Khalid Al-Begain
Dieter Fiems
Jean-Marc Vincent (Eds.)

LNCS 7314

Analytical and Stochastic Modeling Techniques and Applications

19th International Conference, ASMTA 2012
Grenoble, France, June 2012
Proceedings

Springer

Lecture Notes in Computer Science 7314

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Khalid Al-Begain Dieter Fiems
Jean-Marc Vincent (Eds.)

Analytical and Stochastic Modeling Techniques and Applications

19th International Conference, ASMTA 2012
Grenoble, France, June 4-6, 2012
Proceedings

Springer

Volume Editors

Khalid Al-Begain
University of Glamorgan
Faculty of Advanced Technology
Pontypridd, UK
E-mail: kbegain@glam.ac.uk

Dieter Fiems
Ghent University
Department TELIN
Gent, Belgium
E-mail: dieter.fiems@ugent.be

Jean-Marc Vincent
Laboratoire LIG, projet Mescal
Grenoble, France
E-mail: jean-marc.vincent@imag.fr

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-30781-2 e-ISBN 978-3-642-30782-9
DOI 10.1007/978-3-642-30782-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012938994

CR Subject Classification (1998): C.2, D.2.4, D.2.8, D.4, C.4, H.3, F.1

LNCS Sublibrary: SL 2 – Programming and Software Engineering

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

It is our pleasure to present the proceedings of the 19th International Conference on Analytical and Stochastic Modelling and Applications (ASMTA 2012) which was held in Grenoble, France. ASMTA conferences have become established quality events in the calendar of analytical, numerical and even simulation experts in Europe and well beyond. In addition to regular participants from the main centers of expertise from the UK, Belgium, Germany, Belarus, France, Italy, Latvia, Hungary and many other countries, we received newcomers with interesting contributions from other countries such as Korea, Japan and Sweden.

The quality of this year's program was exceptionally high. The conference committee was extremely selective this year, accepting only 20 papers. As ever, the International Program Committee reviewed the submissions critically and in detail, thereby assisting the Program Chairs in making the final decision as well as in providing the authors with useful comments to improve their papers. We would therefore like to thank every member of the Program Committee for their time and efforts.

We are very grateful for the generous support of the Université Joseph Fourier, and their efforts for organizing the conference.

We thank the authors and participants for their contribution to ASMTA 2012.

4 June 2012

Khalid Al-Begain
Dieter Fiems
Jean-Marc Vincent

Organization

Conference Chair

Khalid Al-Begain	University of Glamorgan, UK

Program Chairs

Dieter Fiems	Ghent University, Belgium
Jean-Marc Vincent	Université Joseph Fourier, France

Program Committee

Sergey Andreev	Tampere University of Technology, Finland
Jonatha Anselmi	Basque Center for Applied Mathematics, Spain
Konstantin Avrachenkov	INRIA Sophia-Anitipolis, France
Jeremy Bradley	Imperial College London, UK
Herwig Bruneel	Ghent University, Belgium
Giuliano Casale	Imperial College, UK
Hind Castel	Institut Telecom-Telecom SudParis, France
Koen De Turck	Ghent University, Belgium
Alexander Dudin	Belarusian State University, Belarus
Antonis Economou	University of Athens, Greece
Paulo Fernandes	Pontifical Catholic University of Rio Grande do Sul, Brazil
Jean-Michel Fourneau	University of Versailles, France
Rossano Gaeta	University of Turin, Italy
Marco Gribaudo	Politecnico di Milano, Italy
Yezekael Hayel	Univeristy of Avignon, France
Andras Horvath	University of Turin, Italy
Helen Karatza	Aristotle University of Thessaloniki, Greece
William Knottenbelt	Imperial College, UK
Lasse Leskelä	University of Jyväskylä, Finland
Remco Litjens	TNO ICT, The Netherlands
Andrea Marin	University Ca' Foscari of Venice, Italy
Don McNickle	University of Canterbury, New Zealand
Panayotis Mertikopoulos	Ecole Polytechnique, Paris, France
Yoni Nazarathy	Swinburne University of Technology, Australia
José Niño-Mora	Universidad Carlos III de Madrid, Spain
Antonio Pacheco	Instituto Superior Tecnico, Portugal
Balakrishna Prabhu	Laas-CNRS, France
Matteo Sereno	University of Turin, Italy

Bruno Sericola	INRIA Rennes - Bretagne Atlantique, France
Janos Sztrik	Lajos Kossuth University, Hungary
Miklós Telek	Budapest University of Technology, Hungary
Nigel Thomas	University of Newcastle, UK
Corinne Touati	INRIA Grenoble, France
Dietmar Tutsch	University of Wuppertal, Germany
Kurt Tutschku	University of Vienna, Austria
Benny van Houdt	University of Antwerp, Belgium
Sabine Wittevrongel	Ghent University, Belgium
Katinka Wolter	Humboldt University of Berlin, Germany
Michele Zorzi	University of Padova, Italy

Table of Contents

Queueing Systems I

Networking Applications

Queueing Systems II

Markov Chains

Stochastic Modelling

Queueing System $MAP/M/N$ as a Model of Call Center with Call-Back Option

Chesoong Kim[1], Olga Dudina[2], Alexander Dudin[2,⋆], and Sergey Dudin[2]

[1] Sangji University, Wonju, Kangwon, 220-702, Korea
[2] Belarusian State University, 4, Nezavisimosti Ave., Minsk, 220030, Belarus
dowoo@sangji.ac.kr, dudina_olga@email.com,
dudin@bsu.by, dudin@madrid.com

Abstract. A multi-server queueing system with a Markovian Arrival Process (MAP), an infinite buffer and impatient customers useful in modeling a call center with a call-back option is investigated. The service time of a customer by a server has an exponential distribution. If all servers are busy at a customer arrival epoch, the customer may leave the system forever or move to the buffer (such a customer is referred to as a real customer), or, alternatively, request for call-back (such a customer is referred to as a virtual customer). During a waiting period, the real customer can be impatient and can leave the system without the service or request for call-back (becomes a virtual customer). An efficient algorithm for calculating the stationary probabilities of system states is proposed. Some key performance measures are calculated. The Laplace-Stieltjes transform of the sojourn time distribution for virtual customers is derived. Some numerical results are presented.

Keywords: Call Center, Call-Back, Markovian Arrival Process, Multi-Server Queueing System.

1 Introduction

A call center is a centralised office used by companies for receiving and servicing their clients' requests by telephone. The call centers are an integral part of the companies whose activities are directly related to the contact with their customers. Under increasing competition among the companies, customer's service is becoming increasingly important, so not only the image but also the profit of the company depends on the effective operation of its call center. The problem of the effective service of a large number of calls with minimal losses of customers and minimal operating costs is of primary importance. This problem can be successfully solved by means of informing the customers about the anticipated delays and providing a so called call-back option. This means that the customer who does not want to wait on the line has an opportunity to leave his (her) phone number and an operator of the call center will contact him (her) for the service

⋆ Corresponding author.

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 1–15, 2012.

later on. This option allows to keep customers' calls, to avoid frustration of the customers, make more smooth load of the operators and increase the effectiveness of their work. As practice shows, in the call centers that do not provide the call-back option some part of the customers' service time is spent on listening the customers' complaints about the long waiting time, while in the call centers with the call-back option the customer has his (her) own choice – to wait for the response of the operator or to leave his (her) phone number for contact and due to this the customers rarely complain. Thus, the use of the call-back option can reduce the average customer's waiting and service time and provide more uniform load of the operators.

Adequate mathematical modeling the call centers leads to substantial increase of their economic efficiency, reduces the maintenance costs and improves the quality of the customers' service. For modeling the call centers, the queueing theory is often used. For the references and the present state-of-art in investigation of the call centers the reader is referred to the survey [1], the papers [2], [3] and references therein.

In this paper we consider a multi-server queueing system with an infinite buffer and impatient customers which can be used for modeling and optimization of the call center with the call-back option. If all operators are busy at a customer arrival epoch, the customer is reported about the current queue length ("visible" queue) and his (her) estimated waiting time, and is requested to decide based on provided information, whether to balk (leave the system permanently without the service), wait on the line, or leave his (her) phone number. In the latter case, an operator of the call center will call back to the customer later when there will be a free operator. Statistic shows that customers that receive information about their place in the buffer or waiting time, are 1.5-2 times more patient, than the customers that do not have such information. As a result, a number of unserviced customers is greatly reduced, therefore consideration of "visible" queue and call-back option is an important point in the modeling the modern call center.

In the papers [4] and [5], an asymptotic analysis of the model of the call center with visible queue and call-back option in case of heavy load for a large number of operators is presented. In our paper, we present an exact analytical analysis of the model without the restriction that the number of operators is enough large. We consider more general customers' arrival process (what allows to take into account bursty nature of flows in modern call centers) and possibility of customers abandonment. Comparing to the paper [6], we also consider more general customers arrival process (for which it is hardly possible to exploit level crossing technique applied in [6]). We allow possibility of leaving the system without the service depending on information about the current queue length. We take into account the additional time necessary to provide the service to customers that choose the call-back option (and possibility that such customers will not accept the offer to get the service) and we provide more complete analysis of characteristics of processing of call back customers.

2 Mathematical Model

The structure of the system under consideration is presented on Figure 1.

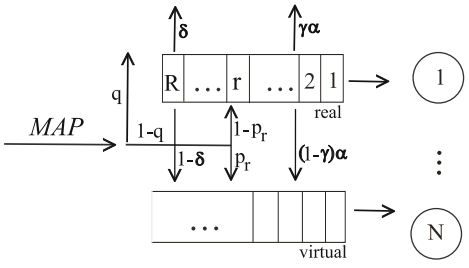


Fig. 1. Structure of the system

The system has N identical operators (servers) and waiting space (buffer, lines).

The customers (calls) arrive to the system according to the MAP. Arrivals can occur at the epochs of transitions of the underlying process ν_t, $t \geq 0$, which is an irreducible continuous time Markov chain with state space $\{0, 1, \ldots, W\}$. The sojourn time of this chain in the state ν is exponentially distributed with the positive finite parameter λ_ν. When the sojourn time in the state ν expires, with probability $p^{(0)}_{\nu,\nu'}$ the process ν_t jumps to the state ν' without a generation of a customer, $\nu, \nu' = \overline{0, W}$, $\nu \neq \nu'$, and with probability $p^{(1)}_{\nu,\nu'}$ the process ν_t jumps to the state ν' with the generation of a customer, $\nu, \nu' = \overline{0, W}$. The notation $\nu = \overline{0, W}$ means that the parameter ν takes the values in the set $\{0, 1, \ldots, W\}$.

The behavior of the MAP is completely characterized by the matrices D_0 and D_1 defined by the entries $(D_0)_{\nu,\nu} = -\lambda_\nu, \nu = \overline{0, W}, (D_0)_{\nu,\nu'} = \lambda_\nu p^{(0)}_{\nu,\nu'}$, $\nu, \nu' = \overline{0, W}, \nu \neq \nu'$, and $(D_1)_{\nu,\nu'} = \lambda_\nu p^{(1)}_{\nu,\nu'}$, $\nu, \nu' = \overline{0, W}$. The matrix $D(1) = D_0 + D_1$ represents the generator of the process $\nu_t, t \geq 0$.

The average arrival rate is given as $\lambda = \boldsymbol{\theta} D_1 \mathbf{e}$, where $\boldsymbol{\theta}$ is the unique solution to the system $\boldsymbol{\theta} D(1) = \mathbf{0}$, $\boldsymbol{\theta}\mathbf{e} = 1$. Here $\mathbf{e}$ is a column vector of appropriate size consisting of 1's and $\mathbf{0}$ is a row vector of appropriate size consisting of zeroes. The squared coefficient of variation of intervals between successive arrivals is given as $c^2_{var} = 2\lambda\boldsymbol{\theta}(-D_0)^{-1}\mathbf{e} - 1$. The coefficient of correlation of two successive intervals between arrivals is given as $c_{cor} = (\lambda\boldsymbol{\theta}(-D_0)^{-1}(D(1) - D_0)(-D_0)^{-1}\mathbf{e} - 1)/c^2_{var}$.

For more information about the MAP see [7].

If, at an arbitrary customer arrival epoch, there is an available server, the customer is admitted to the system and occupies the free server. Otherwise, the customer has the following three possible options: (i) to leave the system without the service, (ii) to become a real customer, i.e., enter the buffer and wait in the system until some server will become available for him (her), or (iii) to become a virtual customer, i.e., to join an infinite size virtual pool of customers and wait until it will be picked up for the service. We assume that both real and virtual

customers are served according to the rule First In - First Out. The difference between two types of customers is that the real customer physically presents in the system (holds a line or occupies a place in a buffer) while the virtual customers just leaves his (her) phone number and an operator will call him (her) and offer the service later on when the system will be not congested. We suggest that the service to a virtual customer is offered only when the queue of real customers is empty and there is a free server.

We assume the following discipline for making a choice among options (i)-(iii). The number of real customers in a buffer can not exceed a certain threshold R. So, if at an arbitrary customer arrival epoch all servers are busy and all R positions in the buffer are occupied, then this customer has only options (i) or (iii). He (she) leaves the system with probability δ or becomes a virtual customer with complementary probability. If, at an arbitrary customer arrival epoch, all servers are busy and there are r, $r = \overline{0, R-1}$, real customers in the buffer, the arriving customer leaves the system with probability q, with complementary probability the customer decides to wait for the service. In latter case he (she) becomes a real customer with probability $1-p_r$, and with probability p_r he (she) becomes a virtual customer. The dependence of probability of joining a buffer on the current number of customers in the buffer realizes the conception of the visible queue, usefulness of which was mentioned in introduction.

The service time at each server has an exponential distribution with the parameter μ, $0 < \mu < \infty$, independently on the type of a customer.

If, at the service completion epoch, there is no real customer in the buffer, a free server starts to dial to the virtual customer placed first into the virtual pool. During a dial time the server is blocked. The dial time has an exponential distribution with the parameter $\tilde{\mu}$, $0 < \tilde{\mu} < \infty$. We assume that the server does not succeed to connect to the customer (the customer's phone is busy or does not answer) with probability h. In this case the virtual customer is considered lost and the blocked server becomes free. With probability $1 - h$ the server connects to the customer and starts the service of the virtual customer.

The real customers are impatient, i.e., the customer leaves the buffer after an exponentially distributed with the parameter α, $0 < \alpha < \infty$, time, conditioned on the fact that this customer is not servicing. In case of leaving the buffer due to impatience, the real customer leaves the system forever with probability γ or becomes a virtual customer with complementary probability.

3 The Process of System States

The behavior of the system under consideration can be described in terms of the regular irreducible continuous-time Markov chain $\xi_t = \{i_t, r_t, n_t, \nu_t\}$, $t \geq 0$, where i_t is the total number of customers in the system, $i_t \geq 0$, r_t is the number of customers in the buffer, $r_t = \overline{0, \max\{0, \min\{i_t - N, R\}\}}$, n_t is the number of servers which process the customers (not accounting the servers that currently are blocked because they dial up to the virtual customers, if any), $n_t = \overline{0, \min\{i_t, N\}}$, ν_t is the state of the underlying process of the MAP, $\nu_t = \overline{0, W}$, at the epoch t, $t \geq 0$.

Lemma 1. *The infinitesimal generator* $Q = (Q_{i,j})_{i,j\geq 0}$ *of the Markov chain* ξ_t, $t \geq 0$, *has the block tridiagonal structure. The non-zero blocks* $Q_{i,j}$, $i, j \geq 0$, *have the following form:*

$$Q_{0,0} = D_0,\ Q_{i,i} = -(\mu C_i + \tilde{\mu}\tilde{C}_i - (1-h)\tilde{\mu}\tilde{C}_i\hat{E}_i) \oplus D_0,\ 1 \leq i < N,$$

$$Q_{N,N} = -(\mu C_N + \tilde{\mu}\tilde{C}_N - (1-h)\tilde{\mu}\tilde{C}_N\hat{E}_N) \oplus (D_0 + qD_1),$$

$$Q_{i,i} = I_{(i-N+1)(N+1)} \otimes (D_0 + qD_1) - I_{i-N+1} \otimes (\mu C_N + \tilde{\mu}\tilde{C}_N - (1-h)\tilde{\mu}\tilde{C}_N\hat{E}_N) \otimes I_{\bar{W}} +$$
$$+(-\alpha C_{i-N} + (1-\gamma)\alpha C_{i-N}E_{i-N}) \otimes I_{(N+1)\bar{W}},\ N < i < N+R,$$

$$Q_{i,i} = Q_1 = I_{(R+1)(N+1)} \otimes D_0 - I_{R+1} \otimes (\mu C_N + \tilde{\mu}\tilde{C}_N - (1-h)\tilde{\mu}\tilde{C}_N\hat{E}_N) \otimes I_{\bar{W}} +$$
$$+\Delta \otimes I_{N+1} \otimes D_1 + (-\alpha C_R + (1-\gamma)\alpha C_R E_R) \otimes I_{(N+1)\bar{W}},\ i \geq N+R,$$

$$Q_{i,i-1} = (\mu C_i E_i^- + h\tilde{\mu}\tilde{C}_i\tilde{E}_i^-) \otimes I_{\bar{W}},\ 1 \leq i \leq N,$$

$$Q_{i,i-1} = \bar{I}_{i-N} \otimes (\mu C_N E_N + h\tilde{\mu}\tilde{C}_N) \otimes I_{\bar{W}} + (E_{i-N}^- - \bar{I}_{i-N}) \otimes (\mu C_N + h\tilde{\mu}\tilde{C}_N\hat{E}_N) \otimes I_{\bar{W}} +$$
$$+\gamma\alpha C_{i-N}E_{i-N}^- \otimes I_{(N+1)\bar{W}},\ N < i \leq N+R,$$

$$Q_{i,i-1} = Q_0 = \hat{I} \otimes (\mu C_N E_N + h\tilde{\mu}\tilde{C}_N) \otimes I_{\bar{W}} +$$
$$+E_R \otimes (\mu C_N + h\tilde{\mu}\tilde{C}_N\hat{E}_N) \otimes I_{\bar{W}} + \gamma\alpha C_R E_R \otimes I_{(N+1)\bar{W}},\ i > N+R,$$

$$Q_{i,i+1} = E_i^+ \otimes D_1,\ 0 \leq i < N,$$

$$Q_{i,i+1} = (1-q)(I_{i-N+1} - \bar{P}_{i-N})E_{i-N}^+ \otimes I_{N+1} \otimes D_1 +$$
$$+(1-q)\bar{P}_{i-N}\tilde{E}_{i-N}^+ \otimes I_{N+1} \otimes D_1,\ N \leq i < N+R,$$

$$Q_{i,i+1} = Q_2 = (\bar{\Delta}\hat{E}_R + \tilde{\Delta}) \otimes I_{N+1} \otimes D_1,\ i \geq N+R,$$

where

I is an identity matrix; $\oplus$ *and* $\otimes$ *are symbols of Kronecker's sum and product respectively, see, e.g., [8];* $\bar{W} = W + 1$;

$C_l = \mathrm{diag}\{0, 1, \ldots, l\}, \tilde{C}_l = \mathrm{diag}\{l, l-1, \ldots, 0\}$, $l = \overline{1, \max\{N, R\}}$;

$\bar{P}_l = \mathrm{diag}\{p_0, p_1, \ldots, p_l\}, l = \overline{0, R-1}$;

$\Delta = \mathrm{diag}\{q, q, \ldots, q, \delta\}$, $\bar{\Delta} = (1-q)\mathrm{diag}\{1-p_0, 1-p_1, \ldots, 1-p_{R-1}, 0\}$, $\tilde{\Delta} = \mathrm{diag}\{(1-q)p_0, (1-q)p_1, \ldots, (1-q)p_{R-1}, 1-\delta\}$;

$E_l^-, \tilde{E}_l^-$, $l = \overline{1, \max\{N, R\}}$, *are the matrices of size* $(l+1) \times l$ *with all zero entries except entries* $(E_l^-)_{0,0}$, $(E_l^-)_{i,i-1}$, $i = \overline{1, l}$, $(\tilde{E}_l^-)_{i,i}$, $i = \overline{0, l}$, *which are equal to 1;*

$E_l^+, \tilde{E}_l^+$, $l = \overline{0, \max\{N, R\} - 1}$, *are the matrices of size* $(l+1) \times (l+2)$ *with all zero entries except entries* $(E_l^+)_{i,i+1}$, $i = \overline{0, l}$, $(\tilde{E}_l^+)_{i,i}$, $i = \overline{0, l-1}$, *which are equal to 1;*

$\hat{E}_l$, $l = \overline{1, \max\{N, R\}}$, *are the square matrices of size* $l+1$ *with all zero entries except entries* $(\hat{E}_l)_{i,i+1}$, $i = \overline{0, l-1}$, *which are equal to 1;*

$E_l, l = \overline{1, \max\{N, R\}}$, *are the square matrices of size* $l+1$ *with all zero entries except entries* $(E)_{i,i-1}$, $i = \overline{1, l}$, *which are equal to 1;*

$\bar{I}_l$ are the matrices of size $(l+1)\times l$ with all zero entries except entry $(\bar{I}_l)_{0,0}$ which is equal to 1;

$\hat{I}$ is the square matrix of size $R+1$ with all zero entries except entry $(\hat{I})_{0,0}$ which is equal to 1.

The proof of Lemma 1 is implemented by means of the analysis of all transitions of the Markov chain $\xi_t, t \geq 0$, during the interval of an infinitesimal length and rewriting intensities of these transitions in the block matrix form.

The Markov chain $\xi_t,\ t \geq 0$, belongs to the class of the continuous time quasi-birth-and-death processes, see, e.g., [9]. It follows from [9] that the ergodicity condition of the quasi-birth-and-death process is the fulfillment of the inequality

$$\boldsymbol{y}Q_0\mathbf{e} > \boldsymbol{y}Q_2\mathbf{e}, \tag{1}$$

where the row vector $\boldsymbol{y} = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_R)$ is the unique solution to the following system of linear algebraic equations

$$\boldsymbol{y}(Q_0 + Q_1 + Q_2) = \mathbf{0},\ \boldsymbol{y}\mathbf{e} = 1. \tag{2}$$

If the dimension of system (2) is small, it can be easily solved on a computer by standard methods. Otherwise, taking into account that the matrix $Q_0+Q_1+Q_2$ has in our case the block tridiagonal structure, to solve this system we can propose the following numerically stable algorithm.

Theorem 1. *The sub-vectors $\boldsymbol{y}_r,\ r = \overline{0,R}$, of the vector $\boldsymbol{y}$ are computed as*

$$\boldsymbol{y}_r = \boldsymbol{y}_{r-1}T_{r-1} = \boldsymbol{y}_0F_r, r = \overline{1,R},$$

where the matrices F_r are calculated using the recurrent formulas

$$F_0 = I,\ F_r = F_{r-1}T_{r-1},\ r = \overline{1,R},$$

the matrices $T_r, r = \overline{0,R-1}$, are calculated using the backward recursion

$$T_r = -\bar{Q}_{r,r+1}(\bar{Q}_{r+1,r+1} + T_{r+1}\bar{Q}_{r+2,r+1})^{-1},\ \ r = R-2, R-3, \ldots, 0,$$

under the initial condition $T_{R-1} = -\bar{Q}_{R-1,R}(\bar{Q}_{R,R})^{-1}$, the vector $\boldsymbol{y}_0$ is the unique solution to the system

$$\boldsymbol{y}_0(\bar{Q}_{0,0} + T_0\bar{Q}_{1,0}) = \mathbf{0},\quad \boldsymbol{y}_0\sum_{r=0}^{R}F_r\mathbf{e} = 1.$$

Here

$$\bar{Q}_{r,r} = -(\mu C_N + \tilde{\mu}\tilde{C}_N - (1-h)\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}} + \delta_{r,0}(\mu C_N E_N + h\tilde{\mu}\tilde{C}_N)\otimes I_{\bar{W}} +$$

$$+I_{N+1}\otimes D_0 + ((1-\delta_{r,R})((1-q)p_r + q) + \delta_{r,R})I_{N+1}\otimes D_1 - r\alpha I_{(N+1)\bar{W}},\ r = \overline{0,R},$$

$$\bar{Q}_{r,r+1} = (1-q)(1-p_r)I_{N+1}\otimes D_1,\ r = \overline{0,R-1},$$

$$\bar{Q}_{r,r-1} = (\mu C_N + h\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}} + r\alpha I_{(N+1)\bar{W}},\ r = \overline{1,R},$$

where $\delta_{i,j}$ is a symbol of Kronecker's delta.

If the ergodicity condition (1) of the Markov chain ξ_t is fulfilled, then the stationary probabilities of the system states exist and are defined as follows:

$$\pi(i,r,n,\nu) = \lim_{t\to\infty} P\{i_t = i,\ r_t = r, n_t = n,\ \nu_t = \nu\},$$

$$i \geq 0,\ r = \overline{0, \max\{0, \min\{i-N, R\}\}},\ n = \overline{0, \min\{i, N\}},\ \nu = \overline{0, W}.$$

Let us form the row vectors $\boldsymbol{\pi}_i$, $i \geq 0$, of the probabilities $\pi(i,r,n,\nu)$, enumerated in the lexicographic order of the components r, n, ν:

$$\boldsymbol{\pi}(i,r,n) = (\pi(i,r,n,0), \pi(i,r,n,1), \ldots, \pi(i,r,n,W)),$$

$$\boldsymbol{\pi}(i,r) = (\boldsymbol{\pi}(i,r,0), \boldsymbol{\pi}(i,r,1), \ldots, \boldsymbol{\pi}(i,r,\min\{i,N\})),$$

$$\boldsymbol{\pi}_i = (\boldsymbol{\pi}(i,0), \boldsymbol{\pi}(i,1), \ldots, \boldsymbol{\pi}(i,\max\{0,\min\{i-N,R\}\})).$$

It is well-known that the probability vectors $\boldsymbol{\pi}_i$, $i \geq 0$, satisfy the following system of linear algebraic equations:

$$(\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_i, \ldots)Q = \mathbf{0},\ \ (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_i, \ldots)\mathbf{e} = 1 \tag{3}$$

where Q is the infinitesimal generator of the Markov chain $\xi_t,\ t \geq 0$.

To solve system (3), the following numerically stable algorithm can be used.

Theorem 2. *The vectors $\boldsymbol{\pi}_i$, $i \geq 0$, are defined as follows*

$$\boldsymbol{\pi}_i = \boldsymbol{\pi}_0 \Phi_i, i \geq 1,$$

where the matrices Φ_i are calculated using the recurrent formulas:

$$\Phi_0 = I,\ \Phi_i = -\Phi_{i-1} Q_{i-1,i} (Q_{i,i} + Q_{i,i+1} G_i)^{-1},\ i = \overline{1, N+R-1},$$

$$\Phi_{N+R} = -\Phi_{N+R-1} Q_{N+R-1,N+R} (Q_1 + Q_2 G)^{-1},$$

$$\Phi_i = -\Phi_{i-1} Q_0 (Q_1 + Q_2 G)^{-1},\ i > N+R,$$

and the vector $\boldsymbol{\pi}_0$ is the unique solution to the system

$$\boldsymbol{\pi}_0 (Q_{0,0} + Q_{0,1} G_0) = \mathbf{0}, \boldsymbol{\pi}_0 \sum_{i=0}^{\infty} \Phi_i \mathbf{e} = 1.$$

The matrices G_i are calculated using the backward recursion

$$G_i = -(Q_{i+1,i+1} + Q_{i+1,i+2} G_{i+1})^{-1} Q_{i+1,i},\ \ i = N+R-1, N+R-2, \ldots, 0,$$

where $G_{N+R} = G$, the matrix G is the minimal nonnegative solution of the matrix equation

$$Q_2 G^2 + Q_1 G + Q_0 = O.$$

Proof. Proof is based on the results of the paper [10] taking into account the special structure of the generator Q.

4 Performance Measures

As soon as the vectors $\boldsymbol{\pi}_i$, $i \geq 0$, have been calculated, we are able to find various performance measures of the system under consideration.

The average number of customers in the system is calculated as $L = \sum_{i=1}^{\infty} i\boldsymbol{\pi}_i\mathbf{e}$.

The average number of (real and virtual) customers, which wait for the service, is calculated as $N^{buffer} = \sum_{i=N+1}^{\infty} (i-N)\boldsymbol{\pi}_i\mathbf{e}$.

The average number of real customers, which wait for the service, is defined as $N_{real}^{buffer} = \sum_{i=N+1}^{\infty} \sum_{r=1}^{\min\{i-N,R\}} r\boldsymbol{\pi}(i,r)\mathbf{e}$.

The average number of virtual customers, which wait for the service, is defined as $N_{virt}^{buffer} = \sum_{i=N+1}^{\infty} \sum_{r=0}^{\min\{i-N,R\}} (i-N-r)\boldsymbol{\pi}(i,r)\mathbf{e}$.

The average number of busy and blocked servers is calculated as $N^{server} = \sum_{i=1}^{\infty} \min\{i,N\}\boldsymbol{\pi}_i\mathbf{e}$.

The average number of busy servers is calculated as

$$N_{busy}^{server} = \sum_{i=1}^{\infty} \sum_{r=0}^{\max\{0,\min\{i-N,R\}\}} \sum_{n=1}^{\min\{i,N\}} n\boldsymbol{\pi}(i,r,n)\mathbf{e}.$$

The average number of blocked servers is calculated as

$$N_{block}^{server} = \sum_{i=1}^{\infty} \sum_{r=0}^{\max\{0,\min\{i-N,R\}\}} \sum_{n=0}^{\min\{i,N-1\}} (\min\{i,N\}-n)\boldsymbol{\pi}(i,r,n)\mathbf{e}.$$

The probability $P_{real}^{esc-loss}$ that an arbitrary real customer arrives when all servers are busy, $r, r < R$, real customers present in the buffer, and this customer does not join the buffer and leaves the system is defined as

$$P_{real}^{esc-loss} = \lambda^{-1} q \sum_{i=N}^{\infty} \sum_{r=0}^{\min\{i-N,R-1\}} \boldsymbol{\pi}(i,r)(I_{N+1} \otimes D_1)\mathbf{e}.$$

The loss probability of an arbitrary real customer at the entrance to the system due to the presence of R real customers in the buffer is calculated as

$$P_{real}^{ent-loss} = \lambda^{-1} \sum_{i=N+R}^{\infty} \boldsymbol{\pi}(i,R)(I_{N+1} \otimes D_1)\mathbf{e}.$$

The probability $P_{real}^{to-virt}$ that an arbitrary customer arrives when all servers are busy, $r, r < R$, real customers present in the buffer, and this customer becomes virtual is defined as

$$P_{real}^{to-virt} = \lambda^{-1}(1-q) \sum_{i=N}^{\infty} \sum_{r=0}^{\min\{i-N,R-1\}} p_r\boldsymbol{\pi}(i,r)(I_{N+1} \otimes D_1)\mathbf{e}.$$

The probability $P_{real}^{imp-loss}$ that an arbitrary real customer arrives when all servers are busy, $r, r < R$, real customers present in the buffer, and this customer will go to the buffer and leave it due to impatience is defined as

$$P_{real}^{imp-loss} = \lambda^{-1}(1-q)\sum_{i=N}^{\infty}\sum_{r=0}^{\min\{i-N,R-1\}}(1-p_r)\sum_{n=0}^{N}\boldsymbol{\pi}(i,r,n)D_1\mathbf{e}(1-z(r+1,n)),$$

where $z(r,n)$ are defined as probabilities that during the waiting time of a real customer in the buffer this customer does not leave the system due to impatience conditioned on the fact that at its arrival epoch there are $r-1$, $r=\overline{0,R-1}$, real customers in the buffer and n customers in the service, $n=\overline{0,N}$.

The probabilities $z(r,n)$ are calculated as follows:

$$z(1,N) = (\alpha+N\mu)^{-1}N\mu,$$

$$z(1,n) = (\alpha+n\mu+(N-n)\tilde{\mu})^{-1}\times$$

$$\times[n\mu+h(N-n)\tilde{\mu}+(1-h)(N-n)\tilde{\mu}z(1,n+1)], n=N-1,N-2,\ldots,0,$$

$$z(r,N) = (r\alpha+N\mu)^{-1}[N\mu+(r-1)\alpha]z(r-1,N),\ r=\overline{2,R},$$

$$z(r,n) = (r\alpha+n\mu+(N-n)\tilde{\mu})^{-1}[(n\mu+(r-1)\alpha)z(r-1,n)+$$

$$+h(N-n)\tilde{\mu}z(r-1,n+1)+(1-h)(N-n)\tilde{\mu}z(r,n+1)], n=N-1,N-2,\ldots,0, r=\overline{2,R}.$$

The loss probability of an arbitrary real customer is calculated as

$$P_{real}^{loss} = P_{real}^{esc-loss} + P_{real}^{ent-loss} + P_{real}^{to-virt} + P_{real}^{imp-loss}.$$

The intensity of flow of customers, which get the service in the system, is calculated as $\lambda^{out} = \mu N_{busy}^{server}$.

The loss probability of an arbitrary customer is calculated as $P^{loss} = 1-\frac{\lambda^{out}}{\lambda}$.

5 Distribution of the Sojourn and Waiting Times of a Virtual Customer

Due to the lack of space, we omit results for distribution of the sojourn time of an arbitrary real customer in the system under study and present only the results for the distribution of the sojourn time of an arbitrary virtual customer.

Let $V_{virt}(x)$ be the distribution function of the sojourn time of an arbitrary virtual customer in the system and $v_{virt}(s) = \int_0^{\infty} e^{-sx}dV_{virt}(x)$, $\mathrm{Re}\ s > 0$, be its Laplace-Stieltjes transform (LST).

Let us tag an arbitrary virtual customer and keep track of its staying in the system. We will derive the expression for the LST $v_{virt}(s)$ by means of the method of collective marks (method of additional event, method of catastrophes) for references, see, e.g., [11], [12]. To this end, we interpret the variable s as the

intensity of some imaginary stationary Poisson flow of catastrophes. So, $v_{virt}(s)$ has the meaning of the probability that no catastrophe arrives during the sojourn time of the tagged virtual customer.

Let $v_{virt}(s, l, r, n, \nu)$ be the the probability that the catastrophe will not arrive during the rest of the tagged customer sojourn time in the system conditioned on the fact that, at the given moment, the tagged customer has the position number $l,\ l \geq 1$, in the system (i.e., there are $l-1$ real or virtual customers that arrived earlier than the tagged customer), the number of the real customers in the buffer is equal to $r,\ r = \overline{0, \min\{l-1, R\}}$, the number of the customers, which are getting the service, is equal to $n,\ n = \overline{0, N}$, and the state of the process $\nu_t,\ t \geq 0$, is ν.

Let us enumerate the probabilities $v_{virt}(s, l, r, n, \nu)$ in the lexicographic order of the component ν and combine them into the column vectors $\mathbf{v}_{virt}(s, l, r, n)$.

Theorem 3. *The LST $v_{virt}(s)$ of the distribution of an arbitrary virtual customer's sojourn time in the system is computed by*

$$v_{virt}(s) = \lambda^{-1}[(1-q)\sum_{i=N}^{\infty}\sum_{r=0}^{\min\{i-N,R-1\}} p_r \sum_{n=0}^{N} \boldsymbol{\pi}(i,r,n) D_1 \mathbf{v}_{virt}(s, i-N+1, r, n) +$$

$$+(1-\delta)\sum_{i=N+R}^{\infty}\sum_{n=0}^{N} \boldsymbol{\pi}(i,R,n) D_1 \mathbf{v}_{virt}(s, i-N+1, R, n)] +$$

$$+(1-\delta_{\gamma,1})(\alpha N_{real}^{buffer})^{-1}\sum_{i=N+1}^{\infty}\sum_{r=1}^{\min\{i-N,R\}} r\alpha \sum_{n=0}^{N} \boldsymbol{\pi}(i,r,n)\mathbf{v}_{virt}(s, i-N+1, r-1, n),$$

where the vectors $\mathbf{v}_{virt}(s, l, r, n)$ can be found from the following system of linear algebraic equations:

$$\mathbf{v}_{virt}(s,l,r,n) = \Big[(s + r\alpha + n\mu + (N-n)\tilde{\mu})I - D_0\Big]^{-1} \times \tag{4}$$

$$\times\Bigg[\delta_{r,0}\Bigg((1-\delta_{l,1})[n\mu\mathbf{v}_{virt}(s,l-1,0,n-1) + h(N-n)\tilde{\mu}\mathbf{v}_{virt}(s,l-1,0,n)] +$$

$$+\delta_{l,1}\mathbf{e}_{\bar{W}}(n\mu + h(N-n)\tilde{\mu})\frac{\tilde{\mu}}{s+\tilde{\mu}}(h + (1-h)\frac{\mu}{s+\mu})\Bigg) +$$

$$+(1-\delta_{r,0})\Bigg((r\alpha+n\mu)\mathbf{v}_{virt}(s,l-1,r-1,n) + h(N-n)\tilde{\mu}\mathbf{v}_{virt}(s,l-1,r-1,n+1)\Bigg) +$$

$$+(1-\delta_{r,R})\Bigg((1-q)(1-p_r)D_1\mathbf{v}_{virt}(s,l+1,r+1,n) + ((1-q)p_r+q)D_1\mathbf{v}_{virt}(s,l,r,n)\Bigg) +$$

$$+\delta_{r,R}D_1\mathbf{v}_{virt}(s,l,r,n) + (1-h)(N-n)\tilde{\mu}\mathbf{v}_{virt}(s,l,r,n+1)\Bigg],$$

$$l > 0,\ r = \overline{0, \min\{l-1, R\}},\ n = \overline{0, N}.$$

To find the solution to system (4) let us introduce the column vectors

$$\mathbf{v}_{virt}(s,l,r) = (\mathbf{v}_{virt}(s,l,r,0),\dots,\mathbf{v}_{virt}(s,l,r,N))^T,$$
$$\mathbf{v}_{virt}(s,l) = (\mathbf{v}_{virt}(s,l,0),\dots,\mathbf{v}_{virt}(s,l,\min\{l-1,R\}))^T,$$
$$\mathbf{v}_{virt}(s) = (\mathbf{v}_{virt}(s,1),\dots,\mathbf{v}_{virt}(s,R),\dots)^T,$$

and rewrite system (4) into the matrix form as

$$-(sI-Q_r^l)\mathbf{v}_{virt}(s,l,r)+\delta_{r,0}(1-\delta_{l,1})\tilde{Q}_0^{l-1}\mathbf{v}_{virt}(s,l-1,0)+ \quad (5)$$
$$+(1-\delta_{r,0})(1-\delta_{l,1})Q_{r-1}^{l-1}\mathbf{v}_{virt}(s,l-1,r-1)+$$
$$+(1-\delta_{r,R})Q_{r+1}^{l+1}\mathbf{v}_{virt}(s,l+1,r+1)+\delta_{r,0}\delta_{l,1}\tilde{\mathbf{a}}(s)=\mathbf{0}^T, l>0,\ r=\overline{0,\min\{l-1,R\}},$$

where

$$Q_r^l = I_{N+1}\otimes(D_0+(1-\delta_{r,R})((1-q)p_r+q)D_1+\delta_{r,R}D_1)-$$
$$-(r\alpha I_{N+1}+\mu C_N+\tilde{\mu}\tilde{C}_N-(1-h)\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}},\ l>0,$$
$$\tilde{Q}_0^{l-1} = (\mu C_N E_N+h\tilde{\mu}\tilde{C}_N)\otimes I_{\bar{W}},\ l>1,$$
$$Q_{r-1}^{l-1} = (r\alpha I_{N+1}+\mu C_N+h\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}},\ l>1,\ r=\overline{1,\min\{l-1,R\}},$$
$$Q_{r+1}^{l+1} = I_{N+1}\otimes(1-q)(1-p_r)D_1,\ l>0,\ \ r=\overline{0,\min\{l,R\}-1},$$
$$\tilde{\mathbf{a}}(s) = \frac{\tilde{\mu}}{s+\tilde{\mu}}(h+(1-h)\frac{\mu}{s+\mu})(\mu C_N+h\tilde{\mu}\tilde{C}_N)\mathbf{e}_{N+1}\otimes\mathbf{e}_{\bar{W}}.$$

Let us also introduce the vector $\mathbf{a}(s)=(\tilde{\mathbf{a}}^T(s),\mathbf{0},\dots,\mathbf{0},\dots)^T$, and the block tridiagonal matrix $\Omega=(\Omega_{l,r})_{l,r\geq 0}$ with the non-zero blocks

$$\Omega_{l,l} = -(1-\delta_{l,1})\alpha C_{l-1}\otimes I_{(N+1)\bar{W}}-I_l\otimes(\mu C_N+\tilde{\mu}\tilde{C}_N-(1-h)\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}}+$$
$$+((1-q)\bar{P}_{l-1}+qI_l)\otimes I_{N+1}\otimes D_1+I_{l(N+1)}\otimes D_0,\ l=\overline{1,R},$$
$$\Omega_{l,l}=\Omega_1=-\alpha C_R\otimes I_{(N+1)\bar{W}}-I_{R+1}\otimes(\mu C_N+\tilde{\mu}\tilde{C}_N-(1-h)\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}}+$$
$$+\bar{E}(\tilde{\Delta}+qI_{R+1})\otimes I_{N+1}\otimes D_1+(I_{R+1}-\bar{E})\otimes I_{N+1}\otimes D_1+I_{(R+1)(N+1)}\otimes D_0,\ l>R,$$
$$\Omega_{l,l-1} = \bar{I}_{l-1}\otimes(\mu C_N E_N+h\tilde{\mu}\tilde{C}_N)\otimes I_{\bar{W}}+$$
$$+(E_{l-1}^- - \bar{I}_{l-1})\otimes(\mu C_N+h\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}}+\alpha C_{l-1}E_{l-1}^-\otimes I_{(N+1)\bar{W}},\ l=\overline{2,R+1},$$
$$\Omega_{l,l-1}=\Omega_0=\hat{I}\otimes(\mu C_N E_N+h\tilde{\mu}\tilde{C}_N)\otimes I_{\bar{W}}+$$
$$+E_R\otimes(\mu C_N+h\tilde{\mu}\tilde{C}_N\hat{E}_N)\otimes I_{\bar{W}}+\alpha C_R E_R\otimes I_{(N+1)\bar{W}},\ l\geq R+2,$$
$$\Omega_{l,l+1} = (1-q)(I_l-\bar{P}_{l-1})E_{l-1}^+\otimes I_{N+1}\otimes D_1,\ l=\overline{1,R},$$
$$\Omega_{l,l+1}=\Omega_2=\bar{E}\bar{\Delta}\hat{E}_R\otimes I_{N+1}\otimes D_1,\ l>R,$$

where $\bar{E}$ is the square matrix of size $R+1$ with all zero entries except the entries $(\bar{E})_{i,i}$, $i=\overline{0,R-1}$, which are equal to 1.

Using this notation we can rewrite system (5) into the form

$$(\Omega-sI)\mathbf{v}_{virt}(s)+\mathbf{a}(s)=\mathbf{0}^T. \quad (6)$$

To solve system (6), the following algorithm can be used.

Theorem 4. *The components* $\mathbf{v}_{virt}(s,l)$, $l \geq 1$, *of the vector* $\mathbf{v}_{virt}(s)$ *can be calculated as follows:*
the vector $\mathbf{v}_{virt}(s,1)$ *is the unique solution to the system*

$$(\Omega_{1,1} - sI + \Omega_{1,2}\mathcal{F}_2(s))\mathbf{v}_{virt}(s,1) = -\tilde{\mathbf{a}}(s),$$

the rest of the components is given by

$$\mathbf{v}_{virt}(s,l) = \mathcal{F}_l(s)\mathbf{v}_{virt}(s,l-1),\ l = 2,3,\ldots,R+1,$$

$$\mathbf{v}_{virt}(s,l) = A(s)\mathbf{v}_{virt}(s,l-1),\ l > R+1,$$

where the matrix functions $\mathcal{F}_l(s)$, $l = \overline{2,R}$, *are calculated using the backward recursion*

$$\mathcal{F}_l(s) = -(\Omega_{l,l} - sI + \Omega_{l,l+1}\mathcal{F}_{l+1}(s))^{-1}\Omega_{l,l-1},\ l = R, R-1,\ldots,2,$$

under the initial condition $\mathcal{F}_{R+1}(s) = -(\Omega_1 - sI + \Omega_2 A(s))^{-1}\Omega_{R+1,R}$.

Here the matrix function $A(s)$ *is the minimal nonnegative solution of the matrix equation*

$$\Omega_2 A^2(s) + (\Omega_1 - sI)A(s) + \Omega_0 = O.$$

Corollary 1. *The average sojourn time* V_{virt}^{soj} *of an arbitrary virtual customer is calculated by*

$$V_{virt}^{soj} = -v'_{virt}(s)|_{s=0} =$$

$$= -\lambda^{-1}[(1-q)\sum_{i=N}^{\infty}\sum_{r=0}^{\min\{i-N,R-1\}} p_r \sum_{n=0}^{N}\boldsymbol{\pi}(i,r,n)D_1\mathbf{v}'_{virt}(s,i-N+1,r,n)|_{s=0}+$$

$$+(1-\delta)\sum_{i=N+R}^{\infty}\sum_{n=0}^{N}\boldsymbol{\pi}(i,R,n)D_1\mathbf{v}'_{virt}(s,i-N+1,R,n)|_{s=0}]+$$

$$-(1-\delta_{\gamma,1})(\alpha N_{real}^{buffer})^{-1}\sum_{i=N}^{\infty}\sum_{r=1}^{\min\{i-N,R\}} r\alpha\sum_{n=0}^{N}\boldsymbol{\pi}(i,r,n)\mathbf{v}'_{virt}(s,i-N+1,r-1,n)|_{s=0}.$$

Here the column vectors $\mathbf{v}'_{virt}(s,l,r,n)|_{s=0}$ *are calculated as the sub-vectors of the vectors* $\mathbf{v}'_{virt}(s,l)|_{s=0}$, $l \geq 1$, *computed by*

$$\mathbf{v}'_{virt}(0,l) = \mathcal{F}'_l(0)\mathbf{e} + \mathcal{F}_l(0)\mathbf{v}'_{virt}(0,l-1),\ l = 2,3,\ldots,R+1,$$

$$\mathbf{v}'_{virt}(0,l) = A'(0)\mathbf{e} + A(0)\mathbf{v}'_{virt}(0,l-1),\ l > R+1,$$

where the matrices $\mathcal{F}'_l(0)$, $l = \overline{2,R}$, *are calculated using the backward recursion*

$$\mathcal{F}'_l(0) = (\Omega_{l,l} + \Omega_{l,l+1}\mathcal{F}_{l+1}(0))^{-1}\times$$

$$\times(\Omega_{l,l+1}\mathcal{F}'_{l+1}(0) - I)(\Omega_{l,l} + \Omega_{l,l+1}\mathcal{F}_{l+1}(0))^{-1}\Omega_{l,l-1},\ l = R, R-1,\ldots,2,$$

under the initial condition

$$\mathcal{F}'_{R+1}(0) = (\Omega_1 + \Omega_2 A(0))^{-1}(\Omega_2 A'(0) - I)(\Omega_1 + \Omega_2 A(0))^{-1}\Omega_{R+1,R},$$

here the matrix function $A'(0)$ is the minimal nonnegative solution of the matrix equation

$$\Omega_2(A'(0)A(0) + A(0)A'(0)) + \Omega_1 A'(0) - A(0) = O,$$

the vector $\mathbf{v}'_{virt}(0,1)$ is the unique solution to the system

$$(\Omega_{1,1} + \Omega_{1,2}\mathcal{F}_2(0))\mathbf{v}'_{virt}(0,1) = -\tilde{\mathbf{a}}'(0) - \Omega_{1,2}\mathcal{F}'_2(0)\mathbf{e} + \mathbf{e}.$$

Corollary 2. *The average waiting time V_{virt}^{wait} of an arbitrary virtual customer, which got a service, is calculated by $V_{virt}^{wait} = V_{virt}^{soj} - \frac{1}{\tilde{\mu}} - \frac{1}{\mu}$.*

Remark 1. By analogy with the vector $\mathbf{v}_{virt}(s)$ of the conditional LSTs $v_{virt}(s,l,r,n,\nu)$ of the sojourn time distribution, it is possible to introduce the vector $\mathbf{w}_{virt}(s)$ of the conditional LSTs of the waiting time distribution. It can be shown that this vector can be computed from the equation

$$(\Omega - sI)\mathbf{w}_{virt}(s) + \hat{\mathbf{a}}(s) = \mathbf{0}^T$$

where the vector $\hat{\mathbf{a}}(s)$ is defined by

$$\hat{\mathbf{a}}(s) = (\bar{\mathbf{a}}^T(s), \mathbf{0}, \ldots, \mathbf{0}, \ldots)^T,\ \bar{\mathbf{a}}(s) = (\mu C_N + h\tilde{\mu}\tilde{C}_N)\mathbf{e}_{N+1} \otimes \mathbf{e}_{\overline{W}}.$$

By means of numerical inverting the components of the vector $\mathbf{w}_{virt}(s)$, see, e.g., [13], it is possible to compute the conditional waiting time distribution of the virtual customer which joins to the busy system at the moment when he (she) gets the position number l, $l \geq 1$, in the system, the number of servers providing the service is equal to n, $n = \overline{0,N}$, and the state of the arrival underlying process is equal to ν, $\nu = \overline{0,W}$. This distribution can be used for informing the arriving customer about the expected time till the moment when he (she) will be called for the service.

6 Optimization Problem

The aim of optimization is maximization of one of the possible cost criteria of the system operation:

$$J(N,R) = a\lambda^{out} - \lambda(b_1 P_{real}^{esc-loss} + b_2\delta P_{real}^{ent-loss} + b_3\gamma P_{real}^{imp-loss}) - cN - dR$$

under the condition on the average waiting time of the virtual customers

$$V_{virt}^{wait} < V. \tag{7}$$

Here a is the average profit obtained by the call center from servicing of one customer, λ^{out} is the intensity of the flow of customers, which got successful service in the system, b_1, b_2, b_3 are the charges of the call center when the arriving

customer is lost due to unwillingness to wait, absence of the place in a buffer and impatience, respectively, c is the charge paid for maintenance of one operator per unit time, d is the charge paid for maintenance of one place in the buffer per unit time, V_{virt}^{wait} is the average waiting time of the virtual customer, V is the admissible waiting time.

We fixed the following cost coefficients: $a = 15$, $b_1 = 10$, $b_2 = 20$, $b_3 = 20$, $c = 3.5$, $d = 0.1$ and assumed that the average waiting time of the virtual customer must be less than $V = 10$.

We consider the MAP defined by the matrices

$$D_0 = \begin{pmatrix} -10.81440 & 0 \\ 0 & -0.35107 \end{pmatrix}, \; D_1 = \begin{pmatrix} 10.74244 & 0.07196 \\ 0.19556 & 0.15551 \end{pmatrix}.$$

It has the fundamental arrival rate $\lambda = 8$, the coefficient of correlation $c_{cor} = 0.2$, and the coefficient of variation $c_{var} = 3.51$.

We assume that the service rate $\mu = 0.72$, the dial rate $\tilde{\mu} = 2$, the intensity of impatience $\alpha = 0.2$, the probabilities $\gamma = 0.4$, $\delta = 0.5$, $q = 0.1$, $h = 0.15$, the probabilities $p_r = 0.2 + 0.02r$, $r = \overline{0, R-1}$.

Figure 2 illustrates the dependence of the cost criterion $J(N, R)$ on the number of servers N and the buffer capacity R.

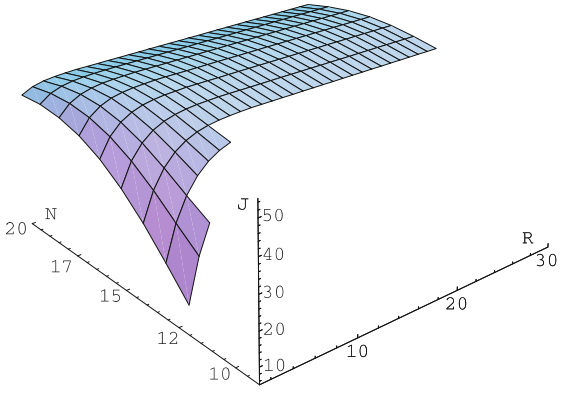


Fig. 2. The dependence of the cost criterion $J(N, R)$ on the number of servers N and the buffer capacity R

Let us note that if $N < 9$ for $R = 1, 2$, $N < 10$ for $R = \overline{3, 5}$ and $N < 11$ for $R = \overline{6, 30}$ ergodicity condition (1) is not fulfilled. For other values of N and R, for which the value of criterion is not represented on the figure 2, condition (1) is fulfilled, but inequality (7) is violated. The optimal value of the cost criterion $J(N, R)$ is equal to 51.27 and the optimal number of servers $N = 16$ and the buffer capacity $R = 9$.

7 Conclusion

The multi-server queueing system with the visible queue, call-back option, and impatient customers is investigated. The main performance measures are calculated. The Laplace-Stieltjes transform of the sojourn time distribution for call-back customers is derived. The obtained results can be used for performance evaluation and optimization of the call centers of banks, emergency and information services, mobile operators.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (Grant No. 2011-0015214).

References

1. Aksin, O.Z., Armony, M., Mehrotra, V.: The Modern Call Centers: a Multi-disciplinary Perspective on Operations Management Research. Working paper, Koc University, Istanbul, Turkey (2007)
2. Kim, J.W., Park, S.C.: Outsourcing strategy in two-stage call centers. Computers & Operations Research 37, 790–805 (2010)
3. Dudin, S., Dudina, O.: Call center operation model as a $MAP/PH/N/R-N$ system with impatient customers. Problems of Information Transmission 47, 364–377 (2011)
4. Armony, M., Maglaras, C.: On Customer Contact Centers with a Call-back Option: Customer Decisions, Routing Rules, and System Design. Operations Research 52(2), 271–292 (2004)
5. Armony, M., Maglaras, C.: Contact Centers with a Call-back Option and Real-Time Delay Information. Operations Research 52(4), 527–545 (2004)
6. Iravani, F., Balcioglu, B.: On Priority Queues with Impatient Customers. Queueing Systems 58, 239–260 (2008)
7. Lucantoni, D.: New Results on the Single Server Queue with a Batch Markovian Arrival Process. Communication in Statistics-Stochastic Models 7, 1–46 (1991)
8. Graham, A.: Kronecker Products and Matrix Calculus with Applications. Ellis Horwood, Chichester (1981)
9. Neuts, M.: Matrix-geometric Solutions in Stochastic Models – An Algorithmic Approach. Johns Hopkins University Press, USA (1981)
10. Klimenok, V.I., Dudin, A.N.: Multi-dimensional Asymptotically Quasi-Toeplitz Markov Chains and their Application in Queueing Theory. Queueing Systems 54, 245–259 (2006)
11. Kesten, H., Runnenburg, J.T.: Priority in Waiting Line Problems. Mathematisch Centrum, Amsterdam (1956)
12. van Danzig, D.: Chaines de Markof dans les ensembles abstraits et applications aux processus avec regions absorbantes et au probleme des boucles. Ann. de l'Inst. H. Poincare 14, 145–199 (1955)
13. Abate, J., Whitt, W.: Numerical Inversion of the Laplace Transforms of Probability Distributons. ORSA Journal on Computing 7(1), 36–43 (1955)

Two Way Communication Retrial Queues with Balanced Call Blending

Tuan Phung-Duc[1] and Wouter Rogiest[2,⋆]

[1] Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
tuan@sys.i.kyoto-u.ac.jp
[2] Department of Telecommunications and Information Processing,
Ghent University
St.-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
wouter.rogiest@telin.ugent.be

Abstract. In call centers, call blending consists in the mixing of incoming and outgoing call activity. Artalejo and Phung-Duc recently provided an apt model for such a setting, with a two way communication retrial queue. However, by assuming a classical (proportional) retrial rate for the incoming calls, the outgoing call activity is largely blocked when many incoming calls are in orbit, which may be unwanted, especially when outgoing calls are vital to the service offered.

In this paper, we assume a balanced way of call blending, through a retrial queue with constant retrial rate for incoming calls. For the single server case (one operator), a generating functions approach enables deriving explicit formulas for the joint stationary distribution of the number of incoming calls and the system state, and also for the factorial moments. This is complemented with a stability analysis, expressions for performance measures, and also recursive formulas, allowing reliable numerical calculation. For the multiserver case (multiple operators), we provide a quasi-birth-and-death process formulation, enabling deriving a sufficient and necessary condition for stability in this case, as well as a numerical recipe to obtain the stationary distribution.

Keywords: Markov chain, retrial queues, single server, multiserver, call centers, call blending.

1 Introduction

Retrial queues have received considerable attention over recent years, providing an apt model for the performance evaluation of call centers, computer networks, and communications systems. An overview is given in [1,2]. Characteristic of retrial queues is the fact that calls (or, in general, customers) that cannot be served upon arrival enter an orbit and request for a retrial after some random time. Due to this, analysis of a retrial queue is more difficult than that of its

⋆ Corresponding author.

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 16–31, 2012.

counterpart model without retrial, and explicit results can only be obtained in a few special cases [3].

Here, we consider a specific retrial queue model with application to call centers. As explained in [4], a central characteristic of a call center is whether it handles inbound traffic, with incoming calls, or outbound traffic, with outgoing calls. Correspondingly, they are referred to as inbound and outbound call centers. Most retrial queue models in literature assume such a system, with one way communication. However, often, call centers are not strictly inbound or outbound, and typically handle a mixture of incoming and outgoing calls. Typically, incoming calls are assigned to operators by an Automatic Call Distributor (ACD). For outgoing calls, calls are either initiated by the ACD (automatically), or by the operators (manually).

The principle of mixing is referred to as call blending, with two way communication, serving several purposes.

- Firstly, it may be added to the regular tasks, as in the case of an inbound call center in which operators utilize their idle time to perform secondary, non-urgent outgoing calls. Then, call blending is primarily a way to increase overall productivity, by increasing operator utilization, potentially through a control policy. A mathematical analysis and optimization of such a policy is presented in [5].
- Secondly, it may occur as an integral part of the tasks performed at the call center. In this case, incoming as well as outgoing calls are vital elements of the service delivered, and should both be performed. This occurs when tasks necessitate several calls in both directions.

Both cases can be modeled with a retrial queue supporting two way communication. More precisely, [6] assumes a model with classical retrial rate for incoming calls (see Section 2.1 for definition of retrial rate). Such a choice results in an apt model for the first case, since the outgoing call activity is indeed largest when few incoming calls are in orbit, and smallest when many are in orbit. However, in the second case, such behavior is undesirable, since the outgoing call activity should also continue regularly while many incoming calls are awaiting service. By assuming a constant retrial rate, outgoing calls are still initiated regularly (either by the ACD or by the operators), even if the number of incoming calls in orbit is high.

Further, note that many types of call blending can be identified; Koole and Mandelbaum [4] provide an excellent overview. A high-level discussion and basic performance analysis is provided in [7]. The paper [8] presents a collection of Markov chain models for call centers, including a discussion of model fidelity and efficacy, in a simulation context. Although different in several ways, Model M1 in [8], with "all blend agents and no mismatches" shares many of the assumptions of the two way communication retrial queue model presented in [6] and here (see also Section 2.1): inbound calls arriving according to a Poisson process, with independent and identically distributed (i.i.d.) service times drawn from an exponential distribution for inbound as well as outbound calls. Further, in [8] and here, multiple identical blend agents (or operators) are assumed (only

1 in [6]), as well as a First-Come-First-Served (FCFS) order for the queue of incoming calls. More precisely, although the model presented here does not make assumptions on the service order for the queue, constant retrial rate is commonly associated with FCFS ordering in the orbit queue (see e.g. [9,10], and [11,12] for discussion), with only the customer at the head of the queue able to request for service. In this regard, FCFS ordering of incoming call requests may be viewed as a more natural (but not only) way to realize a constant retrial rate.

