4,\ \mu_2 = 0.42$) ($\rho_1 = 0.45,\ \rho_2 = 0.429$)

| **N** | **Numerical** | **PW** $*$ | **SDA** | | **SDH** $\equiv$ **SDHI** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 2.8722e-025 | 2.6050e-025 ± 8.64 | 3 | 2.8729e-025 ± 0.06 | 3 | 2.8767e-025 ± 0.13 | 4.11 |
| 50 | 6.0327e-052 | 2.3672e-052 ± 4.67 | 3 | 6.0340e-052 ± 0.07 | 3 | 6.0367e-052 ± 0.12 | 3.89 |
| 100 | 1.3270e-105 | 3.5984e-106 ± 19.5 | 3 | 1.3255e-105 ± 0.07 | 3 | 1.3301e-105 ± 0.17 | 1.63 |

Table 3.7: 2-node tandem network - ERE region. Point IV-1 ($\lambda = 0.04,\ \mu_1 = 0.48,\ \mu_2 = 0.48$) ($\rho_i = 0.083$)

| N | **Numerical** | **PW** | **SDA** | | **SDH ≡ SDHI** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ (exact $RE\%$) | $VRR$ |
| 25 | 7.1526e-07 | 6.8876e-07 ± 3.71 | 7 | 7.1532e-07 ± 0.04 | 5 | 7.1517e-07 ± 0.11 (0.12) | 2.40 |
| | | | | | 6* | 7.1607e-07 ± 0.10 (0.10)* | 3.37 |
| | | | | | 7 | 7.1477e-07 ± 0.10 (0.10) | 2.99 |
| | | | | | 8 | 7.1556e-07 ± 0.12 (0.12) | 2.22 |
| | | | | | 9 | 7.1345e-07 ± 0.13 (0.14) | 1.72 |
| | | | | | 10 | 7.1512e-07 ± 0.16 (0.16) | 1.27 |
| 50 | 4.3521e-14 | 2.7323e-14 ± 5.99 | 7 | 4.3509e-14 ± 0.06 | 6 | 4.3540e-14 ± 0.14 (0.14) | 3.22 |
| | | | | | 7 | 4.3539e-14 ± 0.11 (0.11) | 5.35 |
| | | | | | 8* | 4.3563e-14 ± 0.10 (0.10)* | 6.31 |
| | | | | | 9 | 4.3552e-14 ± 0.11 (0.11) | 5.37 |
| | | | | | 10 | 4.3517e-14 ± 0.13 (0.12) | 3.87 |
| | | | | | 11 | 4.3564e-14 ± 0.14 (0.14) | 3.37 |
| 100 | 7.8097e-29 | 2.5780e-29 ± 11.8 | 7 | 7.8150e-29 ± 0.10 | 7 | 7.8280e-29 ± 0.21 (0.22) | 1.31 |
| | | | | | 8 | 7.8166e-29 ± 0.16 (0.16) | 2.19 |
| | | | | | 9 | 7.8038e-29 ± 0.13 (0.13) | 3.47 |
| | | | | | 10* | 7.8006e-29 ± 0.12 (0.13)* | 3.87 |
| | | | | | 11 | 7.7938e-29 ± 0.13 (0.13) | 3.40 |
| | | | | | 12 | 7.7952e-29 ± 0.14 (0.14) | 3.22 |

Table 3.8: 2-node tandem network - IRE region. Point IV-2 ($\lambda = 0.2$, $\mu_1 = 0.4$, $\mu_2 = 0.4$) ($\rho_i = 0.5$) ('*' denotes $b_{opt}$ or the minimum of exact $RE$)

**Remark 3.5.1.** Experimental results in Tables 3.1 through 3.8 show that the proposed heuristics SDH and SDHI yield estimates with bounded relative error for points with $\mu_1 > \mu_2$, i.e., points in the lower triangle, below the line $\mu_1 = \mu_2$ in Figure 3.7, as well as for points in the upper BRE region. Only for points in the upper ERE/IRE Regions II and III (with $\mu_1 \leq \mu_2$) the SDH and SDHI estimates have a linearly bounded relative error (see Tables 3.4 and 3.6). Note, however, that according to the interchangeability argument in [45], the probability of population overflow is invariant with respect to the placement order of nodes in a Jackson tandem network. Therefore, by interchanging the service rates ($\mu_1$ and $\mu_2$), the overflow probability for an arbitrary point in the upper ERE/IRE Regions II and III can also be estimated with bounded relative error.

#### Sensitivity with respect to $b$

Table 3.8 is extended to include sensitivity results with respect to the dependence range $b$ in the SDH for the 2-node tandem network. For different values of $b$ around the "best" (marked by '*' in the table), we display the resulting estimate along with its relative error, estimated from simulation and computed numerically. The latter is included in parentheses and is obtained from an algorithm similar to that outlined in [25] to compute the variance of the PW importance sampling estimator. Our implementation of this algorithm is adapted to compute the variance of the SDH estimator used in the above tandem network examples. As one can see from Table 3.8 the empirical relative error is consistent with the computed relative error. The accuracy of both the simulation estimates and the computed relative errors is quite robust with respect to $b$. Moreover, they numerically establish the bounded relative error property of the SDH estimator (which is also observed empirically). This is evidenced from the approximately equal computed relative errors at increasing overflow levels.

### 3.5.2 Performance for 3 and 4 queues in tandem

In this section we will demonstrate that SDHI gives performance gain over the PW and SDA methods also in case of 3 and 4 queues in tandem. We do not include results for SDH, since they showed worse performance; for the 2-node case we will experimentally show that in Section 3.6.1.

As for 2 queues in tandem, we divide the feasible parameter space into several regions, depending on the asymptotic properties of the PW change of measure:

BRE region - PW is asymptotically efficient (with bounded relative error)
NAE region - PW is not asymptotically efficient.

The above division, unlike the 2-node case, is based on empirical results, since the only conditions of asymptotic efficiency of the PW heuristic, discussed in [24], are rather strong and do not cover the entire parameter space even for the case of 2 nodes, i.e., not all points may be determined as BRE or NAE. The reason to use the above division is the fact that for 2-node tandem networks, as has been shown empirically in [25], all feasible network parameters are either in BRE or in NAE (for 2-node tandem NAE = ERE $\bigcup$ IRE). Our empirical studies seem to confirm that it holds also for tandem networks with 3 and 4 nodes, i.e., for any feasible set of network parameters, PW is either in BRE or in NAE.

Several experiments were made for 3 and 4 queues (one set of experiments for each network). Each set consists of two parameter points: one in the BRE region and another in the NAE region. In all simulation experiments, the same number of replications, namely $10^6$, is used to obtain estimates of the population overflow probability $\gamma(N)$. These estimates are presented in Table 3.9 through Table 3.12; two tables for each tandem network with a given number of nodes. For each estimate in these tables, we include the relative error (in percent). For the purpose of comparing the SDHI heuristic with SDA, we also include $VRR$ (relative to $RTV$ of SDA). When $VRR > 1$ it implies efficiency gain of SDHI over SDA. As before, we also assumed that the experimental results for SDHI and SDA are obtained using the optimal value of parameter $b$ for each algorithm, where for SDHI, $b$ means that all parameters $b_k$ are the same, i.e., $b_2 = b_3 = b$ for a 3-node tandem network and $b_2 = b_3 = b_4 = b$ for a 4-node tandem network.

Whenever feasible, numerical results (using the algorithm outlined in [32]) are included to verify the correctness of the simulation estimates. We note that numerical results are more difficult to obtain for larger networks and/or higher overflow levels (i.e., for larger state-space). With the available algorithm, we could not obtain numerical results for table entries marked with a '$*$'. In these cases, agreement of different estimators may provide an indication of correctness.

Experimental results in Tables 3.9–3.12 show that both, SDA and SDHI, yield correct and asymptotically efficient estimates. SDHI outperforms SDA in all cases except for level 100 for the $BRE$ region of a 4-node tandem network (Table 3.11). The last case, however, was special (Table 3.11, entry marked as '(!)'). The SDA algorithm (with the restrictions discussed in Section 3.4.2) did not converge and we needed to use a different approach. Apparently, the starting parameters were not adequate and $10^5$ replications were not enough. To overcome this difficulty, it was recommended in [32] to use SDA results of lower levels as an input parameters for higher levels (to improve the starting parameters). Thus, for level 100 the SDA results of level 25 were used and the first iterations were gathered at more replications ($5 \cdot 10^6$). Only in that case SDA converged; even using one of the two improvements did not help. We also tried the above approach for another point of 4-node tandem network and another levels. The results are shown in Tables 3.13–3.14. As one can see it improved the SDA results in some cases ($N = 50$ in Table 3.13 and $N = 100$ in Table 3.14) but made the performance in other ($N = 50$ in Table 3.14) worse. Thus, there is no guarantee for using one or another approach.

In our implementation of SDA we did not use the spline technique (described in [32] and discussed in Chapter 2.5.1), thus, one could argue that the comparison is not fair. To compensate that, we tried to approximate the run time of SDA-with-splines. As claimed in [32], the performance of SDA is better if one starts with splines using small number of replications, and then continues without splines with a larger number of replications. It was also claimed that with splines one could use less replications, namely, $10^4$ instead of $10^5$ and 10 iterations were enough for convergence, i.e., a total of about $10^5$ replications. Thus, for our approximation we took the time of the final two iterations (at $10^6$ cycles) of SDA, and added 10% to it as a rough estimate of the time that would be needed for convergence with splines. Note, that we assumed that SDA with or without splines in the end converged to the same change of measure, since the final iterations at $10^6$ replications would be done without splines in either case. The last column in Tables 3.9–3.12 shows approximation of $VRR$ with the spline technique ($VRR_{spl}$). It is clear that using splines would improve the performance of SDA, but not dramatically. Only in two (from 12) cases, namely, level 50 for for both examples of 3-node tandem networks, it made SDA more efficient than SDHI. For all other cases SDHI is still more efficient than SDA.

**Remark 3.5.2.** ***Employing quadratic convergence property of SDA***
It is important to note, however, that SDA has the quadratic convergence property (see discussion in Section 3.4.2, Remark 3.4.2). Because of it the $RTV$ of SDA decreases with increasing number of replications $n$. Thus, changing the number of replications after some iterations would allow to employ this property of SDA and improve its performance, which could lead to decreasing of the efficiency gain of SDHI over SDA. We tried this approach for the case of 4-node tandem networks, namely, we let SDA converge with $10^5$ replications, then did 2 iterations with $4 \cdot 10^5$, 2 iterations with $1.6 \cdot 10^6$ and 2 iterations with $6.4 \cdot 10^6$ replications. As one can see from Tables 3.15–3.16 it significantly improved the performance of SDA for the BRE point (Table 3.15) and did it a bit for NAE point (Table 3.16), except for level 50. Note, however, that the amount of time spent to achieve this improvement is very unpractical, i.e., hours comparing with seconds/minutes for the SDHI algorithm. Thus, in practice, SDHI is more suitable, though, theoretically, SDA will always win if it is given a lot of time for convergence and for play with number of replications.

| N | Numerical | PW | SDA | | SDHI | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $VRR_{spl}$ |
| 25 | 1.6408e-025 | 1.6406e-025 ± 0.04 | 3 | 1.64081e-025 ± 8e-3 | 0 | 1.6398e-025 ± 0.03 | 3.96 | 2.38 |
| 50 | 1.5144e-051 | 1.5138e-051 ± 0.04 | 3 | 1.51429e-051 ± 6e-3 | 0 | 1.5146e-051 ± 0.03 | 1.07 | 0.66* |
| 100 | * | 1.2907e-103 ± 0.04 | 3 | 1.28994e-103 ± 4e-3 | 0 | 1.2904e-103 ± 0.03 | 0.29 | 0.19 |

Table 3.9: 3-node tandem network - BRE region ($\lambda = 0.01,\ \mu_1 = 0.44,\ \mu_2 = 0.44,\ \mu_3 = 0.11$)

| N | Numerical | PW | SDA | | SDHI | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $VRR_{spl}$ |
| 25 | 5.9531e-020 | 4.8539e-020 ± 12.4 | 3 | 5.9522e-020 ± 0.07 | 4 | 5.8953e-020 ± 0.40 | 3.28 | 1.97 |
| 50 | 6.2176e-042 | 1.2264e-042 ± 8.93 | 3 | 6.2179e-042 ± 0.12 | 4 | 6.2258e-042 ± 0.34 | 3.17 | 0.88* |
| 100 | * | 1.1815e-087 ± 19.9 | 4 | 1.7308e-086 ± 0.09 | 5 | 1.7260e-086 ± 0.44 | 1.71 | 1.13 |

Table 3.10: 3-node tandem network - NAE region ($\lambda = 0.04,\ \mu_1 = 0.32,\ \mu_2 = 0.32,\ \mu_3 = 0.32$)

| N | Numerical | PW | SDA | | SDHI | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $VRR_{spl}$ |
| 25 | 2.7838e-017 | 2.7844e-17 ± 0.05 | 3 | 2.7842e-17 ± 8e-3 | 0 | 2.7862e-17 ± 0.04 | 4.08 | 2.61 |
| 50 | * | 9.3366e-35 ± 0.05 | 3 | 9.3415e-35 ± 0.01 | 0 | 9.3343e-35 ± 0.04 | 6.84 | 4.62 |
| 100 | * | 1.0523e-69 ± 0.05 | 3 | 1.0517e-69 ± 2e-3 (!) | 0 | 1.0508e-69 ± 0.04 | 0.15 (!) | 0.13 |

Table 3.11: 4-node tandem network - BRE region ($\lambda = 0.01,\ \mu_1 = 0.5,\ \mu_2 = 0.3,\ \mu_3 = 0.14,\ \mu_4 = 0.05$)

| N | Numerical | PW | SDA | | SDHI | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $VRR_{spl}$ |
| 25 | 5.0207e-016 | 4.1762e-016 ± 11.6 | 3 | 5.0277e-016 ± 0.14 | 3 | 5.0506e-016 ± 1.38 | 2.43 | 1.68 |
| 50 | * | 3.1024e-035 ± 37.0 | 3 | 1.3532e-034 ± 0.26 | 4 | 1.3403e-034 ± 0.88 | 5.31 | 2.05 |
| 100 | * | 6.7039e-074 ± 62.0 | 3 | 1.2775e-072 ± 0.86 | 5 | 1.3037e-072 ± 1.50 | 17.2 | 12.8 |

Table 3.12: 4-node tandem network - NAE region ($\lambda = 0.04,\ \mu_1 = 0.24,\ \mu_2 = 0.24,\ \mu_3 = 0.24,\ \mu_4 = 0.24$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDHI** | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $VRR_{spl}$ |
| 25 | 2.7838e-017 | 2.7844e-17 ± 0.05 | 3 | 2.7842e-17 ± 8e-3 | 0 | 2.7862e-17 ± 0.04 | 4.08 | 2.61 |
| 50 | * | 9.3366e-35 ± 0.05 | 3 | 9.3410e-35 ± 0.01 | 0 | 9.3343e-35 ± 0.04 | 0.28 | 0.25 |
| 100 | * | 1.0523e-69 ± 0.05 | 3 | 1.0517e-69 ± 2e-3 | 0 | 1.0508e-69 ± 0.04 | 0.15 | 0.13 |

Table 3.13: 4-node tandem network - BRE region ($\lambda = 0.01$, $\mu_1 = 0.5$, $\mu_2 = 0.3$, $\mu_3 = 0.14$, $\mu_4 = 0.05$) with SDA$_{25}$ for N=50, 100 and $5 \cdot 10^5 \rightarrow 10^6$ replications

| **N** | **Numerical** | **PW** | **SDA** | | **SDHI** | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ | $VRR_{spl}$ |
| 25 | 5.0207e-016 | 4.1762e-016 ± 11.6 | 3 | 5.0277e-016 ± 0.14 | 3 | 5.0506e-016 ± 1.38 | 2.43 | 1.68 |
| 50 | * | 3.1024e-035 ± 37.0 | 3 | 1.3472e-034 ± 0.15 | 4 | 1.3403e-034 ± 0.88 | 6.15 | 5.72 |
| 100 | * | 6.7039e-074 ± 62.0 | 3 | 1.2933e-072 ± 0.31 | 5 | 1.3037e-072 ± 1.50 | 5.22 | 5.07 |

Table 3.14: 4-node tandem network - NAE region ($\lambda = 0.04$, $\mu_1 = 0.24$, $\mu_2 = 0.24$, $\mu_3 = 0.24$, $\mu_4 = 0.24$) with SDA$_{25}$ for N=50, 100 and $5 \cdot 10^5 \rightarrow 10^6$ replications

| **N** | **Numerical** | **PW** | **SDA** | | | **SDHI** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | time | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | time | $VRR$ |
| 25 | 2.7838e-017 | 2.7844e-17 ± 0.05 | 3 | 2.78377e-17 ± 4e-4 | ≈ 9h | 0 | 2.7862e-17 ± 0.04 | 40 sec | 0.07 |
| 50 | * | 9.3366e-35 ± 0.05 | 3 | 9.34035e-35 ± 4e-4 | ≈ 9h | 0 | 9.3343e-35 ± 0.04 | 90 sec | 0.09 |
| 100 | * | 1.0523e-69 ± 0.05 | 3 | 1.05161e-69 ± 5e-4 | ≈ 9h | 0 | 1.0508e-69 ± 0.04 | 3 min | 0.05 |

Table 3.15: 4-node tandem network - BRE region ($\lambda = 0.01,\ \mu_1 = 0.5,\ \mu_2 = 0.3,\ \mu_3 = 0.14,\ \mu_4 = 0.05$) with $10^5 \rightarrow 4 \cdot 10^5 \rightarrow 1.6 \cdot 10^6 \rightarrow 6.4 \cdot 10^6$ replications

| **N** | **Numerical** | **PW** | **SDA** | | | **SDHI** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | time | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | time | $VRR$ |
| 25 | 5.0207e-016 | 4.1762e-016 ± 11.6 | 3 | 5.0195e-16 ± 0.04 | ≈ 4h | 3 | 5.0506e-016 ± 1.38 | 20 sec | 0.82 |
| 50 | * | 3.1024e-035 ± 37.0 | 3 | 1.3502e-34 ± 0.05 | ≈ 5h | 4 | 1.3403e-034 ± 0.88 | 1 min | 1.31 |
| 100 | * | 6.7039e-074 ± 62.0 | 3 | 1.3050e-72 ± 0.10 | ≈ 6.5h | 5 | 1.3037e-072 ± 1.50 | 2 min | 0.94 |

Table 3.16: 4-node tandem network - NAE region ($\lambda = 0.04,\ \mu_1 = 0.24,\ \mu_2 = 0.24,\ \mu_3 = 0.24,\ \mu_4 = 0.24$) with $10^5 \rightarrow 4 \cdot 10^5 \rightarrow 1.6 \cdot 10^6 \rightarrow 6.4 \cdot 10^6$ replications

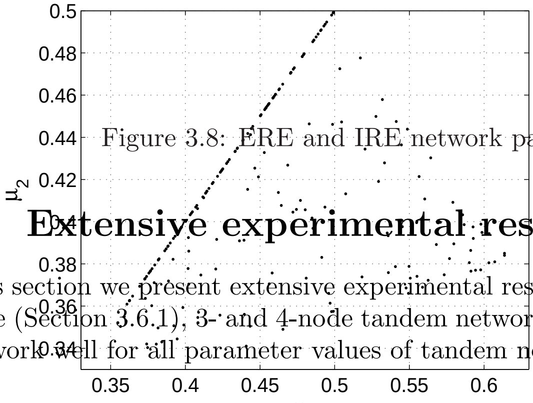


Figure 3.8: ERE and IRE network parameters checked

## 3.6 Extensive experimental results

In this section we present extensive experimental results to validate the heuristics for 2-node (Section 3.6.1), 3- and 4-node tandem networks (Section 3.6.2), i.e., show that they work well for all parameter values of tandem networks.

### 3.6.1 Validation for 2 queues in tandem

In this section we will start with demonstrating that SDHI outperforms SDH for all network parameters (except the case $\mu_1 = \mu_2$ when they are equivalent). After that we will show that SDHI indeed yields asymptotically efficient estimates with bounded or less than linearly growing relative error in the regions where no state-independent heuristic is known to be efficient. We will also present an algorithm to find the optimal value of $b$ and give some practical recommendations to get quick results.

An extensive set of experiments was made to cover the network parameters in the lower ($\mu_1 \geq \mu_2$) ERE and IRE regions in Figure 3.7 (as mentioned in Remark 3.5.1 (see also [45]) the queue order is interchangeable, so it is sufficient to consider only networks with the last queue as the bottleneck, i.e., with $\mu_1 \geq \mu_2$); we also checked some points in the BRE region to show that SDHs are generally applicable.

Experimental results were gathered applying the SDH and SDHI changes of measure for three different values of overflow level $N$, namely 25, 50 and 100.

Different levels were considered to check whether a relative error is bounded, i.e., is the same for all levels. For each level several simulation runs were made: at $10^5$, $4 \cdot 10^5$, $16 \cdot 10^5$ and $64 \cdot 10^5$ cycles to check the stability of the estimator. For a stable estimator if we increase the number of simulation cycles by, say, a factor of 4, the relative error should decrease by the factor of 2 ($\sqrt{4}$), see Equation (3.29) and discussion right after it.

In total there were around 300 randomly chosen (with uniform distribution) points checked: around 250 from the lower ERE and IRE regions (Figure 3.8), half of which was with $\mu_1 = \mu_2$ (in this case SDH $\equiv$ SDHI) and 50 points from the BRE region (not shown in figures).

Several observations that were made based on the obtained experimental results are discussed below (Propositions 1–5).

### Performance of SDHI in comparison with SDH

**Proposition 1.** *For all network parameters SDHI outperforms SDH.*

As a measure of comparison between two changes of measure we will use the variance reduction ratio (see Equation (3.30)). Note, however, that since SDHI and SDH require approximately the same amount of simulation time the value $VRR$ is just a ratio of the squared $RE$ values. $VRR > 1$ means that SDHI works better, otherwise, SDH works better; $VRR = 1$ means that they both work equivalently well. It is clear from Figures 3.9a–b that for all network parameters the value $VRR \geq 1$, which confirms the claim.

All the observations discussed below concern SDHI.

### Dependency of parameter $b_{opt}$ on the overflow level. Optimal value of parameter $b$.

**Proposition 2.** *The optimal parameter* $b$ *is a non-decreasing function of the overflow level* $N$*, i.e.,* $b_{opt}(N_1) \leq b_{opt}(N_2)$ *for* $N_1 \leq N_2$.

In the experiments we checked a broader set of values for parameter $b$ and none of the cases violated Proposition 2.

Since parameter $b$ in SDHI is not known a priori this observation allows us to restrict possible choices of $b$ for higher levels once for some lower level the optimal $b$ was found. Thus, one can use Algorithm 2 (below) for finding $b_{opt}(N_i)$ for a set of ordered levels $N_i$ ($N_i < N_{i+1}$) with $i = 1, ..., m$ (we considered the case $m = 3$ and $N_1 = 25$, $N_2 = 50$, $N_3 = 100$).

**Algorithm 2** Algorithm for choosing the optimal parameter $b$

```
$i := 0$
$b_{opt}(N_0) := 0$
$RTV_{-1} := \infty$
if $i < m$ then
    $b := b_{opt}(N_i) - 1$
    $i := i + 1$
    repeat
        $b := b + 1$
        run the simulation for level $N_i$ and parameter $b$ at $n$ cycles
        (for $n = 10^5$, $4 \cdot 10^5$, $16 \cdot 10^5$, $64 \cdot 10^5$)
    until $RTV$ gets stable and $RTV_b > RTV_{b-1}$
    $b_{opt}(N_i) := b$
end if
```

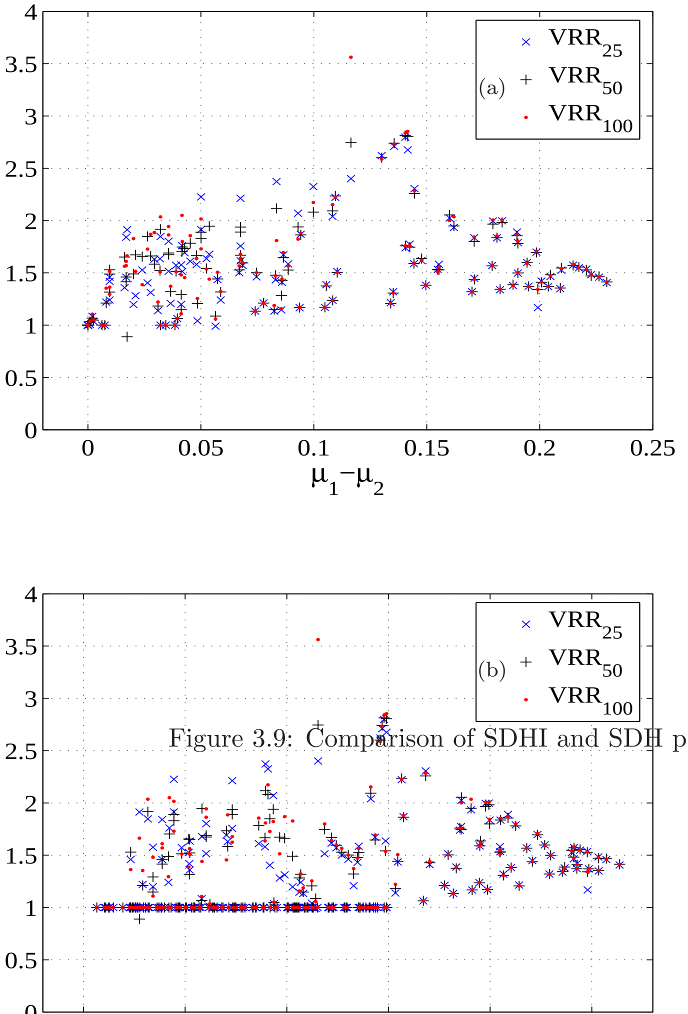


Figure 3.9: Comparison of SDHI and SDH performance

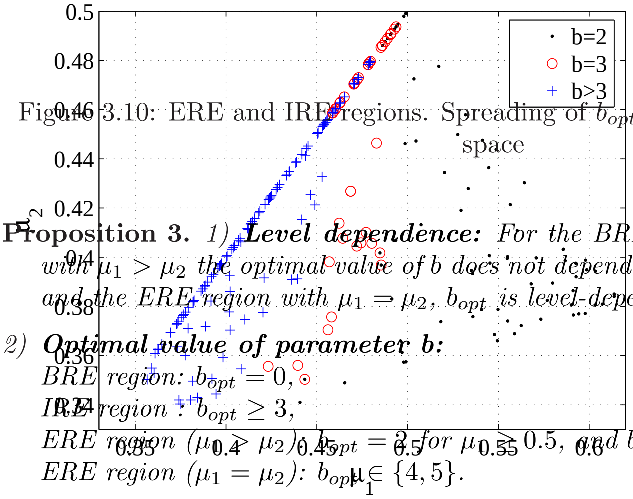


Figure 3.10: ERE and IRE regions. Spreading of $b_{opt}$ over the network parameter space

**Proposition 3.** *1)* ***Level dependence:*** *For the BRE region and the ERE region with $\mu_1 > \mu_2$ the optimal value of $b$ does not depend on level $N$; for the IRE region and the ERE region with $\mu_1 = \mu_2$, $b_{opt}$ is level-dependent;*

*2)* ***Optimal value of parameter b:***
*BRE region: $b_{opt} = 0$,*
*IRE region : $b_{opt} \geq 3$,*
*ERE region ($\mu_1 > \mu_2$): $b_{opt} = 2$ for $\mu_1 \geq 0.5$, and $b_{opt} \in \{2, 3, 4\}$, otherwise,*
*ERE region ($\mu_1 = \mu_2$): $b_{opt} \in \{4, 5\}$.*

To show that $b_{opt}$ depends (or does not depend) on level $N$ consider the difference between values of $b_{opt}$ for consecutively considered levels, i.e., $b_{opt}(100) - b_{opt}(50)$ and $b_{opt}(50) - b_{opt}(25)$. As one can see from Figure 3.11a, these differences are equal to zero for $\mu_1 - \mu_2 > 0$, which means that $b_{opt}$ is the same for all levels (BRE region is not shown since the differences are always zeros). For $\mu_1 - \mu_2 = 0$ they are non-zero (equal to one), i.e., $b_{opt}$ changes with level $N$. For the IRE region (Figures 3.12a-b) the differences are greater or equal to zero, and, thus, $b_{opt}$ is level-dependent.

To see how the values of $b_{opt}$ depend on the network parameters let us look at Figures 3.13–3.14, where $b_{opt}$ is shown as a function of either $\mu_1 - \mu_2$ or $\mu_1$. As one can see (Figure 3.13b) for the ERE region $2 \leq b_{opt} \leq 5$, with $b_{opt} = 2$ for $\mu_1 \geq 0.5$. From Figure 3.13a, one can see that $2 \leq b_{opt} \leq 4$ for $\mu_1 - \mu_2 > 0$ and $4 \leq b_{opt} \leq 5$ for $\mu_1 - \mu_2 = 0$. For IRE region $b_{opt}$ is such that $b_{opt} \geq 3$ as can be clearly seen from Figures 3.14a–b. Figure 3.10 shows how values of $b_{opt}$ are spread over the parameter space ($\mu_1$, $\mu_2$).

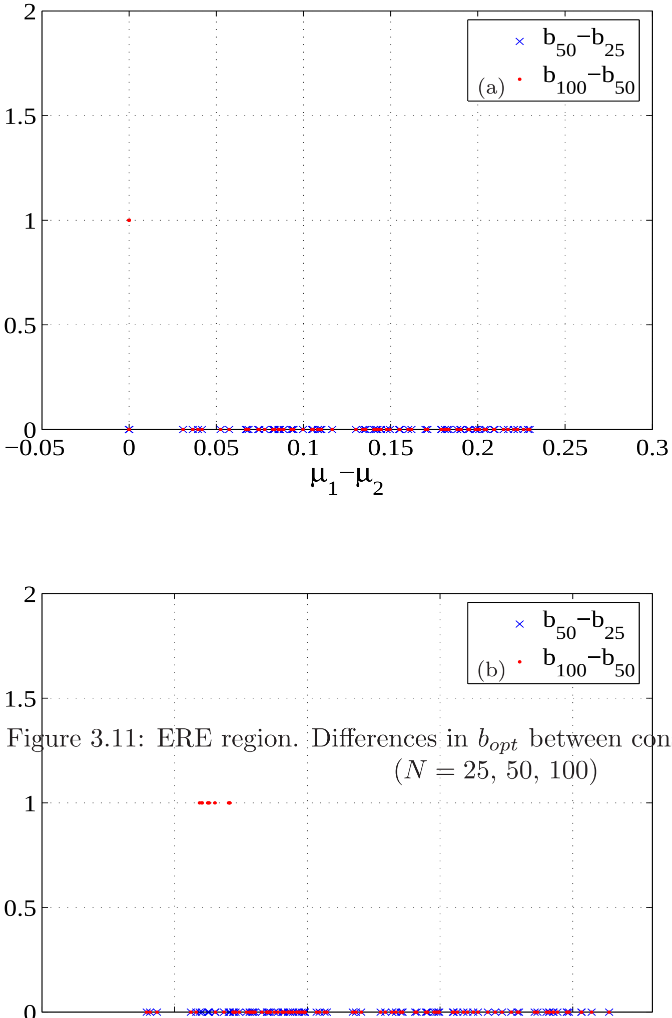


Figure 3.11: ERE region. Differences in $b_{opt}$ between consecutively considered levels ($N$ = 25, 50, 100)

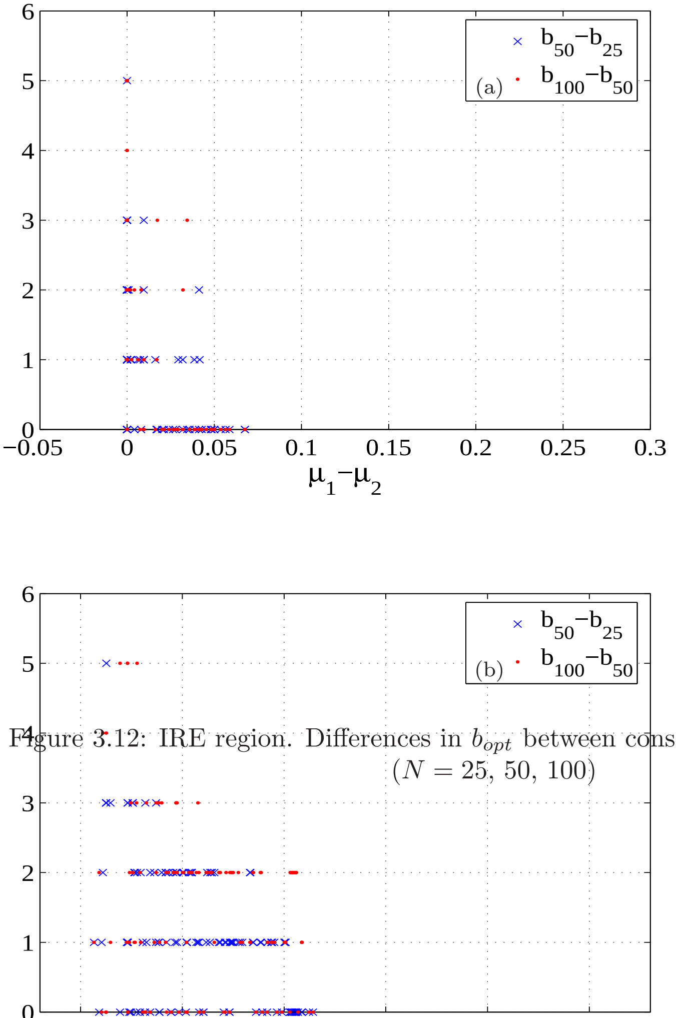


Figure 3.12: IRE region. Differences in $b_{opt}$ between consecutively considered levels ($N = 25, 50, 100$)

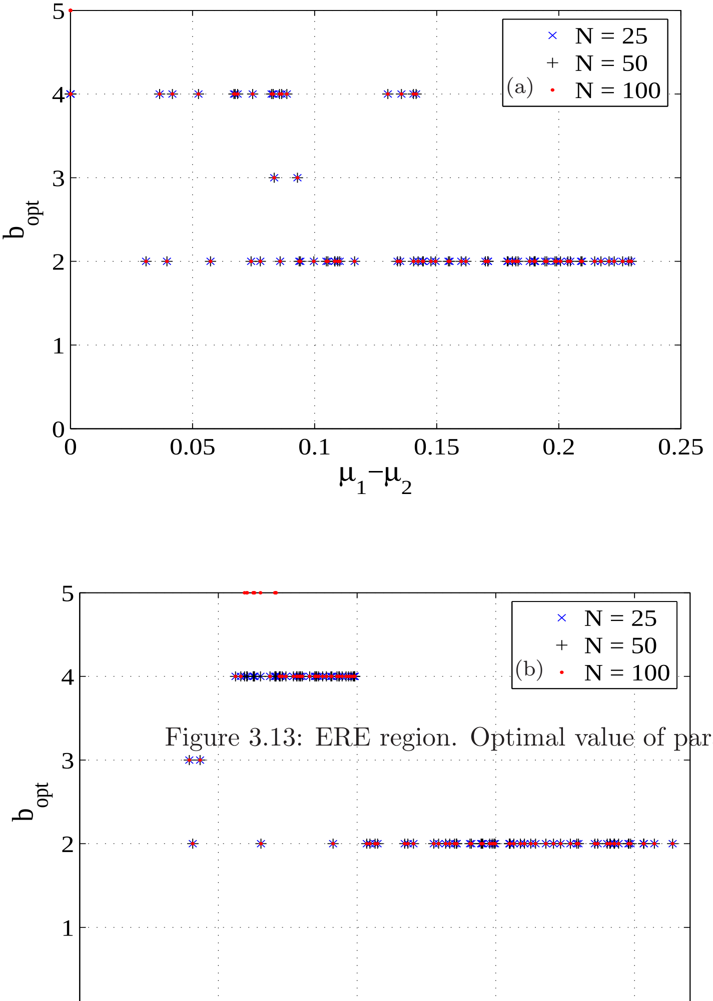


Figure 3.13: ERE region. Optimal value of parameter $b$ ($b_{opt}$)

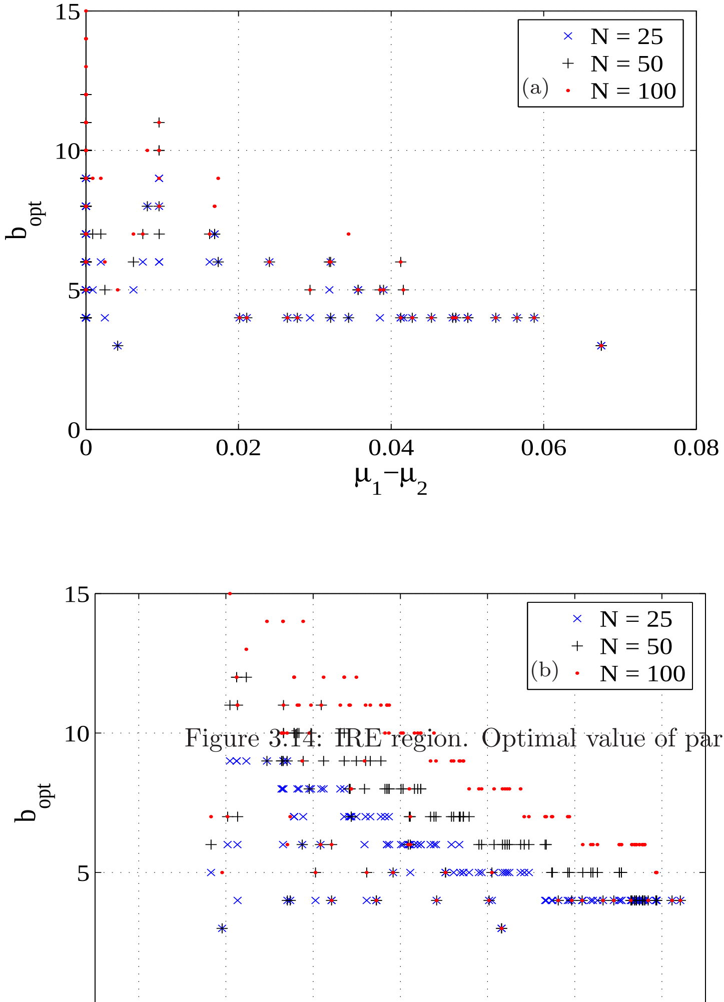


Figure 3.14: IRE region. Optimal value of parameter $b$ ($b_{opt}$)

Propositions 2–3 in practice mean the following: to simulate a 2-node tandem network, one needs, first, to determine to which region (BRE, ERE or IRE, see Figure 3.7) the chosen network belongs; then, simulate the network with new probability distribution SDHI($b$) with parameter $b$ chosen according to Proposition 3.

### Behavior of relative error

**Proposition 4.** *For all network parameters (BRE, ERE and $IRE$ regions) SDHI gives estimates with bounded relative error if $\mu_1 > \mu_2$, and estimates with less than linearly growing relative error if $\mu_1 \approx \mu_2$.*

To show that $RE$ is bounded we calculated the ratios between consecutively considered levels, i.e., $RE_{100}/RE_{50}$ and $RE_{50}/RE_{25}$ (referred in a sequel as $RER$s, which stands for $RE$ ratios). For a bounded $RE$ those ratios should be near 1, i.e., $RE$ does not change when the level is doubled; for a linearly growing $RE$ the ratios should be near 2, i.e., $RE$ grows proportionally to the level growth; for a quadratically growing $RE$ they would be near 4, i.e., $RE$ grows quadratically with the level change. As one can see from Figures 3.15–3.17 RERs $\approx 1$ for $\mu_1 - \mu_2 > 0.05$ and $RER$s $< 2$ for all $\mu_1 - \mu_2 \approx 0$, i.e., SDHI gives bounded $RE$ for $\mu_1 - \mu_2 > 0$, and less than linear growth when $\mu_1 - \mu_2 \approx 0$.

Thus, Proposition 4 tells us that if the value of $b$ for simulating the new network is chosen as $b = b_{opt}$, the relative error of the resulting estimate will not increase (or, will increase at most linearly) when the rare event decays exponentially fast.

### Sensitivity of the heuristic (SDHI)

Now the natural question arises whether $b_{opt}$ is unique and what would happen if we used an “almost optimal” $b$. In other words, *is the value of $b$ crucial for the heuristic to work?* Or, how sensitive is the heuristic for changing parameter $b$? According to experimental results (not proven theoretically) when the number of simulation cycles is large enough, $b_{opt}$ is uniquely determined ($b_{opt}$ corresponds to the minimal $RTV$). To demonstrate this, look at the Figures 3.18a–b, where (for some point) $RTV$ is shown as a function of $b$ for different number of cycles. It is clear that $b_{opt} = 10$ for $N = 50$ and $b_{opt} = 15$ for $N = 100$ since $RTV \approx const$ and takes the minimum value. Note also that $RTV$ is less unstable for $b < b_{opt}$ than for $b > b_{opt}$.