As mentioned earlier, [6] shares many of the assumptions of the single server part of this work. However, assuming constant instead of classical retrial rate leads to completely different expressions for the variables analyzed, with no simple (mathematical) way to relate the obtained results to those of [6]. In terms of analysis, more closely related to this contribution is [13]. In this paper, a service system is analyzed in which a processor must serve two types of impatient units, with either infinitely impatient or infinitely patient customers. Assuming general service times, a variant of Takács' equation is derived which also holds for the system considered in this work, with exponentially distributed service times. In this regard, this paper [13] provides an interesting reference, but does not contain any of the derivations and expressions reported here.

The contribution of this paper is twofold. First, we carry out an extensive analysis for the single server retrial queue with two way communication and constant retrial rate in which we derive explicit expressions for the joint stationary distribution and their partial generating functions as well as recursive formulae. Second, we formulate the multiserver case by a quasi-birth-and-death process for which the stability condition and a numerical recipe are presented.

The remainder of this paper is structured as follows. In Section 2, we set out the model and assumptions of the current work, as well as the balance equations governing the system's behavior. Section 3 presents an exhaustive analysis of the single server case (one operator), including a study of stability, as well as closed-form expressions for the joint stationary distribution of the number of incoming calls and the system state, and several other measures of interest. In Section 4, we consider the multiserver model (multiple operators), through a formulation using a quasi-birth-and-death process. As explained, this allows to apply standard numerical recipes to obtain the stationary distribution as well as the stability condition. Finally, conclusions are drawn in Section 5.

2 Model

In this section, we first list the assumptions made in this work, introducing notation for the parameters involved. This allows to formulate a set of balance equations, which will provide the starting point for the analysis in the next section.

2.1 Assumptions

A single server retrial queue with two way communication is considered. Primary incoming call requests arrive at the server (or operator) according to a Poisson

process with rate λ. Incoming calls finding an idle server receive service instantly. In case of a busy server, the incoming call enters an orbit. Within the orbit, a constant retrial policy is applied, i.e., the arrival rate of customers from the orbit is $\mu(1-\delta_{0,n})$ provided that there are n customers in the orbit, where $\delta_{0,n}$ denotes the Kronecker delta. This is opposed to the case analyzed in [6], with a classical retrial rate $n\mu$, which depends on the number of customers in orbit, n. As mentioned, constant retrial rate occurs when customers form a FCFS queue in the orbit, and only the customer at the head of the queue can request service. In addition, when the server turns idle, it makes an outgoing call after an exponentially distributed time with rate α. The service times of the incoming and outgoing calls are i.i.d., exponentially distributed with rate ν_1 and ν_2 respectively.

2.2 Markov Chain and Balance Equations

Let $S(t)$ denote the state of the server at time t,

$$S(t) = \begin{cases} 0 & \text{if the server is idle,} \\ 1 & \text{if the server is providing an incoming service,} \\ 2 & \text{if the server is providing an outgoing service,} \end{cases}$$

and let $N(t)$ denote the number of calls in orbit at time t. Here, the couple $\{(S(t), N(t)); t \geq 0\}$ forms a Markov chain on the state space $\{0,1,2\} \times \mathbb{Z}_+$, with $\mathbb{Z}_+ = \{0,1,2,\ldots\}$. Given that the Markov chain is aperiodic and irreducible, under the condition that the system is stable, the probability distributions associated with the variables involved converge to a unique stochastic equilibrium for $t \to \infty$, to

$$\pi_{i,j} = \lim_{t\to\infty} \Pr[S(t) = i\,,\, N(t) = j], \qquad (i,j) \in \{0,1,2\} \times \mathbb{Z}_+\,.$$

The condition for stability will be derived in Section 3. Now, the probabilities $\pi_{i,j}$, $(i,j) \in \{0,1,2\}\times\mathbb{Z}_+$, are characterized by following set of balance equations,

$$(\lambda + \alpha + \mu(1-\delta_{0,j}))\pi_{0,j} = \upsilon_1\pi_{1,j} + \upsilon_2\pi_{2,j}, \tag{1}$$

$$(\lambda + \upsilon_1)\pi_{1,j} = \lambda\pi_{0,j} + \mu\pi_{0,j+1} + \lambda\pi_{1,j-1}, \tag{2}$$

$$(\lambda + \nu_2)\pi_{2,j} = \alpha\pi_{0,j} + \lambda\pi_{2,j-1}, \tag{3}$$

for $j \in \mathbb{Z}_+$, with $\delta_{0,j} = 1$ for $j = 0$ and zero elsewhere, and $\pi_{i,-1} = 0$ for $i \in \{1,2\}$.

Let $\Pi_i(z)$, $i \in \{0,1,2\}$, denote the partial generating functions

$$\Pi_i(z) = \sum_{j=0}^{\infty} \pi_{i,j} z^j, \qquad i \in \{0,1,2\}\,,$$

with z a complex number, $z \in \mathbb{C}$. Multiplying the balance equations (1-3) by z^j and taking the sum over $j \in \mathbb{Z}_+$, the balance equations are transformed to the z-domain, yielding

$$(\lambda+\alpha+\mu)\Pi_0(z)-\mu\pi_{0,0}=v_1\Pi_1(z)+v_2\Pi_2(z), \tag{4}$$

$$(\lambda+v_1)\Pi_1(z)=(\lambda+\mu z^{-1})\Pi_0(z)-\mu\pi_{0,0}z^{-1}+\lambda z\Pi_1(z), \tag{5}$$

$$(\lambda+\nu_2)\Pi_2(z)=\alpha\Pi_0(z)+\lambda z\Pi_2(z). \tag{6}$$

Summing (4-6), multiplying with z and dividing by $(z-1)$ leads to an orbit balance equation,

$$\mu(\Pi_0(z)-\pi_{0,0})=\lambda z(\Pi_1(z)+\Pi_2(z))\,, \tag{7}$$

which will prove useful at several points in the analysis below.

3 Analysis

This section provides the analysis of the single-server case. We first derive explicit expressions for the three partial probability generating functions $\Pi_i(z)$, $i\in\{0,1,2\}$, associated with the stationary distribution probabilities $\pi_{i,j}$, $(i,j)\in\{0,1,2\}\times\mathbb{Z}_+$. From these, a stability condition is derived, and inversion of the generating functions to the probability domain yields closed-form expressions for the stationary distribution. Further, we treat the factorial moments, recursive formulas, first moments and a cost model.

3.1 Generating Functions

Looking for explicit expressions for the $\Pi_i(z)$, $i\in\{0,1,2\}$, we first remark that $\Pi_1(z)$ can be expressed in terms of $\Pi_0(z)$ through (4), leading to

$$\Pi_1(z)=\frac{1}{\nu_1}(\lambda+\alpha+\mu-\nu_2\frac{\alpha}{\lambda+\nu_2-\lambda z})\Pi_0(z)-\frac{\mu}{\nu_1}\pi_{0,0}\,, \tag{8}$$

while (6) yields that

$$\Pi_2(z)=\frac{\alpha}{\lambda+\nu_2-\lambda z}\Pi_0(z)\,. \tag{9}$$

Substituting (9) and (8) in (7), we obtain

$$\Pi_0(z)=\pi_{0,0}(1-\frac{\lambda z}{\nu_1})\left(1-\frac{\lambda z}{\mu\nu_1}\left(\lambda+\mu+\alpha\frac{\lambda+\nu_1-\lambda z}{\lambda+\nu_2-\lambda z}\right)\right)^{-1}. \tag{10}$$

Now, only $\pi_{0,0}$ needs to be determined to make (10) explicit. To obtain $\pi_{0,0}$, we evaluate the partial generating functions in $z=1$, and then verify the normalization condition. We obtain

$$\begin{aligned}\Pi_0(1)&=\pi_{0,0}\left(1-\frac{\lambda}{\nu_1}\right)\left(1-\frac{\lambda}{\mu\nu_1}\left(\lambda+\mu+\alpha\frac{\nu_1}{\nu_2}\right)\right)^{-1},\\ \Pi_1(1)&=\frac{\lambda+\mu}{\nu_1}\Pi_0(1)-\frac{\mu}{\nu_1}\pi_{0,0},\\ \Pi_2(1)&=\frac{\alpha}{\nu_2}\Pi_0(1).\end{aligned} \tag{11}$$

Introducing some additional notation,

$$\rho = \frac{\lambda}{\nu_1}, \qquad \sigma = \frac{\alpha}{\nu_2},$$

and requiring that $\sum_{i=0}^{2} \Pi_i(1) = 1$, some calculations yield

$$\pi_{0,0} = \frac{1 - \frac{\lambda}{\mu}(\rho + \sigma + \frac{\mu}{\nu_1})}{1 + \sigma}. \tag{12}$$

Expression (10) and (12) together provide an expression for the partial generating function $\Pi_0(z)$, which is a function of only the model parameters, and thus explicit, as wanted. Explicit expressions for $\Pi_1(z)$ and $\Pi_2(z)$ are readily obtained, by substituting $\Pi_0(z)$ in (8) and (9), respectively.

Finally, with (12), we can simplify (11), to obtain

$$\Pi_0(1) = \frac{1-\rho}{1+\sigma}\,,\ \Pi_1(1) = \rho\,,\ \Pi_2(1) = \sigma\frac{1-\rho}{1+\sigma}. \tag{13}$$

These steady-state probabilities have also been obtained in [6] for the model with classical retrial rate. This is somewhat surprising, since assumptions are different, and all three generating functions ($\Pi_0(z)$, $\Pi_1(z)$ and $\Pi_2(z)$) reported here differ significantly from those in [6]. The fact that the values of (13) match can be understood from the fact that no incoming calls are ever lost (and thus, $\Pi_1(1)$ should amount to the traffic load).

3.2 Stability Condition

With $\pi_{0,0}$ obtained by (12), a characterization of stability is now straightforward. More precisely, requesting $\pi_{0,0}$ to be larger than zero leads to the stability condition for the single-server system,

$$-\mu + (\lambda + \mu)\frac{\lambda}{\nu_1} + \alpha\frac{\lambda}{\nu_2} < 0\,. \tag{14}$$

3.3 Stationary Distribution

At this point, we derive explicit formulae for $\pi_{0,j}$, $\pi_{1,j}$ and $\pi_{2,j}$, $j \in \mathbb{Z}_+$. We already have $\pi_{0,0}$ from (12), and start by deriving $\pi_{0,j}$, $j \geq 1$, from (10). To this end, we transform $\Pi_0(z)$ from (10) as follows

$$\Pi_0(z) = \frac{\pi_{0,0}(1-\rho z)(1-\theta z)}{\frac{1}{b}z^2 - \frac{a}{b}z + 1}, \tag{15}$$

where

$$\theta = \frac{\lambda}{\lambda + \nu_2}\,,\ a = \frac{(\lambda+\mu)(\lambda+\nu_2) + \alpha(\lambda+\nu_1) + \mu\nu_1}{\lambda(\lambda+\alpha+\mu)}\,,\ b = \frac{\mu\nu_1(\lambda+\nu_2)}{\lambda^2(\lambda+\alpha+\mu)}.$$

Through a partial fraction expansion, this can be rewritten as

$$\Pi_0(z) = \pi_{0,0}\left(1 + \frac{z}{z_1}\frac{C}{1-\frac{z}{z_1}} + \frac{z}{z_2}\frac{D}{1-\frac{z}{z_2}}\right), \tag{16}$$

with

$$C = \frac{(1-\rho z)(1-\theta z)}{1-\frac{z}{z_2}}\Big|_{z=z_1} = \frac{z_2(1-\rho z_1)(1-\theta z_1)}{z_2 - z_1},$$
$$D = \frac{(1-\rho z)(1-\theta z)}{1-\frac{z}{z_1}}\Big|_{z=z_2} = \frac{z_1(1-\rho z_2)(1-\theta z_2)}{z_1 - z_2},$$

where z_1 and z_2 denote the real and positive poles of $\Pi_0(z)$, namely

$$z_1 = \frac{a+\sqrt{a^2-4b}}{2}, \qquad z_2 = \frac{a-\sqrt{a^2-4b}}{2},$$

with $z_1 + z_2 = a$, and $z_1 z_2 = b$. If stability condition (14) holds, $z_1 > z_2 > 1$, so enabling inversion of (16) to the probability domain, as

$$\pi_{0,j} = \pi_{0,0}\left[C\left(\frac{1}{z_1}\right)^j + D\left(\frac{1}{z_2}\right)^j\right], \qquad j \geq 1\,. \tag{17}$$

To obtain $\pi_{1,j}$ and $\pi_{2,j}$, we expand the partial generating functions into simpler fractions, which can easily be inverted from the z-domain to the probability domain. A useful expression in the calculation is obtained from (15), by performing a partial fraction expansion, leading to

$$\frac{1}{1-\theta z}\Pi_0(z) = \pi_{0,0}(1-\rho z)\left(\frac{1}{(1-\frac{z_1}{z_2})(1-\frac{z}{z_1})} + \frac{1}{(1-\frac{z_2}{z_1})(1-\frac{z}{z_2})}\right).$$

Using this, and (8) and (17), we obtain

$$\pi_{1,0} = \frac{1}{\nu_1}\pi_{0,0}(\lambda + \alpha - \frac{\alpha\nu_2}{\lambda+\nu_2})\,, \tag{18}$$

$$\pi_{1,j} = \frac{1}{\nu_1}\pi_{0,0}\left[C\left(\lambda+\alpha+\mu-\frac{\alpha\nu_2}{(\lambda+\nu_2)(1-\theta z_1)}\right)\left(\frac{1}{z_1}\right)^j + D\left(\lambda+\alpha+\mu-\frac{\alpha\nu_2}{(\lambda+\nu_2)(1-\theta z_2)}\right)\left(\frac{1}{z_2}\right)^j\right], \quad j \geq 1\,. \tag{19}$$

Similarly, with (9) and (17), we find

$$\pi_{2,0} = \frac{\alpha}{\lambda+\nu_2}\pi_{0,0}\,, \tag{20}$$

$$\pi_{2,j} = \frac{\alpha}{\lambda+\nu_2}\pi_{0,0}\left[\frac{C}{1-\theta z_1}\left(\frac{1}{z_1}\right)^j + \frac{D}{1-\theta z_2}\left(\frac{1}{z_2}\right)^j\right], \quad j \geq 1\,. \tag{21}$$

As such, for $j \in \mathbb{Z}_+$, $\pi_{0,j}$ is given by (12) and (17), $\pi_{1,j}$ by (18) and (19), and $\pi_{2,j}$ by (20) and (21).

Remark 1. In the derivations presented above, we implicitly assumed that $z_1 \notin \{z_2, \rho^{-1}, \theta^{-1}\}$, and $z_2 \notin \{\rho^{-1}, \theta^{-1}\}$. This assumption excludes some minor special cases where the inversion from z-domain to the probability domain requires small modifications.

3.4 Factorial Moments

Next, we derive explicit expressions for the partial factorial moments M_k^i, $(i,k) \in \{0,1,2\} \times \mathbb{Z}_+$, which relate to the coefficients of z^k in the series $\Pi_i(1+z)$ as follows,

$$\Pi_i(1+z) = \sum_{k=0}^{\infty} \frac{M_k^i}{k!} z^k, \qquad i \in \{0,1,2\}\,. \tag{22}$$

For $k = 0$, (13) already provides the answer, since $M_0^i = \Pi_i(1)$. For $k \geq 1$, expressing $\Pi_0(1+z)$ using (16), we obtain

$$\begin{aligned}
M_k^0 &= k!\pi_{0,0} \left(\frac{Cz_1}{(z_1-1)^{k+1}} + \frac{Dz_2}{(z_2-1)^{k+1}} \right), \quad k \geq 1\,, \\
M_k^1 &= \frac{k!\pi_{0,0}}{\nu_1} \left[\left(\lambda + \alpha + \mu - \frac{\alpha\nu_2}{(\lambda+\nu_2)(1-\theta z_1)} \right) \frac{Cz_1}{(z_1-1)^{k+1}} \right. \\
&\qquad \left. + \left(\lambda + \alpha + \mu - \frac{\alpha\nu_2}{(\lambda+\nu_2)(1-\theta z_2)} \right) \frac{Cz_2}{(z_2-1)^{k+1}} \right], \quad k \geq 1\,, \\
M_k^2 &= \frac{\alpha k!\pi_{0,0}}{\lambda+\nu_2} \left[\frac{1}{1-\theta z_1} \frac{Cz_1}{(z_1-1)^{k+1}} + \frac{1}{1-\theta z_2} \frac{Cz_2}{(z_2-1)^{k+1}} \right], \quad k \geq 1\,.
\end{aligned}$$

Together with (13) (with $\Pi_i(1) = M_0^i$), this provides explicit expressions for all M_k^i, $k \in \mathbb{Z}_+$, $i \in \{0,1,2\}$.

3.5 Recursive Formulae

In Sections 3.3 and 3.4, explicit expressions are given for the stationary distribution and the partial factorial moments. However, since the coefficients involved may be either positive or negative, numerical computation may be unreliable. Opposed to this, a recursive computation with only positive terms provides a numerically stable alternative.

The stationary probabilities can be expressed recursively as follows,

$$\pi_{0,j} = \frac{\lambda(\pi_{1,j-1} + \pi_{2,j-1})}{\mu}, \qquad j \geq 1\,, \tag{23}$$

$$\pi_{2,j} = \frac{\alpha\pi_{0,j} + \lambda\pi_{2,j-1}}{\lambda+\nu_2}, \qquad j \geq 1\,, \tag{24}$$

$$\pi_{1,j} = \frac{\lambda(\pi_{0,j} + \pi_{2,j} + \pi_{1,j-1})}{\nu_1}, \qquad j \geq 1\,, \tag{25}$$

where we recall that $\pi_{0,0}$ is given by (12), $\pi_{1,0}$ by (18), and $\pi_{2,0}$ by (20). Expression (23) can be obtained from (7), whereas (24) is derived from (3), and (25) from the combination of (2) and (24).

For the partial factorial moments, starting point is the expression (7), substituting z with $(1+z)$. Appealing to (22), we find that

$$M_k^0 = \frac{\lambda(M_k^1 + M_k^2) + k\lambda(M_{k-1}^1 + M_{k-1}^2)}{\mu}, \qquad k \geq 1\,. \tag{26}$$

Similarly, from (6), substituting z with $(1+z)$, we obtain

$$M_k^2 = \frac{\alpha M_k^0 + k\lambda M_{k-1}^2}{\nu_2}, \qquad k \geq 1\,. \tag{27}$$

Further, adding (7) to the product of (5) and z allows to derive that

$$M_k^1 = \frac{\lambda(M_k^0 + M_k^2 + kM_{k-1}^1)}{\nu_1 - \lambda}, \qquad k \geq 1\,. \tag{28}$$

Substituting (27) and (28) in (26) yields

$$M_k^0 = k\lambda \cdot \frac{\nu_1\nu_2 M_{k-1}^1 + [\lambda\nu_1 + \nu_2(\nu_1 - \lambda)]M_{k-1}^2}{\mu\nu_2(\nu_1 - \lambda) - \lambda(\alpha\nu_1 + \lambda\nu_2)}, \qquad k \geq 1\,. \tag{29}$$

Expressions (27), (28) and (29), and (13) (with $\Pi_i(1) = M_0^i$) together provide the recursive formulation for the partial factorial moments. It should be noted that the denominator of (29) is positive due to the stability condition (14).

3.6 First Moments and Cost Model

In this section, deriving first moments allows formulating a cost model. From (10), we find that

$$M_1^0 = \Pi_0'(1) = \pi_{0,0}\left(\frac{Cz_1}{(1-z_1)^2} + \frac{Dz_2}{(1-z_2)^2}\right),$$

which, after some calculations using $z_1 + z_2 = a$ and $z_1 z_2 = b$, is simplified as

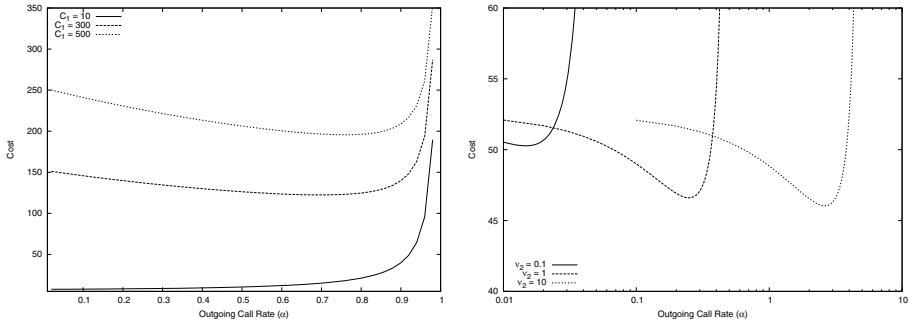
$$M_1^0 = \pi_{0,0} b \frac{a - 2 + (\rho+\theta)(1-b) + \rho\theta(2b-a)}{(1-a+b)^2}. \tag{30}$$

Let $\mathrm{E}[N]$ denote the average number of customers in the orbit, i.e.,

$$\mathrm{E}[N] = M_1^0 + M_1^1 + M_1^2.$$

It follows from (26) that

$$M_1^1 + M_1^2 = \frac{\mu}{\lambda} M_1^0 - (M_0^1 + M_0^2),$$



(a) with various C_1 (b) with various ν_2

Fig. 1. Cost as a function of varying α, for the parameter settings specified in Table 1

leading to

$$\mathrm{E}[N] = \frac{\lambda+\mu}{\lambda} M_1^0 - \frac{\rho+\sigma}{1+\sigma},$$

where M_1^0 is given by (30).

Let U denote the utilization of the server, i.e., $U = M_0^1 + M_0^2$. From a management point of view, we need to minimize $1-U$. At the same time, we also need to minimize the number of customers in the orbit $\mathrm{E}[N]$. These considerations motivate the following cost model,

$$\begin{aligned} \min \quad & f(\alpha) = C_1(1-U) + C_2\mathrm{E}[N], \\ s.t. \quad & -\mu + (\lambda+\mu)\frac{\lambda}{\nu_1} + \alpha\frac{\lambda}{\nu_2} < 0, \qquad \alpha \geq 0\,, \end{aligned}$$

where the inequality comes from the stability condition and C_1 and C_2 are the cost of idle server and of a retrial customer. The cost model formulation boils down to finding the optimal α while keeping all other parameters constant.

Remark 2. In our model, there are a number of free parameters such as $\lambda, \mu, \nu_1, \nu_2$ and α. Thus, the optimization formulation presented above is only one of several options. However, aiming for the optimization of α is natural in the call center context, as it can be controlled by the operator (directly), or by the ACD (automatically).

In Fig. 1, the cost model is evaluated for varying α, under the parameter setting specified in Table 1. For the setting considered in Fig. 1a, cost evaluation yields a non-trivial optimal value for α when $C_1 \in \{300, 500\}$. This corresponds to the case where the cost of idle server (C_1) is (much) larger than the cost of a retrial customer (C_2, fixed to 1). Opposed to this, when the cost of idle server is small, $C_1 = 10$, the cost function is monotonically increasing, and the optimum is trivially found for $\alpha = 0$, with no outgoing call activity, which is intuitive. For various ν_2, illustrated in Fig. 1b, cost evaluation yields clear optima for α,

Table 1. Parameter setting for the numerical examples considered in Fig. 1

Figure	λ	ν_1	ν_2	μ	C_1	C_2
(a)	0.5	1	2	1	various	1
(b)	0.5	1	various	1	100	1

increasing with increasing ν_2, rate parameter of the outgoing call service time distribution. In the optimum, apparently, reduced duration of outgoing calls is matched by an increased rate for outgoing call activity, which is also quite intuitive.

4 Multiple Operators

While the analysis of the previous section assumed a single operator (single server), we now shift focus to the case of multiple operators (multiserver). In particular, we consider an M/M/c/c ($c \geq 1$) retrial queue with constant retrial rate and two way communication, where the notations λ, μ, ν_1, ν_2 and α have the same definitions as above. The behavior of each server in this multiserver model is the same as that of the single server case, i.e., an idle server makes an outgoing call after an exponentially distributed time with rate α. We first provide a Quasi-Birth-and-Death (QBD) process formulation, identifying all the components of the involved infinitesimal generator and block matrices. Next, we highlight stability, and also examine the numerical recipe for the calculation of the stationary probabilities.

4.1 Infinitesimal Generator and Matrices

Let $S_1(t), S_2(t)$ and $N(t)$ denote the numbers of incoming calls and outgoing calls in the servers and the number of customers in the orbit at time t, respectively. It is easy to see that $\{X(t) = (S_1(t), S_2(t), N(t)); t \geq 0\}$ forms a level-independent QBD process in the state space

$$\mathcal{S} = \{(i, j, k); i = 0, 1, \ldots, c, j = 0, 1, \ldots, c - i, j \in \mathbb{Z}_+\}.$$

Let O denote a matrix with an appropriate size with all zero entries. It is easy to see that the infinitesimal generator of $\{X(t); t \geq 0\}$ is given by

$$Q = \begin{pmatrix} A^0 & A^+ & O & O & \cdots \\ A^- & A & A^+ & O & \cdots \\ O & A^- & A & A^+ & \cdots \\ O & O & A^- & A & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

The block matrices A^-, A, A^+ and A^+ are explicitly written as follows,

$$A^- = \begin{pmatrix} O & A_0^- & O & \cdots & O \\ O & O & A_1^- & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & O \\ \vdots & & & O & A_{c-1}^- \\ O & \cdots & \cdots & O & O \end{pmatrix}, \qquad A = \begin{pmatrix} A_{0,1} & A_{0,0} & O & \cdots & \cdots & O \\ A_{1,2} & A_{1,1} & A_{1,0} & \ddots & & \vdots \\ O & A_{2,2} & A_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & O \\ \vdots & & \ddots & \ddots & A_{c-1,1} & A_{c-1,0} \\ O & \cdots & \cdots & O & A_{c,2} & A_{c,1} \end{pmatrix},$$

$$A^+ = \begin{pmatrix} A_0^+ & O & O & \cdots & O \\ O & A_1^+ & O & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & O \\ \vdots & & & A_{c-1}^+ & O \\ O & \cdots & \cdots & O & A_c^+ \end{pmatrix}, \qquad A^0 = \begin{pmatrix} A_{0,1}^0 & A_{0,0} & O & \cdots & \cdots & O \\ A_{1,2} & A_{1,1}^0 & A_{1,0} & \ddots & & \vdots \\ O & A_{2,2} & A_{2,1}^0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & O \\ \vdots & & \ddots & \ddots & A_{c-1,1}^0 & A_{c-1,0} \\ O & \cdots & \cdots & O & A_{c,2} & A_{c,1}^0 \end{pmatrix},$$

where A_i^- $(i = 0, 1, \ldots, c-1)$, A_i^+ $(i = 0, 1, \ldots, c)$, $A_{i,2}$ $(i = 1, 2, \ldots, c)$, $A_{i,1}$, $A_{i,1}^0$ $(i = 0, 1, \ldots, c)$ and $A_{i,0}$ $(i = 0, 1, \ldots, c-1)$ are $(c-i+1) \times (c-i)$, $(c-i+1) \times (c-i+1)$, $(c-i+1) \times (c-i+2)$, $(c-i+1) \times (c-i+1)$, $(c-i+1) \times (c-i+1)$ and $(c-i+1) \times (c-i)$ matrices respectively, with entries given by

$$A_i^-(j, j') = \begin{cases} \mu, & j' = j \ (j = 0, 1, \ldots, c-i-1), \\ 0, & \text{otherwise}, \end{cases}$$

$$A_i^+(j, j') = \begin{cases} \lambda, & j' = j = c-i, \\ 0, & \text{otherwise}, \end{cases}$$

$$A_{i,2}(j, j') = \begin{cases} i\nu_1, & j' = j \ (j = 0, 1, \ldots, c-i), \\ 0, & \text{otherwise}, \end{cases}$$

$$A_{i,1}(j, j') = \begin{cases} (c-i-j)\alpha, & j' = j+1 \ (j = 0, 1, \ldots, c-i-1), \\ j\nu_2, & j' = j-1 \ (j = 1, 2, \ldots, c-i), \\ -\gamma_{i,j}, & j' = j \ (j = 0, 1, \ldots, c-i), \\ 0, & \text{otherwise}, \end{cases}$$

$$A_{i,1}^0(j, j') = \begin{cases} (c-i-j)\alpha, & j' = j+1 \ (j = 0, 1, \ldots, c-i-1), \\ j\nu_2, & j' = j-1 \ (j = 1, 2, \ldots, c-i), \\ -\gamma_{i,j}^0, & j' = j \ (j = 0, 1, \ldots, c-i), \\ 0, & \text{otherwise}, \end{cases}$$

$$A_{i,0}(j, j') = \begin{cases} \lambda, & j' = j \ (j = 0, 1, \ldots, c-i-1), \\ 0, & \text{otherwise}, \end{cases}$$

where $\gamma_{i,j} = \lambda + \mu + i\nu_1 + j\nu_2 + (c-i-j)\alpha$, and $\gamma_{i,j}^0 = \lambda + i\nu_1 + j\nu_2 + (c-i-j)\alpha$.

4.2 Stability Condition

We consider the matrix $P = A^- + A + A^+$, which is the infinitesimal generator of the irreducible Markov chain $\{C(t) = (S_1(t), S_2(t)); t \geq 0\}$ on the state space $\mathcal{V} = \{(i,j); i = 0, 1, \ldots, c, j = 0, 1, \ldots, c-i\}$. It should be noted that this Markov chain represents the behavior of the servers regardless of the number of customers in the orbit when it is large enough. Let $p_{i,j} = \lim_{t\to\infty} \Pr(S_1(t) = i, S_2(t) = j)$ for $(i,j) \in \mathcal{V}$. Furthermore, let $\boldsymbol{p}_i = (p_{i,0}, p_{i,1}, \ldots, p_{i,c-i})$ $(i = 0, 1, \ldots, c)$. In addition, let $\boldsymbol{p} = (\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_c)$ denote the stationary distribution of $\{C(t); t \geq 0\}$, which is the unique solution of the following system of equations.

$$\boldsymbol{p}P = \boldsymbol{0}, \qquad \boldsymbol{p}\boldsymbol{e} = 1,$$

where $\boldsymbol{0}$ and $\boldsymbol{e}$ denote a row and a column vector with an appropriate size with all zero and all one entries, respectively. The necessary and sufficient condition for the stability of $\{X(t); t \geq 0\}$ is given by

$$\boldsymbol{p}A^+\boldsymbol{e} < \boldsymbol{p}A^-\boldsymbol{e}, \tag{31}$$

according to [14]. Because the number of states of $\{C(t); t \geq 0\}$ is finite the stability condition presented by (31) itself is explicit. However, it seems that a simple scalar form in terms of given parameters is not easily obtainable.

Special Case. As a way to verify consistency, we apply the multiserver stability condition to the single-server case. For the matrices, we obtain

$$A^+ = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix}, \qquad A^- = \begin{pmatrix} 0 & 0 & \mu \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$A = \begin{pmatrix} -(\lambda+\alpha+\mu) & \alpha & \lambda \\ \nu_2 & -(\lambda+\nu_2) & 0 \\ \nu_1 & 0 & -\nu_1 \end{pmatrix},$$

$$P = \begin{pmatrix} -(\lambda+\alpha+\mu) & \alpha & \lambda+\mu \\ \nu_2 & -\nu_2 & 0 \\ \nu_1 & 0 & -\nu_1 \end{pmatrix}.$$

We have

$$p_{0,1} = \frac{\alpha}{\nu_2} p_{0,0}, \qquad p_{1,0} = \frac{\lambda+\mu}{\nu_1} p_{0,0}.$$

Thus, the stability condition (31) yields,

$$\lambda\frac{\alpha}{\nu_2} + \lambda\frac{\lambda+\mu}{\nu_1} < \mu,$$

which is consistent with (14), as should.

4.3 Stationary Distribution

In this section, we derive the stationary distribution for $\{X(t); t \geq 0\}$. Under the stability condition derived in the previous section, the stationary distribution exists. Let

$$\begin{aligned}\pi_{i,j,k} &= \lim_{t\to\infty} \Pr(S_1(t) = i, S_2(t) = j, N(t) = k), \qquad (i,j,k) \in \mathcal{S},\\ \boldsymbol{\pi}_{i,k} &= (\pi_{i,0,k}, \pi_{i,1,k}, \ldots, \pi_{i,c-i,k}),\\ \boldsymbol{\pi}_k &= (\boldsymbol{\pi}_{0,k}, \boldsymbol{\pi}_{1,k}, \ldots, \boldsymbol{\pi}_{c,k}).\end{aligned}$$

According to the matrix-analytic method, we have

$$\boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_k R, \qquad k \in \mathbb{Z}_+,$$

where the matrix R is the minimal nonnegative solution of

$$A^+ + AR + A^- R^2 = O,$$

for which several efficient numerical algorithms are available [14]. For example, R can be obtained as $\lim_{n\to\infty} R_n$, where $\{R_n; n \in \mathbb{Z}_+\}$ is defined by

$$R_0 = O, \qquad R_{n+1} = A^{-1}A^+ + A^{-1}A^- R_n^2, \qquad n \in \mathbb{Z}_+.$$

Furthermore, R can be also obtained by the matrix continued fraction approach presented in [15]. Finally, the boundary vector $\boldsymbol{\pi}_0$ is determined by

$$\boldsymbol{\pi}_0 A^0 + \boldsymbol{\pi}_1 A^- = \mathbf{0}, \qquad \sum_{k=0}^{\infty} \boldsymbol{\pi}_k \boldsymbol{e} = 1\,,$$

which is equivalent to

$$\boldsymbol{\pi}_0 (A^0 + RA^-) = \mathbf{0}, \qquad \boldsymbol{\pi}_0 (I - R)^{-1} \boldsymbol{e} = 1.$$

4.4 First Moments and Cost Model

We define the generating function for $\{\boldsymbol{\pi}_k; k \in \mathbb{Z}_+\}$ as

$$\boldsymbol{\pi}(z) = \sum_{k=0}^{\infty} \boldsymbol{\pi}_k z^k = \boldsymbol{\pi}_0 (I - zR)^{-1}.$$

Let $\boldsymbol{M}_n$, $n \in \mathbb{Z}_+$ denote the nth factorial moment vector of partial factorial moments $M_n^{(i,j)}$, $(i,j) \in \mathcal{V}$. We then have

$$\boldsymbol{M}_n = \frac{d}{dz}\boldsymbol{\pi}(z)\bigg|_{z=1} = \boldsymbol{\pi}_0 n! (I - R)^{-(n+1)} R^n.$$

Let $\pi_{i,j}^S = \lim_{t\to\infty} \Pr(S_1(t) = i, S_2(t) = j)$, $(i,j) \in \mathcal{V}$. We also define

$$\boldsymbol{\pi}_i^S = (\pi_{i,0}^S, \pi_{i,1}^S, \ldots, \pi_{i,c-i}^S), \qquad \boldsymbol{\pi}^S = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_c).$$

We then have

$$\boldsymbol{\pi}^S = \sum_{k=0}^{\infty} \boldsymbol{\pi}_k = \boldsymbol{\pi}_0 (I - R)^{-1}.$$

Let $\mathrm{E}[S_1]$ and $\mathrm{E}[S_2]$ denote the average number of incoming and outgoing calls in the servers, respectively. We have

$$\mathrm{E}[S_1] = \sum_{i=0}^{c} i \sum_{j=0}^{c-i} \pi_{i,j}^S, \qquad \mathrm{E}[S_2] = \sum_{i=0}^{c} \sum_{j=0}^{c-i} \pi_{i,j}^S j.$$

On the other hand, Little's formula yields

$$\mathrm{E}[S_1] = \frac{\lambda}{\nu_1}.$$

Let U denote the utilization of a server at the steady state, i.e.,

$$U = \frac{\mathrm{E}[S_1] + \mathrm{E}[S_2]}{c} = \frac{\lambda}{c\nu_1} + \frac{\mathrm{E}[S_2]}{c}.$$

From a management point of view, we need to minimize $1-U$, i.e., the fraction of time where the server is idle. At the same time, from a service point of view, we also need to minimize the average number of customers in the orbit $\mathrm{E}[N] = \boldsymbol{M}_1 \boldsymbol{e}$. These needs motivate us to consider an optimization problem finding the optimal value of the rate of outgoing calls.

$$\begin{aligned} \min \qquad & f(\alpha) = C_1(1-U) + C_2 \mathrm{E}[N], \\ s.t. \qquad & \boldsymbol{p}A^+\boldsymbol{e} < \boldsymbol{p}A^-\boldsymbol{e}, \qquad \alpha \geq 0, \end{aligned}$$

where the inequality is the stability condition, and C_1 and C_2 reflect the cost of idle server and of a retrial customer. Similar to the single-server case, the optimization consists in finding the optimal α while keeping all other parameters constant.

5 Conclusion

This paper presents the analysis of a two way communication retrial queue model applicable to a call center with balanced call blending. By assuming a constant retrial rate for the incoming calls, outgoing call activity is still possible when many incoming calls are in orbit, corresponding to balanced blending.

For the single server case, we derived the partial generating functions associated with the joint stationary distribution of the number of incoming calls and the system state. From this, we extracted explicit (closed-form) expressions for the involved probabilities, and also for the partial factorial moments. Both were also characterized with a recursive formulation. Further, the system's stability condition was derived, and a cost model was proposed. For the multiserver case, a formulation by a quasi-birth-and-death process was assumed. The involved matrices were derived, as well as an expression for the multiserver stability condition. Finally, also a numerical recipe for the stationary distribution was presented, and an associated cost model was proposed.

Acknowledgments. Tuan Phung-Duc is supported by a postdoctoral fellowship of JSPS. Wouter Rogiest is supported by a postdoctoral fellowship of Research Foundation - Flanders (FWO-Vlaanderen). Further, Wouter Rogiest received a travel grant of FWO-Vlaanderen for a work visit to Kyoto University, enabling this collaboration.

References

1. Falin, G.I., Templeton, J.G.C.: Retrial Queues. Chapman and Hall, Boca Raton (1997)
2. Artalejo, J.R., Gomez-Córral, A.: Retrial Queueing Systems: A Computational Approach. Springer, Berlin (2008)
3. Phung-Duc, T., Masuyama, H., Kasahara, S., Takahashi, Y.: State-dependent M/M/c/c+r retrial queues with bernoulli abandonment. Journal of Industrial and Management Optimization 6(3), 517–540 (2010)
4. Koole, G., Mandelbaum, A.: Queueing models of call centers: an introduction. Annals of Operations Research 113(1), 41–59 (2002)
5. Bhulai, S., Koole, G.: A queueing model for call blending in call centers. IEEE Transactions on Automatic Control 48, 1434–1438 (2003)
6. Artalejo, J.R., Phung-Duc, T.: Markovian single server retrial queues with two way communication. In: Proceedings of the 6th International Conference on Queueing Theory and Network Applications, QTNA 2011, pp. 1–7. ACM, New York (2011)
7. Bernett, H.G., Fischer, M.J., Masi, D.M.B.: Blended call center performance analysis. IT Professional 4, 33–38 (2002)
8. Deslauriers, A., L'Ecuyer, P., Pichitlamken, J., Ingolfsson, A., Avramidis, A.N.: Markov chain models of a telephone call center with call blending. Computers & Operations Research 34, 1616–1645 (2007)
9. Atencia, I., Moreno, P.: A single-server retrial queue with general retrial times and Bernoulli schedule. Applied Mathematics and Computation 162(2), 855–880 (2005)
10. Atencia, I., Fortes, I., Moreno, P., Sánchez, S.: An M/G/1 retrial queue with active breakdowns and Bernoulli schedule in the server. Information and Management Sciences 17(1), 1–17 (2006)
11. Fayolle, G.: A simple telephone exchange with delayed feedbacks. In: Proc. of the International Seminar on Teletraffic Analysis and Computer Performance Evaluation, pp. 245–253. North-Holland Publishing Co., Amsterdam (1986)
12. Farahmand, K.: Single line queue with repeated demands. Queueing Systems 6, 223–228 (1990)
13. Martin, M., Artalejo, J.R.: Analysis of an M/G/1 queue with two types of impatient units. Advances in Applied Probability 27(3), 840–861 (1995)
14. Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modelling. SIAM, Philadelphia (1999)
15. Phung-Duc, T., Masuyama, H., Kasahara, S., Takahashi, Y.: A simple algorithm for the rate matrices of level-dependent QBD processes. In: Proceedings of the 5th International Conference on Queueing Theory and Network Applications, QTNA 2010, pp. 46–52. ACM, New York (2010)

Analysis of a Two-Class FCFS Queueing System with Interclass Correlation

Herwig Bruneel, Tom Maertens, Bart Steyaert, Dieter Claeys,
Dieter Fiems, and Joris Walraevens

Ghent University,
Department of Telecommunications and Information Processing,
SMACS Research Group,
Sint-Pietersnieuwstraat 41,
B-9000 Ghent, Belgium
{hb,tmaerten,bs,dclaeys,df,jw}@telin.UGent.be

Abstract. This paper considers a discrete-time queueing system with one server and two classes of customers. All arriving customers are accommodated in one queue, and are served in a First-Come-First-Served order, regardless of their classes. The total numbers of arrivals during consecutive time slots are i.i.d. random variables with arbitrary distribution. The classes of consecutively arriving customers, however, are correlated in a Markovian way, i.e., the probability that a customer belongs to a class depends on the class of the previously arrived customer. Service-time distributions are assumed to be general but class-dependent. We use probability generating functions to study the system analytically. The major aim of the paper is to estimate the impact of the *interclass correlation* in the arrival stream on the queueing performance of the system, in terms of the (average) number of customers in the system and the (average) customer delay and customer waiting time.

1 Introduction

Various types of scheduling disciplines have been investigated within the context of multi-class queueing systems. We mention, among others, priority scheduling (see, e.g., [4,8,11,13,15]), weighted fair queueing (WFQ) (see, e.g., [14,17]), random order of service (ROS) (see, e.g., [1,3,10]), and generalized processor sharing (GPS) (see, e.g., [9,12,16]). Strangely enough, only few results have been derived for multi-class First-Come-First-Served (FCFS) systems, i.e., queueing systems in which the customers of different classes are accommodated in one queue and served in their order of arrival, irrespective of the classes they belong to (a recent paper is [5]). The present paper presents the analysis of a discrete-time model that fits in this category.

In classical multi-class queueing models, furthermore, it is generally assumed that the different classes occur randomly and independently in the arrival stream of customers (this is also the case in [5]). In this paper, however, we explicitly wish to examine the effect of so-called *interclass correlation* (or *class clustering*)

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 32–46, 2012.

in the arrival process. Specifically, we are interested to know whether the degree to which customers of the same class have the tendency to arrive (and be served) closely together (i.e., back-to-back), or, conversely, the degree to which such customers have the tendency to be spread in time and mixed with customers of the other class, has a substantial impact on the performance of a *two-class* FCFS queueing system. In order to do so, we superimpose a two-state Markovian interclass correlation model (with arbitrary transition probabilities) on top of a regular general independent arrival process model for the aggregated customer stream. Service-time distributions are class-dependent but completely general. It is clear that the interclass correlation between consecutive customers can also be viewed as a form of non-independence between service times. One application of this queueing model is obvious: the two customers classes can model, for example, voice and data packets in a heterogeneous telecommunication system. It is common knowledge that data packets are significantly larger than voice packets. Then it is easy to see that if data packets have the tendency to arrive in clusters, the performance of the system may be degraded severely (with respect to voice packets). In this paper, we measure this degradation.

We first derive the probability generating function (pgf) of the total number of customers in the system at customer departure times. From this result, we can easily obtain the corresponding pgf valid at arbitrary slot boundaries. Various performance measures of practical use, such as the mean system content, the mean customer delay and the mean customer waiting time, can be easily derived from these pgf's by applying the moment-generating property of pgf's and by using Little's law. The resulting formulas and a number of numerical examples reveal that the system under study can exhibit two types of stochastic equilibrium, depending on the values of the system parameters: a "strong" equilibrium in which both customer classes individually generate less work than the system can handle (during periods where only such customers arrive), and a "compensated" type of equilibrium whereby one customer class creates overload situations which are compensated by strong under-load periods generated by the other customer class. In the latter case, our results clearly demonstrate the crucial importance of the amount of interclass correlation on the usual performance parameters of the system.

The outline of this paper is as follows. In Section 2, we describe the mathematical model. Section 3 first presents an analysis of the total number of customers in the system at customer departure times; next, the pgf of the system content at random slot boundaries is derived from this result. We discuss the results, both conceptually and quantitatively, in Section 4. Some conclusions are drawn in Section 5.

2 Mathematical Model

We consider a discrete-time queueing system with infinite waiting room, one server, and two classes of customers, named A and B. As in all discrete-time models, the time axis is divided into fixed-length intervals referred to as *slots* in

the sequel. New customers may enter the system at any given (continuous) point on the time axis, but services are synchronized to (i.e., can only start and end at) slot boundaries. Customers are served in their order of arrival, regardless of the class they belong to. We call this service discipline "global FCFS".

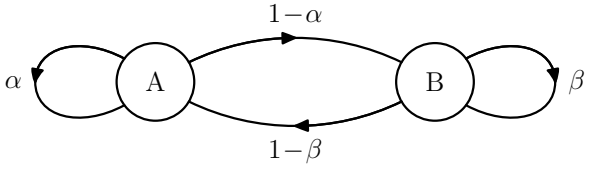


Fig. 1. Two-state Markov chain of the customer classes

The arrival process of new customers in the system is characterized in two steps. First, we model the total (aggregated) arrival stream of new customers by means of a sequence of i.i.d. non-negative discrete random variables (denoting the numbers of arrivals in consecutive slots) with common probability generating function (pgf) $E(z)$. The (total) mean number of arrivals per slot, in the sequel referred to as the (total) mean arrival rate, is given by $\lambda \triangleq E'(1)$. Next, we describe the occurrence of the two classes in the sequence of the consecutively arriving customers. In this study, we assume that both classes of customers account for part of the total load of the system, i.e., both customer classes are "mixed" in the arrival stream, but there may be some degree of "class clustering" in the arrival process, i.e., customers of any given class may (or may not) have a tendency to "arrive back-to-back". Mathematically, this means that the classes of two consecutive customers may be non-independent. Specifically, we assume a first-order Markovian type of correlation between the classes of two consecutively arriving customers, which basically means that the probability that the next customer belongs to a given class depends on the class of the previously arrived customer. Let t_k denote the class of customer k. The transition probabilities of the Markov chain that determines the class of the consecutively arriving customers are then defined as (see Fig. 1)

$$\text{Prob}[t_{k+1} = A \,|\, t_k = A] \triangleq \alpha \ , \tag{1}$$

$$\text{Prob}[t_{k+1} = B \,|\, t_k = A] \triangleq 1 - \alpha \ , \tag{2}$$

$$\text{Prob}[t_{k+1} = A \,|\, t_k = B] \triangleq 1 - \beta \ , \tag{3}$$

$$\text{Prob}[t_{k+1} = B \,|\, t_k = B] \triangleq \beta \ . \tag{4}$$

It is well known that for a two-state Markov chain of this type, the steady-state probabilities t_A and t_B of finding the chain in state A and B are given by

$$t_A \triangleq \lim_{k\to\infty} \text{Prob}[t_k = A] = \frac{1-\beta}{2-\alpha-\beta} \tag{5}$$

and

$$t_B \triangleq \lim_{k\to\infty} \text{Prob}[t_k = B] = \frac{1-\alpha}{2-\alpha-\beta} \ , \tag{6}$$

respectively (see, e.g., [2]). The quantities t_A and t_B can be interpreted as the fractions of class A and class B customers in the arrival stream. The (steady-state) correlation coefficient of the Markov chain, i.e., the amount of correlation between the classes of two consecutively arriving customers (in the steady state), is given by

$$\gamma \triangleq \lim_{k\to\infty} \frac{\text{E}[t_k t_{k+1}] - \text{E}[t_k]\text{E}[t_{k+1}]}{\sqrt{\text{var}[t_k]\,\text{var}[t_{k+1}]}} = \alpha + \beta - 1 \ . \tag{7}$$

We will indicate the parameter γ ($-1 \leq \gamma \leq +1$) as the *interclass correlation* in the sequel. Positive values of γ correspond to a situation whereby the customers of any given class have a tendency to cluster, while negative values of γ refer to arrival streams in which the customers of classes A and B have a tendency to alternate, i.e., be mixed more strongly. The case where $\gamma = 0$, of course, corresponds to the classical assumption that classes of subsequent customers are independent.

The service process of the system is characterized by attaching to each customer a corresponding *service time*, which indicates the number of time slots required to give complete service to the customer at hand. The service times of customers are class-dependent and are modelled as a sequence of independent positive discrete random variables with pgf's $A(z)$ and $B(z)$. The corresponding mean values are given by $\mu_A \triangleq A'(1)$ and $\mu_B \triangleq B'(1)$ for customers of class A and B, respectively.

3 System Analysis

3.1 System Equations at Customer Departure Times

Let u_k denote the total *system content*, i.e., the total number of customers present in the system just after the service completion of the k-th customer, and, as before, let t_k indicate the class customer k belongs to. Then, as a consequence of all the model assumptions in Section 2, the couple (t_k, u_k) forms a Markovian state description of the system (at customer departure times).

The state transitions of the quantities $\{t_k\}$ are governed by the Eqs. (1)-(4), whereas for the quantities $\{u_k\}$, the following recursive system equations can be established (see Figs. 2 and 3):

$$u_{k+1} = \begin{cases} u_k - 1 + g_{k+1} & \text{if } u_k > 0 \\ f_{k+1} + g_{k+1} & \text{if } u_k = 0 \end{cases} . \tag{8}$$

Here, the quantity g_{k+1} is defined as the number of arrivals in the system during the service time of customer $k+1$, while f_{k+1} indicates the number of customers arriving *after* customer $k+1$ in its arrival slot.

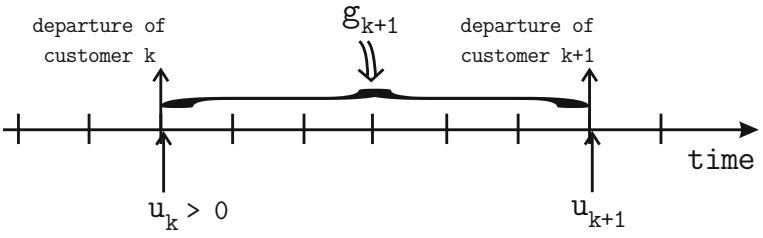


Fig. 2. Relationship between u_k and u_{k+1} when $u_k > 0$

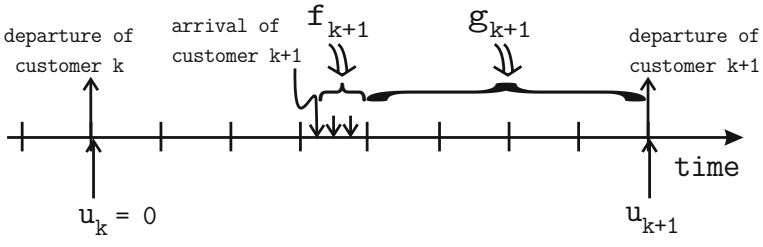


Fig. 3. Relationship between u_k and u_{k+1} when $u_k = 0$

It is easily seen that the pgf of f_{k+1} is given by the pgf of the number of additional arrivals in a slot with at least one arrival, i.e.,

$$F(z) \triangleq \mathrm{E}[z^{f_{k+1}}] = \frac{E(z) - E(0)}{z[1 - E(0)]} \ , \tag{9}$$

regardless of the class of customer $k+1$. The distribution of the quantity g_{k+1}, however, does depend on the class of customer $k+1$. We have

$$G_A(z) \triangleq \mathrm{E}[z^{g_{k+1}} \,|\, t_{k+1} = A] = A(E(z)) \ , \tag{10}$$

$$G_B(z) \triangleq \mathrm{E}[z^{g_{k+1}} \,|\, t_{k+1} = B] = B(E(z)) \ . \tag{11}$$

3.2 System Content at Customer Departure Times

Let us assume that the queueing system at hand is stable, i.e., that the stability condition is fulfilled. Intuitively, it is not difficult to see that the system is stable if and only if the average amount of work entering the system per slot is strictly less than 1, i.e., if and only if $\lambda \mathrm{E}[c] < 1$, with $\mathrm{E}[c]$ the average service time of an arbitrary customer. Expressed in the basic parameters of our system, this is equivalent to the condition

$$\lambda(t_A \mu_A + t_B \mu_B) < 1 \ , \tag{12}$$

where the quantities t_A and t_B are the steady-state probabilities of the arrival Markov chain (see Eqs. (5) and (6)). Assuming this condition fulfilled, we define the joint steady-state probabilities of the Markov chain $\{(t_k, u_k)\}$ as

$$p_A(i) \triangleq \lim_{k\to\infty} \text{Prob}[t_k = A, u_k = i] \tag{13}$$

and

$$p_B(i) \triangleq \lim_{k\to\infty} \text{Prob}[t_k = B, u_k = i] \quad , \tag{14}$$

for all $i \geq 0$. The corresponding partial pgf's are defined as $P_A(z)$ and $P_B(z)$. Then the steady-state pgf $P(z)$ of the total system content at customer departure times is equal to $P_A(z) + P_B(z)$.

We now establish two linear equations for $P_A(z)$ and $P_B(z)$. We depart from the balance equations of the Markov chain $\{(t_k, u_k)\}$ for class A:

$$\begin{aligned} p_A(j) = &\sum_{i=0}^{\infty} p_A(i)\alpha \lim_{k\to\infty} \text{Prob}[u_{k+1} = j \,|\, t_{k+1} = A, u_k = i] \\ &+ \sum_{i=0}^{\infty} p_B(i)(1-\beta) \lim_{k\to\infty} \text{Prob}[u_{k+1} = j \,|\, t_{k+1} = A, u_k = i] \quad . \end{aligned} \tag{15}$$

Next, we introduce pgf's into this equation:

$$\begin{aligned} P_A(z) = &\alpha \sum_{i=0}^{\infty} p_A(i) \lim_{k\to\infty} \text{E}[z^{u_{k+1}} \,|\, t_{k+1} = A, u_k = i] \\ &+ (1-\beta) \sum_{i=0}^{\infty} p_B(i) \lim_{k\to\infty} \text{E}[z^{u_{k+1}} \,|\, t_{k+1} = A, u_k = i] \quad . \end{aligned} \tag{16}$$

The expected values in (16) can be further developed by using the system equations (see Eq. 8):

$$\begin{aligned} \lim_{k\to\infty} \text{E}[z^{u_{k+1}} \,|\, t_{k+1} = A, u_k = i] &= \lim_{k\to\infty} \text{E}[z^{i-1+g_{k+1}} \,|\, t_{k+1} = A] \\ &= z^{i-1} G_A(z) \quad , \end{aligned} \tag{17}$$

for $i \geq 1$, and

$$\begin{aligned} \lim_{k\to\infty} \text{E}[z^{u_{k+1}} \,|\, t_{k+1} = A, u_k = 0] &= \lim_{k\to\infty} \text{E}[z^{f_{k+1}+g_{k+1}} \,|\, t_{k+1} = A] \\ &= F(z) G_A(z) \quad . \end{aligned} \tag{18}$$

Putting everything together, we then obtain

$$\begin{aligned} P_A(z) = &\alpha \frac{G_A(z)}{z} [P_A(z) - P_A(0)] + \alpha P_A(0) F(z) G_A(z) \\ &+ (1-\beta) \frac{G_A(z)}{z} [P_B(z) - P_B(0)] + (1-\beta) P_B(0) F(z) G_A(z) \quad . \end{aligned} \tag{19}$$

Using Eqs. (9)-(11), we finally obtain a first linear equation between $P_A(z)$ and $P_B(z)$:

$$[z - \alpha A(E(z))]\, P_A(z) - (1-\beta)A(E(z))P_B(z) = \frac{E(z)-1}{1-E(0)}\,[\alpha P_A(0) + (1-\beta)P_B(0)]\, A(E(z)) \quad . \tag{20}$$

Starting from the balance equations for class B, we can derive a second, similar equation:

$$[z - \beta B(E(z))]\, P_B(z) - (1-\alpha)B(E(z))P_A(z) = \frac{E(z)-1}{1-E(0)}\,[\beta P_B(0) + (1-\alpha)P_A(0)]\, B(E(z)) \quad . \tag{21}$$

Eqs. (20) and (21) form a set of two linear equations for the two unknown partial pgf's $P_A(z)$ and $P_B(z)$. Expressions for these pgf's can be found by solving the set. Then adding up $P_A(z)$ and $P_B(z)$ leads to the following expression for the pgf $P(z)$:

$$P(z) = \frac{P(0)(E(z)-1)}{1-E(0)} \times \frac{z[p_A A(E(z)) + p_B B(E(z))] + (1-\alpha-\beta)A(E(z))B(E(z))}{z^2 - z[\alpha A(E(z)) + \beta B(E(z))] - (1-\alpha-\beta)A(E(z))B(E(z))} \quad , \tag{22}$$

where the quantities p_A and p_B are defined as

$$p_A \triangleq \frac{\alpha P_A(0) + (1-\beta)P_B(0)}{P(0)} \tag{23}$$

and

$$p_B \triangleq \frac{(1-\alpha)P_A(0) + \beta P_B(0)}{P(0)} \quad , \tag{24}$$

respectively. It is not difficult to see that p_A and p_B denote the conditional probabilities that a customer entering an empty system (in the steady state) belongs to class A or B: $p_X \triangleq \lim_{k\to\infty} \mathrm{Prob}[t_{k+1} = X \mid u_k = 0]$, with $X \in \{A, B\}$.

The probability $P(0)$ can be derived explicitly from the normalization condition of the pgf $P(z)$, i.e., the condition $P(1) = 1$. The result is given by

$$P(0) = \frac{1-E(0)}{\lambda}\,[1 - \lambda(t_A\mu_A + t_B\mu_B)] = \frac{1-E(0)}{\lambda}\,\{1 - \lambda \mathrm{E}[c]\} \quad , \tag{25}$$

where, as before, the quantities t_A and t_B are the steady-state probabilities of the arrival Markov chain, defined in Eqs. (5) and (6), and $\mathrm{E}[c]$ denotes the average service time of an arbitrary customer. It then remains for us to calculate the two unknown probabilities p_A and p_B, of which we know from (23) and (24) that $p_A + p_B = 1$. The unknowns can be determined, in general, by invoking the

well-known property that pgf's such as $P(z)$ are bounded inside the closed unit disk $\{z : |z| \leq 1\}$ of the complex z-plane, at least when the stability condition (12) of the queueing system is met (only in such a case our analysis was justified and $P(z)$ can be viewed as a legitimate pgf). Now, it can be shown by means of Rouché's theorem from complex analysis [2, 7] that the denominator of Eq. (22) has exactly two zeroes inside the closed unit disk of the complex z-plane, one of which is equal to 1, as soon as the stability condition (12) is fulfilled. It is clear that these two zeroes should also be zeroes of the numerator of Eq. (22), as $P(z)$ must remain bounded in those points. For the zero $z = 1$, this condition is fulfilled regardless of the values of the unknowns p_A and p_B, since the numerator of (22) contains a factor $E(z) - 1$. However, for the second zero, say $z = \hat{z}$, the requirement that the numerator should vanish yields a linear equation for the two unknowns. A second linear equation is given by $p_A + p_B = 1$. In general, i.e., when the pgf's $A(z)$ and $B(z)$ are different, the two unknown probabilities p_A and p_B can be found as the solutions of the two established linear equations. We obtain

$$p_A = \frac{\alpha A(E(\hat{z})) - (1-\beta)B(E(\hat{z})) - \hat{z}}{A(E(\hat{z})) - B(E(\hat{z}))} \tag{26}$$

and

$$p_B = \frac{\beta B(E(\hat{z})) - (1-\alpha)A(E(\hat{z})) - \hat{z}}{B(E(\hat{z})) - A(E(\hat{z}))} \ . \tag{27}$$

Once the zero $\hat{z}$ has been computed (numerically), p_A and p_B can be derived from (26) and (27). Substitution of the obtained values and of Eq. (25) in (22) then leads to a fully determined expression of the steady-state pgf $P(z)$ of the total system content at customer departure times.

3.3 System Content at Random Slot Boundaries

It has been shown in [2] that in any discrete-time queueing system with one single server and independent arrivals from slot to slot (with pgf $E(z)$), regardless of the precise characteristics of the service process and the intra-slot details of the arrival process (the position of the arrival instants within the slot, single arrivals or batch arrivals, etc.), the following simple relationship is valid between the pgf $S(z)$ of the system content at random slot boundaries and the pgf $P(z)$ valid at customer departure times:

$$P(z) = \frac{E(z) - 1}{\lambda(z-1)} \ S(z) \ . \tag{28}$$

In the previous subsection, we have found an expression for the pgf $P(z)$. Hence, it is easy to obtain an expression for $S(z)$. From $S(z)$, various performance

measures of practical importance can be derived. For instance, the mean system content at random slot marks can be found as $\mathrm{E}[s] = S'(1)$. After long and tedious calculations, this results in

$$\mathrm{E}[s] = \rho + \frac{\lambda^2 C''(1) + E''(1)C'(1)}{2(1-\rho)} + \frac{\gamma t_A t_B \lambda^2 (\mu_A - \mu_B)^2}{(1-\gamma)(1-\rho)} + \frac{\lambda(p_A - t_A)(\mu_A - \mu_B)}{1-\gamma}, \tag{29}$$

where t_A and t_B are expressed in Eqs. (5) and (6), γ is the interclass correlation defined in (7), $C'(1)$ and $C''(1)$ are derivatives of the pgf $C(z)$ of the service time of an arbitrary customer (i.e., $C(z) \triangleq t_A A(z) + t_B B(z)$), ρ $(= \lambda C'(1))$ is the total load of the queueing system, and p_A and p_B are the unknown probabilities defined in (23) and (24) and calculated as (26) and (27) (as soon as the zero $\hat{z}$ has been determined numerically). The first term (ρ) in Eq. (29) corresponds to the mean number of customers in service, the other three terms account for the mean *queue content*, i.e., the mean number of customers that are actually waiting to be served.

Higher-order moments of the system-content distribution can be obtained by computing higher-order derivatives of the pgf $S(z)$. By applying (the discrete-time version of) Little's law (see, e.g., [6]), the mean *delay* (system time) of an arbitrary customer can be obtained as $\mathrm{E}[d] = \mathrm{E}[s]/\lambda$. The mean *waiting time* of an arbitrary customer can be derived from this as $\mathrm{E}[w] = \mathrm{E}[d] - \mathrm{E}[c]$:

$$\mathrm{E}[w] = \frac{\lambda^2 C''(1) + E''(1)C'(1)}{2\lambda(1-\rho)} + \frac{\gamma t_A t_B \lambda (\mu_A - \mu_B)^2}{(1-\gamma)(1-\rho)} + \frac{(p_A - t_A)(\mu_A - \mu_B)}{1-\gamma}. \tag{30}$$

4 Discussion of Results and Numerical Examples

In this section, we discuss the results, both from a qualitative perspective and by means of some numerical examples. The first interesting result obtained is the form of the stability condition $\lambda < \dfrac{1}{\mathrm{E}[c]} = \dfrac{1}{t_A \mu_A + t_B \mu_B}$, which shows that the maximum achievable throughput of this system, expressed in customers per slot, is completely determined by the mean service time of an arbitrary customer, regardless of the possible interclass correlation.

Next, we focus on the mean system content at random slot marks (see Eq. (29)). This result explicitly and very clearly shows the influence of the various system parameters on the performance of the system. As could be expected intuitively, the mean system content depends on the first two moments of the arrival process (as represented by the quantities λ and $E''(1)$, and to some extent $\rho = \lambda C'(1)$) and the first two moments of the service times (contained in the quantities $C'(1)$, $C''(1)$, μ_A, μ_B, and also $\rho = \lambda C'(1)$). It is not surprising that $\mathrm{E}[s]$ goes to infinity as ρ approaches its limiting value 1, dictated by the stability condition of the system. However, it is striking that $\mathrm{E}[s]$ also seems to increase without bound

if the interclass correlation $\gamma = \alpha + \beta - 1$ approaches the value +1, even when the stability condition $\rho < 1$ is met. Positive interclass correlation appears to be very detrimental for the performance of the system, whereas negative interclass correlation has a very moderate positive effect on the performance.

The first two terms in Eq. (29) correspond to the classical result that would be obtained in a system without interclass correlation and with service-time pgf $C(z)$ (see, e.g., [2]). This means that the third and fourth term in (29) can be fully attributed to the presence of interclass correlation in the arrival process. We note, indeed, that the third term vanishes when $\gamma = 0$; in the fourth term, both t_A and p_A reduce to the same value α when $\gamma = 0$ (see Eqs. (5) and (26), with $\hat{z} = 0$), which implies that the fourth term is equal to zero as well in that case. It is easy to see that the third and fourth term also disappear when all customers have the same service-time distribution, i.e., when $A(z) = B(z)$ and, hence, $\mu_A = \mu_B$, and, finally, when there is only one class of customers in the system, i.e., when either $\alpha = 1$ (and, hence, $p_A = t_A = 1$ and $t_B = 0$) or $\beta = 1$ (and, therefore, $p_A = t_A = 0$).

Let us now consider some numerical results. In a first example, we assume Poisson arrivals (i.e., $E(z) = e^{\lambda(z-1)}$), equal fractions of both classes of customers in the arrival stream (i.e., $t_A = t_B = 0.5$), geometrically distributed service times for both classes, i.e.,

$$X(z) = \frac{z}{\mu_X + (1 - \mu_X)z} \quad , \tag{31}$$

with $X \in \{A, B\}$, and with $\mu_A = 8$ and $\mu_B = 2$. The stability condition is then given by $\rho = \lambda[t_A\mu_A + t_B\mu_B] = 5\lambda < 1$ (i.e., $\lambda < 0.2$).

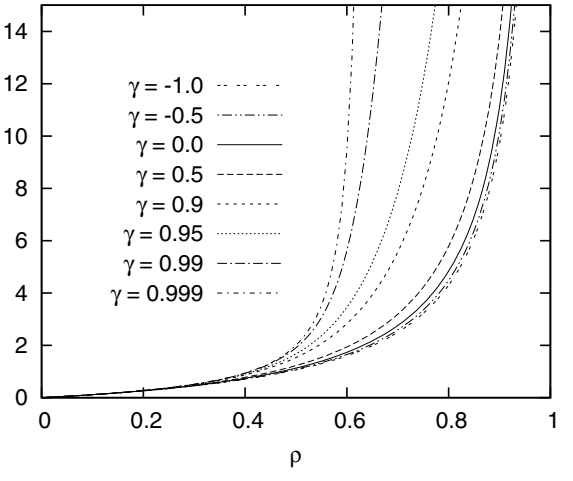


Fig. 4. Mean system content E[s] versus load ρ for various values of the interclass correlation γ

Fig. 4 shows the mean system content $\mathrm{E}[s]$ as a function of the load ρ, for various values of the interclass correlation γ. The figure confirms that, for given values of $\rho < 1$, the parameter γ has a major impact on the results when it is positive and only a minor influence when it is negative. An intuitive explanation of this phenomenon lies in the observation that the numbers of consecutive class-A and class-B customers in the arrival stream both increase dramatically as γ approaches the value $+1$. Indeed, the mean number of class-A (class-B) customers between two consecutive class-B (class-A) customers is given by $\dfrac{1}{1-\alpha} = \dfrac{2}{1-\gamma}$ ($\dfrac{1}{1-\beta} = \dfrac{2}{1-\gamma}$). For negative values of γ, this implies that customers of both classes alternate strongly; for positive values of γ, there may be very long sequences of customers of the same class. During such periods, the momentary load is either given by $\rho_A \triangleq \lambda\mu_A = 8\lambda$ or by $\rho_B \triangleq \lambda\mu_B = 2\lambda$. It is easily seen that the stability condition $\rho < 1$, or $\lambda < 0.2$, guarantees that $\rho_B < 1$, but not necessarily that $\rho_A < 1$. It is clear that if λ or ρ are small enough (more specifically, $\lambda < 0.125$ or $\rho < 0.625$), $\rho_A < 1$ and $\rho_B < 1$, i.e., the system is locally stable both during A- and B-sequences (and, hence, also globally stable - we call this the "strong" equilibrium), while if $0.125 \leq \lambda < 0.2$, or, equivalently, $0.625 \leq \rho < 1$, $\rho_B < 1$ but $\rho_A > 1$, i.e., the system is locally stable during B-sequences but not during A-sequences. In the latter case, labelled as the "compensated" equilibrium, (global) stability is assured because although the queue size builds up during A-sequences (because, on average, more work arrives than the server can perform), it decreases again during B-sequences (when much less work enters than the server can execute). In other words, the overload periods created by the A-customers are *compensated* by the underload periods of the B-customers. When the interclass correlation approaches $+1$, however, the amplitude of these queue size variations goes to infinity, implying that the mean system content does the same.

The same behavior can be observed in Fig. 5, where we have plotted $\mathrm{E}[s]$ as a function of γ for various values of ρ. The figure illustrates very clearly that the system content grows without bound as $\gamma \to +1$ when ρ is higher than its critical value 0.625. When ρ is less than this critical value, on the other hand, the mean system content remains finite for all values of γ. Although we have explained this behaviour intuitively in the previous paragraph, it is somewhat unexpected in view of Eq. (29). Indeed, Eq. (29) seems to say that $\mathrm{E}[s]$ should become unbounded as $\gamma \to +1$, *regardless of the other system parameters*. The third and fourth term in (29) both approach infinity for $\gamma \to 1$; however, when ρ is less than its critical value, the terms cancel each other causing their sum to remain finite.

A second example is treated in Figs. 6 and 7. Again, we assume Poisson arrivals and geometrically distributed service times for both classes. Here, however, $\mu_A = 100$ and $\mu_B = 10$. The interclass correlation γ is kept constant at 0.8. This implies that $\alpha = 0.8 + 0.2t_A$, $\beta = 1 - 0.2t_A$, and $\rho = 10\lambda(1 + 9t_A)$. We now investigate the impact of the parameter t_A, i.e., the fraction of class-A customers in the arrival stream, on the mean system content and the mean waiting times

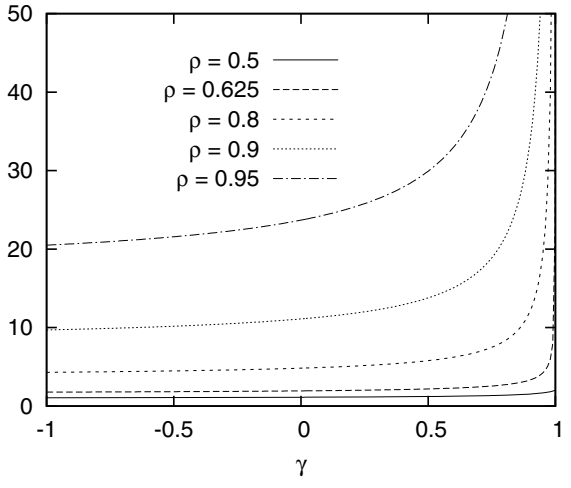


Fig. 5. Mean system content E[s] versus interclass correlation γ for various values of the load ρ

of the customers. Fig. 6 shows the mean system content E[s] versus t_A, for various values of ρ, whereas Fig. 7 illustrates the corresponding results for the mean waiting time E[w] of the customers. Fig. 6 reveals that, for any given value of the total load ρ, the mean system content increases as a function of t_A for "low" values of t_A (more or less in the interval $0 \leq t_A \leq 0.1$), then reaches a maximum value for t_A somewhere around 0.1, and, finally decreases monotonically in the interval $0.1 \leq t_A \leq 1$. An intuitive explanation might be as follows. For $t_A = 0$, all customers belong to class B (with a short service time of 10 slots); as soon as t_A becomes positive, say $0 \leq t_A \leq 0.1$, class-A customers (with a long service time of 100 slots) arrive sporadically and (when in service) somehow block the regular processing of class-B customers, which causes the system content to increase. If, however, t_A increases further (while the total load ρ remains constant), the system receives considerably less customers (for the same amount of work), which explains the decreasing system content in the interval $0.1 \leq t_A \leq 1$.

The behaviour of the mean waiting time (see Fig. 7) is qualitatively a bit similar as for the mean system content. More specifically, it can be observed that the mean waiting times also increase for "low" values of t_A to reach a maximum value and then decrease for "higher" values of t_A. However, the maximum value of the waiting time is attained for t_A around 0.25, whereas the highest mean system content occurs for t_A in the vicinity of 0.1. Also, the rates at which the mean waiting times increase and decrease seem relatively slower than for the mean system content. Intuitively, this can be attributed to the fact that the waiting time reflects the unfinished work in the system (at the arrival instant of a customer), while the system content indicates the number of customers in the

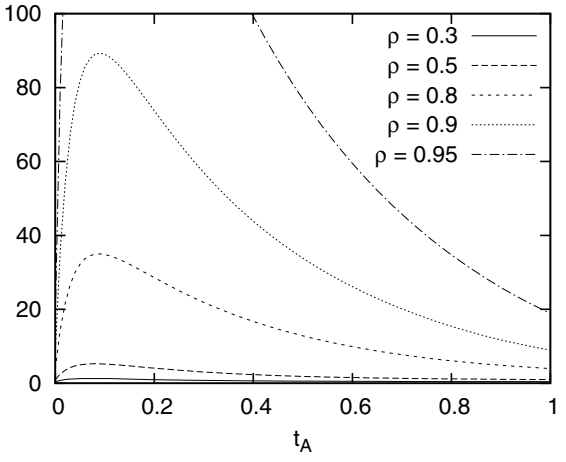


Fig. 6. Mean system content E[s] versus the fraction t_A of class-A customers for various values of the load ρ

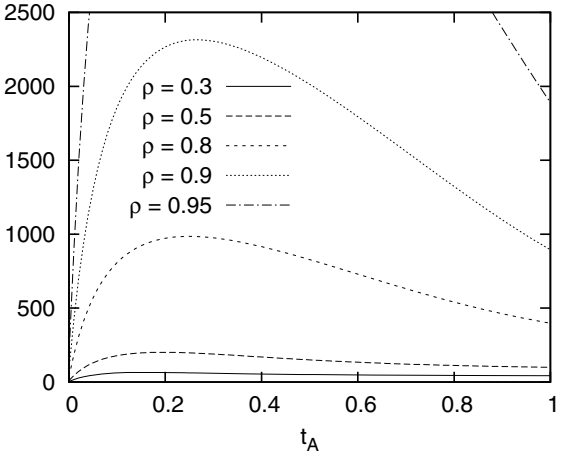


Fig. 7. Mean waiting time E[w] versus the fraction t_A of class-A customers for various values of the load ρ

system, whereby all customers contribute identically, irrespective of their service time, i.e., irrespective of the amount of work they represent. The fact that the mean waiting time (and, hence, the unfinished work in the system) for $t_A = 0$ is substantially smaller than for all other values of t_A can be explained by the higher burstiness of the arrival process of work units if class-B customers (bringing small amounts of work) are alternated with class-A customers (bringing large batches of work at the same time), which happens as soon as t_A gets positive.

5 Conclusions

In this paper, we have studied a discrete-time queueing system with one server and two classes of customers, and operating under the global FCFS service discipline. We have assumed independent (aggregated) arrivals from slot to slot combined with a general first-order Markovian interclass correlation model, and general but class-dependent service-time distributions. We have been able to derive the main performance measures of the system in semi-analytical form, i.e., we have obtained explicit expressions for such quantities as the mean system content and the mean customer waiting time in terms of the basic parameters of the model and one parameter which is only implicitly known through a non-linear equation that it satisfies.

The results reveal that the interclass correlation does not have any effect on the stability condition of the system, but it may have a very direct and great influence on the main performance measures of the system. More specifically, when the system is (globally) stable, we have observed that two different kinds of global equilibrium are possible, depending on the exact value of the load. For "low" values of the load, the system exhibits a "strong" equilibrium, whereas for higher loads, the system reaches a "compensated" type of equilibrium. Especially in the latter case, the impact of strong positive interclass correlation may be devastating for the queueing performance. We therefore believe that the phenomenon of class clustering in the context of multi-class queueing systems deserves more attention than it traditionally has received in the classical queueing literature.

Acknowledgment. The last two authors are Postdoctoral Fellows with the Fund for Scientific Research, Flanders (F.W.O.-Vlaanderen), Belgium.

References

1. Borst, S.C., Boxma, O.J., Morrison, J.A., Queija, R.N.: The equivalence between processor sharing and service in random order. Operations Research Letters 31(4), 254–262 (2003)
2. Bruneel, H., Kim, B.G.: Discrete-time models for communication systems including ATM. Kluwer Academic, Boston (1993)
3. Carter, G.M., Cooper, R.B.: Queues with service in random order. Operations Research 20(2), 389–405 (1970)
4. Chen, J., Guérin, R.: Performance study of an input queueing packet switch with two priority classes. IEEE Transactions on Communications 39(1), 117–126 (1991)
5. De Clercq, S., Laevens, K., Steyaert, B., Bruneel, H.: A multi-class discrete-time queueing system under the FCFS service discipline. Annals of Operations Research (accepted for publication)
6. Fiems, D., Bruneel, H.: A note on the discretization of Little's result. Operations Research Letters 30, 17–18 (2002)
7. Gonzáles, M.O.: Classical complex analysis. Marcel Dekker, New York (1992)
8. Jaiswal, N.: Priority queues. Academic Press, New York (1968)

9. Jin, X., Min, G.: Analytical modelling and evaluation of generalized processor sharing systems with heterogeneous traffic. International Journal of Communication Systems 21(6), 571–586 (2008)
10. Kim, J., Kim, J., Kim, B.: Analysis of the M/G/1 queue with discriminatory random order service policy. Performance Evaluation 68(3), 256–270 (2011)
11. Laevens, K., Bruneel, H.: Discrete-time multiserver queues with priorities. Performance Evaluation 33(4), 249–275 (1998)
12. Lieshout, P., Mandjes, M.: Generalized processor sharing: Characterization of the admissible region and selection of optimal weights. Computers & Operations Research 35(8), 2497–2519 (2008)
13. Maertens, T., Walraevens, J., Bruneel, H.: Performance comparison of several priority schemes with priority jumps. Annals of Operations Research 180(3), 1168–1185 (2008)
14. Shortle, J.F., Fischer, M.J.: Approximation for a two-class weighted fair queueing discipline. Performance Evaluation 67(10), 946–958 (2010)
15. Walraevens, J., Fiems, D., Wittevrongel, S., Bruneel, H.: Calculation of output characteristics of a priority queue through a busy period analysis. European Journal of Operational Research 198(3), 891–898 (2009)
16. Walraevens, J., van Leeuwaarden, J.S.H., Boxma, O.J.: Power series approximations for two-class generalized processor sharing systems. Queueing systems 66(2), 107–130 (2010)
17. Wang, L., Min, G., Kouvatsos, D.D., Jin, X.: Analytical modeling of an integrated priority and WFQ scheduling scheme in multi-service networks. Computer Communications 33, S93–S101 (2010)

The Virtual Waiting Time in a Finite-Buffer Queue with a Single Vacation Policy

Wojciech M. Kempa

Silesian University of Technology,
Institute of Mathematics,
ul. Kaszubska 23, 44-100 Gliwice, Poland
wojciech.kempa@polsl.pl

Abstract. A finite-buffer queueing system with Poisson arrivals and generally distributed service times is considered. Every time when the system empties, a single vacation is initialized, during which the service process is blocked. A system of integral equations for the transient distributions of the virtual waiting time $v(t)$ at a fixed moment t, conditioned by the numbers of packets present in the system at the opening, is derived. A compact formula for the 2-fold Laplace transform of the conditional distribution of $v(t)$ is found and written down using a special-type sequence called a potential. From this representation the stationary distribution of $v(t)$ as $t \to \infty$ and its mean can be easily obtained. Theoretical results are illustrated by numerical examples as well.

Keywords: Finite-buffer queue, Poisson arrivals, Stationary state, Transient state, Virtual waiting time.

1 Introduction

Finite-buffer queueing systems are intensively investigated due to their many applications in the performance evaluation of telecommunication and computer networks. Finite systems with server vacations can be applied in SVC (switched virtual connection) networks where the vacation period can be considered as a time needed for the server to release an SVC or the time for setting up any next SVC (see [14]). Unfortunately, the analysis of such systems is often restricted to the stationary state. However, in practice applications, the investigation of stochastic system characteristics in the transient state becomes more and more desired. Indeed, e.g. permanently changing parameters of IP traffic in telecommunications routers cause that the stationary state of the modelling queueing system, in mathematical sense, quite frequently in practice does not exist.

In the article we study the $M/G/1/N$ system with a single vacation policy and exhaustive service. Every time when the system becomes empty (including the case when the system starts working being empty), a single vacation time of random duration begins. During the vacation the service process is stopped and

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 47–60, 2012.

all packets occurring in this period are accumulated in the buffer queue or, in the case of buffer saturation, are lost. After the vacation the service of packets begins immediately.

In the paper the distribution function of the virtual waiting time $v(t)$, i.e. the time needed to start the service process of the packet that joins the system exactly at time t, is analyzed. Applying the formula of total probability with respect to the first departure epoch after the opening, a system of integral equations for the distributions of the transient virtual waiting time $v(t)$ conditioned by different initial states (numbers of packets) of the system, is build. Next, a specific-type system of equations for the double transforms of conditioned distributions of $v(t)$ is obtained and solved. The solution is written down explicitly by means of a special-type sequence, called a potential, connected with "input" parameters of the system. The main formula can be easily treated numerically. In particular, sample numerical results for the stationary virtual waiting time distribution and its mean are attached.

The review of results for stationary systems with server vacations can be found in monograph [16]. In [6] a general-type system with batch arrivals and exponentially distributed single vacations is considered on the first vacation cycle, using the technique of Wiener-Hopf factorization and integral equations. New results for transient characteristics of the $M^X/G/1$ system with single vacations can be found in [7], [8] and [9]. In particular, in [7] the explicit representation for the 2-fold transform of the departure process is obtained.

Finite systems with server vacations are studied e.g. in [3], [4], [14], [15] and [17]. In particular, in [3] the formula for the mean of the waiting time is derived but only in the steady state of the system. The Laplace-Stieltjes transform of the stationary waiting time in the system with Markovian arrival process is found in [14]. The generalization of this result for the case of BMAP is given in [15] where, additionally, two service disciplines are considered. In [17] the formula for the stationary waiting time distribution in the system with single Poisson arrivals and multiple vacations is derived.

Transient results for finite-buffer models are significantly less frequent in the literature. Stationary results can be found in [16]. In [11] the representation for the Laplace transform of the probability generating function of departure counting process in the $M^X/G/1/N$ queue with batch arrivals is obtained. The formulae for the transient virtual waiting time in a finite system can be found in monograph [2] where the cases of single Poisson, $MMPP$ and $BMAP$ input flows are considered.

The transient waiting time distribution in the general-type queueing system with batch arrivals and infinite buffer is investigated in [5] (virtual waiting time) and [10] (actual waiting time), using the mixed technique based on supplementary variables' approach, Volterra-type integral equations and Wiener-Hopf factorization method.

The article is organized as follows. In the next Section 2 we give a mathematical description of the system and state some auxiliary results. In Section 3 we

build a system of equations for the 2-fold Laplace transform of the conditional virtual waiting time distributions. In Section 4 we state in the explicit form, as the main result, the solution of the system introduced in Section 3. Section 5 is devoted to numerical computations: for sample queueing systems, applying the result from the previous section, we obtain the stationary distributions of the virtual waiting time and their means using, one of algorithms for the numerical Laplace transform inversion. The last Section 6 contains conclusions.

2 The Queueing System and Auxiliary Results

In the article we study a queueing system of the $M/G/1/N$ type with single vacations. The total system capacity is assumed to be equal N i.e. there are $N-1$ places in the buffer queue and one place in service. Let us denote by λ the intensity of the Poisson arrival process and by $F(\cdot)$ a general d.f. (distribution function) of the service time. Assume that single vacation durations are generally distributed with a d.f. $G(\cdot)$. Standardly, it is assumed that all interarrival times, service times and sucessive single vacation durations are totally independent.

Introduce the Laplace-Stieltjes transforms of d.fs $F(\cdot)$ and $G(\cdot)$ as follows:

$$f(s) = \int_0^\infty e^{-st} dF(t), \qquad g(s) = \int_0^\infty e^{-st} dG(t), \quad \mathrm{Re}(s) > 0. \tag{1}$$

Besides, let $F^{j*}(\cdot)$ denotes the j-fold Laplace-Stieltjes convolution of the d.f. $F(\cdot)$ with itself i.e.

$$F^{0*}(t) = 1, \quad F^{1*}(t) = F(t), \quad F^{j*}(t) = \int_0^t F^{(j-1)*}(t-y) dF(y), \tag{2}$$

where $t > 0$.

Finally, let $X(t)$ denotes the number of packets present in the system at time t.

In [12] (see also [2], [13]) the following system of equations is considered:

$$\sum_{k=-1}^{n} a_{k+1} x_{n-k} - x_n = \psi_n, \quad n \geq 0, \tag{3}$$

where sequences $(a_n)_{n=0}^\infty$ $(a_0 \neq 0)$ and $(\psi_n)_{n=0}^\infty$ are known, and the sequence $(x_n)_{n=0}^\infty$ is unknown.

The following theorem (see [2], [12]) states the explicit representation for the nth term of the sequence (x_n) using a known another sequence given in a recurrent way.

Theorem 1. *The solution of the system (3) can be written in the following form:*

$$x_n = C R_{n+1} + \sum_{k=0}^{n} R_{n-k} \psi_k, \quad n \geq 0, \tag{4}$$

where C is a constant independent on n and the sequence $(R_n)_{n=0}^{\infty}$, called a potential, is defined recursively as follows:

$$R_0 = 0, \quad R_1 = a_0^{-1}, \quad R_{n+1} = R_1(R_n - \sum_{k=0}^{n} a_{k+1} R_{n-k}), \quad n \geq 1. \tag{5}$$

As it turns out later, the system of equations for the 2-fold transforms of conditional virtual waiting time distributions can be transformed to the form of (3), and hence it can be solved using the result (4) from Theorem 1.

3 Equations for the 2-Fold Transform of the Virtual Waiting Time Distribution

Let us introduce the following notation:

$$V_n(t,x) = \mathbf{P}\{v(t) < x \,|\, X(0) = n\}, \quad x > 0,\, t > 0,\, 0 \leq n \leq N. \tag{6}$$

Thus, $V_n(t,x)$ is a distribution of the virtual waiting time $v(t)$ conditioned by the number of packets present in the system at the opening.

Note that if the system is empty at time $t = 0$, then the formula of total probability leads to the following equation:

$$\begin{aligned} V_0(t,x) &= \int_0^t \Big[\sum_{k=1}^{N-1} \frac{(\lambda u)^k}{k!} e^{-\lambda u} V_k(t-u,x) + V_N(t-u,x) \sum_{k=N}^{\infty} \frac{(\lambda u)^k}{k!} e^{-\lambda u}\Big] dG(u) \\ &+ \lambda \int_0^t dG(u) \int_u^t e^{-\lambda y} V_1(t-y,x)dy + e^{-\lambda t} G(t) \\ &+ \sum_{k=1}^{N-1} \frac{(\lambda t)^k}{k!} e^{-\lambda t} \int_t^{\infty} F^{k*}(x-u+t) dG(u) \\ &+ \big(1 - G(t)\big) \sum_{k=N}^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} + e^{-\lambda t}\big(G(t+x) - G(t)\big). \end{aligned} \tag{7}$$

Let us briefly comment the last equation. Indeed, the first summand on the right side of (7) presents the situation in that the first single vacation ends before t and during it at least one arrival occurs. Because of the finite system capacity (equal to N), in the case of $k \geq N$ jumps of the Poisson process (describing the input stream of packets) during the vacation, the number of packets at the end of the vacation equals N ($k - N$ packets are lost).

In the second summand on the right side of (7) the vacation also ends before t but the first arrival occurs after the vacation completion epoch. In particular, if the first packet enters after t, then $v(t) = 0$.

The third summand on the right side of (7) describes the case in that the vacation ends after time t but there are some arrivals before t. Then the random event $\{v(t) < x\}$ is equivalent to the event {service time of k packets plus $(u-t) < x\}$, where k denotes the number of packets occurring before t, and u stands for the completion epoch of the single vacation. Of course

$$\mathbf{P}\{\text{service time of } k \text{ packets plus } (u-t) < x\} = F^{k*}(x-u+t).$$

In the fourth summand in (7) the vacation ends after t but the "virtual" packet arriving exactly at time t is lost since the buffer is saturated at this time (at least N jumps of the Poisson arrival process occur before t). Hence $\{v(t) < x\}$ with probability one.

Finally, in the last summand on the right side of (7) the vacation ends after t and, besides, there are no arrivals before t.

As it is well known (see e.g. [1]), the service completion epochs are renewal moments in the single-server queueing system with Poisson arrivals. Applying the formula of total probability with respect to the first service completion epoch after the opening of the system, we obtain the following system of integral equations for the conditional distributions $V_n(t,x)$, where $1 \le n \le N$:

$$\begin{aligned} V_n(t,x) = \int_0^t \Bigg[& \sum_{k=0}^{N-n-1} \frac{(\lambda y)^k}{k!} e^{-\lambda y} V_{n+k-1}(t-y,x) \\ & + V_{N-1}(t-y,x) \sum_{k=N-n}^{\infty} \frac{(\lambda y)^k}{k!} e^{-\lambda y} \Bigg] dF(y) \\ & + \int_t^{\infty} \Bigg[\sum_{k=0}^{N-n-1} \frac{(\lambda t)^k}{k!} e^{-\lambda t} F^{(n+k-1)*}(x-y+t) \\ & + \sum_{k=N-n}^{\infty} \frac{(\lambda t)^k}{k!} e^{-\lambda t} F^{(N-1)*}(x-y+t) \Bigg] dF(y), \end{aligned} \tag{8}$$

The interpretation of the right side of (8) is similar to (7): the first integral on the right side of (8) relates to the situation in that the first service ends before time t, and the second one - to the case of no service completion epochs occurring before t.

Let us introduce into the system of equations (7)–(8) the double Laplace transform of $V_n(t,x)$ (on the arguments t and x) defined as follows:

$$\widehat{V}_n(s,z) = \int_0^{\infty} e^{-st} dt \int_0^{\infty} e^{-zx} V_n(t,x) dx, \quad \mathrm{Re}(s) > 0, \mathrm{Re}(z) > 0, n \ge 0. \tag{9}$$

Next, let us define the following sequences:

$$a_k(s) = \int_0^\infty e^{-(s+\lambda)t} \frac{(\lambda t)^k}{k!} dF(t), \tag{10}$$

$$b_k(s,z) = \int_0^\infty e^{-zx} dx \int_0^\infty e^{-(s+\lambda)t} \frac{(\lambda t)^k}{k!} dt \int_t^\infty F^{k*}(x-u+t) dG(u), \tag{11}$$

$$c_k(s,z) = z^{-1} \int_0^\infty e^{-(s+\lambda)t} \frac{(\lambda t)^k}{k!} \big(1 - G(t)\big) dt, \tag{12}$$

$$d(s,z) = \int_0^\infty e^{-zx} dx \int_0^\infty e^{-(s+\lambda)t} G(t+x) dt, \tag{13}$$

$$h_{k,j}(s,z) = \int_0^\infty e^{-zx} dx \int_0^\infty e^{-(s+\lambda)t} \frac{(\lambda t)^k}{k!} dt \int_t^\infty F^{j*}(x-y+t) dF(y), \tag{14}$$

$$q_k(s,z) = \sum_{j=0}^{N-k-1} h_{j,k+j-1}(s,z) + \sum_{j=N-k}^{\infty} h_{j,N-1}(s,z), \tag{15}$$

$$r_k(s) = \int_0^\infty e^{-(s+\lambda)t} \frac{(\lambda t)^k}{k!} dG(t). \tag{16}$$

Let us note that the following representations hold true:

$$\int_0^\infty e^{-zx} dx \int_0^\infty e^{-st} dt \int_0^t \frac{(\lambda y)^k}{k!} e^{-\lambda y} V_k(t-y,x) dF(y) = a_k(s) \widehat{V}_k(s,z),$$

$$\int_0^\infty e^{-zx} dx \int_0^\infty e^{-st} dt \int_0^t \frac{(\lambda y)^k}{k!} e^{-\lambda y} V_k(t-y,x) dG(y) = r_k(s) \widehat{V}_k(s,z),$$

$$\lambda \int_0^\infty e^{-zx} dx \int_0^\infty e^{-st} dt \int_0^t dG(u) \int_u^t e^{-\lambda y} V_1(t-y,x) dy = \frac{\lambda}{\lambda+s} g(\lambda+s) \widehat{V}_1(s,z).$$

Taking into consideration the above formulae and notations (10)–(16), the system of equations (7)–(8) can be transformed to the following form:

$$\widehat{V}_0(s,z) = \sum_{k=1}^{N-1} r_k(s) \widehat{V}_k(s,z) + \widehat{V}_N(s,z) \sum_{k=N}^{\infty} r_k(s)$$
$$+ \frac{\lambda}{\lambda+s} g(\lambda+s) \widehat{V}_1(s,z) + \sum_{k=1}^{N-1} b_k(s,z) + \sum_{k=N}^{\infty} c_k(s,z) + d(s,z) \tag{17}$$

and

$$\widehat{V}_n(s,z) = \sum_{k=0}^{N-n-1} a_k(s) \widehat{V}_{n+k-1}(s,z) + \widehat{V}_{N-1}(s,z) \sum_{k=N-n}^{\infty} a_k(s) + q_n(s,z), \tag{18}$$

where $1 \le n \le N$.

Substituting now

$$T_n(s,z) = \widehat{V}_{N-n}(s,z), \quad 0 \le n \le N, \tag{19}$$

we can rewrite equations (17)–(18) as

$$T_N(s,z) = \sum_{k=1}^{N-1} r_k(s) T_{N-k}(s,z) + T_0(s,z) \sum_{k=N}^{\infty} r_k(s)$$
$$+ \frac{\lambda}{\lambda+s} g(\lambda+s) T_{N-1}(s,z) + \sum_{k=1}^{N-1} b_k(s,z) + \sum_{k=N}^{\infty} c_k(s,z) + d(s,z) \tag{20}$$

and

$$\sum_{k=-1}^{n} a_{k+1}(s) T_{n-k}(s,z) - T_n(s,z) = \varphi_n(s,z), \tag{21}$$

where $0 \le n \le N-1$, and

$$\varphi_n(s,z) = a_{n+1}(s) T_0(s,z) - T_1(s,z) \sum_{k=n+1}^{\infty} a_k(s) - q_{N-n}(s,z). \tag{22}$$

Let us note that the system (21) has the same form as (3) with unknown functions $T_n(s,z)$, and $a_n(s)$ and $\varphi_n(s,z)$ playing roles of a_k and ψ_n respectively. Of course, since now terms of the sequence (a_n) depend on the argument s, then the potential (R_n) will be dependent on s too. Similarly, since T_n is a function of s and z, then the constant C, in general, will be a function of s and z too.

4 Main Result

From the representation (4) in Theorem 1, adjusting to the system (21), follows that the solution of (21) can be written in the following form:

$$T_n(s,z) = C(s,z) R_{n+1}(s) + \sum_{k=0}^{n} R_{n-k}(s) \varphi_k(s,z), \quad n \ge 0, \tag{23}$$

where the constant $C(s,z)$ is independent on n and the sequence $\big(R_n(s)\big)_{n=0}^{\infty}$ (the potential connected with the sequence $\big(a_n(s)\big)_{n=0}^{\infty}$) is defined as follows (compare to (5)):

$$R_0(s) = 0, \quad R_1(s) = a_0^{-1}(s),$$
$$R_{n+1}(s) = R_1(s)\big(R_n(s) - \sum_{k=0}^{n} a_{k+1}(s) R_{n-k}(s)\big), \quad n \ge 1. \tag{24}$$

To state the formula for $T_n(s,z)$ explicitly, we must find representations for $C(s,z)$, and $T_0(s,z)$ and $T_1(s,z)$ occurring in (22).

Substituting $n = 0$ into the formula (23) we obtain

$$C(s,z) = \frac{T_0(s,z)}{R_1(s)} = a_0(s) T_0(s,z). \tag{25}$$

Similarly, substituting $n = 0$ into the formula (21) we get

$$a_0(s)T_1(s,z) + a_1(s)T_0(s,z) - T_0(s,z) = \varphi_0(s,z) \tag{26}$$

and hence, since $\sum_{k=0}^{\infty} a_k(s) = f(s)$, we obtain

$$T_1(s,z) = \frac{T_0(s,z) - q_N(s,z)}{f(s)}. \tag{27}$$

Now let us substitute (25) and (27) into the formula (23). We obtain (compare [2])

$$\begin{aligned}
T_n(s,z) &= a_0(s)R_{n+1}(s)T_0(s,z) + \sum_{k=0}^{n} R_{n-k}(s) \\
&\times \Big(a_{k+1}(s)T_0(s,z) - T_1(s,z)\sum_{i=k+1}^{\infty} a_i(s) - q_{N-k}(s,z)\Big) \\
&= T_0(s,z)\Big[a_0(s)R_{n+1}(s) + \sum_{k=0}^{n} R_{n-k}(s)\Big(a_{k+1}(s) - \frac{1}{f(s)}\sum_{i=k+1}^{\infty} a_i(s)\Big)\Big] \\
&+ \sum_{k=0}^{n} R_{n-k}(s)\Big(\frac{q_N(s,z)}{f(s)}\sum_{i=k+1}^{\infty} a_i(s) - q_{N-k}(s,z)\Big) \\
&= T_0(s,z)\Theta_n(s) + \Phi_n(s,z),
\end{aligned} \tag{28}$$

where

$$\Theta_n(s) = a_0(s)R_{n+1}(s) + \sum_{k=0}^{n} R_{n-k}(s)\Big(a_{k+1}(s) - \frac{1}{f(s)}\sum_{i=k+1}^{\infty} a_i(s)\Big), \tag{29}$$

$$\Phi_n(s,z) = \sum_{k=0}^{n} R_{n-k}(s)\Big[\frac{q_N(s,z)}{f(s)}\sum_{i=k+1}^{\infty} a_i(s) - q_{N-k}(s,z)\Big]. \tag{30}$$

Applying the formula (28) in (20) we can eliminate $T_0(s,z)$. Indeed, we have

$$\begin{aligned}
&T_0(s,z)\Theta_N(s) + \Phi_N(s,z) \\
&= \sum_{k=1}^{N-1} r_k(s)\Big(T_0(s,z)\Theta_{N-k}(s) + \Phi_{N-k}(s,z)\Big) + T_0(s,z)\sum_{k=N}^{\infty} r_k(s) \\
&+ \frac{\lambda}{\lambda+s} g(\lambda+s)\Big(T_0(s,z)\Theta_{N-1}(s) + \Phi_{N-1}(s,z)\Big) \\
&+ \sum_{k=1}^{N-1} b_k(s,z) + \sum_{k=N}^{\infty} c_k(s,z) + d(s,z)
\end{aligned} \tag{31}$$

and hence we obtain

$$T_0(s,z) = \frac{\sum_{k=1}^{N-1} b_k(s,z) + \sum_{k=N}^{\infty} c_k(s,z) + d(s,z) - D_\Phi(s,z)}{D_\Theta(s) - \sum_{k=N}^{\infty} r_k(s)}, \tag{32}$$

where

$$D_{\Phi}(s,z) = \Phi_N(s,z) - \lambda(\lambda+s)^{-1} g(\lambda+s)\Phi_{N-1}(s,z) - \sum_{k=1}^{N-1} r_k(s)\Phi_{N-k}(s,z), \tag{33}$$

and

$$D_{\Theta}(s) = \Theta_N(s) - \lambda(\lambda+s)^{-1} g(\lambda+s)\Theta_{N-1}(s) - \sum_{k=1}^{N-1} r_k(s)\Theta_{N-k}(s). \tag{34}$$

Now the formulae (19), (28), (29), (30), (32), (33) and (34) lead to the following main theorem:

Theorem 2. *The 2-fold Laplace transform of the conditional virtual waiting time distribution in the $M/G/1/N$-type queueing system with single vacations can be written in the following form:*

$$\begin{aligned}\widehat{V}_n(s,z) &= \int_0^{\infty} e^{-zx}dx \int_0^{\infty} e^{-st}\mathbf{P}\{v(t) < x \mid X(0) = n\}dt \\ &= \frac{\sum_{k=1}^{N-1} b_k(s,z) + \sum_{k=N}^{\infty} c_k(s,z) + d(s,z) - D_{\Phi}(s,z)}{D_{\Theta}(s) - \sum_{k=N}^{\infty} r_k(s)}\Theta_{N-n}(s) + \Phi_{N-n}(s,z),\end{aligned} \tag{35}$$

where $0 \le n \le N$, and the formulae for $b_k(s,z)$, $c_k(s,z)$, $d(s,z)$, $r_k(s)$, $\Theta_n(s)$, $\Phi_n(s,z)$, $D_{\Phi}(s,z)$ and $D_{\Theta}(s)$ are given in (11), (12), (13), (16), (29), (30), (33) and (34) respectively.

Of course, in the case of $G(t) \equiv 1$, $t > 0$, when we get a classical $M/G/1/N$-type system (formally we have a single vacation of zero duration), the formula (35) also holds true.

Since in the last Theorem the double transform $\widehat{V}_n(s,z)$ of the virtual waiting time distribution is given explicitly, then the formula (35) can be applied in practice in different ways.

Firstly, the representation for the Laplace transform of the stationary waiting time distribution (as $t \to \infty$) can be found, using the Tauberian theorem as follows:

$$\int_0^{\infty} e^{-zx}\mathbf{P}\{v(\infty) < x\}dx = \lim_{s \downarrow 0} s\widehat{V}_n(s,z). \tag{36}$$

Similarly, the mean $\mathbf{E}v$ of the stationary waiting time v can be obtained in the following way:

$$\mathbf{E}v = -\frac{d}{dz}\Big(z \lim_{s \downarrow 0} s\widehat{V}_n(s,z)\Big)\Big|_{z=0}, \tag{37}$$

where n in the formulae (36) and (37) can be chosen arbitrarily, since the stationary waiting time distribution is independent on the initial state of the system.

5 Numerical Examples

In this section we present numerical examples in which we derive, using the formulae (35), (36) and (37), the stationary waiting time distribution $V(x) = \mathbf{P}\{v(\infty) < x\}$ and its mean $\mathbf{E}v = \int_0^\infty x dV(x)$ for sample queueing systems. In all computations we use the *Mathematica* environment.

(1) Let us take into cosideration the system in that $N = 5$, $\lambda = 4$, $F(x) = 1 - e^{-2x}$, $x > 0$ (exponentially distributed service time with mean 0.5) and $G(x) = 1 - e^{-x}(1 + x)$, $x > 0$ (2-Erlang distribution with parameter 1 of the single vacation duration).

For such a system, using the formula (35) and next (36), we obtain

$$\int_0^\infty e^{-zx}\mathbf{P}\{v(\infty) < x\}dx = \frac{11412320 + 27777552z + 23664336z^2 + 10049640z^3 + 364710z^4}{356635z(1+z)^2(2+z)^5} + \frac{+964193z^5 + 161956z^6 + 12413z^7}{356635z(1+z)^2(2+z)^5}. \tag{38}$$

Inverting (38) on the argument z we get

$$V(x) = \frac{1}{1069905}\Big(1069905 - 12e^{-x}(-10754 + 9981x) - 2e^{-2x}(580857 + 1018404x + 928392x^2 + 531680x^3 + 178592x^4)\Big) \tag{39}$$

and hence $\mathbf{E}v = 2.066$.

The stationary waiting time distribution (39) is presented in Figure 1.

(2) Assume now that $N = 4$, $\lambda = 8$, service times are exponentially distributed with mean 1 and $G(x)$ is the same as in the case **(1)**.

Using the formulae (35) and (36) we find

$$\int_0^\infty e^{-zx}\mathbf{P}\{v(\infty) < x\}dx = \frac{315004145 + 362270293z + 57365618z^2 + 13427330z^3}{315004145z(1+z)^5} + \frac{+4086349z^4 + 721625z^5}{315004145z(1+z)^5}. \tag{40}$$

Inverting (40) on the argument z we obtain

$$V(x) = \frac{1}{315004145}\Big(315004145 - 24e^{-x}(13095105 + 13075179x + 6448044x^2 + 1910560x^3 + 64x^4)\Big). \tag{41}$$

Hence (see (37)) we get $\mathbf{E}v = 3.84995$.

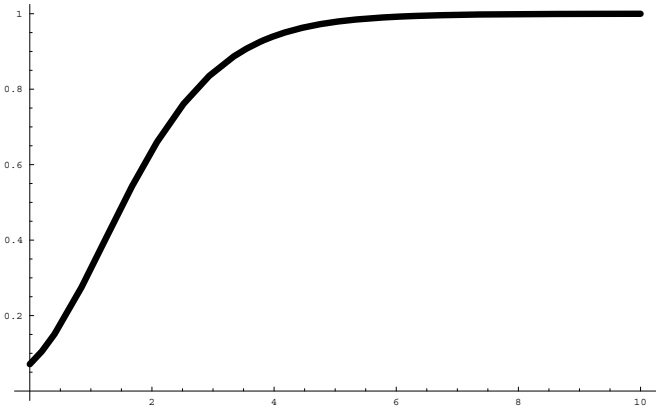


Fig. 1. Stationary waiting time distribution in the case **(1)**

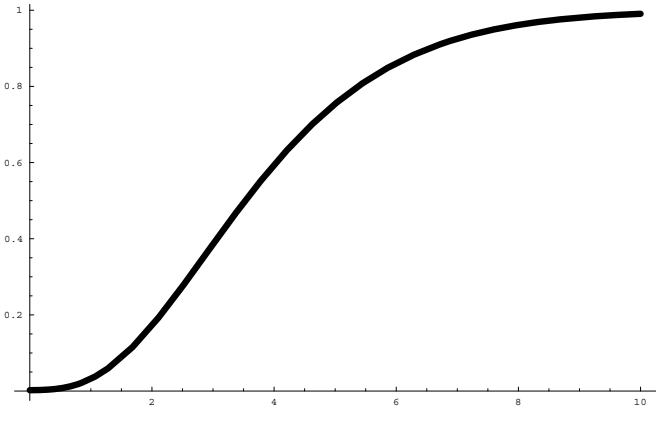


Fig. 2. Stationary waiting time distribution in the case **(2)**

The stationary distribution (41) is presented in Figure 2.

(3) Consider now the system described by exponential distributions, with $\lambda = 3$ (the mean of interarrival time equal to 0.333), and the means of the service time and the vacation duration equal to 1. In Table 1 we present the values of $\mathbf{E}v$, computed using the formula (37), as a function of increasing system capacity N (for $N = 1$ to 10).

Table 1. Mean of the stationary waiting time $\mathbf{E}v$ as a function of the system capacity N

System capacity N	Mean of the stationary waiting time $\mathbf{E}v$
1	0
2	1.54982
3	2.53652
4	3.52184
5	4.51110
6	5.50508
7	6.50217
8	7.50088
9	8.50035
10	9.50013

As one can note, the mean of the time required for a packet to initialize its service increases as the buffer-queue capacity increases. Indeed, intuitively, in the case of small buffer capacity the probability of packet loss increases and, in consequence, the mean of the waiting time is low, since for all lost packets the waiting time equals 0. As the buffer capacity increases, the loss probability of a packet is smaller and smaller, and $\mathbf{E}v$ increases.

(4) Now, let us take into consideration the system of size $N = 3$, and exponentially distributed service times and vacation durations with means 1. In Table 2 we present the mean of the stationary waiting time $\mathbf{E}v$ as a function of different values of λ.

Table 2. Mean of the stationary waiting time $\mathbf{E}v$ as a function of the intensity of Poisson arrivals λ

Intensity of Poisson arrivals λ	Mean of the stationary waiting time $\mathbf{E}v$
0.2	0
1	1.75610
2	2.31380
3	2.53652
5	2.72803
15	2.92002
40	2.97276
80	2.98691
150	2.99316
300	2.99662

Obviously, as one can observe, as the intensity of packet arrivals λ increases, then the mean time of waiting in the queue $\mathbf{E}v$ increases too, but for large λ's stabilizes. This observation has the following intuitive explanation. For relatively small intensity of arrivals, $\mathbf{E}v$ increases as the probability of the buffer overflow decreases. In the case of large values of λ a lot of packets are lost due to the buffer

congestion. Then, in practice, $v(t)$ is positive only for such t for which the system contains $N-1$ packets. Since for large λ's the differences between the waiting times for arrivals are very small, hence the values $\mathbf{E}v$ are close to each other.

(5) Lastly, let us examine the influence of the single vacation duration on the mean of the waiting time in the stationary state of the system. Let us consider the system in which $N=3$, $\lambda=4$ and service times are exponentially distributed with mean 1. Let us assume that durations of successive vacations have exponential distributions with mean γ^{-1}. In Table 3 we present the mean of the stationary waiting time $\mathbf{E}v$ as a function of different values of γ^{-1}.

Table 3. Mean of the stationary waiting time $\mathbf{E}v$ as a function of the mean of vacation duration

Mean of the vacation duration γ^{-1}	Mean of the stationary waiting time $\mathbf{E}v$
0.333	2.68260
0.5	2.67806
1	2.65507
1.25	2.64133
5	2.43077
20	1.89790
66.667	1.32699
1000	0.80270
10000	0.75540
1000000	0

As one can observe, in the considered example, the longer duration of the vacation period, the shorter the mean waiting time. It is intuitively clear: if the service is blocked for a very long time (compared to the intensity of arrivals), then the number of packets lost due to the buffer overflow is very large and, hence, the value of $\mathbf{E}v$ decreases (since for the lost packets the mean of the waiting time equals 0).

6 Conclusions

In the paper the $M/G/1/N$-type queueing system with single vacations and exhaustive service is considered. A system of integral equations for the distributions of the transient virtual waiting time $v(t)$, conditioned by the number of packets present in the system at the opening, is built. The solution of the corresponding system obtained for double transforms of distributions of $v(t)$ is found applying the potential method. The final formula is written down using the sequence, called the potential, defined recursively by means of "input" parameters of the system. The representation is convenient for numerical treatment. Some examples are attached, in which the influence of the Poisson arrival rate, the buffer size and the vacation duration for the mean of the stationary waiting time is examined.

References

1. Cohen, J.W.: The single server queue. North-Holland Publishing Company, Amsterdam (1982)
2. Chydzinski, A.: Queueing characteristics for Markovian traffic models in packet-oriented networks. Silesian University of Technology Press, Gliwice (2007) (in Polish)
3. Gupta, U.C., Banik, A.D., Pathak, S.S.: Complete analysis of $MAP/G/1/N$ queue with single (multiple) vacation(s) under limited service discipline. Journal of Applied Mathematics and Stochastic Analysis 3, 353–373 (2005)
4. Gupta, U.C., Sikdar, K.: Computing queue length distributions in $MAP/G/1/N$ queue under single and multiple vacation. Appl. Math. Comput. 174(2), 1498–1525 (2006)
5. Kempa, W.M.: The virtual waiting time for the batch arrival queueing systems. Stoch. Anal. Appl. 22(5), 1235–1255 (2004)
6. Kempa, W.M.: $GI/G/1/\infty$ batch arrival queueing system with a single exponential vacation. Math. Method. Oper. Res. 69(1), 81–97 (2009)
7. Kempa, W.M.: Some new results for departure process in the $M^X/G/1$ queueing system with a single vacation and exhaustive service, Stoch. Anal. Appl. 28(1), 26–43 (2009)
8. Kempa, W.M.: Characteristics of vacation cycle in the batch arrival queueing system with single vacations and exhaustive service. Int. J. Appl. Math. 23(4), 747–758 (2010)
9. Kempa, W.M.: On departure process in the batch arrival queue with single vacation and setup time. Annales UMCS, Informatica 10(1), 93–102 (2010)
10. Kempa, W.M.: Some results for the actual waiting time in batch arrival queueing systems. Stoch. Models 26(3), 335–356 (2010)
11. Kempa, W.M.: Departure Process in Finite-Buffer Queue with Batch Arrivals. In: Al-Begain, K., Balsamo, S., Fiems, D., Marin, A. (eds.) ASMTA 2011. LNCS, vol. 6751, pp. 1–13. Springer, Heidelberg (2011)
12. Korolyuk, V.S.: Boundary-value problems for complicated Poisson processes. Naukova Dumka, Kiev (1975) (in Russian)
13. Korolyuk, V.S., Bratiichuk, M.S., Pirdzhanov, B.: Boundary-value problems for random walks. Ylym, Ashkhabad (1987) (in Russian)
14. Niu, Z., Takahashi, Y.: A finite-capacity queue with exhaustive vacation/close-down/setup times and Markovian arrival processes. Queueing Syst. 31, 1–23 (1999)
15. Niu, Z., Shu, T., Takahashi, Y.: A vacation queue with setup and close-down times and batch Markovian arrival processes. Perform. Evaluation 54(3), 225–248 (2003)
16. Takagi, H.: Queueing Analysis, vol. 1: Vacation and Priority Systems, vol. 2. Finite Systems. North-Holland, Amsterdam (1993)
17. Takagi, H.: $M/G/1/N$ queues with server vacations and exhaustive service. Oper. Res. 42(5), 926–939 (1994)

Analysis of Periodically Gated Vacation Model and Its Application to IEEE 802.16 Network

Zsolt Saffer[1], Sergey Andreev[2], and Yevgeni Koucheryavy[2]

[1] Budapest University of Technology and Economics (BUTE), Hungary
safferzs@hit.bme.hu
[2] Tampere University of Technology (TUT), Finland
sergey.andreev@tut.fi,
yk@cs.tut.fi

Abstract. In this paper we consider the analysis of an $M/D/1$ vacation queue with periodically gated discipline. The motivation of introducing the new periodically gated discipline comes from the stochastic behavior of a kind of contention-based bandwidth reservation mechanism applied in wireless networks. The analysis utilizes a former result from polling model, which expresses the number of customers at arbitrary epoch in terms of the number of customers at start and end of vacations. The mean and the probability-generating function of the number of customers at arbitrary epoch are determined.

In the last part of the paper we demonstrate the application of the model to the non real-time uplink traffic in IEEE 802.16-based wireless broadband networks.

Keywords: queueing theory, vacation model, Markov chain, contention-based bandwidth reservation, IEEE 802.16.

1 Introduction

Vacation model is an extension of the basic queue, in which the single server takes vacation occasionally. For details on classical vacation models and their solution the reader is referred to the survey of Doshi [4] and to the book of Takagi [10].

Vacation models are widely used queueing theoretical tools. This is mainly due to their generality. On the other hand, vacation models can be tailored to the needs of the application under investigation, which facilitates the applicability of these models. Such examples are multi-server vacation models (see e.g. [11], [13] or [8]) or working vacation models (see e.g. [9], [12] or [7]).

In this paper we introduce a new service discipline for the M/D/1 vacation model. The *periodically gated* discipline is an extension of the gated discipline. The way of the customers in the gated system can be modeled with the help of a gate. The arriving customers are accumulated in a buffer A behind a closed gate. At the end of the vacation the gate is opened and the accumulated customers move to the buffer B, from which they will be served. Then the gate is closed

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 61–75, 2012.

immediately and the service of the customers in the buffer B starts. Thus, in the gated system, only those customers are served in the actual service period which are present at start of that service period. Under the periodically gated discipline the gate may be opened several times also during the service period. This allows moving of newly accumulated customers from behind the gate into the buffer B with stochastic periodicity. When buffer B becomes empty the server goes to vacation. The vacation ends at natural epoch when the gate opens next time according to its prescribed stochastic periodicity. Hence in this model the vacation period depends on the stochastic period, which itself depends on the arrival process.

The motivation of introducing the new periodically gated vacation model lies in the stochastic behavior of a kind of contention-based bandwidth reservation mechanism applied in wireless networks. As part of this kind of reservation mechanism a collision resolution process is invoked. The newly arriving packets (corresponding to customers) must wait during the collision resolution in a reservation buffer (corresponding to buffer A), from which they move to the scheduling buffer (corresponding to buffer B) when the collision resolution process ends successfully. Thus packets move from reservation buffer to scheduling buffer with stochastic periodicity also during scheduling the packets (corresponding to service).

Due to the assumption of constant probability of successful collision resolution (see the independent conditional collision probability assumption proposed by Bianchi [2]) the length of each of these stochastic periods can be modeled as a sum of the time until first arrival during it and a shifted geometrically distributed random variable, whose parameter is the above constant probability. Thus the length of such stochastic period depends also on the arrival process during it. The time interval during which the scheduling buffer (buffer B) is empty corresponds to the vacation period. The vacation period ends next time when packets come from the reservation buffer to the scheduling buffer (next gate opening).

The main contribution of this paper is the analysis of the periodically gated vacation model. In the first step a Markov chain embedded at the gate opening epochs is applied to determine the steady-state probabilities of the number of customers at that epochs. Then the probability-generating function (PGF) and the mean number of customers are determined at both start and end of vacation. Based on them the PGF and the mean of the number of customers at arbitrary epoch are determined, for which the work-conserving property of the model and a former result for polling models ([3], [6]) are utilized. In the last part of the paper we demonstrate the application of the model to the non real-time uplink traffic in IEEE 802.16-based wireless broadband networks ([5]) by establishing the formula for determining the mean packet delay. A detailed evaluation of the application of the periodically gated vacation model to the above network is a topic of future work.

The rest of this paper is organized as follows. In section 2 the model and the notations are introduced and explained. The analysis at characteristic epochs

of the model is presented in section 3. In section 4 the PGF and the mean of the number of customers at arbitrary epoch are derived. Finally the application of the model to IEEE 802.16-based wireless broadband networks is described in section 5.

2 Model Description

We consider a queue with vacations and periodically gated service discipline. Customers arrive to the system according to the Poisson process with rate λ. The infinite length queue is divided into two buffer parts separated by a gate. The customers arrive to buffer A behind the closed gate. Eventually the gate opens, which causes moving of all customers from buffer A into buffer B. Afterwards the gate is closed immediately. Whenever buffer B is not empty its customers are served. In contrast to the gated discipline, in this model the opening of the gate occurs also during the service period, i.e. when buffer B is not empty. The customer service time is constant and denoted by b. We count the time in customer service time units, i.e all the considered time periods in this model are integer multiples of the constant customer service time. Under the periodically gated service discipline the gate opens with stochastic periodicity. The time between two consecutive openings of the gate is the sum of the time until the first customer arrival and the *gate period*. The distribution of the gate period is shifted geometrical with parameter $0 < p \leq 1$ in terms of the constant service time. The gate period also includes the time unit, during which the first customer arrives (see Fig. 1). The epochs of gate openings are called *G-epochs*.

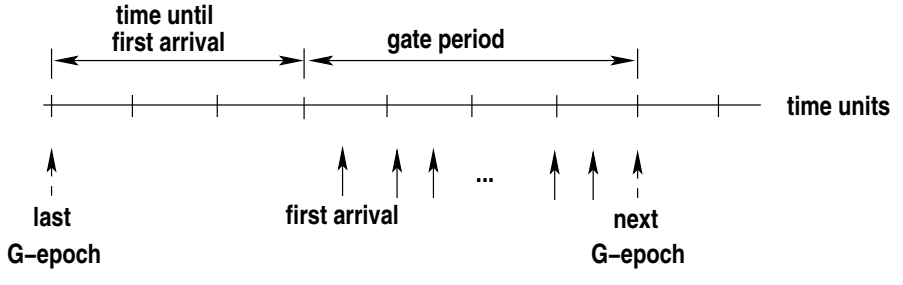


Fig. 1. Gate period in periodically gated vacation model

When buffer B becomes empty the server goes to vacation. The vacation takes until the next G-epoch. Thus the length of vacation period depends on the gate period and it can also depend on the arrival process if buffer A is also empty at start of vacation. In this case the vacation period depends on the arrival process via the time until the first customer arrival during it. The M/D/1 periodically gated vacation model is illustrated in Fig. 2.