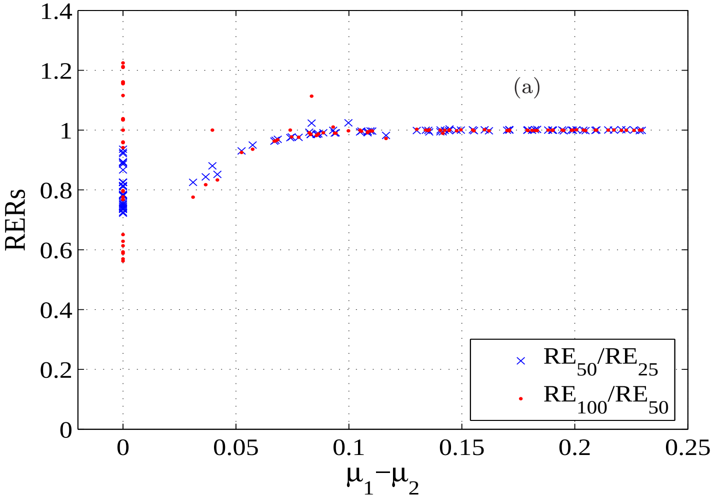
Now, what would happen if the value of $b$ is not the optimal, but is nearly optimal? How crucially will it affect the performance of the heuristics? Proposition 5 answers the question.

**Proposition 5.** *If parameter $b(N)$ for $N = 50, 100$ is chosen as $b(N) = b_{opt}(25)$ then for the BRE region and the ERE region with $\mu_1 > \mu_2$ SDHI gives bounded*[3] *RE;*
*for the ERE region with $\mu_1 \approx \mu_2$ RE grows less than linearly;*
*for the IRE region RE grows less than linearly if $\mu_1 > \mu_2$ and less than quadratically if $\mu_1 \approx \mu_2$.*

Indeed, as one can see from Figures 3.19–3.21, $RER$s $\leq 2$ for ERE region with $\mu_1 \approx \mu_2$ (Figure 3.19a) and for IRE region for $\mu_1 > \mu_2$ (Figures 3.20a); for IRE region with $\mu_1 \approx \mu_2$ $RER$s$\leq 4$, which supports the claim.

---

[3] According to Proposition 3, $b_{opt}$ does not depend on the level, i.e., $b(N) = b_{opt}(25)$ for all $N > 0$;



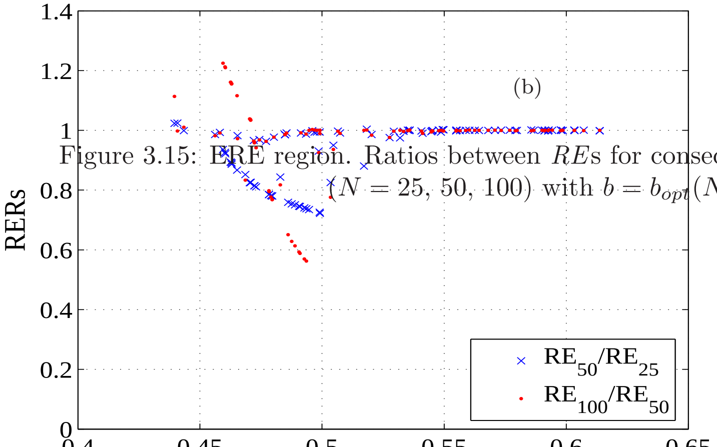


Figure 3.15: ERE region. Ratios between $RE$s for consecutively considered levels ($N = 25, 50, 100$) with $b = b_{opt}(N)$

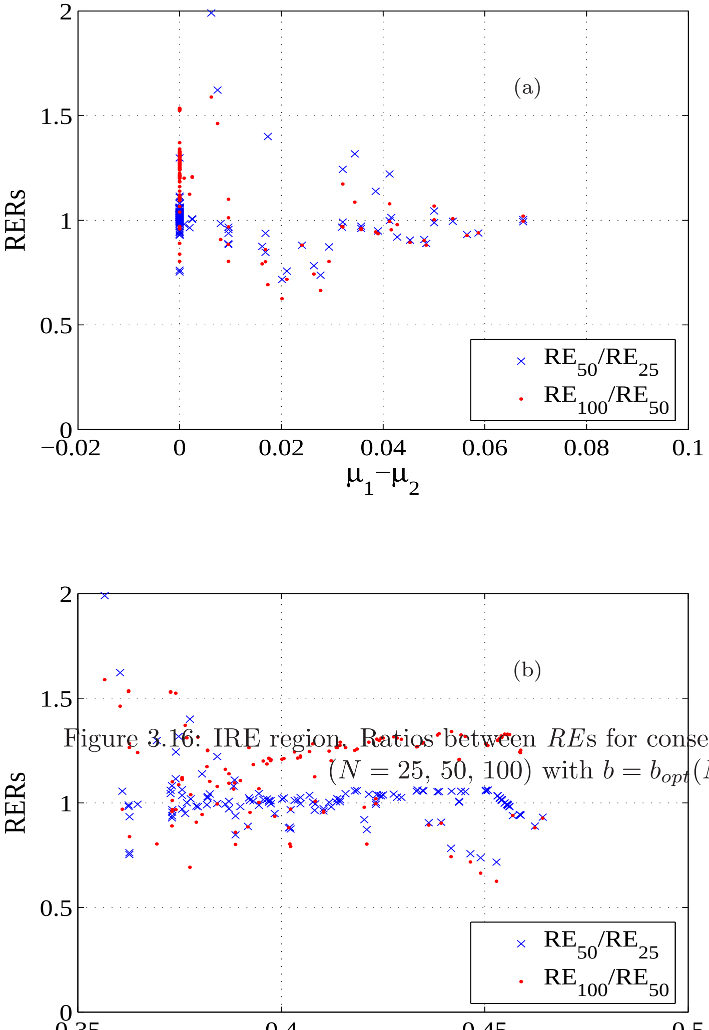


Figure 3.16: IRE region. Ratios between $RE$s for consecutively considered levels ($N$ = 25, 50, 100) with $b = b_{opt}(N)$

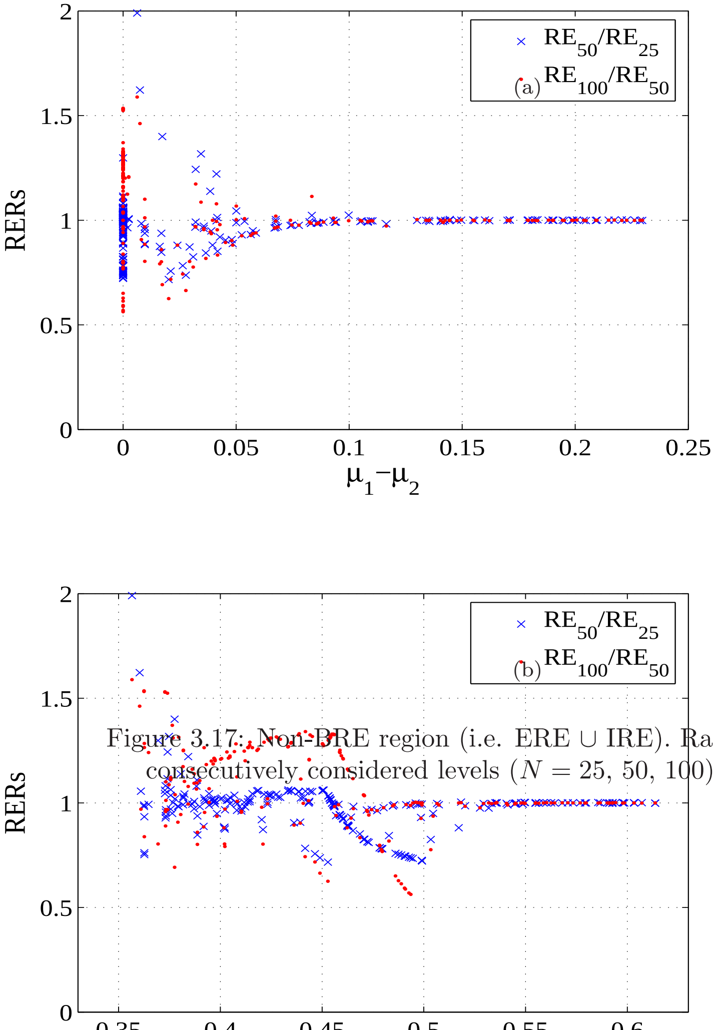
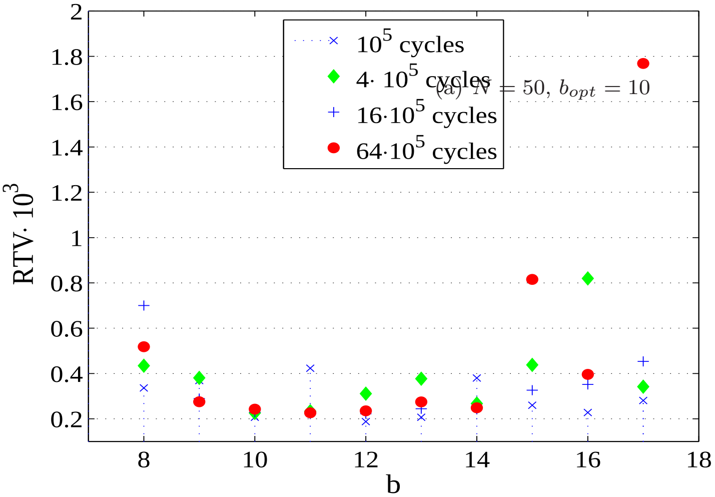
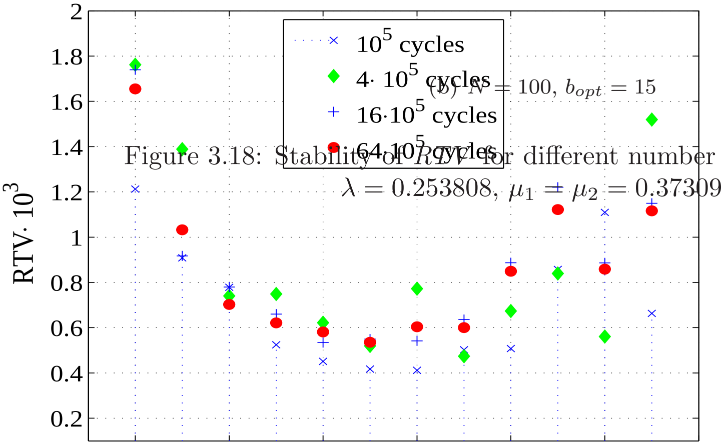


Figure 3.17: Non-BRE region (i.e. ERE $\cup$ IRE). Ratios between *RE*s for consecutively considered levels ($N = 25,\ 50,\ 100$) with $b = b_{opt}(N)$



(a) $N = 50$, $b_{opt} = 10$



(b) $N = 100$, $b_{opt} = 15$

Figure 3.18: Stability of $RTV$ for different number of simulation cycles $\lambda = 0.253808$, $\mu_1 = \mu_2 = 0.373096$

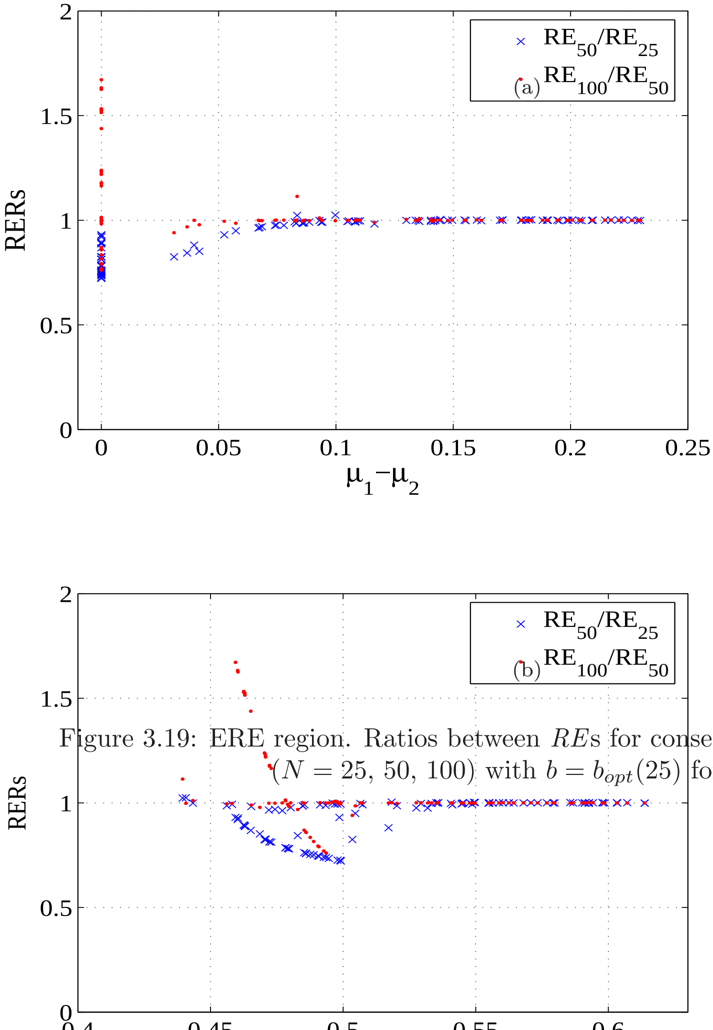
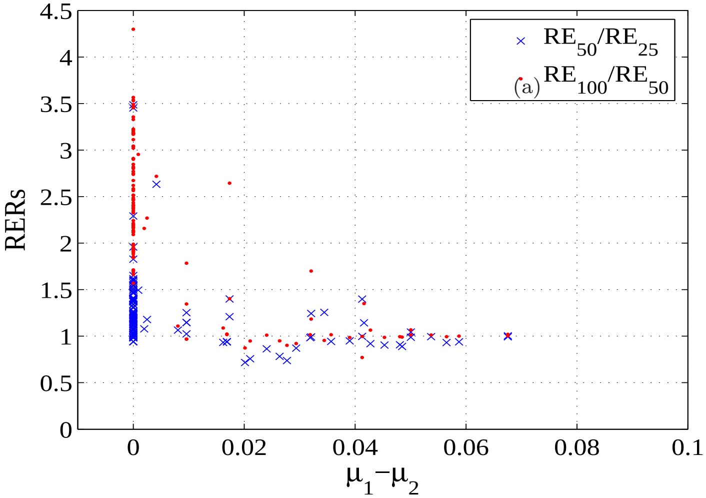
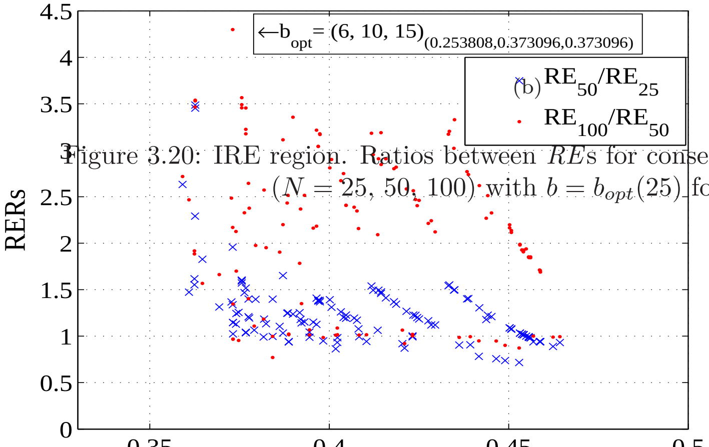


Figure 3.19: ERE region. Ratios between $RE$s for consecutively considered levels ($N = 25$, 50, 100) with $b = b_{opt}(25)$ for all $N$



(a)



(b)

Figure 3.20: IRE region. Ratios between $RE$s for consecutively considered levels ($N$ = 25, 50, 100) with $b = b_{opt}(25)$ for all $N$

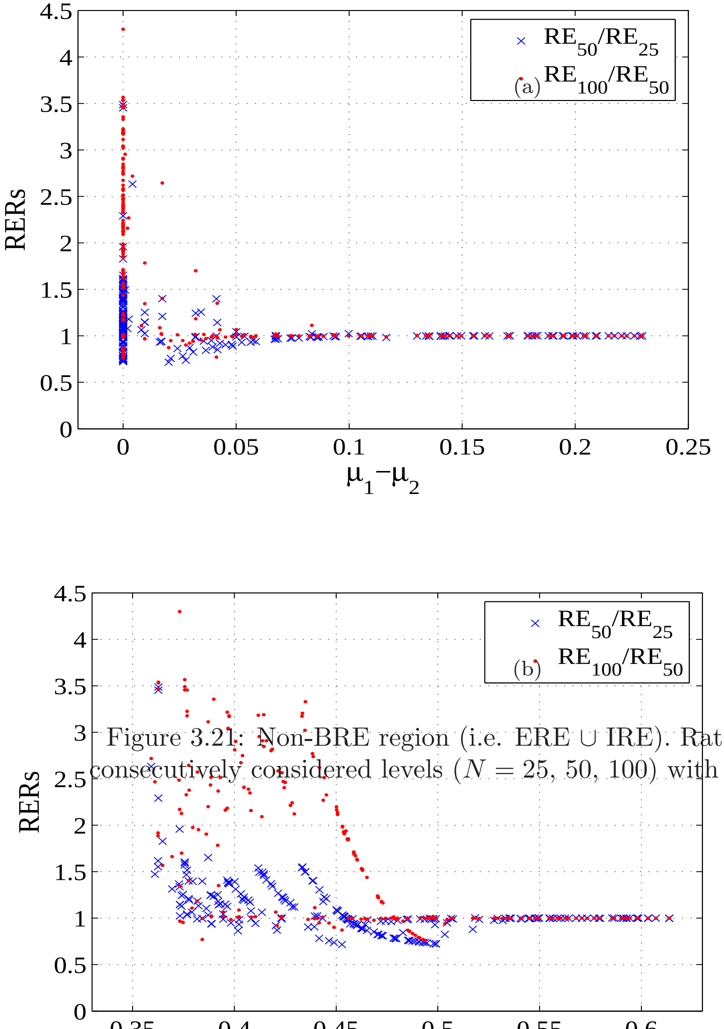


Figure 3.21: Non-BRE region (i.e. ERE $\cup$ IRE). Ratios between $RE$s for consecutively considered levels ($N = 25, 50, 100$) with $b = b_{opt}(25)$ for all $N$

Thus, if one is interested in quick results rather than the best ones, the parameter $b$ for simulation can be chosen as $b_{opt}(25)$ (computer time spent on finding $b_{opt}(25)$ is negligible) and use the same parameter $b$ for finding probabilities for higher levels $N$. Proposition 5 guarantees in this case that the relative error will still grow less than linearly (or, quadratically, depending on the parameter region) when the probability of interest decays exponentially fast.

### 3.6.2 Validation of SDHI for 3 and 4 queues in tandem

This section is designed to demonstrate that SDHI, the change of measure proposed in Section 3.3.2, yields estimates with bounded or less than linearly growing relative error also in cases of 3 and 4 queues in tandem.

A set of experiments was made for different (and known to be more difficult) network parameters (remember that we consider only the case of ordered queues):

1. 3 queues in tandem (around 80 points for each case):
   a) $\mu_1 = \mu_2 = \mu_3$,
   b) $\mu_1 = \mu_2 > \mu_3$,
   c) $\mu_1 > \mu_2 = \mu_3$.
2. 4 queues in tandem (around 60 points for each case):
   a) $\mu_1 = \mu_2 = \mu_3 = \mu_4$,
   b) $\mu_1 = \mu_2 = \mu_3 > \mu_4$,
   c) $\mu_1 = \mu_2 > \mu_3 = \mu_4$,
   d) $\mu_1 > \mu_2 = \mu_3 = \mu_4$.

For each set of parameters the simulation was run for three overflow levels $N$ ($N = 25$, 50 and 100) at $10^5$, $4 \cdot 10^5$, $16 \cdot 10^5$ and $64 \cdot 10^5$ cycles. Observations that were made based on the experimental results are discussed below.

#### Dependency of parameter $b_{opt}$ on the overflow level

First, let us give some remarks about the algorithm, proposed in Section 3.6.1 and based on Proposition 2. For 2 queues in tandem it helped to restrict the amount of candidates on $b_{opt}$ for higher levels once $b_{opt}$ was found for some lower level. Unfortunately, in some cases of 3 and 4 queues in tandem, Proposition 2 was violated, thus, for each level $b_{opt}$ had to be searched separately by consecutively trying all possibilities starting, say, from $b = 1$. The counterpart of Proposition 3 for tandem networks of 3 and 4 queues is stated below.

**Proposition 6.** *For all network parameters of 3- and 4-node tandem networks the optimal value of parameter $b$ ($b_{opt}$) depends on the overflow level $N$.*

---

according to Proposition 4 SDHI gives bounded $RE$ for $b_{opt}$.

That can be easily seen from Figures 3.22–3.23 where the differences in $b_{opt}$ between consecutively considered levels ($N = 25$, 50 and 100) are represented, i.e., $b_{opt}(100) - b_{opt}(50)$ and $b_{opt}(50) - b_{opt}(25)$. The fact that there are network parameters for which these differences are non-zero confirms the statement.

### Behavior of relative error

**Proposition 7.** *1)* ***For a 3-node tandem network*** *SDHI gives estimates with bounded relative error for network parameters satisfying one of the following condition: $\mu_2 > \mu_3$, or $\mu_1 > 0.4$, or $\mu_1 - \mu_2 > 0.18$,*
*and less than linearly growing relative error, otherwise.*

*2)* ***For a 4-node tandem network*** *SDHI gives estimates with less than linearly growing relative error.*

To show this, we again considered relative error ratios (*RER*s) between consecutively considered levels, i.e., $RE_{100}/RE_{50}$ and $RE_{50}/RE_{25}$. For bounded relative error those ratios should be near 1, for linearly growing relative error they should be near 2. As one can see from Figures 3.24–3.25 (representing the results for a 3-node tandem network) for all $\mu_2 - \mu_3 > 0$ those ratios are indeed near 1 and only for $\mu_2 - \mu_3 = 0$ they are larger but still less than 2, which confirms the first part of Proposition 7. Figures 3.26–3.27 show that *RER*s for a 4-node network are mostly between 0.5 and 1.5 and the rest is less than 2, so the second part of Proposition 7 is also supported.

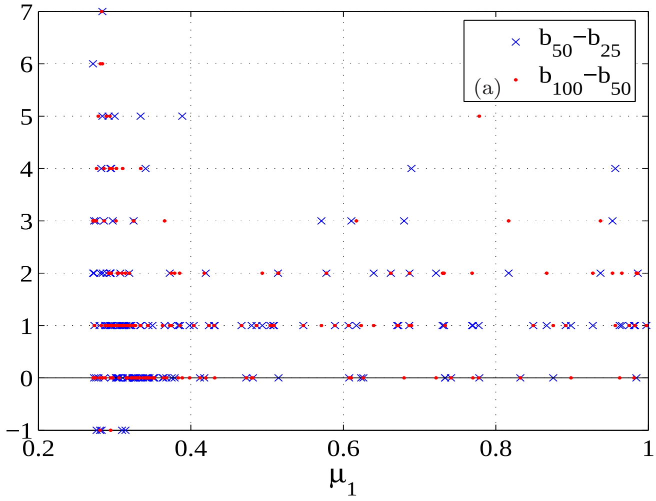
### Guideline for choosing $b$

Note, that for a 2-node tandem network we have only one parameter $b$ to choose ($b_2$). When the number of nodes in the network grows, we obtain more parameters, namely, two for a 3-node network ($b_2$ and $b_3$) and three for 4-node network ($b_2$, $b_3$ and $b_4$). In general, they can be different, which makes finding the optimal values of $b_i$ more complicated. However, as we show below, in most of the cases they can be chosen the same. The proposition below summarizes the guideline for choosing the optimal values of $b_i$.

**Proposition 8.** *1)* ***For a 3-node tandem network:***
*if $\mu_1 > 1.5 \cdot \mu_2$, then $b_2 = \infty$, $b_3 = b$ (to be found); otherwise $b_2 = b_3 = b$;*

*2)* ***For a 4-node tandem network:***
*if $\mu_1 > 1.5 \cdot \mu_2$ then $b_2 = \infty$, $b_3 = b_4 = b$;*
*if $\mu_2 > 1.5 \cdot \mu_3$ then $b_2 = b_3 = \infty$, $b_4 = b$;*
*otherwise, $b_2 = b_3 = b_4 = b$.*

For the heuristics SDHs the equality $b_i = \infty$ means that we do not “push” queue $i$. For 4-node tandem networks $b_2 = \infty$ and $b_3 = b_4 = b$ means that we treat the last 3 queues as a 3-node tandem network. We do that only in cases where the service rate at the first queue is large enough so that it does not influence the traffic stream much. Apparently, by experiments, $\mu_1 > 1.5 \cdot \mu_2$ was enough to satisfy this condition (for a 3-node tandem network it means that we treat the last two nodes as a 2-node tandem network). The same was true when $\mu_2 > 1.5 \cdot \mu_3$. In that case the first two queues did not influence the stream, thus we could treat 4-node tandem network as a 2-node network ($b_2 = b_3 = \infty$).



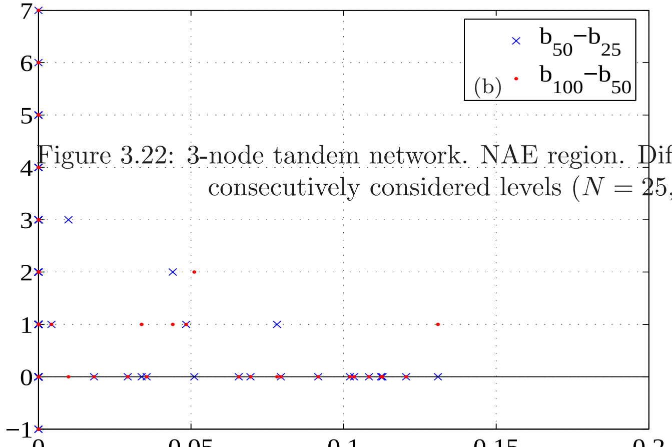


Figure 3.22: 3-node tandem network. NAE region. Differences in $b_{opt}$ between consecutively considered levels ($N = 25, 50, 100$)

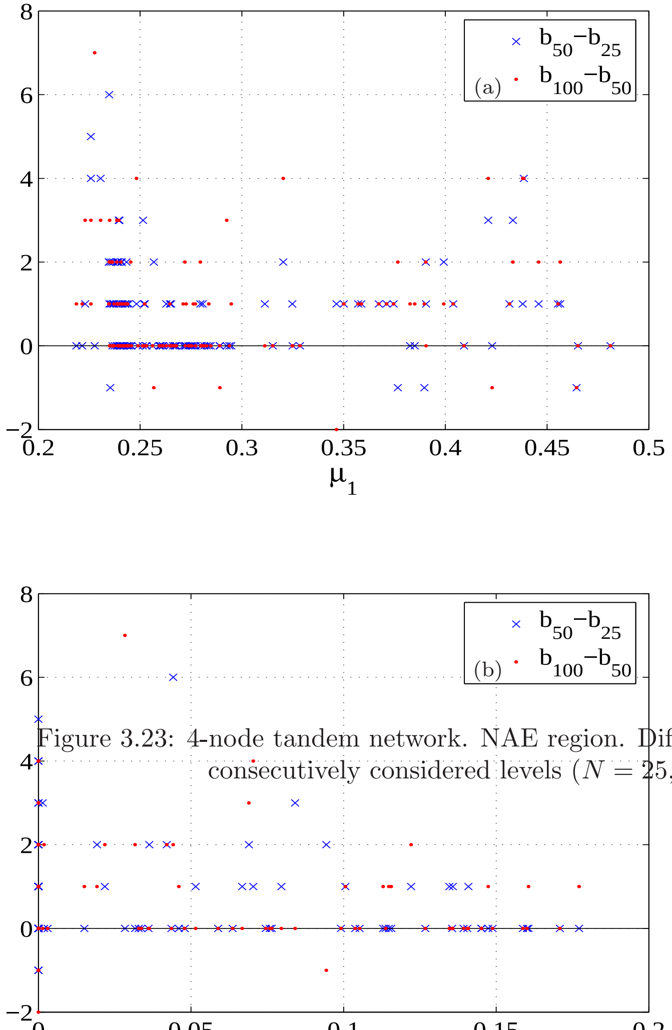
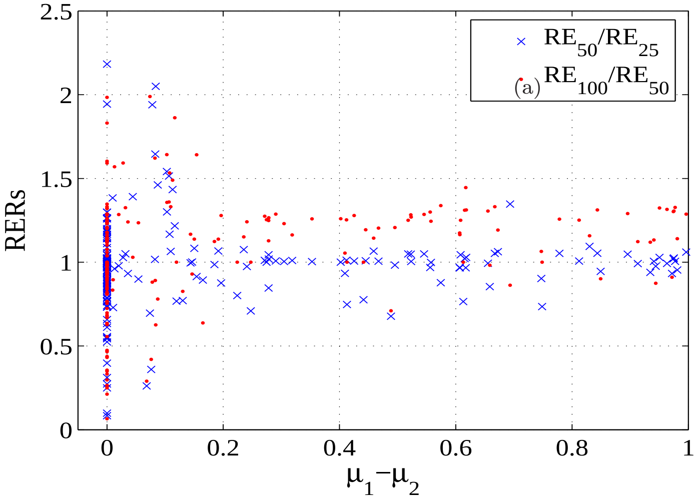
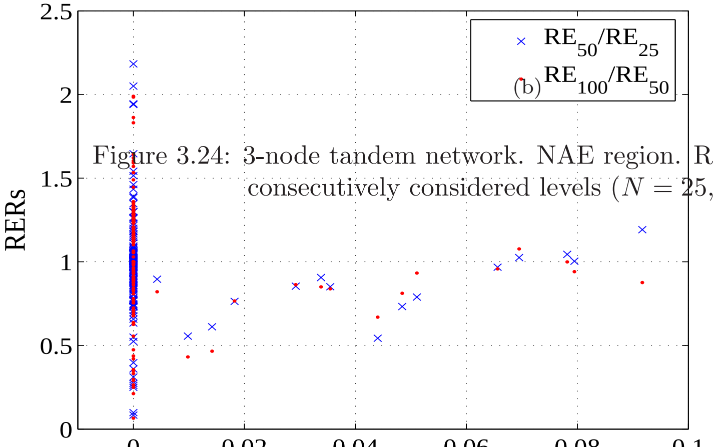


Figure 3.23: 4-node tandem network. NAE region. Differences in $b_{opt}$ between consecutively considered levels ($N$ = 25, 50, 100)

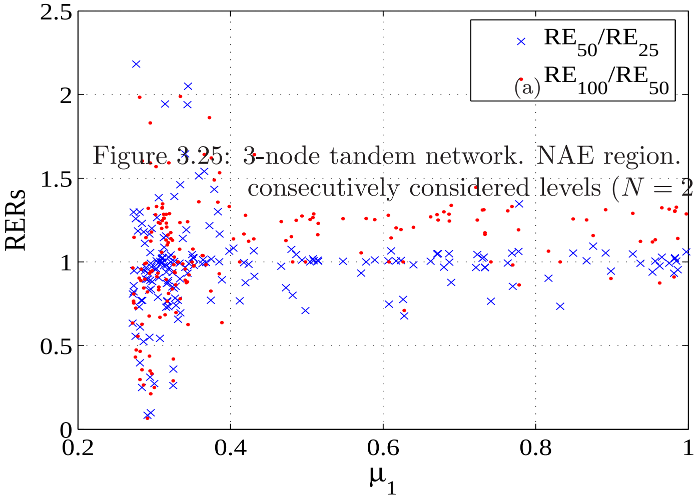


(a)



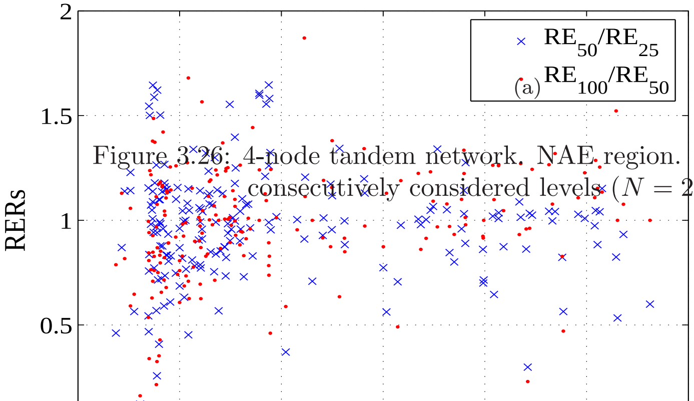
(b)

Figure 3.24: 3-node tandem network. NAE region. Ratios between *RE*s for consecutively considered levels ($N$ = 25, 50, 100)



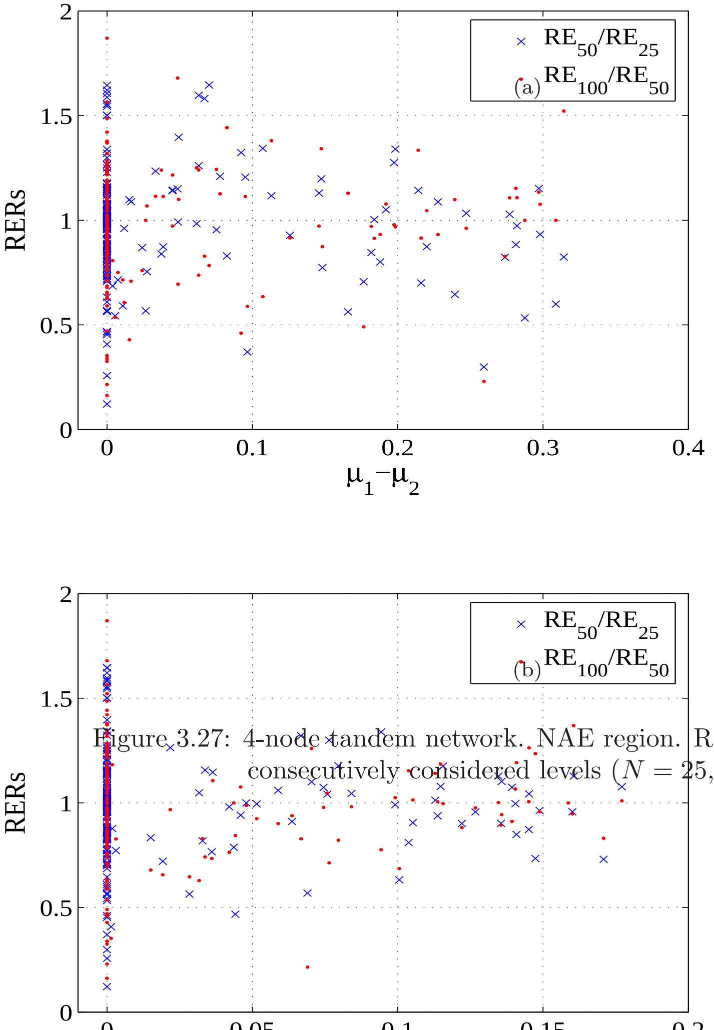
(a)

Figure 3.25: 3-node tandem network. NAE region. Ratios between *RE*s for consecutively considered levels ($N$ = 25, 50, 100)



(a)

Figure 3.26: 4-node tandem network. NAE region. Ratios between *RE*s for consecutively considered levels ($N$ = 25, 50, 100)



(a)

(b)

Figure 3.27: 4-node tandem network. NAE region. Ratios between *RE*s for consecutively considered levels ($N = 25, 50, 100$)

### Final remarks

Since for 3- and 4-node networks parameter $b$ always depends on the level, and since Proposition 2 does not have a counterpart for 3 and 4 queues in tandem we have not investigated sensitivity of SDHI on parameter $b$.

## 3.7 Conclusion

In this chapter we developed two state-dependent heuristics for estimating population overflow probabilities in tandem networks. Each of the heuristics is a set of changes of measure, parametrized by $b$, the number of boundary levels for which the change of measure depends on the system state. There exists an optimal value of $b$ ($b_{opt}$), i.e., the one which gives estimates with smallest $RTV$. Depending on the network parameters the heuristic parametrized by this value (denoted as SDHI($b_{opt}$)) gives estimates with bounded or less than linearly growing relative error. For 2 queues in tandem the parameter $b$ was fully investigated by experiments and $b_{opt}$ was found for all network parameters. For 3- and 4-node networks the optimal value of $b$ has to be found by trial and error.

Thus, we showed that the proposed heuristics are generally applicable for all network parameters of 2-, 3- and 4-node tandem networks. We also compared our approach with the state-dependent adaptive method and showed that in most of the cases our heuristics work better. It would be interesting to find the theoretical proof of asymptotic efficiency of the developed heuristics and have a straightforward rule to find the optimal value of parameter $b$ (e.g., some dependence of $b_{opt}$ on the network parameters $\lambda$, $\mu_1$,...,$\mu_d$).

# Chapter 4

# State-dependent heuristic for queues in parallel

In this chapter we develop a state-dependent heuristic for simulating population overflow in networks of queues in parallel. No theoretical results are yet known to be efficient for this case. The chapter is organized as follows. In Section 4.1 we describe the model and notation. Sections 4.2.1 and 4.2.2 give, respectively, a motivation and a time-reversal argument behind the proposed heuristic. The heuristic itself is described in Section 4.3, whereas experimental results supporting the heuristic are presented in Section 4.4. Final remarks and conclusions are discussed in Section 4.5.

## 4.1 Model and notation

Consider a network of $d$ queues in parallel. Customers arrive to the network according to the Poisson process with rate $\lambda$ and are routed to queue $i$ with probability $p_i > 0$ with $\sum_i^d p_i = 1$. Let $\lambda_i = \lambda \cdot p_i$ denote the arrival rate at node $i$ and $\mu_i$ denote the service rate at node $i$. We assume that the service times are exponentially distributed and all nodes are stable, i.e., for all $i$ ($i = 1, \ldots, d$) the traffic intensity $\rho_i$ satisfies

$$\rho_i = \frac{\lambda_i}{\mu_i} < 1. \tag{4.1}$$

Without loss of generality we assume that the rates are normalized, i.e.,

$$\sum_{i=1}^{d} (\lambda_i + \mu_i) = 1. \tag{4.2}$$

Since the inter-arrival and service times are exponentially distributed and we are not interested in any time dependent quantities, we simplify our model to the discrete time Markov chain. Thus, let $X = (x_1, ..., x_d)$ be the system state and $S = \sum_{i=1}^d x_i$ be the total number of customers in the network. Again, we are interested in the probability $\gamma(N)$ that starting from the empty state $X_0 = (0, ..., 0)$ the network population reaches level $N$ before returning to $X_0$.

The arrival and service rates at node $i$ under the proposed change of measure we denote by $\tilde{\lambda}_i$ and $\tilde{\mu}_i$.

## 4.2 Preliminary discussion

In this section we will discuss the motivation and time-reversal argument behind the heuristic proposed in this chapter.

### 4.2.1 Motivation of the heuristic

There are no theoretical results yet known to be efficient for all parameters of networks of queues in parallel. The PW change of measure discussed in Section 2.4.2 is effective only for a very limited set of network parameters, namely, when the load at the bottleneck node is much larger than at the other nodes. In that case interchanging the arrival rate with the bottleneck service rate provides efficient estimates. When the service rates are equal or almost equal, this heuristic does not work at all and there has not been reported any other heuristic that would help in that case. Recent studies (e.g., [6], [13]) indicate that the "optimal" change of measure depends on the system state and this dependence is strong along the boundaries (i.e., when one (or more) of the nodes is empty). By knowing a change of measure on the boundaries and in the interior of the state space one might be able to construct a change of measure that approximates the optimal one over the entire state space. We already succeeded to do that for tandem queuing networks in Chapter 3 and now do so for networks of queues in parallel.

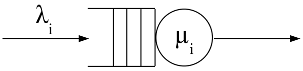
### 4.2.2 Time reversal argument

In this section, similar to what we have done for tandem networks, the time reversal argument in [44] is applied to motivate the change of measure that we will introduce in the next section. Again, it is not a formal proof of its asymptotic efficiency.

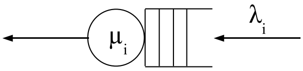
For networks of queues in parallel, the reverse time process (RTPr for short) is even simpler than for tandem networks. Since all queues are independent the RTPr of a network of $d$ queues in parallel is a "combination" of $d$ RTProcesses each of which is an RTPr for a single node (which is a process with the arrival and service rates interchanged, see Figure 4.1), i.e., it is a network of $d$ nodes in parallel where all arrival rates are interchanged with the corresponding service rates.

According to [38] the most likely path to the rare set in the forward time process is, in the limit, i.e., as level $N \to \infty$, the same path by which the reverse time process evolves, starting from the rare set. Since we do not know at which state exactly the rare set is going to be hit we consider the general case, i.e., that it is hit at point $(n_1, ..., n_d)$ with $\sum_{i=1}^{d} n_i = N$ where $N$ is a target level and all $n_i \geq 0$.

Consider the behavior of the RTPr for a single queue $i$ (see Figure 4.1). Customers arrive with the rate $\lambda_i$ and are served at the rate $\mu_i$. When $n_i$ is large enough, i.e., enough to make the server busy all the time and work at its full capacity, the output rate from node $i$ is equal to $\mu_i$. Then, in the forward time process (FTPr for short) the arrival rate at node $i$ is equal to $\mu_i$ (remember that the output rates in the RTPr are the input rates in the FTPr and vice versa). When queue $i$ empties, i.e., $n_i = 0$,



a) Forward time process at queue *i*



a) Reverse time process at queue *i*

Figure 4.1: Time reversal of a single queue

the output rate from queue $i$ in the RTPr is equal to the input rate ($\lambda_i$). Thus, in the FTPr we have the arrival rate at node $i$ equal to $\lambda_i$ and the service rate equal to $\mu_i$ (since the sum of the arrival and service rates at node $i$ is the same for the FTPr and the RTPr and is equal to $\lambda_i + \mu_i$, i.e., remains unchanged. Thus, for each queue $i$ in the network we have a changes of measure for $n_i \gg 0$ such that $\tilde{\lambda}_i = \mu_i$, $\tilde{\mu}_i = \lambda_i$, and no change of measure for $n_i = 0$, i.e., $\tilde{\lambda}_i = \lambda_i$, $\tilde{\mu}_i = \mu_i$. This applies for all $i$ ($i = 1, ..., d$) since the queues are independent. In the following section we will construct the state-dependent change of measure to simulate the population overflow probability in the network of queues in parallel which is a combination of these two changes of measure.

## 4.3 State-dependent heuristic

Denote by $\mathbf{SDH}_i$ the $2 \times 2$ linear operator (matrix) transforming the original rates into the new rates at node $i$ ($i = 1, \ldots, d$). As before, define $[a]^+ = \max(a, 0)$ and $[a]^1 = \min(a, 1)$. Then the change of measure at node $i$ ($i = 1, \ldots, d$) is given by:

$$\begin{bmatrix} \tilde{\lambda}_i \\ \tilde{\mu}_i \end{bmatrix} = \mathbf{SDH}_i \begin{bmatrix} \lambda_i \\ \mu_i \end{bmatrix}, \tag{4.3}$$

$$\mathbf{SDH}_i = \left[\frac{b_i - x_i}{b_i}\right]^+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \left[\frac{x_i}{b_i}\right]^1 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{4.4}$$

$$\mu_i(x_1, ..., x_d) = 0, \text{ for } x_i = 1, x_j = 0\,(i \neq j), \tag{4.5}$$

where $b_i \geq 1$ ($i = 1, \ldots, d$) is some integer number. The first matrix is the identity matrix, corresponding to no change of measure. The second matrix is the identity matrix with the first and the second rows interchanged, which corresponds to interchanging the arrival and service rates at node $i$. Note that the new rates remain to be normalized.

The parameter $b_i$ in the above heuristic is the number of boundary levels for which the change of measure at node $i$ depends on its content $x_i$. Proper selection of the $b_i$'s is crucial for achieving asymptotic efficiency. In general, the "optimal" $b_i$'s (yielding estimates with lowest variance) depend on the set of network parameters as well as the overflow level $N$. Empirical results (Section 4.4) suggest dependence on the traffic intensities $\rho_i$'s at all network nodes.

According to the above change of measure, all nodes may be "pushed" (overloaded) simultaneously, however, to different extent depending on their respective ratios of content $x_i$ relative to $b_i$. This is a state-dependent change of measure, by which empty nodes are not "pushed" at all, and busy nodes are "pushed" harder for higher $x_i/b_i$.

**Remark 4.3.1.** The PW (state-independent) heuristic suggests interchanging the arrival and the service rates at the bottleneck node (node with the highest $\rho_i$). For a single node, say, node $i$, our change of measure, with $b_i = 1$, is identical to PW.

## 4.4 Experimental results

The experiments in this section are designed to demonstrate that the state-dependent change of measure proposed in Section 4.3 always yields asymptotically efficient estimates with less than linearly growing relative error. In Section 4.4.1 we consider the performance of the proposed heuristic in comparison with other methods; in Section 4.4.3 we experimentally verify its validity for all sets of network parameters.

### 4.4.1 Performance

In this section, similarly to what we have done in Section 3.5 for tandem networks, we will present the comparison of the proposed heuristic (Section 4.3) with the PW heuristic and the SDA algorithm (Section 2.5). All notation used in Section 3.4.1 apply here, as well as the restriction, discussed in Section 3.4.2.

All simulation experiments are gathered with the same number of replications, namely, $10^6$. The results are shown in Tables 4.1–4.9. For each estimate in these tables we include the relative error (*RE*) (in percent). For the purpose of comparing SDH and SDA we also include *VRR* (relative to SDA). Hence, $VRR > 1$ implies efficiency gain of SDH over SDA. Estimates obtained using the PW heuristic are also presented, however, these are not necessarily accurate or stable. In general, numerical results are difficult to obtain for larger and/or higher overflow levels. Whenever feasible, numerical results (using the algorithm outlined in [32], [27]) are included to verify the correctness of the simulation estimates. Otherwise, the corresponding table entry is marked with an asterix ($*$). In these cases, agreement of the SDH and SDA estimates may be an indication of correctness.

We experimented with 2-, 3- and 4-node (symmetric and asymmetric) networks of queues in parallel For each case network parameters are chosen in such a way that the PW heuristic is not effective. Typically, this is the case for symmetric networks of queues in parallel, i.e., all nodes have the same utilization, or when the higher utilizations are sufficiently close to each other.

The experimental results are presented in Tables 4.1–4.3 for 2-node, in Tables 4.4–4.6 for 3-node and in Tables 4.7–4.9 for 4-node networks of queues in parallel. For each network we have two symmetric cases (with low and high loads) and one asymmetric case (different loads and different routing probabilities).

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 6.4837e-14 | 2.4899e-14 ± 7.45 | 3 | 6.4826e-14 ± 0.06 | 3 | 6.4800e-14 ± 0.12 | 2.97 |
| 50 | 1.1675e-28 | 3.7971e-29 ± 36.2 | 4 | 1.1684e-28 ± 0.06 | 4 | 1.1667e-28 ± 0.15 | 1.84 |
| 100 | 1.8445e-58 | 1.9774e-59 ± 14.5 | 5 | 1.8527e-58 ± 0.08 | 5 | 1.8480e-58 ± 0.25 | 0.79 |

Table 4.1: 2-node parallel network - symmetric ($\lambda_i = 0.1,\ \mu_i = 0.4$) ($\rho_i = 0.25$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 1.9796e-08 | 1.1928e-08 ± 11.7 | 4 | 1.9800e-08 ± 0.06 | 4 | 1.9814e-08 ± 0.14 | 1.58 |
| 50 | 2.5813e-17 | 8.5168e-18 ± 12.7 | 5 | 2.5834e-17 ± 0.06 | 6 | 2.5904e-17 ± 0.17 | 0.98 |
| 100 | 2.0926e-35 | 2.3032e-35 ± 86.2 | 6 | 2.0923e-35 ± 0.07 | 7 | 2.0895e-35 ± 0.26 | 0.66 |

Table 4.2: 2-node parallel network - symmetric ($\lambda_i = 0.15,\ \mu_i = 0.35$) ($\rho_i = 0.43$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b_1$ | $b_2$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 5.6704e-13 | 7.2661e-13 ± 22.1 | 3 | 5.6480e-013 ± 0.12 | 2 | 5 | 5.6600e-13 ± 0.15 | 5.87 |
| 50 | 4.8047e-26 | 4.7674e-26 ± 3.88 | 3 | 4.7993e-026 ± 0.16 | 2 | 7 | 4.8188e-26 ± 0.20 | 3.30 |
| 100 | 3.4493e-52 | 3.3333e-52 ± 3.01 | 3 | 3.4434e-052 ± 0.21 | 2 | 10 | 3.4563e-52 ± 0.28 | 3.23 |

Table 4.3: 2-node parallel network - asymmetric ($\lambda_1 = 0.12,\ \lambda_2 = 0.08,\ \mu_i = 0.4$) ($\rho_1 = 0.3,\ \rho_2 = 0.2$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 4.2209e-15 | 8.6265e-16 ± 21.7 | 3 | 4.2222e-15 ± 0.06 | 3 | 4.2057e-015 ± 0.20 | 7.40 |
| 50 | * | 2.0868e-33 ± 9.46 | 3 | 5.6552e-32 ± 0.09 | 4 | 5.6263e-032 ± 0.26 | 5.29 |
| 100 | * | 2.0758e-68 ± 11.3 | 3 | 2.5319e-66 ± 0.31 | 5 | 2.5110e-066 ± 0.43 | 14.8 |

Table 4.4: 3-node parallel network - symmetric ($\lambda_i = 0.05,\ \mu_i = 0.25$) ($\rho_i = 0.2$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 8.3550e-06 | 4.9906e-06 ± 20.8 | 5 | 8.3574e-06 ± 0.07 | 7 | 8.3550e-06 ± 0.19 | 7.26 |
| 50 | * | 2.7409e-13 ± 17.0 | 4 | 1.0608e-12 ± 0.38 | 8 | 1.0566e-12 ± 0.22 | 64.0 |
| 100 | * | 1.5623e-28 ± 8.69 | 5 | 3.7658e-27 ± 0.93 | 9 | 3.8483e-27 ± 0.32 | 114. |

Table 4.5: 3-node parallel network - symmetric ($\lambda_i = 0.1,\ \mu_i = 0.2$) ($\rho_i = 0.5$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b_1$ | $b_2, b_3$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 8.2980e-10 | 7.8038e-10 ± 2.17 | 3 | 8.29758e-010 ± 0.16 | 2 | 6 | 8.3082e-10 ± 0.19 | 27.4 |
| 50 | * | 9.1128e-20 ± 2.64 | 3 | 9.31422e-020 ± 0.16 | 2 | 10 | 9.3459e-20 ± 0.25 | 7.39 |
| 100 | * | 1.1746e-39 ± 2.97 | 3 | 1.15892e-039 ± 0.39 | 2 | 14 | 1.1798e-39 ± 0.38 | 7.11 |

Table 4.6: 3-node parallel network - asymmetric ($\lambda_1 = 0.1,\ \lambda_2 = 0.075,\ \lambda_3 = 0.025,\ \mu_i = 0.25$) ($\rho_1 = 0.4,\ \rho_2 = 0.3,\ \rho_3 = 0.1$)

| N | Numerical | PW | SDA | | SDH | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | * | 8.5099e-13 ± 12.0 | 4 | 7.3197e-12 ± 0.08 | 4 | 7.3465e-12 ± 0.30 | 33.8 |
| 50 | * | 1.8289e-27 ± 48.1 | 4 | 5.0880e-26 ± 0.14 | 5 | 5.1083e-26 ± 0.41 | 43.0 |
| 100 | * | 4.6236e-58 ± 7.58 | 5 | 3.1658e-55 ± 0.14 | 5 | 3.1384e-55 ± 0.78 | 19.2 |

Table 4.7: 4-node parallel network - symmetric ($\lambda_i = 0.05,\ \mu_i = 0.2$) ($\rho_i = 0.25$)

| N | Numerical | PW | SDA | | SDH | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | * | 7.2898e-06 ± 9.09 | 3 | 1.7915e-05 ± 0.13 | 8 | 1.7924e-05 ± 0.24 | 23.5 |
| 50 | * | 1.1733e-13 ± 18.0 | 4 | 9.8414e-13 ± 0.26 | 8 | 9.8027e-13 ± 0.31 | 120. |
| 100 | * | 4.1708e-30 ± 16.5 | 5 | 3.4284e-28 ± 0.59 | 9 | 3.4008e-28 ± 0.49 | 386. |

Table 4.8: 4-node parallel network - symmetric ($\lambda_i = 0.08,\ \mu_i = 0.17$) ($\rho_i = 0.47$)

| N | Numerical | PW | SDA | | SDH | | | |
|---|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b_1$ | $b_2, b_3, b_4$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | * | 2.8583e-12 ± 18.9 | 4 | 2.4917e-12 ± 0.15 | 2 | 6 | 2.5012e-12 ± 0.35 | 135. |
| 50 | * | 1.8266e-25 ± 2.59 | 4 | 2.1002e-25 ± 0.22 | 2 | 8 | 2.1268e-25 ± 0.64 | 56.7 |
| 100 | * | 1.4262e-51 ± 7.11 | 4 | 1.3031e-51 ± 0.37 | 2 | 10 | 1.5248e-51 ± 1.38 | 22.0 |

Table 4.9: 4-node parallel network - asymmetric ($\lambda_1 = 0.06,\ \lambda_2 = \lambda_3 = 0.04,\ \lambda_4 = 0.02,\ \mu_i = 0.2$)
($\rho_1 = 0.3,\ \rho_2 = \rho_3 = 0.2,\ \rho_4 = 0.1$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 6.4837e-14 | 2.4899e-14 ± 7.45 | 3 | 6.4826e-14 ± 0.06 | 2 | 6.5053e-14 ± 0.30 (0.32) | 0.54 |
| | | | | | 3* | 6.4800e-14 ± 0.12 (0.12)* | 2.97 |
| | | | | | 4 | 6.4964e-14 ± 0.13 (0.13) | 2.02 |
| | | | | | 5 | 6.5054e-14 ± 0.19 (0.19) | 0.82 |
| | | | | | 6 | 6.4777e-14 ± 0.29 (0.29) | 0.34 |
| 50 | 1.1675e-28 | 3.7971e-29 ± 36.2 | 4 | 1.1684e-28 ± 0.06 | 2 | 1.1567e-28 ± 0.92 (2.45) | 0.06 |
| | | | | | 3 | 1.1670e-28 ± 0.23 (0.23) | 0.89 |
| | | | | | 4* | 1.1667e-28 ± 0.15 (0.16)* | 1.84 |
| | | | | | 5 | 1.1681e-28 ± 0.18 (0.18) | 1.09 |
| | | | | | 6 | 1.1680e-28 ± 0.26 (0.26) | 0.50 |
| 100 | 1.8445e-58 | 1.9774e-59 ± 14.5 | 5 | 1.8527e-58 ± 0.08 | 3 | 1.8374e-58 ± 1.00 (1.24) | 0.06 |
| | | | | | 4 | 1.8527e-58 ± 0.33 (0.34) | 0.53 |
| | | | | | 5* | 1.8480e-58 ± 0.25 (0.26)* | 0.79 |
| | | | | | 6 | 1.8538e-58 ± 0.28 (0.28) | 0.56 |
| | | | | | 7 | 1.8613e-58 ± 0.37 (0.37) | 0.29 |

Table 4.10: 2-node parallel network - symmetric ($\lambda_i = 0.1$, $\mu_i = 0.4$) ($\rho_i = 0.25$)
('*' denotes $b_{opt}$ or the minimum of exact $RE$)

Results in Tables 4.1 and 4.3 show that unlike PW, SDH yields correct (compared with numerical results), stable, and asymptotically efficient estimates with relative error increasing (sub-)linearly in the overflow level $N$. Note that the best $b_1$ and $b_2$ are equal only in the symmetric case. In the asymmetric case, $b_1 = 2$, $b_2 > b_1$ and increases with the overflow level $N$. SDA produces correct and stable results as well; however, it appears to be less efficient than SDH (as indicated by $VRR$ ratios significantly higher than one).

Experimental results in Tables 4.4 and 4.9 show that the PW heuristic can not be applied to obtain reliable estimates. At the same time, SDH yields correct, stable, and asymptotically efficient estimates with relative error increasing (sub-)linearly in the overflow level $N$. The correctness can be assumed by the agreement with the SDA estimates since numerical results are not feasible (except for level $N = 25$ for 3-node case). Again, the "best" $b_i$ $(i = 1, ..., d)$ are equal only in the symmetric case. In the asymmetric case, $b_1 = 2$ and $b_i > b_1$ (for $i \geq 2$) and increases with the overflow level $N$.

**Remark 4.4.1.** It is noteworthy that for networks of queues in parallel (in comparison to tandem) the performance gain of SDH over the SDA algorithm is much larger, especially in cases of 3 and 4 nodes.

### 4.4.2 Sensitivity with respect to $b$

To check the sensitivity of SDH with respect to $b$, the experiments have been gathered for different values of $b$ for an example of a 2-node network. The results are represented in Table 4.10. For different values of $b$ around the "best" (marked by '*' in the table), the resulting estimate is displayed together with its relative error, estimated from simulation and computed numerically (shown between parentheses, where the best numerically computed $RE$ is marked with '*'). The numerical $RE$ is obtained from an algorithm similar to that outlined in [25] but adapted to compute the variance of the SDH estimator for an example of a network of 2 nodes in parallel. As one can see from Table 4.10 the empirical relative error is consistent with the computed relative error. One can also see that the accuracy of the simulation estimates is not too sensitive with respect to $b$.

### 4.4.3 Validation

In this section we will experimentally demonstrate that the state-dependent heuristic we proposed in Section 4.3 leads to asymptotically efficient estimates for all network parameters. Note that in case of queues in parallel the freedom for choosing network parameters is larger than for tandem networks, since there are more variables to play with. For 2 queues in parallel there are four parameters: two arrival rates and two service rates, but only three of them are independent due to normalization Equation (4.2). For 4 queues in parallel the number of parameters is even larger (four arrival and four service rates, i.e., seven independent parameters).

There are two possible cases of networks of queues in parallel: symmetric networks (equal loads at all network nodes, i.e., $\rho_i = \lambda_i/\mu_i = \rho_j$ for all $i, j$), and asymmetric networks (different loads). We made the experiments for both cases assuming for

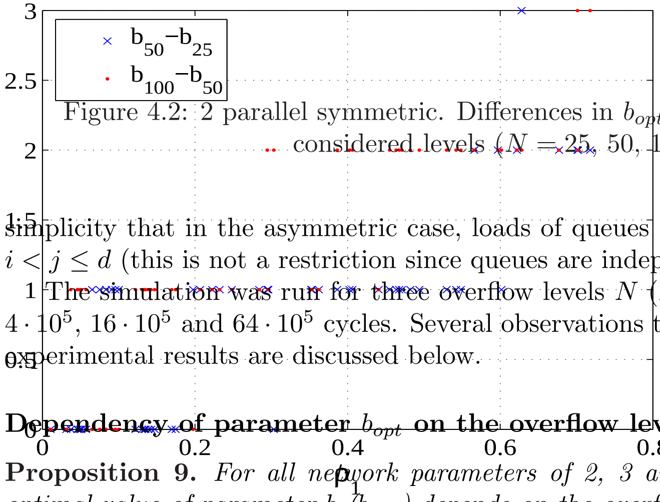


Figure 4.2: 2 parallel symmetric. Differences in $b_{opt}$ between consecutively considered levels ($N = 25$, 50, 100)

simplicity that in the asymmetric case, loads of queues are ordered, i.e., $\rho_i \geq \rho_j$ for $i < j \leq d$ (this is not a restriction since queues are independent).

The simulation was run for three overflow levels $N$ ($N = 25$, 50 and 100) at $10^5$, $4 \cdot 10^5$, $16 \cdot 10^5$ and $64 \cdot 10^5$ cycles. Several observations that were made based on the experimental results are discussed below.

### Dependency of parameter $b_{opt}$ on the overflow level

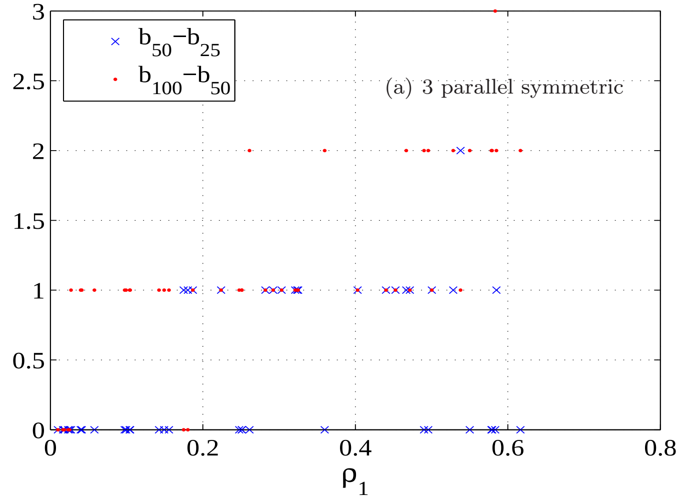
**Proposition 9.** *For all network parameters of 2, 3 and 4 queues in parallel, the optimal value of parameter b ($b_{opt}$) depends on the overflow level N.*

To see this, consider the difference between values $b_{opt}$ for levels 25, 50 and 100, i.e., $b_{opt}(100) - b_{opt}(50)$ and $b_{opt}(50) - b_{opt}(25)$. Non-zero difference means that $b_{opt}$ changes with level $N$. As one can see from Figures 4.2–4.3, there are network parameters (not all) with one of the differences being non-zero, which supports the claim.

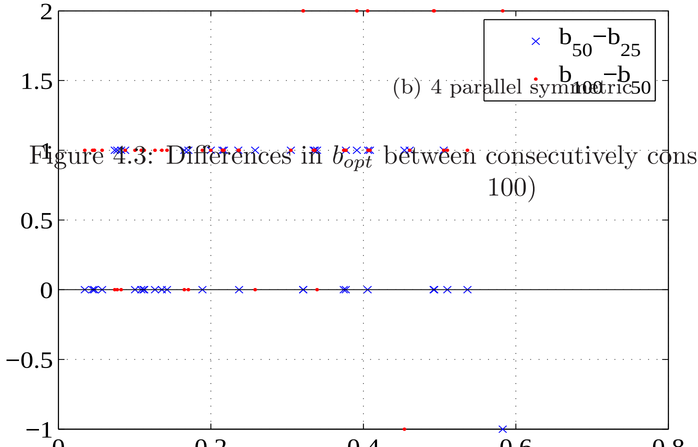
### Behavior of relative error

**Proposition 10.** *For all network parameters of 2, 3 and 4 queues in parallel, the heuristic proposed in Section 4.3 gives estimates with less than linearly growing relative error.*

To be able to see this, we considered the relative error ratios ($RER$s) between levels $N = 100$ and $N = 50$, and between levels $N = 50$ and $N = 25$, i.e., $RE_{100}/RE_{50}$ and $RE_{50}/RE_{25}$ (see similar discussion in Section 3.6.1, Proposition 4). For bounded $RE$ these ratios should be near 1, for linearly growing $RE$ they should be near 2. $RER$s that are less than 2 indicate that $RE$ grows less than linearly with level $N$. Figures 4.4a–4.5a-b clearly show the validity of Proposition 10.

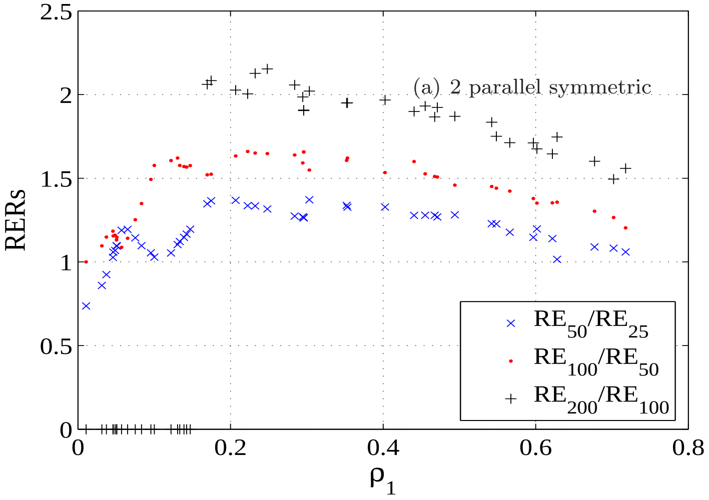


(a) 3 parallel symmetric

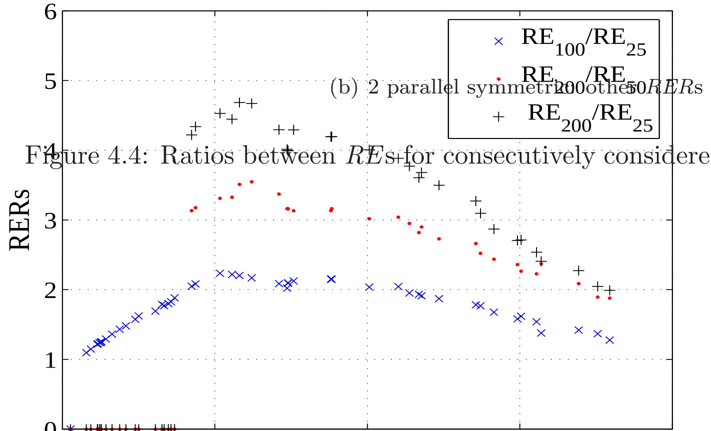


(b) 4 parallel symmetric

Figure 4.3: Differences in $b_{opt}$ between consecutively considered levels ($N = 25, 50, 100$)



(a) 2 parallel symmetric



(b) 2 parallel symmetric other $RER$s

Figure 4.4: Ratios between $RE$s for consecutively considered levels ($N = 25$, 50, 100)

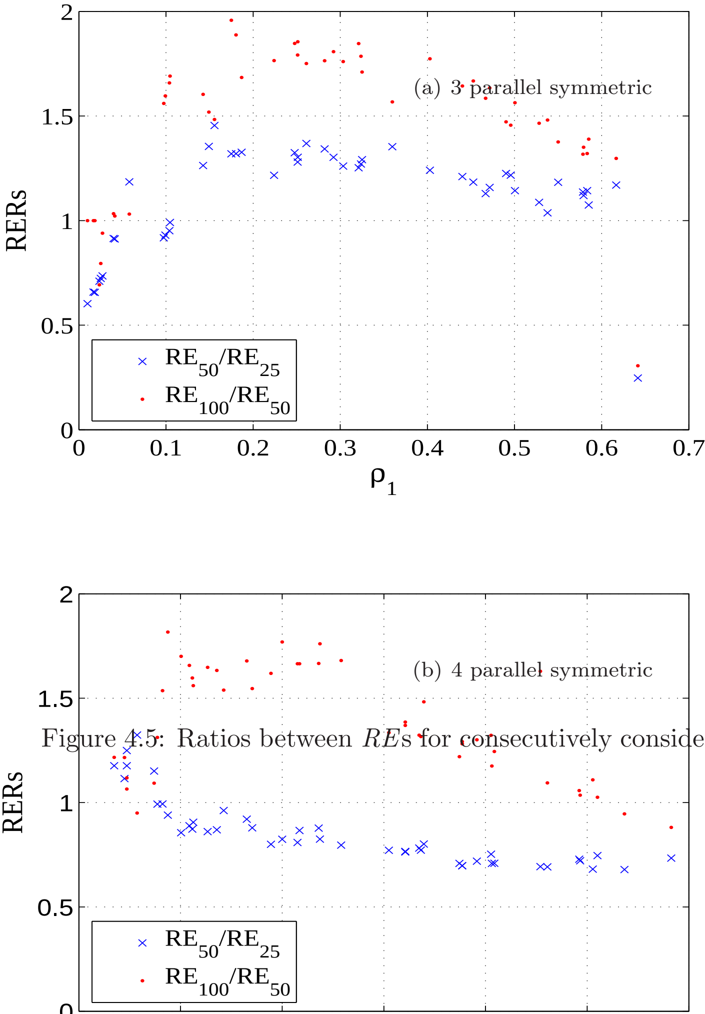


Figure 4.5: Ratios between $RE$s for consecutively considered levels ($N = 25$, 50, 100)

**Remark 4.4.2.** Note that the fact that the *RER*s of level 100 to level 50 are higher than *RER*s of level 50 to 25 does not contradict Proposition 10. To see that consider the additional value of level $N$, namely, $N = 200$ and $RE_{200}/RE_{100}$ (Figure 4.4a, the case of 2-nodes in parallel). Although, $RE_{200}/RE_{100} > RE_{100}/RE_{50}$, i.e., the same tendency continues, comparing *RER*s of $RE_{200}/RE_{50}$, $RE_{100}/RE_{25}$ (Figure 4.4b), in which case the level was quadrupled and, hence, for the sublinear growth *RER*s should be less than 4, one can clearly see that the sublinear growth property is actually satisfied. Also, since $RE_{200}/RE_{50} < 4$ and $RE_{50}/RE_{25} < 2$ the ratio $RE_{200}/RE_{25} < 8$ (as one can see from Figure 4.4b) and, for 3 and 4-node cases (Figure 4.5a–b), since $RE_{100}/RE_{50} < 2$ and $RE_{50}/RE_{25} < 2$ the ratio $RE_{100}/RE_{25}$ will be less than 4. Thus, Proposition 10 is supported.

### Optimal value of parameter $b$

**Proposition 11.** *For all network parameters of 2-, 3- and 4-node symmetric networks of queues in parallel (i.e., $\rho_i = \rho$ for all $i \leq d$), the optimal value of parameter $b$ grows with the load $\rho$.*

The validity of the proposition can be easily seen from Figures 4.6–4.7. The optimal value of parameter $b_i$ $(i = 1, ..., d)$ for symmetric networks of queues in parallel is the same for each queue $i$ and can be found with Algorithm 2 (Section 3.6.1).

### Guideline for choosing $b$ in case of asymmetric networks

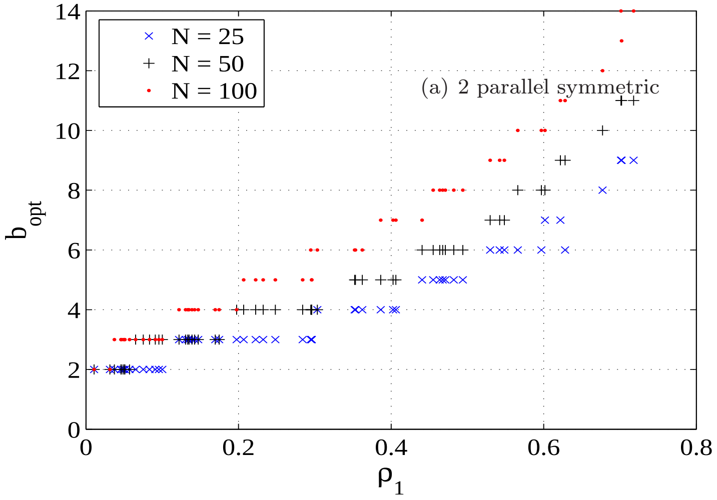
Unlike the symmetric case the value $b_{i,opt}$ for asymmetric networks can be different for each queue depending on how loaded the queue is. That makes it more time consuming to find $b_{i,opt}$ by simply checking all the possibilities. In practice, however, we saw that the number of possibilities could be restricted by considering only the cases with $b_i \leq b_j$ for $i < j$, i.e., for a higher loaded node the value $b_{opt}$ is at most as large as for the lower loaded nodes. (Remember that we consider the case when queues are ordered by their loads.) This restriction, however, does not help much since the number of possible combinations of $b_{i,opt}$ is still large and grows very quickly with the number of nodes in the network.

From a practical point of view, it can be good enough to obtain not necessary the optimal but just good estimates. So, we restrict ourselves to show that for given parameters $b_i$ $(i = 1, ..., d)$ the proposed heuristic (SDH) gives reliable estimates. We experimented only with two different sets of parameters $b_i$, namely,

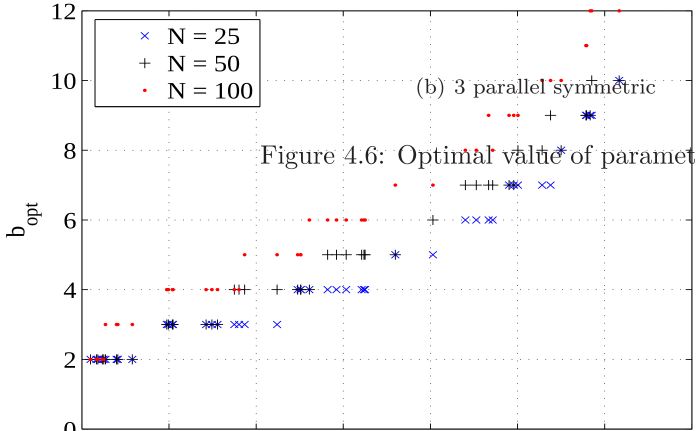
1) all $b_i$'s equal, i.e., $b_1 = b_2 = ... = b_d$,

2) all but the first $b_i$'s equal and $b_1 = 2$ (remember that the first queue is the bottleneck), i.e., $b_1 = 2$, $b_2 = ... = b_d$ and $b_i > 2$ for $i \geq 2$.

The reason for this choice was that in most of the cases considering equal $b_i$'s was enough to obtain good estimates. In cases when it did not work we tried, for a couple of examples, different values of $b_i$'s. It happened that $b_1$ needed to be smaller and the optimal value of $b_1$ was equal to 2. Then, for cases where choosing all $b_i$'s equal was not sufficiently good we tried $b_1 = 2$ and it happened to work well, so we decided to restrict ourselves to that case. The practical conclusion is formulated in the following proposition.

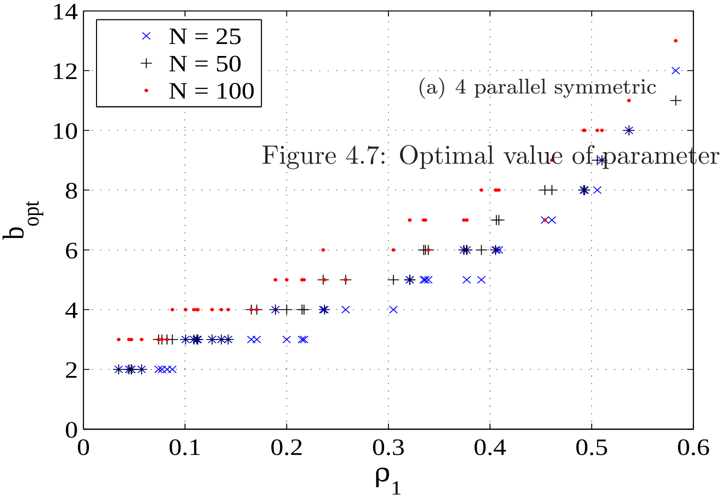


(a) 2 parallel symmetric



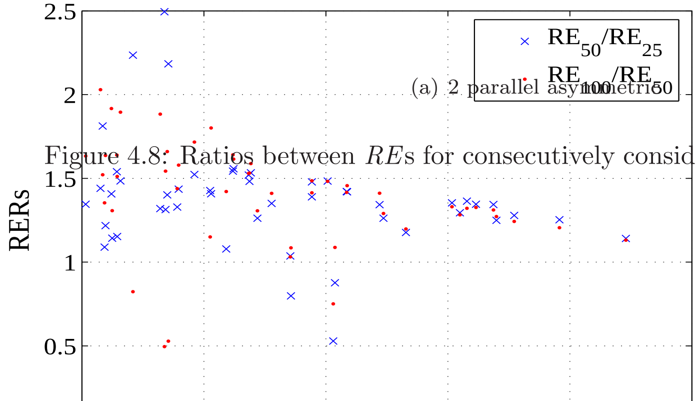
(b) 3 parallel symmetric

Figure 4.6: Optimal value of parameter $b$ ($b_{opt}$)



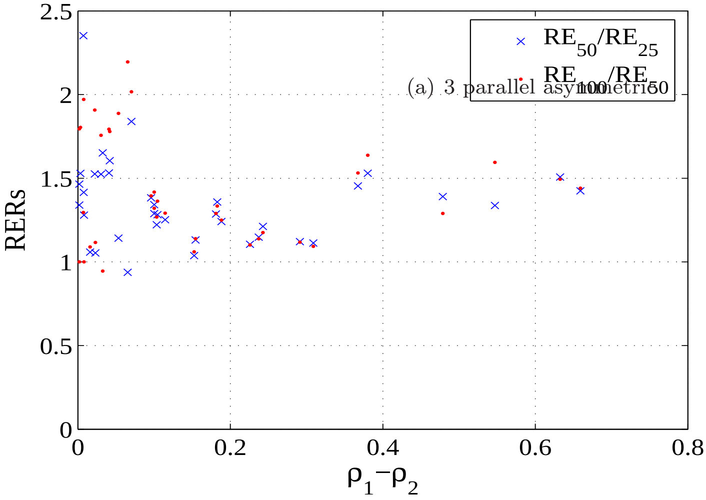
(a) 4 parallel symmetric

Figure 4.7: Optimal value of parameter $b$ ($b_{opt}$)

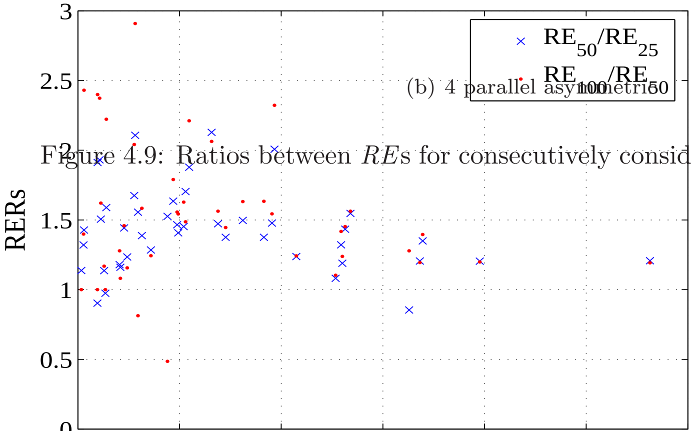


(a) 2 parallel asymmetric

Figure 4.8: Ratios between $RE$s for consecutively considered levels ($N = 25, 50, 100$)



(a) 3 parallel asymmetric



(b) 4 parallel asymmetric

Figure 4.9: Ratios between $REs$ for consecutively considered levels ($N = 25, 50, 100$)

**Proposition 12.** *For most of the network parameters of 2-, 3- and 4-node asymmetric networks of queues in parallel, SDH for at least one of the cases 1)-2) gives estimates with less than linear growth.*

Figures 4.8–4.9 support the above proposition.

## 4.5 Conclusion

In this chapter we proposed the state-dependent heuristic to estimate the probability of population overflow in networks of queues in parallel. We showed experimentally that SDH gives asymptotically efficient estimates with less than linearly growing relative error. There was no other heuristic known to be efficient for all network parameters. We also compared our heuristic (SDH) with the heuristic obtained using the adaptive technique (SDA) and showed that SDH has more advantages: it is quicker, more straightforward and easier to implement, and it is also more efficient. Moreover, its effectiveness does not diminish for larger networks. It is still an open question how to find the optimal value of parameter $b$. We used trial and error, but there might be an exact dependence of $b_{opt}$ on the network parameters. Investigating that would be of much interest. Another open issue is the theoretical proof of asymptotic efficiency, which can give better understanding of the system behavior and help to extend the heuristic to non-Markovian networks of queues in parallel.

# Chapter 5

# State-dependent heuristics for Jackson networks

In this chapter we develop a state-dependent change of measure for efficient simulation of network overflow probabilities in general Jackson queuing networks. In Section 5.1 we introduce the model and notation; in Section 5.2 we describe the heuristic proposed in [22] to simulate the probability of an *arbitrary buffer* overflow in a Jackson network. This heuristic is used to derive our change of measure, described in Section 5.3, for estimating *total network* overflow probability. In Section 5.4 we present experimental results and discuss the performance of our heuristics in comparison with the SDA algorithm and the PW state-independent heuristic. For feed-forward networks we do an extensive experimentation and discuss the results in Section 5.5. We conclude in Section 5.6.

## 5.1 Model and notation

Consider a Jackson network consisting of $d$ nodes (queues), each having its own buffer of infinite size. Customers arrive at node $i$ $(i = 1, \ldots, d)$ according to a Poisson process with rate $\lambda_i$. The service time of a customer at node $i$ is exponentially distributed with rate $\mu_i$ $(= 1, \ldots, d)$. Customers that leave node $i$ join node $j$ with probability $p_{ij}$ $(i, j = 1, \ldots, d)$ or leave the network with probability $p_{ie}$ $(i = 1, \ldots, d)$. Without loss of generality we assume that

$$\sum_{i=1}^{d} (\lambda_i + \mu_i) = 1. \tag{5.1}$$

We also assume that the queuing network is stable, i.e., $\gamma_i < \mu_i$ for all $i = 1, \ldots, d$, where $\gamma_i$ is the total arrival rate at node $i$, as determined from the traffic equations

$$\gamma_i = \lambda_i + \sum_{j=1}^{d} \gamma_j \, p_{ji}, \; i = 1, ..., d.$$

Let $x_{i,t}$ $(i = 1,\dots,d)$ denote the number of customers at node $i$ at time $t \geq 0$ (including those in service). Then the vector $\mathbf{X}_t = (x_{1,t}, x_{2,t}, ..., x_{d,t})$ is a Markov process representing the state of the network at time $t$. Denote by $S_t$ the total number of customers in the network at time $t$, i.e., $S_t = \sum_{i=1}^{d} x_{i,t}$. Since inter-arrival and service times are exponentially distributed and we are not interested in quantities dependent on time (e.g., waiting time) the model can be simplified to the discrete time Markov chain. Namely, $X = (x_1, ..., x_d)$ is the system state and $S = \sum_{i=1}^{d} x_i$ is the total number of customers in the network, called *the network population.*

Assuming that the initial network state is $\mathbf{X}_0 = (0, 0, ..., 0)$, corresponding to an empty network, we are interested in the probability that the network population reaches some high level $N \in \mathbb{N}$ before becoming empty. We denote this probability by $\gamma(N)$ and refer to it as the *population overflow probability*, starting from the initial state $\mathbf{X}_0$. Since the associated event is typically rare, importance sampling may be used to efficiently estimate this probability.