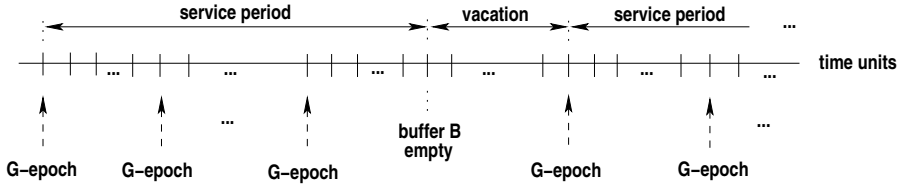


Fig. 2. Periodically gated vacation model

We impose the following assumptions on the periodically gated vacation model:

A.1 The arrival rate and the customer service time are positive and finite, i.e. $0 < \lambda < \infty$, $0 < b < \infty$.

A.2 The arrival process and the customer service times are mutually independent.

A.3 Work-conserving property: If the service begins then it is work conserving up to end of the service according to the periodically gated service discipline.

We assume that the model is stable. Only finite number of customers can be accumulated during the vacation period, since both the arrival rate and the mean gate period are finite. Hence to ensure the stability of the model only the service period must be considered. This model is stable if the arrival rate does not exceed the mean service rate ($\frac{1}{b}$). Thus the condition of the stability can be given as

$$\rho < 1, \tag{1}$$

where $\rho = \lambda b$ is the server utilization.

When $\widehat{y}(z)$ is a PGF, $\widehat{y}'(z)$ denotes its first derivative with respect to z. Furthermore for the PGF $\widehat{y}(z)$, $y^{(k)}$ denotes its k-th derivative at $z = 1$ for $k \geq 1$, i.e., $y^{(k)} = \frac{d^k}{dz^k}\widehat{y}(z)|_{z=1}$. Additionally $[\mathbf{Y}]_{j,l}$ stands for the j,l-th element of matrix $\mathbf{Y}$. Similarly $[\mathbf{y}]_j$ denotes the j-th element of vector $\mathbf{y}$.

3 Analysis at Characteristic Epochs of the System

In this section we analyze the number of customers at G-epochs, at start of vacation and at end of vacation. Throughout this paper the G-epoch stands for the epoch just after the gate openings. Similarly under start and end of vacations we understand the epochs just after start and end of vacations, respectively.

In the following we establish an embedded Markov chain in order to determine the distribution of the number of customers at G-epochs. Then we relate the number of customers at start of vacation to the number of customers at G-epochs. Finally we give a relationship to determine the distribution of the number of customers at end of vacation from the distribution of the number of customers at start of vacation.

3.1 The Number of Customers at G-Epochs

Let $q^{(G)}(\ell)$ be the number of customers in the system at l-th G-epochs for $\ell > 0$. The sequence $\{q^G(\ell), \ell > 0\}$ is a homogenous embedded Markov chain on the state space $(\{1, 2, \ldots\})$. Let $p^{(G)}(j,k)$ denote the probability of transition from state j to state k in this Markov chain, i.e.

$$p^{(G)}(j,k) = \; P\{q^{(G)}(\ell+1) = k \mid q^{(G)}(\ell) = j\}, \quad \ell \geq 1, \quad j,k \geq 1.$$

Theorem 1. *In the stable M/D/1 periodically gated vacation model satisfying assumptions* **A.1** - **A.3** *the transition probabilities of the Markov chain embedded at G-epochs are given as*

$$\begin{aligned} p^{(G)}(j,k) = & \sum_{m=\max(j-k+1,1)}^{j-1} \sum_{n=0}^{m-1} (1-p)^n p \left(e^{-\rho}\right)^{m-n-1} \\ & \sum_{l=1}^{k-j+m} \frac{\rho^l}{l!} e^{-\rho} \frac{(\rho n)^{k-j+m-l}}{(k-j+m-l)!} e^{-\rho n} \\ & + \sum_{m=j}^{\infty} \sum_{n=0}^{m-1} (1-p)^n p \left(e^{-\rho}\right)^{m-n-1} \sum_{l=1}^{k} \frac{\rho^l}{l!} e^{-\rho} \frac{(\rho n)^{k-l}}{(k-l)!} e^{-\rho n} \quad j,k \geq 1, \end{aligned} \tag{2}$$

where $\max(a,b)$ *stands for the highest value of set* (a,b).

Proof. We partition the transition probability from state j to state k on the number of time units in between, which is denoted by $m \geq 1$. The operation of the model implies that there must be at least one arrival during the transition. Let n denote the number of time units during the transition after the time unit of the first arrival. Clearly $0 \leq n \leq m-1$. The transition from state j to state k and the notations are shown in Fig. 3.

Either all the j customers are served during the transition, in which case $m \geq j$, or at least one of them is still in the system at the end of the transition, which implies $m \leq j-1$.

Let us firstly consider the latter case, i.e. when $m \leq j-1$. In this case there are at least $(j-m)$ customers present in the system at the end of the transition. Hence there are $k-(j-m)$ arrivals. It follows that $m \geq j-k+1$, since $k-j+m \geq 1$. However also $m \geq 1$ must hold and thus for the range of m we get $\max(j-k+1,1) \leq m \leq j-1$.

The gate period consists of the time unit of the first arrival and the next n time units. According to its shifted geometrical distribution the corresponding probability is given by $(1-p)^n p$. Assuming $l \geq 1$ arrivals in the time unit of the first arrival implies $k-j+m-l$ arrivals during the next n time units. It also implies that there is no arrival in the first $m-n-1$ time units of the transition. The corresponding probability summing over the possible values of l yields

$$\left(e^{-\lambda b}\right)^{m-n-1} \sum_{l=1}^{k-j+m} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-j+m-l}}{(k-j+m-l)!} e^{-\lambda n b}.$$

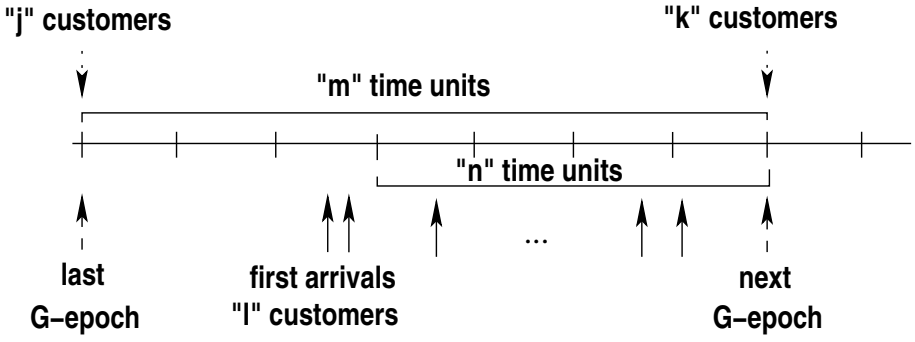


Fig. 3. Transition between two consecutive G-epochs

Taking into account also the probability of gate period and summing over the possible values of n gives the partial transition probability from state j to state k, given that the number of time units during the transition is m, and $m \leq j-1$, as

$$\sum_{n=0}^{m-1} (1-p)^n p \left(e^{-\lambda b}\right)^{m-n-1} \sum_{l=1}^{k-j+m} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-j+m-l}}{(k-j+m-l)!} e^{-\lambda n b}. \quad (3)$$

We follow the same line of arguments for the case of $m \geq j$. Now the boundary for m is given as $m \geq j$, since it is the only constraint for m. In this case there are k arrivals during the transition and thus l goes up to k. Therefore the partial transition probability from state j to state k, given that the number of time units during the transition is m, and $m \geq j$, can be expressed as

$$\sum_{n=0}^{m-1} (1-p)^n p \left(e^{-\lambda b}\right)^{m-n-1} \sum_{l=1}^{k} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-l}}{(k-l)!} e^{-\lambda n b}. \quad (4)$$

Utilizing $\rho = \lambda b$, the statement of the theorem comes from (3) and (4) by summing out m over the corresponding ranges. □

Let $p_k^{(G)}$ denote the equilibrium probability that the Markov chain embedded at G-epochs is in state k. To keep the computation of the equilibrium probabilities tractable we apply an upper limit X on the number of customers in the system, i.e. $k \leq X$. This results in finite number of equilibrium probabilities and transition probabilities and hence also finite number of equilibrium equations. We remark here that the resulted embedded Markov chain is always stable, since it has finite number of states due to the truncation and it is irreducible (see (2)). The proper value of X depends on the required precision and can be determined in iterative manner until the difference of consecutive values of probabilities $p_k^{(G)}$, for every $k \leq X$, becomes less than the specified error. In the computation, the probabilities $p_k^{(G)}$ for $k > X$ are set 0, since they can be neglected.

We define the $1 \times X$ vector $\boldsymbol{\theta}$, representing the equilibrium probabilities of the above Markov chain, by its k-th element as $[\boldsymbol{\theta}]_k = p_k^{(G)}$. We also define the $X \times X$ matrix $\boldsymbol{\Pi}$ representing the transition probabilities of the above Markov chain as $[\boldsymbol{\Pi}]_{j,k} = p^{(G)}(j,k)$. Due to the truncation, the row sums of matrix $\boldsymbol{\Pi}$ becomes less than 1. Hence, in the computation, these row sums are corrected to 1 (e.g. by applying normalization) ensuring that matrix $\boldsymbol{\Pi}$ remains stochastic, which is needed for the right computation.

The equilibrium probabilities of the Markov chain embedded at G-epochs can be uniquely determined from the following system of linear equations

$$\boldsymbol{\theta}\boldsymbol{\Pi} = \boldsymbol{\theta}, \quad \boldsymbol{\theta}\mathbf{e} = \sum_{k=1}^{X} p_k^{(G)} = 1, \tag{5}$$

where $\mathbf{e}$ denotes the $X \times 1$ column vector having all elements equal to one. Based on the $p^{(G)}(k)$ equilibrium probabilities the steady-state PGF of the number of customers at the G-epochs, $\widehat{g}(z)$, is defined as

$$\widehat{g}(z) = \sum_{k=1}^{\infty} p_k^{(G)} z^k, \quad |z| \leq 1.$$

3.2 The Number of Customers at Start of Vacation

Let $q_k^{(m)}$ be the steady-state probability that the number of customers in the system at start of vacation is k for $k > 0$. We define $\widehat{m}^*(z)$ as

$$\widehat{m}^*(z) = \sum_{k=0}^{\infty} q_k^{(m)} z^k, \quad |z| \leq 1.$$

Note that the probabilities $q_k^{(m)}$ covers only the cases when vacation is reached, i.e. buffer B becomes empty between two consecutive G-epochs. However buffer B does not necessarily become empty in these periods, only in a fraction of all possible stochastic sample paths. It follows that $\sum_{k=0}^{\infty} q_k^{(m)} < 1$ and hence the probabilities $q_k^{(m)}$ and $\widehat{m}^*(z)$ must be normalized. Therefore we also define $p_k^{(m)}$ as the steady-state conditional probability that the number of customers in the system at start of vacation is k, given that there is a vacation, for $k > 0$. The $p_k^{(m)}$ probabilities can be computed from the $q_k^{(m)}$ probabilities as

$$p_k^{(m)} = \frac{q_k^{(m)}}{\sum_{k=0}^{\infty} q_k^{(m)}}. \tag{6}$$

The PGF corresponding to the probabilities $p_k^{(m)}$ is defined as

$$\widehat{m}(z) = \sum_{k=0}^{\infty} p_k^{(m)} z^k, \quad |z| \leq 1.$$

The definitions of $\widehat{m}(z)$ and $\widehat{m}^*(z)$ as well as (6) imply that $\widehat{m}(z)$ can be expressed from $\widehat{m}^*(z)$ as

$$\widehat{m}(z) = \frac{\widehat{m}^*(z)}{\widehat{m}^*(1)}. \tag{7}$$

Theorem 2. *In the stable M/D/1 periodically gated vacation model satisfying assumptions* **A.1** - **A.3** *the relation between the PGFs of the number of customers at start of vacation and at G-epochs is given as*

$$\widehat{m}(z) = \frac{1-(1-p)\,e^{\rho}}{1-(1-p)\,e^{\rho z}}\,\frac{p\,\widehat{g}\left(e^{-\rho}\right)-(1-p)\left(e^{\rho z}-1\right)\widehat{g}\left((1-p)\,e^{-\rho(1-z)}\right)}{p\,\widehat{g}\left(e^{-\rho}\right)-(1-p)\left(e^{\rho}-1\right)\widehat{g}\left(1-p\right)}. \tag{8}$$

Proof. Let $j \geq 1$ be the number of customers at the G-epoch preceding the considered start of vacation. We condition the steady-state probability $q_k^{(m)}$ on j. The vacation starts after serving all j customers. Therefore the interval up to the start of vacation from the last G-epoch preceding it consists of exactly j time units. During this interval there is no new gate opening. Therefore either there is no arrival during these j time units or the gate period exceeds this interval.

The probability of no arrivals during j time units can be given as $\left(e^{-\lambda b}\right)^j$. Unconditioning on j gives the probability $q_0^{(m)}$ as

$$q_0^{(m)} = \sum_{j=1}^{\infty} p_j^{(G)} \left(e^{-\lambda b}\right)^j. \tag{9}$$

Now let us consider the other case. In this case there are k arrivals during the j time units. Let n denote the number of time units after the time unit of the first arrival. It follows that $0 \leq n \leq j-1$. The transition from the last G-epoch to the start of vacation and the notations are shown in Fig. 4.

The probability that the gate period exceeds the time until the start of vacation is given as $(1-p)^{n+1}$. Assuming $l \geq 1$ arrivals in the time unit of the first arrival implies $k-l$ arrivals during the next n time units. It also implies that there is no arrival in the first $j-n-1$ time units of the transition. The corresponding probability summing over the possible values of l results in

$$\left(e^{-\lambda b}\right)^{j-n-1} \sum_{l=1}^{k} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-l}}{(k-l)!} e^{-\lambda n b}.$$

Taking into account also the probability of no gate openings and summing over the possible values of n gives the conditional probability of having k customers in the system at start of vacation, given that j customers are present in the system at last G-epoch, as

$$\sum_{n=0}^{j-1} (1-p)^{n+1} \left(e^{-\lambda b}\right)^{j-n-1} \sum_{l=1}^{k} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-l}}{(k-l)!} e^{-\lambda n b}.$$

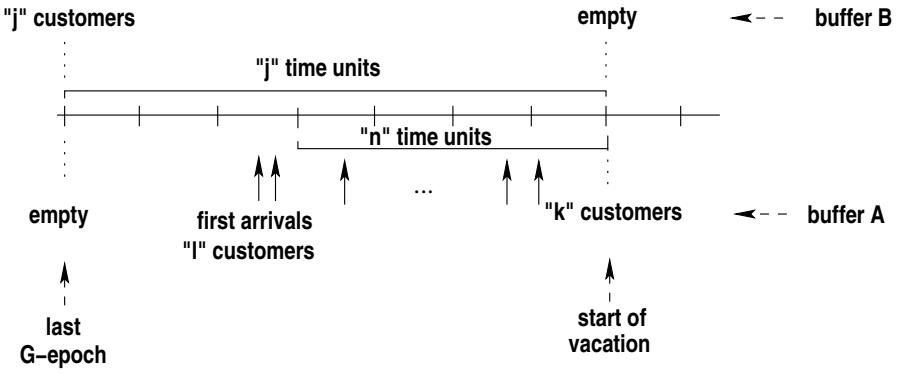


Fig. 4. Transition from last G-epoch to start of vacation

Unconditioning on j results in the probability $q_k^{(m)}$ for $k \geq 1$ as

$$q_k^{(m)} = \sum_{j=1}^{\infty} p_j^{(G)} \sum_{n=0}^{j-1} (1-p)^{n+1} \left(e^{-\lambda b}\right)^{j-n-1} \sum_{l=1}^{k} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-l}}{(k-l)!} e^{-\lambda n b}. \quad (10)$$

Using (9), (10) and the definition of $\widehat{m}^*(z)$, as well as rearranging yield

$$\widehat{m}^*(z) = \frac{p\ \widehat{g}\,(e^{-\rho}) - (1-p)\,(e^{\rho z}-1)\,\widehat{g}\left((1-p)\ e^{-\rho(1-z)}\right)}{1-(1-p)\ e^{\rho z}}. \quad (11)$$

Applying (11) to (7) results in the statement of the theorem. □

The mean of the steady-state number of customers at start of vacation can be computed by taking the first derivative of (8) at $z = 1$, which results in

$$\begin{aligned} m^{(1)} = & \frac{(1-p)\rho\,(p\ e^{\rho}\widehat{g}\,(e^{-\rho}) - p\ e^{\rho}\widehat{g}(1-p))}{(1-(1-p)\ e^{\rho})\,(p\widehat{g}\,(e^{-\rho}) - (e^{\rho}-1)\,(1-p)\widehat{g}(1-p))} \\ & - \frac{(1-p)\rho\,((e^{\rho}-1)\,(1-(1-p)\ e^{\rho})\,(1-p)\widehat{g}'(1-p))}{(1-(1-p)\ e^{\rho})\,(p\widehat{g}\,(e^{-\rho}) - (e^{\rho}-1)\,(1-p)\widehat{g}(1-p))} \end{aligned} \quad (12)$$

3.3 The Number of Customers at End of Vacation

Let $p_k^{(f)}$ be the steady-state probability that the number of customers in the system at end of vacation is k for $k > 0$. The corresponding PGF of the steady-state number of customers at end of vacation, $\widehat{f}(z)$, is defined as

$$\widehat{f}(z) = \sum_{k=1}^{\infty} p_k^{(f)} z^k, \quad |z| \leq 1.$$

Theorem 3. *In the stable M/D/1 periodically gated vacation model satisfying assumptions* **A.1** - **A.3** $\widehat{f}(z)$ *can be expressed by means of* $\widehat{m}(z)$ *as*

$$\widehat{f}(z) = \frac{pe^{-\rho(1-z)}\widehat{m}(z) - p\ p_0^{(m)} e^{-\rho\frac{1-e^{-\rho(1-z)}}{1-e^{-\rho}}}}{1-(1-p)e^{-\rho(1-z)}}, \tag{13}$$

where $p_0^{(m)}$ *is given as*

$$p_0^{(m)} = \frac{(1-(1-p)\ e^{\rho})\ \widehat{g}\,(e^{-\rho})}{p\ \widehat{g}\,(e^{-\rho}) - (1-p)\,(e^{\rho}-1)\,\widehat{g}\,(1-p)}. \tag{14}$$

Proof. The number of customers at end of vacation is at least one, since also the end of vacation is a G-epoch. The number of customers at end of vacation is the sum of the number of customers at start of vacation and those which arrive during the vacation period.

Either there are no customers in the system at start of vacation, in which case the gate can not be opened before the time unit of the first arrival, or there is at least one customer present at start of vacation, in which case the gate opening can occur at end of any of subsequent time units.

We start with the case when no customers are present in the system at start of vacation. Assuming m time units before the time unit of the first arrival and n time units afterwards as well as l arrivals in the time unit of the first arrival (see Fig. 5), the corresponding partial probability of having k customers at end of vacation can be expressed as

$$p_0^{(m)} \sum_{m=0}^{\infty} \left(e^{-\lambda b}\right)^m \sum_{n=0}^{\infty} (1-p)^n p \sum_{l=1}^{k} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-l}}{(k-l)!} e^{-\lambda n b}. \tag{15}$$

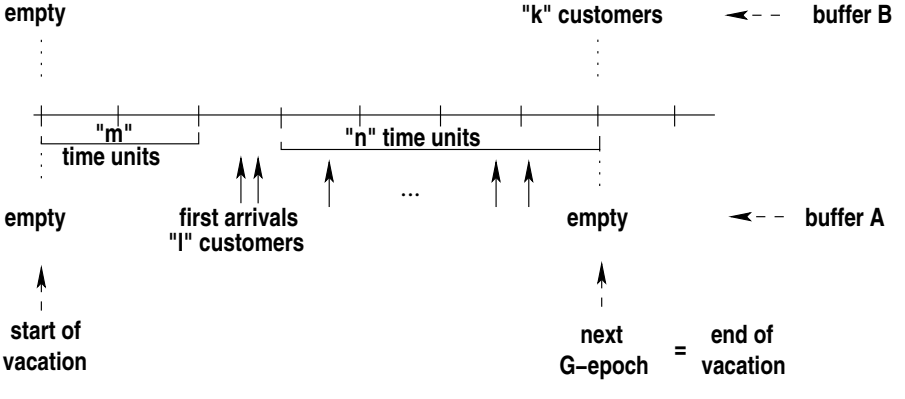


Fig. 5. Transition during vacation when no customer present at start of vacation

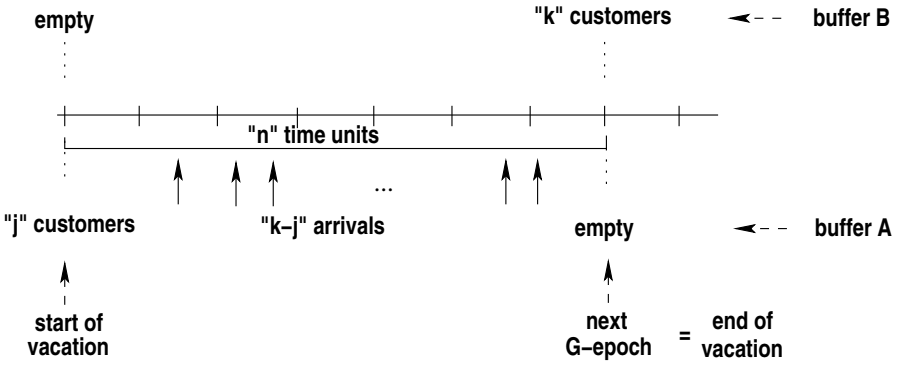


Fig. 6. Transition during vacation when at least one customer present at start of vacation

In the other case there are $j \geq 1$ customers in the system at start of vacation. Hence there are $k - j$ arrivals during the vacation consisting of n time units. Thus the gate period ends exactly after n time units (see Fig. 6). Hence the corresponding partial probability of having k customers at end of vacation can be given for this case as

$$\sum_{j=1}^{k} p_j^{(m)} \sum_{n=1}^{\infty} (1-p)^{n-1} p \frac{(\lambda n b)^{k-j}}{(k-j)!} e^{-\lambda n b}. \tag{16}$$

Putting (15) and (16) together the probability of having k customers at end of vacation can be expressed as

$$\begin{aligned} p_k^{(f)} = p_0^{(m)} \sum_{m=0}^{\infty} \left(e^{-\lambda b}\right)^m \sum_{n=0}^{\infty} (1-p)^n p \sum_{l=1}^{k} \frac{(\lambda b)^l}{l!} e^{-\lambda b} \frac{(\lambda n b)^{k-l}}{(k-l)!} e^{-\lambda n b} \\ + \sum_{j=1}^{k} p_j^{(m)} \sum_{n=1}^{\infty} (1-p)^{n-1} p \frac{(\lambda n b)^{k-j}}{(k-j)!} e^{-\lambda n b}. \end{aligned} \tag{17}$$

Starting from the definition of $\widehat{f}(z)$, using (17) and performing several rearrangements results in the first statement of the theorem. The second statement comes by setting $z = 0$ in (8) and rearranging it. □

The mean of the steady-state number of customers at end of vacation can be computed by taking the first derivative of (13) at $z = 1$, which results in

$$f^{(1)} = \frac{\left(p\, p_0^{(m)} + e^{\rho} - 1\right) \rho + p\,(e^{\rho} - 1)\, m^{(1)}}{p\,(e^{\rho} - 1)} \tag{18}$$

4 Analysis at Arbitrary Epoch

In this section we give the expressions of the PGF and the mean of the steady-state number of customers in an arbitrary epoch. Afterwards we summarize the procedure for computing the above mentioned mean.

4.1 The Number of Customers at Arbitrary Epoch

Let $N(t)$ be the number of customers in the system at time t for $t \geq 0$. We define $\widehat{q}(z)$ as the PGF of the steady-state number of customers in an arbitrary epoch as

$$\widehat{q}(z) = \lim_{t\to\infty} \sum_{k=0}^{\infty} P\{N(t) = k\} z^k, \quad |z| \leq 1.$$

Theorem 4. *In the stable M/D/1 periodically gated vacation model satisfying assumptions* **A.1** - **A.3** *the PGF of the steady-state number of customers in an arbitrary epoch can be expressed as*

$$\widehat{q}(z) = \frac{(1-\rho)e^{-\rho(1-z)}}{e^{-\rho(1-z)} - z} \frac{p\,(e^{\rho}-1)}{\left(p\; p_0^{(m)} + e^{\rho} - 1\right)\rho} \times \frac{\left(1 - e^{-\rho(1-z)}\right)\widehat{m}(z) + p\; p_0^{(m)} e^{-\rho} \frac{1-e^{-\rho(1-z)}}{1-e^{-\rho}}}{1-(1-p)e^{-\rho(1-z)}}, \tag{19}$$

where $\widehat{m}(z)$ *and* $p_0^{(m)}$ *are given by (8) and (14), respectively.*

Proof. In the classical vacation model the steady-state number of customers in an arbitrary epoch can be expressed in terms of the steady-state number of customers at start and end of vacation (see in [3] and [6]) as

$$\widehat{q}(z) = \frac{(1-\rho)\widetilde{B}(\lambda-\lambda z)}{\widetilde{B}(\lambda-\lambda z) - z} \frac{\widehat{m}(z) - \widehat{f}(z)}{f^{(1)} - m^{(1)}}, \tag{20}$$

where $\widetilde{B}(s)$ is the Laplace-Stieljes transform (LST) of the customer service time. In fact the arguments used for the proof of this relation in [3] are valid for models, in which the work-conserving property holds during the service period, i.e. this relation holds in broader settings. It follows that it holds also for the periodically gated vacation model.

Due to constant customer service time $\widetilde{B}(s) = e^{-sb}$. Using it and applying (13) and (18) in (20) gives the statement of the theorem. □

Corollary 1. *In the stable M/D/1 periodically gated vacation model satisfying assumptions* **A.1** - **A.3** *the mean of the steady-state number of customers in an arbitrary epoch can be expressed as*

$$q^{(1)} = \rho + \frac{\rho^2}{2(1-\rho)} + \frac{(2-p)\rho^2\left(p\ p_0^{(m)} + e^{\rho} - 1\right) + 2p\ \rho\,(e^{\rho}-1)\,m^{(1)}}{2p\left(p\ p_0^{(m)} + e^{\rho} - 1\right)\rho}, \quad (21)$$

where $m^{(1)}$ *and* $p_0^{(m)}$ *are given by (12) and (14), respectively.*

Proof. The statement comes by taking the first derivative of (19) at $z = 1$. □

4.2 The Computational Procedure

The computational steps of the mean steady-state number of customers of this model can be summarized as follows:

1. Calculation of the equilibrium probabilities of the Markov chain embedded at G-epochs, $p_k^{(G)}$ for $1 \leq k \leq X$, from the system of linear equations (5).
2. Calculation of the moment $m^{(1)}$ and the probability $p_0^{(m)}$ from the probabilities $p_k^{(G)}$ by applying (12), (14) and the definition of $\widehat{g}(z)$.
3. Computation of $q^{(1)}$ from the quantities $m^{(1)}$ and $p_0^{(m)}$ by applying (21).

4.3 Numerical Example

Fig. 7 shows the dependency of $q^{(1)}$ on the parameter p for several values of ρ.

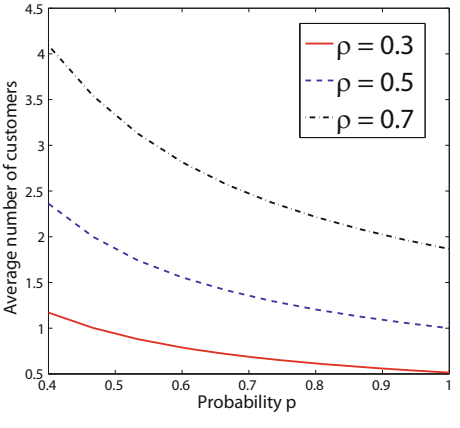


Fig. 7. The dependency of the mean steady-state number of customers on the parameter p for several values of ρ

5 Application to the IEEE 802.16 Network

We apply the periodically gated vacation model to the uplink (UL) non real-time (nrtPS) traffic in IEEE 802.16 based network. We apply the framework of [1].

5.1 Analytical Model of the Uplink Non Real-Time Traffic in the IEEE 802.16 Network

We assume that the probability of the successful BW-Req transmission at each SS (denoted by $p_i^{(b)}$) is constant (this is based on [2]). As a consequence of it the statistical behavior of each SS can be considered independently of the other SSs. According to it we consider the behavior of the tagged SS i.

The tagged SS i corresponds to the periodically gated vacation model. Packets arriving to SS i are the customers of the model. The reservation buffer and the scheduling buffer corresponds to buffer A and B of the queue, respectively. The successful BW-Req transmission is represented by a gate opening event. The scheduling of a packet is represented by a customer service. The constant length frame corresponds to b. The probability of successful bandwidth reservation ($p_i^{(b)}$) is the parameter p of the model. Finally the packet arrival rate at SS i corresponds to λ of the model.

5.2 The Mean Packet Delay

The *overall delay* is composed of several parts, from which the major parts are the *reservation delay*, W_i^r, and the *scheduling delay*, W_i^s. The reservation and the scheduling delay is the time period the packet takes in the reservation and in the scheduling buffer, respectively. Therefore $E\left[W_i^r + W_i^s\right]$ can be computed from the steady-state number of customers in the queue of the periodically gated vacation model ($r^{(1)}$) by applying Little's law as

$$E\left[W_i^r + W_i^s\right] = \frac{r^{(1)}}{\lambda_i}, \tag{22}$$

where $r^{(1)}$ can be expressed by applying (21) as

$$r^{(1)} = q^{(1)} - \rho = \frac{\rho^2}{2(1-\rho)} + \frac{(2-p)\rho^2\left(p\,p_0^{(m)} + e^{\rho} - 1\right) + 2p\,\rho\,(e^{\rho}-1)\,m^{(1)}}{2p\left(p\,p_0^{(m)} + e^{\rho} - 1\right)\rho}. \tag{23}$$

References

1. Andreev, S., Saffer, Z., Turlikov, A.: Delay Analysis of Wireless Broadband Networks with Non Real-Time Traffic. In: Sacchi, C., Bellalta, B., Vinel, A., Schlegel, C., Granelli, F., Zhang, Y. (eds.) MACOM 2011. LNCS, vol. 6886, pp. 206–217. Springer, Heidelberg (2011)
2. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications 18(3), 535–547 (2000)
3. Borst, S.C., Boxma, O.J.: Polling models with and without switchover times. Operations Research 45, 536–543 (1997)
4. Doshi, B.T.: Queueing systems with vacations - a survey. Queueing Systems 1, 29–66 (1986)

5. IEEE 802.16-2009, Part 16: Air Interface for Broadband Wireless Access Systems, Standard for Local and Metropolitan Area Networks (May 2009)
6. Saffer, Z.: An introduction to classical cyclic polling model. In: Proc. of the 14th Int. Conf. on Analytical and Stochastic Modelling Techniques and Applications (ASMTA 2007), pp. 59–64 (2007)
7. Saffer, Z., Telek, M.: M/G/1 Queue with Exponential Working Vacation and Gated Service. In: Al-Begain, K., Balsamo, S., Fiems, D., Marin, A. (eds.) ASMTA 2011. LNCS, vol. 6751, pp. 28–42. Springer, Heidelberg (2011)
8. Saffer, Z., Yue, W.: Analysis of Multi-Server Queue with Synchronous Vacations and Gated Discipline. In: 6th International Conference on Queueing Theory and Network Applications (QTNA 2011), Seoul, Korea (2011)
9. Servi, L.D., Finn, S.G.: M/M/1 queue with working vacations (M/M/1/WV). Performance Evaluation 50, 41–52 (2002)
10. Takagi, H.: Queueing Analysis - A Foundation of Performance Evaluation, Vacation and Prority Systems, vol. 1. North-Holland, New York (1991)
11. Tian, N., Zhang, G.: Vacation Queueing Models: Theory and Applications. International Series in Operations Research & Management Science, vol. 93, XII. Springer, New York (2006)
12. Wu, D., Takagi, H.: M/G/1 queue with multiple working vacations. Performance Evaluation 63, 654–681 (2006)
13. Yue, W., Takahashi, Y., Takagi, H.: Advances in Queueing Theory and Network Applications. Springer Science + Business Media, New York (2010)

Combined CAC and Forced Handoff for Mobile Network Performability

Idriss-Ismael Aouled[1,2] and Hind Castel-Taleb[1]

[1] INSTITUT TELECOM, Telecom SudParis/SAMOVAR
9, rue Charles Fourier 91011 Evry Cedex, France
{hind.castel,idriss.ismael}@it-sudparis.eu
[2] LACL, Université Paris-Est, Créteil Val de Marne
61, av. du Général de Gaulle, 94010 Créteil Cedex, France

Abstract. In mobile networks, call admission control (CAC) is widely used in reaching of the quality of service (QoS) requirements. However, as the CACs schemes give priority to the handoff calls, the blocking probability is degraded. In this paper we propose a new scheme which is based on the combinaison of CAC scheme and load sharing policy between a cluster of surrounding cells. Our scheme forces some calls to handover, with conditions, to neighboring cells in order to avoid the blocking states in the serving cell. Thus we prove, in the case of one cell, that our scheme permits to improve both the dropping and blocking probabilities.

We use multidimensional's Markov chains to model the systems because of the consideration of occupation and failure/reparations of channels. Therefore, it is difficult to deduce intuitively the relevance of our scheme versus others in the literatures. So, we apply a mathematical method based on stochastic comparisons in other to prove that our scheme provides better performance measures. We illustrate these proofs by numerical results in order to show the relevance of our mechanism to improve QoS of mobile networks.

Keywords: Performability, Handoff, Load sharing, QoS, Stochastic comparisons, Markov chains.

1 Introduction

In cellular and mobile communication system, user's mobility during service access is a key element. Sometimes, it may happen that a moving mobile terminal (MT) could not receive a good signal from its serving base station (BS). Then the control element of system may decide to hand over this MT to one other better serving BS. Unfortunately, the handoff call may be dropped due to the lack of good quality signal or unavailable channel from target cell (adjacent). Since dropping a call in progress is considered to have a more negative impact from the user's perspective than a new incoming call, then an extensive work has been done in the definition of call admission control (CAC) mechanisms [1,2] in

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 76–90, 2012.

order to reduce the dropping probability of ongoing calls. Other techniques related to the signal quality are very relevant to ensure that the handoff call is not dropped. The MAHO (Mobile Assisted HandOff) [3] is based on collective decision between the MT and the surrounding BS. The MT reports periodically their RSSI/BER (received signal strenght indicator, bit error rate) values to assist the serving BS for handoff decision. Although this scheme ensures that the MT has a good signal before handoff, it doesn't provide information about the channel availability. Channel reservation for handoff calls are efficient techniques to limit the call dropping probability. In [4], one or more channels are reserved (called the guard channels - GC), and the CAC mechanism is implemented. Guard channels based scheme had focused only on the channel availability while making handoff decision. Therefore the handed call faces the risk of being dropped in case of the target BS offers poor signal quality. Since these two techniques are based on single criterion, signal quality for MAHO and number of guard channel for GC, in [5] authors propose a multi-criteria scheme to address this limitation and reduce the handoff dropping probability. This scheme, called MG and which combines MAHO and GC, ensures that a call is handed off if the target cell is able to provide a free channel as well as an acceptable signal quality. All of these techniques contribute certainly to reduce the dropping of ongoing calls but unfortunately increase the blocking probability of the new calls. Further techniques based on load or resource availability in home cell and neighbour cells are proposed in [6,7] without significant improvement of the system capacity compared to the introduced complexity [8]. A comprehensive survey on handoff and CAC techniques can be found in [1]. To address the debasement of the blocking probability because of the mechanism of guard channel, we propose a new scheme to improve the dropping probability of ongoing calls and, at the same time, to limit the blocking probability of new calls. The proposed technique, called LMG (Load sharing Maho with Guard channels) includes two mechanisms. The first one is the CAC mechanism based on a number g of guard channels reserved for the handovers. If the cell is under-loaded (when the number of free channels is upper or equal to the guard channel threshold g) then both new and handoff calls are accepted for services. Whenever the number of free channels goes lower than g, the new calls are refused and only handoff calls get services. Moreover, MG makes sure that MT has a good signal quality and available channel from target cell. The second one implements the load sharing policy when the cell is overloaded. We consider the threshold s which is the minimum number of free channels in the cell from which we decide to force the handover for calls in the edge of the cell. The handovers are forced to adjacent cells which are able to provide acceptable signal quality until the total number of free channels goes upper than s. Therefore, we reduce the load of the cell and we improve the blocking probability of new calls.

In practice, the systems we consider are usually represented by multidimensional processes with very large state spaces, so quantitative analysis is very difficult if there is no specific solution form [9]. We propose to use a mathematical method based on stochastic comparisons of Markov processes [10]. Stochastic

comparison method can be applied for the performance evaluation of different kinds of network architectures [11].

This method can be used to compare two different systems from their performance or reliability, or also to build from a complex system, bounding systems easier to solve as an example on a reduced state space [12]. More recently, metrics like first flush occurrence on G-networks with catastrophe [13] have been studied using the stochastic comparison. In [14], bounds are defined by removing links between queues in a Jackson network in order to compute the transient probability distribution. In this paper, we use stochastic comparisons in order to prove mathematically that our system (LMG) provides better performance measures (handoff-call blocking probability and new call blocking probability) than MG system. The systems that we study are complex to analyse. Because they are multidimensional and moreover they evolve with many events as we consider both occupation and failures/reparation of the channels. So we need to apply a mathematical method in order to prove that the LMG system is better than MG. The stochastic comparison is very efficient, as we compare the equation evolutions of the systems using the coupling by events [15]. We give numerical results on the performance measures in order to show that our mechanism is efficient. This paper is organized as follows: in the section after, we describe MAHO, MG, and LMG systems. After that, we present the stochastic comparison method, and we prove that our system provides better performance measures. In section IV, we give the numerical results on the performance measures. Achieved results and extension of this work are discussed in the conclusion.

2 Systems Description

The system we study is represented by a cellular network where each cell contains n channels. Under the condition that all neighbouring cells are statistically identical, a single cell in isolation is representative and all interactions will be captured through a handoff call arrival process [16,17]. The system is prone to the channel failure and reparation. If failure occurs on an occupied channel, the communication is switched to another free channel if available and lost otherwise [18]. We consider a channel failure by the unavailability of such channel due to excessively interference, multi-path, fading, etc. We assume that the arrival process of new calls and handoff calls are Poisson. The call holding time and cell residency for both types of calls is exponentially distributed. The same is for times to channel failure and reparation. We assume that all channels share a single repair facility. Let the set $N = \{0, \ldots, n\}$. We denote by $x_c \in N$ and $x_e \in N$ denote the number of occupied channels in the center of cell and the edge of the cell. If $x \in N$ denotes the total idle channels into the cell, then $(x_c + x_e) \leq x$. Thus the model described above is a composite model for the combined performance and availability analysis. It is an homogeneous irreducible Continuous-time Markov Chain (CTMC), taking values in the finite state space $A = N^3$ and each state of cell is defined by the tuple $\boldsymbol{x} = (x, x_c, x_e)$.

The system is described by the following parameters.

λ_c	new call arrival rate is in the center of cell
λ_e	new call arrival rate in the edge of the cell
λ_h	handoff-call arrival rate in the edge of the cell
μ_t	the rate for a call to terminate into the cell
μ_r	channel residency time
γ	failure rate
$1/\tau$	mean channel repair time

Note that a new call can be either destined to the center or to the edge of cell with rate λ_c and λ_e respectively. In the other hand, handoff arrivals (with rate λ_h) and departure (with rate μ_r) from cell can only occur on the edge. We assume that there is no local handover between center and edge area in the same cell [19]. Because of the tehchnique of frequency hopping, the interference level is the same for all channels. Therfore, the intracell mobility not provides significant improvement and, it is possible to switch off the corresponding Flag.

In the next section, we describe in detail the different schemes that we want to compare.

2.1 MAHO Scheme Based Model

Since the standard MAHO is based only on the signal quality, new and ongoing calls access fairly to the resources when there are available channels. In this scheme, we consider both occupation and availability of the resource due to failures and reparations. When MAHO is applied on the understudied model, we have the following cases for the transitions :

- The arrival rate of new calls in the center of the cell is λ_c if $x_c + x_e < x$, otherwise it is null. So the evolution equation of the system is such that x_c increases by one (see (1)).
- The arrival rate of new calls in the edge of the cell is λ_e if $x_c + x_e < x$, otherwise it is null. The evolution equation is given by transition (2).
- The arrival rate of handoff calls in the edge of the cell is λ_h if $x_c + x_e < x$, otherwise it is null. The evolution equation is given by transition (2).
- The total service rate denoted μ_c for calls in the center of the cell is the rate to have a free channel in the center of the cell. As we have x_c busy channels then $\mu_c = x_c * \mu_t$. The evolution equation is given by transition (3).
- The total service rate denoted μ_e for calls in the edge of the cell is the sum of the rate for a call to terminate and the handoff departure rate so it is $\mu_e = x_e * (\mu_t + \mu_r)$. The evolution equation is given by transition (4).
- The failure rate is $x_c\gamma$ for busy channels in the center of the cell if all the channels are busy : $x = x_c + x_e$. So in this case both x and x_c decreases (see equation 5). The evolution equation is given by transition (5). It is similar for the calls at the edge of the cell, see transition (6).

- The failure rate is $(x-(x_c+x_e))*\gamma+(x_c+x_e)\gamma$ equivalent of fail of a free channel, or a busy channel in the center or in the edge, if $x-(x_c+x_e)>0$. In the case where a busy channel fails, the communication is switched to another free channel, so only the number of available channels x decreases by one. The evolution equation is given by transition (7).
- The repair rate is τ, so x increases by one (see transition (8)).

So the evolution equations are as follows :

$$
\begin{aligned}
\boldsymbol{x} &\rightarrow (x, min\{n, x_c+1\}, x_e), with\ rate\ \lambda_c, && (1)\\
&\quad if\ x_c+x_e<x.\\
&\rightarrow (x, x_c, min\{n, x_e+1\}),\ with\ rate\ \lambda_e+\lambda_h, && (2)\\
&\quad if\ x_c+x_e<x.\\
&\rightarrow (x, max\{0, x_c-1\}, x_e),\ with\ rate\ \mu_c=x_c\mu_t. && (3)\\
&\rightarrow (x, x_c, max\{0, x_e-1\}),\ with\ rate\ \mu_e=x_e(\mu_t+\mu_r). && (4)\\
&\rightarrow (max\{0, x-1\}, max\{0, x_c-1\}, x_e), && (5)\\
&\quad with\ rate\ x_c\gamma\ if\ x=x_c+x_e.\\
&\rightarrow (max\{0, x-1\}, x_c, max\{0, x_e-1\}), && (6)\\
&\quad with\ rate\ x_e\gamma\ if\ x=x_c+x_e.\\
&\rightarrow (max\{0, x-1\}, x_c, x_e), && (7)\\
&\quad with\ rate\ (x_e+x_c)\gamma+(x-(x_c+x_e))\gamma,\\
&\quad if\ x>(x_c+x_e).\\
&\rightarrow (min\{n, x+1\}, x_c, x_e), && (8)\\
&\quad with\ rate\ \tau.
\end{aligned}
$$

For this model, we have represented the corresponding CTMC for $n=3$ in Fig.1. We can see easily the complexity of the system from the number of states and kinds of events. The handoff calls dropping probabilities P_d^{MAHO} and new calls blocking probabilities P_b^{MAHO}, are computed on the same states (precisely on states $\boldsymbol{x}$ such that $x=x_c+x_e$). We denote by $\pi_{i,j,k}^{MAHO}$ the probability of the state $\boldsymbol{x}=(x, x_c, x_e)$ to have the value (i, j, k). Obviously, both P_d^{MAHO} and P_b^{MAHO} are equals and are written as follows:

$$P_d^{MAHO}=P_b^{MAHO}=\sum_{j=0}^{n}\sum_{i=j}^{n}\pi_{i,j,i-j}^{MAHO} \qquad (9)$$

Next, we present the MG scheme which combines the MAHO system, with a CAC procedure based on g guard channels (GC) in order to give the priority to handoff calls over new calls. The goal is to reduce the handoff calls dropping probability.

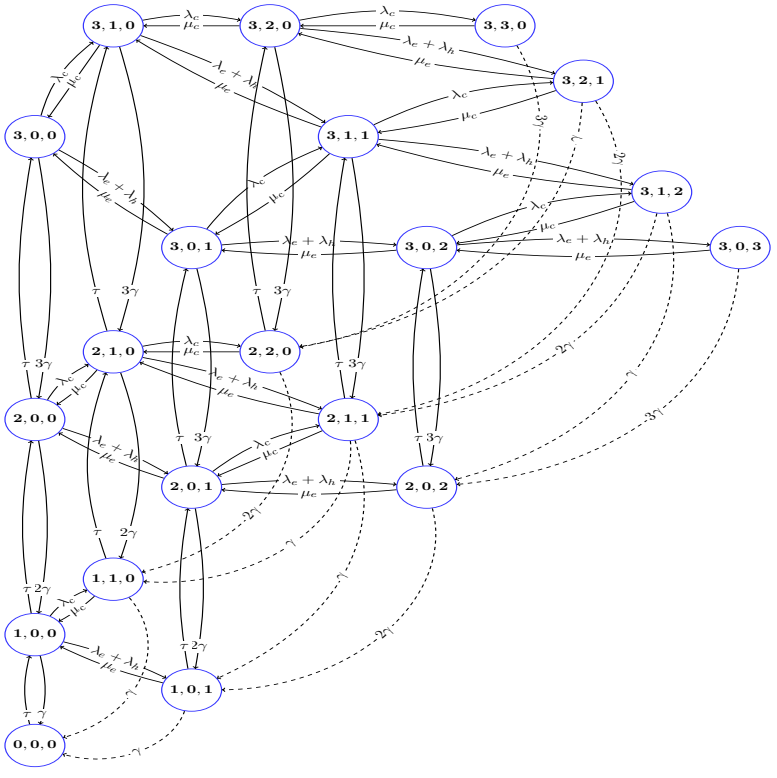


Fig. 1. The MAHO Markov Chain model

2.2 MG Scheme: Combined MAHO and GC Schemes

The MG scheme considers both MAHO scheme and a CAC (Call Admission Control) procedure, based on a guard channel scheme (GC) which reserves a subset of g channels for handoff calls. Whenever the number of free channels is lower or equal than the threshold g, CAC rejects new calls and accepts only handoff calls until the number of free channels goes upper than the threshold. In the case of all the channels are occupied, both new and handoff calls are rejected. MAHO scheme ensures that a handed call has always an acceptable signal quality. It means that $\alpha = 1$, where α is the probability that the handoff call will be a success [5]. From state $\boldsymbol{x} = (x, x_c, x_e)$, the different transition rates are defined as follows when MG is applied:

1. The total arrival rate of new calls in the center of the cell is λ_c if $x-(x_c+x_e) > g$ otherwise it is null (see transition (10)).
2. The total arrival rate in the edge of the cell is the sum of the new call arrival rate in the edge of the cell and handoff arrival rate $\lambda_e + \lambda_h$ if the number of free channels is upper than the number of guard channels ($x-(x_c+x_e) > g$), (see transition (11)). Otherwise, only handed off calls are admitted, so the arrival rate is λ_h (see transition 12).

3. The total service rate for calls in the center of the cell is $\mu_c = x_c * \mu_t$, (see transition 13).
4. The total service rate for calls in the edge of the cell is $\mu_e = x_e * (\mu_t + \mu_r)$, (see transition 14)
5. The failure rate is $x_c\gamma$ for busy channels in the center of the cell if all the channels are busy : $x = x_c + x_e$. The evolution equation is given by transition (15). It is similar for the calls at the edge of the cell, see transition (16).
6. The failure rate is $(x - (x_c + x_e)) * \gamma + (x_c + x_e)\gamma$ equivalent of fail of a free channel, or a busy channel in the center or in the edge, if $x - (x_c + x_e) > 0$. In the case where a busy channel fails, the communication is switched to another free channel, so only the number of available channels x decreases by one. The evolution equation is given by transition (17).
7. The repair rate is τ, so x increases by one (see transition (18)).

And the evolution equations of the system are as follows:

$$
\begin{aligned}
\boldsymbol{x} &\rightarrow (x, min\{n, x_c + 1\}, x_e), with\ rate\ \lambda_c, && (10)\\
&\quad if\ x - (x_c + x_e) > g.\\
&\rightarrow (x, x_c, min\{n, x_e + 1\}),\ with\ rate\ \lambda_e + \lambda_h, && (11)\\
&\quad if\ x - (x_c + x_e) > g.\\
&\rightarrow (x, x_c, min\{n, x_e + 1\}),\ with\ rate\ \lambda_h, && (12)\\
&\quad if\ x - (x_c + x_e) \leq g.\\
&\rightarrow (x, max\{0, x_c - 1\}, x_e),\ with\ rate\ \mu_c = x_c\mu_t. && (13)\\
&\rightarrow (x, x_c, max\{0, x_e - 1\}),\ with\ rate\ \mu_e = x_e(\mu_t + \mu_r). && (14)\\
&\rightarrow (max\{0, x - 1\}, x_c, max\{0, x_e - 1\}), && (15)\\
&\quad with\ rate\ x_e\gamma\ if\ x = x_c + x_e.\\
&\rightarrow (max\{0, x - 1\}, max\{0, x_c - 1\}, x_e), && (16)\\
&\quad with\ rate\ x_c\gamma\ if\ x = x_c + x_e.\\
&\rightarrow (max\{0, x - 1\}, x_c, x_e), && (17)\\
&\quad with\ rate\ (x_e + x_c)\gamma + (x - (x_c + x_e))\gamma,\\
&\quad if\ x > (x_c + x_e).\\
&\rightarrow (min\{n, x + 1\}, x_c, x_e), && (18)\\
&\quad with\ rate\ \tau.
\end{aligned}
$$

For this model, the handoff calls dropping probabilities P_d^{MG} are computed on states $\boldsymbol{x}$ such that $x = x_c + x_e$. We denote by $\pi^{MG}(i, j, k)$ the probability of the state $\boldsymbol{x} = (x, x_c, x_e)$ to have the value (i, j, k). So P_d^{MG} is computed by summing the probabilities of states (i, j, k) such that $k = i - j$, by considering that $0 \leq j \leq n - g$ due to the reserved channels for handoff calls. For P_b^{MG}, the probability is computed by summing the probabilities $\pi_{i,j,k}^{MG}$ such that $i - (j + k) \leq g$.

The handoff call dropping probabilities and the new call blocking probabilities are then given by:

$$P_d^{MG} = \sum_{j=0}^{n-g} \sum_{i=j}^{n} \pi_{i,j,i-j}^{MG} \; and \; P_b^{MG} = \sum_{j=0}^{n-g} \sum_{k=0}^{g} \sum_{i=j+k}^{n} \pi_{i,j,i-j-k}^{MG}. \tag{19}$$

We can notice that, although MG improves the dropping probability, the exclusive reservation of guard channels for handoff calls, degrades the blocking probability of new calls.

2.3 Our Proposed Scheme: LMG (Load Sharing Maho with Guard Channels)

The relevance of our scheme is to combine both the CAC mechanism with forced handoff in order to improve both the dropping probability and the blocking probability. We consider two thresholds : g for the number of channels reserved for handovers, and s which represents the minimum number of free channels from which we decide to force the calls located in the edge of cell to handoff to adjacent cells. In order to facilitate the presentation, we suppose that (g=s).

The forced handoff works by modifying the channel residency time for calls that are on the edge of the cell according to the load. If the number of free channels exceeds g, the channel residency rate is μ_r. So the total service rate is the sum of the rate for a call to terminate and the channel residency rate is $\mu_e = x_e * (\mu_t + \mu_r)$ (see equation (24)). In the case where the number of free channels is lower than g, our scheme forces calls located at edge of the cell to handed off to the adjacent cells with rate μ_{r0}, under conditions of good signal and channel availability. The total service rate is $\mu_{e0} = x_e * (\mu_t + \mu_r + \mu_{r0})$ and the transition is given in equation (25). In this scheme, users of overloaded cell will be served by adjacent cells to avoid loss. Load share can be achieved by controlling various radio access parameters and push to the less loaded cell with threshold mechanism that can be used for load sharing.

From the assumptions on the call arrivals and the services, we deduce that the system is represented by a Markov chain denoted $\{X^{LMG}(t), t \geq 0\}$. From state $\boldsymbol{x} = (x, x_c, x_e)$, the different transitions are as follows :

$$\begin{aligned}
\boldsymbol{x} &\rightarrow (x, min\{n, x_c + 1\}, x_e), with\ rate\ \lambda_c, && (20)\\
&\quad if\ x - (x_c + x_e) > g.\\
&\rightarrow (x, x_c, min\{n, x_e + 1\}),\ with\ rate\ \lambda_e + \lambda_h, && (21)\\
&\quad if\ x - (x_c + x_e) > g.\\
&\rightarrow (x, x_c, min\{n, x_e + 1\}),\ with\ rate\ \lambda_h, && (22)\\
&\quad if\ x - (x_c + x_e) \leq g.\\
&\rightarrow (x, max\{0, x_c - 1\}, x_e),\ with\ rate\ \mu_c = x_c\mu_t. && (23)\\
&\rightarrow (x, x_c, max\{0, x_e - 1\}),\ with\ rate\ \mu_e = x_e(\mu_t + \mu_r), && (24)\\
&\quad if\ x - (x_c + x_e) > g.
\end{aligned}$$

$$\rightarrow (x, x_c, max\{0, x_e - 1\}),\ with\ rate\ \mu_{e0} = x_e(\mu_t + \mu_r + \mu_{r0}), \quad (25)$$
$$if\ x - (x_c + x_e) \leq g.$$
$$\rightarrow (max\{0, x - 1\}, x_c, max\{0, x_e - 1\}), \quad (26)$$
$$with\ rate\ x_e\gamma\ if\ x = x_c + x_e.$$
$$\rightarrow (max\{0, x - 1\}, max\{0, x_c - 1\}, x_e), \quad (27)$$
$$with\ rate\ x_c\gamma\ if\ x = x_c + x_e.$$
$$\rightarrow (max\{0, x - 1\}, x_c, x_e), \quad (28)$$
$$with\ rate\ (x_e + x_c)\gamma + (x - (x_c + x_e))\gamma,$$
$$if\ x > (x_c + x_e).$$
$$\rightarrow (min\{n, x + 1\}, x_c, x_e),\ with\ rate\ \tau. \quad (29)$$

The Fig.2 shows the markov chain model of the system with the LMG features and such that g=s=1 and n=3. For this model, the handoff calls dropping probabilities P_d^{LMG} and $\mathbb{P}_b^{LMG}$ are computed on the same states than in the case of MG mechanism. The formulas for dropping and blocking probabilities are:

$$P_d^{LMG} = \sum_{j=0}^{n-g} \sum_{i=j}^{n} \pi_{i,j,i-j}^{LMG}\ and\ P_b^{LMG} = \sum_{j=0}^{n-g} \sum_{k=0}^{g} \sum_{i=j+k}^{n} \pi_{i,j,i-j-k}^{LMG} \quad (30)$$

3 Bounding Systems and Stochastic Comparisons

The goal of this section is to prove that our system (LMG) has a lower handoff call dropping probability and new call blocking probability than MG system. We apply the stochastic comparison method in order to prove the comparison between these performance measures. The stochastic comparison is based on the stochastic ordering theory. Next, we give some definitions and theorems which will be applied for our proof.

3.1 Stochastic Ordering Theory

Let A be a discrete, and countable state space, and $\preceq$ be at least a preorder (reflexive,transitive but not necessarily an anti-symmetric binary relation) on A. We consider two random variables X and Y defined respectively on A. The well-known sample path ordering $\preceq_{st}$ is defined as follows [10]:

Definition 1. $X \preceq_{st} Y \Leftrightarrow E[(f(X))] \leq E[(f(Y))]\ \forall f : A \rightarrow \mathbb{R}^+$, $\preceq$ *−increasing whenever the expectations exist.*

Now we focus on the stochastic comparisons of multidimensional Continuous Time Markov Chains (CTMC)s. Let $\{X_1(t), t \geq 0\}$ (resp. $\{X_2(t), t \geq 0\}$) be a CTMC taking values on A. The stochastic comparison in the sense of $\preceq_{st}$ ordering is defined as follows [10]:

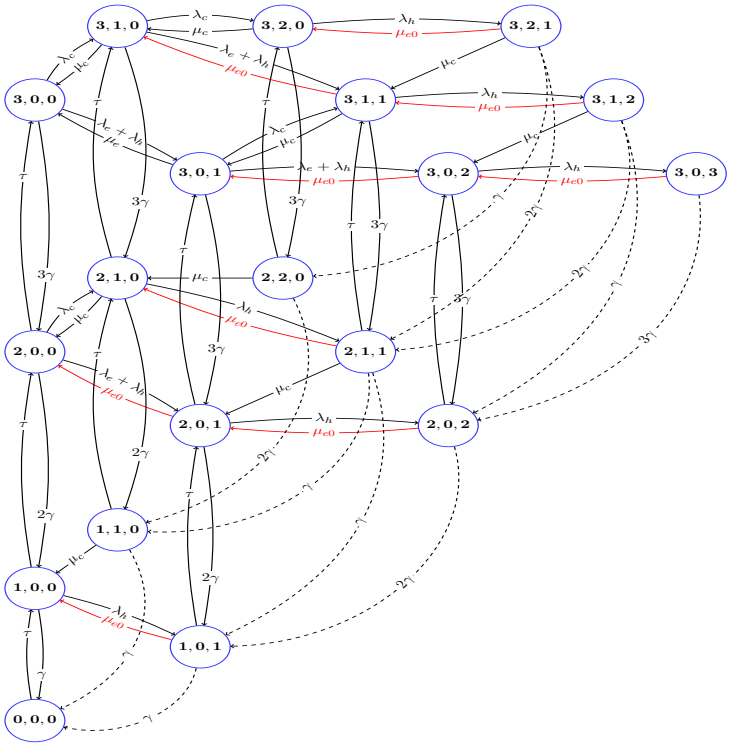


Fig. 2. The LMG Markov Chain model

Definition 2. $\{X_1(t), t \geq 0\}$ *is said to be less in the sense of the stochastic ordering* $\preceq_{st}$ *than* $\{X_2(t), t \geq 0\}$ *written as* $(\{X_1(t), t \geq 0\} \preceq_{st} \{X_2(t), t \geq 0\})$, *if:*

$$X_1(0) \preceq_{st} X_2(0) \Longrightarrow X_1(t) \preceq_{st} X_2(t), \forall t > 0$$

For the stochastic comparison of Markov processes, the coupling method is very used. It is applied for the definition of the $\preceq_{st}$ ordering. This method is equivalent to the definition of a coupled version of the processes in order to compare their sample paths. For the coupling of $\{X(t), t \geq 0\}$ and $\{Y(t), t \geq 0\}$, we define two other Markov chains on A: $\left\{\widehat{X}(t), t \geq 0\right\}$ (resp. $\left\{\widehat{Y}(t), t \geq 0\right\}$) with the same infinitesimal generator then $\{X(t), t \geq 0\}$ (resp. $\{Y(t), t \geq 0\}$). The $\preceq_{st}$ comparison by coupling is established by the following theorem [15]:

Theorem 1. *The following propositions are equivalent:*

1. $\{X(t), t \geq 0\} \preceq_{st} \{Y(t), t \geq 0\}$

2. there exists the coupling $\{(\widehat{X}(t), \widehat{Y}(t), t \geq 0\}$ *such that* $\forall \omega \in \Omega$ *:*

$$\widehat{X}(0)(\omega) \preceq \widehat{Y}(0)(\omega) \Rightarrow \widehat{X}(t)(\omega) \preceq \widehat{Y}(t)(\omega),\ \forall t > 0$$

Next, we apply this theory in order to prove that our mechanism provides lower performance measures.