## 5.2 Buffer overflow at an arbitrary node

In this section we present an asymptotically efficient change of measure (as proposed in [22]) to simulate buffer overflow probability at an arbitrary node. This change of measure plays a key role in the heuristics proposed in this chapter. We need to introduce some notation.

Consider a Jackson network as described in Section 5.1 and let all nodes in the network be indexed by the set $\mathcal{H}$. These nodes are further categorized by one (arbitrary) "target" node indexed by $k$ and the remaining "feeder" nodes indexed by the set $\mathcal{F}$. Thus, $\mathcal{H} \equiv \{k\} \cup \mathcal{F}$. In [22] a state-independent change of measure is proposed to estimate the probability that the buffer content at the target node exceeds a large level during its busy period (a busy period of the target queue is initiated when an arrival to it finds it empty, and ends when the target queue re-empties). Under this change of measure, the simulated queuing network is again a Jackson network in which the original inter-arrival and service time distributions are exponentially twisted to achieve asymptotic efficiency. Moreover, only the target node (node $k$) is unstable while each of the other (feeder) nodes is either stable (in the set $\mathcal{S} = \mathcal{F} - \mathcal{C}$) or critical, i.e., an input rate is equal to an output rate (in the set $\mathcal{C} \subseteq \mathcal{F}$).

Let $\tilde{\lambda}_i$, $\tilde{\mu}_i$, and $\tilde{p}_{ij}$ $(i = 1,\dots,d$ and $j = 1,\dots,d)$ be the new external arrival rates, service rates, and routing probabilities, respectively. Also, define the constants $c_i \geq 1$ for $i \in \mathcal{H}$, and let $\mathcal{D} \subset \mathcal{H}$ denote the set $\{i : \lambda_i = 0\}$. The change of measure in [22] is characterized as follows:

$$\tilde{\lambda}_i = c_i \lambda_i,\ i \in \mathcal{H}, \tag{5.2}$$

$$\tilde{\lambda}_i = 0,\ i \in \mathcal{D}. \tag{5.3}$$

$$\tilde{p}_{ij} = \frac{c_j}{c_i} \frac{\mu_i}{\tilde{\mu}_i} p_{ij},\ i, j \in \mathcal{H}, \tag{5.4}$$

$$\tilde{p}_{ie} = \frac{1}{c_i} \frac{\mu_i}{\tilde{\mu}_i} p_{ie},\ i \in \mathcal{H}, \tag{5.5}$$

where the new service rates $\tilde{\mu}_i$ and the unknown constants $c_i$ $(i = 1, \ldots, d)$ are determined from the non-linear program given as follows:
Maximize $c_k$ subject to the following constraints:

$$\sum_{i \in \mathcal{H}} (\tilde{\lambda}_i + \tilde{\mu}_i) = \sum_{i \in \mathcal{H}} (\lambda_i + \mu_i), \tag{5.6}$$

$$\sum_{j \in \mathcal{H}} \tilde{p}_{ij} + \tilde{p}_{ie} = 1, \tag{5.7}$$

$$\tilde{\gamma}_i = \tilde{\lambda}_i + \sum_{j \in \mathcal{F}} \tilde{p}_{ji} \tilde{\gamma}_j + \tilde{\mu}_k \tilde{p}_{ki}. \tag{5.8}$$

The constants $c_i$ and the new service rates $\tilde{\mu}_i$ are such that the feeder nodes under the change of measure are either stable, i.e.,

$$\tilde{\mu}_i > \tilde{\gamma}_i, \; i \in \mathcal{S}, \tag{5.9}$$

or, critical, i.e.,

$$\tilde{\mu}_i = \tilde{\gamma}_i, \; i \in \mathcal{C}. \tag{5.10}$$

Equation (5.8) is the traffic equation for the system under the change of measure, and (5.6) assures that the sum of all rates in the new system is equal to the sum of all rates in the original system.

Under the restriction that the queue lengths at the feeder nodes are initially bounded, the change of measure characterized above is proven to be asymptotically efficient for estimating the probability of overflow at the target node (node $k$) during its busy cycle (see [22]). In the sequel of this chapter we refer to it as the $\mathbf{JN}_k$ change of measure, where $k$ is the index of the arbitrary (target) node in the network.

**Remark 5.2.1.** When the service rates at the feeder nodes are sufficiently large (for example, when the target node has higher load than all its feeder nodes, see Equation (5.11)) this change of measure can be determined explicitly (see [22] and discussion below); when the target node is the bottleneck of the (whole) network this change of measure is identical to PW ([15]) and the change of measure proposed in [17] to simulate network population overflow.

Formally, let $R = (r_{ij} : i, j \in \mathcal{H})$ equal $(I - P)^{-1}$ ($R$ is a $d \times d$ matrix). Since the network is stable, $r_{ij}$ is the expected number of visits to queue $j$ by a customer starting from queue $i$, before it leaves the system. Note that $r_{ik} \leq r_{kk}$. If, for each $i \in \mathcal{F}$, the service rates at the feeder nodes satisfy the inequality

$$\mu_i > \gamma_i \left( 1 + \frac{r_{ik}}{r_{kk}} \left( \frac{\mu_k}{\gamma_k} - 1 \right) \right), \tag{5.11}$$

then, the change of measure characterized above is determined explicitly as follows:

- all feeder nodes are stable, i.e., the set $\mathcal{C}$ is empty, and $\tilde{\mu}_i = \mu_i$ for $i \in \mathcal{F}$, i.e. the service rates of the feeder nodes do not change under change of measure and are equal to the original service rates. The target node (node $k$) is unstable, with

$$\tilde{\mu}_k = \frac{(r_{kk} - 1)\mu_k + \gamma_k}{r_{kk}}. \tag{5.12}$$

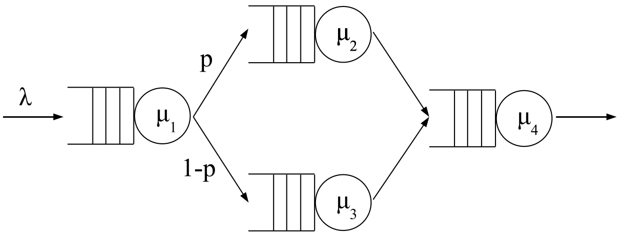


Figure 5.1: 4-node feed-forward network

- for each $i \in \mathcal{H}$,

$$c_i = 1 + \frac{r_{ik}}{r_{kk}} \left( \frac{\mu_k}{\gamma_k} - 1 \right), \tag{5.13}$$

$$\tilde{\lambda}_i = c_i \lambda_i, \tag{5.14}$$

  and

$$\tilde{\gamma}_i = c_i \gamma_i. \tag{5.15}$$

- for $i, j \in \mathcal{H}$,

$$\tilde{p}_{ij} = \frac{c_j}{c_i} \frac{\mu_i}{\tilde{\mu}_i} p_{ij}, \tag{5.16}$$

  and

$$\tilde{p}_{ie} = \frac{1}{c_i} \frac{\mu_i}{\tilde{\mu}_i} p_{ie}. \tag{5.17}$$

The above explicit change of measure can be applied for any network topologies (including tandem, parallel, etc) and is asymptotically efficient to simulate overflow at the bottleneck node. However, for simulating network population overflow it is not always asymptotically efficient, as shown in [24], [25] for 2-node tandem networks; for feed-forward and feedback topologies one can see that from the experiments presented below (cf. Section 5.4).

## 5.3 State-dependent heuristics

In this section we present heuristic state-dependent probability measures to efficiently simulate Jackson networks of feed-forward (Section 5.3.1) and feedback (Section 5.3.2, one small case) topologies. In Section 5.3.3 we describe the generalization of the heuristic for any feed forward network.

### 5.3.1 SDH for a feed-forward network

To describe our state-dependent heuristic for feed-forward Jackson networks, we use the specific example depicted on Figure 5.1. The traffic intensity at node $i$ is $\rho_i = \gamma_i / \mu_i$, where $\gamma_i$ is the total arrival rate at node $i$ ($i = 1, 2, 3, 4$). We also assume that $\rho_1 \le \rho_2 \le \rho_3 \le \rho_4$, since in that case the inequality (5.11) holds and $\mathbf{JN}_i$ can be explicitly calculated for all $i = 1, ..., 4$ from Remark 5.2.1.

### Basic idea

The basic idea behind the heuristic is very simple: in a given feed-forward network we look at each subnetwork of nodes in parallel as one "big node". The feed-forward network then becomes a tandem network, for which we apply the heuristic from Appendix A. We "push" each node depending on the number of customers in it. For the "big node" we use the heuristic for queues in parallel from Section 4.3.

**Remark 5.3.1.** Note, that for a tandem network a fully state-dependent (Appendix A) and not a partly state-dependent (Section 3.3.1) heuristic is used. The former has been developed in the context of feed-forward networks and has experimentally shown better performance. At the same time, this fully state-dependent heuristic applied to real tandem networks (with $b_i = b$ for $i = 1, ..., d$) did not improve the performance compared to the already developed heuristics (Section 3.3.1). See Appendix A for more detail.

Below we introduce the notation and apply the heuristic for the example of a feed-forward network in Figure 5.1.

### Notation

Let $\mathbf{\Theta}$ be a vector with the arrival rate, the service rates at nodes 1, 2, 3, 4, and the routing probability $p$ (of going from node 1 to node 2), respectively, under the original probability measure, i.e.,

- $\mathbf{\Theta} = [\lambda, \mu_1, \mu_2, \mu_3, \mu_4, p]$,
  i.e., the parameter vector corresponding to no change of measure.

The state-independent (and asymptotically efficient) change of measure to overflow (only) node $k$ in the feed-forward network is denoted by $\tilde{\mathbf{\Theta}}_k$ ($k = 1, 2, 3, 4$); it is characterized in [22] and can be determined as described in Section 5.2. It follows that

- $\tilde{\mathbf{\Theta}}_1 = [\mu_1, \lambda, \mu_2, \mu_3, \mu_4, p]$,
  i.e., $\lambda$ and $\mu_1$ interchanged to overflow node 1
- $\tilde{\mathbf{\Theta}}_2 = [\mu_2 + \lambda(1-p), \mu_1, \lambda p, \mu_3, \mu_4, \mu_2/(\mu_2 + \lambda(1-p))]$,
  i.e., $\lambda p$ and $\mu_2$ interchanged to overflow node 2
- $\tilde{\mathbf{\Theta}}_3 = [\mu_3 + \lambda p, \mu_1, \mu_2, \lambda(1-p), \mu_4, \lambda p/(\lambda p + \mu_3)]$,
  i.e., $\lambda(1-p)$ and $\mu_3$ interchanged to overflow node 3
- $\tilde{\mathbf{\Theta}}_4 = [\mu_4, \mu_1, \mu_2, \mu_3, \lambda, p]$,
  i.e., $\lambda$ and $\mu_4$ interchanged to overflow node 4.

We also need to identify an asymptotically efficient change of measure which simultaneously overflows nodes 2 and 3, i.e., the parallel section in the feed-forward network of Figure 5.1. This state-dependent change of measure was defined in Section 4.3. For each node it is a combination of no change of measure and the PW change of measure, depending on the number of customers at that node. To make the vector notation similar to the one used above we leave out the dependence part (we include it into our formula for feed-forward network) and denote by $\tilde{\mathbf{\Theta}}_{23}$ the vector that simultaneously overflows node 2 and 3 (cf. (4.3)–(4.4) with $x_i \geq b_i$), i.e.,

- $\tilde{\mathbf{\Theta}}_{23} = [\mu_2 + \mu_3, \mu_1, \lambda p, \lambda(1-p), \mu_4, \mu_2/(\mu_2+\mu_3)]\,.$

With the preceding definitions, a state-dependent change of measure for the feed-forward network in Figure 5.1 can now be given.

### SDH for a feed-forward network

Let $\tilde{\mathbf{\Theta}}(\mathbf{x}) = \left[\tilde{\lambda}, \tilde{\mu}_1, \tilde{\mu}_2, \tilde{\mu}_3, \tilde{\mu}_4, \tilde{p}\right]$ be a vector with the corresponding arrival and service rates at the respective nodes as well as the routing probability under the new change of measure to simulate network population overflow. Then,

$$\begin{aligned}\tilde{\mathbf{\Theta}}(\mathbf{x}) = \left[\frac{x_4}{b_4}\right]^1 \tilde{\mathbf{\Theta}}_4 + \left[\frac{b_4 - x_4}{b_4}\right]^+ \cdot \Bigg( & \left[\frac{x_2}{b_2}\right]^1 \cdot \left[\frac{x_3}{b_3}\right]^1 \tilde{\mathbf{\Theta}}_{23} + \\ & + \left[\frac{x_2}{b_2}\right]^1 \cdot \left[\frac{b_3 - x_3}{b_3}\right]^+ \tilde{\mathbf{\Theta}}_2 + \left[\frac{b_2 - x_2}{b_2}\right]^+ \cdot \left[\frac{x_3}{b_3}\right]^1 \tilde{\mathbf{\Theta}}_3 + \\ & + \left[\frac{b_2 - x_2}{b_2}\right]^+ \cdot \left[\frac{b_3 - x_3}{b_3}\right]^+ \cdot \left\{ \left[\frac{x_1}{b_1}\right]^1 \tilde{\mathbf{\Theta}}_1 + \left[\frac{b_1 - x_1}{b_1}\right]^+ \mathbf{\Theta} \right\} \Bigg),\end{aligned} \tag{5.18}$$

$$\tilde{\mu}_4(0,0,0,1) = 0. \tag{5.19}$$

Note that all vectors on the right side of (5.18) are state-independent. The only dependence is on a state (0,0,0,1) (Eq. (5.19)), which guarantees that all cycles hit the overflow level $N$ during the simulation. However, $\tilde{\mathbf{\Theta}}(\mathbf{x})$, and so the new parameters to simulate the network under importance sampling ($\tilde{\lambda}(\mathbf{x})$, $\tilde{\mu}_i(\mathbf{x})$, ($i = 1,2,3,4$), and $\tilde{p}(\mathbf{x})$) are state-dependent. Moreover, the equality $\sum_{i=1}^{n} (\tilde{\lambda}_i(\mathbf{x}) + \tilde{\mu}_i(\mathbf{x})) = 1$ still holds under the above change of measure (except for the state (0,0,0,1) where they still need to be normalized).

The above change of measure (5.18) is constructed as follows. In the feed-forward network in Figure 5.1 we consider parallel nodes in the middle (node 2 and 3) as one "big node", called node 2-3 (see Figure 5.2). By doing so, our feed-forward network becomes a 3-node tandem network (with nodes 1, 2-3 and 4), for which we use the change of measure from Appendix A (see Remark 5.3.1). This change of measure "pushes" each node when there are enough customers in it. Now we only need to define how to "push" node 2-3. Since it was formed from nodes 2 and 3, two queues in parallel, we can use our heuristic for queues in parallel. The only difference is that for queues in parallel in case when all nodes are empty we use no change of measure. In the current case of feed-forward network we have another node (node 1) in front of nodes 2 and 3 (or, node 2-3), and we need to "push" it when both nodes 2 and 3 are empty.

Thus, we "push" node 4 if it has enough customers ($x_4 \geq b_4$). Otherwise, if nodes 2 and 3 have enough customers, we "push" them together ($\tilde{\mathbf{\Theta}}_{23}$), or, "push" node 2 (respectively, node 3) if it has enough customers, i.e., $x_2 \geq b_2$ (respectively, $x_3 \geq b_3$). If both nodes 2 and 3 are empty, we "push" node 1 or use no change of measure depending on $x_1$.

According to the above change of measure, all nodes are overloaded simultaneously, depending on their buffer contents. Since $\rho_1 \leq \rho_2 \leq \rho_3 \leq \rho_4$, dependence on $x_4$ supersedes dependence on $x_3$ and $x_2$ which supersedes dependence on $x_1$.

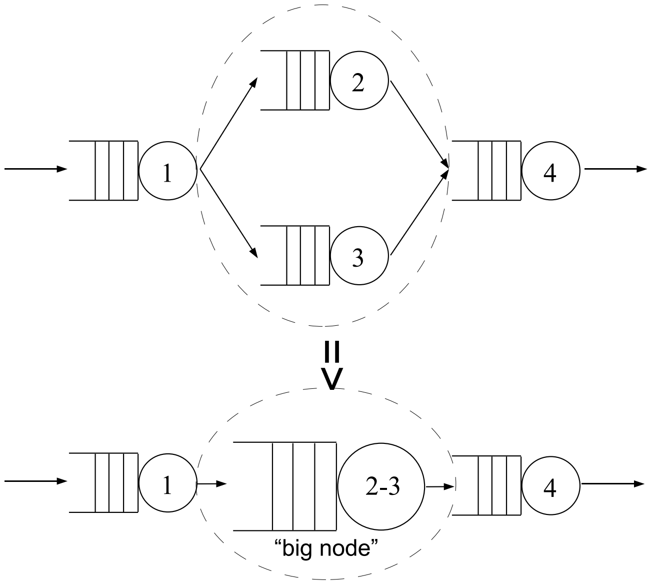


Figure 5.2: Defining a "big node"

The above change of measure implies that, upon arrival at an empty network, node 1 is gradually overloaded according to $[x_1/b_1]^1\,\tilde{\boldsymbol{\Theta}}_1$, i.e., gradually interchange $\lambda$ and $\mu_1$ depending on $x_1$. As $x_2$ (resp. $x_3$) increases, node 1 is gradually downloaded while node 2 (resp. node 3) is gradually overloaded according to $[x_2/b_2]^1\,\tilde{\boldsymbol{\Theta}}_2$ (resp. $[x_3/b_3]^1\,\tilde{\boldsymbol{\Theta}}_3$). As $x_4$ increases, node 2 (resp. node 3) is gradually downloaded while node 4 is gradually overloaded according to $[x_4/b_4]^1\,\tilde{\boldsymbol{\Theta}}_4$.

The effectiveness of the heuristic depends on the choice of the variables $b_i$. In general, $b_i$ can be different for each node $i$. However, as we show in the simulation experiments (Section 5.4), if we set all $b_i$'s in the heuristic equal to some $b$, the resulting estimates, corresponding to the best $b$, seem quite stable, thus suggesting robustness with respect to the choice of $b_i$'s. This appears to imply that, either the heuristic is sensitive to only one $b_i$, and robust with respect to the other parameters, or, the optimal values of $b_i$ are equal.

## 5.3.2 SDH for a feedback network

To describe a state-dependent heuristic for feedback Jackson networks, we use the specific example depicted on Figure 5.3 (a similar feedback network was considered in [46]). Without loss of generality we assume that $\sum_{i=1}^{2}(\lambda_i+\mu_i)=1$. The traffic intensity at node $i$ is $\rho_i=\gamma_i/\mu_i$, where $\gamma_i$ is the total arrival rate at node $i$ ($i=1,2$). We also assume that $\rho_1\le\rho_2$.

To construct our heuristic for a feedback network we use similar logic as for a tandem network, i.e., we "push" each node depending on the number of customers in it. The more customers the node has, the more we "push" it. Again, the bottleneck node

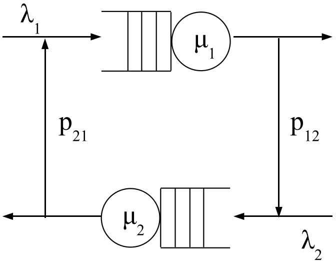


Figure 5.3: 2-node feedback network

is considered "more important", so the dependence on the number of customers in it supersedes the dependence on the number of customers in the non-bottleneck node. Each node is "pushed" according to the change of measure described in Section 5.2 (and [22]).

Let us describe the heuristic formally. Let $\mathbf{\Theta} = [\lambda_1, \lambda_2, \mu_1, \mu_2, p_{12}, p_{21}]$ be a vector with the external arrival rates, the service rates at nodes 1 and 2, and the routing probabilities, respectively, in the original network. For the feedback network in Figure 5.3, denote by $\tilde{\mathbf{\Theta}}_i$ the (asymptotically efficient) state-independent change of measure to simulate buffer overflow at node $i$ $(i = 1, 2)$, as determined from Section 5.2. Thus,

- $\tilde{\mathbf{\Theta}}_1 = \left[\tilde{\lambda}_1^{[1]}, \tilde{\lambda}_2^{[1]}, \tilde{\mu}_1^{[1]}, \tilde{\mu}_2^{[1]}, \tilde{p}_{12}^{[1]}, \tilde{p}_{21}^{[1]}\right]$,
- $\tilde{\mathbf{\Theta}}_2 = \left[\tilde{\lambda}_1^{[2]}, \tilde{\lambda}_2^{[2]}, \tilde{\mu}_1^{[2]}, \tilde{\mu}_2^{[2]}, \tilde{p}_{12}^{[2]}, \tilde{p}_{21}^{[2]}\right]$,

where $\tilde{\lambda}_i^{[k]}$, $\tilde{\mu}_i^{[k]}$, $\tilde{p}_{ij}^{[k]}$ $(i, j = 1, 2)$ are defined from Equations (5.12)–(5.17) and $k$ $(k = 1, 2)$ is a target node.

Let $\tilde{\mathbf{\Theta}}(\mathbf{x}) = \left[\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1, \tilde{\mu}_2, \tilde{p}_{12}, \tilde{p}_{21}\right]$ be a vector with the corresponding network parameters under SDH to simulate population overflow in the feedback network depicted on Figure 5.3; it is given in the following heuristic.

**The heuristic for a feedback network**

$$\tilde{\mathbf{\Theta}}(\mathbf{x}) = \left[\frac{x_2}{b_2}\right]^1 \tilde{\mathbf{\Theta}}_2 + \left[\frac{b_2 - x_2}{b_2}\right]^+ \left\{ \left[\frac{x_1}{b_1}\right]^1 \tilde{\mathbf{\Theta}}_1 + \left[\frac{b_1 - x_1}{b_1}\right]^+ \mathbf{\Theta} \right\}. \tag{5.20}$$

$$\tilde{p}_{10}(1, 0) = 0, \tag{5.21}$$

$$\tilde{p}_{12}(1, 0) = 1, \tag{5.22}$$

$$\tilde{p}_{20}(0, 1) = 0, \tag{5.23}$$

$$\tilde{p}_{21}(0, 1) = 1. \tag{5.24}$$

The new network parameters $(\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\mu}_1, \tilde{\mu}_2, \tilde{p}_{12}, \tilde{p}_{21})$ to be used in importance sampling are state-dependent. Moreover, the equality $\sum_{i=1}^{n} (\tilde{\lambda}_i(\mathbf{x}) + \tilde{\mu}_i(\mathbf{x})) = 1$ still holds under the above change of measure.

In Equation (5.20) the probability measures $\tilde{\mathbf{\Theta}}_i$ $(i = 1, 2)$ are state-independent and nested in the order of traffic intensities at the respective nodes. Since $\rho_2 > \rho_1$, $\tilde{\mathbf{\Theta}}_2$ is nested at the highest level, followed by $\tilde{\mathbf{\Theta}}_1$, then $\mathbf{\Theta}$ (no change of measure). The nesting of $\tilde{\mathbf{\Theta}}_2$ and $\tilde{\mathbf{\Theta}}_1$ would be reversed if node 1 were the bottleneck. Descriptively: if $x_2 \geq b_2$, then only node 2 is fully overloaded. Otherwise, if $x_1 \geq b_1$, then both node 1 and node 2 are overloaded depending on $x_2$. When both $x_1 < b_1$ and $x_2 < b_2$, $\tilde{\mathbf{\Theta}}(\mathbf{x})$ is a combination, with corresponding coefficients, of two changes of measure ($\tilde{\mathbf{\Theta}}_1$ and $\tilde{\mathbf{\Theta}}_2$) and no change of measure ($\mathbf{\Theta}$). Thus, starting from an empty network, as $x_1$ increases, we gradually overload node 1. As $x_2$ increases, we gradually and proportionately unload node 1 while overloading node 2. When the number of customers at node 2 is sufficiently large ($x_2 \geq b$), only node 2 is overloaded. Equations (5.21)–(5.24) guarantee that all cycles reach the overflow level $N$ during the simulation.

Here again, the effectiveness of the heuristic is influenced by the dependence ranges, $b_1$ and $b_2$, which must be set appropriately. Experimental results in Section 5.4.2 with $b_1 = 1$ suggest robustness with respect to $b_2$.

**Remark 5.3.2.** For the simple 2-node feedback network in Figure 5.3, the proposed change of measure appears to be very effective (see Section 5.4.2). However, a straightforward generalization of it for larger feedback networks (with more than 2 nodes) has not been tried and, thus, can not be guaranteed.

### 5.3.3 Possible generalization

Now we will describe how the heuristic proposed in Section 5.3.1 can be generalized for some types (see description below) of feed-forward networks. *A feed-forward network* is a queuing network when indices of queues can be chosen such that $r_{ij} = 0$ if $j \leq i$, i.e., node with lower index "feed" nodes with higher index (where $r_{ij}$ are the elements of the matrix $R = (I - P)^{-1}$).

To describe the heuristic we need to formalize the definition of a "big node" used in Section 5.3.1. Let $\mathcal{H}$ be the set of all nodes in the network. Suppose that all nodes $i \in \mathcal{H}$ are indexed such that $r_{ij} = 0$ if $j \leq i$. Let $\mathcal{F}_j$ denote the set of all direct feeder nodes of node $j$ $(j = 1, .., d)$, i.e., $p_{ij} > 0$ for all $i \in \mathcal{F}_j$. Suppose that for all $l, m \in \mathcal{F}_j$ the probability $p_{lm} = 0$, i.e., the direct feeder nodes of node $j$ "feed" only the node $j$ and not each other. Then, the subnetwork $\mathcal{B} \subset \mathcal{H}$ of a feed-forward network is called a *"big node"* if it is either a tandem network or a network of nodes in parallel such that 1) no nodes can be added to keep this property, i.e., if $\mathcal{B}$ is a network of nodes in tandem (resp., parallel) then for all $i \in \mathcal{H} - \mathcal{B}$ the networks of nodes $i \cup \mathcal{B}$ is not a tandem (resp., parallel) network, 2) if $\mathcal{B}$ is a network of nodes in parallel, then there exists $i \in \mathcal{B}$ and $m \in \mathcal{F}_i$ such that $m \in \mathcal{F}_j$ for all $j \in \mathcal{B}$, i.e., there exists at least one node that feeds all nodes in $\mathcal{B}$.

**Proposition 13.** ***(Sufficient condition for generalization)***
*If the feed-forward network is constructed in such a way that after defining all "big nodes" the network becomes a network of nodes in tandem or nodes in parallel than the heuristic can be generalized.*

The generalization is done recursively as follows. For the final network in which each node is either a single node or a "big node", we apply the corresponding heuristic for nodes in tandem or parallel (Appendix A, or Section 4.3). Then, we unfold each "big

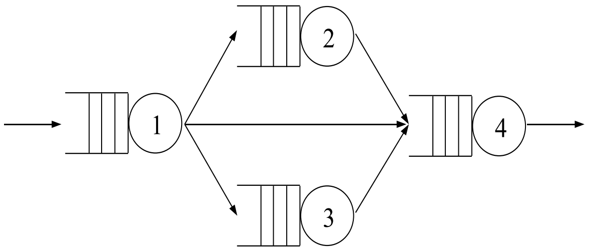


Figure 5.4: An example of a network not satisfying Proposition 13

node" by applying the corresponding heuristic until all nodes become single nodes. In the end we get a change of measure that is used to estimate the overflow probability in the feed-forward network.

Note, that the generalization described above does not cover all possible types of feed-forward networks. For example, the type of network depicted in Figure 5.4 is not included, since node 1 feeds not only its direct nodes (2 and 3) but also a node after them (node 4), so we can not separate a subnetwork of nodes in tandem or parallel.

We also do not have theoretical or empirical proof of its asymptotic efficiency since it was tested only on small networks of up to four nodes. However, for all types that were tested it worked very well (see experimental results below for some examples).

## 5.4 Experimental results

In this section we compare the performance of our heuristics with the PW and SDA algorithms (Sections 5.4.1–5.4.2) for feed-forward and feedback networks, respectively.

### 5.4.1 Performance for a feed-forward network

In this section we present experimental results performed on the feed-forward network depicted on Figure 5.1. All the experiments were made with $10^6$ replications. We consider four sets of feasible network parameters in the NAE region (this is verified empirically by showing that the PW heuristic yields wrong or unstable estimates). For consistency with the assumption made in Section 5.3.1, in the following experiments we also choose the network parameters such that $\rho_1 \leq \rho_2 \leq \rho_3 \leq \rho_4$.

For simplicity, in the following experiments, we set all $b_i$'s equal to some $b$ (see Section 5.5.1 for more discussion). Each estimate displayed in Tables 5.1–5.4 is obtained with the corresponding best setting of $b$ (which, of course, may be different for SDH and SDA).

The network parameters were chosen such that there are two cases where all loads are different (Table 5.1–5.2), one case with loads in the middle equal (Table 5.3) and one case with one of the loads in the middle equal to the bottleneck node (Table 5.4).

Experimental results in tables show that SDH (as described in Section 5.3.1) works very well and yields stable estimates with small, and bounded relative errors. Correctness is verified by agreement with SDA estimates, which are also accurate with small and less than linearly growing relative errors.

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | * | 3.9476e-07 ± 2.05 | 3 | 4.0064e-07 ± 0.07 | 6 | 4.0075e-07 ± 0.25 | 2.86 |
| 50 | * | 1.2825e-14 ± 3.44 | 4 | 1.3298e-14 ± 0.05 | 5 | 1.3270e-14 ± 0.24 | 4.65 |
| 100 | * | 1.1920e-29 ± 3.50 | 4 | 1.2568e-29 ± 0.06 | 6 | 1.2521e-29 ± 0.25 | 2.08 |

Table 5.1: 4-node feedforward network ($\lambda = 0.0455$, $\mu_1 = 0.7272$, $\mu_2 = 0.0455$, $\mu_3 = 0.0909$, $\mu_4 = 0.0909$, $p = 0.1$) ($\rho_1 = 0.06$, $\rho_2 = 0.1$, $\rho_3 = 0.45$, $\rho_4 = 0.5$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | * | 2.0830e-08 ± 5.89 | 3 | 1.9674e-08 ± 0.03 | 6 | 1.9671e-08 ± 0.22 | 1.33 |
| 50 | * | 5.2291e-17 ± 0.73 | 4 | 5.2223e-17 ± 0.02 | 6 | 5.2188e-17 ± 0.23 | 0.94 |
| 100 | * | 3.6552e-34 ± 0.67 | 4 | 3.6807e-34 ± 0.02 | 5 | 3.6846e-34 ± 0.24 | 0.43 |

Table 5.2: 4-node feedforward network ($\lambda = 0.064$, $\mu_1 = 0.564$, $\mu_2 = 0.039$, $\mu_3 = 0.192$, $\mu_4 = 0.141$, $p = 0.1$) ($\rho_1 = 0.11$, $\rho_2 = 0.16$, $\rho_3 = 0.3$, $\rho_4 = 0.45$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | * | 1.9267e-07 ± 3.75 | 5 | 2.1408e-07 ± 0.04 | 10 | 2.1485e-07 ± 0.50 | 2.18 |
| 50 | * | 3.9177e-15 ± 2.82 | 4 | 4.4567e-15 ± 0.07 | 10 | 4.4481e-15 ± 0.51 | 2.03 |
| 100 | * | 1.9168e-30 ± 12.0 | 4 | 1.9285e-30 ± 0.10 | 10 | 1.9341e-30 ± 0.69 | 0.51 |

Table 5.3: 4-node feedforward network ($\lambda = 0.069,\ \mu_1 = 0.571,\ \mu_2 = 0.022,\ \mu_3 = 0.198,\ \mu_4 = 0.14,\ p = 0.1$) ($\rho_1 = 0.12,\ \rho_2 = \rho_3 = 0.31,\ \rho_4 = 0.49$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | * | 1.6065e-06 ± 8.08 | 4 | 1.6963e-06 ± 0.11 | 9 | 1.7121e-06 ± 0.41 | 10.2 |
| 50 | * | 5.7736e-14 ± 31.6 | 4 | 7.7208e-14 ± 0.40 | 9 | 7.7058e-14 ± 0.39 | 137. |
| 100 | * | 2.2486e-29 ± 14.9 | 4 | 7.2017e-29 ± 0.56 | 10 | 7.1816e-29 ± 0.51 | 105. |

Table 5.4: 4-node feedforward network ($\lambda = 0.074,\ \mu_1 = 0.617,\ \mu_2 = 0.024,\ \mu_3 = 0.135,\ \mu_4 = 0.15,\ p = 0.1$) ($\rho_1 = 0.12,\ \rho_2 = 0.31,\ \rho_3 = \rho_4 = 0.49$)

Estimates using PW do not always agree with those using SDH and SDA. In fact, for an increasing number of replications (beyond $10^6$ used in all simulation runs), PW estimates eventually exhibit unstable behavior indicating a large variance.

*VRR* ratios of the SDA to SDH algorithms are mostly higher than one, implying that SDH is more efficient than SDA. Note, however, that we needed to use some changes in our procedure of comparing the SDA and SDH algorithms (cf. Section 3.4.2). Namely, for levels $N = 50$ and $N = 100$ we used the PW change of measure as a starting point for the SDA algorithm (it helped to speed up the convergence). We also increased the number of replications for the first iterations of the SDA algorithm, namely, $5 \cdot 10^5$ instead of $10^5$ as this was done previously (cf. Section 3.4.2), otherwise, the SDA algorithm did not converge. Only for $N = 100$ in Table 5.3 SDA converged already with $10^5$ replications and gave better results than with $5 \cdot 10^5$ replications, so we used that. This also might explain why *VRR* dropped to less than one in this case.

In Table 5.2 *VRR* for $N = 50$ and $N = 100$ is also less than one, meaning that SDA is better for this case than SDH. Note, however, that this case is also a bit different, namely, SDA showed very good convergence (it converged already with $10^5$ replications and without using PW as starting point).

On the contrary, in Table 5.4 *VRR* for $N = 50$ and $N = 100$ increased by order 10 compared with $N = 25$ making the SDH algorithm for this example far more efficient than the SDA algorithm. Since *RE*'s for the SDH and SDA algorithms for levels $N = 50$ and $N = 100$ are very close, this bad performance of SDA can be a result of its convergence problems. Some smoothing techniques which are known to help in other cases might help here (for more information about smoothing techniques see [32] and [14]). This shows, however, how much the performance of SDA algorithm depends on its convergence properties.

In the end, one can conclude that if the SDA algorithm shows signs of good convergence, it might work better than SDH. Otherwise (in most of the cases), the SDH algorithm is (sometimes by far) more efficient.

### 5.4.2 Performance for a feedback network

In this section we present experimental results performed on the feedback network depicted in Figure 5.3. As before, for all the experiments we used the same number of replications, namely, $10^6$. The PW heuristic in [15] may be asymptotically efficient (AE) only in some regions of the feasible parameter space; it is not asymptotically efficient (NAE) in other regions. These AE/NAE regions have not yet been formally characterized. However, the asymptotic efficiency of PW can be empirically tested for any arbitrary point in the parameter space. For the same network in Figure 5.3, in [46] a small region of the feasible parameter space is identified as NAE, i.e., a subset in which PW is provably asymptotically inefficient.

In the following experiments, we consider two sets of feasible network parameters. One of them is in the provably NAE region (Table 5.5), another one is in experimentally (not proven) NAE region (Table 5.6). To be consistent with the assumption made in Section 5.3.2, we also choose the network parameters such that $\rho_1 \leq \rho_2$. For SDH, we set $b_1 = 1$ and $b_2 = b$. Each SDH and SDA estimate displayed in the tables is obtained with the corresponding best setting of $b$ (which, of course, may be different for SDH and SDA).

Two types of network parameters were chosen: a network with low loads (Table 5.5) and a network with high loads (Tables 5.6). Experimental results in Tables 5.5 and 5.6 show that whereas PW gives incorrect estimates, SDH yields correct and asymptotically efficient estimates with bounded relative error.

Note, that SDH is more efficient for one case (Table 5.6, $VRR > 1$) and less efficient for another (Table 5.5, $VRR < 1$). It can be explained by very good convergence of SDA for the network in Table 5.5, so the $RE$'s are about 20 times as small as for SDH. However, for another example (Table 5.6), despite the fact that $RE$ of SDA for level $N = 25$ is smaller than $RE$ of SDH, the variance reduction ratio $VRR > 1$, which shows that SDH is more efficient than SDA. The efficiency gain also grows with the overflow level $N$, since $RE$ of SDA grows linearly with $N$ and $RE$ of SDH stays bounded. Thus, it is impossible to predict which algorithm (SDH or SDA) will work better, all depends on the specific parameter settings and convergence of SDA. If it converges good it might work better than SDH, otherwise, SDH is better.

### Sensitivity with respect to $b$

In Table 5.6 the results for different values of $b$ are presented to show the sensitivity of SDH for a feedback network with respect to $b$. For different values of $b$ around the "best" (marked by an '*' in the table), we display the resulting estimate along with its relative error, estimated from simulation and computed numerically (shown between parentheses, where the best numerically computed $RE$ is marked with '*'). The numerical $RE$ is obtained from an algorithm similar to that outlined in [25] but adapted to compute the variance of the SDH estimator for the above feedback network example. The empirical relative error is consistent with the relative error calculated numerically.

It is interesting to note that the accuracy of the simulation estimates is not too sensitive with respect to $b$. This can also be concluded from the computed relative errors. Moreover, both empirical and numerical results suggest bounded relative error of the SDH estimator.

The feedback network example considered is relatively small, yet it helps to illustrate that our approach may indeed be useful where no other heuristics are known to be effective. However, the heuristic has not been checked extensively for different parameters settings, hence, its effectiveness can not be guaranteed for other network parameters.

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b_2 = b$ | $\tilde{\gamma}(N) \pm RE\%$ | $VRR$ |
| 25 | 7.9508e-021 | 2.4772e-020 ± 77.4 | 5 | 7.8930e-21 ± 0.02 | 4 | 7.9670e-21 ± 0.26 | 0.33 |
| 50 | 1.3885e-042 | 8.7117e-043 ± 7.25 | 5 | 1.3883e-42 ± 0.01 | 5 | 1.3928e-42 ± 0.25 | 0.09 |
| 100 | 3.9811e-086 | 2.3596e-086 ± 2.60 | 5 | 3.9535e-86 ± 0.01 | 4 | 3.9859e-86 ± 0.27 | 0.07 |

Table 5.5: 2-node feedback network ($\lambda_1 = 0.01,\ \mu_1 = 0.13,\ \lambda_2 = 0.09,\ \mu_2 = 0.77,\ p_{12} = 0.9,\ p_{21} = 0.05$) ($\rho_1 = 0.117,\ \rho_2 = 0.135$)

| **N** | **Numerical** | **PW** | **SDA** | | **SDH** | | |
|---|---|---|---|---|---|---|---|
| | $\gamma(N)$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b_2 = b$ | $\tilde{\gamma}(N) \pm RE\%$ (exact $RE\%$) | $VRR$ |
| 25 | 9.9890e-006 | 7.3235e-006 ± 9.21 | 5 | 9.9834e-06 ± 0.09 | 9 | 1.0005e-05 ± 0.28 (0.31) | 2.55 |
| | | | | | 10 | 1.0016e-05 ± 0.26 (0.27) | 2.71 |
| | | | | | 11* | 9.9321e-06 ± 0.25 (0.25)* | 2.75 |
| | | | | | 12 | 9.9452e-06 ± 0.25 (0.26) | 2.36 |
| | | | | | 13 | 9.9863e-06 ± 0.26 (0.27) | 2.02 |
| | | | | | 14 | 1.0011e-05 ± 0.28 (0.28) | 1.81 |
| 50 | 1.4634e-011 | 8.4150e-012 ± 10.4 | 5 | 1.4585e-11 ± 0.29 | 10 | 1.4589e-11 ± 0.32 (0.36) | 12.7 |
| | | | | | 11 | 1.4590e-11 ± 0.27 (0.29) | 16.7 |
| | | | | | 12* | 1.4655e-11 ± 0.26 (0.26) | 16.2 |
| | | | | | 13 | 1.4642e-11 ± 0.28 (0.26)* | 13.3 |
| | | | | | 14 | 1.4650e-11 ± 0.26 (0.27) | 13.5 |
| | | | | | 15 | 1.4595e-11 ± 0.28 (0.28) | 12.3 |
| 100 | 2.0500e-023 | 1.6348e-023 ± 33.3 | 5 | 2.0064e-23 ± 0.57 | 11 | 2.0416e-23 ± 0.30 (0.36) | 23.8 |
| | | | | | 12 | 2.0480e-23 ± 0.29 (0.29) | 27.3 |
| | | | | | 13* | 2.0594e-23 ± 0.28 (0.27)* | 27.8 |
| | | | | | 14 | 2.0550e-23 ± 0.28 (0.28) | 24.2 |
| | | | | | 15 | 2.0437e-23 ± 0.29 (0.29) | 21.4 |
| | | | | | 16 | 2.0555e-23 ± 0.30 (0.31) | 20.0 |

Table 5.6: 2-node feedback network ($\lambda_1 = 0.06,\ \mu_1 = 0.14,\ \lambda_2 = 0.25,\ \mu_2 = 0.55,\ p_{12} = 0.9,\ p_{21} = 0.05$) ($\rho_1 = 0.542,\ \rho_2 = 0.579$) ('*' denotes $b_{opt}$ or the minimum of exact $RE$)

## 5.5 Extensive experimental results

In this section we check the heuristic for the feed-forward network with 4 nodes (as depicted in Figure 5.1) by experiments for many different parameter settings. We show that for most of the network parameters our heuristic gives bounded $RE$. We also show that there are, however, some cases for which the heuristic is not asymptotically efficient. This is a so called *completely symmetric* case, i.e., when loads at all network nodes are equal and outgoing probabilities from node 1 to node 2 and from node 1 to node 3 are the same (i.e., $\rho_i = \rho$ for $i = 1, ..4$, $p = 0.5$).

The statistics were gathered for three different levels, namely, $N = 25$, 50 and 100 with $10^6$, $4 \cdot 10^6$, $16 \cdot 10^6$ and $64 \cdot 10^6$ replications (similar to what was done for queues in tandem and in parallel, Sections 3.6 and 4.4.3). There were about 130 different settings of network parameters checked. Around 30 points for a completely symmetric case (as a most difficult one) and around 100 points randomly chosen (according to uniform distribution) from all possible network parameters. Below we discuss the observations that were made based on the experimental results.

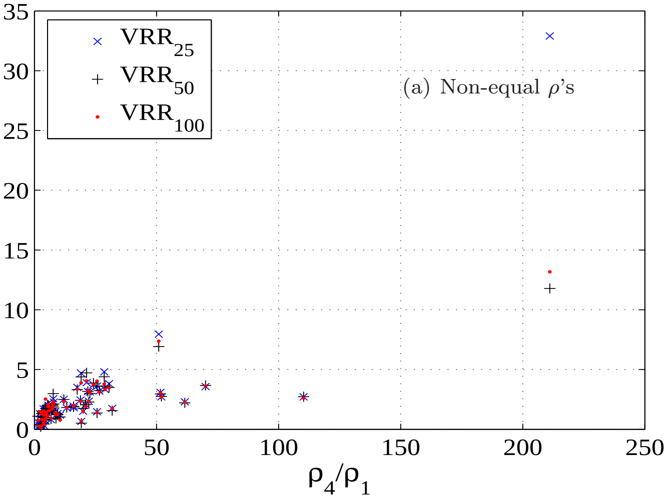
First, we talk about sensitivity of the heuristic with respect to $b_i$ (Section 5.5.1). Then, we discuss the behavior of relative error (Section 5.5.2) and dependency of $b_{opt}$ on the overflow level (Section 5.5.3). At the end we give a guideline for finding $b_{opt}$ (Section 5.5.4).

### 5.5.1 Sensitivity with respect to $b_i$

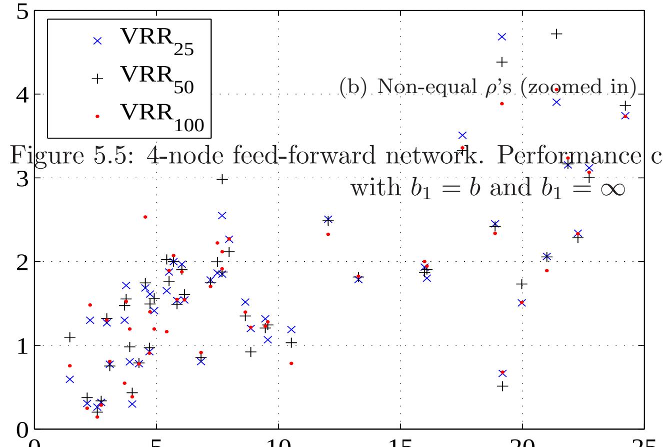
In general, the optimal values of $b_i$ can be different for all $i = 1, .., d$. Even if we assume that $b_i \leq L$ the number of all possibilities grows very quickly with the number of nodes in the network (it is of order $N^d$). In all our experiments we restrict ourselves to the case $b_i = b$ for all $i = 1, .., d$. We show that even with equal $b_i$'s the proposed heuristic gives reliable estimates. We also show that, in general, the estimation results can be improved if we allow $b_i$'s to be different.

Let us consider the case $b_1 = \infty$ (as an extreme example) and compare $RTV$'s (cf. (3.29)) of the heuristic with equal $b_i$'s ($b_i = b = b_{opt}$ for $i = 1, .., 4$) and $b_1 = \infty$, $b_i = b = b_{opt}$ for $i = 2, 3, 4$ (note, that $b_{opt}$ for cases $b_1 = \infty$ and $b_1 = b_i$ might be different). Since the simulation time was almost equal for both of the cases we calculate *VRR* (cf. 3.30) as a squared ratio of $RE$'s, i.e., *VRR*$=(RE_{b_1=b}/RE_{b_1=\infty})^2$. Figures 5.5a–b show *VRR* results. *VRR* values greater than one mean that the heuristic with $b_1 = \infty$ is better.

From the figures one can see that when the ratio $\rho_4/\rho_1 \geq 5$ the heuristic with $b_1 = \infty$ "works" better. That can be easily explained. The boundaries $b_i$ in the heuristic indicate how much the change of measure depends on the number of customers at node $i$. The value $b_i = 1$, for example, means that the heuristic is very dependent on the content of node $i$ (so we "push" it as soon as there is one customer there) and the value $b_1 = \infty$ means that we do not "push" this node (the heuristic does not depend on the number of customers there). Figures 5.5a–b show that as soon as the load at the bottleneck node (node 4) is five times larger than the load at node 1 ($\rho_4/\rho_1 \geq 5$) it is almost always more efficient not to "push" node 1, i.e., use $b_1 = \infty$.

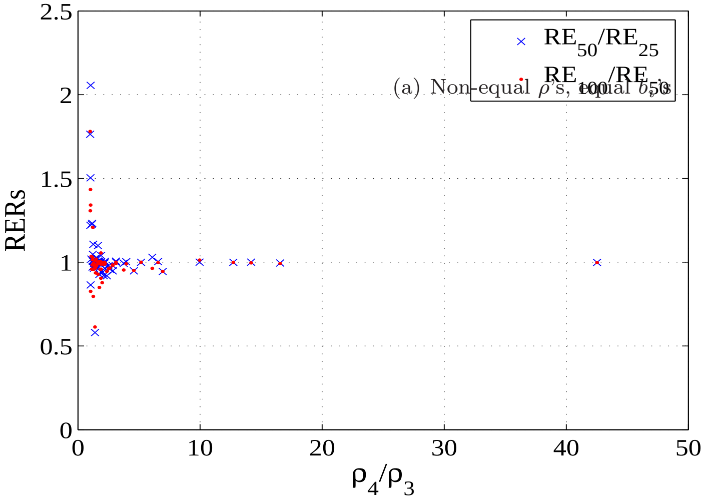


(a) Non-equal $\rho$'s

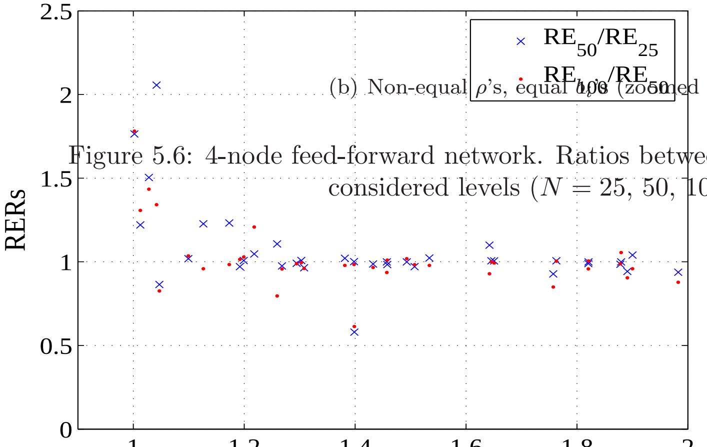


(b) Non-equal $\rho$'s (zoomed in)

Figure 5.5: 4-node feed-forward network. Performance comparison of the heuristic with $b_1 = b$ and $b_1 = \infty$

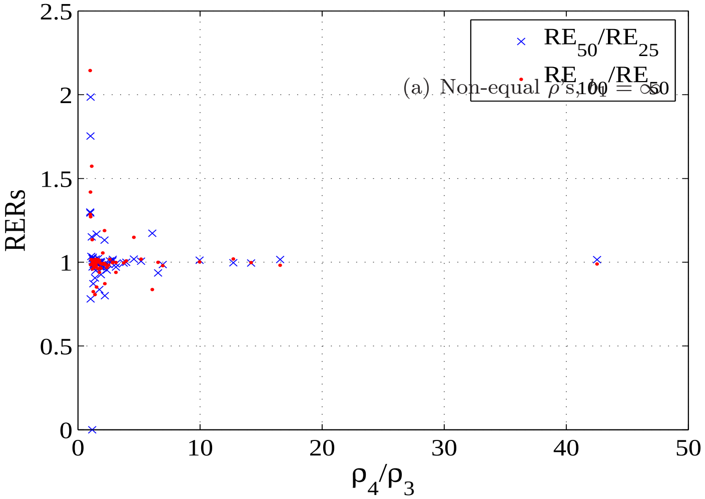


(a) Non-equal $\rho$'s, equal $b$'s

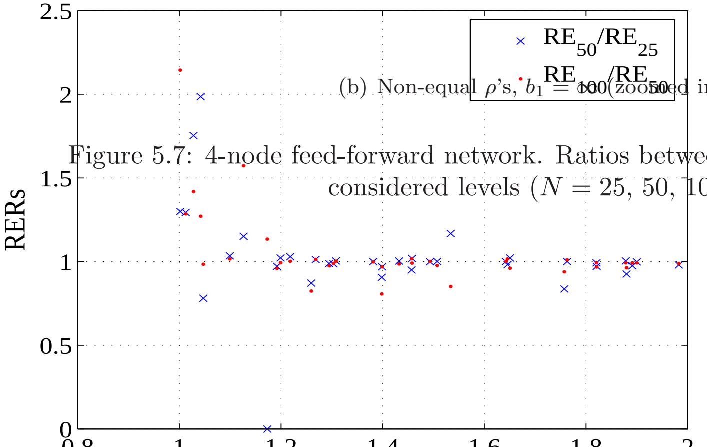


(b) Non-equal $\rho$'s, equal $b$'s (zoomed in)

Figure 5.6: 4-node feed-forward network. Ratios between $RE$s for consecutively considered levels ($N$ = 25, 50, 100)



(a) Non-equal $\rho$'s, $b_1 = \infty$



(b) Non-equal $\rho$'s, $b_1 = \infty$ (zoomed in)

Figure 5.7: 4-node feed-forward network. Ratios between $RE$s for consecutively considered levels ($N = 25, 50, 100$)

Despite this fact in all the experiments we set $b_i = b$ and try to find $b = b_{opt}$, the value $b$ that minimizes the variance of the estimator. This was done for simplicity and because the heuristic was working even with equal $b_i$'s. Moreover, it showed the same type of behavior (see Figures 5.6–5.7). All the conclusions made below correspond to the case $b_i = b_{opt}$ for $i = 1, .., 4$.

### 5.5.2 Behavior of relative error

**Proposition 14.** *For all network parameters of the 4-node feed-forward network depicted in Figure 5.1 satisfying the condition* $\rho_4 > \rho_3$, *the heuristic proposed in Section 5.3.1 gives estimates with bounded RE;*
*for network parameters with equal loads* ($\rho_1 = \rho_3 = \rho_3 = \rho_4$) *the proposed heuristic might give estimates with bounded or less than linearly growing RE; however, for most of the case it is not asymptotically efficient.*

To see that, let us consider the relative error ratios ($RER$s) between levels $N = 100$ and $N = 50$, and between levels $N = 50$ and $N = 25$, i.e., $RE_{100}/RE_{50}$ and $RE_{50}/RE_{25}$ (see similar discussion in Section 3.6.1, Proposition 4). These ratios should be near 1 for bounded $RE$. $RER$s that are less than 2 indicate that $RE$ grows less than linearly with level $N$. As one can see from Figures 5.6a-b the $RER$s are very close to one for all network parameters with $\rho_4/\rho_3 > 1$. In case $\rho_4/\rho_3 = 1$, i.e., when the load at the bottleneck node (the last node in the feed-forward network) is equal to the load at the second bottleneck node (one of the nodes in the middle), these ratios are closer to 2, suggesting linearly growing $RE$.

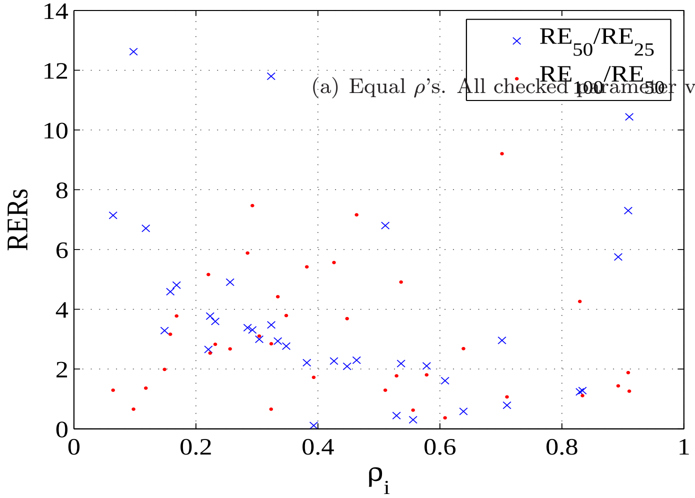
In Figures 5.8a–b the $RER$s for network parameters with equal loads are depicted. One can clearly see that the heuristic works not as well as for the non-symmetric case ($RER$s are very often greater than one). Nevertheless, there are network parameters for which the heuristic gives bounded $RE$ (3 out of 36 checked points), or less than linearly growing $RE$ (3 out of 36 checked points), see Figure 5.8b. Unfortunately, we were not able to specify exactly, or, numerically, the regions of bounded or linearly growing $RE$.

### 5.5.3 Dependence of $b_{opt}$ on the overflow level

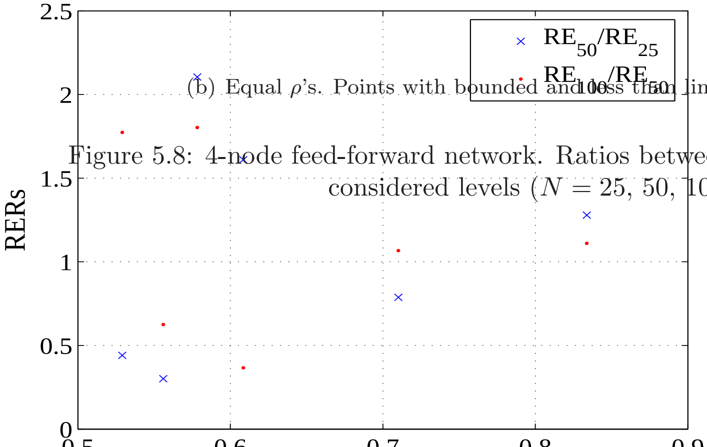
In this section we consider the dependence of $b_{opt}$ on the overflow level. We suppose that $b_{opt}$ is already known (the guideline to find $b_{opt}$ is described in the next Section, Proposition 16). The conclusions are made only for the case of non-equal loads, since according to Proposition 14 in this case the heuristic is (experimentally) asymptotically efficient.

**Proposition 15.** *Consider the 4-node feed-forward network depicted in Figure 5.1. Suppose that all network parameters satisfy the condition that at least one of the inequalities* $\rho_4 \geq \rho_3 \geq \rho_2 \geq \rho_1$ *is strict, i.e.,* $\rho_4 > \rho_1$ *(not all loads are equal). Then, if the last node is a strong bottleneck* ($\rho_4 \geq 10 \cdot \rho_3$) *then* $b_{opt}$ *does not depend on the overflow level* $N$*;*
*otherwise,* $b_{opt}$ *can change with the level* $N$.

To see this we consider the difference between values $b_{opt}$ for levels 25, 50 and 100, i.e., $b_{opt}(100) - b_{opt}(50)$ and $b_{opt}(50) - b_{opt}(25)$. From Figures 5.9a–b one can

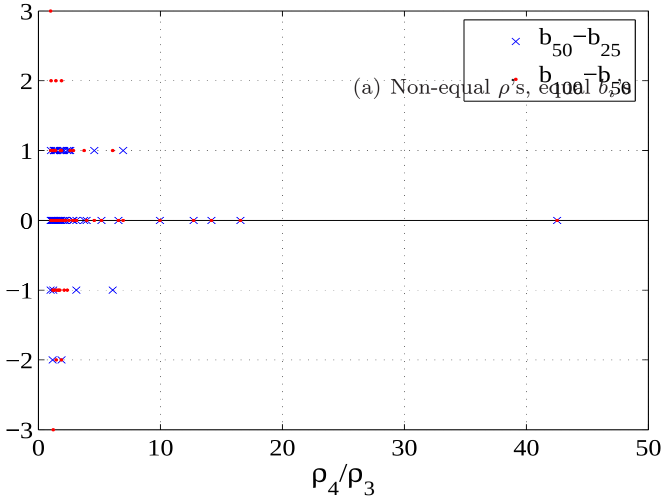


(a) Equal $\rho$'s. All checked parameter values

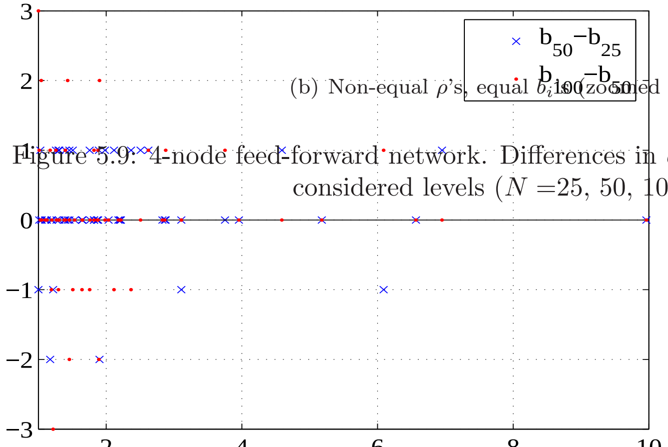


(b) Equal $\rho$'s. Points with bounded and less than linearlygrowing $RE$

Figure 5.8: 4-node feed-forward network. Ratios between $RE$s for consecutively considered levels ($N = 25, 50, 100$)

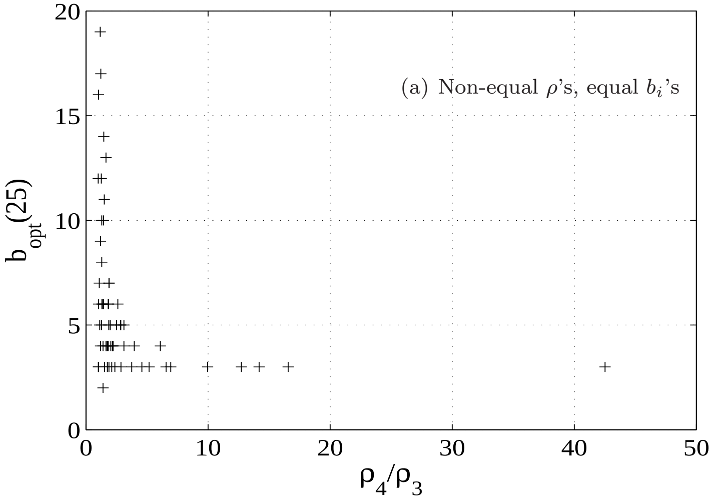


(a) Non-equal $\rho$'s, equal $b_i$'s



(b) Non-equal $\rho$'s, equal $b_i$'s (zoomed in)

Figure 5.9: 4-node feed-forward network. Differences in $b_{opt}$ between consecutively considered levels ($N$ =25, 50, 100)



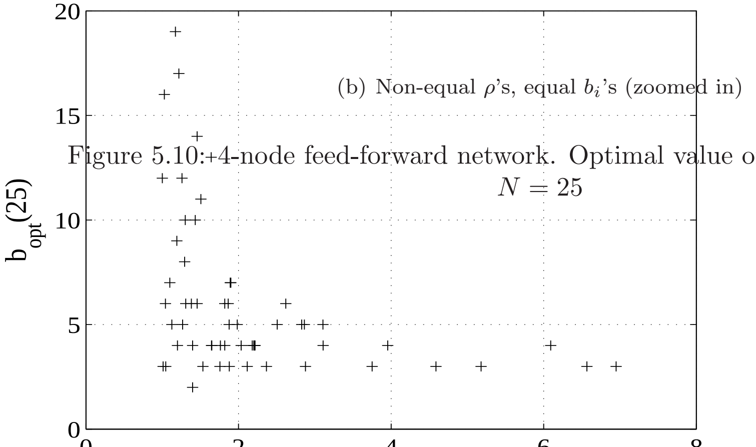


Figure 5.10: 4-node feed-forward network. Optimal value of parameter $b$ for level $N = 25$

see that these differences are equal to zero for $\rho_4/\rho_3 \geq 10$ meaning that $b_{opt}$ does not changes with level $N$. For $\rho_4/\rho_3 < 10$ these differences can be non-zero, i.e., $b_{opt}$ can change with level $N$. However, when $2 \leq \rho_4/\rho_3 < 10$ these differences are very small (between -1 and 1), i.e., $b_{opt}$ almost does not change with level $N$. This makes it easier to find $b_{opt}$ for higher levels (as $N = 50$ or 100) once we know $b_{opt}$ for some low level (as $N = 25$). One needs to check only three values: $b_{opt}(25) - 1$, $b_{opt}(25)$ and $b_{opt}(25) + 1$. Note, however, that we have not experimented with levels higher than $N = 100$, so these differences might be larger.

### 5.5.4 Guideline for finding $b_{opt}$

In this section we will discuss how to find $b_{opt}$ for level 25. Using this knowledge and Proposition 15 one can find $b_{opt}$ for higher levels.

**Proposition 16.** *Consider the feed-forward network depicted in Figure 5.1. If the last node is a strong bottleneck* ($\rho_4/\rho_3 \geq 10$) *then* $b_{opt}(25) = 3$*;*
*if* $\rho_4/\rho_3 \in [4, 10)$ *then* $b_{opt}(25) \in [3, 4]$*;*
*if* $\rho_4/\rho_3 \in [2, 4)$ *then* $b_{opt}(25) \in [2, 6]$*;*
*if* $\rho_4/\rho_3 \in [1, 2)$ *then* $b_{opt}(25) \geq 2$*;*

Figure 5.10 shows how $b_{25}$ depends on the network parameters. One can clearly see that the Proposition 16 is satisfied.

## 5.6 Conclusion

In this chapter we have proposed and experimented with the heuristic changes of measure to estimate the probability of population overflow in feed-forward and feedback networks. Extensive experimental results indicate that for most of the cases the heuristics yield asymptotically efficient estimates, with relative error growing at most linearly with the overflow level. The efficiency of the obtained changes of measure compares well with those determined using adaptive importance sampling methodologies: the heuristics are simpler to apply and in most of the cases are more efficient.

In this chapter a possible generalization of the heuristic for specific types of feed-forward networks was also proposed. It is still an open question, however, how it can be generalized to any Jackson network. That would be a very interesting topic for further research. Another interesting question is a formal proof of asymptotic efficiency. Also, a theoretical dependency of a number of boundary layers on the network parameters is another challenge that needs to be addressed.

# Chapter 6

# Exploring further

In the previous chapters we have proposed state-dependent heuristics to simulate population overflow in different types of Markovian queuing networks. For most of the cases we could conclude, based on extensive experimentation, that the heuristics were asymptotically efficient. Such a good performance, in return, triggers two questions. First, can the experimentally shown efficiency be theoretically proven? Second, can these heuristics be applied for non-Markovian networks?

In this chapter we address both of these issues. In Section 6.1 we discuss directions to prove asymptotic efficiency of the changes of measure presented in Chapter 3. In Section 6.2 we present extension of the heuristics proposed for a 2-node tandem Jackson queuing network (Chapter 3, Appendix A) to non-Markovian queuing networks.

## 6.1 A proof of asymptotic efficiency?

In this Section we look in more detail whether the proof approaches in [7] and [47] for different change of measure for a 2-node tandem network can be applied in our case. In Section 6.1.1 we discuss the main ideas from [7] and [47]. In Section 6.1.2 we show that the reverse version of the arguments in [7] can not be applied for our case. In Section 6.1.3 by falling short to use the approach from [47], some understanding in the behavior of the changes of measure is gained.

### 6.1.1 Related work

In a recent paper [7] a first attempt of constructing a provably asymptotically efficient change of measure for a 2-node tandem network has been presented. This change of measure (for a 2-node tandem Markovian queuing network) is constructed based on game theory results. Instead of minimizing the second moment (necessary for asymptotic efficiency), the authors propose to look at the problem of finding a change of measure as a stochastic control problem. Using the corresponding theory, they write down the Dynamic Programming Equation (DPE), the solution of which (through its gradient) defines the zero-variance (thus, asymptotically efficient) change of measure. The authors find the solution approximately and show that the approximation is good enough to give an asymptotically efficient change of measure. The proof, however, is very complicated and includes a lot of new notation. In [47] the same change of measure was looked at from a different point of view and a simpler proof was presented.

Below we discuss if the results in [7] and [47] can be used to prove asymptotic efficiency of our changes of measure from Sections 3.2.3–3.2.4.

### 6.1.2 First approach for a proof

We aim for proving asymptotic efficiency of our changes of measure for a 2-node tandem network (Chapter 3). Our first attempt is to use the same approach as in [7]. However, it is not clear whether in our case the same type of arguments can be applied. The problem is that in [7] the change of measure (at each state) is *constructed* from the gradient of an approximate solution, some function $W(x_1, x_2)$ on a state space, of the so-called DPE. Thus, it is *found.* In our case, we already *have* a change of measure for each state (constructed using some heuristic arguments) and we would like to show its asymptotic efficiency. We show below that such a function $W(x_1, x_2)$ can *not* be constructed, and, thus, this approach can not be used to prove asymptotic efficiency of our heuristics.

Let $(x_1, x_2)$ denote a system state and $(\lambda, \mu_1, \mu_2)$ denote, respectively, the arrival rate to node 1 and the service rates at nodes 1 and 2. Let $p = (p_1, p_2)$ be a vector in $\mathbb{R}$. In [7] the change of measure for estimating the total network overflow probability in a 2-node tandem queuing network is defined through some function $W(x_1, x_2)$ on the state space as follows.

$$\begin{cases} \tilde{\lambda} = & \lambda e^{-p_1/2} \cdot N(p), \\ \tilde{\mu}_1 = & \mu_1 e^{(p_1 - p_2)/2} \cdot N(p), \\ \tilde{\mu}_2 = & \mu_2 e^{p_2/2} \cdot N(p), \end{cases} \tag{6.1}$$

where

$$N(p) = \frac{1}{\lambda e^{-p_1/2} + \mu_1 e^{(p_1 - p_2)/2} + \mu_2 e^{p_2/2}}, \tag{6.2}$$

and $p = (p_1, p_2) = (\partial W / \partial x_1, \partial W / \partial x_2)$.

We will follow the construction in [7] in the other direction as follows. For our changes of measure we use Equation (6.1) (Proposition 3.4 in [7], or, Equation 7 in [47]) to find a vector $p$. Then, the problem is to find a function $W(x_1, x_2)$, such that its gradient is equal to $p$ and that satisfies the same conditions as mentioned in [7]. We show below that such a function does not exist.

Let us consider simplified versions, namely, without condition $\tilde{\mu}_2(0, 1) = 0$, of SDH and SDHI changes of measure for a 2-node tandem network (Sections 3.2.3–3.2.4). Then, using Equation (6.1) one can find that the corresponding derivatives look as follows. For the simplified SDH change of measure, we have

$$\frac{\partial W}{\partial x_1} = \frac{2}{3} \ln \left( \frac{\lambda^2}{\mu_1 \mu_2} \cdot \frac{\left(\mu_2 + (\lambda - \mu_2) \cdot \left[\frac{x_2}{b}\right]^+\right) \cdot \left(\lambda + (\mu_1 - \lambda) \cdot \left[\frac{x_2}{b}\right]^+\right)}{\left(\mu_1 + (\mu_2 - \mu_1) \cdot \left[\frac{x_2}{b}\right]^+\right)^2} \right), \tag{6.3}$$

$$\frac{\partial W}{\partial x_2} = \frac{2}{3} \ln \left( \frac{\lambda \mu_1}{{\mu_2}^2} \cdot \frac{\left(\mu_2 + (\lambda - \mu_2) \cdot \left[\frac{x_2}{b}\right]^+\right)^2}{\left(\lambda + (\mu_1 - \lambda) \cdot \left[\frac{x_2}{b}\right]^+\right) \cdot \left(\mu_1 + (\mu_2 - \mu_1) \cdot \left[\frac{x_2}{b}\right]^+\right)} \right), \tag{6.4}$$

and for the simplified SDHI change of measure, we have

$$\frac{\partial W}{\partial x_1} = \frac{2}{3} \ln \left( \frac{\lambda^2}{\mu_1 \mu_2} \cdot \frac{\left(\mu_1 + (\lambda - \mu_1) \cdot \left[\frac{x_2}{b}\right]^+\right) \cdot \left(\lambda + (\mu_1 - \lambda) \cdot \left[\frac{x_2}{b}\right]^+\right)}{\mu_2^2} \right), \tag{6.5}$$

$$\frac{\partial W}{\partial x_2} = \frac{2}{3} \ln \left( \frac{\lambda \mu_1}{{\mu_2}^2} \cdot \frac{\left(\mu_1 + (\lambda - \mu_1) \cdot \left[\frac{x_2}{b}\right]^+\right)^2}{\mu_2 \left(\lambda + (\mu_1 - \lambda) \cdot \left[\frac{x_2}{b}\right]^+\right)} \right). \tag{6.6}$$

Integrating (6.3) (respectively, (6.5) for the SDHI change of measure) with respect to $x_1$ we obtain

$$W(x_1, x_2) = \frac{\partial W}{\partial x_1} \cdot x_1 + C_2(x_2) = F_1(x_2) \cdot x_1 + C_2(x_2), \tag{6.7}$$

and integrating (6.4) (respectively, (6.6)) with respect to $x_2$ we obtain

$$W(x_1, x_2) = \int_0^N \frac{\partial W}{\partial x_2} dx_2 + C_1(x_1) = F_2(x_2) + C_1(x_1), \tag{6.8}$$

where the functions $C_1(x_1)$ and $C_2(x_2)$ are yet unknown. Thus, the following equality should be true for all $x_1, x_2 \geq 0$

$$F_1(x_2) \cdot x_1 + C_2(x_2) = F_2(x_2) + C_1(x_1). \tag{6.9}$$

This is possible only if $F_1(x_2) = const$ for all $x_2 \geq 0$. In our case $F_1(x_2) = const$ only for $x_2 = 0$ and $x_2 \geq b$, since $F_1(x_2) = \partial W / \partial x_1$ and it does depend on $x_2$ for $0 < x_2 < b$ (cf., (6.3), (6.5)).

Thus, a suitable function $W(x_1, x_2)$ can *not* be constructed (even for simplified versions of our changes of measure) and, hence, the approach to follow a reverse version of the argument in [7] can not be used to prove asymptotic efficiency of our heuristics.

### 6.1.3 Second approach for a proof

A second approach could be to follow the proof in [47], which is an alternative to the proof in [7] for the same change of measure for a 2-node tandem network. The authors of [47] looked at the likelihood ratio of a sample path under the new change of measure and showed that it depends only on the initial and final points and is “largely independent” of the exact shape of the path. This means, for example, that it is almost independent of cycles, and, hence, the likelihood ratio of any cycle is equal to one, or, is approximately equal to one.

Let us consider the behavior of likelihood ratios of SDH, SDHI (Sections 3.2.3–3.2.4) and PW [15] changes of measure (the latter does not have anything to do with

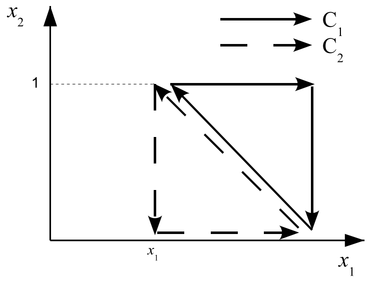


Figure 6.1: Cycles $C_1$ and $C_2$

our proof attempt and is here only to show the difference in behavior of likelihood ratios of "good working" (SDH and SDHI) and "not always working" (PW) changes of measure).

We want to see whether likelihood ratios of our changes of measure are nearly equal to one. If they are, then we can, otherwise, we can *not* use the approach in [47] to prove asymptotic efficiency of SDH and SDHI changes of measure.

Below we show that likelihood ratios of cycles for the SDH and SDHI changes of measure are not always equal to one and they do not go to one in the limit when overflow level $N \to \infty$. Hence, the approach in [47] can not be used to prove asymptotic efficiency of our changes of measure. The observations are, however, useful in understanding the efficiency of our changes of measure.

### Behavior of likelihood ratios for cycles

**Observation 1.** *For both the SDH and SDHI changes of measure, any cycle that goes only through states with at least b customers at node 2, i.e.,* $x_2 \geq b$*, has a likelihood ratio equal to 1.*

This is always true for PW change of measure for cycles that do not touch the boundary $x_2 = 0$, since the PW rates are state-independent and are equal to $\mu_2$, $\mu_1$ and $\lambda$ (respectively, for the arrival rate, the service rate at node 1 and the service rate at node 2). Any cycle of length $3 \cdot c$ has $c$ arrivals, $c$ departures from node 2 and $c$ departures from node 2. Hence,

$$LR^{PW} = \left( \frac{\lambda}{\mu_2} \frac{\mu_1}{\mu_1} \frac{\mu_2}{\lambda} \right)^c = 1. \tag{6.10}$$

Our SDH and SDHI changes of measure are equal to PW for $x_2 \geq b$. Thus, cycles that go only through states with $x_2 \geq b$ do not influence the likelihood ratio of a path.

**Observation 2.** *For the SDH change of measure, every cycle of length three that touches the boundary* $x_2 = 0$ *has likelihood ratio larger than 1.*

Let us consider cycles of length three. There are only two types of them, namely, cycle $C_1$: $(x_1, 1) \to (x_1 + 1, 1) \to (x_1 + 1, 0) \to (x_1, 1)$ and cycle $C_2$: $(x_1, 1) \to (x_1, 0) \to (x_1 + 1, 0) \to (x_1, 1)$ (Figure 6.1). The corresponding SDH likelihood ratios are equal to

$$LR_{C_1}^{SDH} = \frac{\lambda \mu_2 \frac{\mu_1}{\lambda+\mu_1}}{\left(\mu_1 + \frac{\mu_2-\mu_1}{b}\right)\left(\mu_2 + \frac{\lambda-\mu_2}{b}\right)\frac{\lambda}{\mu_1+\lambda}} = \frac{\mu_1\mu_2}{\left(\mu_1 + \frac{\mu_2-\mu_1}{b}\right)\left(\mu_2 + \frac{\lambda-\mu_2}{b}\right)} > 1, \tag{6.11}$$

and

$$LR_{C_2}^{SDH} = \frac{\mu_2 \frac{\lambda}{\lambda+\mu_1}\frac{\mu_1}{\lambda+\mu_1}}{\left(\mu_2 + \frac{\lambda-\mu_2}{b}\right)\frac{\mu_1}{\mu_1+\lambda}\frac{\lambda}{\mu_1+\lambda}} = \frac{\mu_2}{\mu_2 + \frac{\lambda-\mu_2}{b}} > 1. \tag{6.12}$$

Remember, that, as before, we consider only the case $\mu_2 \leq \mu_1$ (Remark 3.5.1).

The difference with the PW change of measure (which shows good performance only for some network parameters) is that for PW *every* cycle (not only cycles of length three) that touches the boundary $x_2 = 0$ has a likelihood ratio larger than 1, i.e.,

$$LR^{PW} = \frac{\mu_2 + \mu_1}{\lambda + \mu_1} > 1, \tag{6.13}$$

for *all* cycles that touch the boundary $x_2 = 0$.

If the number of those cycles is significant, the final estimate can have a large variance. For SDH, only cycles of length *three* that touch the boundary have a likelihood ratio larger than one. Hence, the probability of those cycles to be in a path is smaller. This explains why the SDH change of measure shows much better performance than the PW change of measure.

Still, this observation shows that the likelihood ratio of a path for the SDH change of measure does depend on cycles of length three that touch the boundary $x_2 = 0$. This can dramatically influence the likelihood ratio of the path. In principle, it can go to infinity if the path goes an infinite number of times through this type of cycles.

**Observation 3.** *For the SDHI change of measure, cycles of length three that touch the boundary $x_2 = 0$ have likelihood ratios either smaller or larger than 1, depending on the cycle type and $b$ value.*

Namely, for cycle $C_1$:

$$LR_{C_1}^{SDHI} = \frac{\lambda \mu_2 \frac{\mu_1}{\lambda+\mu_1}}{\mu_2 \left(\mu_1 + \frac{\lambda-\mu_1}{b}\right)\frac{\lambda}{\mu_2+\lambda}} = \frac{\mu_1(\lambda + \mu_2)}{\left(\mu_1 + \frac{\lambda-\mu_1}{b}\right)(\lambda + \mu_1)}, \tag{6.14}$$

and

$$\begin{cases} LR_{C_1}^{SDHI} = 1 \text{ for } b_{C_1} = \dfrac{\lambda^2 - {\mu_1}^2}{\mu_1(\mu_2 - \mu_1)}, \\ LR_{C_1}^{SDHI} < 1 \text{ for } b \in (b_{C_1}, \infty), \\ LR_{C_1}^{SDHI} > 1 \text{ for } b \in (0, b_{C_1}). \end{cases} \tag{6.15}$$

For cycle $C_2$:

$$LR_{C_2}^{SDHI} = \frac{\mu_2 \frac{\lambda}{\lambda+\mu_1}\frac{\mu_1}{\lambda+\mu_1}}{\left(\mu_1 + \frac{\lambda-\mu_1}{b}\right)\frac{\mu_2}{\mu_2+\lambda}\frac{\lambda}{\mu_2+\lambda}} = \frac{\mu_1(\lambda + \mu_2)^2}{\left(\mu_1 + \frac{\lambda-\mu_1}{b}\right)(\lambda + \mu_1)^2}, \tag{6.16}$$

and

$$\begin{cases} LR_{C_2}^{SDHI} = 1 \text{ for } b_{C_2} = \dfrac{(\lambda^2 - {\mu_1}^2)(\lambda + \mu_1)}{\mu_1(\mu_2 - \mu_1)(2\lambda + \mu_1 + \mu_2)}, \\ LR_{C_2}^{SDHI} < 1 \text{ for } b \in (b_{C_2}, \infty)\,, \\ LR_{C_2}^{SDHI} > 1 \text{ for } b \in (0, b_{C_2})\,. \end{cases} \tag{6.17}$$

Thus, only for $b = b_{C_1}$ for cycles of type $C_1$ and for $b = b_{C_2}$ for cycles of type $C_2$, likelihood ratios of those cycles are equal to 1. Since the equivalence $b_{C_1} = b_{C_2}$ is impossible for any network parameters (otherwise, $(\lambda + \mu_1)/(2\lambda + \mu_1 + \mu_2) = 1$, and, hence, $\lambda + \mu_2 = 0$), a likelihood ratio of a path also for the SDHI change of measure does depend on cycles if those cycles have a length three and touch the boundary $x_2 = 0$.

**Summary of the previous observations**

Thus, from Observations 1–3 we see that

1. cycles that go only through states with $x_2 \geq b$ are “harmless” for the likelihood ratio of the whole path;
2. for $0 \leq x_2 < b$, cycles of length 3 that touch boundary $x_2 = 0$ have likelihood ratios not equal to 1, hence, are “problematic”. However, the probability of those cycles can be very small if a path does not go near the border $x_2 = 0$ all the way, and, hence, may be not that “harmful”.

**Remark 6.1.1.** Since the term “harmful” is used later in this section, we define it here. We call a cycle *“harmful”* (for the likelihood ratio of the path) if its likelihood ratio is not equal to one.

Now, the question arises, what is the behavior of other cycles, namely, those, that do *not* touch the boundary $x_2 = 0$?

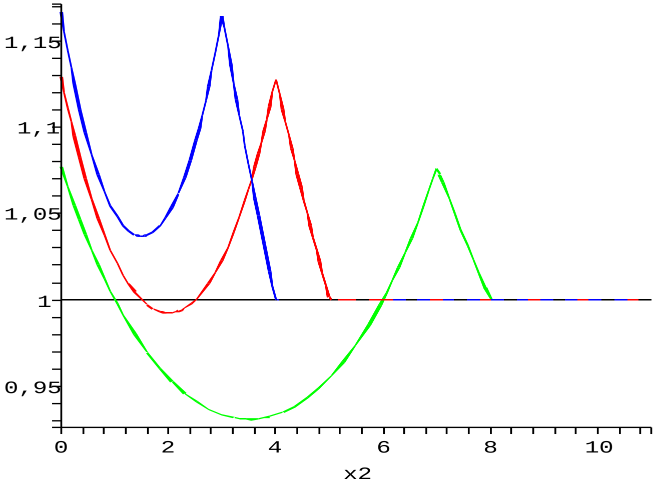
**Observation 4.** *Any cycle of length* $3 \cdot c$ *(for* $c > 1$*) can be decomposed to* $c$ *cycles of length 3 with some multiplicand that is a ratio of arrival rates at different states. This multiplicand is equal to 1 for SDHI, and can be smaller or larger than 1 for the SDH change of measure depending on the cycle type.*

The proof of the above observation is given in Appendix B. There we, first, show that this is true for all cycles of length six and then use induction to show that this is also true for every cycle of length more than six.