3.2 Stochastic Comparison of the Mechanisms

As the LMG system is based on the system MG we propose to compare them using stochastic comparisons (it is obvious that MG provides better performances than MAHO system). We denote by $\{X^{MG}(t), t \geq 0\}$ the Markov process representing the MG system, and $\{X^{LMG}(t), t \geq 0\}$ the Markov process representing the LMG system. Let A be the state space of the processes. As the dropping probability is written as the sum of probabilities for states $\mathbf{x} = (x, x_c, x_e) \in A$ such that $x = x_c + x_e$, then we can write it as a reward function on the probability distribution such that the reward equals one for states x such that $x = x_c + x_e$, and 0 for other states. For the blocking probability, then it is also written from a reward function which equals 1 on states such that $x - (x_c + x_e) \leq g$ and 0 for others. As the comparison of these performance measures for MG and LMG systems can be made only if the rewards functions are increasing functions according to the preorder defined on the state space A, then the preorder is such that when the number of free channels decreases, then the states are upper.

For all $\boldsymbol{x}, \boldsymbol{y} \in A$, we define the preorder $\preceq$. So $\boldsymbol{x} \prec \boldsymbol{y}$ if :

$$x - (x_e + x_c) > y - (y_e + y_c)$$

or

$$x - (x_e + x_c) = y - (y_e + y_c)\ \ and\ x_e > y_e,\ x_c > y_c$$

and $\boldsymbol{x} = \boldsymbol{y}$ if $x = y$, and $x_c = y_c$, $x_e = y_e$.

We will prove that :

$$\{X^{LMG}(t), t \geq 0\} \preceq_{st} \{X^{MG}(t), t \geq 0\} \tag{31}$$

We apply Theorem 1, so we define $\{\widehat{X^{LMG}}(t), t \geq 0\}$ (resp. $\{\widehat{X^{MG}}(t), t \geq 0\}$) with the same infini tesimal generator than $\{X^{LMG}(t), t \geq 0\}$ (resp. $\{X^{MG}(t), t \geq 0\}$).

We suppose that : $\widehat{X^{LMG}}(t) \preceq \widehat{X^{MG}}(t)$, and we prove that :

$$\widehat{X^{LMG}}(t + dt) \preceq \widehat{X^{MG}}(t + dt) \tag{32}$$

We consider the different events happening in the systems :

– arrivals in the center of the cell of new calls : as the arrival rates are the same (λ_c), then if we have an arrival in $X^{LMG}(t)$, then we have also an arrival in $X^{MG}(t)$, so inequality (32) is verified. For other kinds of arrivals (in the edge of new calls or handovers) then the order is also verified as the processes work equivalently.

- for the ends of communications in the center of the cell, if we have an end of communication in $X^{MG}(t)$, then we have also an end of communication in $X^{LMG}(t)$ as the processes work equivalently. So inequality (32) is verified.
- for the ends of communications in the edge of the cell or handoff, in the case where $x-(x_e+x_c) > g$, the processes work equivalently and with the same rates : so if these events happen in $X^{MG}(t)$ then they happen also in $X^{LMG}(t)$. In the case where $x-(x_e+x_c) \leq g$, due to the forced handoff, as the total service rate to make free an edge channel is $x_e(\mu_t+\mu_r+\mu_{r0})$ in $X^{LMG}(t)$ so upper than in $X^{MG}(t)$ which is $x_e(\mu_t+\mu_r)$, then the inequality 32 is obviously verified. And intuitively due to the preorder $\preceq$, we see that $X^{LMG}(t)$ has more free channels than $X^{MG}(t)$.
- for the failure rates and reparations, the processes are similar so inequality (32) is verified

So we deduce that equation (31) is verified. And for the stationary probability distributions we have $\Pi^{LMG} \preceq_{st} \Pi^{MG}$. As the handoff-call dropping probabilities of LMG and MG have the same reward functions and are increasing functions according to the preorder $\preceq$ then we deduce that :

$$P_d^{LMG} \leq P_d^{MG}, \text{ and } P_b^{LMG} \leq P_b^{MG} \tag{33}$$

Next, we give some numerical results on the different systems in order to compare the performance measures.

4 Numerical Results

We take the following values for the parameters : the number of channels considered in the cell is $n = 10$, and we denote by λ as the total average arrival rate (the center and the edge) in the cell. λ varies from 10 to 40 calls per minute. We take $\lambda_c = 0.6 * \lambda$, $\lambda_e = 0.4 * \lambda$. For the arrivals of the handovers, we suppose that $\lambda_h = 0.1 * \lambda$. The call holding time is $1/\mu_t = 3$ *minutes*, the channel residency time $1/\mu_r = 1.5$ *minutes*. The rate of failures is $\gamma = 1/hour$, and the repair time $1/\tau$ is 30 minutes. The number of guard channels is $g = 2$. We take two values for the forced handoff rate μ_{r0} : 1 and 2 per minute, in order to see the influence of this parameter.

To show the improvement provided by the new scheme, we compare the performance measures for the three different schemes : MAHO, MG, and LMG.

The numerical values are obtained through Matlab/Simulink simulator. We consider with 95% confidence the indifference region with width 0,01 of the stimated value. In other words, if the estimated value is $\hat{X}$, the exact value X lies in $\hat{X} - 0.01\hat{X} \leq X \leq \hat{X} + 0.01\hat{X}$ with probability 0.95.

In Fig.3 and Fig.4, we compare respectively the blocking probability of the new calls and the dropping handoff calls probability for the different schemes : MAHO, MG, and our scheme LMG.

We observe that MG provides lower handoff call blocking probability than MAHO, but the new call blocking probability is upper because of channels reserved for handoff calls.

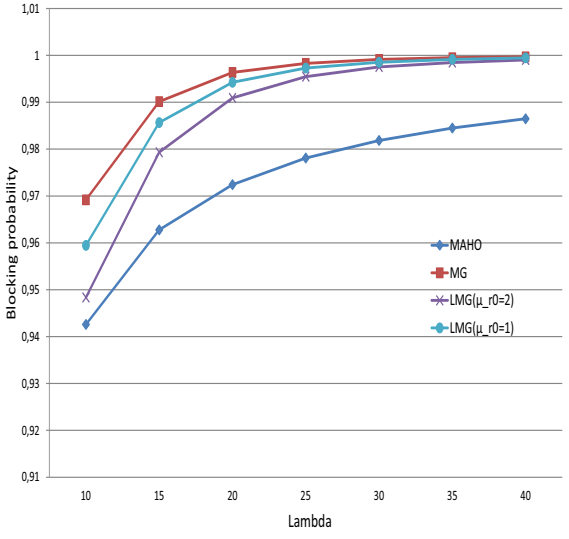


Fig. 3. New call blocking probability

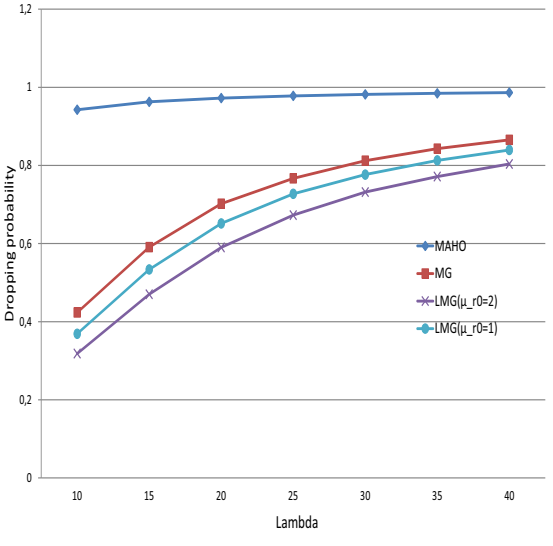


Fig. 4. Handoff call dropping probability

Also, we can see the relevance of the LMG scheme which provides both a lower new call blocking probability than MG, and also a better handoff call dropping probability. This is due to the fact that LMG releases resources by forcing some MT located in the edge of cell to move to the adjacent cells.

We remark also the impact of the forced handoff rate μ_{r0} : we emphasize that when it increases, the performance measures are improved.

5 Conclusion

In this paper, we propose a new mechanism based on a CAC procedure with guard channels and forced handoff in order to improve at the same time the handoff call dropping probability and the new call blocking probability. We apply the stochastic comparisons in order to prove that the performance of our mechanism provides better results than others. We give some numerical results in order to show clearly the relevance of our mechanism.

As a future work, we aim to consider the case of multiple cells in order to obtain a more realistic system with interactions between cells. We could envisage also to adapt LMG to COMB-type networks where users, in the edge of cells, will receive composite signals from a pool of surrounding cells.

References

1. Ahmed, M.H.: Call admission control in wireless networks: a comprehensive survey. IEEE Communication Surveys and Tutorials 7(1), 50–69 (2005)
2. Ekiz, N., Salih, T., Kucukoner, S., Fidanboylu, K.: Overview of handoff techniques in cellular networks. International Journal of Information Technology 2(3), 132–136 (2005)
3. Mouly, M., Paulet, M.B.: The GSM system for mobile communication, M. Mouly, 49 rue Louise Brunner, Palaise, France (1992)
4. Lin, Y.-B., Mohan, S., Noerpel, A.: PCS channel assignmnet strategies for hand-off and initial access. IEEE Personal Comm. 3, 47–56 (1994)
5. Madan, B.B., Dhamaraja, S., Trivedi, K.S.: Combined Guard Channel and Mobile Assisted Handoff for Cellular Networks. IEEE Transactions on Vehicular Technology 57, 502–510 (2008)
6. Acampora, A., Naghshineh, M.: An Architecture and Methodology for Mobile-Executed Handoff in Cellular ATM Networks. IEEE JSAC 12(8), 1365–1375 (1994)
7. Lin, H., Tzeng, S.: Double-Threshold Admission Control in Cluster-based Micro/Picocellular Wireless Networks. In: Proc. IEEE Vehic. Tech. Conf. (VTC 2000-Spring), Tokyo, vol. 2, pp. 1440–1444 (May 2000)
8. Koyuncu, O., Das, S., Ernam, H.: Dynamic Resource Assignment using Network Flows in Wireless Data Networks. In: Proc. IEEE Vehic. Tech. Conf. (VTC 1999), vol. 1, pp. 1–5 (1999)
9. Stewart, W.J.: Introduction to the numerical solution of Markov chains. Princeton University Press (1994)
10. Muller, A., Stoyan, D.: Comparison methods for Stochastic Models and Risks. J. Wiley and son in Probability and Statistics (2002)
11. Mokdad, L., Castel-Taleb, H.: Stochastic comparisons: a methodology for the performance evaluation of fixed and mobile networks. Computer Communications 31(17) (November 2008)
12. Fourneau, J.-M., Pekergin, N.: An Algorithmic Approach to Stochastic Bounds. In: Calzarossa, M.C., Tucci, S. (eds.) Performance 2002. LNCS, vol. 2459, pp. 64–88. Springer, Heidelberg (2002)
13. Castel-Taleb, H., Ismael-Aouled, I., Pekergin, N.: Bounding techniques for transient analysis of G-Networks with catastrophes. In: 5th International ICST Conference on Performance Evaluation Methodologies and Tools, Valuetools 2011. ACM Sigmetrics, May 16-20 (2011)

14. Massey, W.: Stochastic orderings for Markov processes on partially ordered spaces. Mathematics of Operations Research 12(2) (May 1987)
15. Lindvall, T.: Lectures on the coupling method. Wiley series in Probability and Mathematical Statistics (1994)
16. Hong, D., Rappaport, S.S.: Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures. IEEE Transactions on Vehicular Technology 35(3), 77–92 (1986)
17. McMillan, D.: Traffic modeling and analysis for cellular mobile networks. In: Proc. of 13th Int. Teletraffic Congress, pp. 627–632 (1991)
18. Ma, Y., Ro, C.W., Trivedi, K.S.: Performability analysis of channel allocation with channel recovery strategy in cellular networks. In: Proc. of IEEE 1998 (ICUPC 1998), Florence, Italy, October 5-9, pp. 71–75 (1998)
19. Nielsen, T.T., Wigard, J.: Performance enhancements in a frequency hopping GSM network. Kluwer Academic Publishers (2000)

Modeling CSMA/CA in VANET

Anh Tuan Giang and Anthony Busson

Laboratory of Signals and Systems
Université Paris Sud, Supélec, CNRS

Abstract. In this paper, we propose a simple theoretical model to compute the maximum spatial reuse feasible in a VANET. We focus on the ad hoc mode of the IEEE 802.11p standard. Our model offers simple and closed formulae on the maximum number of simultaneous transmitters, and on the distribution of the distance between them. It leads to an accurate upper bound on the maximum capacity. In order to validate our approach, results from the analytical models are compared to simulations performed with the network simulator NS-3. We take into account different traffic distributions (traffic of vehicles) and we study the impact of this traffic on capacity.

1 Introduction

In recent years, Inter-Vehicle Communication (IVC) has become an intense research area, as part of Intelligent Transportation Systems. It assumes that all or a subset of the vehicles is equipped with radio devices, enabling communication between them. Although classical 802.11 can be used for IVC, specific technologies such as IEEE 802.11p [1] (also referred to as Wireless Access in Vehicular Environments, WAVE) have been standardized to support these communications. This standard includes data exchanges between vehicles (ad hoc mode) and between infrastructure and vehicles. When the ad hoc mode is used, the network formed by the vehicles is called a Vehicular Ad hoc NETwork (VANET).

VANET can be used by two families of applications. The first family is user oriented. In this case the VANET may be used to advertise restaurants, gas stations, traffic condition, etc. But the most important applications are related to road safety. Information on road conditions, speed, traffic or alert messages (signalling an accident) may be exchanged in the VANET allowing drivers to anticipate dangerous situations [2]. Data from embedded sensors may also be exchanged in order to increase the perception of the environment. This helps drivers to make appropriate decisions, as it increases the information available on road conditions and traffic situations. The amount of data which can be exchanged between vehicles is thus crucial. Design of these applications has to take into account the limited capacity of the VANET to control the quantity of information which can be sent to other vehicles. In such networks, capacity is mainly limited by the 802.11p spatial reuse. As channels are shared by all the nodes, only a subset of nodes, sufficiently far from each other, can emit at the same time.

In this paper, we evaluate the maximum spatial reuse of the 802.11p technology. Our approach can be presented through a simple example. Let us consider the vehicles depicted in Figure 1. We suppose that we are in a saturated case where all these

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 91–105, 2012.

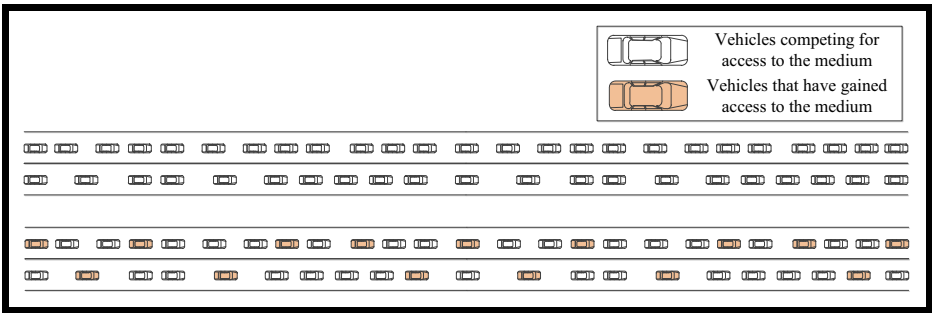


Fig. 1. Example of concurrent transmissions: the 802.11p MAC layer (CSMA/CA) set the rules to access the medium. Only orange vehicles are allowed to transmit frames at the same time.

vehicles wish to send a frame. The MAC layer of the 802.11p standard will select a subset of vehicles which will be allowed to transmit their frames (they are colored in orange in the figure). It selects vehicles in such a way that distances between concurrent transmitters is sufficiently great to avoid interference between the transmissions. At the same time, the capacity is directly related to these distances as they limit the number of simultaneous transmitters. This paper aims to propose a simple model to evaluate the distribution of these distances. We propose a Markovian model where locations of transmitting nodes are built recursively according to the rules used by the 802.11 MAC layer. The equilibrium distribution of this Markov chain allows us to deduce the mean intensity of the concurrent transmitters, i.e. the mean number of transmitting nodes per kilometer. Also, it leads to an estimate of the capacity. The capacity is defined here as the maximum number of frames per second that the network is able to send. Unlike classical approaches dealing with the asymptotic behavior of the capacity, our approach offers accurate estimates of this capacity. Results from the analytical model are then compared to simulations performed with the network simulator NS-3 [3]. We take into account different traffic scenarios (traffic of vehicles). The first scenario assumes that the distance between vehicles is constant and the second one uses a traffic simulator to emulate drivers' behavior on a highway. The combination of NS-3 and the traffic simulator allows us to obtain simulations that are as realistic as possible.

The paper is organized as follows. In Section 2 we present the technological context of this study. Section 3 overviews related works dealing with capacity of ad hoc networks and VANET. Our contributions with regard to the existing approaches are highlighted in the same section. The models are presented in Section 4. Theoretical estimations of the capacity and simulation results are compared in Section 5. We conclude in Section 6.

2 CSMA/CA in 802.11p

The IEEE 802.11p spectrum is composed of six service channels and one control channel. The control channel will be used for broadcast communications dedicated to high priority data and management frames, especially for safety communications. It should

be the privileged channel used to disseminate messages from safety applications. The service channels can be used for safety and service applications, broadcast and unicast communications. The MAC layer in 802.11p is similar to the IEEE 802.11e Quality of Service extension. Application messages are categorized into one of four different queues depending on their level of priority. Each queue uses the classical CSMA/CA (Carrier Sense Multiple Access/Congestion Avoidance) mechanism to access the medium, but CSMA/CA parameters (backoff, etc.) are different from one queue to another in order to favour frames with high priority. In CSMA/CA, a candidate transmitter senses the channel before effectively transmitting. Depending on the channel state, idle or busy, the transmission is started or postponed. *Clear Channel Assessment* (CCA) depends on the MAC protocol and the terminal settings. For the CSMA/CA protocols used in IEEE 802.11, CCA is performed according to one of these three methods.

1. CCA Mode 1: *Energy above threshold*. CCA shall report a busy medium upon detecting any energy above the Energy Detection (ED) threshold. In this case, the channel occupancy is related to the total interference level.
2. CCA Mode 2: *Carrier sense only*. CCA shall report a busy medium only upon the detection of a signal compliant with its own standard, i.e. same physical layer (PHY) characteristics, such as modulation or spreading. Note that depending on threshold values, this signal may be above or below the ED threshold.
3. CCA Mode 3: *Carrier sense with energy above threshold*. CCA shall report a busy medium using a logical combination (e.g. AND or OR) of Detection of a compliant signal AND/OR Energy above the ED threshold.

The CCA mechanism ensures that there is a minimal distance between simultaneous transmitters (except when a collision occurs). If the receiver is in the transmitter radio range, it guarantees a low interference level at the receiver location. Also, it limits the number of simultaneous transmitters in a given area, and thus the number of frames that can be sent per second. Therefore, there is a direct relationship between the spatial reuse imposed by the CCA mechanism and the network capacity.

3 Related Works

A theoretical bound on the capacity of ad hoc networks was initially investigated in [4] where the authors prove that, in a network of n nodes, a capacity of $\Omega\left(\frac{1}{\sqrt{n \cdot \log n}}\right)$ is feasible. In [5], the authors improved this bound and proved that an asymptotic capacity of $\Omega\left(\frac{1}{\sqrt{n}}\right)$ is feasible. In these two articles, the capacity is reached by means of a particular transmission scheduling and routing scheme. In [6] and [7], more realistic link models have been used, both leading to a maximum asymptotic capacity of $O\left(\frac{1}{n}\right)$. In particular, the authors of [7] have shown that when there is a non-zero probability of erroneous frame reception, the cumulative impact of packet losses over intermediate links results in a lower capacity. Finally, it is shown in [5], that when the path-loss

function is bounded, the capacity is also $O\left(\frac{1}{n}\right)$. However these last two results also suppose particular transmission scheduling and routing schemes. Moreover, all these studies deal with the asymptotic behavior of the capacity with regard to the number of nodes and do not propose precise estimates of this capacity.

On the other hand, in CSMA/CA based wireless networks, the transmission scheduling is distributed and asynchronous. It is not planned in advance and depends on the link conditions, interference, etc. at the time a node wants to emit its frame. The number of simultaneous transmitters is thus closely related to the CSMA/CA mechanism which limits the spatial reuse of the channel. The total number of frames sent in the whole network is thus bounded by a constant C whatever the number of nodes and the type of routing schemes. This constant has been evaluated in [8]. Therefore, CSMA/CA multi hop wireless networks would offer a capacity of $O\left(\frac{1}{n}\right)$.

However all these studies focus on networks where nodes are distributed on the plane or in a 2-dimensional observation window. VANETs have very different topologies as the vehicles/nodes are distributed along roads and highways. Radio range of the nodes (about 700 meters with 802.11p in rural environment) being much greater than the road width, we can consider that the topology is distributed on a line rather than in a 2 dimensional space. Lines, grids or topologies composed of a set of lines (to model streets in a city) are thus more appropriate to model VANET topologies.

In [9,10], the authors propose a bound on VANET capacity. They show that when nodes are at constant intervals or exponentially distributed along a line, the capacity is $\Omega\left(\frac{1}{n}\right)$ and $\Omega\left(\frac{1}{n \cdot ln(n)}\right)$ in downtown (city) grids. But it is also an asymptotic bound. Moreover, physical and MAC layers are unrealistic, radio ranges are constant and the same for all the nodes, interference is not taken into account and they assume a perfect transmission scheduling between the nodes. Thus, this bound cannot be applied to 802.11p networks.

In [11], the broadcast capacity of a VANET is estimated. The idea is similar to this paper; an estimation of the number of simultaneous transmitters is proposed. But this evaluation is based on numerical evaluation only, using integer programming.

The contributions of this paper are as follow. We propose two simple models to evaluate the maximum capacity of VANET. The first one, presented in Section 4.1, estimates the number of simultaneous transmitters for the CCA mode 2 of the 802.11. It is based on a existing mathematical model known as *the packing problem*. Since the extension of this model is not tractable for the CCA mode 1, we propose instead a Markovian approach. It is presented in Section 4.2. For this Markov chain, we deduce the transmitter intensity and the mean capacity. Also, we are able to compute the exact distribution of the distance between transmitters. To validate our approach, the theoretical results are compared to realistic simulations performed with NS-3. They focus on the CCA mode 1. Simulations show that our approach is suitable for evaluating the maximum capacity of VANET precisely. It gives precise estimates of CSMA/CA performances, rather than just the asymptotic behaviors, and can consequently be used as a dimensioning or parametrizing tool.

4 Modeling CCA Mode 1 and 2

4.1 Model for CCA Mode 2

When CCA mode 2 is used, the medium is assumed to be busy when a 802.11p frame is detected. This corresponds to cases where the node sensing the medium is at a distance where the signal from the transmitter is detected and compliant to the 802.11 standard. In this case, this approach is rather sensitive to the highest interfering signal rather than the overall interference level. A simple model consists of considering that the maximum distance at which a 802.11 frame is detected is constant. Let R be this distance. The medium is then busy if there is a transmitting node located at a distance less than R. With this model, the problem about the maximum number of simultaneous transmitters comes down to the following question: how many segment with size $2 \cdot R$ can we put in a certain interval $[a, b]$ under the constraint that the centers of these segment cannot be covered by another segment? The answer is simple. If we consider that the first point is located at a, we just have to set a segment at a distance R from the previous one until reaching b. But in a VANET, underlying transmitters are randomly distributed on the line, and transmitters are chosen randomly (it depends on the applications, backoffs, etc.). A more appropriate model consists in placing the segments randomly in $[a, b]$. The first segment is placed uniformly in $[a, b]$. Then, we place the second segment uniformly into all points x of $[a, b]$ such that a segment at x does not cover the center of the previous segment, and so on. The process terminates when there are no gaps in $[a, b]$ large enough to host another segment. This model is referred to as *the packing problem*. A rigorous analysis [12] shows that the mean number of segments divided by the interval length $(b - a)$ tends to a constant $c \approx 0.7476$ when $(b - a) \to +\infty$. The number of simultaneous transmitters with CCA mode 2 can then be estimated as $(b - a)\frac{c}{2 \cdot R}$ for $b - a$ large enough.

4.2 A Markovian Approach for CCA Mode 1

For CCA mode 1, where the sum of signals from all the current transmitters (i.e. Interference) is taken into account, assumptions about radio environment are required to model the signal strengths received from the current transmitters. Interference at a node located at x is generally considered as the sum of all interfering signals:

$$I(x) = \sum_{x_i \in \Phi} l(\|x_i - x\|) \tag{1}$$

where Φ is the set of concurrent transmitters, $\|x_i - x\|$ is the Euclidian distance between the nodes at x and x_i, and $l(.)$ is the path-loss function describing the received signal strength as a function of the distance. The medium is considered idle for a node at x if $I(x) < \theta$ where θ is the Energy Threshold (ED). In this case, the node at x can transmit its frame and becomes a transmitter. An approach similar to the packing problem could be considered in this case. For a given interval $[a, b]$, we sequentially add points uniformly distributed in all points x of $[a, b]$ such that $I(x) < \theta$. But this classical packing approach does not seem tractable. Therefore, we propose a tractable model, based on a Markov chain, to represent transmitters' location.

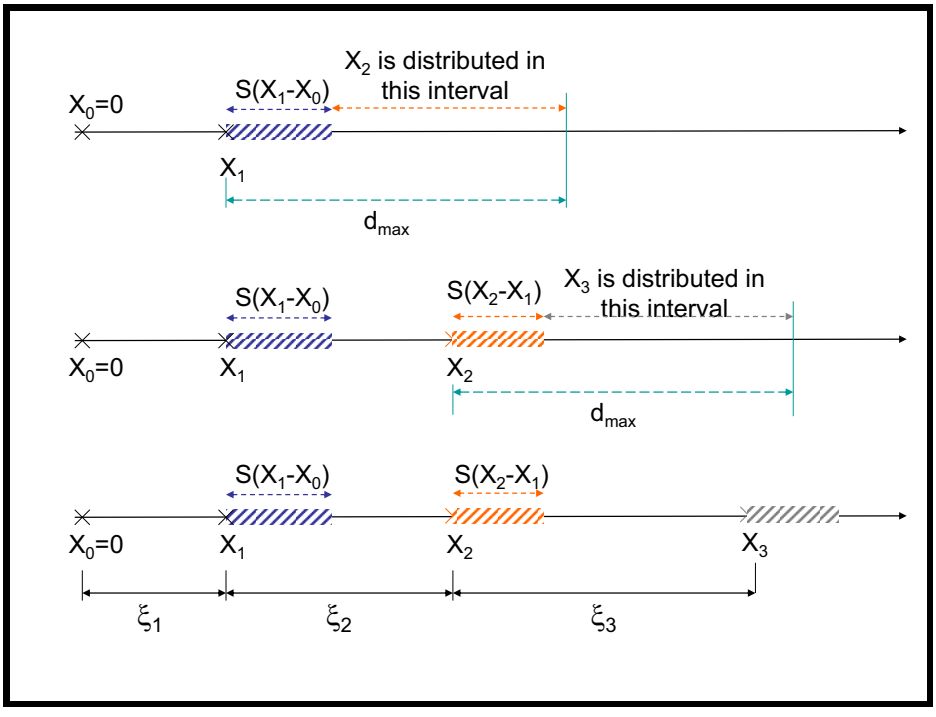


Fig. 2. Notations used in the model. The figure shows how the points X_2 and X_3 are distributed.

This model aims to evaluate the maximum number of simultaneous transmitters in a CSMA/CA network using CCA mode 1. First, we present the different assumptions on the path-loss function and Interference. Then, we define the intervals where the random variables of the Markov chain take their values. In the last paragraph, we present the transition density function and the main results (in Theorem 1).

a) Assumptions. We assume that the medium is detected idle for a node at $X \in I\!R^+$ if:

$$I(X) < \theta \tag{2}$$

where $I(X)$ is the interference at X and θ is the ED threshold (CCA mode 1). Here, $I(X)$ is defined as:

$$I(X) = l(X - L) + l(R - X) \tag{3}$$

where L and R are the locations of the two closest transmitters from X, the closest ones on the left and on the right. Function $l(.)$ is the path-loss function. In our model, Interference is thus computed from the signal strength of the two closest interferers. For the parameters of 802.11p technology, this model is very similar to a model where Interference from all the transmitters is taken into account. Indeed, as there is a significant distance between two successive transmitting nodes (due to the CCA mechanism), Interference generated by distant interferers is negligible with regard to the closest ones

(in 802.11p and in a rural environment, the second interferer in a given direction will be at least 1 km away from the first one).

We assume that the path-loss function verifies the following conditions:

- $l(.)$ is continuous,
- $l(.)$ is a decreasing function,
- $l(0) > \theta$, where θ is a positive constant (the ED threshold),
- $\lim_{d \to +\infty} l(u) = 0$,
- there exists $u \in I\!R^+$ such that $l(u) > \theta$ and $l(v)$ is strictly decreasing and differentiable for all $v \in [u, +\infty)$.

These conditions hold for path-loss functions with the form: $l(u) = P_T \ min(1, c/u^\alpha)$ where P_T is the transmitting power (with $P_T > \theta$), c and d are two positive constants ($c > 0$ and $\alpha > 2.0$).

b) State space of the Markov chain. The chain is denoted $(X_n)_{n \in I\!N}$ with $X_{n-1} < X_n$. It represents the simultaneous transmitters of a CSMA/CA network and consists in a sequence of random points distributed on the line. Since all these transmitters/points have detected the medium idle, Interference at each point X_n must be less than the CCA threshold θ:

$$I(X_n) < \theta \ \ \forall n \geq 0$$

There is thus a minimal distance between the points of the process. We define a function $S(.)$ to describe this distance. According to equation (3) and the CCA condition, $S(u)$ is defined as the solution of

$$l(u) + l(S(u)) = \theta \tag{4}$$

where u corresponds to the distance between the two previous transmitters. X_n is thus distributed in $[X_{n-1} + S(X_{n-2} - X_{n-1}), +\infty]$.

A second assumption allows us to bound this interval. Since we are trying to estimate the maximum number of simultaneous transmitters, we shall distribute the points in such a way that it is not possible to add more points which could detect the medium idle. Consequently, the distance between transmitters must be bound by a maximal distance in order to prevent the presence of intermediate transmitters. Let d_{max} be this distance, it is solution of

$$2 \cdot l\left(\frac{d_{max}}{2}\right) = \theta \tag{5}$$

Thus, each point X_n ($n > 1$) belongs to the interval $[X_{n-1} + S(X_{n-1} - X_{n-2}), X_{n-1} + d_{max}]$. Distances between the transmitters are denoted $\xi_i = X_i - X_{i-1}$.

c) Building the point process. The point process is built as follows. The first two transmitters are located at $X_0 = 0$ and at X_1 with $X_1 \leq d_{max}$ almost surely. Assumptions about the distribution of X_1 are given in the theorem below.

The other points are built recursively. The location of a transmitter X_n ($n > 1$) is distributed in $[X_{n-1} + S(X_{n-1} - X_{n-2}), X_{n-1} + d_{max}]$. For convenience, we consider the sequence $\xi_n = X_n - X_{n-1}$ rather than X_n. ξ_n ($n > 1$) is thus distributed in $[S(\xi_{n-1}), d_{max}]$. It is possible to consider a different distribution on this interval leading to a different density of transmitters. As we do not know a priori the distribution of

the distance between the transmitters, we have considered different distributions. In this paper, only the most accurate distribution, which has been determined by simulations, is presented. This distribution is the linear distribution in $[S(\xi_{n-1}), d_{max}]$. By linear distribution we mean an affine function, positive in $[S(\xi_{n-1}), d_{max}]$, null at d_{max}, and such that its integral on $[S(\xi_{n-1}), d_{max}]$ is 1. The pdf $f_{\xi_n|\xi_{n-1}}(.)$ of $\xi_n = X_n - X_{n-1}$ given $\xi_{n-1} = X_{n-1} - X_{n-2}$ is then:

$$f_{\xi_n|\xi_{n-1}=s}(u) = \Big(\frac{-2}{(d_{max} - S(s))^2}u + \frac{2d_{max}}{(d_{max} - S(s))^2}\Big)1_{u\in[S(s),d_{max}]} \tag{6}$$

where $1_{u\in[S(s),d_{max}]}$ is the indicator function, equals to 1 if $u \in [S(s), d_{max}]$ and 0 otherwise. The sequence $(\xi_n)_{n\geq 0}$ is thus a Markov chain which takes its values in the continuous state space $[S(d_{max}), d_{max}]$. In Figure 2, we present an example of this point process and the different notations. The stationary distribution of this Markov chain is given in the following theorem:

Theorem 1. *The process $(\xi_n)_{n\geq 0}$ defined in this Section is a Markov chain. The stationary distribution of ξ_n is $\pi(s)$ with:*

$$\pi(s) = a \cdot (d_{max} - s)(d_{max} - S(s))^2 1_{s\in[S(d_{max}),d_{max}]} \tag{7}$$

where a is a normalizing factor. The chain $(\xi_n)_{n>0}$ converges in total variation to the distribution $\pi(s)$ for all initial distribution of ξ_1 in $[S(d_{max}), d_{max}]$. If ξ_1 follows the stationary distribution $\pi(.)$ then ξ_n follows the distribution $\pi(.)$ for all n with $n > 0$.

The proof of this theorem is given in the appendix. In the following, we assume that ξ_1 follows the distribution $\pi(.)$. The intensity λ of the point process $(X_n)_{n\in\mathbb{N}}$, i.e. the mean number of point per unit length, is then given by:

$$\lambda = \frac{1}{\mathbb{E}[\xi_1]} = \left(\int_{S(d_{max})}^{d_{max}} s\pi(s)ds\right)^{-1} \tag{8}$$

The inverse of this intensity λ is the mean distance between two consecutive transmitters. Hence, the number of simultaneous transmitters over a road with length d will be $\lambda \times d$. Consequently, the capacity which is defined as the mean number of frames sent per second in the network can be estimated as:

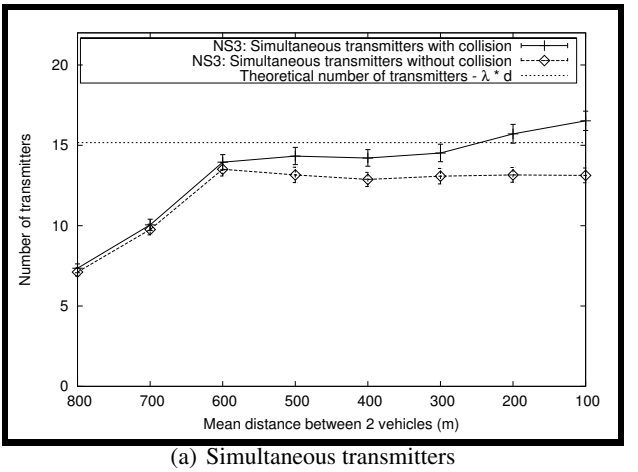
$$Capacity(d) = \frac{\lambda \times d}{T} \tag{9}$$

where λ is the intensity given by equation (8), d is the length of the road and T is the mean time to transmit a frame. This time takes into account the DIFS, the time to transmit the frame, the SIFS and the acknowledgement. We could wonder if it is pertinent to consider the number of transmitted frames rather than the number of received frames for the capacity. In practice, the ED threshold is significantly less than the signal strength required for correct reception. Therefore, when the transmitters respect the CCA rules, Interference does not disturb reception and the number of transmitted frames corresponds to the number of received ones. This will be validated by simulations in the next Section. Our simulations have shown that the only time, frames are not received properly is when collisions occur, i.e. when the CCA rules are not respected.

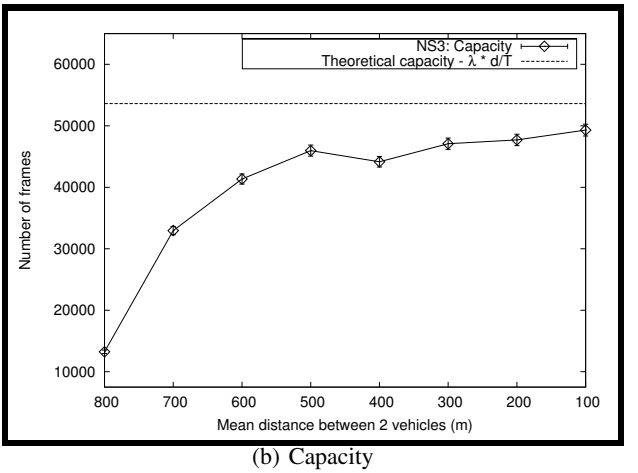
5 Simulations

In this Section, we compare the theoretical evaluation of the capacity to simulations performed with the network simulator NS-3 [3]. In the theoretical model, we consider the path-loss function used in NS-3. We compute for this path-loss the corresponding functions $S(.)$, $\pi(.)$, and the different constants (d_{max}, λ, T, etc.). We compute for all the simulations a confidence interval of 95%. For the simulations, all of the nodes transmit frames to a neighbor with a constant bit rate. All parameters are given in Table 1 and are set according to the IEEE 802.11p standard.

For vehicle locations, we take into account two scenarios: a scenario where the distances between vehicles are constant, and a scenario where vehicle locations are

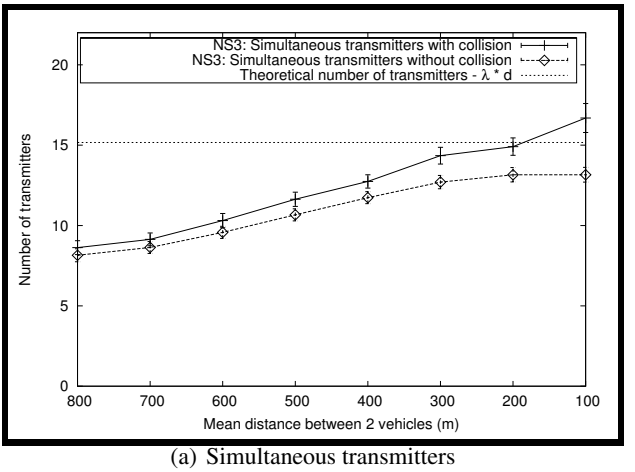


(a) Simultaneous transmitters

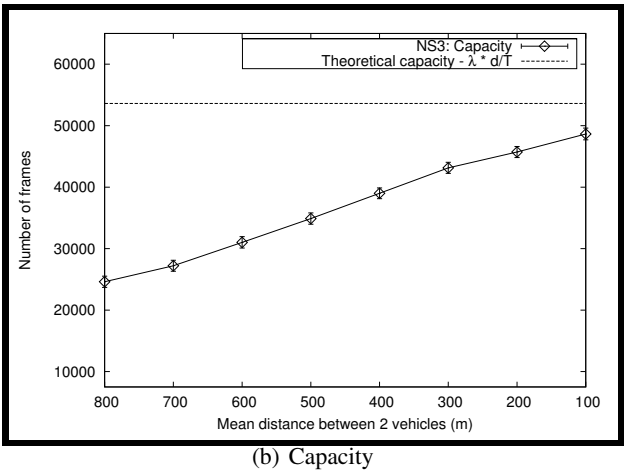


(b) Capacity

Fig. 3. Mean number of simultaneous transmitters and capacity for constant inter-distances



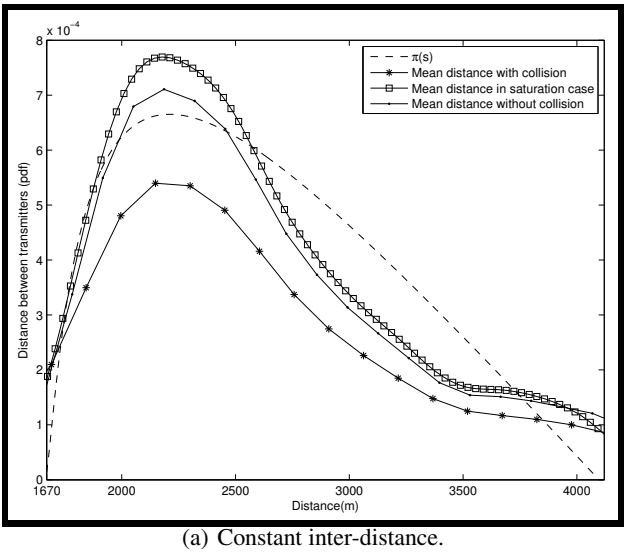
(a) Simultaneous transmitters



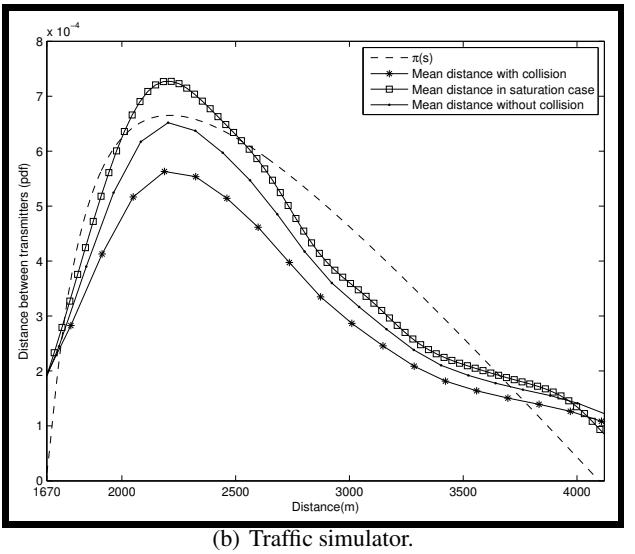
(b) Capacity

Fig. 4. Mean number of simultaneous transmitters and capacity for the traffic simulator

obtained from a realistic traffic simulator. This traffic simulator allows us to faithfully emulate driver behavior. On a highway, driver behavior is limited to accelerating, braking and changing lanes. We assume that there is no off-ramp on the section of highway. A desired speed is associated with each vehicle. It corresponds to the speed that the driver would reach if he was alone in his lane. If the driver is alone (the downstream vehicle is sufficiently far), he adapts his acceleration to reach his desired speed (free flow regime). If he is not alone, he adapts his acceleration to the vehicles around (car following regime). He can also change lanes if the conditions of another lane seem better. All these decisions are functions of traffic condition (speed and distance) and random variables used to introduce a different behavior for each vehicle. This kind of simulation is called micro simulation [13], and the model we used which has been tuned and validated with regard to real data collected on a highway is presented in detail in [14].



(a) Constant inter-distance.



(b) Traffic simulator.

Fig. 5. Distribution of the distances between concurrent transmitters

With the traffic simulator, we simulated a road/highway of 50 km with 2 lanes. The desired speed of the vehicles follows a Normal distribution with mean 120 km/h and standard deviation $\sigma = 10$. The distance shown on the x-axis in the figures corresponds to the mean distance between two successive vehicles.

a) Intensity and capacity results. In Figures 3 and 4 we plotted the mean number of transmitters and the capacity. The different figures correspond to the two kind of traffic: constant inter-distance and trajectories generated by the traffic simulator. It is worth

Table 1. Simulation parameters

Theoretical and NS-3 Parameters	Numerical Values
IEEE 802.11std	802.11p - CCH channel
Path-loss function	$l(d) = P_t \cdot min\left(1, \frac{10^{-4.5677}}{d^3}\right)$
CCA mode	CCA mode 1
ED Threshold (θ)	-82 dBm
Emission power P_t	43 dBm
Number of samples per point	100
Length of the packet	1024 bytes
Duration of the simulation	4 sec
$S(u)$	$\left(2.29 \times 10^{-10} - u^{-3}\right)^{\frac{-1}{3}}$
d_{max}	4120 m
λ	0.379×10^{-3}
DIFS	34 μs
Road length (d)	50 km
SIFS	16 μs

noting that the two traffic distributions (constant and traffic simulator) do not impact the results. This counter intuitive result is explained by the fact that the radio range and detection distance of the 802.11p technology are really greater than the mean distance between nodes. Comments are thus the same for these two traffic scenarios. When we processed the results from the NS-3 simulator, we distinguished transmitters provoking a collision and the ones respecting the CCA rules. When we do not take into account collisions, the theoretical model gives an accurate bound on both intensity and capacity. For the capacity, the difference is only 4% for 10 veh/km (distance between vehicles=100 meters) in Figure 3(b). The theoretical bound is thus approached even for very low density traffic as 10 veh/km corresponds to very sparse traffic. It was difficult to increase this density as the simulated highway is 50km (we already have 500 vehicles when the density is 10 veh/km). When we consider all the transmitters, the transmitters' intensity obtained by simulations exceeds the theoretical one. This is caused by transmitters provoking collisions, which by definition does not respect the CCA rules.

b) Distribution of the distance between transmitters. In Figure 5, we plotted the distributions of the distance between transmitters obtained with NS-3, and the distribution π. The abscissa is $[S(d_{max}), d_{max}]$. The simulated highway is 50 km with 2 lanes and 10 vehicles per kilometer in average. We collected distances between transmitters from 100 samples. For each sample we collected the distances between the transmitters and we plotted the corresponding empirical probability density function. The shape of the distribution for the transmitters without collisions fits very well with the stationary distribution $\pi(.)$. Nevertheless, we can observe a small difference when the function is decreasing. This difference is caused by samples greater than d_{max}. Indeed, it is very difficult to reach the absolute saturation of the network, where the medium is busy at every location, all the time. Therefore, sometimes there are regions where the medium is idle. Even if we simulated an important CBR for each source, nodes do not try to

access the medium all the time because they are in the backoff procedure, they have nothing to send, etc. However if we consider only samples less than d_{max}, we obtain the curve in Figure 5(b). This allows us to estimate the distribution in the saturated case since we neglect the network parts where the medium is idle. It appears that it fits with the theoretical distribution $\pi(.)$ closely. If we compute the mean value of these samples, we obtain a mean inter-distance equal to 2.7 km corresponding to the mean inter-distance proposed in our model (2.64 km). It empirically proves that the theoretical model corresponds to a case where the CCA rule is respected by all the nodes (no collisions), and where the medium is spatially busy. Even if these conditions are not feasible in practice, the proposed Markovian approach still offers accurate bounds on the number of transmitters and capacity of VANET.

6 Conclusion

The particular topology of VANET, where nodes are distributed along a line, allows us to derive a simple model based on the Markov chain. It models distances between concurrent transmitters. Comparisons to realistic simulations show that the model is accurate and that it is quite independent of the traffic distribution. The theoretical intensity of the number of transmitters offers a very good upper bound on capacity, i.e. on the maximum number of frames that can be transmitted per second and per unit length. Our model can be used to tune the CSMA/CA parameters in order to optimize the capacity. Also, the distribution of the distance between two transmitters can be combined to elaborate radio models to evaluate Interference, Bit or Frame Error Rates. In this paper, the path-loss function does not take into account the multipath and fading properties of wireless link. We are currently working on an extension of this model to take into account more elaborate wireless models.

References

1. Fisher, W. (ed.), Armstrong, L. (chair): Status of project ieee 802.11 task group p. wireless access in vehicular environments (wave), http://grouper.ieee.org/groups/802/11/Reports/tgp_update.htm
2. Hartenstein, H., Laberteaux, K.K.: VANET Vehicular Applications and Inter-Networking Technologies. Wiley (2009)
3. Network simulator 3 - ns3, http://www.nsnam.org
4. Gupta, P., Kumar, P.: Capacity of wireless networks. IEEE Transactions on Information Theory 46(2), 388–404 (2000)
5. Franceschetti, M., Dousse, O., Tse, D., Thiran, P.: Closing the gap in the capacity of wireless networks via percolation theory. IEEE Transactions on Information Theory 53(3), 1009–1018 (2007)
6. Dousse, O., Thiran, P.: Connectivity vs capacity in dense ad hoc networks. In: Conference on Computer Communications (INFOCOM), Hong Kong, China. IEEE (March 2004)
7. Mhatre, V., Rosenberg, C., Mazumdar, R.: On the capacity of ad-hoc networks under random packet losses. IEEE Transactions on Information Theory 55(6), 2494–2498 (2009)
8. Busson, A., Chelius, G.: Point processes for interference modeling in csma/ca ad-hoc networks. In: Sixth ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN 2009), Tenerife, Spain (October 2009)

9. Pishro-Nik, H., Ganz, A., Ni, D.: The capacity of vehicular ad hoc networks. In: 45th Annual Allerton Conference on Communication, Control and Computing, Allerton, USA (September 2007)
10. Nekaoui, M., Eslami, A., Pishro-Nik, H.: Scaling laws for distance limited communications in vehicular ad hoc networks. In: IEEE International Conference on Communications, ICC 2008, Beijing, China (May 2008)
11. Du, L., Ukkusri, S., Yushimito Del Valle, W.F., Kalyanaraman, S.: Optimization models to characterize the broadcast capacity of vehicular ad hoc networks. Transportation Research, Part C, Emerging Technologies 17(6), 571–585 (2009)
12. Hall, P.: Introduction to the Theory of Coverage Processes. Wiley (1988)
13. Druitt, S.: An introduction to microsimulation. Traffic Engineering and Control 39(9) (1998)
14. Ahmed, K.I.: Modeling Drivers' Acceleration and Lane Changing Behavior. PhD thesis, Massachusetts Institute of Technology (1999)
15. Diaconis, P., Freedman, D.: On markov chains with continuous state space. Mathematics Statistics Library (501), 1–11 (1995)

Proof. Proof of Theorem 1. First, we prove that if the initial distribution of the Markov chain (the distribution of ξ_1) is π, ξ_n follows the distribution π for all $n > 0$. It suffices to show that π is the stationary distribution for this chain. We need to prove that

$$\pi(s) = \int_{S(d_{max})}^{d_{max}} f_{\xi_n|\xi_{n-1}=y}(s)\pi(y)dy \tag{10}$$

with $\pi(s) = a\left(d_{max} - S(s)\right)^2 (d_{max} - s)$ and $f_{\xi_n|\xi_{n-1}=y}(s)$ given by equation 6.

We get,

$$\int_{S(d_{max})}^{d_{max}} f_{\xi_n|\xi_{n-1}=y}(s)\pi(y)dy$$

$$= \int_{S(d_{max})}^{d_{max}} \left(\frac{-2}{(d_{max} - S(y))^2}s + \frac{2d_{max}}{(d_{max} - S(y))^2}\right) \tag{11}$$

$$\times\, 1_{s\in[S(y),d_{max}]} a(d_{max} - y)(d_{max} - S(y))^2 dy \tag{12}$$

$$= 2a(d_{max} - s)\int_{S^{-1}(s)}^{d_{max}} (d_{max} - y)dy \tag{13}$$

$$= a\left(d_{max} - s\right)\left(d_{max} - S^{-1}(s)\right)^2 \tag{14}$$

where $S^{-1}(.)$ is the inverse function of $S(.)$. This function exists since due to the properties of the function $l(.)$, $S(u)$ is bijective, differentiable and strictly decreasing in $[S(d_{max}), d_{max}]$. To conclude, note that $S^{-1}(x) = S(x)$.

$$a\left(d_{max} - s\right)\left(d_{max} - S^{-1}(s)\right)^2$$
$$= a\left(d_{max} - s\right)\left(d_{max} - S(s)\right)^2 = \pi(s) \tag{15}$$

Also, we prove that ξ_n converges in total variation (it implies convergence in distribution) to π for any initial distribution of ξ_1 in $(S(d_{max}), d_{max}]$. We apply the Theorem 1 in [15] to prove this convergence. Since we have proved that π was the stationary distribution, it suffices to prove that the kernel P of this Markov chain is strongly

$\pi-$irreducible, i.e. $\forall x \in (S(d_{max}), d_{max}]$ and $A \subset [S(d_{max}), d_{max}]$ with $\pi(A) > 0$, there is a positive integer n_{xA} such that $P^n(x, A) > 0 \ \forall n \geq n_{xA}$. In our case, $\pi(A) > 0$ with $A \subset [S(d_{max}), d_{max}]$ is equivalent to $\nu(A) > 0$ where $\nu(.)$ is the Lebesgue measure in $\mathbb{R}^+$. The kernel P describes the transition probabilities, in our case it is formally defined as:

$$P(x, A) = \int_A f_{\xi_2|\xi_1=x}(y)dy \tag{16}$$

with $A \subset [S(d_{max}), d_{max}]$. $P^n(.,.)$ is the distribution of ξ_n $(n > 1)$ given ξ_1. It may be defined recursively:

$$P^n(x, A) = \int_{S(d_{max})}^{d_{max}} P(x, dy)P^{n-1}(y, A) \tag{17}$$

First, note that if $P^m(x, A) > 0$ with $m > 0$, $P^n(x, A) > 0 \ \forall n \geq m$. It can be easily proved by recurrence: Since $P^m(x, A) > 0 \ \forall y \in [S(d_{max}), d_{max}]$ and $P(x, dy) = f_{\xi_2|\xi_1=x}(y)dy$ with $f_{\xi_2|\xi_1=x}(y) > 0 \ \forall y \in [S(x), d_{max}]$, $P^{m+1}(x, A)$ expressed as

$$P^{m+1}(x, A) = \int_{S(d_{max})}^{d_{max}} P(x, dy)P^m(y, A) \tag{18}$$

will be positive if $\nu([S(x), d_{max}]) > 0$, in other words if $x > S(d_{max})$. We prove now that $P^2(x, A)$ for all $x \in [S(x), d_{max}]$ and $A \subset [S(x), d_{max}]$ with $\nu(A) > 0$. n_{xA} can thus be chosen equal to 2. Let $a = min\{u, u \in A\}$,

$$P^2(x, A) = \int_{S(d_{max})}^{d_{max}} P(y, A) f_{\xi_2|\xi_1=x}(y)dy \tag{19}$$

$$\geq \int_{S(min(x,a))}^{d_{max}} P(y, A) f_{\xi_2|\xi_1=x}(y)dy \tag{20}$$

$$> 0$$

Indeed, $P(y, A) > 0$ and $f_{\xi_2|\xi_1=x}(y) > 0$ for all y in $[S(min(x, a)), d_{max}]$. Equation (20) is thus positive when $\nu([S(min(x, a)), d_{max}]) > 0$, i.e. when $x > S(d_{max})$. This proves that the Markov chain is strongly $\pi-$irreducible, and thus μP^n converges in total variation to π when $n \to +\infty$ for any initial distribution μ in $(S(d_{max}), d_{max}]$.

Consolidation and Replication of VMs Matching Performance Objectives

Marco Gribaudo, Pietro Piazzolla, and Giuseppe Serazzi

Dip. di Elettronica e Informazione, Politecnico di Milano,
via Ponzio 34/5, 20133 Milano, Italy
{gribaudo,piazzolla,serazzi}@elet.polimi.it

Abstract. The users of actual computing infrastructures allowing the resource provision (such as clouds) are often asked to decide about the proper amount of equipment (virtual machines, VMs) required to execute their requests while satisfying a set of performance objectives. These types of decisions are particularly difficult since the direct correlation between the resources allocated and the performance offered is influenced by a number of factors such as the characteristic of the different class of requests, the capacity of the resources, the workload sharing the same physical hardware, the dynamic variation of the mix of requests of the different classes in concurrent execution. In this paper we derive the impact on several performance indexes by two popular techniques, namely, consolidation and replication, adopted in virtual computing infrastructures. In particular we present an analytical model to determine the best consolidation or replication options that matches given performance objectives specified through a set of constraints.

1 Introduction

Consolidation and replication techniques are commonly used to manage efficiently large datacenters. According to the former technique, the load of several systems are merged in a reduced number of servers minimizing operational costs. The latter technique partition the load among several physical machines executing replicated applications: in this way, the requests flow each server has to handle is reduced and the performance improved.

Both these techniques have several positive aspects but also they may result in complex management and technical problems that require wide knowledge in several computer science topics to be satisfactorily solved.

While the introduction of virtualization concept alleviated some of the difficulties related to the management of large infrastructures (see, e.g., [16,17]), it also increased the logical distance between the users and the physical resources making more complex the performance forecast. This problem is particularly evident in virtual environments, such as clouds, where users have a limited or no control of the hardware allocated to execute their requests. These drawbacks, coupled with the heterogeneity of the actual workload service demands [8] and the variability of arrival patterns [14] make from a user perspective the matching of its performance expectations a very difficult task.

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 106–120, 2012.

This paper explores the relationships between the resources consolidation/replication actions and the performance experienced by users in systems running mixes of different classes of applications. Indeed, these actions play a fundamental role in determining the overall performance since they have a direct impact on the bottleneck creation and migration.

In the considered infrastructure the subjects of consolidation and replication actions are Virtual Machines (VMs) that users may startup or shutdown. Users provision VMs in a quantity assumed sufficient to satisfy their requirements. The number of instanced VMs has a strong impact on the performance experienced. Under-provisioning will provide unsatisfactory performance, that may lead to violating its expectations, while over-provisioning will result in a waste of money.

We will focus on the forecast of performance resulting from consolidation and replication actions from a user perspective. In particular, we present an analytical technique that allows to determine the optimal consolidation or replication actions to match a user performance objectives subject to a number of constraints.

In the literature, there are several works that deal with the optimal allocation of resources in virtual environments. Several techniques and models focus on database consolidation, some like in [7] by means of workload monitoring for load balancing, others like in [11] using data migration and task scheduling. Other techniques, as in [2,3,10] are aimed to maintain acceptable application performance levels while minimizing the costs of migration/consolidation of resources. Many works propose different approaches to enable autonomic controller to satisfy service level objectives by dynamically provisioning resources, as [6,15,18]. In particular, in [5,13] the dynamic allocation of VMs in cloud environment is described.

The technique proposed in this paper is different from the previous one since we study the impact of consolidation/replication actions on performance indexes subject to constraints considering VMs executing concurrently applications having heterogeneous service demands, i.e., running a multiclass workload. Also, the suggested approach to the identification of the optimal number of VMs that satisfy performance objectives is proactive while the approaches proposed in literature are reactive.

The structure of the paper is as follows. In the next section we describe the technique used to model consolidation and replication actions. In Section 3 we derive the minimum number of replications needed to handle a given multiclass workload, and we extend the methodology in Section 4 to deal with performance constraints. Examples are analyzed in Section 5. Section 6 concludes the paper.

2 The System Model

Consider a system with a multiclass open workload composed by R resources and C customer classes. Let $\mathcal{R} = \{1, 2, ..., R\}$ be the set of resource indexes and $\mathcal{C} = \{1, 2, ..., C\}$ be the set of customer class indexes. The mean service demand of a class c job at resource r is defined as the product of the mean

service time of a class c job for each visit to resource r and the mean number of visits by a class c job to resource r, and it is denoted by a $R \times C$ matrix D^* whose element d^*_{rc} represents the service demand that a class c job requires from the r-th resource, i.e., the mean time required by resource c to its complete execution. The jobs arrive to the system with a global rate Λ, and are subdivided among the classes according to the values β_c, with $\sum_{c=1}^{C} \beta_c = 1$. The vector $\boldsymbol{\beta} = |\beta_1 \dots \beta_C|$, referred to as the *population mix*, allows the definition of the arrival vector $\boldsymbol{\lambda} = |\lambda_1 \lambda_C|$, with $\lambda_c = \Lambda \cdot \beta_c$. We assume that resources can be either *consolidated* or *replicated*. Two resources are consolidated when they are implemented as two different virtual machines on the same physical system. To simplify the presentation, we consider that the services that are consolidated in a single physical machine are the ones of indexes $R-1$ and R.

We assume that the effects of resource consolidation is the sum of the service demands of the two service centers. With this assumption, the matrix $\mathbf{D}^{\#}$ resulting from the consolidation of resources $R-1$ and R has $R-1$ rows; row $R-1$ represents the consolidated resource:

$$\begin{array}{ll} d^{\#}_{r,c} = d^*_{r,c} & \forall c \in \{1,2,...,C\}, \forall r \in \{1,2,...,R-2\} \\ d^{\#}_{R-1,c} = d^*_{R-1,c} + d^*_{R,c} & \forall c \in \{1,2,...,C\} \end{array} \tag{1}$$

With the replication technique a service is deployed through several physical machines reducing the workload each server has to handle. We assume to have m_r instances for each resource r with $m_r \geq 1, \forall r \in \{1,2,...,R\}$. Let m be the total number of service centers that the system with replications will have $m = \sum_{r=1}^{R} m_r$. With these assumptions we have that $m \geq R$. We assume that traffic is equally shared among the m_r instances of the r-th resource. Then, the service demands of a system with replications is described by a matrix $\mathbf{D}$ with m rows and C columns. Rows are partitioned in R groups, each of them composed by m_r identical rows, corresponding to the m_r instances of the r-th resource. The demand associated to each row in a group can be derived from $\mathbf{D}^*$, by considering that each instance of the server has the same demand as the original model, with a visit ratio equal to $\frac{1}{m_r}$. In particular, we can define:

$$\mathbf{D} = \left| \begin{array}{l} m \left\{ \begin{array}{l} m_1 \left\{ \begin{array}{ccc} \frac{d^*_{11}}{m_1} & \cdots & \frac{d^*_{1C}}{m_1} \\ & \vdots & \\ \frac{d^*_{11}}{m_1} & \cdots & \frac{d^*_{1C}}{m_1} \end{array} \right. \\ \vdots \\ m_R \left\{ \begin{array}{ccc} \frac{d^*_{R1}}{m_R} & \cdots & \frac{d^*_{R1}}{m_R} \\ & \vdots & \\ \frac{d^*_{R1}}{m_R} & \cdots & \frac{d^*_{RC}}{m_R} \end{array} \right. \end{array} \right. \end{array} \right| \tag{2}$$

The purpose of consolidation is to reduce the number of physical machines required to handle workloads characterized by very low demands. Replication on

the other hand allows to share requests among several machines to handle very high workloads. It is based on the assumption that a load balancer can equally share the demands among the replicated resources: of course if the load is extremely high, the load balancer becomes the bottleneck of the system, and it must be replicated as well.