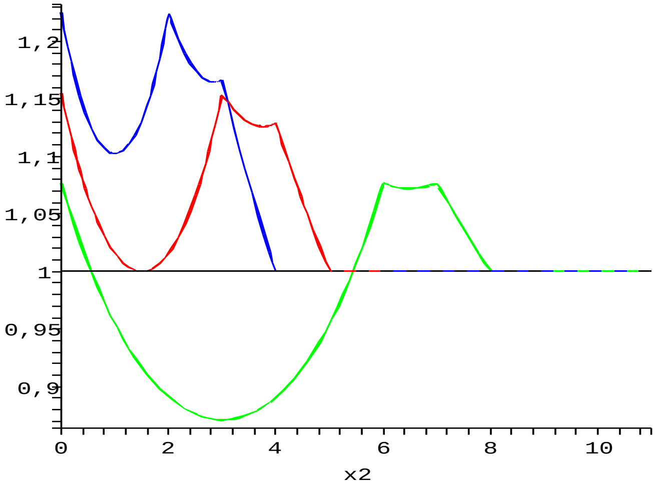
The above observation is more useful for the SDHI change of measure since it guarantees that this is sufficient to study only cycles of length three.

**Observation 5.** *Likelihood ratios of cycles going through states* $0 < x_2 < b$ *can be larger than 1 for some values of* $b$ *and larger or smaller than 1, depending on* $x_2$*, for other values of* $b$ *for both the SDH and SDHI changes of measure.*

Let us consider an example of a 2-node tandem network with $\lambda = 0.18$, $\mu_1 = 0.42$, $\mu_2 = 0.4$. To prove the above observation, let us look at likelihood ratios for cycles of length three (Figures 6.2a, 6.3a) and six (Figures 6.2b, 6.3b) for the SDH and SDHI changes of measure, respectively.

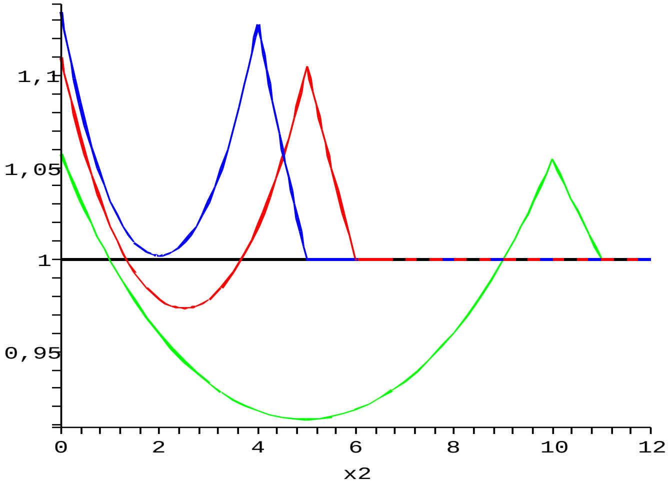


(a) LRs of cycles of length 3 with $b' = 5, b_{opt} = 6, b^* = 11$

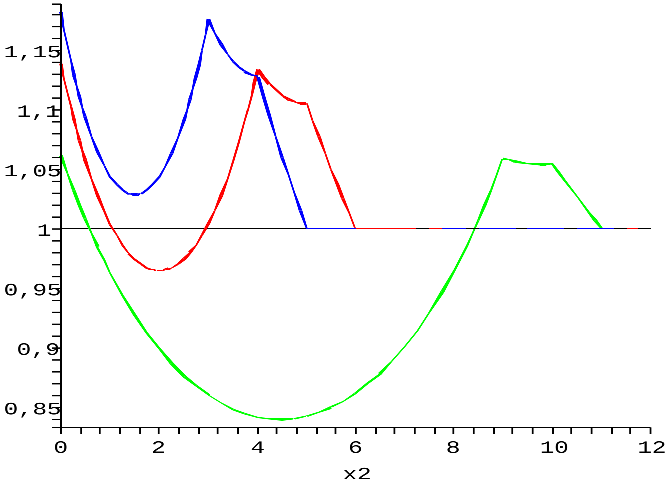


(b) LRs of cycles of length 6 with $b' = 5, b_{opt} = 6, b^* = 11$

Figure 6.2: 2 tandem ($\lambda = 0.18, \mu_1 = 0.42, \mu_2 = 0.4$). SDH



(a) LRs of cycles of length 3 with $b' = 4, b_{opt} = 5, b^* = 8$



(b) LRs of cycles of length 6 with $b' = 4, b_{opt} = 5, b^* = 8$

Figure 6.3: 2 tandem ($\lambda = 0.18, \mu_1 = 0.42, \mu_2 = 0.4$). SDHI

**Remark 6.1.2.** The parameter $x_2$ on the horizontal axis corresponds to the minimum $x_2$ for all $x_2$ from the cycle.

*Red lines* in these figures correspond to likelihood ratios of SDH or SDHI changes of measure with $b = b_{opt}$ found by simulation. One can see that for $x_2 \geq b$ they are equal to one and for $0 < x_2 < b$ they take values both above and below one.

For other values of $b$, we saw experimentally that curves of likelihood ratios go up when $b$ becomes smaller and go down when $b$ increases. Thus, we picked some other values of $b$, namely, the two extreme cases. The first is the case when likelihood ratios are always larger than (or equal to) one. This is $b = b'$, corresponding to the maximum of those values and depicted with *blue lines.*

For $x_2 = 0$ or $x_2 = b - 2$, likelihood ratios turned out to be larger than one for all $b$. This corresponds to cycles that touch boundary $x_2 = 0$ and $x_2 = b$. Thus, we considered only the cycles that do not touch those boundaries and tried to find the minimum $b = b^*$ for which likelihood ratios are smaller (or equal to) one. These are states with $1 \leq x_2 \leq b-2$ for cycles of length three and states with $1 \leq x_2 \leq b-3$ for cycles of length six. The likelihood ratios corresponding to $b = b^*$ are depicted with *green lines.*

As one can see, the optimal value of $b$ ($b_{opt}$, obtained via simulation) is such that likelihood ratios for cycles of length three and six are smaller than one for some and larger than one for other values of $x_2$. It is interesting to note that (for this example of arrival and service rates) $b_{opt} = b' + 1$, i.e., it is one more than the maximum $b$ for which likelihood ratios (for cycles of length three and six) are larger than one, or, in other words, it is the minimum $b$ for which likelihood ratios become smaller than one.

Note, however, that there is no proof that $b_{opt}$ is always such that $b_{opt} = b' + 1$ and that $b'$, $b^*$ exist for all network parameters of a 2-node tandem network.

The main purpose of this observation was to show that the likelihood ratio of a cycle that does not touch the boundary $x_2 = 0$ is not equal to one. We showed this on one network example and also observed this for others (not included here).

**Summary of the previous observations**

Thus, from Observations 1–5 we see that

1. cycles going only through states with $x_2 \geq b$ are “harmless” for the likelihood ratio of the whole path;
2. cycles of length 3 that touch boundary $x_2 = 0$ and go through states with $0 \leq x_2 < b$ are “problematic” but unlikely;
3. cycles of length more than 3 and going through states with $0 < x_2 < b$ can be larger and smaller than 1, hence, also “problematic”.

   Thus, *all* cycles that go through states with $0 < x_2 < b$ are “harmful” for a likelihood ratio of a path. Now, the question is, does this number grow with the overflow level $N$? If it is small enough compared to the likely states that a path visits this might be not that “harmful”.

**Observation 6.** *Likelihood ratios of cycles for both the SDH and SDHI change of measure do not go to 1 in the limit as overflow level $N \to \infty$.*

We saw from the previous observations that a likelihood ratio of a cycle going through states with $x_2 \geq b$ is equal to one, hence, these states are "harmless" for the likelihood ratio of the path. A likelihood ratio of a cycle going through states with $x_2 < b$ is not equal to one, thus, these states are "harmful" and their number depends on $b$.

From the experimental results in Section 3.6 we saw that for some network parameters the optimal $b$ ($b = b_{opt}$) depends on the overflow level $N$ and grows slowly as $N \to \infty$ (cf. Figure 3.14). It was also observed experimentally (one can already see that from Figures 6.2–6.3) that as $b$ increases, the lowest point of the curves does not increase, i.e., a likelihood ratio does not go to one when $b$ increases, and, hence, "harmful" states stay to be such. Since the number of them depends on $b$ and $b$ grows with the overflow level $N$, the number of these states also grows as $N \to \infty$, and the probability that a cycle go through these states is not rare.

Consequently, likelihood ratios of cycles for both the SDH and SDHI change of measure are *not* always equal to one, nor *go to* one in the limit as the overflow level $N \to \infty$. This means that we can not use some crucial parts of the reasoning in [47] to prove the asymptotic efficiency of our changes of measure. Hence, unfortunately, the second proof approach can not be applied, either.

The observations above are, however, helpful in explaining why our changes of measure showed good performance. The value $b_{opt}$ found by simulation might be such that on average cycles have likelihood ratios almost equal to one, hence, the likelihood ratio of a path does not depend much on the number of cycles it has. Another explanation could be that cycles mostly go through states with $x_2 \geq b$ (for which likelihood ratios are equal to one according to Observation 1).

### 6.1.4 Final remarks

As we saw from the previous discussion, neither of the two approaches for proving asymptotic efficiency can be applied in our case. However, we presented them for two reasons. First, to show that these approaches fall short as proofs for our changes of measure. Second, to give more understanding on the behavior of changes of measure. These observations may form a good step as a starting point for further research.

## 6.2 Non-Markovian networks

In this section we discuss the extension of our heuristics to non-Markovian queuing networks. In Section 6.2.1 we describe the two models that we are going to consider. In Section 6.2.2 we repeat a known result for simulating a $GI/GI/1$ queue, which we will use later. In Section 6.2.3 the specifics of state-dependent simulation for non-Markovian queuing networks are discussed in general. In Section 6.2.4 we propose two specific changes of measure. In Sections 6.2.7–6.2.8 its performance is experimentally validated for the two examples.

## 6.2.1 Non-Markovian models

In the previous chapters all the networks we considered were Jackson queuing networks. Modeling arrival and service processes as Markovian is only an approximation to a real communication networks behavior, though possibly realistic one for an arrival process in some cases (when users independently send about the same amount of traffic). At the same time, assuming that the service process is exponentially distributed is less realistic. It is more reasonable to suggest that it is bi-modally distributed, i.e., that packets are of two different kinds, say, the acknowledgment packets and the real data packets and each type of packet has some specified amount of time that it needs to be served.

More realistic arrival processes can be achieved if we allow some burstiness, i.e., packets can come in batches. This can be implemented by modeling the inter-arrival times as hyper-exponentially distributed random variable.

In the end, our goal is to consider a fully non-Markovian network, i.e., both, inter-arrival and service times being non-exponentially distributed. This would be, for example, the case with inter-arrival times being hyper-exponentially distributed and service times being bi-modally distributed. As an intermediate step, we look at a case in which one of the two processes is Markovian. We choose the case when the arrival process is such, i.e., service times are bi-modally distributed and inter-arrival times are exponentially distributed.

#### Service process

Consider the service times to be *bi-modally distributed.* Suppose that the transmission speed is $\nu$ (bits/sec) and packets have length $l_1$ with probability $p$ and length $l_2$ with probability $1-p$. We call them packets of *type 1* and *2*, respectively. Then,

$$\text{service time} = \begin{cases} t_1 = l_1/\nu, & \text{with probability } p, \\ t_2 = l_2/\nu, & \text{with probability } (1-p), \end{cases}$$

and the moment generating function of the service process is

$$M_{serv}(\theta) = e^{\theta t_1} p + e^{\theta t_2}(1-p). \tag{6.18}$$

An Importance Sampling (IS) change of measure to be used for simulating bi-modally distributed service times changes the probability $p$, and neither, the transmission speed, nor packets lengths can be changed, since the latter two are not random. This also follows from the definition of IS. If in the new system an event is possible such that in the original system it has a zero probability, the corresponding likelihood ratio is equal to zero. Thus, only the events that are possible in the original system have a non-zero probability in the new system.

#### Arrival process

If the inter-arrival times are *exponentially distributed* with rate $\lambda$, then, its density function is

$$f(x) = \lambda e^{-\lambda x}, \tag{6.19}$$

and the moment generating function is equal to

$$M_{ar}(\theta) = \frac{\lambda}{\lambda - \theta}. \tag{6.20}$$

If the inter-arrival times are *hyper-exponentially distributed* with parameters $\lambda_1$, $\lambda_2$, i.e., the inter-arrival time is with probability $q$ an exponentially distributed random variable $X_1$ with mean $1/\lambda_1$ and with probability $(1-q)$ an exponentially distributed random variable $X_2$ with mean $1/\lambda_2$, then, its density function is

$$f(x) = q\lambda_1 e^{-\lambda_1 x} + (1-q)\lambda_2 e^{-\lambda_2 x}, \tag{6.21}$$

and the moment generating function

$$M_{ar}(\theta) = \frac{q\lambda_1}{\lambda_1 - \theta} + \frac{(1-q)\lambda_2}{\lambda_2 - \theta}. \tag{6.22}$$

An IS change of measure changes the arrival rate $\lambda$ for exponentially distributed inter-arrival times, and the rates $\lambda_1$, $\lambda_2$ and the probability $q$ for hyper-exponentially distributed inter-arrival times.

#### Model 1: M/Bim/1 → ·/Bim/1

The first network model we consider is a 2-node tandem queuing network with inter-arrival times being exponentially distributed and the service times at node $i$ being bi-modally distributed with probability $p$ (for both nodes) and packet types $l_1$ and $l_2$. In a real system the outgoing process from node 1 determines the incoming process to node 2, i.e., if a packet of type $k$ enters node 1 it goes to node 2 with probability one, thus, generally speaking node 2 does not have a probabilistic choice of a packet type. In our simplified model we ignore this and assume that node 2 still has a probabilistic choice (with the same probability $p$), i.e., we assume that nodes operate independently of each other. This is know as *Kleinrock's independence assumption* and has been shown by numerous simulation results to be reasonable for networks of moderate connectivity [48].

#### Model 2: $H_2$/Bim/1 → ·/Bim/1

The second model we consider is also a 2-node tandem queuing network with the service times at each node being bi-modally distributed with parameters $l_1$, $l_2$, $p$, but the inter-arrival times being hyper-exponentially distributed with the rates $\lambda_1$, $\lambda_2$ and the probabilities $q$, $(1-q)$. The above assumption of independence regarding service times at node 2 is also made.

### 6.2.2 Optimal change of measure for $GI/GI/1$ queue

As discussed in Section 2.4.1 and shown in [2], a provably asymptotically efficient change of measure for simulating a $GI/GI/1$ queue, is *an exponential change of measure* (exponential twist), proposed in [15] and defined as

$$d\tilde{F}(x) = \frac{e^{\theta x} dF(x)}{M(\theta)}, \text{ with } \theta \in \mathbb{R}, \tag{6.23}$$

where $F(x)$ is the original distribution and $M(\theta) = \mathbb{E}e^{\theta x}$ is the moment generating function. The best change of measure is the one for which parameter $\theta = \theta^*$, with $\theta^*$ being a solution of the equation

$$M_{ar}(-\theta)M_{serv}(\theta) = 1, \tag{6.24}$$

with $M_{ar}(\theta)$ and $M_{serv}(\theta)$ being the moment generating functions, for the arrival and service process, respectively. According to (6.23), the new inter-arrival time distribution is given by

$$d\tilde{F}_{ar}(x) = \frac{e^{-\theta^* x} dF_{ar}(x)}{M_{ar}(-\theta^*)}, \tag{6.25}$$

and the new service time distribution is given by

$$d\tilde{F}_{serv}(x) = \frac{e^{\theta^* x} dF_{serv}(x)}{M_{serv}(\theta^*)}. \tag{6.26}$$

This is the PW change of measure which for an $M/M/1$ queue exchanges arrival and service rate. We used this change of measure to construct our heuristics for Markovian networks and are going to do the same for non-Markovian ones.

**Model 1**: When the arrival process is exponentially distributed and the service process is bi-modally distributed the value $\theta^*$ for exponentially twisted change of measure is found as a solution of Equation (6.24), i.e.,

$$\frac{\lambda}{\lambda + \theta}\left(e^{\theta t_1} p + e^{\theta t_2}(1-p)\right) = 1. \tag{6.27}$$

One of the two non-negative solutions [23] is $\theta = 0$ which corresponds to no change of measure, and, hence, can not be considered. Another one is $\theta > 0$, which can be found numerically.

**Model 2**: If the arrival process is hyper-exponentially distributed and the service process is bi-modally distributed, then $\theta^*$ is found from

$$\left(\frac{q\lambda_1}{\lambda_1 + \theta} + \frac{(1-q)\lambda_2}{\lambda_2 + \theta}\right) \cdot \left(e^{\theta t_1} p + e^{\theta t_2}(1-p)\right) = 1. \tag{6.28}$$

Again, we are not interested in the solution $\theta = 0$ and the solution $\theta > 0$ can be found numerically.

### 6.2.3 Simulating non-Markovian networks

In the previous chapters all the networks under consideration had exponentially distributed inter-arrival and service times, thus, could be simulated as discrete time Markov chains (DTMC). The natural advantage of it, of course, was the irrelevance of time for a full system state description. For non-Markovian networks a state of the system does include time, and, thus, at each moment in time (and, specifically, at each moment that some event happens) the system state is represented not only by the number of customers at each node, but also by the time elapsed since the previous service completion at each node (if the service time has a non-exponential distribution) and the time elapsed from the previous arrival to the network (if the inter-arrival time has a non-exponential distribution). Hence, every time a new event is scheduled the elapsed times for every node have to be taken into account and the new event has to be scheduled *conditioned* on it. This makes programming a non-Markovian simulator much more complicated.

Another complication is the existence of two ways to schedule the next event. This gives two different types of changes of measure. The first one, called a change of measure *without rescheduling*, does the following. At each moment the state changes due to a service completion at some node or an arrival to the network, the next event is scheduled only for this specific node and the events scheduled for other nodes are unchanged. The second way, called a change of measure *with rescheduling*, changes the time of the next event for *each node*, every time the system state changes. Besides rescheduling, a state-dependent change of measure can be either partly state-dependent, i.e., depends only on the number of customers in the network and not on the elapsed times, or fully state-dependent, i.e., depends also on the elapsed times. The second one is difficult to implement since it would require continuous rescheduling. For more detailed information on the exact algorithm description and different types of changes of measure see [49]. According to [49] the most efficient and practical one is a *partially* state-dependent change of measure *with* rescheduling. This is the one we use in our case. For experiments with non-Markovian networks in Sections 6.2.7–6.2.8 we used the simulation program of the author of [49].

### 6.2.4 Extension for tandem networks. General case

To construct a change of measure for a 2-node non-Markovian tandem queuing network we apply the same procedure as for the Markovian case. There our change of measure is a combination (depending on a number of customers in each node) of changes of measure which are proven to be the best to "push" a single node, i.e., the exponential twist (Equation (6.23)). For the $M/M/1$ queue, the exponentially twisted change of measure simply means interchanging the arrival and service rates. For non-Markovian node the interpretation depends on the inter-arrival and service time distribution.

The change of measure for a 2-node non-Markovian tandem network (cf. Section 3.2.3 for the Markovian case) can be written as

$$COM_{x_2} = \left[\frac{x_2}{b}\right]^1 COM_2 + \left[\frac{b - x_2}{b}\right]^+ COM_1, \tag{6.29}$$

where $COM_i$ is the exponentially twisted change of measure for node $i$ (Equation (6.23)), $[a]^1 = \min(a, 1)$, $[a]^+ = \max(a, 0)$ and $x_2$ is the number of customers at node 2. As before the change of measure depends on parameter $b$, which are yet to be determined.

The alternative change of measure (equivalent to the one in Appendix A, i.e., with dependence on the number of customers ($x_1$ and $x_2$) at both nodes) is

$$\begin{aligned} COM_{x_1,x_2} = \left[\frac{x_2}{b_2}\right]^1 COM_2 + \left[\frac{b_2 - x_2}{b_2}\right]^+ \times \\ \times \left( \left[\frac{x_1}{b_1}\right]^1 COM_1 + \left[\frac{b_1 - x_1}{b_1}\right]^+ COM_0 \right), \end{aligned} \tag{6.30}$$

where $COM_0$ corresponds to the original network parameters, i.e., no change of measure and $b_1$, $b_2$ being some integer numbers to be determined.

Equations (6.29)–(6.30) represent the dependence of the new network parameters on the original ones. Note, however, that the exact representation of $COM_i$ still needs to be defined, since parameters to be changed need to be chosen. There are two possible ways.

The first possibility is to change the arrival and the service processes *directly through their parameters*, i.e., let $COM_i$ represent the new arrival and service parameters for the whole network. The second possibility is to change the arrival and the service processes *indirectly*, i.e., through the *twisting parameter* $\theta^*$ and let $COM_i$ represent the new twisting parameters.

Let us describe these two possibilities in more detail. Let $\theta_i^*$ denote the optimal twisting parameter for simulating node $i$ as a single node (the non-zero solution of Equation (6.24) with $M_{serv}(\theta) = M_{serv_i}(\theta)$ where $M_{serv_i}(\theta)$ is the moment generating function for the service distribution at node $i$).

### The change of measure linear in the parameters: COM$^p$

The first possibility, called *the change of measure linear in the parameters* and denoted by **COM**$^p$ (i.e., $COM^p_{x_2}$ and $COM^p_{x_1,x_2}$ for (6.29) and (6.30), respectively) is represented in the following steps.

1. Find the twisting parameters $\theta_i^*$ for each node $i$.
2. Use Equations (6.25)–(6.26), for each node $i$ to calculate the new inter-arrival and service time distributions.
3. Use the results from the previous step to define $COM_i$ as a vector of the new arrival and service parameters *to "push" node* $i$ as a single node.
4. Define the change of measure for the whole network as a vector of *the new arrival and service parameters* using Equation (6.29) for $COM^p_{x_2}$ (respectively, Equation (6.30) for $COM^p_{x_1,x_2}$) and $COM_i$ defined in the previous step.

In other words, the change of measure for the whole network is a combination *of changes of measure* for each node.

### The change of measure linear in $\theta$: COM$^\theta$

The second possibility, called *the change of measure linear in* $\theta$ and denoted by **COM**$^\theta$ (i.e., $COM^\theta_{x_2}$ and $COM^\theta_{x_1,x_2}$ for (6.29) and (6.30) changes of measure, respectively), is defined as follows. In essence, in this approach, we change the order of the steps proposed above by moving step 2 to the last position.

1. Find the twisting parameters $\theta_i^*$ for each node $i$.
2. Define $COM_i$ as a vector of *twisting parameters to "push" node $i$* as a single node in the network.
3. Define a vector of *twisting parameters for the whole network* using Equation (6.29) for $COM_{x_2}^{\theta}$ (respectively, Equation (6.30) for $COM_{x_1,x_2}^{\theta}$).
4. The change of measure for the whole 2-node tandem network is defined from Equations (6.25)–(6.26) with the twisting parameter parameter found in the previous step.

In other words, the change of measure for the whole network is defined as exponential twist with parameter $\theta^*$ found as a combination *of twisting parameters* for each node.

Note that for Markovian models, the above two possibilities are equivalent, since the new (exponentially twisted) arrival and service rates depend linearly on $\theta^*$ as $\tilde{\lambda} = \lambda + \theta^*$ and $\tilde{\mu} = \mu - \theta^*$. Below we present the exact equations of the proposed heuristics for both our models.

### 6.2.5 Exact calculation of COM for Model 1: M/Bim/1 → ·/Bim/1

Let $\theta_1^*$, $\theta_2^*$ denote the twisting parameters for node 1 and 2 (found from Equation (6.23)). Let

$$t_{i,k} = \frac{l_k}{\nu_i} \tag{6.31}$$

denote the service time of packet of type $k$ at node $i$ (remember that $l_k$ is a length of a packet of type $k$ and $\nu_i$ is a service rate at node $i$). Let $\tilde{\lambda}_i$ denote the new arrival rate and $\tilde{p}_i$ denote the new probability for the service process for node $i$.

#### The change of measure linear in the parameters: COM$^p$

COM$^p$ changes the arrival rate and the probability $p_i$ for node $i$, i.e.,

$$\tilde{\lambda}_i = \lambda + \theta_i^*, \tag{6.32}$$

$$\tilde{p}_i = \frac{e^{\theta_i^* t_{i,1}} p}{M_{serv_i}(\theta_i^*)}. \tag{6.33}$$

Then, the optimal exponentially twisted change of measure $COM_i$ for a single node $i$ ($i = 1, 2$) determines the new arrival rate, the new probability at node 1 and the new probability at node 2 as

$$COM_1 = (\tilde{\lambda}_1, \tilde{p}_1, p), \tag{6.34}$$

$$COM_2 = (\tilde{\lambda}_2, p, \tilde{p}_2), \tag{6.35}$$

$$COM_0 = (\lambda, p, p), \tag{6.36}$$

where $COM_0$ corresponds to no change of measure. The resulting $COM_{x_2}^{p}$ for Model 1 is equal to (substituting (6.34) and (6.35) into (6.29))

$$COM^p_{x_2}: \begin{cases} \tilde{\lambda} = \lambda + \left[\frac{x_2}{b}\right]^1 \theta_2^* + \left[\frac{b-x_2}{b}\right]^+ \theta_1^*, \\ \tilde{p}_1 = p\left(\left[\frac{x_2}{b}\right]^1 + \left[\frac{b-x_2}{b}\right]^+ \frac{e^{\theta_1^* t_{1,1}}}{M_{serv_1}(\theta_1^*)}\right), \\ \tilde{p}_2 = p\left(\left[\frac{x_2}{b}\right]^1 \frac{e^{\theta_2^* t_{2,1}}}{M_{serv_2}(\theta_2^*)} + \left[\frac{b-x_2}{b}\right]^+\right). \end{cases}$$

The change of measure $COM^p_{x_1,x_2}$ for Model 1 can be written by substituting (6.34), (6.35) into (6.30).

### The change of measure linear in $\theta$: COM$^\theta$

COM$^\theta$ is defined as follows. $COM_i$ denotes the new twisting parameters, i.e.,

$$COM_1 = (-\theta_1^*, \theta_1^*, 0), \tag{6.37}$$

$$COM_2 = (-\theta_2^*, 0, \theta_2^*), \tag{6.38}$$

$$COM_0 = (0, 0, 0). \tag{6.39}$$

Thus, $COM_1$ twists node 1, change of measure $COM_2$ twists node 2 and $COM_0$ corresponds to no change of measure.

The linear in $\theta$ change of measure is defined as the exponentially twisted distribution with parameter $\theta^*$ defined as follows

$$COM^\theta_{x_2}: \begin{cases} \tilde{\theta}_{ar} = -\left[\frac{x_2}{b}\right]^1 \theta_2^* - \left[\frac{b-x_2}{b}\right]^+ \theta_1^*, \\ \tilde{\theta}_{serv_1} = \left[\frac{b-x_2}{b}\right]^+ \theta_1^*, \\ \tilde{\theta}_{serv_2} = \left[\frac{x_2}{b}\right]^1 \theta_2^*, \end{cases} \tag{6.40}$$

$$COM^\theta_{x_1,x_2}: \begin{cases} \tilde{\theta}_{ar} = -\left[\frac{x_2}{b_2}\right]^1 \theta_2^* - \left[\frac{b_2-x_2}{b_2}\right]^+ \left[\frac{x_1}{b_1}\right]^1 \theta_1^*, \\ \tilde{\theta}_{serv_1} = \left[\frac{b_2-x_2}{b_2}\right]^+ \left[\frac{x_1}{b_1}\right]^1 \theta_1^*, \\ \tilde{\theta}_{serv_2} = \left[\frac{x_2}{b_2}\right]^1 \theta_2^*. \end{cases} \tag{6.41}$$

This changes the arrival and service parameters as follows

$$\begin{cases} \tilde{\lambda} = \lambda + \tilde{\theta}_{ar}, \\ \tilde{p}_1 = \frac{e^{\tilde{\theta}_{serv_1} t_{1,1}} p}{M_{serv_1}(\tilde{\theta}_{serv_1})}, \\ \tilde{p}_2 = \frac{e^{\tilde{\theta}_{serv_2} t_{2,1}} p}{M_{serv_2}(\tilde{\theta}_{serv_2})}, \end{cases} \tag{6.42}$$

where twisting parameters $\tilde{\theta}_{ar}$, $\tilde{\theta}_{serv_1}$ and $\tilde{\theta}_{serv_2}$ are found from the set of Equations (6.40) for $COM^\theta_{x_2}$ and from the set of Equations (6.41) for $COM^\theta_{x_1,x_2}$.

### 6.2.6 Exact calculation of COM for Model 2: $\mathrm{H}_2$/Bim/1 → ·/Bim/1

Let $\tilde{\lambda}_{i,1}$, $\tilde{\lambda}_{i,2}$, $\tilde{q}_i$, $(1-\tilde{q}_i)$ denote the new arrival rates and probabilities for node $i$, and $\tilde{p}_i$ denote the new probability for the service process for node $i$.

#### The change of measure linear in the parameters: COM$^p$

For Model 2 COM$^p$ changes the arrival rates $\lambda_1$ and $\lambda_2$, the arrival probabilities $q$ and $(1-q)$ and the service process probability $p_i$ (for node $i$). From (6.25)–(6.26) the new network parameters can be found as follows

$$\tilde{\lambda}_{i,k} = \lambda_k + \theta_i^*, \tag{6.43}$$

$$\tilde{q}_i = \frac{q\lambda_1(\lambda_2+\theta_i^*)}{q\lambda_1(\lambda_2+\theta_i^*)+(1-q)\lambda_2(\lambda_1+\theta_i^*)}, \tag{6.44}$$

$$\tilde{p}_i = \frac{e^{\theta_i^* t_{i,1}}p}{M_{serv_i}(\theta_i^*)}. \tag{6.45}$$

The changes of measure $COM_i$ for each node $i$ correspond to the new arrival and service processes and are equal to

$$COM_1 = (\tilde{\lambda}_{1,1}, \tilde{\lambda}_{1,2}, \tilde{q}_1, \tilde{p}_1, p), \tag{6.46}$$

$$COM_2 = (\tilde{\lambda}_{2,1}, \tilde{\lambda}_{2,2}, \tilde{q}_2, p, \tilde{p}_2), \tag{6.47}$$

$$COM_0 = (\lambda_1, \lambda_2, q, p, p), \tag{6.48}$$

where $COM_0$ corresponds to no change of measure.

The change of measure $COM^p_{x_2}$ for Model 2 is defined as (substituting (6.46) and (6.47) into (6.29))

$$COM^p_{x_2}: \begin{cases} \tilde{\lambda}_1 = \lambda_1 + \left[\frac{x_2}{b}\right]^1 \theta_2^* + \left[\frac{b-x_2}{b}\right]^+ \theta_1^*, \\ \tilde{\lambda}_2 = \lambda_2 + \left[\frac{x_2}{b}\right]^1 \theta_2^* + \left[\frac{b-x_2}{b}\right]^+ \theta_1^*, \\ \tilde{q} \ = q\lambda_1 \left( \left[\frac{x_2}{b}\right]^1 \frac{(\lambda_2+\theta_2^*)}{q\lambda_1(\lambda_2+\theta_2^*)+(1-q)\lambda_2(\lambda_1+\theta_2^*)} + \right. \\ \qquad \left. + \left[\frac{b-x_2}{b}\right]^+ \frac{(\lambda_2+\theta_1^*)}{q\lambda_1(\lambda_2+\theta_1^*)+(1-q)\lambda_2(\lambda_1+\theta_1^*)} \right), \\ \tilde{p}_1 = p\left( \left[\frac{x_2}{b}\right]^1 + \left[\frac{b-x_2}{b}\right]^+ \frac{e^{\theta_1^* t_{1,1}}}{M_{serv_1}(\theta_1^*)} \right), \\ \tilde{p}_2 = p\left( \left[\frac{x_2}{b}\right]^1 \frac{e^{\theta_2^* t_{2,1}}}{M_{serv_2}(\theta_2^*)} + \left[\frac{b-x_2}{b}\right]^+ \right), \end{cases}$$

The change of measure $COM^p_{x_1,x_2}$ for Model 2 can be written by substituting (6.46)–(6.48) into (6.30).

### The change of measure linear in $\theta$: COM$^\theta$

COM$^\theta$ is defined from Equation (6.40) for $COM^\theta_{x_2}$ and from Equation (6.41) for $COM^\theta_{x_1,x_2}$. The arrival and service processes of the whole network change as follows.

$$\begin{cases} \tilde{\lambda}_1 = \lambda_1 + \theta^*, \\ \tilde{\lambda}_2 = \lambda_2 + \theta^*, \\ \tilde{q} \ = \dfrac{q\lambda_1(\lambda_2+\theta^*)}{q\lambda_1(\lambda_2+\theta^*) + (1-q)\lambda_2(\lambda_1+\theta^*)}, \\ \tilde{p}_1 = \dfrac{e^{\theta^* t_{1,1}} p}{M_{serv_1}(\theta^*)}, \\ \tilde{p}_2 = \dfrac{e^{\theta^* t_{2,1}} p}{M_{serv_2}(\theta^*)}. \end{cases} \tag{6.49}$$

### 6.2.7 Experimental results for Model 1: exponential arrivals and bi-modal service rates

In this section we consider the experimental results for Model 1 described in Section 6.2.1. For each example we compare the performance of all four changes of measure. For changes of measure dependent on both, $x_1$ and $x_2$, i.e., for $COM^p_{x_1,x_2}$ and $COM^\theta_{x_1,x_2}$, we choose $b_1 = b_2$.

#### Network parameters

Consider the example of Model 1 with the following parameters

$$\begin{cases} \lambda \ = 5000 \text{ packets/s}, \\ l_1 = 50 \text{ bytes } = 400 \text{ bits}, \\ l_2 = 1500 \text{ bytes } = 12000 \text{ bits}, \\ p \ = 0.4, \\ \nu_1 = 81 \text{ Mbit/s } = 8.1 \cdot 10^7 \text{ bit/s}, \\ \nu_2 = 80 \text{ Mbit/s } = 8 \cdot 10^7 \text{ bit/s}. \end{cases} \tag{6.50}$$

The network parameters are chosen to be close to ones that could be observed on an Ethernet for the TCP protocol. The packet length $l_2$ corresponds to a data packet and the packet length $l_1$ corresponds to an acknowledgment packet. Parameter $p = 0.4$ means that 40% of sent packets are acknowlegments, and 60% are the real data packets.

The case of equal service rates at the two nodes is known to be the most difficult for simulation. We have chosen our service rates to be *almost equal* since for equal service rates two events may be scheduled at exactly the same time, which would need special handling in the simulation program.

The service time $t_{i,k} = l_k/\nu_i$ (Equation 6.31) of a packet of type $k$ at node $i$ is

$$\begin{cases} t_{1,1} = 4.938 \cdot 10^{-6} \text{ sec}, \\ t_{1,2} = 14.815 \cdot 10^{-5} \text{ sec}, \\ t_{2,1} = 5 \cdot 10^{-6} \text{ sec}, \\ t_{2,2} = 15 \cdot 10^{-5} \text{ sec}, \end{cases} \tag{6.51}$$

and the system utilizations are equal to $\rho_i = \lambda(pt_{i,1} + (1-p)t_{i,2})$, with

$$\begin{cases} \rho_1 = 0.454, \\ \rho_2 = 0.460. \end{cases} \tag{6.52}$$

### Calculation of COMs

The numerical solutions of Equation (6.27) (rounded to integer values for simplicity) for nodes 1 and 2 are, respectively,

$$\begin{cases} \theta_1^* = 9674, \\ \theta_2^* = 9419. \end{cases} \tag{6.53}$$

The new arrival rates and service probabilities for COM$^p$ are found from Equations (6.32)–(6.33) and are equal to

$$\begin{cases} \tilde{\lambda}_1 = \lambda + \theta_1^* = 14674 \text{ (bits/s)}, \\ \tilde{\lambda}_2 = \lambda + \theta_2^* = 14419 \text{ (bits/s)}, \\ \tilde{p}_1 = \dfrac{e^{\theta_1^* t_{1,1}} p}{M_{serv_1}(\theta_1^*)} = 0.143, \\ \tilde{p}_2 = \dfrac{e^{\theta_2^* t_{2,1}} p}{M_{serv_2}(\theta_2^*)} = 0.145. \end{cases} \tag{6.54}$$

(i.e., in the new system small packets have only around 14% of the traffic). The new utilizations (of the system under the change of measure) are equal to

$$\begin{cases} \tilde{\rho}_1 = 1.874, \\ \tilde{\rho}_2 = 1.859. \end{cases} \tag{6.55}$$

Thus,

$$COM_1 = (\tilde{\lambda}_1, \tilde{p}_1, p) = (14674, 0.143, 0.4), \tag{6.56}$$

$$COM_2 = (\tilde{\lambda}_2, p, \tilde{p}_2) = (14419, 0.4, 0.145), \tag{6.57}$$

$$COM_0 = (\lambda, p, p) \quad = (5000, 0.4, 0.4), \tag{6.58}$$

and the **COM**$^p$ for simulating the whole system is found through Equation (6.29) for $COM^p_{x_2}$ and through Equation (6.29) for $COM^p_{x_1,x_2}$. The **COM**$^\theta$ is defined from the set of Equations (6.42) together with Equation(6.40) for $COM^\theta_{x_2}$ and Equation(6.41) for $COM^\theta_{x_1,x_2}$ with

$$COM_1 = (-9674, 9674, 0), \tag{6.59}$$

$$COM_2 = (-9419, 0, 9419), \tag{6.60}$$

$$COM_0 = (0, 0, 0). \tag{6.61}$$

| **N** | $\boldsymbol{COM^{p}_{x_1,x_2}}$ | | $\boldsymbol{COM^{\theta}_{x_1,x_2}}$ | |
|---|---|---|---|---|
| | $b_{opt}$ | $\tilde{\gamma}(N) \pm RE\%$ | $b_{opt}$ | $\tilde{\gamma}(N) \pm RE\%$ |
| 10 | 4 | 9.1776e-04 ± 0.14 | 4 | 9.1795e-04 ± 0.15 |
| 25 | 4 | 2.5937e-10 ± 0.17 | 5 | 2.5882e-10 ± 0.19 |
| 50 | 5 | 1.3566e-21 ± 0.21 | 5 | 1.3527e-21 ± 0.22 |
| 100 | 6 | 1.9320e-44 ± 0.27 | 6 | 1.9347e-44 ± 0.29 |

Table 6.1: Model 1

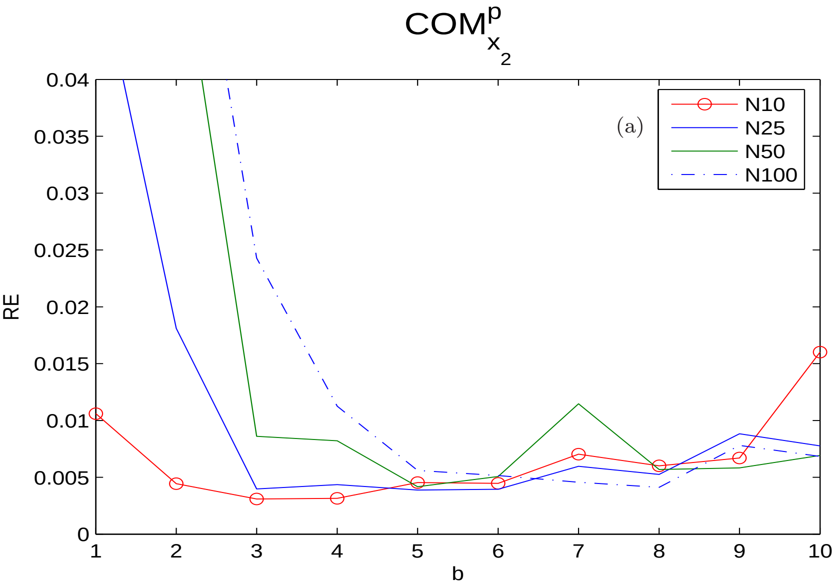
### Results description

In Figures 6.4–6.5 the simulation results are presented for four different values of the overflow level $N$. The dependence of the relative error ($RE$) of the estimator on the parameter $b$ is shown for all four proposed heuristics. The results are gathered with $10^5$ replications.

The purpose of including the figures is to show and (also check) the stability of the heuristics. As one can see, all of them give small $RE$s but the heuristics with the dependence only on the number of customers at node 2 are clearly less stable, hence, less reliable. At the same time the heuristics that depend on the number of customers at both nodes, i.e., $COM^{p}_{x_1,x_2}$ and $COM^{\theta}_{x_1,x_2}$ are clearly very stable. Moreover, there is one $b = b_{opt}$ for each level, at which the minimum $RE$ is achieved.

In Figure 6.6 the results for $COM^{p}_{x_1,x_2}$ and $COM^{\theta}_{x_1,x_2}$ (two best performing changes of measure) with more replications ($16 \cdot 10^5$) are shown. One can see that they perform almost the same, though the first one is a bit more smooth for the overflow level $N = 100$. One can also see that $RE$ of an estimator for low level $N = 10$ is not very sensitive with respect to $b$. Starting from $b = 3$ $RE$ changes very slow with $b$. At the same time, the curves of $RE$ start to be more “deep” near the minimum for higher levels $N$, which means that it is more important for the efficiency of the heuristics to choose the right $b$. It is also interesting to note, that $RE$ is more sensitive to the lower than to the higher than $b_{opt}$ values of $b$, i.e., for $b < b_{opt}$, $RE$ grows very fast when $b$ decreases whereas for $b > b_{opt}$, $RE$ is slowly growing with increasing $b$.

Table 6.1 shows estimated overflow probabilities obtained with $16 \cdot 10^5$ replications for $COM^{p}_{x_1,x_2}$ and $COM^{\theta}_{x_1,x_2}$. The relative errors, shown in percent, are very small, and grow less than linearly with the overflow level $N$.



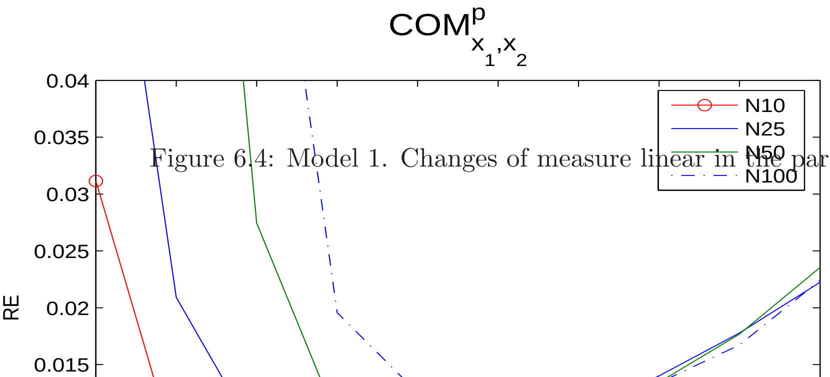


Figure 6.4: Model 1. Changes of measure linear in the parameters. $10^5$ replications

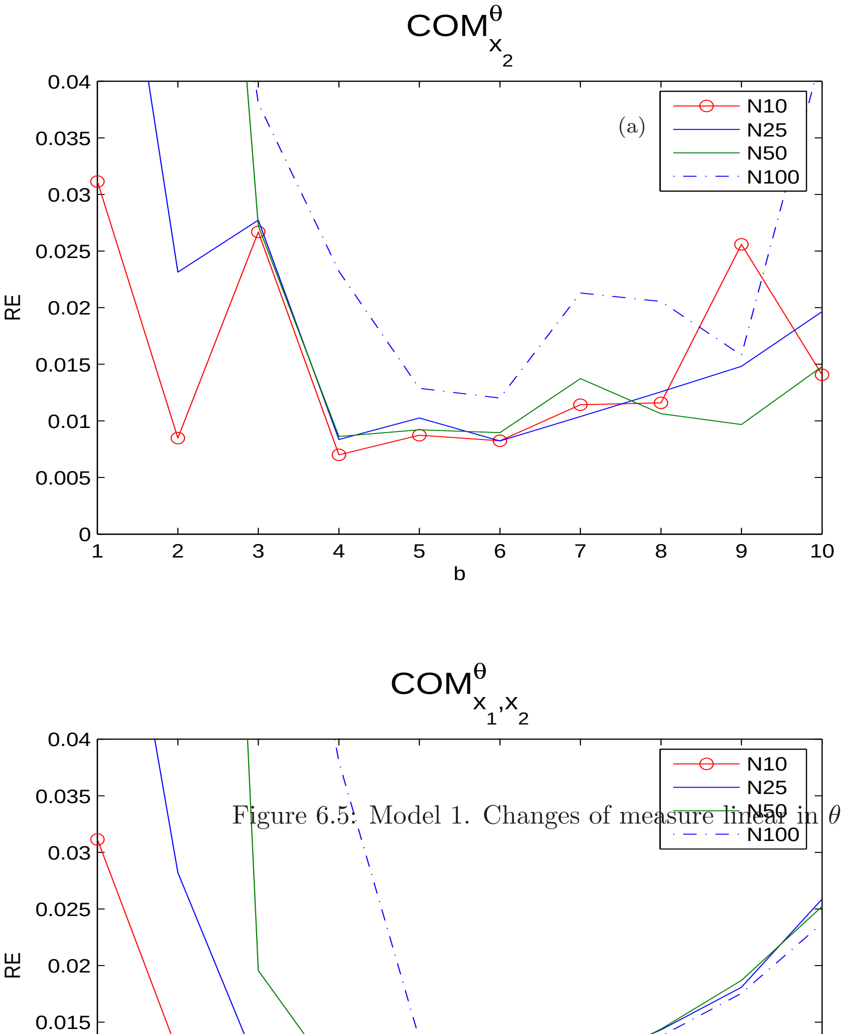


Figure 6.5: Model 1. Changes of measure linear in $\theta$. $10^5$ replications

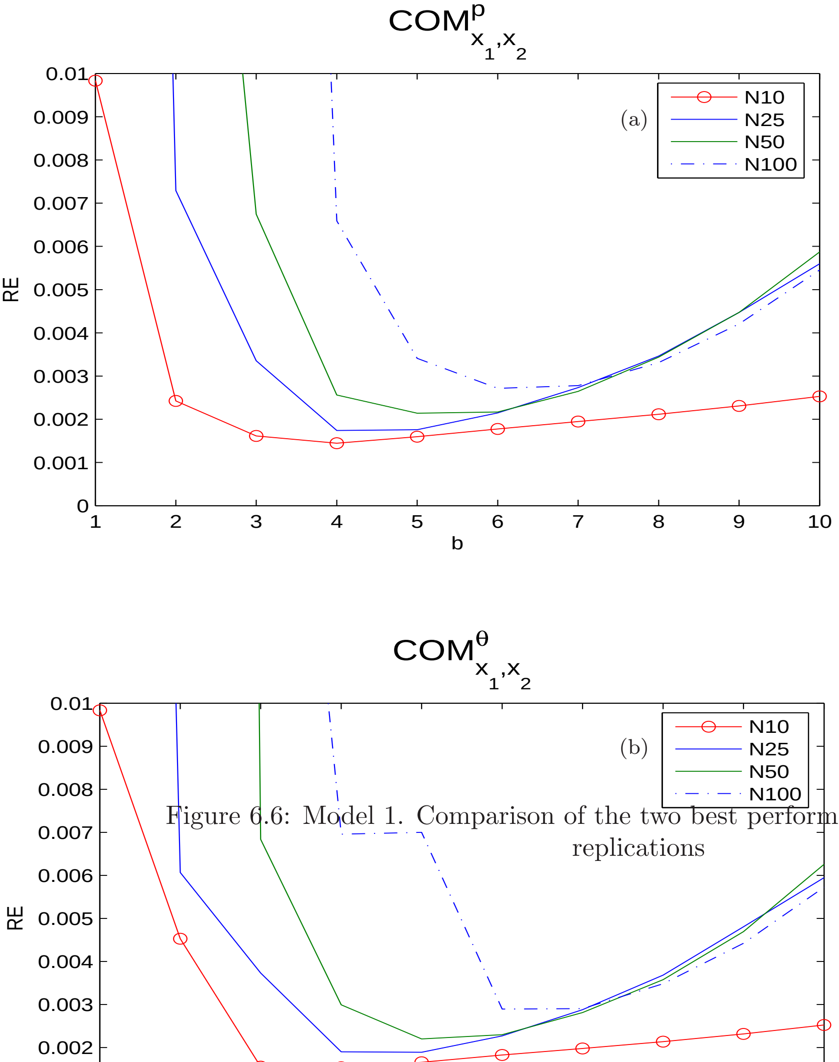


Figure 6.6: Model 1. Comparison of the two best performing COMs with $16 \cdot 10^5$ replications

### 6.2.8 Experimental results for Model 2: bursty arrivals and bi-modal service rates

In this section we consider the experimental results for Model 2 described in Section 6.2.1. For each example we compare the performance of all four changes of measure. For changes of measure dependent on both, $x_1$ and $x_2$, i.e., for $COM^p_{x_1,x_2}$ and $COM^\theta_{x_1,x_2}$, we choose $b_1 = b_2$.

**Network parameters**

Consider the example of Model 2 with the parameters

$$\begin{cases} \lambda_1 = 200 \text{ packets/s}, \\ \lambda_2 = 4000 \text{ packets/s}, \\ q \ \ = 0.1, \\ l_1 \ = 400 \text{ bits}, \\ l_2 \ = 12000 \text{ bits}, \\ p \ \ = 0.3, \\ \nu_1 = 9.01 \cdot 10^7 \text{ bit/s}, \\ \nu_2 = 9 \cdot 10^7 \text{ bit/s}. \end{cases} \tag{6.62}$$

Hence, the service time of a packet of type $k$ at node $i$ is $t_{i,k} = l_k/\nu_i$ (Equation 6.31) with

$$\begin{cases} t_{1,1} = 4.440 \cdot 10^{-6} \text{ sec}, \\ t_{1,2} = 1.332 \cdot 10^{-4} \text{ sec}, \\ t_{2,1} = 4.444 \cdot 10^{-6} \text{ sec}, \\ t_{2,2} = 1.333 \cdot 10^{-4} \text{ sec}. \end{cases} \tag{6.63}$$

The system utilizations are equal to $\rho_i = (\lambda_1 q + \lambda_2(1-q)) \cdot (pt_{i,1} + (1-p)t_{i,2})$, which yields

$$\begin{cases} \rho_1 = 0.34231, \\ \rho_2 = 0.34269. \end{cases} \tag{6.64}$$

**Calculation of COMs**

The numerical solutions of Equation (6.27) (rounded to integer values for simplicity) for nodes 1 and 2 are, respectively,

$$\begin{cases} \theta_1^* = 14376, \\ \theta_2^* = 14347. \end{cases} \tag{6.65}$$

The new arrival and service parameters for linear in parameters change of measure are found from Equations (6.43)–(6.45) and are equal to

$$\begin{cases} \tilde{\lambda}_{1,1} = \lambda_1 + \theta_1^* = 14576 \text{ (bits/s)}, \\ \tilde{\lambda}_{1,2} = \lambda_2 + \theta_1^* = 18376 \text{ (bits/s)}, \\ \tilde{\lambda}_{2,1} = \lambda_1 + \theta_2^* = 14547 \text{ (bits/s)}, \\ \tilde{\lambda}_{2,2} = \lambda_2 + \theta_2^* = 18347 \text{ (bits/s)}, \\ \tilde{q}_1 \quad = 0.007, \\ \tilde{q}_2 \quad = 0.007, \\ \tilde{p}_1 \quad = \dfrac{e^{\theta_1^* t_{1,1}} p}{M_{serv_1}(\theta_1^*)} = 0.06308, \\ \tilde{p}_2 \quad = \dfrac{e^{\theta_2^* t_{2,1}} p}{M_{serv_2}(\theta_2^*)} = 0.06318. \end{cases} \tag{6.66}$$

The new utilizations (of the system under the change of measure) are equal to

$$\begin{cases} \tilde{\rho}_1 = 2.295, \\ \tilde{\rho}_2 = 2.294. \end{cases} \tag{6.67}$$

Thus,

$$COM_1 = (\tilde{\lambda}_{1,1}, \tilde{\lambda}_{1,2}, \tilde{q}_1, \tilde{p}_1, p) = (14576, 18376, 0.007, 0.0631, 0.3), \tag{6.68}$$

$$COM_2 = (\tilde{\lambda}_{2,1}, \tilde{\lambda}_{2,2}, \tilde{q}_2, p, \tilde{p}_2) = (14547, 18347, 0.007, 0.3, 0.0632), \tag{6.69}$$

$$COM_0 = (\lambda_1, \lambda_2, q, p, p) \qquad = (200, 4000, 0.1, 0.3, 0.3), \tag{6.70}$$

and $\mathbf{COM}^p$ for simulating the whole system is found through Equation (6.29) for $COM^p_{x_2}$ and through Equation (6.29) for $COM^p_{x_1,x_2}$.

### Results description

As for Model 1 we considered four different values of the overflow level $N$ for all four proposed heuristics. Figures 6.7–6.8 show the dependence of a relative error of an estimator ($RE$) on a parameter $b$. The simulation results are gathered for $10^5$ replications.

One can see from the figures, that, as for Model 1, the heuristics with the dependence only on the number of customers at node 2 are less stable. At the same time the heuristics with dependence on the number of customers at both nodes, i.e., $COM^p_{x_1,x_2}$ and $COM^\theta_{x_1,x_2}$, are clearly very stable, especially $COM^p_{x_1,x_2}$, even for the small number of replications ($10^5$). Again, there is one $b = b_{opt}$ for each level $N$ at which the minimum $RE$ is achieved.

In Figure 6.9 the results for the two best performing changes of measure ($COM^p_{x_1,x_2}$ and $COM^\theta_{x_1,x_2}$) with more replications ($16 \cdot 10^5$) are presented. One can see that they perform almost the same. The behavior of $RE$ is very similar to the one observed for Model 1. Namely, $RE$ of an estimator is not very sensitive with

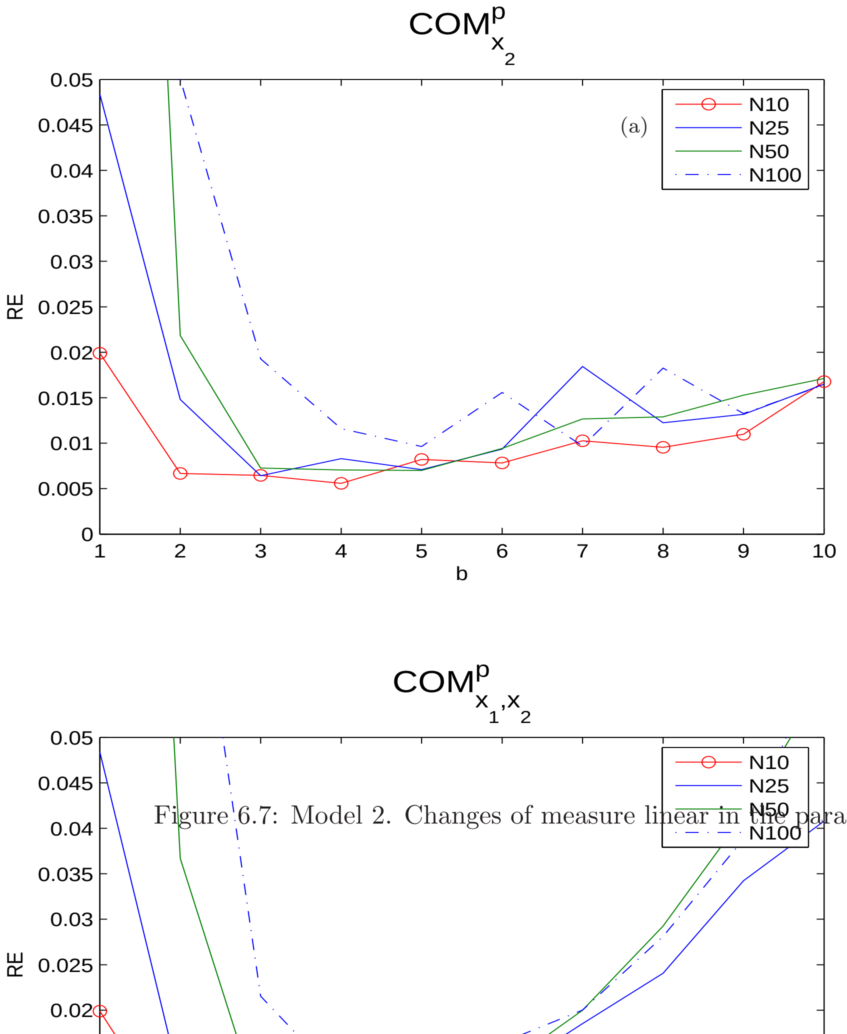
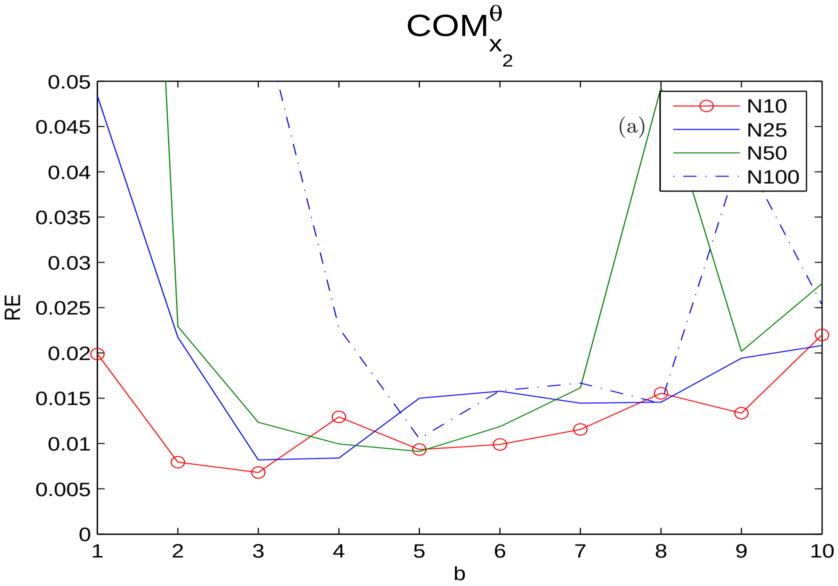


Figure 6.7: Model 2. Changes of measure linear in the parameters. $10^5$ replications



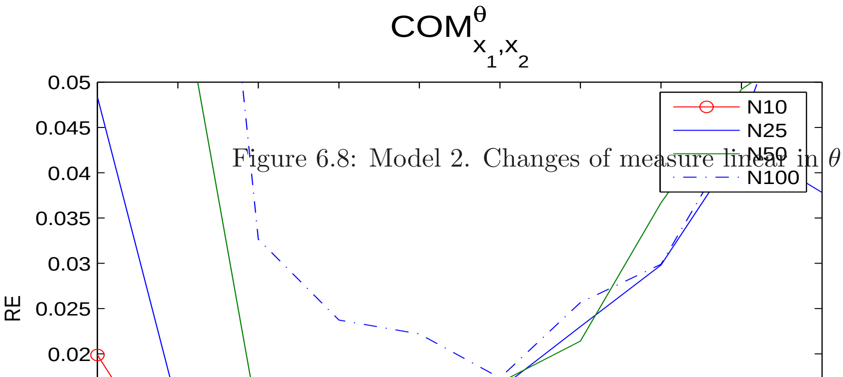


Figure 6.8: Model 2. Changes of measure linear in $\theta$. $10^5$ replications

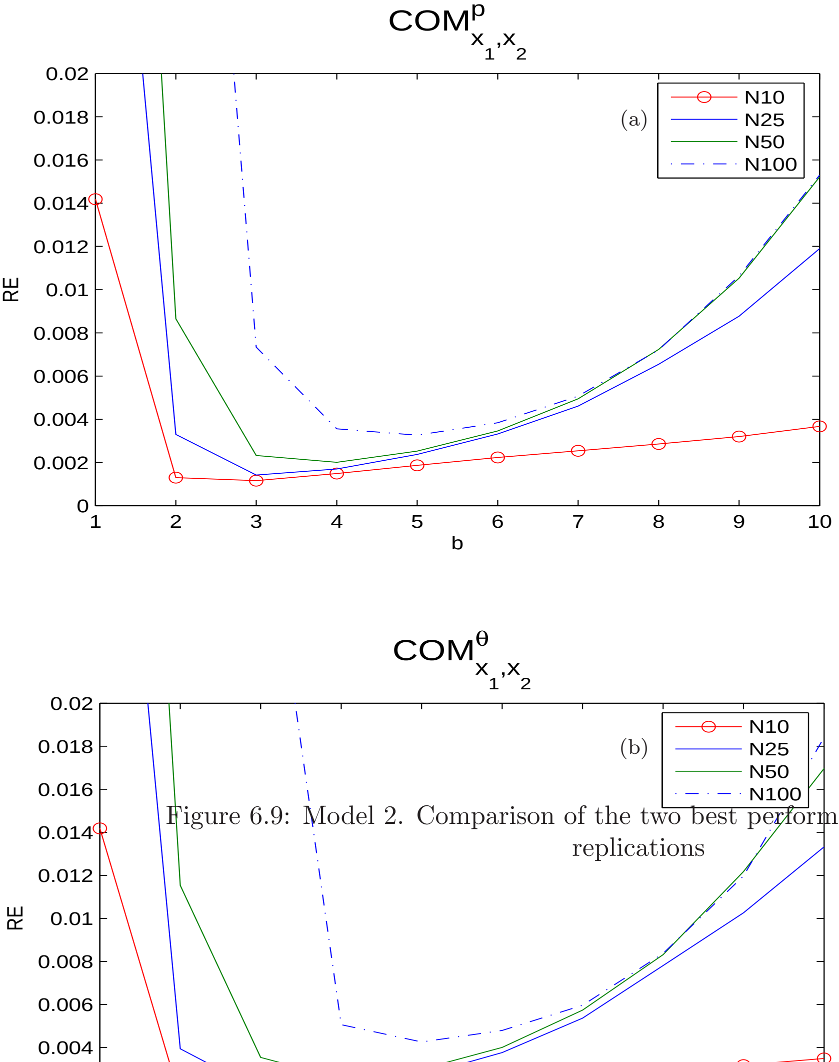


Figure 6.9: Model 2. Comparison of the two best performing COMs with $16 \cdot 10^5$ replications

| **N** | $\boldsymbol{COM^{p}_{x_1,x_2}}$ | | $\boldsymbol{COM^{\theta}_{x_1,x_2}}$ | |
|---|---|---|---|---|
| | $b$ | $\tilde{\gamma}(N) \pm RE\%$ | $b$ | $\tilde{\gamma}(N) \pm RE\%$ |
| 10 | 3 | 1.3068e-05 ± 0.12 | 3 | 1.30985e-05 ± 0.14 |
| 25 | 3 | 8.1645e-16 ± 0.14 | 3 | 8.11988e-16 ± 0.20 |
| 50 | 4 | 3.8084e-33 ± 0.20 | 4 | 3.81196e-33 ± 0.25 |
| 100 | 5 | 4.4188e-68 ± 0.32 | 5 | 4.45316e-68 ± 0.43 |

Table 6.2: Model 2

respect to $b$ for low level $N = 10$, and starts to be more sensitive when level $N$ increases. Again, $RE$ is more sensitive to the lower than to the higher than $b_{opt}$ values of $b$.

Table 6.2 shows estimated overflow probabilities obtained with $16 \cdot 10^5$ replications for $COM^{p}_{x_1,x_2}$ and $COM^{\theta}_{x_1,x_2}$. The relative errors, shown in percent, are very small, and grow less than linearly with the overflow level $N$. It is very similar to what has been observed for Model 1. Since both models have the same service but different inter-arrival time distributions, this suggests, that there might be no or very little dependence on the inter-arrival time distribution. This is, however, only guess, which yet needs to be checked.

### 6.2.9 Final remarks

In Section 6.2 we have proposed the four state-dependent heuristics to simulate the total population overflow in a 2-node non-Markovian tandem queuing network. Two of them are partly state-dependent, i.e., depend only on the number of customers at node 2, and other two are fully state dependent. Two types of models have been considered and the heuristics have been checked experimentally. For both of the examples the heuristics with full state-dependence showed very good performance even for the worst known network parameters setting, namely, equal service rates. This is a very promising result. More experiments are needed, however, to validate the heuristics for both considered models, and, for other non-Markovian queuing networks.

# Chapter 7

# Conclusions

In this chapter we list the main contributions of the thesis and discuss possibilities to extend the research.

## 7.1 Contributions of the thesis

The main problem studied in this thesis is the estimation of the overflow probability in queuing networks. The estimation was done using importance sampling simulation for different types of networks.

The main result of the thesis is the invention of asymptotically efficient state-dependent heuristics for different types of network topologies. The heuristics have been developed for

- Jackson queuing networks of nodes in tandem,
- Jackson queuing networks of nodes in parallel,
- some specific types of feed-forward Jackson queuing networks,
- a 2-node feedback Jackson queuing network and
- a 2-node non-Markovian tandem queuing network.

Each of the heuristics is parametrized by parameters $b_i$ and there exists some optimal parameter $b_{opt}$ (needs to be found by experiments) for which the heuristic with $b_i = b_{opt}$ (for all network nodes) shows the best performance. In more detail the results were the following.

For tandem networks, networks of nodes in parallel and feed-forward Jackson networks an extensive experimentation for networks of up to four nodes was done. Thus, the heuristics were experimentally validated. The experimental results showed very good performance for all possible settings of network parameters.

The heuristics for a 2-node tandem Jackson network were fully investigated and $b_{opt}$ was found exactly for some network settings, and the scope of possibilities was given for other network settings.

The changes of measure for a 2-node tandem queuing network were investigated further. Two approaches for a theoretical proof of their asymptotic efficiency were considered. It was shown that none of the two can be applied. Despite that, this consideration was useful to gain some understanding in the behavior of changes of measure.

For all types of Jackson queuing networks considered, some examples were chosen (for each of the network type) and the heuristics have been compared with a specific state-dependent adaptive algorithm. For the majority of the cases the heuristics showed similar or better performance. At the same time, they have an advantage of being much simpler and easier to implement compared with the adaptive method. It is very interesting that such simple heuristics, namely, with only linear dependence on the network parameters, work so well where the adaptive algorithm tries to find a good approximation. At the same time, the heuristics do depend, and sometimes, very strongly, on a right choice of a parameter $b$ which needs to be found by experiments. In that sense the adaptive algorithm, which also involves some parameter $b$, is not that sensitive to the right choice of it.

## 7.2 Future work

Below we list several possibilities to extend the research done in this thesis.

The first direction is a theoretical proof of asymptotic efficiency of the proposed heuristics. This is the most valuable but also the most difficult direction.

Another possible direction of research is finding the exact dependency of the parameter $b_{opt}$ on network parameters, i.e., some kind of dependency $b_{opt}(\lambda, \mu_1, .., \mu_d)$. If it is known, one can estimate the probability of total population overflow by applying the heuristics with $b = b_{opt}(\lambda, \mu_1, .., \mu_d)$ and obtain a reliable estimate immediately, with no need to first search for $b_{opt}$.

A promising direction for further investigation is the extension of the heuristic proposed for the specific types of feed-forward network to any type of feed-forward network, and the extension of the heuristic from a 2-node feedback network to a network with any number of nodes. Once this is done, all possible network topologies are covered.

Last but not least, more experiments with non-Markovian networks can be done. First, for a 2-node tandem non-Markovian network to experimentally validate the heuristic proposed in this thesis. Second, for other non-Markovian network topologies to check whether the heuristics proposed for Markovian networks can be extended to the non-Markovian case.

# Appendix A

# Fully state-dependent heuristic for tandem networks

In this appendix we present a fully state-dependent change of measure for a $d$-node Markovian tandem queuing network. This heuristic has been developed during the experiments with a 4-node feed-forward queuing network in Section 5.3.1. As a part of the heuristic for a feed-forward network the fully state-dependent change of measure has shown much better performance than the heuristics developed in Section 3.3.1–3.3.2. The equivalent of the fully state-dependent heuristic has also been applied for a 2-node *non*-Markovian tandem queuing network (Section 6.2.4) and has shown very good performance, as well.

Below we present experimental results on a 2-node Markovian tandem queuing network for different parameters settings. We compare it with the heuristic proposed in Section 3.2.4 and show that the fully state-dependent change of measure is better only for some cases of equal service rates. This fact has been the reason of including it here, and not in Chapter 3.

## Heuristic for a 2-node Markovian tandem network

Let $COM_i$ denote the PW change of measure to "push" node $i$ ($i = 1, 2$) as a single node, and let $COM_0$ correspond to the original network parameters, i.e., no change of measure:

$$COM_0 = (\lambda, \mu_1, \mu_2)\,,$$

$$COM_1 = (\mu_1, \lambda, \mu_2)\,,$$

$$COM_2 = (\mu_2, \mu_1, \lambda)\,.$$

Then, *the fully state-dependent heuristic* $SDH_{x_1,x_2}$ for a 2-node tandem network is given as follows

$$\begin{aligned} SDH_{x_1,x_2} &= \left[\frac{x_2}{b_2}\right]^1 COM_2 + \left[\frac{b_2-x_2}{b_2}\right]^+ \\ &\times \left(\left[\frac{x_1}{b_1}\right]^1 COM_1 + \left[\frac{b_1-x_1}{b_1}\right]^+ COM_0\right), \qquad (A.1) \\ SDH_{0,1} &= (\tilde{\lambda}, \tilde{\mu}_1, 0), \qquad (A.2) \end{aligned}$$

where $b_1$, $b_2$ are integer numbers to be determined and $[a]^1 = \min(a,1)$, $[a]^+ = \max(a,0)$. Equation (A.2) is added to ensure that all cycles during the simulation reach the rare event. For the new arrival and service rates the change of measure means the following.

$$\begin{cases} \tilde{\lambda} = \left[\frac{x_2}{b_2}\right]^1 \cdot \mu_2 + \left[\frac{b_2-x_2}{b_2}\right]^+ \cdot \left(\left[\frac{x_1}{b_1}\right]^1 \cdot \mu_1 + \left[\frac{b_1-x_1}{b_1}\right]^+ \lambda\right), \\ \tilde{\mu}_1 = \left[\frac{x_2}{b_2}\right]^1 \cdot \mu_1 + \left[\frac{b_2-x_2}{b_2}\right]^+ \cdot \left(\left[\frac{x_1}{b_1}\right]^1 \cdot \lambda + \left[\frac{b_1-x_1}{b_1}\right]^+ \mu_2\right), \\ \tilde{\mu}_2 = \left[\frac{x_2}{b_2}\right]^1 \cdot \lambda + \left[\frac{b_2-x_2}{b_2}\right]^+ \cdot \left(\left[\frac{x_1}{b_1}\right]^1 \cdot \mu_2 + \left[\frac{b_1-x_1}{b_1}\right]^+ \mu_2\right), \\ \tilde{\mu}_2(0,1) = 0, \end{cases}$$

Note that the above change of measure is very similar to the one in Section 3.2.3, with the only difference that it also depends on the number of customers at node 1 (hence, the name *fully* state-dependent).

### Performance for a 2-node Markovian tandem network

Section 3.6.1 reported on the comparison of two state-dependent changes of measure proposed in Sections 3.2.3–3.2.4. The variance reduction ratio ($VRR$, Equation (3.30)) was used as a measure of comparison. It was shown (Proposition 1) that SDHI outperforms SDH. Here, we compare the performance of the $SDH_{x_1,x_2}$ change of measure with the SDHI change of measure, i.e., the one that turned out to be the best.

Two sets of experiments have been done, namely, with $b_1 = b_2 = b$ and $b_1 = 1$, $b_2 = b$ with $b$ found during the simulation. Figures A.1–A.2 depict the variance reduction ratios. $VRR > 1$ means that $SDH_{x_1,x_2}$ performs better, otherwise, SDHI is better. One can clearly see from Figure A.1 that for $b_1 = b_2 = b$, $SDH_{x_1,x_2}$ performs worse than SDHI, since for almost all checked parameters $VRR < 1$ . Only for some points near $\mu_1 = 0.35$ for level $N = 100$ (depicted in red dots) $VRR > 1$, but those points correspond to non-rare events (since $\mu_1 \approx 0.35$ and $\mu_1 = \mu_2$, thus $\lambda \approx \mu_1 \approx \mu_2$ and the loads at both nodes are close to one).

The experiments with $b_1 = 1$ (Figure A.2) showed a bit better performance than with $b_1 = b_2$, however, still for most of the cases SDHI outperforms $SDH_{x_1,x_2}$. Only for some parameters with $\mu_1 \approx \mu_2$ (Figure A.2b) $SDH_{x_1,x_2}$ performs better than SDHI. Thus, for a 2-node Markovian tandem network SDHI stays the best performing change of measure.

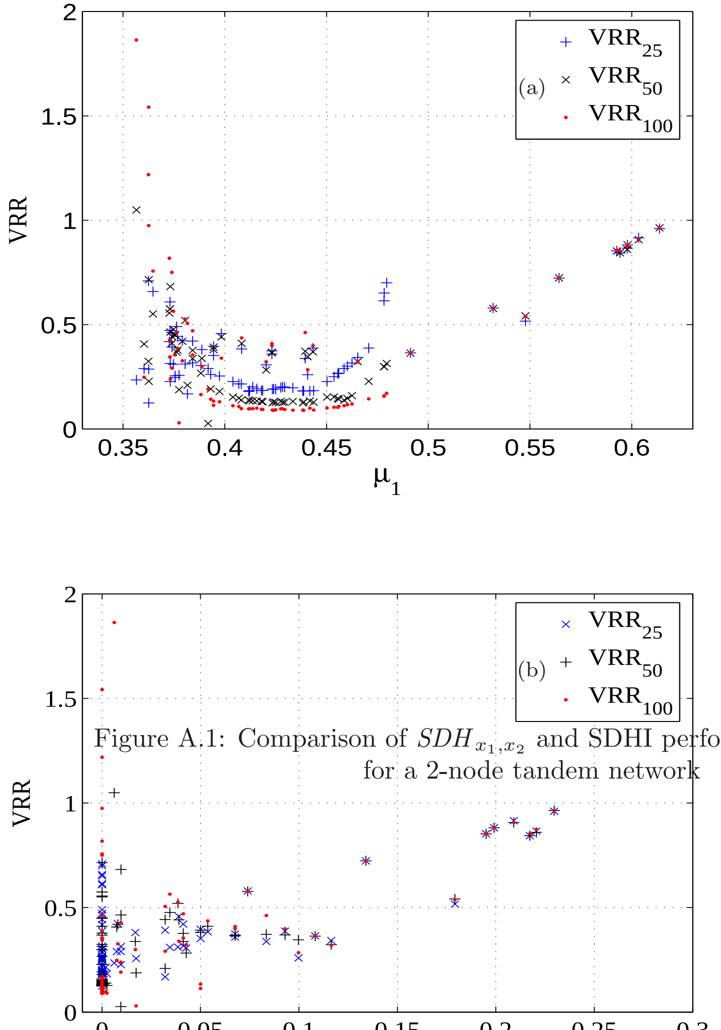


Figure A.1: Comparison of $SDH_{x_1,x_2}$ and SDHI performance ($b_1 = b_2$) for a 2-node tandem network

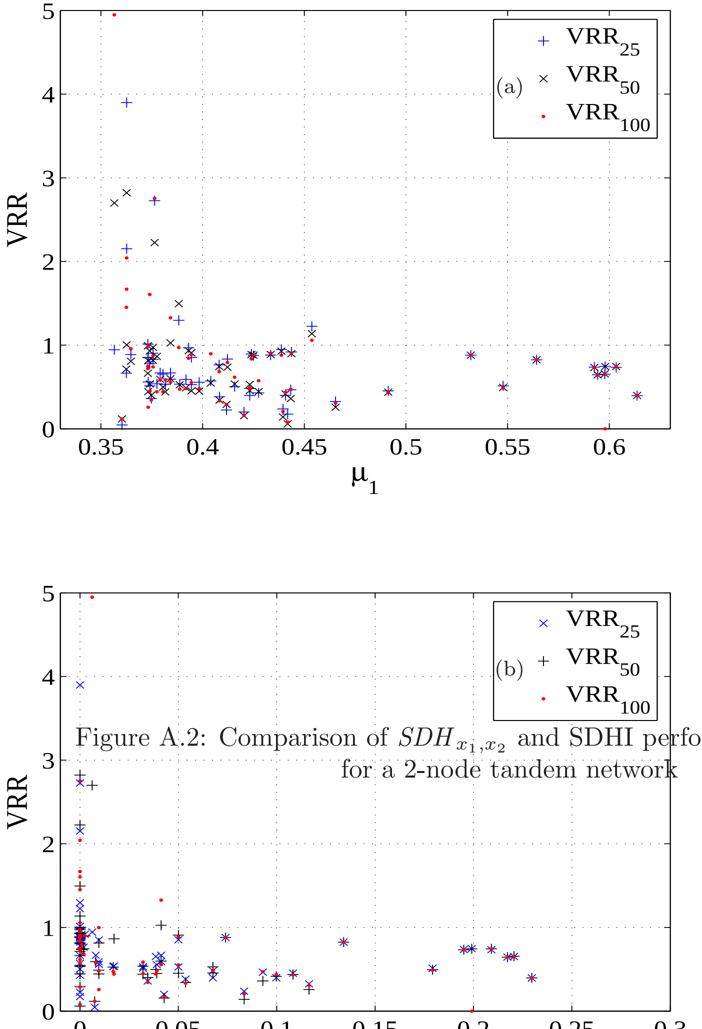


Figure A.2: Comparison of $SDH_{x_1,x_2}$ and SDHI performance ($b_1 = 1$) for a 2-node tandem network

The experiments for more nodes in tandem have not been done. However, below we present the general heuristic for a $d$-node tandem network, since it is has been used to construct the heuristic for a feed-forward network (Section 5.3.1) and might be useful for tandem *non*-Markovian networks.

## Generalization to a $d$-node tandem network

The fully state-dependent heuristic for a $d$-node tandem network can be described as follows:

$$\begin{aligned} SDH_{x_1,x_2} = & \left[\frac{x_d}{b_d}\right]^1 COM_d + \left[\frac{b_d - x_d}{b_d}\right]^+ \\ & \times \left( \left[\frac{x_{d-1}}{b_{d-1}}\right]^1 COM_{d-1} + \left[\frac{b_{d-1} - x_{d-1}}{b_{d-1}}\right]^+ \right. \\ & \dots \\ & \times \left. \left( \left[\frac{x_1}{b_1}\right]^1 COM_1 + \left[\frac{b_1 - x_1}{b_1}\right]^+ COM_0 \right) \dots \right), \end{aligned}$$

where

$$\begin{aligned} COM_0 &= (\lambda, \mu_1, \dots, \mu_d), \\ COM_j &= (\mu_j, \mu_1, \dots, \mu_{j-1}, \lambda, \mu_{j+1}, \dots, \mu_d),\ j = 1, \dots, d. \end{aligned}$$

The parameters $b_j$ ($j = 1, \dots, d$) are integer numbers that need to be found. For $d > 2$ the above fully state-dependent heuristic has only been validated on a 3-node Markovian tandem network as a part of a feed-forward heuristic described in Section 5.3.1. Note, however, that if we define $\left[\frac{x_1}{b_1}\right]^1 = 1$, $\left[\frac{b_1 - x_1}{b_1}\right]^+ = 0$ for $b_1 = 0$, then $SDH_{x_1,x_2} \equiv$ SDH, as proposed in Section 3.3.1 (hence, this case has been validated).

The above heuristic can also be applied for a *non*-Markovian tandem network, in which case $COM_j$, $j = 1, \dots, d$, is found from Equation (6.23). See more discussion in Section 6.2.4.

# Appendix B

# Proof of Observation 4 (Section 6.1.3)

We define *a cycle* ($C$) of length $n$ as a sequence of states $(x_1, x_2) \rightarrow (x_1^{(2)}, x_2^{(2)}) \rightarrow \dots \rightarrow (x_1^{(n)}, x_2^{(n)}) \rightarrow (x_1, x_2)$ such that for all $i = 1, \dots, d$, $(x_1^{(i)}, x_2^{(i)}) \neq (x_1, x_2)$. For each $i = 2, \dots, d-1$ the state $(x_1^{(i-1)}, x_2^{(i-1)})$ is called *a predecessor state* and the state $(x_1^{(i+1)}, x_2^{(i+1)})$ is called *a successor state.* Thus, Figure B.1 represents an example of a cycle, and Figure B.2 represents two cycles connected in one point.

To prove Observation 4 we first show that it is true for all cycles of length six and then use induction to show that it is also true for every cycle of length more than six.