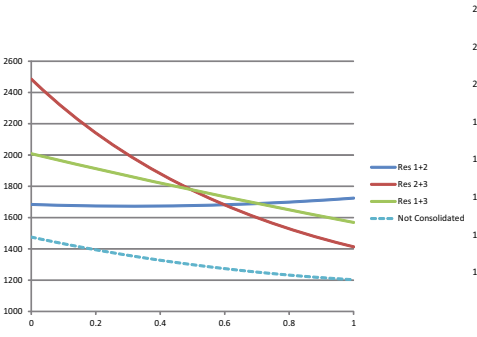
2.1 Example of Consolidation/Replication Actions

Let us consider a system with a multiclass open workload composed of $C = 2$ classes of customers, and $R = 3$ resources. The demand matrix $\boldsymbol{D}^*$ in *msec* is:

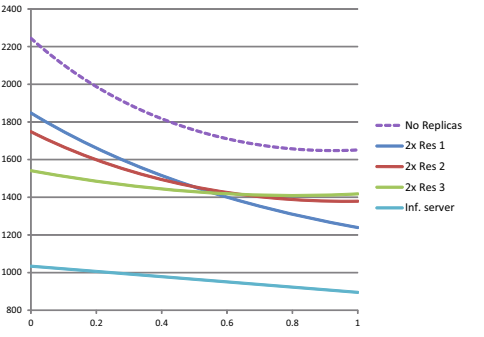
$$\boldsymbol{D}^* = \begin{vmatrix} 391 & 238 \\ 281 & 346 \\ 223 & 450 \end{vmatrix} \tag{3}$$

Depending on the population mix $\boldsymbol{\beta}$, resource 1 or resource 3 can be saturated. The system performs a bottleneck switch at $\beta_1 = 0.5579$ (see [1] for the computation of the bottleneck switching point). This means that with $0 \leq \beta_1 < 0.5579$, resource 1 is the bottleneck of the system, otherwise the bottleneck is resource 3. Using standard queueing theory results (see e.g.,[12,9]), in Figure 1 we plot the system response time as function of the population mix $\boldsymbol{\beta} = |\beta_1 \;\; (1-\beta_1)|$ for different arrival rate Λ and different replication or consolidation patterns. The actual demand matrices, obtained after consolidation and replication are shown in Figure 1(c) and (f) respectively. Figure 1(a) shows the effect of consolidation when the system is lightly utilized, i.e., the global arrival rate is low with respect to the maximum load that the system can handle. This system is stable for all the different population mixes β, and the response time increases with the number of resources consolidated on the same physical machine: this is caused by the fact that no workload partitioning can be applied, which is natural when the server runs on separate hardware. It is interesting to see that the choice of the particular consolidation pattern affects the performance, and that the best choice is function of the population mix. When class 1 jobs are dominant, i.e., ($\beta_1 \approx 1$), then consolidating resource 2 and 3 is the action that gives the best results in terms of response time, with respect of the combination of the other resources, e.g., 1 and 3 or 1 and 2 or 1,2,3. This is behavior is emphasized in Figure 1(b) where the system is unstable when all the resources are consolidated in a single physical resource, and cannot be stable for $\beta < 0.13$ when resource 2 and 3 are consolidated. Indeed, the best choice is always to consolidate the machines that are not bottleneck for a particular population mix $\boldsymbol{\beta}$.

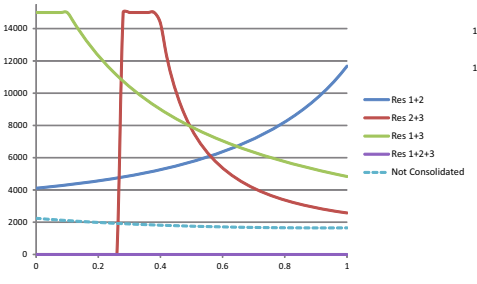
Replication on the other hand reduces the response time. As shown in Figure 1(d), the best choice, again depending on the population mix β, correspond to the replication of the bottleneck resource. Figure 1(e) shows the same replication scheme when the system is very heavily loaded, i.e., the global arrival rate is close to the maximum load that the system can handle. In this case replication can make stable a system otherwise unstable. It is also interesting to see that the replication of resource 2, the one that is never a bottleneck, has the effect of



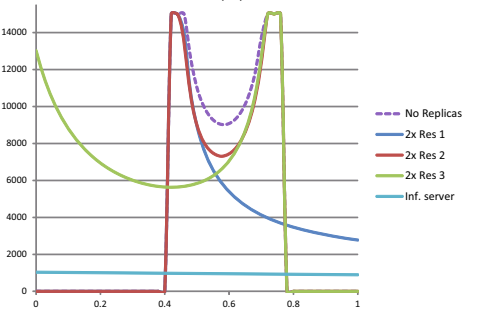
(a)



(d)



(b)



(e)

Consolidations of resources							
Res 1+2		Res 2+3		Res 1+3		Res 1+2+3	
672	584	391	238	614	688	895	1034
223	450	504	796	281	346		

(c)

Replication of resources					
2x Res 1		2x Res 2		2x Res 3	
195.5	119	391	238	391	238
195.5	119	140.5	173	281	346
281	346	140.5	173	111.5	225
223	450	223	450	111.5	225

(f)

Fig. 1. Response time in *msec* vs population mix β_1 of various consolidation and replication configurations of the system described by the demand matrix $\boldsymbol{D}^*$ of Eq. 3 for different arrival rate Λ: (a) consolidation with $\Lambda = 0.008$; (b) consolidation with $\lambda = 0.0014$; (c) service demands for the consolidated cases; (d) replication with $\Lambda = 0.0014$; (e) replication with $\Lambda = 0.0028$; (f) service demands for the replicated cases. NOTE: in (b) and (e) the plotted response time is capped at $15000msec$ in order to avoid out-of-scale values.

reducing the response time when $\beta \in [0.4, 0.8]$, but it does not extend the stability region of the system which remains the same as the one of the non-replicated case. The response time of a replicated system has a lower bound, which can be computed by considering all the resources as infinite server resources. The minimum response time of the infinite server case is also shown in Figure 1(d) and (e) to emphasize the difference between the obtained response time and its lower bound.

2.2 Maximum Consolidation Workload

With a light load, all the resources can be consolidated in a single physical machine. In this case, the demand matrix reduces to a $1 \times C$ vector $\mathbf{D}^K = |d_{1c}^K|$, with each element defined as:

$$d_{1c}^K = \sum_{r=1}^{R} d_{rc}^*, \qquad \forall c \in \{1, 2, ..., C\}. \tag{4}$$

The utilization of the resources is $U^K = \boldsymbol{\lambda}\mathbf{D}^K = \Lambda\boldsymbol{\beta}\mathbf{D}^K$. Since the utilization must be ≤ 1, we can compute the maximum arrival rate that the consolidated system can handle $\Lambda^K(\boldsymbol{\beta})$ as:

$$\Lambda^K(\boldsymbol{\beta}) = \frac{1}{\boldsymbol{\beta}\mathbf{D}^K} = \left(\sum_{c=1}^{C} \beta_c d_{1c}^K \right)^{-1}. \tag{5}$$

In other words, given a population mix $\boldsymbol{\beta}$, all the virtual machines can be consolidate in a single physical machine if $\Lambda < \Lambda^K(\boldsymbol{\beta})$. For this reason $\Lambda_K(\boldsymbol{\beta})$ will be referred as the *maximum consolidation workload*.

2.3 Minimum Number of Physical Machines

Suppose now that we have a high workload Λ for which some resource of the system must be replicated. We can prove that the theoretical minimum number of physical machines required to handle the workload $m_{\min}^T(\boldsymbol{\beta})$ is:

$$m_{\min}^T(\boldsymbol{\beta}) = \left\lceil \frac{\Lambda}{\Lambda_k(\boldsymbol{\beta})} \right\rceil = \left\lceil \Lambda\boldsymbol{\beta}\mathbf{D}^K \right\rceil. \tag{6}$$

Proof. The minimum number of virtual machines $m = \sum_{r=1}^{R} m_r$ required to handle a workload of intensity Λ and population mix $\boldsymbol{\beta}$ must guarantee that the utilization of all the resources is strictly less than one:

$$\sum_{c=1}^{C} \lambda_c \frac{d_{rc}^*}{m_r} < 1 \qquad \forall 1 \leq r \leq R. \tag{7}$$

from which we can compute m_r:

$$m_r > \sum_{c=1}^{C} \lambda_c d_{rc}^* = \Lambda \sum_{c=1}^{C} \beta_c d_{rc}^*. \tag{8}$$

If we apply the definition of m we obtain:

$$m = \sum_{r=1}^{R} m_r > \sum_{r=1}^{R} \Lambda \sum_{c=1}^{C} \beta_c d_{rc}^* = \Lambda \sum_{c=1}^{C} \beta_c \sum_{r=1}^{R} d_{rc}^* = \Lambda \sum_{c=1}^{C} \beta_c d_{1c}^K = \frac{\Lambda}{\Lambda_k(\boldsymbol{\beta})} \tag{9}$$

If we consider that the minimum number of physical machines should be an integer, and we round m to the closest higher integer, we obtain the definition of $m_{\min}^T(\boldsymbol{\beta})$ give in Eq. 6.

The theoretical minimum requires the replication of a server that consolidates all the resources, which can be unpractical. If we require that each resource holds at most one service, then the theoretical minimum is just a lower bound to the actual minimum, which could be a little bit higher. In Section 3 the actual minimum will be considered, and in Section 5 both cases will be compared.

3 Computing the Best Number of Replications

We want to study the system as the arrival rate increases. We can express the number of servers for resource r as a fraction of the total number of allocated servers m. In particular we define:

$$\gamma_r = \frac{m_r}{m}, \qquad \text{with: } \sum_{r=1}^{R} \gamma_r = 1. \tag{10}$$

Let $\boldsymbol{\gamma} = |\gamma_1 \dots \gamma_R|$ be the vector representing the *instances mix*. As we have seen, in order to maintain the system stable, the number of VMs must grow accordingly to the increased arrival rate. In particular, reversing the definition of $\Lambda^K(\boldsymbol{\beta})$ given in Eq. 5, we may express the total arrival rate as a function of m (the total number of VMs):

$$\Lambda = m\Lambda^K(\boldsymbol{\beta}). \tag{11}$$

We can thus define the *stability condition*, that is the condition of the system in which the utilization of all the resources should be strictly less than one:

$$\max_r \left\{ \sum_{c=1}^{C} \frac{\Lambda \beta_c d^*_{rc}}{m_r} \right\} = \max_r \left\{ \frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \sum_{c=1}^{C} \beta_c d^*_{rc} \right\} < 1. \tag{12}$$

The best allocation strategy would saturate all the available physical machines, raising their utilization to 1. In other words, it will be:

$$\forall r \in \{1, 2, ..., R\} : \frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \sum_{c=1}^{C} \beta_c d^*_{rc} = 1. \tag{13}$$

From Eq. 13, we can then compute γ_r:

$$\gamma_r = \Lambda^K(\boldsymbol{\beta}) \sum_{c=1}^{C} \beta_c d^*_{rc}. \tag{14}$$

It can be easily proven that with the definition given in Eq. 14 is consistent with the definition of γ_r, that is that $\sum_{r=1}^{R} \gamma_r = 1$.

Proof. If we sum the γ_r for all the resources we obtain:

$$\sum_{r=1}^{R} \gamma_r = \sum_{r=1}^{R} \Lambda^K(\boldsymbol{\beta}) \sum_{c=1}^{C} \beta_c d^*_{rc} = \Lambda^K(\boldsymbol{\beta}) \sum_{c=1}^{C} \beta_c \sum_{r=1}^{R} d^*_{rc} = \Lambda^K(\boldsymbol{\beta}) \frac{1}{\Lambda^K(\boldsymbol{\beta})} = 1 \tag{15}$$

Eq. 6 and Eq. 14 are very important, because they tell us how many virtual machines m should be provisioned, and which fraction of these machines should be used to host a particular service r, to be able to serve an input workload of intensity Λ distributed according to a given population mix $\boldsymbol{\beta}$. In particular, inserting Eq. 6 in 14, we can obtain:

$$m_r = \lceil m\gamma_r \rceil = \left\lceil \Lambda \sum_{c=1}^{C} \beta_c d_{rc}^* \right\rceil . \tag{16}$$

We can use the results from Eq. 16 to compute the actual minimum number of physical machines $m_{\min}^A(\boldsymbol{\beta})$ required to handle a workload Λ as:

$$m_{\min}^A(\boldsymbol{\beta}) = \sum_{r=1}^{R} m_r. \tag{17}$$

Note that by definition, we have that $m_{\min}^T(\boldsymbol{\beta}) \leq m_{\min}^A(\boldsymbol{\beta})$, but the relative difference between $m_{\min}^T(\boldsymbol{\beta})$ and $m_{\min}^A(\boldsymbol{\beta})$ tends to 0 as Λ tends to infinity.

4 Matching Performance Objectives

The technique presented in Section 3 can be extended to take into account *Performance Constraints* (PCs). In particular, m and $\boldsymbol{\gamma}$ can be computed to not only guaranty stability, but also to ensure that a given set of constraints are respected. Several PCs can be defined by a user in order to match his own expectations or objectives. In this paper we will focus on requirements concerning: the utilization of a resource r by a class c, the utilization of a resource r, the mean residence time of a resource r, and the mean system response time.

4.1 Constraints on the Utilization of a Class in a Resource

In Eq. 12, the parameters m and $\boldsymbol{\gamma}$ were compute to make the system stable. If instead of saturating all the resource, we want to limit the utilization of the class c at station r to a value $0 \leq u_{rc} < 0$ (with $\sum_{c=1}^{C} u_{rc} \leq 1, \forall r$), Eq. 12 becomes:

$$\frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \beta_c d_{rc}^* < u_{rc}. \tag{18}$$

Eq. 18 should be valid for all the classes c. We can thus find the minimum value of γ_r that satisfy the PCs on the utilization for all the classes as:

$$\gamma_r = \max_c \left\{ \frac{\Lambda^K(\boldsymbol{\beta})}{u_{rc}} \beta_c d_{rc}^* \right\}. \tag{19}$$

In this case however, we can have that $\sum_{r=1}^{R} \gamma_r > 1$. The number of replicas m_r for the r-th resource can be computed exactly as in Eq. 16:

$$m_r = \lceil m\gamma_r \rceil = \left\lceil \frac{\Lambda\gamma_r}{\Lambda_k(\boldsymbol{\beta})} \right\rceil . \tag{20}$$

The minimum number of servers m that respect the PCs $m_{\min}^{\text{PCs}}(\boldsymbol{\beta})$ can thus be computed as follows:

$$m_{\min}^{\text{PCs}}(\boldsymbol{\beta}) = \sum_{r=1}^{R} \left\lceil \frac{\Lambda \gamma_r}{\Lambda_k(\boldsymbol{\beta})} \right\rceil . \tag{21}$$

4.2 Constraints on the Total Utilization of a Resource

Suppose instead that we want to limit the total utilization of a resource r to be at most u_r, and to be equally shared among the classes. In this case we will have that:

$$\frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \sum_{c=1}^{C} \beta_c d_{rc}^* < u_r, \tag{22}$$

from which we can easily determine γ_r:

$$\gamma_r = \frac{\Lambda^K(\boldsymbol{\beta})}{u_r} \sum_{c=1}^{C} \beta_c d_{rc}^*. \tag{23}$$

The same considerations given in Section 4.1 about the possibility of having $\sum_{r=1}^{R} \gamma_r > 1$ and its implications are also valid for this PC and for the ones considered in the following sections.

4.3 Constraints on Mean Resource Residence Time

Now, let us consider a PC that imposes that the mean residence time of a resource should be less than a given ϑ_r. Using the standard queueing theory results, we can formulate this requirement as:

$$m_r \sum_{c'=1}^{C} \frac{\lambda_{c'}}{\Lambda} \frac{\frac{d_{rc'}^*}{m_r}}{1 - \frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \sum_{c=1}^{C} \beta_c d_{rc}^*} = \frac{\sum_{c=1}^{C} \beta_c d_{rc}^*}{1 - \frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \sum_{c=1}^{C} \beta_c d_{rc}^*} < \vartheta_r, \tag{24}$$

Note that, since the resource is split in m_r replicas, we have to consider the sum of the residence time at all the replicas, and this is why the first member on the left hand side of Eq. 24 is multiplied by m_r. We may compute γ_r, as:

$$\gamma_r = \frac{\vartheta_r \Lambda^K(\boldsymbol{\beta}) \sum_{c=1}^{C} \beta_c d_{rc}^*}{\vartheta_r - \sum_{c=1}^{C} \beta_c d_{rc}^*}. \tag{25}$$

4.4 Constraints on the System Response Time

If we require that the mean system response time should be less than a given threshold ϑ, we can express this constraint as:

$$\sum_{r=1}^{R} m_r \sum_{c'=1}^{C} \frac{\lambda_{c'}}{\Lambda} \frac{\frac{d^*_{rc'}}{m_r}}{1 - \frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \sum_{c=1}^{C} \beta_c d^*_{rc}} = \sum_{r=1}^{R} \frac{\sum_{c=1}^{C} \beta_c d^*_{rc}}{1 - \frac{\Lambda^K(\boldsymbol{\beta})}{\gamma_r} \sum_{c=1}^{C} \beta_c d^*_{rc}} < \vartheta, \tag{26}$$

The previous Eq. has infinite solutions in γ_r. Determining the optimal value (i.e., the one that minimizes the total number of physical machines), requires the solution of a non-linear optimization problem. We can however very easily compute one of the solutions (which might be sub-optimal). If we define γ_r as:

$$\gamma_r = \frac{1}{\alpha} \Lambda^K(\boldsymbol{\beta}) \sum_{c=1}^{C} \beta_c d^*_{rc}. \tag{27}$$

then Eq. 26 becomes:

$$\sum_{r=1}^{R} \sum_{c=1}^{C} \beta_c d^*_{rc} \frac{1}{1-\alpha} = \frac{1}{\Lambda^K(\boldsymbol{\beta})(1-\alpha)} < \vartheta, \tag{28}$$

we can compute α:

$$\alpha = \frac{1}{\vartheta} \left(\vartheta - \frac{1}{\Lambda^K(\boldsymbol{\beta})} \right) \tag{29}$$

from which we derive:

$$\gamma_r = \frac{\vartheta \Lambda^K(\boldsymbol{\beta}) \sum_{c=1}^{C} \beta_c d^*_{rc}}{\vartheta - \frac{1}{\Lambda^K(\boldsymbol{\beta})}}. \tag{30}$$

5 Examples

In this section we will first show how the Equations given in Section 4 can be used to properly choose the number of replicas required to handle a given workload while respecting a set of PCs. We then apply the proposed results on a test system to investigate if is it better to replicate consolidated services, or to replicate separate services. The test system was simulated using the JMT tool [4]: confidence interval at 99% were evaluated, but only mean values are shown to simplify the presentation.

5.1 Sizing a System

Let us consider the three-tier system with $C = 2$ classes, and $R = 3$ resources, characterized by the demand matrix $\boldsymbol{D}$ of Eq. 3. Suppose that the utilization of each resource r for each class c must be less than the following given values u_{rc}:

$$|u_{rc}| = \begin{vmatrix} 0.5 & 0.5 \\ 0.3 & 0.7 \\ 0.65 & 0.35 \end{vmatrix}. \tag{31}$$

If we apply the results presented in Section 4.1, we obtain that the number of replica for each service m_r should be $|m_r| = |1 \;\; 1 \;\; 3|$. In the following, to simplify the notation, we will denote this particular configuration of replicas simply as 1-1-3. Figure 2(a) shows the utilization of all the combination of classes and resources for the 1-1-2 and the 1-1-3 configurations, together with the target value required by the PC. As it can be seen, the 1-1-3 configuration respects all the PCs, while the 1-1-2 violate the constraint on the second class for the third resource (C2-R3), where the utilization is about 40% and the requirement should be less than 35%. Next we put the requirement on the utilizations of single resources to be less than u_r defined as $|u_r| = |0.5 \;\; 0.3 \;\; 0.8|$. Using the results presented in Section 4.2, we can see that at least 2-4-2 replicas are required to satisfy the PCs. In Figure 2(b) we present the utilization of the resources for three configurations: 2-2-2, 2-3-2 and 2-4-2. Clearly the 2-2-2 configuration violates the PC on the second resource. At first sight, the 2-3-2 configuration would seem to be adequate to satisfy all the constraints. However, at a closer look, we can see that with this configuration the utilization of the second resource would be 30.26%, slightly higher than the 30% required by the PC. Constraints on the response time of the single resources is considered in Figure 2(c), where there PCs are set according to the following values, in *msec*: $|\vartheta_r| = |1000 \;\; 500 \;\; 1050|$. In this case, applying the results presented in Section 4.3, we have that minimum configuration should have at least 2-3-2, and a 2-2-2 configuration will violate the PC on the second resource. Finally, we consider the system response time and we use the expression presented in Section 4.4. In particular we examine a series of possible constraints, $\vartheta = 14000msec$, $\vartheta = 8200msec$, $\vartheta = 4200msec$, $\vartheta = 2200msec$, $\vartheta = 1200msec$ and we compute the configuration required to obtain such system response time. They are 1-1-2, 1-2-2, 2-2-2, 2-2-2 and 5,6,6 respectively. Figure 2(d) shows that by using the proposed configuration the system response time is always lower than the requirement. However, since the Equations given in Section 4.4 compute only a sub-optimal solution, there could be cases where the constraint can be met with a smaller set of machines. In this case this happens for the $\vartheta = 4200msec$ constraint, which is met not only for the 2-2-2 configuration (the one computed by Eq. 30), but also for the 1-2-2 configuration that uses one machine less.

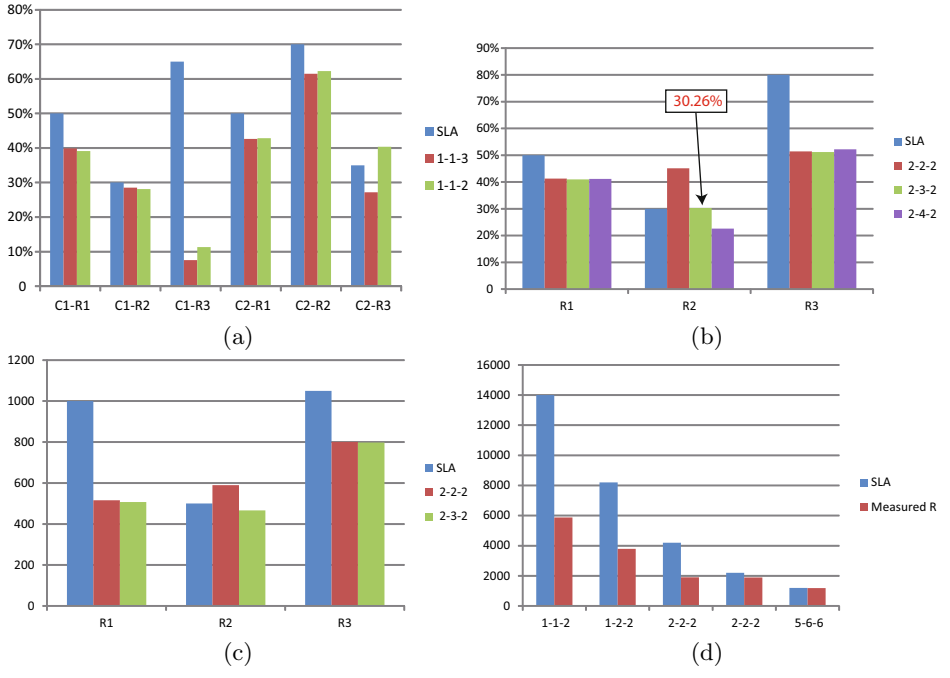


Fig. 2. Minimum number of physical machine to handle a workload arrival rate $\Lambda = 0.0028$ and population mix $\boldsymbol{\beta} = |0.36, 0.64|$ with the following constraints on the: (a) utilization of a resource r by a class c; (b) total utilization of a resource r; (c) response time of resource r; (d) system response time. The last two are expressed in *msec*.

5.2 Replication of Consolidated Servers vs. Replication of Single Servers

In Eq. 6 and Eq. 17 we have seen that there are two minimum number of machines that can be computed from the definition of a system: the theoretical minimum $m^T_{\min}(\boldsymbol{\beta})$, and the actual minimum $m^A_{\min}(\boldsymbol{\beta})$. In principle it would be desirable to use $m^T_{\min}(\boldsymbol{\beta})$, since it is always smaller. Such minimum could be achieved by first consolidating all the servers on a single resource, and then by replicating the consolidated resource as much as required. This solution can be unpractical since it would require to install many services on a single physical system, making it more complex to maintain, and more resource consuming. However, if we do not consider the practical issues, and we focus on the performance of the consolidated system, will a system where we replicate a consolidated server be better than a system where we replicate the single services? To investigate this question, we use a simple demand matrix with values in *msec*:

$$\boldsymbol{D}^* = \begin{vmatrix} 200 & 100 \\ 100 & 100 \\ 100 & 300 \end{vmatrix}, \tag{32}$$

that can be consolidated as: $\boldsymbol{D}^K = |400 \quad 500|$. This matrix has been chosen to have the values computed by Eq. 16 integer for $\Lambda = 0.04$, to avoid the effects of the smallest higher integer operator. We consider however two slightly smaller workload intensities with $\Lambda = 0.038$ and $\Lambda = 0.035$ respectively. Figure 3 shows

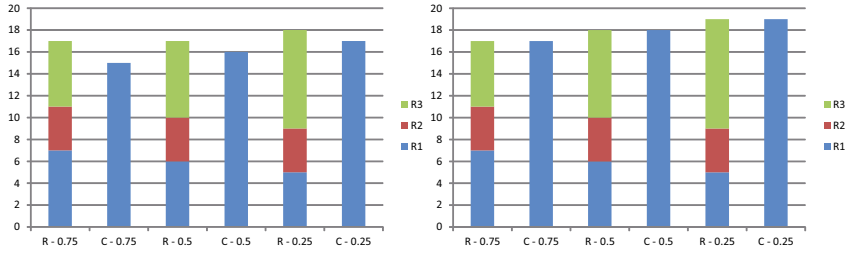


Fig. 3. Minimum number of physical machines for the consolidated or the replicated case. Arrival rates: (a) $\Lambda = 0.035$; (b) $\Lambda = 0.038$.

the minimum number of machines required in both cases. As we can see, for the highest load (Figure 3(b)), both the pure replicated and the consolidated case have the same number of machines. With a smaller workload however, Figure 3(a), the replicated case is able to handle the same rate of requests with a few (one or two) machines less. Let us focus on the utilization first (Figure 4). In the consolidated case the system is always fully utilized (always greater than 95%), while in the replicated case, except for the bottleneck resource, the other are less congested. Response time (Figure 5) is generally lower for the pure

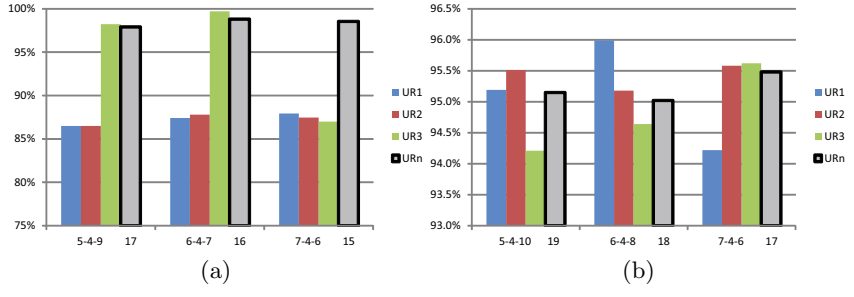


Fig. 4. Utilization of the resources for the consolidated or the replicated case. Arrival rates: (a) $\Lambda = 0.035$; (b) $\Lambda = 0.038$.

replication case when the workload is smaller ($\Lambda = 0.035$), and it becomes larger when the system is closer to saturation ($\Lambda = 0.038$). The only exception is for $\boldsymbol{\beta} = |0.5 \quad 0.5|$, where the system experience high response times even for the pure replication case with $\Lambda = 0.0035$: this however is due to the third resource which is utilized more than 99% as can be seen from Figure 4(a).

So, to summarize the results, it would seem that from a pure performance point of view, the replication of the consolidated server will always perform better with respect to the pure replication case, allowing the user to either use a smaller number of machines, or to experience shorter response time with the same number of machines. However, since replication of consolidated machines cannot be as practical as replication of servers, we can observe that the decrease in performance of the latter is not so significant, and the difference between the two solutions tends to decrease as the total number of machines increases.

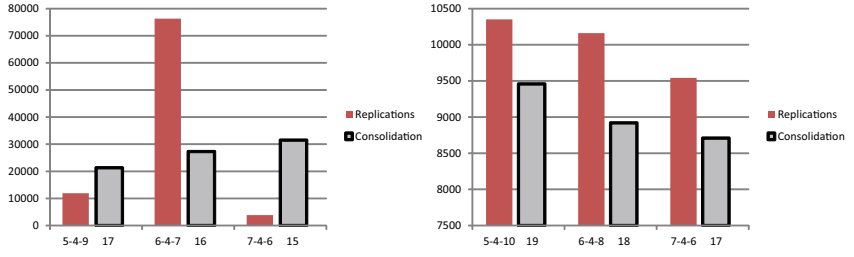


Fig. 5. System response time for the consolidated or the replicated cases with arrival rate $\Lambda = 0.035$ (a) and $\Lambda = 0.038$ (b)

6 Conclusions

In this paper we have considered the topic of consolidation and replication from a end-user point of view. In particular we have proposed very simple expressions that can be used to predict the effect of consolidation, and to appropriately dimension a system, in terms of replication of service, to match a given set of performance objectives.

Future works will address more complex performance objectives, and will consider more complex types of resources, to better capture the internal parallelization characteristics related to multi-core and multi-threaded resources.

References

1. Balbo, G., Serazzi, G.: Asymptotic analysis of multiclass closed queueing networks: Common bottlenecks. Performance Evaluation 26(1), 51–72 (1996)
2. Benevenuto, F., Fernandes, C., Santos, M., Almeida, V., Almeida, J., Janakiraman, G(J.), Santos, J.R.: Performance Models for Virtualized Applications. In: Min, G., Di Martino, B., Yang, L.T., Guo, M., Rünger, G. (eds.) ISPA 2006. LNCS, vol. 4331, pp. 427–439. Springer, Heidelberg (2006)
3. Bennani, M., Menascé, D.: Resource allocation for autonomic data centers using analytic performance models. In: Autonomic Computing, ICAC 2005, pp. 229–240 (June 2005)

4. Bertoli, M., Casale, G., Serazzi, G.: Java modelling tools: an open source suite for queueing network modelling and workload analysis. In: Proc. of the 3rd Conf. on Quantitative Evaluation of Systems (QEST), pp. 119–120. IEEE (2006)
5. Bobroff, N., Kochut, A., Beaty, K.: Dynamic placement of virtual machines for managing sla violations. In: 10th IFIP/IEEE International Symposium on Integrated Network Management, IM 2007, pp. 119–128 (21, 2007-yearly 25, 2007)
6. Bushehrian, O.: The Application of FSP Models in Automatic Optimization of Software Deployment. In: Al-Begain, K., Balsamo, S., Fiems, D., Marin, A. (eds.) ASMTA 2011. LNCS, vol. 6751, pp. 43–54. Springer, Heidelberg (2011)
7. Curino, C., Jones, E.P., Madden, S., Balakrishnan, H.: Workload-aware database monitoring and consolidation. In: Proc. of the International Conference on Management of Data, SIGMOD 2011, pp. 313–324. ACM, New York (2011)
8. Ganapathi, A., Chen, Y., Fox, A., Katz, R., Patterson, D.: Statistics-driven workload modeling for the cloud. In: 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW), pp. 87–92 (March 2010)
9. Jackson, J.R.: Jobshop-like queueing systems. Management Science 10(1), 131–142 (1963)
10. Khanna, G., Beaty, K.A., Kar, G., Kochut, A.: Application performance management in virtualized server environments. In: NOMS, pp. 373–381 (2006)
11. Kokkinos, P., Christodoulopoulos, K., Kretsis, A., Varvarigos, E.: Data consolidation: A task scheduling and data migration technique for grid networks. In: Proc. of the 8th IEEE Int. Symposium on Cluster Computing and the Grid, pp. 722–727. IEEE Computer Society, Washington, DC (2008)
12. Lazowska, E.D., Zahorjan, J., Graham, G.S., Sevcik, K.C.: Quantitative System Performance. Prentice-Hall (1984)
13. Menascé, D.A.: Virtualization: Concepts, applications, and performance modeling (2005)
14. Mi, N., Casale, G., Cherkasova, L., Smirni, E.: Sizing multi-tier systems with temporal dependence: benchmarks and analytic models. J. Internet Services and Applications 1(2), 117–134 (2010)
15. Padala, P., Shin, K.G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., Salem, K.: Adaptive control of virtualized resources in utility computing environments. In: Proc. of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems, EuroSys 2007, pp. 289–302. ACM, New York (2007)
16. VirtualBox, http://www.virtualbox.org
17. VMware, http://www.vmware.com
18. Watson, B.J., Marwah, M., Gmach, D., Chen, Y., Arlitt, M., Wang, Z.: Probabilistic performance modeling of virtualized resource allocation. In: Proc. of the 7th International Conference on Autonomic Computing, ICAC 2010, pp. 99–108. ACM, NY (2010)

Analysis of a Discrete-Time Queue with Geometrically Distributed Service Capacities

Herwig Bruneel, Joris Walraevens, Dieter Claeys, and Sabine Wittevrongel

Department of Telecommunications and Information Processing (TELIN),
Ghent University,
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
{hb,jw,dclaeys,sw}@telin.ugent.be

Abstract. We consider a discrete-time queueing model whereby the service capacity of the system, i.e., the number of work units that the system can perform per time slot, is variable from slot to slot. Specifically, we study the case where service capacities are independent from slot to slot and geometrically distributed. New customers enter the system according to a general independent arrival process. Service demands of the customers are i.i.d. and arbitrarily distributed. For this (non-classical) queueing model, we obtain explicit expressions for the probability generating functions (pgf's) of the unfinished work in the system and the queueing delay of an arbitrary customer. In case of geometric service demands, we also obtain the pgf of the number of customers in the system explicitly. By means of some numerical examples, we discuss the impact of the service process of the customers on the system behavior.

Keywords: Discrete-time queueing model, Variable service capacity, Analytic study, Closed-form results.

1 Introduction

In classical queueing models, it is generally assumed that some kind of customers require some kind of service from a given service facility, containing one or multiple servers, which are each able to provide service to one customer at a time. A large body of work exists especially on the analysis of *single-server* queues with various types of arrival processes, service times, queueing disciplines, storage capacities, etc., both in continuous-time and (to a lesser extent) discrete-time settings. The study of *multiserver* queues, however, has traditionally received much less attention, one important reason being that multiserver queues are notoriously hard to analyze.

An important subclass of queueing models consists of queues with *server interruptions*, whereby these server interruptions may be either driven by external processes (such as breakdowns or higher-priority customers demanding service) or they may correspond to vacations deliberately taken by the servers (e.g. when the system becomes empty after a busy period). Here, again, the case of one single server subject to interruptions has been the most commonly studied setting,

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 121–135, 2012.

although some work has also been reported on multiserver systems with server breakdowns [6,10,15,19]. The current paper is related to the latter type of models in the sense that we consider a system whose service facility is able to deliver more than one unit of work (i.e., service time) per time slot (as in other multiserver models) and that the number of work units that can be executed per time slot is not constant over time (which is typical for systems with server interruptions). More specifically, we consider a model in which the so-called *service capacity*, i.e., the number of work units that the service facility is able to perform, changes randomly from slot to slot. Service capacities are independent from slot to slot and *geometrically distributed*. Customers demanding variable amounts of work (as specified by the service-demand distribution, which can be arbitrary) enter the system and are served in First-Come-First-Served (FCFS) order, i.e., during each slot the service facility executes as many work units as possible (as specified by the momentary service capacity) to the customers present in the system, in their order of arrival. The service of a next customer is only started when the previous customer has received complete service.

Possible application areas of this type of queueing model are numerous. First, it is closely related to the "effective bandwidth" or "effective capacity" concepts in telecommunication networks, to model the time-varying capacity of stations in wireless networks/LANs [7,13,14]. A wireless station can indeed be regarded as a server with varying capacity, due to rate fluctuations of the physical channel or at the MAC layer. A second potential application domain is the modeling of a varying production capacity of a production system with a single product line [1,12,18], in order to estimate the influence of this variability on the holding times. The model also allows to evaluate the impact of a variable number of workers in an HR-environment.

It turns out that, although the model dealt with in this paper belongs to the category of very hard multiserver-type queues with server interruptions, the assumption of geometric service capacities allows for a completely analytic solution of the problem. Remarkably simple explicit expressions can be derived for most quantities of interest, such as the pgf's and mean values of the unfinished work, the queueing delay and (with some restrictions) the buffer occupancy.

The structure of the paper is as follows. In Sect. 2, we describe the queueing model under study. Section 3 presents the analysis of the (steady-state) *unfinished work* in the system, resulting in an explicit expression for the pgf of this quantity. In Sect. 4, we derive the pgf of the (steady-state) *delay* of an arbitrary customer from the pgf of the unfinished work. All these results are valid for arbitrary service-demand distributions. It turns out that the derivation of the pgf of the (steady-state) *buffer occupancy* is much harder. This is the topic of Sect. 5. Section 6 is devoted to the special case where the service-demand distribution is geometric; in this case, we do succeed to derive an explicit expression for the pgf of the buffer occupancy. We discuss the results both conceptually and quantitatively in Sect. 7. Finally, some conclusions and directions for future work are given in Sect. 8.

2 Queueing Model Description

We consider a discrete-time queueing system with infinite waiting room and a service facility (henceforth also referred to as the "server" of the system) which can deliver a variable amount of service as time goes by. The time axis is divided into fixed-length intervals referred to as *time slots* or, simply, *slots*, in the sequel. New customers may enter the system at any given (continuous) point on the time axis, but services are synchronized to (i.e., can only start and end at) slot boundaries.

The arrival process of new customers in the system is characterized by means of a sequence of independent and identically distributed (i.i.d.) nonnegative discrete random variables with common probability mass function (pmf) $a(n)$ and common probability generating function (pgf) $A(z)$. More specifically,

$$a(n) \triangleq \text{Prob}[n \text{ customer arrivals in one slot}] \quad , \quad n \geq 0 \ ; \tag{1}$$

$$A(z) \triangleq \sum_{n=0}^{\infty} a(n)\, z^n \quad . \tag{2}$$

The mean number of customers entering the system per slot, in the sequel referred to as the *mean arrival rate*, is given by $\lambda \triangleq A'(1)$.

The service process of the customers is described in two steps. First, we characterize the *demand* that customers place upon the resources of the system, by attaching to each customer a corresponding *service demand*, which indicates the number of *work units* required to give complete service to the customer at hand. The service demands of consecutive customers arriving at the system are modeled as a sequence of i.i.d. positive discrete random variables with common pmf $s(n)$ and common pgf $S(z)$. More specifically,

$$s(n) \triangleq \text{Prob}[\text{service demand equals } n \text{ work units}] \quad , \quad n \geq 1 \ ; \tag{3}$$

$$S(z) \triangleq \sum_{n=1}^{\infty} s(n)\, z^n \quad . \tag{4}$$

The mean service demand of the customers is given by $1/\sigma \triangleq S'(1)$.

Next, we describe the (variable) *resources* of the server, by attaching to each slot a corresponding *service capacity*, which indicates the number of work units that the server is capable of delivering in this slot. We assume that service capacities are nonnegative random variables, independent from slot to slot and geometrically distributed, with common pmf $r(n)$ and common pgf $R(z)$. More specifically,

$$r(n) \triangleq \text{Prob}[\text{service capacity of } n \text{ work units}] = \frac{1}{1+\mu}\left(\frac{\mu}{1+\mu}\right)^n , \ n \geq 0 \, ; \tag{5}$$

$$R(z) \triangleq \sum_{n=0}^{\infty} r(n)\, z^n = \frac{1}{1+\mu-\mu z} \quad . \tag{6}$$

The mean service capacity of the system (per slot) is given by $\mu = R'(1)$.

Note that in traditional queueing models, the terms *service demand* and *service capacity* are usually not used in the sense defined here. Instead, the term *service time* is used to indicate the total time needed to serve one customer, i.e., the service time is the time a server (with an unspecified service capacity) needs to serve a customer (with an implicit service demand). Since *service time* is an ambiguous concept in our model, we do not use this term in the remainder.

The operation of the queueing system is as follows. Customers arrive in the system according to an uncorrelated arrival process, characterized by the pgf $A(z)$, and take place in the queue in their order of arrival. The amount of service required by each customer (expressed in work units) is given by their corresponding service demand, described by the pgf $S(z)$. The server serves customers from the queue one by one in FCFS order, spending no more work units in each slot than the available service capacity for that slot, which is described by the pgf $R(z)$. If the server disposes of less work units than needed to complete the service of the customer being served in a slot, the service of that customer continues in the next slot. If, on the contrary, the service capacity of the server in a slot is higher than the remaining service demand of the customer in service, then the server starts the service of the next customer in the queue (if any) or (else) becomes idle. We assume that the service of a customer can start during the slot following his arrival slot at the earliest.

3 Unfinished Work

3.1 System Equations

We start the analysis by defining a number of important random variables. Specifically, let u_k denote the unfinished work, i.e., the total number of work units "present in" the system at the beginning of the k-th slot, and e_k the total amount of work entering the system during this slot. Furthermore, let r_k denote the service capacity during the k-th slot. Then, the following recursive system equation can be established:

$$u_{k+1} = e_k + (u_k - r_k)^+ \ , \tag{7}$$

where the notation $(\ldots)^+$ indicates the quantity $\max(0, \ldots)$.

In (7), the r_k's are i.i.d. random variables with (known) pgf $R(z)$, as defined in (5). The random variables $\{e_k\}$, on the other hand, can be obtained as

$$e_k = \sum_{i=1}^{a_k} s_{k,i} \ , \tag{8}$$

where a_k indicates the number of customers entering the system during slot k (with known pgf $A(z)$), and the $s_{k,i}$'s are the service demands (with common pgf $S(z)$) of these customers. It is easily seen that the e_k's are i.i.d. with pgf

$$E(z) = A(S(z)) \ . \tag{9}$$

3.2 Analysis of the Unfinished Work

For all k, let $U_k(z)$ denote the pgf of u_k. Then, from (7) we can derive

$$U_{k+1}(z) = E(z) \cdot E\left[z^{(u_k - r_k)^+}\right] \quad , \tag{10}$$

with $E[\cdot]$ the expectation operator. The second factor in the right hand side of (10) can be expanded further by means of the law of total probability (using also the mutual independence of u_k and r_k):

$$E\left[z^{(u_k - r_k)^+}\right] = \sum_{n=0}^{\infty} r(n) \sum_{i=0}^{\infty} u_k(i) z^{(i-n)^+} \quad , \tag{11}$$

where, for all $i \geq 0$,

$$u_k(i) \triangleq \text{Prob}[u_k = i] \quad . \tag{12}$$

Removing the $(.)^+$ operator and introducing (5), we get

$$\begin{aligned} E\left[z^{(u_k - r_k)^+}\right] &= \sum_{n=0}^{\infty} \frac{1}{1+\mu} \left(\frac{\mu}{1+\mu}\right)^n \left(\sum_{i=0}^{n} u_k(i) + \sum_{i=n+1}^{\infty} u_k(i) z^{i-n}\right) \\ &= \sum_{i=0}^{\infty} u_k(i) \left(\frac{\mu}{1+\mu}\right)^i + \frac{1}{1+\mu} \sum_{i=1}^{\infty} u_k(i) z^i \frac{\left(\frac{\mu}{(1+\mu)z}\right)^i - 1}{\left(\frac{\mu}{(1+\mu)z}\right) - 1} \\ &= U_k\left(\frac{\mu}{1+\mu}\right) + \frac{z}{\mu - (1+\mu)z} \left[U_k\left(\frac{\mu}{(1+\mu)}\right) - U_k(z)\right] \quad . \end{aligned} \tag{13}$$

Combination of (10) and (13) then leads to

$$[\mu - (1+\mu)z] U_{k+1}(z) = E(z) \left(\mu(1-z) U_k\left(\frac{\mu}{1+\mu}\right) - z U_k(z)\right) \quad . \tag{14}$$

Now, let us assume that the queueing system at hand is stable, i.e., that the stability condition [5,17] is fulfilled. The system is stable if and only if the mean number of work units entering the system per slot, given by $E'(1)$, is strictly less than the mean service capacity per slot, given by $R'(1)$, or, expressed in the basic parameters of our system, if and only if

$$\lambda < \mu\sigma \quad . \tag{15}$$

This inequality says that the supremum of the achievable throughput of the system, expressed in customers per slot, is given by $\mu\sigma$.

We now let the time parameter k go to infinity. Assuming the system reaches a steady state, then both functions $U_k(\cdot)$ and $U_{k+1}(\cdot)$ converge to a common limit function $U(\cdot)$, which denotes the pgf of the unfinished work at the beginning of an arbitrary slot in steady state. As a result, (14) translates into a linear equation for $U(z)$, with solution

$$U(z) = \mu U\left(\frac{\mu}{1+\mu}\right) \frac{(z-1)E(z)}{\mu(z-1) - z[E(z)-1]} \quad . \tag{16}$$

This expression is fully determined, except for the unknown quantity $U(\frac{\mu}{1+\mu})$, which, in turn, can be obtained by invoking the normalizing condition of the pgf $U(z)$, i.e., the condition $U(1) = 1$. It is not difficult to obtain

$$U\left(\frac{\mu}{1+\mu}\right) = 1 - \frac{\lambda}{\mu\sigma} \quad . \tag{17}$$

With (9) and (17), we then finally get the following closed-form expression for the steady-state pgf of the unfinished work in the system:

$$U(z) = \frac{(\mu - \frac{\lambda}{\sigma})(z-1)A(S(z))}{\mu(z-1) - z[A(S(z)) - 1]} \quad . \tag{18}$$

Various performance measures of the system, related to the unfinished work, can be derived in explicit form from the above result. For instance, the mean unfinished work in the system is given by

$$E[u] = U'(1) = \frac{2\lambda[(\mu+1)\sigma - \lambda] + A''(1) + \lambda\sigma^2 S''(1)}{2\sigma(\mu\sigma - \lambda)} \quad . \tag{19}$$

Higher-order moments of the unfinished-work distribution can be obtained similarly, by computing higher-order derivatives of $U(z)$.

4 Customer Delay

We now turn to the analysis of the probability distribution of the delay (expressed in slots) customers incur in the system. More specifically, let C denote an arbitrary customer entering the system in steady state, and let J denote the slot during which C arrives. In the sequel, customer C will be referred to as the "tagged customer". We define the (discrete) delay d of customer C as the total number of (full) slots between the arrival instant of C in the system and the departure time of C from the system, i.e., d indicates the number of slots between the end of slot J and the end of the slot during which the last work unit of the service demand of C is actually being executed.

Owing to the FCFS queueing discipline used in the system, the delay d of the tagged customer C is equal to the time needed to execute the unfinished work present in the system just after slot J, but to be performed before or during the service of customer C. In the next subsections, we first compute the pgf of this amount of work. Next, from this, we derive the pgf of d.

4.1 Work to Be Performed before the Tagged Customer

Let $\tilde{u}$ denote the unfinished work at the beginning of slot J, $\tilde{r}$ the available service capacity during slot J (with pgf $R(z)$ as defined in (5)), and f the number of customers entering the system during slot J but to be served before C. Then, the total amount of work to be performed before the service of the tagged customer

C, still "present in" the system just after slot J, i.e., at the moment when the delay d of customer C starts running, is given by

$$v = (\tilde{u} - \tilde{r})^+ + \sum_{i=1}^{f} \tilde{s}_i \ , \tag{20}$$

where the quantities $\tilde{s}_i$ refer to the service demands of the f customers entering the system during slot J, but to be served before the tagged customer C.

It is well-known, see e.g. [2,9,15], that the pgf of f is given by

$$F(z) \triangleq E\left[z^f\right] = \frac{A(z) - 1}{\lambda(z-1)} \ . \tag{21}$$

On the other hand, the independent nature of the arrival process (from slot to slot) implies that the probability distribution of $\tilde{u}$, i.e., the unfinished work at the beginning of the *arrival slot* of the tagged customer C, is identical to the probability distribution of the unfinished work at the beginning of an *arbitrary slot* in the steady state. This implies that the pgf of $\tilde{u}$ is equal to the function $U(z)$ determined earlier (see (18)). For the same reason, the random variables f and $\tilde{u}$ are mutually independent. Putting all these elements together, we conclude that the pgf of v can be obtained as

$$V(z) \triangleq E[z^v] = E\left[z^{(\tilde{u}-\tilde{r})^+}\right] \cdot E\left[z^{\sum_{i=1}^{f} \tilde{s}_i}\right] = \frac{U(z)}{A(S(z))} \cdot F(S(z)) \ , \tag{22}$$

where, in the last step, we have used (9) and equation (10) for $k \to \infty$.

Using (18), (21) and (22), we find the following explicit expression for $V(z)$:

$$V(z) = \frac{(\mu - \frac{\lambda}{\sigma})(z-1)[A(S(z)) - 1]}{\lambda[S(z) - 1]\left\{\mu(z-1) - z[A(S(z)) - 1]\right\}} \ . \tag{23}$$

4.2 Analysis of the Delay

The delay d of customer C is nothing else than the number of slots required to perform the remaining service demands of the customers in front of C just after slot J (which takes v work units) together with customer C itself (which requires one full service demand $\tilde{s}$). That is, the delay d of customer C is equal to the time needed to perform $v + \tilde{s}$ work units. If we denote by $\tilde{r}_j$ the available service capacity in the j-th slot after slot J, and by q_i the total service capacity available during i consecutive slots (just after slot J), then it is easy to see that

$$q_i = \sum_{j=1}^{i} \tilde{r}_j \ , \tag{24}$$

with corresponding pgf

$$Q_i(z) = R(z)^i \ . \tag{25}$$

The distribution of the delay d can then be obtained as follows. First, we express the tail distribution, for $i \geq 0$, as

$$\text{Prob}[d > i] = \text{Prob}[q_i < v + \tilde{s}] = \sum_{n=0}^{\infty} \text{Prob}[q_i = n] \, \text{Prob}[v + \tilde{s} > n] \quad . \tag{26}$$

The reasoning behind this equation is that more than i slots are required to remove $v + \tilde{s}$ units of work from the system, if and only if at most $v + \tilde{s} - 1$ work units can be performed during i slots. We now z-transform the above equation and represent $\text{Prob}[q_i = n]$ as a derivative, i.e.,

$$\text{Prob}[q_i = n] = \left[\frac{1}{n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} Q_i(x) \right]_{x=0} = \left[\frac{1}{n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} R(x)^i \right]_{x=0} \tag{27}$$

to obtain an equation for the pgf $D(z)$ of the delay d:

$$\begin{aligned} \frac{D(z) - 1}{z - 1} &= \sum_{i=0}^{\infty} z^i \ \text{Prob}[d > i] \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \ \text{Prob}[v + \tilde{s} > n] \left[\frac{\partial^n}{\partial x^n} \sum_{i=0}^{\infty} z^i R(x)^i \right]_{x=0} \quad . \end{aligned} \tag{28}$$

In the above equation, the partial derivative can be expressed as

$$\begin{aligned} \left[\frac{\partial^n}{\partial x^n} \sum_{i=0}^{\infty} z^i R(x)^i \right]_{x=0} &= \left[\frac{\partial^n}{\partial x^n} \frac{1}{1 - zR(x)} \right]_{x=0} = \left[\frac{\partial^n}{\partial x^n} \frac{1 + \mu - \mu x}{1 - z + \mu - \mu x} \right]_{x=0} \\ &= \delta(n) + \frac{n! \mu^n z}{(1 - z + \mu)^{n+1}} \quad , \end{aligned} \tag{29}$$

where $\delta(n)$ is the well-known Kronecker delta-function, which equals 1 for $n = 0$ and 0 for all $n > 0$. Combination of (28) and (29) then leads to

$$\begin{aligned} \frac{D(z) - 1}{z - 1} &= 1 + \frac{z}{1 - z + \mu} \sum_{n=0}^{\infty} \left(\frac{\mu}{1 - z + \mu} \right)^n \text{Prob}[v + \tilde{s} > n] \\ &= 1 + \frac{z}{1 - z + \mu} \left[\frac{V(y) S(y) - 1}{y - 1} \right]_{y = \frac{\mu}{1 - z + \mu}} \quad . \end{aligned} \tag{30}$$

Using the formula we obtained earlier for the pgf $V(\cdot)$ in (23), this translates into the following explicit expression for the pgf of the queueing delay:

$$D(z) = \frac{\mu\sigma - \lambda}{\lambda\mu\sigma} \ \frac{z(z-1) S(\frac{\mu}{1-z+\mu}) \left[A(S(\frac{\mu}{1-z+\mu})) - 1 \right]}{\left[S(\frac{\mu}{1-z+\mu}) - 1 \right] \left[z - A(S(\frac{\mu}{1-z+\mu})) \right]} \quad . \tag{31}$$

Various delay-related performance measures can be derived from (31) in explicit form. For instance, the mean delay can be obtained as

$$E[d] = D'(1) = 1 + \frac{\lambda\sigma + (\mu\sigma - \lambda)}{\mu\sigma(\mu\sigma - \lambda)} + \frac{\mu A''(1) + \lambda^2 \sigma S''(1)}{2\lambda\mu(\mu\sigma - \lambda)} \quad . \tag{32}$$

Higher-order moments of the delay can be obtained by computing higher-order derivatives of $D(z)$ at $z = 1$.

5 Buffer Occupancy

The *buffer occupancy* is defined as the number of customers present in the system. In this section, we try to find an expression for the pgf and/or the mean value of the buffer occupancy from the results obtained in the previous sections.

Let us define b_k as the buffer occupancy at the beginning of the k-th slot. Then, clearly, the following relationships can be established between b_k and the unfinished work u_k, defined in Sect. 3:

$$u_k = 0 \quad , \quad \text{if} \quad b_k = 0 \ ; \tag{33}$$

$$u_k = \hat{s}_1 + s_2 + \ldots + s_{b_k} \quad , \quad \text{if} \quad b_k > 0 \ , \tag{34}$$

where $\hat{s}_1$ indicates the remaining service demand of the customer in service at the beginning of slot k and $s_2, \ldots, s_{b_k}$ denote the (full) service demands of the other $b_k - 1$ customers in the system at the beginning of slot k.

It is not straightforward to derive from the above equations a relationship between the pgf's of the unfinished work and the buffer occupancy. There are two main reasons for this. First, it is not obvious how to find the distribution (or pgf) of the random variable $\hat{s}_1$: the classical results from renewal theory [4,16] on the distribution of the residual lifetime in a sequence of i.i.d. random variables are not applicable here, as the remaining service demand of an ongoing service does not simply decrease by one unit per slot in the system under study (because the service capacities are variable). Second, the random variables $\hat{s}_1$ and b_k, appearing in (33) and (34), are not necessarily independent. In the next section, however, we shall see that these obstacles do not exist if the service demands have a geometric distribution, and we shall be able to determine the pgf of the buffer occupancy completely in that case.

It should also be noted that the *mean* buffer occupancy can always be derived from the mean delay, by applying (the discrete-time version of) Little's result [8]:

$$E[b] = \lambda E[d] = \lambda + \frac{\lambda^2\sigma + \lambda(\mu\sigma - \lambda)}{\mu\sigma(\mu\sigma - \lambda)} + \frac{\mu A''(1) + \lambda^2\sigma S''(1)}{2\mu(\mu\sigma - \lambda)} \ . \tag{35}$$

6 Geometric Service Demands

In the previous sections, we have made no specific assumptions as to the precise nature of the service-demand distribution, i.e., the pgf $S(z)$ was arbitrary. In this section, we explore the special case where the service demands are geometrically distributed with mean value $1/\sigma$, such that

$$s(n) = \sigma(1-\sigma)^{n-1} \quad , \quad n \geq 1 \ ; \quad S(z) = \frac{\sigma z}{1 - (1-\sigma)z} \ . \tag{36}$$

6.1 Performance Measures

The pgf of the unfinished work in the system at the beginning of an arbitrary slot in steady state, given in general by (18), cannot be much simplified in case of geometric service demands, i.e., the assumption of geometric service demands does not seem to have much impact on the form of this pgf. The result reads

$$U(z) = \frac{(\mu - \frac{\lambda}{\sigma})(z-1)A(\frac{\sigma z}{1-(1-\sigma)z})}{\mu(z-1) - z[A(\frac{\sigma z}{1-(1-\sigma)z}) - 1]} \quad . \tag{37}$$

The corresponding expression for the mean unfinished work can be found from (19) by the substitution $S''(1) = 2(1-\sigma)/\sigma^2$ as

$$E[u] = \frac{2\lambda(1+\mu\sigma-\lambda) + A''(1)}{2\sigma(\mu\sigma-\lambda)} \quad . \tag{38}$$

The general expressions of the pgf and the mean value of the customer delay, given in (31) and (32), simplify considerably in case the service demands are geometrically distributed. The results are given by

$$D(z) = \frac{(\mu\sigma-\lambda)z[A(\frac{\mu\sigma}{1+\mu\sigma-z}) - 1]}{\lambda[z - A(\frac{\mu\sigma}{1+\mu\sigma-z})]} \quad ; \tag{39}$$

$$E[d] = 1 + \frac{2\lambda + A''(1)}{2\lambda(\mu\sigma-\lambda)} \quad . \tag{40}$$

With reference to the analysis of the buffer occupancy in Sect. 5, the assumption of geometric service demands brings about substantial simplifications. The main reason for this lies in the memoryless nature of the geometric distribution [16], which implies that the distribution of the remaining service demand ($\hat{s}_1$) is identical to the distribution of a full service demand (such as $s_2, \ldots, s_{b_k}$). Also, the distribution of $\hat{s}_1$, in this case, is not influenced by the value of b_k: although b_k may be correlated with the received amount of service of the customer in service (at the beginning of slot k), this does not determine in any way the distribution of the remaining service demand, owing to the memoryless property of the geometric distribution. As a consequence, (33) and (34) are equivalent to

$$u_k = \sum_{i=1}^{b_k} s_i \quad , \tag{41}$$

where the s_i's are i.i.d. with geometric distribution (with mean $1/\sigma$), which, in addition, are independent of b_k. The (steady-state) pgf of u_k then simply follows as

$$U(z) = B\left(\frac{\sigma z}{1-(1-\sigma)z}\right) \quad , \tag{42}$$

which leads to

$$B(x) = U\left(\frac{x}{\sigma + (1-\sigma)x}\right) \quad . \tag{43}$$

Using (37), we then get the pgf of the buffer occupancy explicitly as

$$B(z) = \frac{(\mu\sigma - \lambda)(z-1)A(z)}{\mu\sigma(z-1) - z[A(z)-1]} \quad . \tag{44}$$

The mean buffer occupancy is derived as

$$E[b] = B'(1) = \lambda + \frac{2\lambda + A''(1)}{2(\mu\sigma - \lambda)} \quad . \tag{45}$$

The same result can also be obtained by applying Little's result, i.e., $E[b] = \lambda E[d]$, to (40), which supports our confidence in the results of our analysis.