## Cycles of length six

Let us consider state $(x_1, x_2)$. There are only two types of cycles of length six starting from state $(x_1, x_2)$, i.e., a cycle $C_6$ (cf. Figure B.3):

$$\begin{aligned} C_6 : (x_1, x_2) &\rightarrow (x_1 + 1, x_2) \rightarrow (x_1 + 2, x_2) \rightarrow (x_1 + 2, x_2 - 1) \rightarrow \\ &\rightarrow (x_1 + 2, x_2 - 2) \rightarrow (x_1 + 1, x_2 - 1) \rightarrow (x_1, x_2), \end{aligned}$$

and a cycle $C_6'$ (cf. Figure B.4):

$$\begin{aligned} C_6' : (x_1, x_2) &\rightarrow (x_1 - 1, x_2 + 1) \rightarrow (x_1 - 2, x_2 + 2) \rightarrow (x_1 - 2, x_2 + 1) \rightarrow \\ &\rightarrow (x_1 - 2, x_2) \rightarrow (x_1 - 1, x_2) \rightarrow (x_1, x_2) \end{aligned}$$

The corresponding likelihood ratios are equal to

$$LR_{C_6} = \frac{\lambda^2 {\mu_2}^2 {\mu_1}^2}{\tilde{\lambda}^2(\cdot, x_2)\tilde{\mu}_2(\cdot, x_2)\tilde{\mu}_2(\cdot, x_2 - 1)\tilde{\mu}_1(\cdot, x_2 - 2)\tilde{\mu}_1(\cdot, x_2 - 1)},$$

$$LR_{C_6'} = \frac{{\mu_1}^2 {\mu_2}^2 \lambda^2}{\tilde{\mu}_1(\cdot, x_2)\tilde{\mu}_1(\cdot, x_2 + 1)\tilde{\mu}_2(\cdot, x_2 + 1)\tilde{\mu}_2(\cdot, x_2)\tilde{\lambda}^2(\cdot, x_2)}.$$

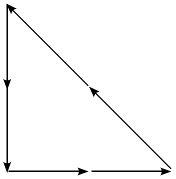


Figure B.1: Cycle

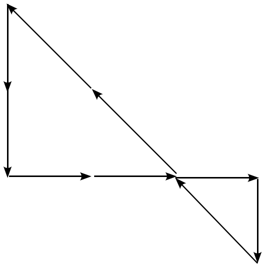


Figure B.2: Two connected cycles

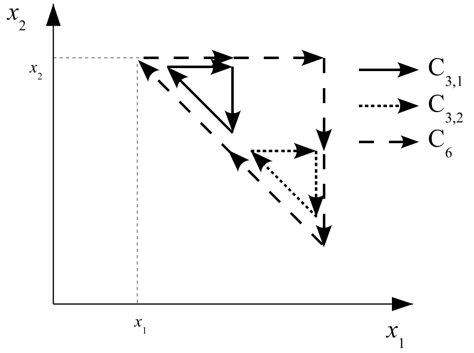


Figure B.3: Cycle $C_6$

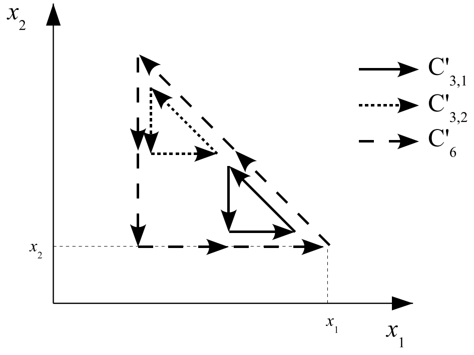


Figure B.4: Cycle $C'_6$

Note that we skip index $x_1$ to make the equations more readable, since both the SDH and SDHI changes of measure depend only on $x_2$, for all states except state (0,1). At state (0,1), $\tilde{\mu}_2(0,1) = 0$, i.e., the transition to state (0,0) is not allowed, therefore, there are no cycles that include this transition.

Now consider cycles $C_{3,i}$ and $C'_{3,i}$ (first index is for the cycle length, second index is for the cycle number):

$$C_{3,1} : (x_1, x_2) \to (x_1+1, x_2) \to (x_1+1, x_2-1) \to (x_1, x_2),$$

$$\begin{aligned} C_{3,2} : (x_1+1, x_2-1) &\to (x_1+2, x_2-1) \to \\ &\to (x_1+2, x_2-2) \to (x_1+1, x_2-1), \end{aligned}$$

and,

$$C'_{3,1} : (x_1, x_2) \to (x_1-1, x_2+1) \to (x_1-1, x_2) \to (x_1, x_2),$$

$$\begin{aligned} C'_{3,2} : (x_1-1, x_2+1) &\to (x_1-2, x_2+2) \to \\ &\to (x_1-2, x_2+1) \to (x_1-1, x_2+1). \end{aligned}$$

The corresponding likelihood ratios are equal to

$$LR_{C_{3,1}} = \frac{\lambda\mu_2\mu_1}{\tilde{\lambda}(\cdot, x_2)\tilde{\mu}_2(\cdot, x_2)\tilde{\mu}_1(\cdot, x_2-1)},$$

$$LR_{C_{3,2}} = \frac{\lambda\mu_2\mu_1}{\tilde{\lambda}(\cdot, x_2-1)\tilde{\mu}_2(\cdot, x_2-1)\tilde{\mu}_1(\cdot, x_2-2)},$$

$$LR_{C'_{3,1}} = \frac{\mu_1\mu_2\lambda}{\tilde{\mu}_1(\cdot, x_2)\tilde{\mu}_2(\cdot, x_2+1)\tilde{\lambda}(\cdot, x_2)},$$

and

$$LR_{C'_{3,2}} = \frac{\mu_1\mu_2\lambda}{\tilde{\mu}_1(\cdot, x_2+1)\tilde{\mu}_2(\cdot, x_2+2)\tilde{\lambda}(\cdot, x_2+1)}.$$

Now one can see that

$$LR_{C_6} = LR_{C_{3,1}} \cdot LR_{C_{3,2}} \cdot \frac{\tilde{\lambda}(\cdot, x_2-1)}{\tilde{\lambda}(\cdot, x_2)},$$

and

$$LR_{C'_6} = LR_{C'_{3,1}} \cdot LR_{C'_{3,2}} \cdot \frac{\tilde{\lambda}(\cdot, x_2+1)}{\tilde{\lambda}(\cdot, x_2)}.$$

For the SDH change of measure, $\tilde{\lambda}(\cdot, x_2) = \mu_1 + (\mu_2 - \mu_1) \cdot x_2/b$. Hence,

$$\frac{\tilde{\lambda}(\cdot, x_2-1)}{\tilde{\lambda}(\cdot, x_2)} > 1,$$

$$\frac{\tilde{\lambda}(\cdot, x_2+1)}{\tilde{\lambda}(\cdot, x_2)} < 1,$$

and

$$LR_{C_6} > LR_{C_{3,1}} \cdot LR_{C_{3,2}}$$

$$LR_{C'_6} < LR_{C'_{3,1}} \cdot LR_{C'_{3,2}}$$

For SDHI change of measure $\tilde{\lambda}(\cdot,\cdot) = \mu_2$. Hence,

$$\frac{\tilde{\lambda}(\cdot, x_2 - 1)}{\tilde{\lambda}(\cdot, x_2)} = 1,$$

and

$$LR_{C_6} = LR_{C_{3,1}} \cdot LR_{C_{3,2}}$$

$$LR_{C'_6} = LR_{C'_{3,1}} \cdot LR_{C'_{3,2}}$$

Thus, we have proved that Observation 4 is true for all cycles of length six. Now, let us do the induction step.

## The induction step

Let us consider a cycle of length $3 \cdot (k+1)$ and suppose that Observation 4 is true for all cycles of length $3 \cdot k$. Cycle length $3 \cdot k$ means that there are exactly $k$ arrivals, $k$ departures from node 1, and $k$ departures from node 2.

Suppose, that the following lemma is true (we use it now and prove it at the end of this appendix).

**Lemma 1.** *For every cycle $C$ there exists a state $(j_1, j_2)$ such that*

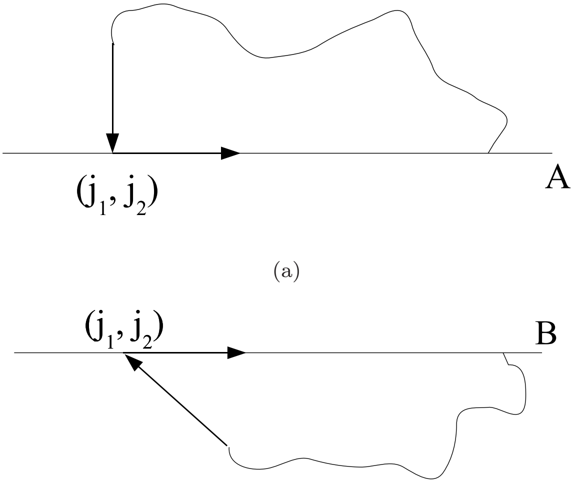
1) *transition $(j_1, j_2) \rightarrow (j_1 + 1, j_2) \in C$, and one of the following two statements is true*

   *(a) $\forall\ x_1, x_2$: $(x_1, x_2) \in C$, $x_2 \leq j_2$, or*

   *(b) $\forall\ x_1, x_2$: $(x_1, x_2) \in C$, $x_2 \geq j_2$;*

2) *state $(j_1, j_2)$ has such a property that*
   *its predecessor state is equal to $(j_1 + 1, j_2 - 1)$ if statement (a) is true, or,*
   *its predecessor state is equal to $(j_1, j_2 + 1)$ if statement (b) is true.*

Informally, the above lemma means the following. If we locate all states of a cycle $C$ at a coordinate plane with a horizontal coordinate $x_1$ and a vertical coordinate $x_2$, then there exists a state $(j_1, j_2) \in C$ with the outgoing transition being an arrival, such that all other states of the cycle $C$ lie on one direction from it with respect to the horizontal line going through this state. This means that either all states of the cycle $C$ are above the line (Figure B.5a), or, all states of the cycle $C$ are below the line (Figure B.5b), and the predecessor state to state $(j_1, j_2)$ is as depicted on Figures B.5a–b.



(a)

(b)

Figure B.5: “Border” state

### Proof of induction step

Let $i$ be the number of consecutive arrivals starting from state $(j_1, j_2)$, i.e., for all $j = j_1, \ldots, j_1 + i - 1$ there is an arrival at state $(j, j_2)$ and at the state $(j_1 + i, j_2)$, there is either departure from node 1 (Figure B.6), or from node 2 (Figure B.7).

Then, the last step of the induction is straightforward. Suppose, that at state $(j_1 + i, j_2)$ the departure is from node 1 (Figure B.6). Then, the cycle enters the state $(j_1, j_2)$ by departure from state $(j_1, j_2 + 1)$ by Lemma 1. Now, if we move all arrival transitions at states $(j, j_2)$ for $j = j_1, \ldots, j_1 + i - 1$ up, i.e., replace all the transitions $(j, j_2) \to (j+1, j_2)$ by $(j, j_2+1) \to (j+1, j_2+1)$, for $j = j_1, \ldots, j_1+i-1$, as shown on Figure B.6 (in red), the cycle of length $3 \cdot (k + 1)$ becomes a cycle of length $3 \cdot (k + 1) - 3 = 3 \cdot k$, for which, by induction, we suppose that Observation 4 is true. Thus, the likelihood ratio is multiplied by the ratio $\tilde{\lambda}(j, j_2 + 1)/\tilde{\lambda}(j, j_2)$, $j = j_1, \ldots, j_1 + i - 1$, which is smaller than one for the SDH and equal to one for the SDHI change of measure. Three rates that are left from the original cycle, namely, $\tilde{\lambda}(j_1 + i - 1, j_2)$, $\tilde{\mu}_1(j_1 + i, j_2)$ and $\tilde{\mu}_2(j_1, j_2 - 1)$ can form a cycle of length 3 if we replace $\tilde{\mu}_2(j_1, j_2-1)$ by $\tilde{\mu}_2(j_1+i-1, j_2-1)$ which does not affect the likelihood ratio. On Figure B.6 the transitions that are replaced are shown by dashed lines and the transitions by which they are replaced are shown in red.

**Remark 1.** Note, that outgoing transitions from states $(j, j_2+1)$ for $j = j_1, \ldots, j_1+i-1$ can not be arrivals, since the predecessor state of a state $(j, j_2 + 1)$ can only be the state $(j, j_2 + 2)$ (i.e., $(j, j_2 + 1)$ is entered by a departure from node 2), and the successor state of a state $(j, j_2+1)$ can only be the state $(j-1, j_2+2)$ (i.e., $(j, j_2+1)$ is left by a departure from node 1), otherwise, our cycle would have a smaller subcycle (which contradicts with the definition of a cycle). Note also, that if there are states $(j, j_2 + 1) \in C$ for $j = j_1, \ldots, j_1 + i - 1$ (where a departure from node 2 is followed by a departure from node 1), then, replacing transitions $(j, j_2) \to (j + 1, j_2)$

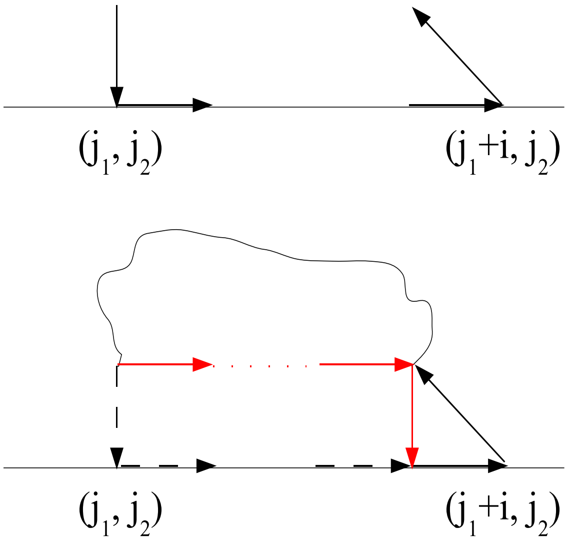


Figure B.6: Consecutive arrivals. Cycle lies above

by $(j, j_2+1) \rightarrow (j+1, j_2+1)$ will make several smaller than $3 \cdot k$ sub-cycles (for which Observation 4 is also true by induction).

Now, suppose, that at the state $(j_1+i, j_2)$ the departure is from *node 2* (Figure B.7). Then, by Lemma 1, the cycle enters the state $(j_1, j_2)$ by departure from state $(j_1+1, j_2-1)$. Similarly, by moving all arrival transitions at states $(j, j_2)$ for $j = j_1+1, \ldots, j_1+i-1$ *down*, i.e., replacing all the transitions $(j, j_2) \rightarrow (j+1, j_2)$ by $(j, j_2-1) \rightarrow (j+1, j_2-1)$ for $j = j_1+1, \ldots, j_1+i-1$ our cycle $3 \cdot (k+1)$ becomes a cycle of length $3 \cdot (k+1) - 3 = 3 \cdot k$ and the likelihood ratio is multiplied by ratio $\tilde{\lambda}(j, j_2-1)/\tilde{\lambda}(j, j_2)$, $j = j_1+1, \ldots, j_1+i-1$, which is larger than one for the SDH and equal to one for the SDHI change of measure. Rates $\tilde{\lambda}(j_1, j_2)$, $\tilde{\mu}_1(j_1+1, j_2-1)$ and $\tilde{\mu}_2(j_1+i, j_2)$ can form a cycle of length 3 if we replace $\tilde{\mu}_2(j_1+i, j_2)$ by $\tilde{\mu}_2(j_1, j_2)$ which does not affect the likelihood ratio. On Figure B.7 the transitions that are replaced are shown by dashed lines and the transitions by which they are replaced are shown in red.

Remark 1 is also valid in another direction, i.e., for states $(j, j_2-1)$, $j = j_1+1, \ldots, j_1+i-1$ (which can be entered only by a departure form node 1 and left by a departure from node 2).

This finishes the induction step and, hence, proves Observation 4. Now we have to prove Lemma 1 that we have already used. We do that below using a proof by contradiction.

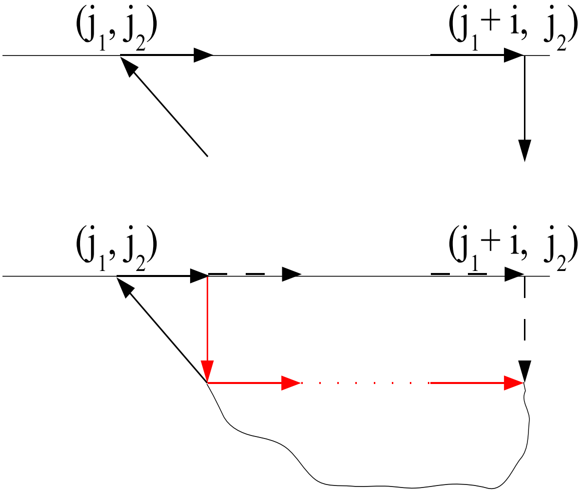


Figure B.7: Consecutive arrivals. Cycle lies below

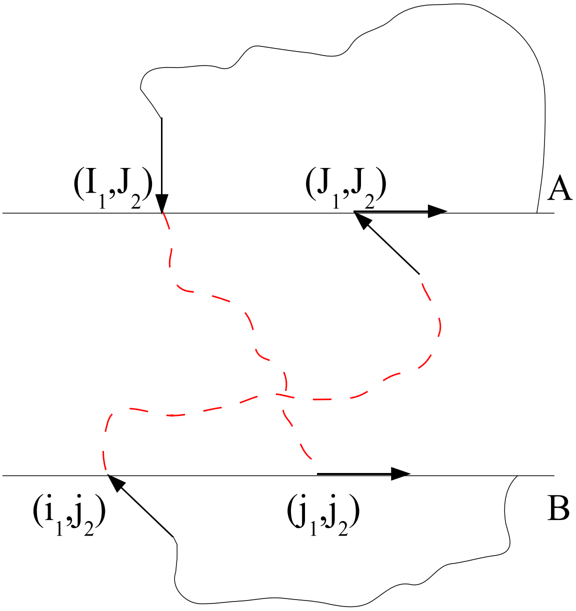


Figure B.8: Behavior of a cycle if Lemma 1 is violated

### Proof of Lemma 1

Suppose that the first statement of the lemma is not true. Then, let $(J_1, J_2)$ and $(j_1, j_2)$ denote, respectively, the state of the cycle with the highest and the lowest $x_2$ value, and such that the outgoing transition from this state is an arrival. Thus, all states from the cycle with value $x_2$ higher than $J_2$ have no outgoing arrival transitions. Similar is true for the state $(j_1, j_2)$, i.e., all states from the cycle with value $x_2$ smaller than $j_2$ have no outgoing arrival transitions.

Let $A$ and $B$ be the horizontal lines going through states $(J_1, J_2)$ and $(j_1, j_2)$, respectively (Figure B.8). If part 1) of Lemma 1 is not true, then there are states from the cycle that lie above the line $A$ and states (from the cycle) that lie below the line $B$. By the definition of the state $(J_1, J_2)$, the outgoing transitions for all states that lie above the line $A$ are departures. The same is true for all states below the line $B$.

Let us consider the line $A$. Every departure from node 1 decreases $J_1$ by 1 and increases $J_2$ by 1, i.e., the cycle goes up and to the left. However, since $J_1$ is finite, at some point in time the departures from node 2 will happen. Hence, the cycle necessarily crosses the line $A$ at some point $(I_1, J_2)$ with $I_1 < J_1$, otherwise, there would be a smaller sub-cycle above the line $A$ (which contradicts with our definition of a cycle). Similar is true for the line $B$. Every departure from node 2 decreases $j_2$ by 1, thus, cycle goes down. Since $j_2$ is finite, at some point the departure from node 1 occurs. Hence, the cycle necessarily crosses the line $B$ at some point $(i_1, j_2)$ with $i_1 < j_1$ (otherwise, there would be a smaller sub-cycle below the line $B$) as depicted in Figure B.8.

Since all the transitions form the cycle, the following states need to be connected by a path, the states $(i_1, j_2)$ and $(J_1, J_2)$, and the states $(I_1, J_2)$ and $(j_1, j_2)$. Hence, since $i_1 < j_1$ and $I_1 < J_1$, these two paths necessarily cross each other. Thus, the cycle has originally consisted of at least two smaller cycles, which contradicts with the definition of a cycle. This proves statement 1) of Lemma 1.

Now, part 2) of Lemma 1 is a direct consequence of part 1). If the state $(j_1, j_2)$ is such that its outgoing transition is an arrival and it has the highest $x_2$ value, then all states of the cycle lie below it (i.e., statement (a) is true) and the only way the cycle can enter it from below is by departure from state $(j_1+1, j_2-1)$, thus, the predecessor state is $(j_1+1, j_2-1)$. Similar is true in case when the state $(j_1, j_2)$ has the lowest $x_2$ value and the outgoing transition is an arrival (i.e., statement (b) is true). Then, all states of the cycle lie above the state $(j_1, j_2)$, and the only way the cycle can enter it from above is by departure from state $(j_1, j_2+1)$, thus, the predecessor state is equal to $(j_1, j_2+1)$. This finishes the proof of Lemma 1.

# Acknowledgments / Благодарности

First of all, I would like to thank Victor Nicola, my first supervisor, with whom I started my journey to the rare event simulation. I am very thankful to him for all the time he spent with me, for all the knowledge he opened to me and that helped in writing this thesis. I also thank him for being not only my supervisor but also a very good friend, for all the moral support in difficult moments of my life.

Due to the circumstances I was finishing to write my diploma under supervision of other people. Though in the beginning this was a bit difficult, this gave me a chance to know and work closely with very competent people from whom I learned a lot. I want to thank Pieter-Tjerk de Boer and Boudewijn Haverkort who directed me in writing the thesis. I want to say vast thanks to Pieter-Tjerk for very accurate and exact remarks and detailed text review, for severe attitude toward writing and for all the time we spent in discussion. They were very useful. I am very thankful to Boudewijn for his remarks and very useful ideas which helped to write what the reader is keeping in hands now. That was a real pleasure to work with all of you.

I also want to thank all my colleagues for a warm atmosphere in the group and all the nice moments we spent together. I want to thank all my friends, in Russia, Holland and other places, who supported me and helped during the difficult moments. And especially, I want to thank my family, my beloved parents who brought me up to become what I am now. They always inspired and encouraged me not to stop on what is achieved. I also want to thank my husband who was always with me when needed, for his love, support and for just being there. And, of course, I want to thank our little charming daughter Elisabeth, our flower, who gives us delight and joy every day.

Thanks to all of you. Without you this would be impossible.

Прежде всего я хотела бы поблагодарить Виктора Николу, моего самого первого научного руководителя, с которым я начала свой путь в дебри моделирования редких событий. Я очень благодарна ему за все то время, что он посвятил мне, за те знания, которые он мне открыл и которые помогли в написании данной работы. Я так же очень благодарна ему за то, что он был не просто моим руководителем, но и хорошим другом, за моральную поддержку в трудные минуты моей жизни.

Так распорядилась судьба, что заканчивать написание дипломной работы мне пришлось под другим руководством. И хотя вначале это было немного трудно,

благодаря этому мне удалось тесно сотрудничать и перенять опыт нескольких очень знающих людей. Я хочу поблагодарить Питер-Черка де Бура и Баудевайна Хаверкорта, которые направляли меня на пути написания дипломной работы. Я хочу сказать огромное спасибо Питер-Черку за очень верные замечания и детальный разбор текста, за требовательный подход к написанию и за все то время, которое мы провели в дискуссиях. Они были очень полезны. Я очень благодарна ему за поддержку, которую он оказал мне в этот трудный период. Огромное спасибо Баудевайну за верные замечания и очень полезные идеи, которые помогли написать то, что читатель держит сейчас в руках. Мне было очень приятно работать с вами.

Я также хочу поблагодарить моих коллег за теплую атмосферу в коллективе и за все приятные моменты, которые мы провели вместе. Я хочу поблагодарить всех моих друзей, в России, Голландии и других странах, которые поддерживали меня и помогали мне в трудные минуты. И особенную благодарность я хочу выразить моей семье, моим любимым родителям, которые воспитали меня и сделали меня тем, что я есть, которые всегда вдохновляли меня и поощряли не останавливаться на достигнутом. Я хочу выразить благодарность моему супругу, который всегда был рядом, когда это было необходимо, за его любовь, поддержку и просто за то, что он есть. И, конечно же, я благодарна нашей маленькой дочурке Элизабэт, нашему цветочку, который радует и восхищает нас каждый день.

Спасибо вам всем, без вас это было бы невозможно!

# Curriculum Vitae

**January, 25 1978** born in Yaroslavl, Soviet Union (Russia).

**September 1985 - June 1995** undergraduate school, Yaroslavl, Russia. Finished with an honored diploma.

**September 1995 - June 1999** Yaroslavl State University, Yaroslavl, Russia. Received bachelor diploma with honor in “Mathematics, Applied Mathematics”.

**September 1999 - June 2001** Yaroslavl State University, Yaroslavl, Russia. Received Master diploma with honor in “Mathematics, Applied Mathematics”.

**August 2001 - November 2007** University of Twente, Enschede, the Netherlands. The author was doing reasearch at Design and Analysis of Communication Systems group (DACS) with a break in research for giving a birth and taking care about a lovely daughter Elisabeth.

# Bibliography

[1] S. Asmussen and R. Y. Rubinstein. Complexity properties of steady-state rare-events simulation in queueing models. In J. Dshalalow, editor, *Advances in Queueing: Theory, Methods and Open Problems*, pages 429–462. CRC Press, 1995.

[2] M. Cottrell, J.-C. Fort, and G. Malgouyres. Large deviations and rare events in the study of stochastic algorithms. *IEEE Transactions on Automatic Control*, 28(9):907–920, 1983.

[3] P. Glynn and D. L. Iglehart. Importance sampling for stochastic simulations. *Management Science*, 35:1367–1392, 1989.

[4] P. Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation*, 5(1):43–85, 1995.

[5] D. P. Kroese and V. F. Nicola. Efficient simulation of a tandem Jackson network. In *The 1999 Winter Simulation Conference*, pages 411–419, New York, NY, USA, 1999. ACM.

[6] D. P. Kroese and V. F. Nicola. Efficient simulation of a tandem Jackson network. *ACM Transactions of Modeling and Computer Simulation*, 12(2):119–141, 2002.

[7] P. Dupuis, A. D. Sezer, and H. Wang. Dynamic importance sampling for queueing networks. *Annals of Applied Probability*, to appear, 2007.

[8] D. Lieber. *The cross-entropy method for estimating probabilities of rare events.* PhD thesis, William Davidson Faculty of Industrial Engineering and Management, Technion, Israel, 1999.

[9] R. Y. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1:127–190, 1999.

[10] P. T. de Boer, V. F. Nicola, and R. Y. Rubinstein. Adaptive importance sampling simulation of queueing networks. In *2000 Winter Simulation Conference*, pages 646–655. IEEE Computer Society Press, 2000.

[11] R. Y. Rubinstein. The cross-entropy method and rare events for maximal cut and bipartition problems. *ACM Transactions of Modeling and Computer Simulation*, 12(1):27–53, 2002.

[12] T. P. I. Ahamed, V. S. Borkar, and S. K. Juneja. Adaptive importance sampling technique for Markov chains using stochastic approximation. *Operations Research*, 54(3):489–504, 2006.

[13] P. T. de Boer and V. F. Nicola. Adaptive state-dependent importance sampling simulation of Markovian queuing networks. *European Transactions on Telecommunications*, 13(4):303–315, 2002.

[14] E. Woudt, P. T. de Boer, and J. K. van Ommeren. Improving adaptive importance sampling simulation of Markovian queueing models using non-parametric smoothing. *Simulation*, to appear, 2007.

[15] S. Parekh and J. Walrand. A quick simulation method for excessive backlogs in networks of queues. *IEEE Transactions on Automatic Control*, 34:54–66, 1989.

[16] M. R. Frater and B. D. O. Anderson. Fast estimation of the statistics of excessive backlogs in tandem networks of queues. *Australian Telecommunications Research*, 23(1):49–55, 1989.

[17] M. R. Frater, T. M. Lennon, and B. D. O. Anderson. Optimally efficient estimation of the statistics of rare events in queueing networks. *IEEE Transactions on Automatic Control*, 36:1395–1405, 1991.

[18] C. S. Chang, P. Heidelberger, S. Juneja, and P. Shahabuddin. Effective bandwidth and fast simulation of ATM in-tree networks. *Performance Evaluation*, 20:45–65, 1994.

[19] G. De Veciana, C. Courcoubetis, and J. Walrand. Decoupling bandwidths for networks: A decomposition approach to resource management for networks. In *The IEEE Conference on Computer Communication (INFOCOM'94)*, pages 466–473, Toronto, 1994. IEEE Computer Society Press.

[20] P. L'Ecuyer and Y. Champoux. Estimating small cell loss ratios in ATM switches via importance sampling. *ACM Transactions of Modeling and Computer Simulation*, 11(1):76–105, 2001.

[21] R. Dhamodaran and B. C. Shultes. Efficient analysis of rare events associated with individual buffers in a tandem Jackson network. In *The 2003 Winter Simulation Conference*, pages 503– 511, Vol.1, 2003.

[22] S. K. Juneja and V. F. Nicola. Efficient simulation of buffer overflow probabilities in Jackson networks with feedback. *ACM Transactions on Modeling and Computer Simulation*, 15(4):281–315, 2005.

[23] J. S. Sadowsky. Large deviations theory and efficient simulation of excessive backlogs in a $GI/GI/m$ queue. *IEEE Transaction on Automatic Control*, 36:1383–1394, 1991.

[24] P. Glasserman and S.-G. Kou. Analysis of an importance sampling estimator for tandem queues. *ACM Transactions on Modeling and Computer Simulation*, 5(1):22–42, January 1995.

[25] P. T. de Boer. Analysis of state-independent importance-sampling measures for the two-node tandem queue. *ACM Transactions on Modeling and Computer Simulation*, 16(3):225–250, 2006.

[26] B. R. Haverkort. *Performance of computer communication systems.* John Wiley and Sons, Ltd, 1998.

[27] M. J. J. Garvels. *The splitting method in rare event simulation.* PhD thesis, University of Twente, 2000.

[28] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. A large deviation perspective on the efficiency of multilevel splitting. *IEEE Transactions on Automatic Control*, 43(12):1666–1679, 1998.

[29] C. Görg and O. Fuß. Comparison and optimization of RESTART run time strategies. *AEU International Journal of Electronics and Communications*, 52:197–204, 1998.

[30] M. Villén-Altamirano and J. Villén-Altamirano. RESTART: A method for accelerating rare event simulation. In J.W.Cohen and C.D.Pack, editors, *The 13th International Teletraffic Congress, Queuing Performance and Control in ATM*, pages 71–76, 1991.

[31] M. Villén-Altamirano and J. Villén-Altamirano. About the efficiency of RESTART. In *Second International Workshop on Rare Event Simulation, RESIM99*, pages 99–128, 1999.

[32] P. T. de Boer. *Analysis and efficient simulation of queueing models of telecommunication systems.* PhD thesis, University of Twente, 2000.

[33] S. R. S. Varadhan. Large deviations and applications, 1984. Philadelphia, Pa.: Society for Industrial and Applied Mathematics. Notes, based on lectures given at the University of Southern Illinois at Carbondale during June 1982.

[34] F. den Hollander. *Large Deviations.* American Mathematical Society, 2000.

[35] J. Lewis and R. Russell. An introduction to large deviations for teletraffic engineers, 1996.

[36] N. O'Connell. Large deviations with applications to telecommunications, 1999. Lecture notes for a course given at Uppsala University.

[37] M. R. Frater and B. D. O. Anderson. Fast simulation of buffer overflows in tandem networks of $GI/GI/1$ queues. *Annals of Operations Research*, 49:207–220, 1994.

[38] V. Anantharam, P. Heidelberger, and P. Tsoucas. Analysis of rare events in continuous time Markov chains via time reversal and fluid approximation, 1990. IBM Research Report RC 16280, Yorktown Heights, New York.

[39] P. E. Heegaard. A scheme for adaptive biasing in importance sampling. *AEÜ International Journal of Electronics and Communications*, 52:172–182, 1998.

[40] M. Devetsikiotis and J. K. Townsend. An algorithmic approach to the optimization of importance sampling parameters in digital communication system simulation. *IEEE Transactions on Communications*, 41:1464–1473, 1993.

[41] M. Devetsikiotis and J. K. Townsend. Statistical optimization of dynamic importance sampling parameters for efficient simulation of communication networks. *IEEE/ACM Transactions on Networking*, 1:293–305, 1993.

[42] W. A. Al-Qaq, M. Devetsikiotis, and J. K. Townsend. Stochastic gradient optimization of importance sampling for the efficient simulation of digital communication systems. *IEEE Transactions on Communications*, 43:2975–2985, 1995.

[43] R. Y. Rubinstein. Rare event simulation via cross-entropy and importance sampling. In *Second International Workshop on Rare Event Simulation, RESIM'99*, pages 1–17, 1999.

[44] F. P. Kelly. *Reversibility and Stochastic Networks.* John Wiley and Sons, New York, 1979.

[45] R. R. Weber. The interchangeability of $\cdot/M/1$ queues in series. *Journal of Applied Probability*, 16:690–695, 1979.

[46] R. S. Randhawa and S. K. Juneja. Combining importance sampling and temporal difference control variates to simulate Markov chains. *ACM Transactions of Modeling and Computer Simulation*, 14(1):1–30, 2004.

[47] P. T. de Boer and W. R. W. Scheinhardt. Alternative proof and interpretations for a recent state-dependent importance sampling scheme. *Queueing Systems: Theory and Applications (QUESTA)*, to appear, 2007.

[48] L. Kleinrock. *Queuing Systems. Volume II: Computer Applications.* John Wiley and Sons, 1976.

[49] P. T. de Boer. Rare-event simulation of non-Markovian queueing networks using a state-dependent change of measure determined using cross-entropy. *Annals of Operations Research*, 134(1):69–100, 2005.

[50] V. F. Nicola and T. S. Zaburnenko. State-dependent importance sampling heuristic for simulating population overflow in tandem networks. In *The International Workshop on Rare Event Simulation (RESIM'04)*, Budapest, Hungary, 2004.

[51] V. F. Nicola and T. S. Zaburnenko. Importance sampling simulation of population overflow in two-node tandem networks. In *The 2nd International Conference on the Quantitative Evaluation of Systems (QEST'05)*, pages 220–229, Torino, Italy, 2005.

[52] V. F. Nicola and T. S. Zaburnenko. Efficient importance sampling heuristics for the simulation of population overflow in Jackson networks. In *The 2005 Winter Simulation Conference (WSC 2005)*, pages 538–546. IEEE Computer Society Press, 2005.

[53] T. S. Zaburnenko and V. F. Nicola. State-dependent importance sampling heuristic for simulating rare events in tandem networks. In *The 5th St. Petersburg Workshop on Simulation (SPWS '05)*, pages 755–764, St. Petersburg, Russia, 2005.

[54] V. F. Nicola and T. S. Zaburnenko. Efficient heuristics for the simulation of population overflow in series and parallel queues. In *The 1st International Conference on Performance Evaluation Methodolgies and Tools (Valuetools'06)*, volume 180, Pisa, Italy, 2006. ACM.

[55] V. F. Nicola and T. S. Zaburnenko. Efficient simulation of population overflow in parallel queues. In *The 2006 Winter Simulation Conference (WSC 2006)*, pages 398–403, Monterey, California, USA, 2006.

[56] V. F. Nicola and T. S. Zaburnenko. Efficient importance sampling heuristics for the simulation of population overflow in feed-forward queueing networks. In *The 6th International Workshop on Rare Event Simulation (RESIM'06)*, pages 144–152, Otto-Friedrich University, Bamberg, Germany, 2006.

[57] T. S. Zaburnenko and V. F. Nicola. Efficient heuristics for simulating population overflow in parallel networks, (extended abstract). In *The 2006 Russian-Scandinavian Symposium on Probability Theory and Applied Probability*, pages 86–93, Petrozavodsk, Russia, 2006.

[58] V. F. Nicola and T. S. Zaburnenko. Efficient importance sampling heuristics for the simulation of population overflow in Jackson networks. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 17(2), 2007.

# Index

- adaptive
  - algorithm, 2, 5, 15, 144
  - method, 2, 14, 17, 32, 69, 144
- AE region, *see* region, AE
- algorithm
  - adaptive, *see* adaptive, algorithm
  - numerical, 32, 39, 40, 74, 79, 102
- approximation, 15, 16, 40, 41, 113, 144
- asymptotic efficiency, 3, 9, **9**, 13, 39, 113, 144

- "big node", 93, 94, 97, **97**
- boundary layer, 16, 28, 32
- boundary level, **24**, 69, 73
- BRE region, *see* region, BRE
- busy cycle, 35, 91

- change of measure, 2, **7**, 11, 17
  - exponential, *see* twisted, exponentially
  - optimal, **8**, 11
  - state-dependent, *see* state-dependent, change of measure
  - state-independent, *see* state-independent, change of measure
  - zero-variance, *see* zero-variance
- $\mathbf{COM}^p$, 127
- $\mathbf{COM}^\theta$, 127
- $COM_{x_2}$, $COM_{x_1,x_2}$, 126
- convergence, 2, 15, 16, 40, 101, 102
  - quadratic, **32**, 33, 41
  - rate, 10
- cross-entropy, **14**
  - method, *see* method, cross-entropy

- Discrete Time Markov Chain (DTMC), 125

- efficiency gain, **31**, 35, 40, 41, 74, 102
- ERE region, *see* region, ERE
- estimator, 2, 6, **8**, 31
  - asymptotic behavior, 9
  - asymptotically efficient, **9**, 20
  - efficient, 31
  - stability, 46
  - stable, *see* stable estimator
  - unbiased, 7, 8, 30
  - variance, 6, 8, **8**, 14, 108
- exponential twist, *see* twisted, exponentially

- feed-forward, 92–94, **97**, 143, 145, 149
- feedback, 92, 95–97, 143

- GI/GI/1 queue, 13

- heuristic, 2
  - PW, **12**, 32, 74, 101, 145
  - SDH, *see* SDH
  - SDHI, *see* SDHI
  - state-dependent, *see* state-dependent, heuristic
  - state-independent, *see* state-independent, heuristic

- importance sampling, 2, 90
  - change of measure, 8, 23, 123
  - density, **8**
  - estimator, 30, 31
  - heuristic, 31
  - method, **7**
  - simulation, 30, 143
  - state-dependent, *see* state-dependent, importance sampling

  state-independent, *see* state-independent, importance sampling
indicator function, 5, 7, 14, 30
IRE region, *see* region, IRE

large deviation
  lower bound, 10
  principle, 10
  rate function, 10
  theory (LDT), 10
  upper bound, 10
likelihood ratio, **7**, 8, 12, 30
  behavior, 115, 116
  of a busy cycle, 30
  of a cycle, 115, 116, 151
  of a path, 14, 115, 117
local average, 16, 32

M/M/1 queue, 11, 12
Markov process, 5, 12, 19, 90
method
  adaptive, *see* adaptive, method
  cross-entropy, 10, 14
  importance sampling, *see* importance sampling, method
  Monte Carlo, *see* simulation, Monte Carlo
  splitting, 6, 7
moment generating function, 10, 11, 123–125, 127
most likely path, 11, 13, 21, 22, 25, 72

NAE region, *see* region, NAE
non-Markovian
  change of measure, 125, 126
  model, 123, 124
  network, 3, 113, 144
  simulator, 126
  tandem, *see* tandem, non-Markovian

overflow probability, 12
  arbitrary node, 90
  level, 34
  population, **20**, 73, 90, *see* total population
  single buffer, 13
  total network, 2, 89, 114
  total population, 2, 13

parallel, 12, 71, 72, 93, 94, 97, 143
  asymmetric, 74
  reverse time process, 72
  symmetric, 74
performance comparison, 30
"push", 24

rare event, 2, 5, 7
  probability, 1, 5, **7**
  set, 7
  simulation, 5, 6
  state, 6
region
  AE, 101
  BRE, 34, 39, 47, 49, 54
  ERE, 34, 39, 46, 49, 54
  IRE, 34, 39, 46, 49, 54
  NAE, 39, 98, 101
relative error, 6, 16
  $RE$, **9**, 30, 31, 33, 54, 63, 80, 108, 133
  bounded, **9**, 33, 63, 98, 102
  computed, 39, 79, 102
  exponentially growing, 34
  infinite, 34
  linearly growing, 63, 74, 79, 98
  ratio ($RER$), **54**, 63, 80, 108
relative time variance ($RTV$), **31**, 32, 40, 104
RESTART, 6
restart level, 6
reverse time process, 21, 72
routing probability, 74, 90, 93, 94, 96

sample path, 7, 14
SDA, 32
SDH, 23, 94, 95
$\mathbf{SDH}_i$, 73
$\mathbf{SDH}_d$, 27
SDHI, 25
$\mathbf{SDHI}_d$, 29
$SDH_{x_1,x_2}$, 146
simulation, 1, 6
  cycle, 31, 46
  estimate, 32

  importance sampling, *see* importance sampling, simulation
  Monte Carlo, 5, 17
  non-visited states, 16
  run, 46, 80, 101
  speed up, 2
  technique, 6
  time, 31, 32, 47
smoothing, 16, 101
spline, 16, 32
  fitting, 16, 32
  technique, 40
splitting, *see* method, splitting
stable estimator, **31**, 32, 46
standard deviation, 6
state dependency, 16
state-dependent
  adaptive, *see* SDA
  change of measure, **2**, 9, 13, 20, 32, 34, 73, 74, 94, 95
  fully, 93, 126, 145
  heuristic, 13, 19, 71, 92, 143
  importance sampling, 2
  partly, 93, 126
state-independent
  algorithm, 15
  change of measure, **2**, 9, 34, 90, 93, 96
  distribution, 9
  heuristic, 12, 13, 46, 74
  importance sampling, 2, 12
  rates, 116
  vectors, 94

tandem, 12, 13, 19, 113, 143
  Markovian, 2, 3, 145
  non-Markovian, 2, 124, 143, 149
threshold, 6
twisted, 12
  exponentially, **11**, 90, 125, 126, 128, 129

variance reduction, 8
variance reduction ratio (*VRR*), **31**, 47, 102, 146

weight, 2, 7, 28

zero-variance, 2, 14, 17, 113