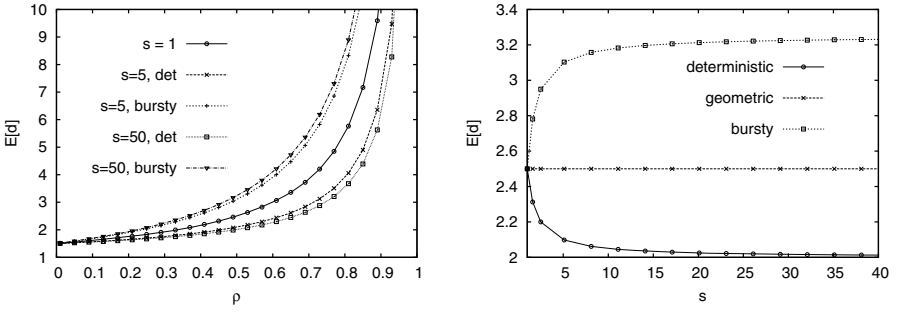
6.2 Geometric Invariance Property

We conclude this section on geometric service demands with the observation that, in this particular case, the distributions of the buffer occupancy and the delay depend on the service capacities and the service demands only through the product $\mu\sigma$, as can be clearly seen from (39), (40), (44) and (45). This is not true for the distribution of the unfinished work, though. We will refer to this remarkable property with the term "geometric invariance" in the sequel. This result implies that, for instance, doubling the mean service demands of the customers (i.e., dividing σ by 2) and at the same time doubling the mean per-slot service capacity of the system (i.e., multiplying μ by 2), does not alter the delay and the buffer occupancy of the system, but it does double the mean unfinished work in the system. Intuitively, this is acceptable, because higher service demands appear to be compensated by the proportionally higher service capacity, and hence, the effective service times of the customers (expressed in slots) basically remain the same. The average amount of work in the system, of course, doubles, as all work-related quantities scale up. The geometric invariance property can also be interpreted as follows: if we replace the geometric service demands (with mean $1/\sigma$ work units) with deterministic service demands equal to 1 work unit each and we slow down the server from an average service capacity of μ work units per slot to $\mu\sigma$ work units per slot, the delay and the buffer occupancy remain the same. Further, we will discover that all these conclusions are not necessarily true for non-geometric service-demand distributions.

7 Discussion of Numerical Examples

In this section, we discuss the results obtained above by means of some numerical examples. Specifically, we investigate the behavior of the system when one of the parameters s ($\triangleq 1/\sigma$), μ and λ is varied. In order to make for a fair comparison, we scale one of the other parameters to keep the load ρ ($\rho = \lambda s/\mu$) constant.

First, we examine the influence of varying the mean service demand s and the mean service capacity μ while keeping their ratio constant. In Fig. 1(a), the mean delay is depicted versus ρ, for the case of Poisson arrivals, $s/\mu = 0.5$,



(a) vs. the load ρ; $s = 1, 5$ or 50 (b) vs. the mean demand s; $\rho = 0.5$

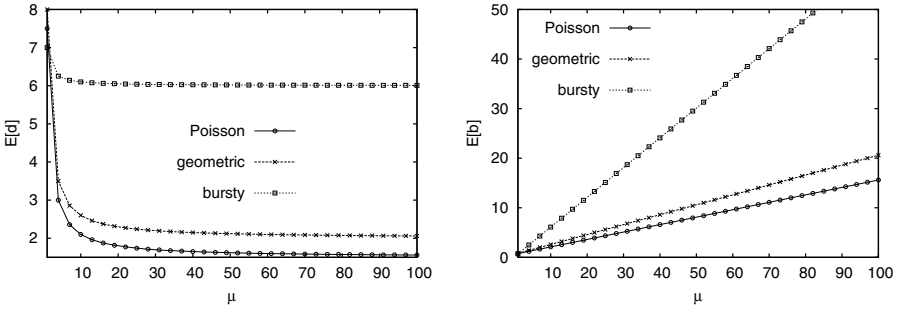
Fig. 1. Mean delay $E[d]$ for Poisson arrivals and $s/\mu = 0.5$

$s = 1, 5$ and 50 and for three possible distributions of the service demand, namely deterministic, geometric or "bursty". Here, "bursty" refers to the case where the service demand of a customer is either 1 or $5 \cdot s$ work units:

$$S(z) = \frac{4s}{5s-1}\ z + \frac{s-1}{5s-1}\ z^{5s}\ . \tag{46}$$

When $\rho \to 0$, the mean delay tends to the constant value $1 + s/\mu$, irrespective of the absolute values of s and μ, or of the service-demand distribution. This can be understood as follows: when $\rho \to 0$, an arriving customer arrives in an empty system. He therefore has to wait 1 slot extended with, on average, an extra $1/\mu$ slots per work unit, for a total of s/μ slots. Also, the mean delay increases with increasing ρ, as expected. For $s = 1$, the service demand is deterministically equal to 1. Because of the geometric invariance property (see Sect. 6), the mean delay does not depend on the absolute values of s and μ for geometric demands, if s/μ is kept constant. For these two reasons, the curve for $s = 1$ in Fig. 1(a) is identical to the curves for other values of s and geometric demands, as well as for $s = 1$ and the two 'other' service-demand distributions (which are all the same for $s = 1$). When the demands are not geometric and $s > 1$, we see a completely different picture; the mean delay can differ drastically when s varies, both in positive and in negative sense. For service-demand distributions with less variance than the geometric distribution (e.g., deterministic), larger demands and larger service capacities are beneficial, while for more bursty demands, it is favorable to keep the demands small. So, the geometric distribution seems to be a turning point. These conclusions can also be drawn from Fig. 1(b). Here, we depict the mean delay versus s, for $\rho = 0.5$ and, otherwise, the same assumptions as in Fig. 1(a). The geometric invariance property is nicely illustrated by the constant line, while it is demonstrated that less bursty (more bursty, respectively) distributions for the service demand lead to *lower* mean delay *decreasing* with s (*higher* mean delay *increasing* with s, respectively).

Second, the impact of a change in the mean arrival rate λ and the mean service capacity μ is studied while maintaining their ratio constant. In Fig. 2, the mean



(a) Mean delay $E[d]$ (b) Mean buffer occupancy $E[b]$

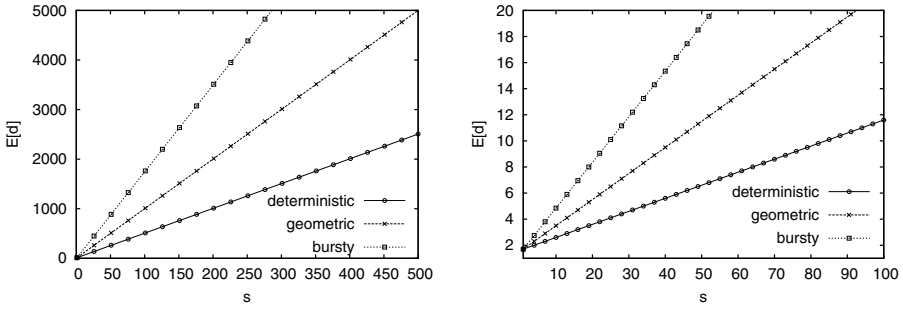
Fig. 2. Mean delay $E[d]$ and mean buffer occupancy $E[b]$ vs. the mean service capacity μ for Poisson, geometric and bursty arrivals, deterministic demands $s = 50$ and $\rho = 0.5$

delay and the mean buffer occupancy are plotted versus μ, for deterministic service demands equal to $s = 50$, $\rho = 0.5$ and three possible distributions for the number of per-slot arrivals, namely Poisson, geometric or "bursty". In the latter case, the number of arrivals is either 0 or $10 \cdot \lambda$:

$$A(z) = 0.9 + 0.1z^{10\lambda} \quad . \tag{47}$$

From Fig. 2, we observe that the mean delay goes to ∞ when μ and λ go to 0, while the mean buffer occupancy tends to 0. This is logical, since an arriving customer (requiring $s = 50$ work units) needs a high number of slots to be served completely when the capacity is low (even an infinite number of slots when the capacity goes to 0). However, the arrival rate is small and, therefore, the mean buffer occupancy is small as well (and zero if λ goes to 0). So, for low μ and λ, the mean delay is almost entirely a result of one customer in the system not having enough service capacity, rather than other customers being present that have to be served before this customer. When μ increases slightly, most of the customers are still waiting 'alone' in the system, so the mean delay decreases (dramatically). For even higher μ, the positive effect of increasing μ is reduced by increasing λ. For increasing λ, the mean buffer occupancy, therefore, increases, while the mean delay evolves to a constant value for μ going to ∞. We further note that a higher variance of the number of per-slot arrivals leads to higher mean delays and buffer occupancies.

Finally, we keep the mean capacity μ constant and vary the mean arrival rate λ and the mean service demand s, so as to keep the product λs constant. In Fig. 3, we depict the mean delay as a function of s, when the service demands are deterministic, geometric or bursty (according to (46)), for Poisson arrivals, $\rho = 0.5$ and for $\mu = 0.2$ (Fig. 3(a)) and $\mu = 10$ (Fig. 3(b)). We perceive that the mean delay increases dramatically when s increases, even when λ decreases inversely proportionally. This is easily explained intuitively; increasing s and decreasing λ entails less customer arrivals and a larger number of work units per customer. As a result, the work-arrival process (i.e., the sequence of work



(a) for $\mu = 0.2$ (b) for $\mu = 10$

Fig. 3. Mean delay $E[d]$ vs. the mean demand s for λs a constant, Poisson arrivals and $\rho = 0.5$

units that enter the system during consecutive slots) is burstier and thus leads to larger delays. We also notice that a higher variance of the service demands leads to a higher increment of the mean delay when s increases.

8 Conclusions

We have obtained explicit expressions for the main performance measures of a basic discrete-time queueing model with variable service capacity. Numbers of arrivals per slot and service demands have general distributions. The main restriction seems to be the assumption that service capacities are *i.i.d.* and *geometrically distributed*. Although these assumptions may not be as general as one would like, they do play a role in the analytic tractability of the model, and they allow to study the behavior of a variable service-capacity system in conceptual terms. We have discovered the remarkable "geometric invariance" property in case of geometric service demands. Numerical results show, however, that in more general conditions, changing one of the system parameters can have an undeniable impact on the performance, even when the load is kept constant.

As future work, *more general service-capacity distributions* can be considered than the geometric distribution. In fact, some literature exists on similar models whereby the service capacity is variable and not geometric. E.g., in [3,11,15], a discrete-time queueing model is studied with constant service times of 1 slot each and a constant number of servers, of which a variable number is available from slot to slot. In terms of our model, this comes down to assuming that $S(z) = z$ and the service-capacity distribution has *finite support*, i.e., $R(z)$ is a *polynomial*. Although these papers consider non-geometric service capacities, they are not more general than the present study: the analysis in these papers relies heavily on the polynomial nature of $R(z)$ and on the deterministic (single-slot) nature of the service times (two restrictions that the present model does not have).

Acknowledgment. The second author is a Postdoctoral Fellow with the Research Foundation - Flanders (FWO-Vlaanderen), Belgium.

References

1. Balkhi, Z.T.: On the global optimal solution to an integrated inventory system with general time varying demand, production and deterioration rates. European Journal of Operations Research 114, 29–37 (1999)
2. Bruneel, H.: Buffers with stochastic output interruptions. Electronics Letters 19, 735–737 (1983)
3. Bruneel, H.: A general model for the behaviour of infinite buffers with periodic service opportunities. European Journal of Operational Research 16(1), 98–106 (1984)
4. Bruneel, H.: Performance of discrete-time queueing systems. Computers and Operations Research 20(3), 303–320 (1993)
5. Bruneel, H., Kim, B.G.: Discrete-time models for communication systems including ATM. Kluwer Academic, Boston (1993)
6. Chakravarthy, S.R.: Analysis of a multi-server queue with Markovian arrivals and synchronous phase type vacations. Asia-Pacific Journal of Operational Research 26(1), 85–113 (2009)
7. Chang, C.S., Thomas, J.A.: Effective bandwidth in high-speed digital networks. IEEE Journal on Selected Areas in Communications 13, 1091–1100 (1995)
8. Fiems, D., Bruneel, H.: A note on the discretization of Little's result. Operations Research Letters 30, 17–18 (2002)
9. Gao, P., Wittevrongel, S., Bruneel, H.: Discrete-time multiserver queues with geometric service times. Computers and Operations Research 31, 81–99 (2004)
10. Gao, P., Wittevrongel, S., Laevens, K., De Vleeschauwer, D., Bruneel, H.: Distributional Little's law for queues with heterogeneous server interruptions. Electronics Letters 46, 763–764 (2010)
11. Georganas, N.D.: Buffer behavior with poisson arrivals and bulk geometric service. IEEE Transactions on Communications 24, 938–940 (1976)
12. Glock, C.H.: Batch sizing with controllable production rates. International Journal of Production Research 48, 5925–5942 (2010)
13. Jin, X., Min, G., Velentzas, S.R.: An analytical queuing model for long range dependent arrivals and variable service capacity. In: Proceedings of IEEE International Conference on Communications (ICC 2008), Beijing, pp. 230–234 (May 2008)
14. Kafetzakis, E., Kontovasilis, K., Stavrakakis, I.: Effective-capacity-based stochastic delay guarantees for systems with time-varying servers, with an application to IEEE 802.11 WLANs. Performance Evaluation 68, 614–628 (2011)
15. Laevens, K., Bruneel, H.: Delay analysis for discrete-time queueing systems with multiple randomly interrupted servers. European Journal of Operations Research 85, 161–177 (1995)
16. Mitrani, I.: Modelling of Computer and Communication Systems. Cambridge University Press, Cambridge (1987)
17. Takagi, H.: Queueing Analysis, A Foundation of Performance Evaluation. Discrete-time systems, vol. 3. North-Holland, Amsterdam (1993)
18. Yang, H.-L.: A partial backlogging production-inventory lot-size model for deteriorating items with time-varying production and demand rate over a finite time horizon. International Journal of Systems Science 42, 1397–1407 (2011)
19. Yang, X.L., Alfa, A.S.: A class of multi-server queueing system with server failures. Computers & Industrial Engineering 56(1), 33–43 (2009)

Perfect Sampling of Networks with Finite and Infinite Capacity Queues

Ana Bušić[1], Bruno Gaujal[2], and Florence Perronnin[3]

[1] INRIA - ENS, Paris, France
ana.busic@inria.fr
[2] INRIA Grenoble - Rhône-Alpes, Montbonnot, France
bruno.gaujal@inria.fr
[3] Joseph Fourier University, Grenoble, France
Florence.Perronnin@imag.fr

Abstract. We consider open Jackson queueing networks with mixed finite and infinite buffers and analyze the efficiency of sampling from their exact stationary distribution. We show that perfect sampling is possible, although the underlying Markov chain has a large or even infinite state space. The main idea is to use a Jackson network with infinite buffers (that has a product form stationary distribution) to bound the number of initial conditions to be considered in the coupling from the past scheme. We also provide bounds on the sampling time of this new perfect sampling algorithm under *hyper-stability* conditions (to be defined in the paper) for each queue. These bounds show that the new algorithm is considerably more efficient than existing perfect samplers even in the case where all queues are finite. We illustrate this efficiency through numerical experiments.

1 Introduction

The stationary behavior of queueing networks (QNs) can only be obtained quite efficiently under specific assumptions that yield the so called product-form property; e.g., [7]. This property means that the stationary probability distribution of these networks can be decomposed, up to a normalizing constant, in the product of the marginal distributions of each network node (or queue). In several cases, product-form QNs are restrictive because they often assume that nodes have infinite buffer sizes or that the behavior of a network node does not depend on the state of other nodes; e.g., [6]. In the context of Internet networks, blocking and rejection mechanisms arise due to finite-buffer constraints and state-dependent routing. While it is possible to obtain the stationary distribution of non-product-form QNs through the solution of a set of linear equations, i.e., the global-balance equations [2], the huge size of their state space makes this approach of practical interest only for small networks. In this setting, simulation is a useful approach to obtain robust measures and insights on the stationary performance.

At the cost of a slightly higher computational complexity than Monte Carlo simulation, the exact stationary distribution can be sampled in finite time using

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 136–149, 2012.

a technique called coupling from the past (see the seminal work by Propp and Wilson [11]). Unlike Monte Carlo simulation, this powerful technique produces independent samples of ergodic finite Markov chains *exactly distributed* according to their stationary distribution. For this reason, this technique is also known as perfect sampling algorithm (PSA) and will be denoted by PSA in the remainder of the paper.

This technique has been used to design simulation algorithms for queueing networks with finite capacity buffers and rather general routing policies [12]. The main assumption needed to make these algorithms work is that the state space is finite (or equivalently that the buffer capacities are all finite).

In this paper we present a new network simulation algorithm that can handle finite and infinite buffers at the same time, with rejection of jobs arriving at a saturated queue. We also derive a bound on its sampling time complexity and show that it does not depend on the size of the state space, under a *hyper-stability* condition (defined in section 4).

Related Work. As mentioned before, the original perfect sampling algorithm has been adapted for the simulation of monotone queueing networks in [12]. The complexity of this algorithm has been analyzed in [4] for acyclic networks, while cyclic networks have been studied in [1].

A series of papers [3, 5, 9, 10] propose new perfect sampling algorithms (introducing new ideas such as *envelopes*, *splitting* and *skipping*) for non-monotone queueing networks. However, in all cases, it is essential that the underlying Markov chain has a finite state space.

In [8], the concept of bounding process is introduced. The coupling between the original process and the bounding process is different from what we propose here. Actually we could not see how to use the coupling proposed in [8] when the bounding process is not reversible. Our technique to construct the forward and the backward chains does not use the same events for both chains but is based on the same random innovations.

Outline. In section 2, we show how to use this concept for open Jackson queueing networks with mixed finite and infinite buffers and in section 3 we propose a new PSA based on this approach. We prove that this algorithm constructs samples of the exact stationary distribution of the number of jobs in all queues. In section 4 we establish a theoretical bound on the expected simulation time. Finally, in section 5 we show that in the case where all buffers are finite our algorithm clearly outperforms previous perfect samplers that all have larger simulation times. We illustrate this with numerical experiments in section 6.

2 Queueing Network Model

We consider an open Jackson queueing network (JQN) $\mathcal{J}$ with M queues. The vector $\mathbf{C} = (C_1, \ldots, C_M)$ denotes the buffer size of each queue. Note that for any i, $C_i \in \mathbb{N} \cup \infty$, *i.e.* the buffers can either be finite or infinite. For $1 \leqslant i \leqslant M$,

we denote by $\mathbf{e}_i$ the vector in $\mathbb{Z}^M$ with all the components equal to 0, except for component i that is equal to 1.

An infinite stream of jobs that follow a Poisson process with rate λ_0 joins the JQN from an external source (numbered 0 in the following). The fact that the external source is seen as a node of the network (numbered 0) will unify the notations in the rest of the paper. To take this one step further, we also denote $\mu_0 \stackrel{\text{def}}{=} \lambda_0$. The probability that a job joins queue i, upon arrival to the network, is p_{0i}. In queue i ($i > 0$), each job requires some processing for an exponentially distributed amount of time with mean service rate μ_i. The service discipline of each queue i is work-conserving. Upon completion of service at queue i, a job is sent to queue j with probability p_{ij}, and it is accepted if queue j has an available slot (i.e., if it is non-saturated), otherwise the job is lost. The probability that a job leaves the network after service at i is p_{i0}. We make the classical assumption that the matrix $P = (p_{i,j})_{i,j\in\{0,\dots,M\}}$ is irreducible. This implies that all queues always get new jobs and that all jobs eventually leave the network.

Under the foregoing assumptions, the stochastic process

$$\{(X_1(t), \dots, X_M(t)) \in \mathbb{Z}^M : 0 \le X_i(t) \le C_i, \forall i\}_{t\ge 0},$$

is a continuous-time Markov chain where $X_i(t)$ denotes the number of jobs in queue i at time t. The space of all the possible states is $\mathcal{S} \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{Z}^M : 0 \le x_i \le C_i, \forall i\}$. Our main notation is summarized in Table 1.

2.1 Discrete-Event Definition of JQN

The JQN $\mathcal{J}$ with M queues described above can be seen as a discrete-event system with a single type of events, namely a_{ij}, $(i, j \in \{0, 1, \dots, M\})$ corresponding to the service of one job in queue i that then joins queue j. The dummy queue 0 corresponds to the outside world: An event of type a_{0j} is an exogenous arrival in queue j and an event of type a_{i0} corresponds to the departure of a job from queue i. If queue i is empty then event a_{ij} has no effect on the system. The set of all events is denoted by $\mathcal{A}$.

The rate of event a_{ij} is γ_{ij} and it is independent of M and $\mathbf{C}$, for any i, j. Using the previous description of a JQN, for all $i, j \in \{0, 1, \dots, M\}$, $\gamma_{ij} = \mu_i p_{ij}$. The total event rate $\Gamma \stackrel{\text{def}}{=} \sum_{i,j=0}^{M} \gamma_{ij}$ is finite (we set $\gamma_{00} = 0$).

The continuous-time Markov chain described above can be transformed into a discrete-time Markov chain $(X_n)_{n\in\mathbb{N}}$ with the same stationary distribution using uniformization by constant Γ. Using the foregoing assumption that the routing matrix P is irreducible, this discrete chain is irreducible and aperiodic. The evolution of the Markov chain $(X_n)_{n\in\mathbb{N}}$ can be written under the form $X_{n+1} = \phi(X_n, U_{n+1})$ where $(U_n)_{n\geqslant 0}$ are i.i.d random variables, uniformly distributed over $[0, 1]$.

The interval $[0, 1]$ is partitioned into intervals A_{ij}, corresponding to the events a_{ij}, $0 \leqslant i, j \leqslant M$. The interval A_{ij} is of length γ_{ij}/Γ, corresponding to the probability of event a_{ij}. The forward transition function of the chain $\phi : \mathcal{S} \times [0, 1] \to \mathcal{S}$ is defined as follows:

Table 1. Notation (in order of appearance)

QN	Queuing network
JQN	Jackson queuing network
PSA	Perfect sampling algorithm [11]
$\mathcal{J}$	Jackson network to be sampled (finite/infinite queues)
M	Number of queues
C_i	Capacity of queue i
$\mathbf{C}$	Vector of queue capacities
$\mathbf{e}_i$	Unit vector: with zero coordinates except the i-th one that is equal to 1.
λ_0	Total exogenous job arrival rate (also $\mu_0 = \lambda_0$).
p_{0i}	Fraction of exogenous jobs arriving at queue i
p_{ij}	Routing probability from queue i to j
p_{i0}	Probability that after service completion at queue i a job leaves the system
μ_i	Service rate of queue i
$\mathcal{S}$	State space of the CTMC
a_{ij}	Event "job leaving queue i and joining queue j"
$\mathcal{A}$	Set of all possible events
γ_{ij}	Rate of event a_{ij} ($\gamma_{ij} = \mu_i p_{i,j}$)
Γ	Total event rate ($\Gamma \stackrel{\text{def}}{=} \sum_{ij} \gamma_{ij}$)
X_n	Uniformized, discrete-time Markov chain: number of jobs in each queue at time n
$\phi(\mathbf{x}, u)$	Forward transition function from state $\mathbf{x}$ with innovation u
U_n	Sequence of i.i.d. random variables uniformly distributed over $[0, 1]$
$\mathcal{J}^\infty$	Infinite-queue JQN
X_n^∞	Uniformized DTMC for $\mathcal{J}^\infty$, the infinite-queue JQN
$\phi^\infty(\mathbf{x}, u)$	Forward transition function for $\mathcal{J}^\infty$, the infinite-queue JQN
$\preceq$	componentwise partial order
λ_i	Total arrival rate in queue i ($\lambda_i = \sum_{j=0}^{M} \lambda_j p_{ji}$).
q_{ij}	Routing probabilities for the *reversed* JQN.
θ_{0j}	Exogenous arrival rate in queue j for the *reversed* JQN.
$\beta^\infty(\mathbf{x}, u)$	Backward transition function
$\mathbf{1}_A$	Indicator function: 1 if proposition A is true, 0 otherwise
$X^{(1)}$	Upper bounding process
$X^{(2)}$	Lower bounding process
PSA-BP	Perfect sampling with bounding process (proposed algorithm)
Y_n^∞	Reversed chain of X_n^∞
Y_n	Reversed chain of X_n
$\tau(X^{(2)}, X^{(1)})$	Coalescence time of the trajectories of the chain X_n starting in $X^{(1)}$ and $X^{(2)}$.
S_i	Expected stationary size in queue i of the infinite-queue JQN
S	Maximum expected stationary queue size: $S \stackrel{\text{def}}{=} \max_i S_i$

- If $u \in A_{ij}$ then $\phi(\mathbf{x}, u) = ((\mathbf{x} - \mathbf{e}_i) \vee 0 + \mathbf{e}_j \mathbf{1}_{\mathbf{x} - \mathbf{e}_i \geqslant 0}) \wedge \mathbf{C}$;
- If $u \in A_{0j}$ then $\phi(\mathbf{x}, u) = (\mathbf{x} + \mathbf{e}_j) \wedge \mathbf{C}$;
- If $u \in A_{i0}$ then $\phi(\mathbf{x}, u) = (\mathbf{x} - \mathbf{e}_i) \vee 0$.

This transition function can be extended to any finite sequence $(u_1, u_2, \ldots, u_k) \in [0, 1]^k,\ k \in \mathbb{N}$, by defining $\phi : \mathcal{S} \times [0, 1]^k \to \mathcal{S}$ recursively:

$$\phi(\mathbf{x}, u_1, u_2, \ldots, u_k) \stackrel{\text{def}}{=} \phi(\phi(\mathbf{x}, u_1), u_2, \ldots, u_k).$$

2.2 Network with Infinite Buffers

Starting from $\mathcal{J}$, we construct a new network $\mathcal{J}^\infty$ that is identical to $\mathcal{J}$ except for the buffer sizes: in $\mathcal{J}^\infty$ all queues have infinite capacities. The state space of this new network will be denoted by $\mathcal{S}^\infty \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{Z}^M : x_i \geq 0, \forall i\}$.

We further assume here that each queue of the infinite JQN is stable, i.e., the total arrival rate at queue i λ_i, determined the traffic equation $\lambda_j = \sum_{i=0}^{M} \lambda_i p_{ij}$ for $1 \leqslant i \leqslant M$, satisfies $\lambda_i < \mu_i$ for $1 \leqslant i \leqslant M$.

The corresponding discrete-time Markov chain is denoted $(X_n^\infty)_{n\in\mathbb{N}}$. The evolution of the Markov chain $(X_n^\infty)_{n\in\mathbb{N}}$ can be written under the form $X_{n+1}^\infty = \phi^\infty(X_n^\infty, U_{n+1})$ where:

- If $u \in A_{ij}$ then $\phi^\infty(\mathbf{x}, u) = (\mathbf{x} - \mathbf{e}_i) \vee 0 + \mathbf{e}_j \mathbf{1}_{\mathbf{x}-\mathbf{e}_i \geqslant 0}$;
- If $u \in A_{0j}$ then $\phi^\infty(\mathbf{x}, u) = \mathbf{x} + \mathbf{e}_j$;
- If $u \in A_{i0}$ then $\phi^\infty(\mathbf{x}, u) = (\mathbf{x} - \mathbf{e}_i) \vee 0$.

Similarly as before, $\phi^\infty : \mathcal{S}^\infty \times [0,1]^k \to \mathcal{S}^\infty$ is defined recursively:

$$\phi^\infty(\mathbf{x}, u_1, u_2, \ldots, u_k) \stackrel{\text{def}}{=} \phi^\infty(\phi^\infty(\mathbf{x}, u_1), u_2, \ldots, u_k).$$

We consider the usual product partial order of states: for $\mathbf{x}, \mathbf{y} \in \mathcal{S}^\infty$,

$$\mathbf{x} \preceq \mathbf{y} \quad \text{iff} \quad x_i \leqslant y_i,\ 1 \leqslant i \leqslant M.$$

This new chain is an upper bound for the original system:

Lemma 1. *For any $u \in [0,1]$, and any $\mathbf{x} \in \mathcal{S}$, functions ϕ and ϕ^∞ satisfy:*

$$\phi(\mathbf{x}, u) \preceq \phi^\infty(\mathbf{x}, u).$$

Proof. This follows directly from the fact that for any $u \in [0,1]$, and any $\mathbf{x} \in \mathcal{S}$, $\phi(x, u) = \phi^\infty(x, u) \wedge \mathbf{C}$. □

To establish the comparison between the two chains, we also use the fact that ϕ^∞ is monotone in $\mathbf{x}$. (Actually, the original system is also monotone, but we do not need it for the proof of the bound.)

Lemma 2. *For any $u \in [0,1]$, and any $\mathbf{x}, \mathbf{y} \in \mathcal{S}^\infty$,*

$$\mathbf{x} \preceq \mathbf{y} \ \Rightarrow\ \phi^\infty(\mathbf{x}, u) \preceq \phi^\infty(\mathbf{y}, u).$$

The proof is straightforward.

Combining the two lemmas, we get the sample path comparison of the two systems:

Proposition 1. *For any $(u_{-t+1}, \ldots, u_0) \in [0,1]^t$, and any $\mathbf{x} \in \mathcal{S}$, $\mathbf{y} \in \mathcal{S}^\infty$,*

$$\mathbf{x} \preceq \mathbf{y} \ \Rightarrow\ \phi(\mathbf{x}, u_{-t+1}, \ldots, u_0) \preceq \phi^\infty(\mathbf{y}, u_{-t+1}, \ldots, u_0).$$

Proof. By induction on t. For $t = 1$, and any $\mathbf{x} \preceq \mathbf{y}$, using first Lemma 1 and then Lemma 2, we have $\phi(\mathbf{x}, u_0) \preceq \phi^\infty(\mathbf{x}, u_0) \preceq \phi^\infty(\mathbf{y}, u_0)$.

Assume now the induction statement is valid for $t-1$. Let $\mathbf{x} \preceq \mathbf{y}$, and denote $\mathbf{x}' = \phi(\mathbf{x}, u_{-t+1})$, and $\mathbf{y}' = \phi^\infty(\mathbf{y}, u_{-t+1})$. Using Lemmas 1 and 2, $\mathbf{x}' = \phi(\mathbf{x}, u_{-t+1}) \preceq \phi^\infty(\mathbf{x}, u_{-t+1}) \preceq \phi^\infty(\mathbf{y}, u_{-t+1}) = \mathbf{y}'$. Now $\phi(\mathbf{x}, u_{-t+1}, \ldots, u_0) = \phi(\mathbf{x}', u_{-t+2}, \ldots, u_0)$ and $\phi^\infty(\mathbf{y}, u_{-t+1}, \ldots, u_0) = \phi^\infty(\mathbf{y}', u_{-t+2}, \ldots, u_0)$, so

$$\phi(\mathbf{x}, u_{-t+1}, \ldots, u_0) \preceq \phi^\infty(\mathbf{y}', u_{-t+2}, \ldots, u_0)$$

by induction hypothesis. □

The new Markov chain $(X_n^\infty)_{n\in\mathbb{N}}$ has three interesting properties:

1. As mentioned above, the process $(X_n^\infty)_{n\in\mathbb{N}}$ is a bounding process of the original process $(X_n)_{n\in\mathbb{N}}$;
2. $(X_n^\infty)_{n\in\mathbb{N}}$ has a unique stationary distribution with the product form property:
$$\pi^\infty(x_1, \ldots, x_M) = \prod_{i=1}^{M} (1-\rho_i)\rho_i^{x_i}, \tag{1}$$
where ρ_i is the ratio of the total arrival rate in queue i, λ_i over the service rate μ_i. The arrival rate in queue i satisfies the routing balance equations: $\lambda_i = \sum_{j=0}^{M} p_{ji}\lambda_j$, for all i.
3. The construction of Markov chain $(X_n^\infty)_{n\in\mathbb{N}}$ can be reversed in time.

Reversed Chain. The reversed chain can also be modelled as a Jackson network with M queues with service rate μ_i in queue i, whose routing probabilities $q_{ij} \stackrel{\text{def}}{=} p_{ji}.\lambda_j/\lambda_i$, the probability that a customer leaves the system at queue i is $q_{i0} \stackrel{\text{def}}{=} p_{0i}.\lambda_0/\lambda_i$, and the exogenous arrival rate in queue j is $\theta_{0j} \stackrel{\text{def}}{=} p_{j0}\lambda_j$.

Similarly as for the forward construction of the Markov chain, we define events $b_{i,j}$ and corresponding intervals $B_{i,j}$, partitioning $[0,1]$, of length $\mu_i q_{ij}/\Gamma$ if $i \neq 0$ and θ_{0j}/Γ otherwise.

The backward transition function $\beta^\infty : \mathcal{S}^\infty \times [0,1] \to \mathcal{S}^\infty$ is defined as follows:

- If $u \in B_{ij}^\infty$ then $\beta^\infty(\mathbf{x}, u) = (\mathbf{x} - \mathbf{e}_i) \vee 0 + \mathbf{e}_j \mathbf{1}_{\mathbf{x}-\mathbf{e}_i \geqslant 0}$;
- If $u \in B_{0j}^\infty$ then $\beta^\infty(\mathbf{x}, u) = \mathbf{x} + \mathbf{e}_j$;
- If $u \in B_{i0}^\infty$ then $\beta^\infty(\mathbf{x}, u) = (\mathbf{x} - \mathbf{e}_i) \vee 0$.

3 Perfect Sampling Algorithm

One can use the chain X_n^∞ to obtain the bounds for the initial condition of the original chain X_n, as shown in Algorithm 1. Here is how this algorithm works. First, generate at time 0 a stationary sample Y^∞ of the infinite system according to the product form distribution π^∞. Starting from Y^∞, construct a backward trajectory up to time $-t$. At time $-t$, start a forward simulation using the same innovation sequence from states $X^{(2)}(-t) = 0$ and $X^{(1)}(-t) = Y^\infty(-t)$. If the

Algorithm 1. PSA with a Bounding Process (PSA-BP)

Data:
(1) An infinite i.i.d. sequence $\{U_{-n}\}_{n\in\mathbb{N}}$ with U_{-k} uniformly distributed in $[0,1]$.
(2) A state Y^∞ generated according to the distribution π^∞.
Result: A state sampled from the stationary distribution of $\mathcal{J}$
begin
  $t := 1$;
  repeat
    for $k = \lfloor t/2 \rfloor$ **to** $t-1$ **do**
      $Y^\infty := \beta^\infty(Y^\infty, U_{-k})$;
    $X^{(1)} := Y^\infty \wedge \mathbf{C}$; $X^{(2)} := 0$;
    for $k = t-1$ **downto** 0 **do**
      $X^{(1)} := \phi(X^{(1)}, U_{-i})$;
      $X^{(2)} := \phi(X^{(2)}, U_{-i})$;
    $t := 2t$;
  until $X^{(1)} = X^{(2)}$;
  return $X^{(1)}$;
end

trajectories do not coalesce at time 0 (*i.e.* $X^{(2)}(0)$ and $X^{(1)}(0)$ are not equal), then double the length of the simulation time to $2t$. As shown in the proof below, this does not introduce a bias in the output sample.

Theorem 1. *The Algorithm PSA-BP terminates with probability* 1. *The output of Algorithm PSA-BP is a state whose distribution is the stationary distribution over the network* $\mathcal{J}$.

Proof. We first assume that the Algorithm PSA-BP terminates with probability 1, and show that in that case, the output of Algorithm PSA-BP is a state whose distribution is the stationary distribution over the network $\mathcal{J}$. Then we will prove that the Algorithm PSA-BP terminates with probability 1.

Let us define the Markov chain $(Y_n^\infty)_{n\in\mathbb{N}}$ to be the reversed chain of $(X_n^\infty)_{n\in\mathbb{N}}$, and $(Y_n)_{n\in\mathbb{N}}$ to be the reversed chain of $(X_n)_{n\in\mathbb{N}}$.

Since X_n^∞ is a bounding process of X_n, the reversed process Y_n^∞ is also a bounding process of the reversed process Y_n. This implies that there exists a backward transition function $\beta : \mathcal{S} \times [0,1] \to \mathcal{S}$ for the chain Y_n such that for any $(u_0, \ldots, u_{-t+1}) \in [0,1]^t$,

$$\beta(\mathbf{x}, u_0, u_{-1}, \ldots, u_{-t+1}) \preceq \beta^\infty(\mathbf{y}, u_0, u_{-1}, \ldots, u_{-t+1}), \tag{2}$$

where $\beta(\mathbf{x}, u_0, \ldots, u_{-t+1}) \stackrel{\text{def}}{=} \beta(\beta(\mathbf{x}, u_0), u_{-1}, \ldots, u_{-t+1})$ is the usual extension of the function β to any finite sequence of innovations.

This backward transition function cannot be constructed explicitly in general. Such a construction is not needed here, only the existence property is needed.

In the proposed PSA-BP Algorithm, the variable $Y^\infty(0)$ is generated according to the stationary distribution of X_n^∞. It is possible to define $Y(0)$ such that

$Y(0) \preceq Y^{\infty}(0)$ and $Y(0)$ has the stationary distribution of Y_n (and of X_n). Now, for any deterministic time t, let us define $Y(-t) = \beta(Y(0), u_0, u_{-1}, \ldots, u_{-t+1})$, using the backward transition function β of the chain Y_n (see the blue dotted line in Figure 1), so that it also has the stationary distribution of X_n. By definition of β, from relation (2) one has $Y(-t) \preceq Y^{\infty}(-t)$. Now, starting from time $-t$ one has

$$Y^{\infty}(-t) = X^{(1)}(-t) \succeq Y(-t) \succeq X^{(2)}(-t) = 0.$$

Moving forward in time, this implies that

$$X^{(1)}(0) \succeq \phi(Y(-t), u_{-t+1}, \ldots, u_0) \succeq X^{(2)}(0).$$

Since, $\phi(Y(-t), u_{-t+1}, \ldots, u_0)$ has the stationary distribution of X_n, this is also the case for $X^{(1)}(0)$ and for $X^{(2)}(0)$, when they coalesce. This construction is illustrated in Figure 1.

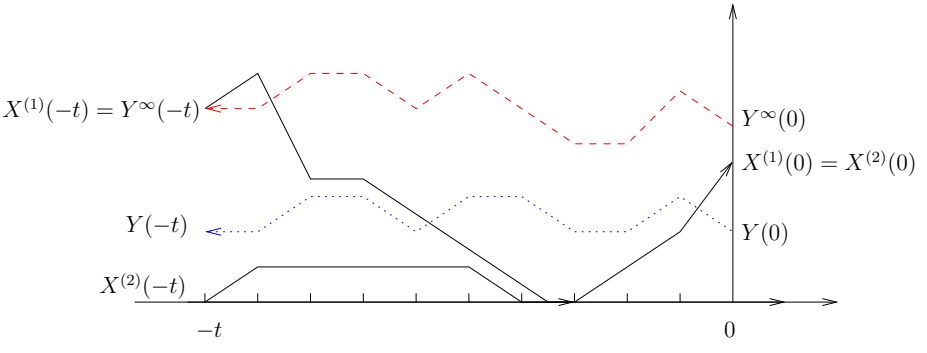


Fig. 1. Illustration of the proof. The variable $Y(-t)$ has the stationary distribution of X_n and is below $Y^{\infty}(-t)$.

The result being proved for any deterministic time t, the rest of the proof for finite buffers is similar to the proof of the classical perfect sampling algorithm (see for example the original proof in [11]).

Finally, we prove that the Algorithm PSA-BP terminates with probability 1. The random variable $Y^{\infty}(-t)$ is distributed according to the stationary distribution π^{∞}. Thus, the stability assumption for chain $(Y^{\infty})_{n\in\mathbb{N}}$ implies that we have $P(Y^{\infty}(-t) = 0) > 0$ (the zero state is to be understood componentwise). In that case, since $X^{(1)}$ and $X^{(2)}$ are both bounded by $(Y^{\infty})_{n\in\mathbb{N}}$, we have that $X^{(1)}(-t) = X^{(2)}(-t)$, so we also have $X^{(1)}(0) = X^{(2)}(0)$ and the algorithm terminates. By the Borel-Cantelli lemma, this happens almost surely in finite time so Algorithm PSA-BP terminates with probability 1. □

4 Complexity Analysis

The simulation time of PSA-BP can be decomposed into three steps:

1. The generation of a sample Y^∞ with the stationary distribution of the bounding process. Since this distribution is product form and since for each queue the geometric law can be easily sampled in constant time, the time complexity of this first step is $\mathcal{O}(M)$.
2. To obtain random events from the innovation sequence $\{U_{-n}\}_{n\in\mathbb{N}}$, we use the Walker's alias method [13] (for sampling discrete random variables in $\mathcal{O}(1)$ time). The construction of alias table is linear in number of events, so this pretreatment can be done in $\mathcal{O}(M^2)$ time. Given the alias tables for events $a_{i,j}$ and $b_{i,j}$, the values of $\phi(\mathbf{x}, u)$ and $\beta^\infty(\mathbf{x}, u)$ can be computed in $\mathcal{O}(1)$ time.
3. The construction of the backward trajectory of Y^∞ as well as the forward trajectories of $X^{(1)}$ and $X^{(2)}$ have the same expected duration, smaller than $12\mathbb{E}\tau(X^{(2)}, X^{(1)})$, where $\tau(X^{(2)}, X^{(1)})$ is the coalescence time of two trajectories of the Markov chain X_n, starting in $X^{(1)}$ and $X^{(2)}$ respectively, under the same sequence of events.

A queue $j \geq 1$ in a Jackson network $\mathcal{J}$ is *hyper-stable* if $\gamma_j \stackrel{\text{def}}{=} \sum_{i=0}^{M} \gamma_{ij} < \mu_j$. The network is hyper-stable if all queues $j \geq 1$ are hyper-stable. Note that hyper-stability implies stability since $\lambda_j \leqslant \gamma_j,\ j \geq 1$.

Theorem 2. *If $\mathcal{J}$ is hyper-stable, then the expected sampling time of PSA-BP satisfies $\mathbb{E}\tau(X^{(2)}, X^{(1)}) \leqslant c\sigma^2 M \sum_{j=1}^{M} S_j$, where c and σ are given in Equations (5) and (6) resp. and S_j is the expected stationary size of queue j of the infinite system $\mathcal{J}^\infty$.*

Proof. Starting from any state $\mathbf{x}$ and using any unit vector $\mathbf{e}_i$, let us consider the trajectories $X_1(t)$, $X_2(t)$ and $X_0(t)$ of the forward Markov chain starting respectively from states $\mathbf{x}+\mathbf{e}_i$, $\mathbf{x}$ and 0, under the same sequence of innovations $u_1, \ldots u_t, \ldots$. up to the time when they all coalesce. By definition, $X_1(0) = \mathbf{x}+\mathbf{e}_i$, $X_2(0) = \mathbf{x}$, $X_0(0) = 0$, and for all $t > 0$ and $i = 0, 1, 2$, $X_i(t) \stackrel{\text{def}}{=} \phi(X_i(t-1), u_t)$. At the coalescence time t_c, $X_1(t_c) = X_2(t_c) = X_0(t_c)$.

Let us also define the coalescence time t_1 of the first two trajectories: $X_1(t_1) = X_2(t_1)$ and t_0 of the last two trajectories: $X_2(t_2) = X_0(t_2)$. By definition, $t_1 \leqslant t_c$ and $t_0 \leqslant t_c$.

Now only two things can happen: either $X_2(t)$ and $X_0(t)$ meet first ($t_0 < t_1$) or $X_1(t)$ and $X_2(t)$ meet first ($t_1 \leqslant t_0$).

In the first case, the global coalescence time can be decomposed into t_0 plus the coalescence time of the two remaining states $X_1' \stackrel{\text{def}}{=} X_1(t_0)$ and $X_0' \stackrel{\text{def}}{=} X_0(t_0)$, equal to $t_c - t_0$. In the second case, clearly $t_0 = t_c$. In both cases, t_c can decomposed into $t_c = t_0 + (t_c - t_0)$ (that is degenerated in the second case). Using the definitions of the coalescence time, this equality can be re-written

$$\mathbb{E}\tau(\mathbf{x}+\mathbf{e}_i, 0) = \mathbb{E}\tau(\mathbf{x}, 0) + \mathbb{E}\tau(X_1', X_0'). \tag{3}$$

By inspecting all the possible events, it is easy to see that, since the L_1 distance between the original states $\mathbf{x}$ and $\mathbf{x}+\mathbf{e}_i$ is one, then the distance between X_1'

and X_0' is also at most one. This means that in the first case, there exists a unit vector $\mathbf{e}_k$, $k \in \{1, \ldots, M\}$ such that $X_1' = X_0' + \mathbf{e}_k$. In the second case, $X_0' = X_1'$, so that $\mathbb{E}\tau(X_1', X_0') = 0$. It was shown in [1] that if all queues are hyper-stable, for any state $\mathbf{x}$ and any unit vector e_k, then

$$\mathbb{E}\tau(\mathbf{x} + \mathbf{e}_k, \mathbf{x}) \leq cM \left(\frac{\max_i \frac{\gamma_i}{\mu_i}}{1 - \max_i \frac{\gamma_i}{\mu_i}} \right)^2, \tag{4}$$

where

$$c \stackrel{\text{def}}{=} \frac{2}{\min_{i:p_{i0}>0} p_{i0}}. \tag{5}$$

We also define

$$\sigma \stackrel{\text{def}}{=} \frac{\max_i \frac{\gamma_i}{\mu_i}}{1 - \max_i \frac{\gamma_i}{\mu_i}}. \tag{6}$$

Since this bound (4) is uniform in $\mathbf{x}$ and k, then using the recurrence equations (3) for any state $\mathbf{x}$,

$$\begin{aligned} \mathbb{E}\tau(\mathbf{x}, 0) &= \mathbb{E}\tau(\mathbf{x} - \mathbf{e}_i, 0) + \mathbb{E}\tau(X_1', X_0') && (7)\\ &\leqslant \mathbb{E}\tau(\mathbf{x} - \mathbf{e}_i, 0) + cM\sigma^2 && (8)\\ &= cM\sigma^2 \sum_i \mathbf{x}_i. && (9) \end{aligned}$$

Now, let us consider the sampling time of PSA-BP. As mentioned before, this time is less than $12\mathbb{E}\tau(X^{(1)}, 0)$, where $X^{(1)}$ is equal to state $Y^{\infty}(-t)$. At each time t, $Y^{\infty}(-t)$ is stationary and by monotonicity of the coalescence time with respect to its starting point, this means that if $t_c = \tau(X^{(1)}, 0)$ then $\mathbb{E}Y^{\infty}(-t_c) \leqslant (S_1, \ldots, S_M)$ where $S_i \stackrel{\text{def}}{=} \frac{\frac{\lambda_i}{\mu_i}}{1 - \frac{\lambda_i}{\mu_i}}$ is the expected size in queue i in the network $\mathcal{J}^{\infty}$. Therefore,

$$\mathbb{E}\tau(X^{(1)}, 0) \leqslant cM\sigma^2 \sum_{j=1}^{M} S_j. \tag{10}$$

5 Comparison with the Classical Perfect Sampler

As discussed earlier, our new algorithm allows one to get perfect sampling of Jackson networks with finite and infinite buffers. This was not possible with the classical perfect sampling algorithms. Actually, this new approach also brings improvements in the case where all buffers are finite. In that case, it reduces the sampling time by a factor corresponding to the ratio between the maximum capacity of the buffers over the expected size of the queues under the stationary law of the bounding process. This is detailed below.

Let us first recall the classical perfect sampler for monotone finite Markov chains derived from Jackson queueing networks.

Algorithm 2. Coupling from the past (PSA) [11]

Data: $\{U_{-n}\}_{n\in\mathbb{N}}$
Result: A state sampled from the stationary distribution of the JQN
begin
  $t := 1$;
  repeat
    $Z^{(1)} := \mathbf{C}$; $Z^{(2)} := 0$;
    for $i = t-1$ **downto** 0 **do**
      $Z^{(1)} := \phi(Z^{(1)}, U_{-i})$;
      $Z^{(2)} := \phi(Z^{(2)}, U_{-i})$;
    $t := 2t$;
  until $Z^{(1)} = Z^{(2)}$;
  return $Z^{(1)}$;
end

It is well known that Algorithm 2 outputs samples of the stationary distribution of the network as long as all buffers are finite. Otherwise, if $X^{(1)}$ has some infinite coordinates, the algorithm does not converge.

The time complexity in the case where $\mathcal{J}$ is hyper-stable with finite queues has been studied in [1]. The expected simulation time to get one sample is $\mathcal{O}(M^2 f(M) \sum_{i=1}^{M} C_i)$ where $f(M)$ is the average number of connections per node.

As shown in Theorem 2, the expected coalescence time of our new sampler is $\mathcal{O}(M\sigma^2 \sum_{i=1}^{M} S_i)$ (where S_i is the expected stationary size of queue i in the infinite system $\mathcal{J}^\infty$). This may result in a big improvement over the classical algorithm. Indeed, if the load in all queues is smaller than 0.8 (this is a typical situation), then $S \leqslant 4$, which is usually much smaller than the buffer size (typically of order 100, 1000 or more).

Finally, the coalescence time of PSA-BP is stochastically smaller than the coalescence time of PSA since the extreme starting point $X^{(1)}(-t) = Y^\infty(-t) \wedge \mathbf{C}$ of the former is always smaller than the starting time of the latter, $Z^{(1)}(-t) = \mathbf{C}$.

In the next section, we show on a numerical example that the PSA-BP algorithm 1 drastically outperforms the classical PSA Algorithm 2 with respect to efficiency.

6 Numerical Experiments

We have implemented both algorithms over a standard laptop computer and tested them over a queueing network where the M queues form a cycle, with the same service time in each queue, $\mu_i = \mu$, the same exogenous arrival rate in each queue (λ) and the same probability of leaving the system at each queue: $p_{i0} = p$.

In that case, the infinite-queue system is stable if and only if $\lambda/p < \mu$. We have run both simulators and we measured the coalescence time (number of steps t in Algorithm 1), as well as the simulation time (in seconds). Figures 2 and 3

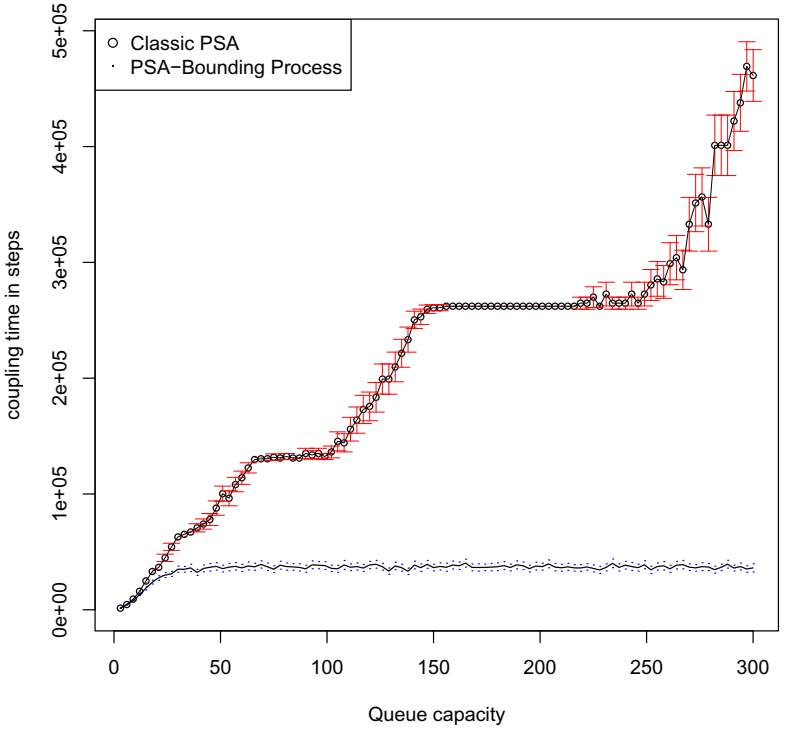


Fig. 2. Comparative coalescence time (number of steps) of Bounding-Process PSA and Classic PSA. Circular networks with M=50 queues, $\lambda = 4, \mu = 10, p_{i0} = 0.5$. The capacity C of each queue varies from 1 to 300.

respectively present those values (with 95 % confidence intervals computed with 100 experiments per setting) when the capacity of the queues ranges from 1 to 300. One can notice that the coalescence time as well as the simulation time of the classical perfect sampling algorithm grow linearly with the capacity of the buffers, as predicted in [1]. The succession of flat parts with steep parts comes from the doubling period scheme of the algorithm: the coalescence time of the algorithm is the smallest power of 2 larger than the actual coalescence time.

As for the new algorithm, the coalescence time remains bounded (as shown by Theorem 2) and the performance gap with the classical case becomes significant as soon as the capacity becomes larger than 10 (on the simulated example).

On an absolute basis, we think it is remarkable to sample the stationary distribution of such a queueing network, whose state space is $300^{50} \approx 10^{123}$ in less than 50 milliseconds over a standard laptop computer.

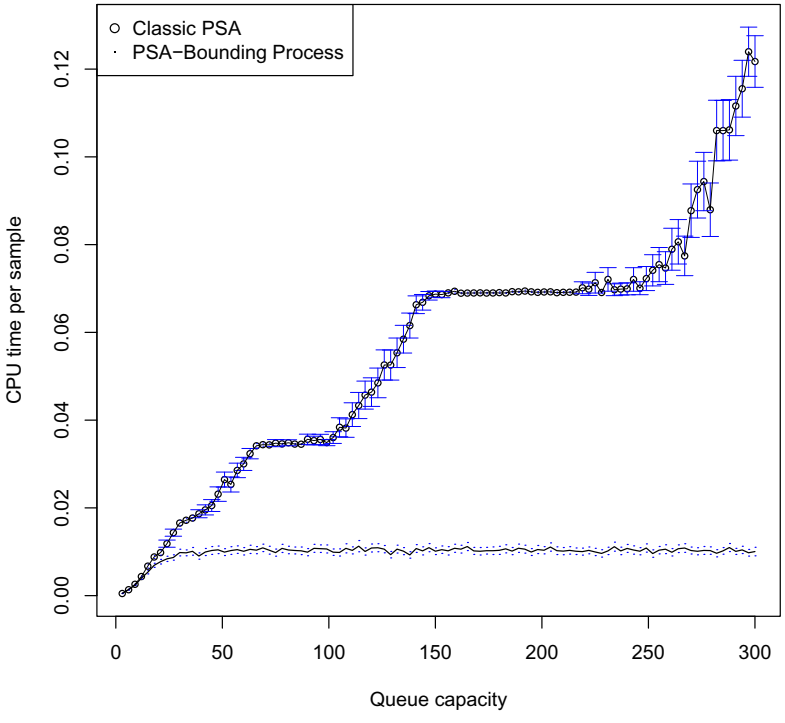


Fig. 3. Comparative sampling time (CPU time) of Bounding-Process PSA and Classic PSA. Circular networks with M=50 queues, $\lambda = 4, \mu = 10, p_{i0} = 0.5$. The capacity C of each queue varies from 1 to 300.

7 Conclusion

In this paper we have presented a new perfect sampler for Jackson queueing networks with finite and infinite capacities in queues. A complexity analysis of the algorithm shows that its expected sampling time does not depend on the capacities. This is a remarkable improvement over classical perfect samplers whose sampling time increase at least linearly in the capacities, or fails when some capacities are infinite. Actually our approach is quite general and can be used for any Markov chain for which a bounding process with a computable stationary distribution exists. In the future, we plan to construct similar bounding processes for a larger class of networks than the open Jackson networks with losses.

Bibliography

1. Anselmi, J., Gaujal, B.: On the efficiency of perfect simulation in monotone queueing networks. In: IFIP Performance: 29th International Symposium on Computer Performance, Modeling, Measurements and Evaluation, Amsterdam (October 2011), ACM Performance Evaluation Review

2. Bolch, G., Greiner, S., de Meer, H., Trivedi, K.: Queueing Networks and Markov Chains. Wiley-Interscience (2005)
3. Busic, A., Gaujal, B., Vincent, J.-M.: Perfect simulation and non-monotone Markovian systems. In: 3rd International Conference Valuetools 2008, Athens, Greece. ICST (October 2008)
4. Dopper, J., Gaujal, B., Vincent, J.-M.: Bounds for the coupling time in queueing networks perfect simulation. Celebration of the 100th Anniversary of Markov, pp. 117–136 (2006)
5. Gaujal, B., Gorgo, G., Vincent, J.-M.: Perfect Sampling of Phase-Type Servers Using Bounding Envelopes. In: Al-Begain, K., Balsamo, S., Fiems, D., Marin, A. (eds.) ASMTA 2011. LNCS, vol. 6751, pp. 189–203. Springer, Heidelberg (2011)
6. Jackson, J.R.: Job shop-like queueing systems. Management Sci. 10, 131 (1963)
7. Kelly, F.: Reversibility and Stochastic Networks. John Wiley & Sons Ltd (1979)
8. Kendall, W.S.: Notes on perfect simulation. Dept. of statistics, University of Warwick (2005)
9. Pin, F., Busic, A., Gaujal, B.: Perfect sampling of Markov chains with piecewise homogeneous events. Technical Report 163442, arXiv (2010)
10. Pin, F., Busic, A., Gaujal, B.: Acceleration of perfect sampling for skipping events. In: Valuetools, Paris (2011)
11. Propp, J.G., Wilson, D.B.: Exact sampling with coupled markov chains and applications to statistical mechanics. Rand. Struct. Alg. 9(1-2), 223–252 (1996)
12. Vincent, J.-M.: Perfect simulation of monotone systems for rare event probability estimation. In: WSC 2005: Proceedings of the 37th Conference on Winter Simulation Conference, pp. 528–537 (2005)
13. Walker, A.J.: An efficient method for generating discrete random variables with general distributions. ACM Trans. Math. Softw. 3, 253–256 (1977)

A Queueing Theoretic Approach to Decoupling Inventory

Eline De Cuypere, Koen De Turck, and Dieter Fiems

Department of Telecommunications and Information Processing, Ghent University,
St-Pietersnieuwstraat 41, B-9000, Belgium
{eline.decuypere,koen.deturck,dieter.fiems}@telin.ugent.be

Abstract. This paper investigates the performance of different hybrid push-pull systems with a decoupling inventory at the semi-finished products and reordering thresholds. Raw materials are 'pushed' into the semi-finished product inventory and customers 'pull' products by placing orders. Furthermore, production of semi-finished products starts when the inventory goes below a certain level, referred to as the threshold value and stops when the inventory attains stock capacity. As performance of the decoupling stock is critical to the overall cost and performance of manufacturing systems, this paper introduces a Markovian model for hybrid push-pull systems. In particular, we focus on a queueing model with two buffers, thereby accounting for both the decoupling stock as well as for possible backlog of orders. By means of numerical examples, we assess the impact of different reordering policies, irregular order arrivals, the set-up time distribution and the order processing time distribution on the performance of hybrid push-pull systems.

1 Introduction

In a make-to-stock system (push type), products are stocked in advance, while in a make-to-order system (pull type), a product only starts to be manufactured when a customer order is placed, see a.o [24,16,12,25,7]. Nowadays, as a means to respond quickly to growing variety, shorter product life cycles while keeping inventory costs as low as possible, hybrid push-pull systems are introduced [23]. An important issue in the overall performance of such hybrid systems is the position of the decoupling point [23,20]. Hoekstra et. al [10] defined the customer order decoupling point (CODP) concept. These authors considered market, product and production related factors as well as the desired service level and associated inventory costs to locate the optimal decoupling point. Under different hybrid push/pull control policies, Pandey and Khokhajaikiat [19] conducted a case study concerning the design and performance evaluation of a multistage production system. Results indicated that the choice of the optimal decoupling positions changes with the extent of raw material constraint operating at the stages and the demand lead time variabilities. To account for a degree of customisation and short delivery times, Blecker and Abdelkafi [2] considered a decoupling point at the inventory of semi-finished products. Here, after an order

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 150–164, 2012.

is received, only the final completion step still needs to be done. A case study at Phoenix showed that, by a hybrid approach, the company would save 20 to 25 percent of the total late costs and inventory costs compared to a pure push approach, which was at that time being used [5]. Research on the performance of the decoupling inventory in a hybrid push-pull system is therefore of main importance. This is the subject of the present paper.

In the present setting, we use a queuing theoretic approach to study the hybrid push-pull system. Queuing theory has already been successfully applied to assess decoupling points. Kaminsky and Kaya [13] considered a variety of combined make-to-order (MTO) and make-to-stock (MTS) supply chains with a single manufacturer and a single supplier in order to minimise a function of the total inventory, lead times and tardiness. The arrival process at the manufacturer is treated as a single facility with multiple classes of Poisson arrivals scheduled FCFS. As in previous research, they concluded that costs can be cut dramatically by using a combined system instead of pure MTO or MTS systems. Ohta and al. [18] analysed a multi-product inventory system where demand for each item arrive according to a Poisson process and the production time has an Erlang distribution. An optimality condition that specifies whether each product should be produced MTS or MTO is proposed. Bell [1] investigated a decoupling inventory between two successive production stages, the demand at stage 2 being independent from production at stage 1. The stages are decoupled by storing intermediate products. Limits on the available storage capacity and the rates of flow production into and out of the decoupling inventory are set, which enables the firm to determine the optimum capacities for the storage facility and to determine the value of an additional supply of intermediate product. Chang and Lu [4] studied a one-station production system consistent with MTO and MTS productions and dealing with two types of random demands: ordinary demand and specific demand. In this system, both types of demand arrive according to a Poisson process and production times of the workstation are exponentially distributed. Specific demand has a higher priority with respect to ordinary demand and the performance of this system is studied by means of matrix-geometric methods.

The present study of the decoupling stock closely relates to literature on two-part assembly systems, sometimes termed paired queues or kitting processes. For such systems, there are two queues, each storing a specific part, and production only starts when both part buffers are non-empty. In the current setting, one part-buffer corresponds to the decoupling stock, while the other corresponds to the list of backlogged orders. Also, production only starts when both buffers are non-empty. Indeed, each delivery of a finished product requires both the order specifications and a semi-finished product and can only be satisfied if both are present. If both part-buffers have unlimited capacity, Harrison [9] was the first to prove that, assuming no arrival control strategy, this queueing system is inherently unstable. In particular, he studied the multiple-input extension of the GI/G/1 queue in which arrivals in each stream are described by an independent renewal process and service times are independent and identically distributed.

He showed that part waiting times converge to non-defective limiting distributions only if the buffer capacities are bounded. This was also demonstrated by Latouche [14] who termed the two-part assembly system as waiting lines with paired customers. He considered a system of infinite capacity queues with Poisson arrivals for both parts and exponential services. The steady state is attained, i.e. the system is stable, if the arrival rates depend on the difference between queue lengths. [3] extended Latouche's research by considering two exponential distributions, one for the part processing distribution, i.e. the synchronisation phase, and the other for the assembly operation distribution. Approximations for the throughput rate and average queue length were given. Lipper and Sengupta [17] is another extension of the work of Latouche. In this paper, multiple Poisson input streams arrive in buffers with finite capacity. A more general structure in which parts are withdrawn from infinite pools and processed prior to assembly has been studied by Hopp et. al [11] and Som et. al [22]. Som and Wilhem [21] studied a two-queue system in which each part is processed according to an exponential distribution and the assembly operation times are generally distributed. They follow a matrix-geometric approach to numerically determine the marginal distributions of both kit and end-product inventory positions. Finally, assuming finite part-buffers, a two-part assembly system in a Markovian environment is studied in [6] by numerically solving the corresponding Markov chains by the generalized minimal residual method.

Furthermore, this article analyses hybrid push-pull systems with a threshold inventory: once the stock of semi-finished products drops below some level, this is either communicated to the production department if the parts are produced in-house or an order is placed with a third-party company if this is not the case. In both cases, it may take some time, the reordering time, before the inventory is replenished. Then, production stops when the semi-finished product inventory level attains stock capacity. The studied inventory control system closely relates to the well-known economic order quantity (EOQ) model [8]. This is a deterministic fluid-model for a single inventory and determines optimal reordering policies which balance the purchase, order and storage costs. While the single-part inventory problem is well understood, both in a deterministic and a stochastic setting, many issues of optimal inventory management in the multi-queue inventory case remain unresolved, most prominently in the stochastic setting.

In contrast to previous research, this paper investigates a two-queue system with one finite and one infinite buffer. Indeed, to limit involved costs, the decoupling stock needs to be sufficiently small. Hence, finite capacity is assumed. However, no such assumption is imposed for the other queue: the order backlog queue has an infinite capacity. Assuming a finite capacity product queue also assures the existence of a steady-state solution, provided that the arrival rate of orders is limited. In particular, this article analyses hybrid push-pull systems under different threshold policies, assuming that production stops when the inventory level reaches maximum capacity. Comparing versatility and numerical tractability, we study the decoupling stock in a Markovian environment as in [6]. This approach allows for assessing the effect of variability in the production

process of semi-finished products, the ordering process and the delivery process on the performance of the decoupling stock.

The remainder of this paper is organised as follows. Section 2 describes the decoupling stock model at hand. In section 3, the decoupling inventory system is analysed as a quasi-birth-and-death-process (QBD) and a number of specific application scenarios for the decoupling inventory system are introduced. Also, the numerical solution methodology is discussed and relevant performance measures are determined. To illustrate our approach, section 4 considers some numerical examples. Finally, conclusions are drawn in section 5.

2 Model Description

The decoupling stock is modelled as a queueing model with two queues, as depicted in Figure 1. The product queue has finite capacity C_p and stores the semi-finished products prior to being processed to finished products. Moreover, production of semi-finished starts when the inventory goes below the threshold value T_p and stops when the inventory level reaches capacity C_p. The order queue keeps track of the orders that have not yet been delivered and has infinite capacity. Arriving orders are served in accordance with a first-come-first-served queueing discipline. Each order takes a semi-finished product from the product queue and completes the product in accordance with order specifications. Note that the two queues in the model at hand are tightly coupled. Departures from the product queue are only possible when there are orders. Similarly, departures from the order queue are only possible if there are semi-finished products in the product queue.

Arrivals at both queues are modelled according to possibly dependent arrival processes and order completion is not instantaneous. For ease of modelling, it is assumed that there is a modulating Markov chain, arrival and service rates depending on the state of this modulating chain. To be more precise, the decoupling inventory system is a three-dimensional continuous-time Markov Chain with infinite state space $\mathbb{N} \times \{0, 1, 2, \ldots, C_p\} \times \mathcal{K}$, $\mathcal{K} = \{0, 1, \ldots, K\}$ being the

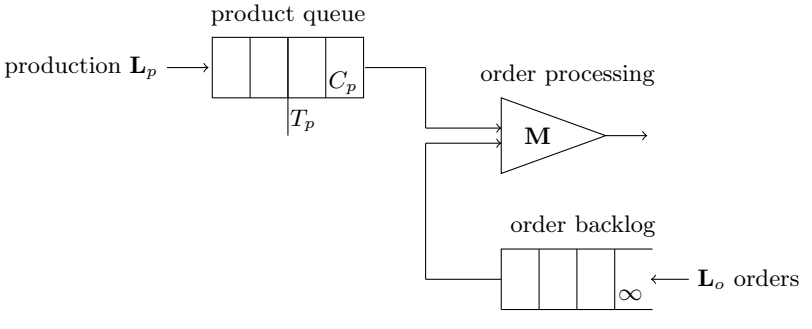


Fig. 1. Decoupling inventory of semi-finished products in a hybrid push-pull system

state space of the modulating chain. At any time, the state of the decoupling inventory system is described by the triplet $[n, m, i]$, n being the number of backlogged orders, m being the number of semi-finished products and i being the state of the modulating chain. We now describe the state transitions.

- The state of the modulating chain can change when there are neither arrivals nor departures. Let α_{ij} denote the transition rate from state i to state j $(i, j \in \mathcal{K}, i \neq j)$. Further, for ease of notation, let

$$\alpha_{ii} = -\sum_{j \neq i} \alpha_{ij}.$$

  and let $\mathbf{A} = [\alpha_{ij}]_{i,j \in \mathcal{K}}$ denote the corresponding generator matrix. Further, it is assumed that when either of the queues is empty, different transition rates (when there are neither arrivals nor departures) can be specified: let $\hat{\alpha}_{ij}$ and $\hat{\mathbf{A}}$ denote the transition rate from state i to state j and the corresponding generator matrix, respectively.
- The state of the modulating chain may remain the same or may change when there is an arrival. Let $\lambda_{ij}^{(p)}$ and $\lambda_{ij}^{(o)}$ denote the (marked) transition rate from state i to state j when there is an arrival at the product queue and the order queue, respectively. Moreover, let $\boldsymbol{\Lambda}_p = [\lambda_{ij}^{(p)}]_{i,j \in \mathcal{K}}$ and $\boldsymbol{\Lambda}_o = [\lambda_{ij}^{(o)}]_{i,j \in \mathcal{K}}$ denote the corresponding generator matrices. Note that marked self transitions from state i to state i are allowed.
- Analogously, the state of the modulating chain may remain the same or may change when there is a departure (in each buffer). Let μ_{ij} and $\underline{\mathbf{M}}$ denote the corresponding transition rate and generator matrix respectively.

Remark 1. The transition rates are dependent on the product queue size, the state of the modulating chain and whether the order queue is empty, e.g. there are no product arrivals when the queue is full, production starts only when the semi-finished product inventory level goes below the threshold value and there are only departures if both queues are non-empty.

3 Analysis

3.1 Quasi-Birth-Death Process

The studied Markov process is a homogeneous quasi-birth-and-death process (QBD), see [15]. In the present setting, the so-called level or block-row number, indicates the number of backlogged orders while the phase, i.e. the index within a block element, indicates both the content of the decoupling stock and the state of the Markovian environment. The one-step transitions are restricted to states in the same level (from state $(n, *, *)$ to state $(n, *, *)$) or in two adjacent levels (from state $(n, *, *)$ to state $(n + 1, *, *)$ or state $(n - 1, *, *)$).

We then find that the generator matrix of the Markov chain has the following block matrix representation,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{L}'_p & \mathbf{L}_o & \mathbf{0} & 0 & \cdots \\ \mathbf{W} & \mathbf{L}_p & \mathbf{L}_o & 0 & \cdots \\ 0 & \mathbf{W} & \mathbf{L}_p & \mathbf{L}_o & \cdots \\ 0 & 0 & \mathbf{W} & \mathbf{L}_p & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{1}$$

The blocks are given by,

$$\mathbf{L}_o = \begin{bmatrix} \boldsymbol{\Lambda}_o^{(0)} & 0 & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Lambda}_o^{(1)} & 0 & \cdots & 0 \\ 0 & 0 & \boldsymbol{\Lambda}_o^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \boldsymbol{\Lambda}_o^{(C_p)} \end{bmatrix}, \quad \mathbf{L}_p = \begin{bmatrix} \underline{\mathbf{D}}^{(0)} & \boldsymbol{\Lambda}_p^{(0)} & 0 & \cdots & 0 \\ 0 & \mathbf{D}^{(1)} & \boldsymbol{\Lambda}_p^{(1)} & \cdots & 0 \\ 0 & 0 & \mathbf{D}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{D}^{(C_p)} \end{bmatrix}, \tag{2}$$

$$\mathbf{L}'_p = \begin{bmatrix} \underline{\mathbf{D}}^{(0)} & \boldsymbol{\Lambda}_p^{(0)} & 0 & \cdots & 0 \\ 0 & \underline{\mathbf{D}}^{(1)} & \boldsymbol{\Lambda}_p^{(1)} & \cdots & 0 \\ 0 & 0 & \underline{\mathbf{D}}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \underline{\mathbf{D}}^{(C_p)} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \mathbf{M}^{(1)} & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{M}^{(2)} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathbf{M}^{(C_p)} & 0 \end{bmatrix}. \tag{3}$$

with $\mathbf{D}^{(m)} = \mathbf{A}^{(m)} - \partial\boldsymbol{\Lambda}_o^{(m)} - \partial\boldsymbol{\Lambda}_p^{(m)} - \partial\mathbf{M}^{(m)}$ and $\underline{\mathbf{D}}^{(m)} = \hat{\mathbf{A}}^{(m)} - \partial\boldsymbol{\Lambda}_o^{(m)} - \partial\boldsymbol{\Lambda}_p^{(m)}$ with $m = (0, 1, 2, \ldots, C_p)$ being the number of semi-finished products in the buffer. Note that $\partial\mathbf{X}$ represents a diagonal matrix with diagonal elements equal to the row sums of $\mathbf{X}$. Intensities in the generator matrices Λ_o, Λ_p, $\underline{\mathbf{D}}$, $\mathbf{D}$ and $\mathbf{M}$ are dependent of the product buffer content m. Therefore, we introduce the superscript $^{(m)}$ to make this dependence explicit. Note that if no superscript is indicated, the intensities in the generator matrix are equal for all numbers of semi-finished products in the queue.

To simplify notation, states representing an inactive production and a product queue content equal or less than the threshold value, are taken into account in the generator matrix structure. However, as production is always active when the semi-finished product inventory level is below the threshold value, the next transition changes the given inactive background state to an active one. The matrix structure also considers states where the number of semi-finished products equals capacity and the background state is active. Again, the next transition changes the background state into an inactive state.

In the general case, arrivals and departures at both queues are modelled according to possibly dependent Markovian arrival processes (MAP) and phase-type distributed order processing times, respectively. The Markovian arrival processes are described by the generator matrices $\mathbf{B}_1^{(m)}$ and $\mathbf{B}_3^{(m)}$ with arrivals of semi-finished products and orders respectively and the generator matrices $\mathbf{B}_0^{(m)}$

and $\mathbf{B}_2^{(m)}$ without arrivals at the decoupling stock and the queue of backlogged orders respectively. The phase-type distribution is completely characterised by an initial probability vector $\boldsymbol{\tau}$ and the matrix $\mathbf{T}$ which corresponds to non-absorbing transitions [15]. Let $\mathbf{t}' = -\mathbf{T}\mathbf{e}'$ be the column vector with the rates to the absorbing state with $\mathbf{e}$ a row vector of ones. We have,

$$\boldsymbol{\Lambda}_p = \mathbf{B}_1^{(m)}, \quad \boldsymbol{\Lambda}_o = \mathbf{B}_3^{(m)}, \quad \mathbf{A} = \mathbf{B}_0^{(m)} + \mathbf{B}_2^{(m)} + \mathbf{T}, \quad \hat{\mathbf{A}} = \mathbf{B}_0^{(m)} + \mathbf{B}_2^{(m)}, \quad \mathbf{M} = \mathbf{t}'\boldsymbol{\tau}.$$

Before proceeding, we introduce a number of specific application scenarios of the decoupling inventory system at hand.

Example 1. In the most basic setting, when the semi-finished product inventory level goes below a given threshold value, semi-finished products arrive in the queues in accordance with an independent Poisson process with parameter λ_p and production stops when the stock capacity is reached. Orders arrive according to an independent Poisson process with parameter λ_o and order processing times are exponentially distributed with parameter μ. In this case, the modulating state indicates whether the production of semi-finished products is active or not. We have,

$$\boldsymbol{\Lambda}_p = \lambda_p \mathbf{I}, \quad \boldsymbol{\Lambda}_o = \lambda_o \mathbf{I}, \quad \mathbf{M} = \mu \mathbf{I}, \quad \mathbf{A} = \hat{\mathbf{A}} = \mathbf{0}.$$

Here $\mathbf{I}$ denotes the identity matrix.

Example 2. To account for variability in the production times of semi-finished products, we consider a Markovian arrival process with the generator matrices $B_0^{(m)}$ and $B_1^{(m)}$ as described above. Assuming Poisson arrivals of orders with parameter λ_o and order processing times exponentially distributed with parameter μ, we have,

$$\boldsymbol{\Lambda}_p = \mathbf{B}_1^{(m)}, \quad \boldsymbol{\Lambda}_o = \lambda_o \mathbf{I}, \quad \mathbf{A} = \hat{\mathbf{A}} = \mathbf{B}_0^{(m)}, \quad \mathbf{M} = \mu \mathbf{I}.$$

Example 3. Unreliability in the ordering process can also be modelled by a Markovian arrival process. Here, the MAP is described by the generator matrix $\mathbf{B}_3^{(m)}$ of transitions with order arrivals and the generator matrix $\mathbf{B}_2^{(m)}$ without arrivals. Retaining exponentially distributed order processing times and assuming Poisson arrivals of semi-finished products, we have,

$$\boldsymbol{\Lambda}_p = \lambda_p \mathbf{I}, \quad \boldsymbol{\Lambda}_o = \mathbf{B}_3^{(m)}, \quad \mathbf{A} = \hat{\mathbf{A}} = \mathbf{B}_2^{(m)}, \quad \mathbf{M} = \mu \mathbf{I}.$$

Example 4. As for the arrival processes, the model at hand is sufficiently flexible to include phase-type distributed order processing times. The phase-type distribution is completely characterised by an initial probability vector $\boldsymbol{\tau}$ and the matrix $\mathbf{T}$ which corresponds to non-absorbing transitions [15]. Let $\mathbf{t}' = -\mathbf{T}\mathbf{e}'$ be the column vector with the rates to the absorbing state with $\mathbf{e}$ a row vector of ones. Assuming Poisson arrivals in both buffers (with rate λ_p and λ_o, respectively), we get the following matrices,

$$\boldsymbol{\Lambda}_p = \lambda_p \mathbf{I}, \quad \boldsymbol{\Lambda}_o = \lambda_o \mathbf{I}, \quad \mathbf{A} = \mathbf{T}, \quad \hat{\mathbf{A}} = \mathbf{0} \quad \mathbf{M} = \mathbf{t}'\boldsymbol{\tau}.$$

3.2 Methodology: The Matrix-Geometric Technique

Consider the above defined Markov process on the three-dimensional state space $\{(n, m, i) \mid n \geq 0, 0 \leq m \leq C_p, i = 0, 1, \ldots, K\}$ where i denotes the state of the modulating chain, as the phase set i is defined in the finite state space $\mathcal{K}$ (see section 2). A well-known method for finding the stationary distribution of QBD processes is the matrix-geometric method. With $\pi(n, m, i)$ the stationary probability of the process being in state (n, m, i), and using the vector notation $\boldsymbol{\pi}_k = (\pi(k, 0, 0), \pi(k, 0, 1), \ldots, \pi(k, C_p, K))$, the probability vectors can be expressed as,

$$\boldsymbol{\pi}_k = \boldsymbol{\pi}_0 \mathbf{R}^k. \tag{4}$$

where the so-called rate matrix $\mathbf{R}$ is the minimal non-negative solution of the non-linear matrix equation $\mathbf{R}^2\mathbf{W} + \mathbf{R}\mathbf{L}_p + \mathbf{L}_o = \mathbf{0}$. Here, we compute the rate matrix by implementing the iterative algorithm of [15, chapter 8].

3.3 Performance Measures

Once the steady state probabilities have been determined numerically, we can calculate a number of interesting performance measures for the decoupling inventory system. For ease of notation, we introduce the marginal probability mass functions of the queue content of the product queue and the order queue: $\pi^{(p)}(m) = \sum_{i \in \mathcal{K}} \sum_{n=0}^{\infty} \pi(n, m, i)$ and $\pi^{(o)}(m) = \sum_{i \in \mathcal{K}} \sum_{m=0}^{C_p} \pi(n, m, i)$.

Note that as the queue of the backlogged orders is infinite, the throughput of the decoupling inventory system η equals the order arrival rate λ_o. In addition, we have the following performance measures.

- The mean semi-finished product queue and the order backlog content: $\mathrm{E}\, Q_p$ and $\mathrm{E}\, Q_o$ respectively,

$$\mathrm{E}\, Q_p = \sum_{m}^{C_p} \pi^{(p)}(m) m\,, \quad \mathrm{E}\, Q_o = \sum_{n}^{\infty} \pi^{(o)}(n) n\,.$$

- The variance of the semi-finished product queue and the order backlog content: $\mathrm{Var}\, Q_p$ and $\mathrm{Var}\, Q_o$ respectively,

$$\mathrm{Var}\, Q_p = \sum_{m}^{C_p} \pi^{(p)}(m) m^2 - (\mathrm{E}\, Q_p)^2\,,$$

$$\mathrm{Var}\, Q_o = \sum_{n}^{\infty} \pi^{(o)}(n) n^2 - (\mathrm{E}\, Q_o)^2\,.$$

- The mean lead time LT (calculated based on Little's theorem) is the average amount of time between the placement of an order and the completion to a finished product:

$$\mathrm{LT} = \frac{\mathrm{E}\, Q_o}{\lambda_o}$$

– As the product queue has finite capacity, production prior to the decoupling stock may be blocked. This happens when there is a product arrival and the queue is full. Hence, blocking corresponds to the loss probability in the product buffer. The loss probability is most easily expressed in terms of the throughput η. We have,

$$b_p = \frac{\lambda_p - \eta}{\lambda_p} = \frac{\lambda_p - \lambda_o}{\lambda_p}.$$

We now illustrate our approach by means of some numerical examples.

4 Numerical Examples

4.1 Poisson Arrivals and Exponential Order Processing Times

As a first example, the difference between the mean semi-finished product queue and the mean order backlog content versus the threshold value of the semi-finished product inventory is depicted in figure 2(a). We assume that semi-finished products and orders arrive according to a Poisson process with parameter $\lambda_p = 1$ and $\lambda_o = 0.85$, respectively. The inventory capacity C_p equals 20 and order processing times are exponentially distributed with service rate μ equal to 1 for all curves. The described model is a decoupling inventory system with Poisson arrivals and exponential order processing times as described in example 1 of section 3. As the figure shows, the threshold value of 5 results on average in no backlogged orders and no semi-finished products in stock. Under and above this level, products and orders are on average backlogged, respectively. Obviously, there is on average more stock of semi-finished products and less backlog of orders as the threshold value increases.

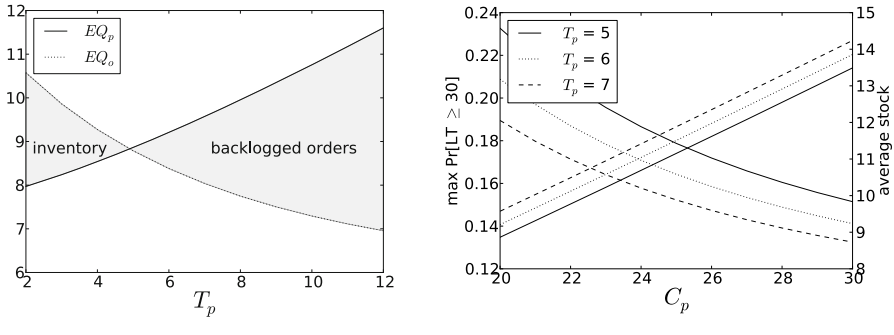


Fig. 2. There is a trade-off between the average stock of the semi-finished products and the average number of backlogged orders and between the lead time

Figure 2(b) represents the trade-off between the maximum probability to have the lead time higher or equal to 30 (left side) and the average stock of the semi-finished products (right side). Note that we calculated the lead time distribution

by using the one-sided Chebyshev's inequality. Under the same parameter assumptions of figure 2(a), the maximum probability to have the lead time higher or equal to 30 decreases and the average stock increases as the inventory capacity increases for each threshold value. Indeed, if more buffer capacity is available, it will be used – the mean semi-finished product queue increases such that there is on average less time required to deliver an order. Finally, in this numerical example, we observe that the highest threshold value give the average best results: the intersection between the two performance measures and the necessary stock capacity have the lowest value.

4.2 Erlang Distributed Set-Up Times

The second numerical example quantifies the impact of variability in the production process of semi-finished products on the decoupling inventory system. In particular, we here study Erlang-distributed set-up times – the set-up time starts when the semi-finished product inventory goes below a certain level and stops after some Erlang distributed time. Then, the semi-finished products arrive according to a Poisson process with arrival rate λ_p until the stock capacity is reached. The described model is a decoupling inventory system with Markovian arrivals as described in example 2 of section 3.

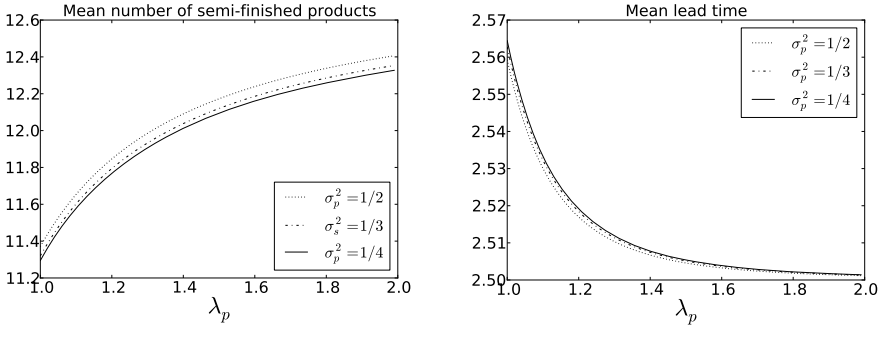


Fig. 3. The shape of the set-up time distribution has a small effect on the mean number of semi-finished products and on the mean lead time

Figure 3(a) 3(b) show the mean number of semi-finished products in the buffer and the mean lead time of the system with a buffer capacity equal to 20 and a threshold value equal to 5. In both figures, the arrival rate is varied and different values of the variance of the set-up time process are assumed as indicated. The order arrival rate λ_o equals 0.6, order processing times are assumed to be exponentially distributed with service rate μ equal to 1 and the mean set-up time equals 1. As expected, the mean number of semi-finished products increases and the mean lead time decreases as the arrival rate of the semi-finished products

λ_p increases. Furthermore, the shape of the set-up time distribution has a very small effect on both performance measures. In particular, the mean number of semi-finished products and the mean lead time show respectively a slight decrease and increase as the variance of the set-up time distribution σ_p^2 increases. This is due to the fact that the regularly the set-up time, the less semi-finished products are on average in stock and the more orders are on average backlogged.

4.3 Markovian Arrival Process for Orders

We also quantify the impact of irregular order arrivals. To this end, we compare both buffers with Poisson arrivals to corresponding decoupling inventory systems with interrupted Poisson arrivals for the orders and Poisson arrivals for the semi-finished products. The arrival interruptions account for inefficiency in the ordering process. Order processing times are assumed to be exponentially distributed with service rate μ equal to 1, this value being independent of the number of products and orders in the different buffers. This numerical example fits example 3 of section 3.

The interrupted Poisson process considered here is a two-state Markovian process. In the active state, new orders arrive in accordance with a Poisson process with rate λ_o whereas no new orders arrive in the inactive state. Let α and β denote the rate from the active to the inactive state and vice versa, respectively. We then use the following parameters to characterise the interrupted Poisson process (IPP),

$$\sigma = \frac{\beta}{\alpha+\beta}\,, \quad \kappa = \frac{1}{\alpha} + \frac{1}{\beta}\,, \quad \rho_o = \lambda_o \sigma\,.$$

Note that σ is the fraction of time that the interrupted Poisson process is active, the absolute time parameter κ is the average duration of an active and an inactive period, and ρ_o is the arrival load of the orders.

Figure 4 shows the mean number of backlogged orders versus the arrival rate of semi-finished products with buffer capacity C_p equal to 20 and the threshold value T_p equal to 5 and 7 for Poisson arrivals as well as for interrupted Poisson arrivals of orders. Order processing times are exponentially distributed with service rate μ equal to one for all curves. In addition, we set $\sigma = 0.8$ and $\kappa = 10$ for the interrupted Poisson processes (λ_o equals 0.6 for Poisson arrivals and 0.75 for interrupted Poisson arrivals). As expected, the mean number of backlogged orders decreases as the arrival rate of semi-finished products increases. Furthermore, the impact of the threshold value on the average number of backlogged orders decreases as the arrival rate of semi-finished products increases – both Markovian models converge to some value for T_p equal to 5 and 7. Finally, comparing interrupted Poisson and Poisson processes, burstiness in the ordering process has a negative impact on performance – there is on average more time required to deliver an order.

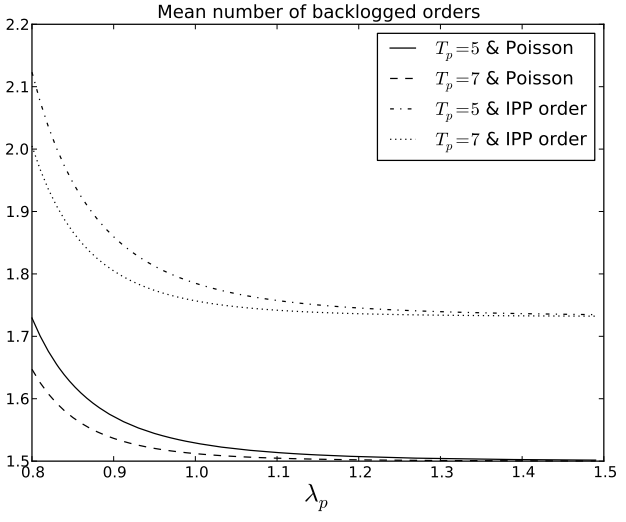


Fig. 4. Irregular order arrivals result in a higher average number of backlogged orders

4.4 Phase-Type Distributed Order Processing Times

The last numerical example quantifies the impact of the distribution of the order processing times on the decoupling inventory performance. In particular, we here study Erlang-distributed order processing times. This numerical example fits example 4 of section 3.

Figure 5(a) and 5(b) depict the mean number of semi-finished products in the buffer and the mean lead time of the decoupling inventory system. In both figures, the arrival rate of semi-finished products is varied and different values of the order processing time distribution are assumed as indicated. The service rate μ equals 1 for all curves, the order arrival rate λ_o equals 0.6, the inventory capacity C_p equals 20 and the threshold value T_p is equal to 5. Clearly, figure 5(a) and 5(b) show respectively that the buffer content of semi-finished products converges to capacity and the lead time decreases until a certain value as the arrival rate of semi-finished products increases. Concerning the mean number of semi-finished products, we can conclude that the order processing time distribution has no significant effect on this performance measure. Indeed, we observe that the difference is very small and that it decreases as the arrival rate of semi-finished products increases. However, the difference between a variance σ_s^2 equal to 1, 1/2 and 1/4 for the mean lead time remains constant and is significant, especially when the arrival rate λ_p is smaller than 0.7. Furthermore, in this numerical example, the mean number of semi-finished products decreases and the mean lead time increases as the variance increases. Indeed, the results of figure 6 show that the zero probability increases slightly as the variance of the order processing time distribution increases. As for Erlang distributed set-up times in

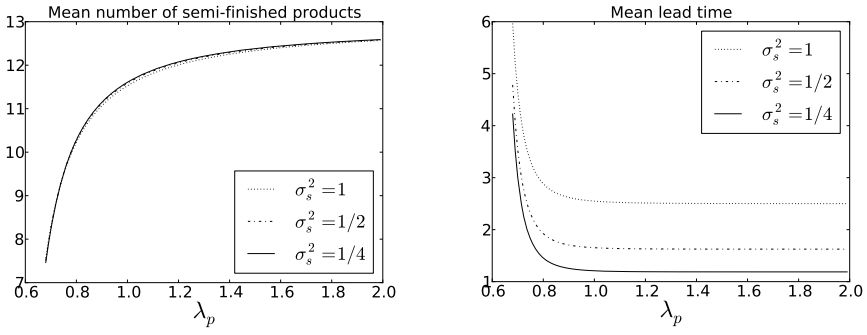


Fig. 5. The shape of the order processing time distribution is not significant for the mean number of semi-finished products and is significant for the mean lead time

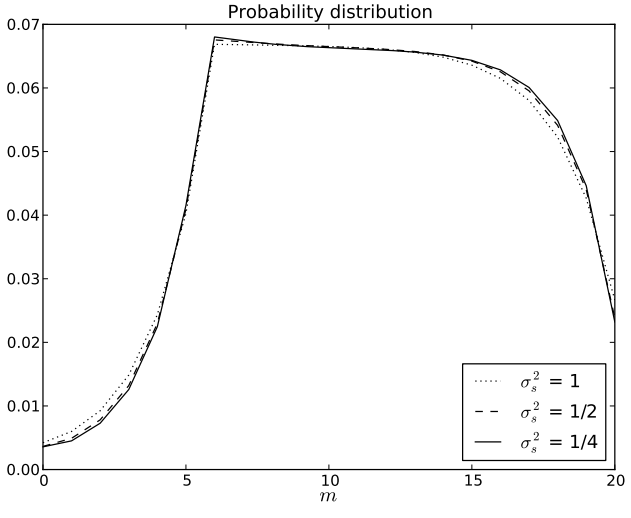


Fig. 6. The zero probability increases when the variance of the order processing time distribution decreases

section 4.2, we have a coupling effect between both performance measures – the mean number of semi-finished products increases such that the mean number of backlogged orders (and thus the mean lead time) decreases.

5 Conclusion

In this paper, we evaluate the performance of different hybrid push-pull systems with a decoupling inventory at the semi-finished products and reordering thresholds. In particular, we investigate the impact of different reordering policies, irregular order arrivals as well as the set-up time distribution and the order

processing time distribution on the performance of hybrid push-pull systems. In the studied hybrid push-pull systems, production of semi-finished products starts when the inventory goes below the so-called threshold value and stops when the inventory attains stock capacity. Decoupling means that the completion of a semi-finished product is only possible when there is an order. These orders are backlogged and can be satisfied only when the semi-finished products are available. Therefore, the studied push-pull system is modelled as a homogeneous quasi-birth-and-death process (QBD) and solved with matrix-analytic methods.

As our numerical examples show, there is trade-off to be made between the inventory cost and the service level, as expected – e.g. a higher threshold value causes on average a higher inventory cost and a smaller lead time. Furthermore, irregular order arrivals have a negative impact on the performance of the hybrid push-pull system. However, system performance is relatively insensitive to variation in the set-up time distribution and partially insensitive to variation in the order processing time distribution. Future work will focus on determining the total cost of the studied hybrid push-pull systems.

References

1. Bell, P.: A decoupling inventory problem with storage capacity constraints. Operations Research 28, 476–488 (1980)
2. Blecker, T., Abdelkafi, N.: Complexity and variety in mass customization systems: analysis and recommendations. Management Decision 44(7), 908–929 (2006)
3. Bonomi, F.: An approximate analysis for a class of assembly-like queues. Queueing Systems Theory and Applications, 289–309 (1987)
4. Chang, K., Lu, Y.: Queueing analysis on a single-station make-to-stock/make-to-order inventory-production system. Applied Mathematical Modelling 34, 978–991 (2010)
5. Cochran, J., Kim, S.: Optimizing a serially combined push and pull manufacturing system by simulated annealing. In: Second International Conference on Engineering Design and Automation (1998)
6. De Cuypere, E., Fiems, D.: Performance Evaluation of a Kitting Process. In: Al-Begain, K., Balsamo, S., Fiems, D., Marin, A. (eds.) ASMTA 2011. LNCS, vol. 6751, pp. 175–188. Springer, Heidelberg (2011)
7. Ghrayeb, O., Phojanamongkolkij, N., Tan, B.: A hybrid push/pull system in assemble-to-order manufacturing environment. Journal of Intelligent Manufacturing 20, 379–387 (2009)
8. Harris, F.: How many parts to make at once. Factory, the Magazine of Management 10(2), 135–136 (1913)
9. Harrison, J.: Assembly-like queues. Journal of Applied Probability 10(2), 354–367 (1973)
10. Hoekstra, S., Romme, J., Argelo, S.: Integral logistic structures: developing customer-oriented goods flow. McGraw-Hill (1992)
11. Hopp, W.J., Simon, J.T.: Bounds and heuristics for assembly-like queues. Queueing Systems 4, 137–156 (1989)
12. Hopp, W., Spearman, M.: Factory physics: Foundations of manufacturing management. The McGraw-Hill Companies, Inc. (2000)

13. Kaminsky, P., Kaya, O.: Combined make-to-order/make-to-stock supply chains. IIE Transactions 41, 103–119 (2009)
14. Latouche, G.: Queues with paired customers. Journal of Applied Probability 18, 684–696 (1981)
15. Latouche, G., Ramaswami, V.: Introduction to Matrix Analytic Methods in Stochastic Modeling. SIAM (1999)
16. Lee, C.: A recent development of the integrated manufacturing system: A hybrid of mrp and jit. International Journal of Operations and Production Management 13(4), 3–17 (1993)
17. Lipper, E., Sengupta, B.: Assembly-like queues with finite capacity: bounds, asymptotics and aproximations. Queueing Systems: Theory and Applications 18, 684 (1986)
18. Ohta, H., Hirota, T., Rahim, A.: Optimal production-inventory policy for make-to-order versus make-to-stock based on the m/er/1 queuing model. International Journal of Advanced Manufacturing Technologies 33, 36–41 (2007)
19. Pandey, P., Khokhajaikiat, P.: Performance modeling of multistage production systems operating under hybrid push-pull control. International Journal Production Economics 43, 115–126 (1995)
20. Ramachandran, K., Whitman, L., Ramachandran, A.: Criteria for determining the push-pull boundary. In: Industrial Engineering Research Conference. Orlando, FL, USA (2002)
21. Som, P., Wilhelm, W.: Analysis of stochastic assembly with GI-distributed assembly time. INFORMS Journal on Computing 11, 104–116 (1999)
22. Som, P., Wilhelm, W., Disney, R.: Kitting process in a stochastic assembly system. Queueing Systems 17, 471–490 (1994)
23. Soman, C., van Donk, D., Gaalman, G.: Combined make-to-order and make-to-stock in a food production system. International Journal of Production Economics 90, 223–235 (2004)
24. Spearman, M., Zazamis, M.: Push and pull production systems: Issues and comparisons. Operations Research 3, 521–532 (1992)
25. Takahashi, K., Nakamura, N.: Push pull, or hybrid control in supply chain management. International Journal of Computer Integrated Manufacturing 17(2), 126–140 (2004)

Controlling Variability in Split-Merge Systems

Iryna Tsimashenka, William Knottenbelt, and Peter Harrison

Imperial College London, 180 Queen's Gate,
London SW7 2AZ, United Kingdom
{it09,wjk,pgh}@doc.ic.ac.uk

Abstract. We consider split-merge systems with heterogeneous subtask service times and limited output buffer space in which to hold completed but as yet unmerged subtasks. An important practical problem in such systems is to limit utilisation of the output buffer. This can be achieved by judiciously delaying the processing of subtasks in order to cluster subtask completion times. In this paper we present a methodology to find those deterministic subtask processing delays which minimise any given percentile of the difference in times of appearance of the first and the last subtasks in the output buffer. Technically this is achieved in three main steps: firstly, we define an expression for the distribution of the range of samples drawn from n independent heterogeneous service time distributions. This is a generalisation of the well-known order statistic result for the distribution of the range of n samples taken from the same distribution. Secondly, we extend our model to incorporate deterministic delays applied to the processing of subtasks. Finally, we present an optimisation scheme to find that vector of delays which minimises a given percentile of the range of arrival times of subtasks in the output buffer. A case study illustrates the applicability of our proposed approach.

1 Introduction

Numerous physical systems of practical interest feature a queue of incoming tasks which split into synchronised subtasks that are processed in parallel at a set of (potentially heterogeneous) servers. Subtasks that complete service are held in an output buffer until all of its siblings have completed service. Examples of such systems include the processing of logical I/O requests by a RAID enclosure housing several physical disk drives [10], parallel job processing in MapReduce environments comprising several compute nodes [19], and the assembly of customer orders made up of multiple items in the highly-automated warehouses of large online retailers [15].

The importance of performance prediction in such systems has been long appreciated by performance modellers who have devised appropriate abstractions for their representation, most notably *split-merge* queueing systems and their less synchronised – but analytically much less tractable – counterparts, *fork-join* queueing systems [2]. Understandably, for both kinds of model, the primary focus of research work to date has been on the computation of moments of response time, most especially the mean. For example, Harrison and Zertal present an

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 165–177, 2012.

approximation for moments of the maximum of service times in a split-merge queueing system with general heterogeneous service times [8]; this gives an exact result in the case of exponential queues. For fork-join systems with homogeneous Markovian service time distributions, Nelson and Tantawi describe a technique which yields approximate upper and lower bounds on the mean response time as a function of the number of servers [13]. For the same system, Varki et al. [17] present approximate bounds on mean response time. Varma and Makowski [18] use interpolation between light and heavy traffic modes to approximate the mean response time for a homogeneous system of fork-join M/M/1 queues. The same fork-join system was considered in [9], where the maximum order statistic provides an easily-computable upper bound on response time.

By contrast, the focus of the present paper is not response time computation; rather it concerns ways to control the variability of subtask completion time (that is the difference in time between the arrival of the first and last subtasks of a task in the output buffer) in split-merge systems. The idea is to try to cluster the arrival of subtasks in the output buffer by applying judiciously chosen deterministic delays to subtasks before they are dispatched to the parallel servers. This has especial relevance for systems that involve the retrieval of orders comprising multiple items from automated warehouses [15], since partially completed subtasks must be held in a physical buffer space that is often limited and highly utilised; consequently it is difficult to manage. Despite this, to the best of our knowledge, this problem has not received significant attention in the literature. Our previous work [16] presented a simple mean-based methodology for computing the vector of deterministic subtask delays that minimises a cost function given by the difference between the expected maximum and expected minimum subtask completion times (across all subtasks arising from a particular task). However, an expected value does not always satisfy service level objectives; in addition there is a dependence between the maximum and minimum subtask completion times which must be taken into account for any distributional analysis. The methodology we present here yields the set of subtask delays which minimises any given percentile of the distribution of the difference in the time of appearance of the first and last subtasks in the output buffer.

The technical contribution of this paper begins with a generalisation of the well-known order statistics result for the distribution of the range when n samples are taken from a given distribution $F(t)$. In particular, we present an exact analytical expression for the distribution of the range of n samples taken from heterogeneous distributions $F_i(t)$ $(i = 1, \ldots, n)$. Having extended this theory to incorporate deterministic subtask processing delays, we show how an optimisation procedure can be applied to a split-merge system to find that vector of subtask delays which minimises a given percentile of the range of subtask completion times.

The rest of the paper is organised as follows. Section 2 describes essential preliminaries including a definition of split-merge systems and selected results from the theory of order statistics. Section 3 presents various heterogeneous order statistic results, including the distribution of the range. Section 4 shows how the

basic split-merge model can be enhanced to support deterministic delays, defines an appropriate objective function, and presents a related optimisation procedure. Section 5 presents a case study which demonstrates the applicability of our work. Section 6 concludes and considers appropriate directions for future work.

2 Essential Preliminaries

2.1 Parallel Systems

A split-merge system (see Fig. 1) is a composition of a queue of waiting tasks (assumed to arrive according to a Poisson process with mean rate λ), a split point, several heterogeneous servers (which serve their allocated subtask with general service time distribution with mean service rate $1/\mu_i$), buffers for completed subtasks (merge buffers) and a merge point [2]. We note that in practice in physical systems it is not uncommon for the merge buffers to share the same physical space which is managed as a single logical output buffer. When the queue of

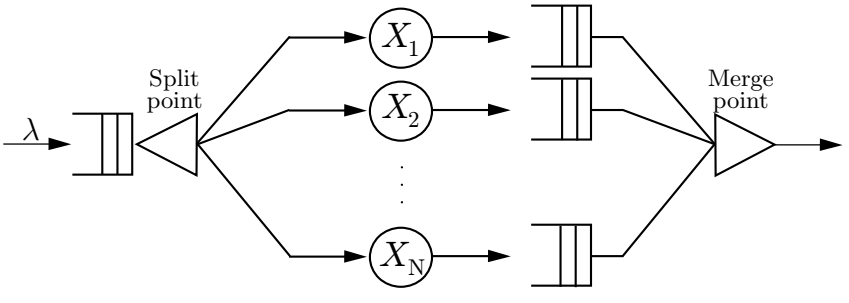


Fig. 1. Split–Merge queueing model

waiting tasks is not empty and the parallel servers are idle, a task is injected into the system from the head of the queue. The task is split into n subtasks at the *split* point and the subtasks arrive simultaneously at the n parallel servers to receive service. Completed subtasks join a merge buffer. Only after all subtasks (belonging to a particular task) are present in the merge buffers does the original task depart the system via the merge point. We note that this split-merge system is a more synchronised type of fork-join system. In split-merge systems parallel servers are blocked after they have served a subtask while the original task is in the system, whereas in fork-join systems there is no queue of waiting tasks, but there is a queue of subtasks at each parallel server. Split-merge systems can also be said to be a more conservative type of fork-join system in the sense that analysis of task response time in a split-merge system yields an upper bound on task response time in a fork-join system [9].

2.2 Theory of Order Statistics [6]

Definition: Let the increasing sequence $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$ be a permutation of the real valued random variables $X_1, X_2, \ldots, X_n$, i.e. the X_i arranged in ascending order $X_{(1)} \leqslant X_{(2)} \leqslant \ldots \leqslant X_{(n)}$. Then $X_{(i)}$ is called the *ith order statistic*, for $i = 1, 2, \ldots, n$. The first and last order statistics, $X_{(1)}$ and $X_{(n)}$, are the minimum and maximum respectively, which are also called the *extremes*. $T = X_{(n)} - X_{(1)}$ is the *range*.

We assume initially that the random variables X_i are identically distributed as well as independent (iid), but of course the $X_{(i)}$ are dependent because of the ordering.

Distribution of the *k*th-Order Statistic (iid case)

If $X_1, X_2, \ldots, X_n$ are n independent random variables, the cumulative distribution function (cdf) of the maximum order statistic (the maximum) is simply given by

$$F_n(x) = Pr\{X_{(n)} \leqslant x\} = Pr\{X_i \leqslant x, 1 \leqslant i \leqslant n\} = F^n(x)$$

Likewise, the cdf of the minimum statistic is:

$$F_1(x) = Pr\{X_{(1)} \leqslant x\} = 1 - Pr\{X_{(1)} > x\} = 1 - Pr\{X_i > x, 1 \leqslant i \leqslant n\} = 1 - [1 - F(x)]^n$$

These are special cases of the general cdf of the rth order statistic, $F_r(x)$, which can be expressed as:

$$\begin{aligned} F_r(x) = Pr\{X_{(r)} \leqslant x\} &= Pr\{\text{at least } r \text{ of the } X_i \leqslant x\} \\ &= \sum_{i=r}^{n} \binom{n}{i} F(x)^i [1 - F(x)]^{n-i} \end{aligned} \tag{1}$$

The pdf of X_r, $f_r(x) = F_r'(x)$, where the prime denotes the derivative with respect to x, when it exists, is then:

$$f_r(x) = \frac{n!}{(r-1)!(n-r)!} F^{r-1}(x) f(x) [1 - F(x)]^{n-r}.$$

Multiplying both sides by "small" ϵ, this result follows intuitively from noting that we require one of the X_i to take a value in the interval $(x, x + \epsilon]$, exactly $r - 1$ of the X_i to be less than or equal to x and exactly $n - r$ of them to be greater than x. The coefficient $n!/((r-1)!1!(n-r)!)$ is the number of ways of doing this, given that the X_i are stochastically indistinguishable.

The joint density function of the rth and sth order statistics $X_{(r)}, X_{(s)}$, where $(1 \leqslant r < s \leqslant n)$, is:

$$f_{rs}(x, y) = S_{rs} F^{r-1}(x) f(x) [F(y) - F(x)]^{s-r-1} f(y) [1 - F(y)]^{n-s} \tag{2}$$

where $S_{rs} = \frac{n!}{(r-1)!(s-r-1)!(n-s)!}$, by similar reasoning. The corresponding joint cdf $F_{rs}(x, y)$ of $X_{(r)}$ and $X_{(s)}$ may be obtained by integration of the pdf or, alternatively, for $x < y$ we have:

$$F_{rs}(x,y) = Pr\{\text{at least } r \text{ of the } X_i \leqslant x, \text{ at most } n-s \text{ of the } X_i > y\}$$
$$= \sum_{j=s}^{n}\sum_{i=r}^{j} Pr\{\text{exactly } i \text{ of the } X_i \leqslant x, \text{ exactly } n-j \text{ of the } X_i > y\}$$
$$= \sum_{j=s}^{n}\sum_{i=r}^{j} \frac{n!}{i!(j-i)!(n-j)!} F^i(x)[F(y)-F(x)]^{j-i}[1-F(y)]^{n-j}$$

Finally, the joint pdf for the k order statistics $X_{(n_1)},\ldots,X_{(n_k)}$, $1 \leqslant n_1 < \ldots < n_k \leqslant n$), is, similarly, for $x_1 \leqslant \ldots \leqslant x_k$:

$$f_{n_1,\ldots,n_k}(x_1,\ldots,x_k) = S_{n_1,\ldots,n_k} F^{n_1-1}(x_1) f(x_1)[F(x_2)-F(x_1)]^{n_2-n_1-1} f(x_2)\cdots$$
$$[F(x_k)-F(x_{k-1})]^{n_k-n_{k-1}-1} f(x_k)[1-F(x_k)]^{n-n_k}$$

where $S_{n_1,\ldots,n_k} = \frac{n!}{(n_1-1)!(n_2-n_1-1)!\ldots(n_k-n_{k-1}-1)!(n-n_k)!}$.

Distribution of the Range

The pdf $f_{T_{rs}}(x)$ of the interval $T_{rs} = X_{(s)} - X_{(r)}$ follows from the joint pdf of the rth and sth order statistics in Eq. 2 by setting $y = x + t_{rs}$ and integrating over x, giving:

$$f_{T_{rs}}(t_{rs}) = S_{rs}\int_{-\infty}^{\infty} F^{r-1}(x) f(x)[F(x+t_{rs})-F(x)]^{s-r-1} f(x+t_{rs})[1-F(x+t_{rs})]^{n-s} dx$$

In the special case when $r = 1$ and $s = n$, T_{rs} is the range $T = X_{(n)} - X_{(1)}$ and the pdf simplifies to:

$$f(t) = n(n-1)\int_{-\infty}^{\infty} f(x)[F(x+t)-F(x)]^{n-2} f(x+t)dx$$

The cdf of T then follows by integrating inside the integral with respect to x, giving:

$$F(t) = n\int_{-\infty}^{\infty} f(x)\int_0^t (n-1) f(x+t')[F(x+t')-F(x)]^{n-2} dt' dx$$
$$= n\int_{-\infty}^{\infty} f(x)\left[[F(x+t')-F(x)]^{n-1}\right]_{t'=0}^{t'=t} dx$$
$$= n\int_{-\infty}^{\infty} f(x)[F(x+t)-F(x)]^{n-1} dx \qquad (3)$$

As noted in [6], this equation follows intuitively by noting that the integrand (multiplied by an infinitesimal quantity dx) is the probability that X_i falls into the interval $(x, x+dx]$ (for some i) and the remaining $n-1$ of the $X_j, j \neq i$ fall into $(x, x+t]$. There are n ways of choosing i, giving the factor n.

3 Heterogeneous Order Statistics

We now consider n independent, real-valued random variables $X_1,\ldots,X_n$ where each X_i has an arbitrary probability distribution $F_i(x)$ and probability density

function $f_i(x) = F_i'(x)$. In this case of "heterogeneous" (or independent, but not necessarily identically distributed) random variables, we call the order statistics *heterogeneous order statistics* to distinguish them from the better known results where the random variables are implicitly assumed to be identically distributed.

Recent decades have seen increasing consideration given to the heterogeneous case in the literature. Key theoretical results for the distribution and density functions of heterogeneous order statistics are summarised in [7]. This includes the work of Sen [14], who derived bounds on the median and the tails of the distribution of heterogeneous order statistics. Practical issues related to the numerical computation of the ith heterogeneous order statistic are considered in [5], with special consideration of recurrence relations among distribution functions of order statistics.

Distribution of the rth Heterogeneous Order Statistic

The rth heterogeneous order statistic, derived similarly to Eq. 1, has the following cdf:

$$\begin{aligned} F_{(r)}(x) &= Pr\{X_{(r)} \leqslant x\} = Pr\{\text{at least } r \text{ of the } X_i \leqslant x\} \\ &= \sum_{i=r}^{n} \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2\} \in \mathcal{P}_i} \prod_{k=1}^{i} F_{\ell_{1k}}(x) \prod_{k=1}^{n-i} [1 - F_{\ell_{2k}}(x)] \end{aligned} \tag{4}$$

where $\mathcal{P}_i$ is the set of all two-set partitions $\{D, E\}$ of $\{1, 2, \ldots, n\}$ with $|D| = i$ and $|E| = n - i$, and ℓ_{hk} is the kth component of the vector $\boldsymbol{\ell_h}$ for $h = 1, 2$.

Similarly to the homogeneous case, the minimum and maximum order statistics are respectively given by:

$$\begin{aligned} F_{(1)}(x) = Pr\{X_{(1)} \leqslant x\} = 1 - Pr\{X_{(1)} > x\} = \\ 1 - Pr\{X_i > x \mid 1 \leq i \leq n\} = 1 - \prod_{i=1}^{n} [1 - F_i(x)], \end{aligned}$$

and

$$F_{(n)}(x) = Pr\{X_{(n)} \leqslant x\} = Pr\{X_i \leqslant x \mid 1 \leq i \leq n\} = \prod_{i=1}^{n} F_i(x).$$

Differentiating Eq. 4 and simplifying yields the pdf:

$$f_{(r)}(x) = \sum_{i=r}^{n} \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2\} \in \mathcal{P}_i} \left[\sum_{j=1}^{i} \prod_{k=1, k \neq j}^{i} F_{\ell_{1k}}(x) \prod_{k=1}^{n-i} [1 - F_{\ell_{2k}}(x)] f_{\ell_{1j}}(x) - \right.$$
$$\left. \sum_{j=1}^{n-i} \prod_{k=1}^{i} F_{\ell_{1k}}(x) \prod_{k=1, k \neq j}^{n-i} [1 - F_{\ell_{2k}}(x)] f_{\ell_{2j}}(x) \right]$$

$$= \sum_{i=r}^{n} \sum_{h=1}^{n} \Bigg[\sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2\} \in \mathcal{P}_{i-1}^{h-}} \prod_{k=1}^{i-1} F_{\ell_{1k}}(x) \prod_{k=1}^{n-i} [1 - F_{\ell_{2k}}(x)] f_h(x) -$$

$$I_{i<n} \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2\} \in \mathcal{P}_{i}^{h-}} \prod_{k=1}^{i} F_{\ell_{1k}}(x) \prod_{k=1}^{n-i-1} [1 - F_{\ell_{2k}}(x)] f_h(x) \Bigg]$$

$$= \sum_{h=1}^{n} f_h(x) \Bigg[\sum_{i=r}^{n} \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2\} \in \mathcal{P}_{i-1}^{h-}} \prod_{k=1}^{i-1} F_{\ell_{1k}}(x) \prod_{k=1}^{n-i} [1 - F_{\ell_{2k}}(x)] -$$

$$\sum_{i=r+1}^{n} \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2\} \in \mathcal{P}_{i-1}^{h-}} \prod_{k=1}^{i-1} F_{\ell_{1k}}(x) \prod_{k=1}^{n-i} [1 - F_{\ell_{2k}}(x)] \Bigg]$$

$$= \sum_{h=1}^{n} f_h(x) \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2\} \in \mathcal{P}_{r-1}^{h-}} \prod_{k=1}^{r-1} F_{\ell_{1k}}(x) \prod_{k=1}^{n-r} [1 - F_{\ell_{2k}}(x)]$$

where $I_\bullet$ is the indicator function and $\mathcal{P}_i^{h-}$ is the set of all 2-set partitions of $\{1, 2, \ldots, n\} \setminus \{h\}$ with i elements in the first set and $1 \leq h \leq n$. In fact this result also follows from an intuitive argument using the infinitesimal interval $(x, x + \epsilon]$, as in the homogeneous case.

The joint density function $f_{rs}(x, y)$ of two order statistics, $X_{(r)}$ and $X_{(s)}$, for $1 \leqslant r < s \leqslant n$, follows similarly as:

$$f_{(r)(s)}(x, y) = \sum_{1 \leq h_1 \neq h_2 \leq n} f_{h_1}(x) f_{h_2}(y) \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3\} \in \mathcal{P}_{r-1, s-r-1}^{h_1-, h_2-}} \prod_{k=1}^{r-1} F_{\ell_{1k}}(x) \times \prod_{k=1}^{s-r-1} [F_{\ell_{2k}}(y) - F_{\ell_{2k}}(x)] \prod_{k=1}^{n-s} [1 - F_{\ell_{3k}}(y)] \tag{5}$$

where $\mathcal{P}_{i_1, i_2}^{h_1-, h_2-}$ is the set of all 3-set partitions of $\{1, 2, \ldots, n\} \setminus \{h_1, h_2\}$ with i_1 elements in the first set, i_2 elements in the second set, and so $n - i_1 - i_2 - 2$ in the third, and $1 \leq h_1 \neq h_2 \leq n$.

Distribution of the Range for Heterogeneous Order Statistics

From the joint pdf of two heterogeneous order statistics in Eq. 5, we obtain the pdf of the interval $T_{rs} = X_{(r)} - X_{(s)}$ by setting $t_{rs} = y - x$:

$$f_{(r:s)}(t_{rs}) = \sum_{1 \leq h_1 \neq h_2 \leq n} \int_{-\infty}^{\infty} f_{h_1}(x) f_{h_2}(x + t_{rs}) \sum_{\{\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3\} \in \mathcal{P}_{r-1, s-r-1}^{h_1-, h_2-}} \prod_{k=1}^{r-1} F_{\ell_{1k}}(x) \prod_{k=1}^{s-r-1} [F_{\ell_{2k}}(x + t_{rs}) - F_{\ell_{2k}}(x)] \prod_{k=1}^{n-s} [1 - F_{\ell_{3k}}(x + t_{rs})] dx \tag{6}$$

For the range, we want the special case in which $r = 1,\ s = n$ and $T = X_{(n)} - X_{(1)}$, giving the pdf:

$$f_{(1:n)}(t) = \sum_{1\le h_1\ne h_2\le n} \int_{-\infty}^{\infty} f_{h_1}(x) f_{h_2}(x+t) \sum_{\{\boldsymbol{\ell}_1,\boldsymbol{\ell}_2,\boldsymbol{\ell}_3\}\in \mathcal{P}_{0,n-2}^{h_1-,h_2-}} \prod_{k=1}^{n-2} [F_{\ell_{2k}}(x+t) - F_{\ell_{2k}}(x)]dx$$

$$= \sum_{1\le h_1\ne h_2\le n} \int_{-\infty}^{\infty} f_{h_1}(x) f_{h_2}(x+t) \prod_{k\ne h_1,h_2} [F_k(x+t) - F_k(x)]dx \tag{7}$$

The cdf now follows by integration (inside the sum and integral with respect to x):

$$F_{(1:n)}(t) = \sum_{1\le h_1\ne h_2\le n} \int_{-\infty}^{\infty} f_{h_1}(x) \int_0^t f_{h_2}(x+t') \prod_{k\ne h_1,h_2} [F_k(x+t') - F_k(x)]dx\ dt'$$

$$= \sum_{1\le h_1\le n} \int_{-\infty}^{\infty} f_{h_1}(x) \prod_{k\ne h_1} [F_k(x+t) - F_k(x)]dx \tag{8}$$

In fact, the same result can be obtained by noting that Eq. 3 generalises using the argument given immediately following it. This is that, given a particular choice of $i = 1, 2, \ldots, n$, the integrand (multiplied by an infinitesimal quantity $\mathrm{d}x$) is the probability that X_i falls into the interval $(x, x + dx]$ and the other $X_j, j \ne i$ fall into $(x, x+t]$. Of course there are n ways of choosing i, and so we have to sum over n terms; in the homogeneous case, all these terms are the same, which gave the factor n. For heterogeneous order statistics, we therefore obtain:

$$F_{range}(t) = F_{(1:n)}(t) = \sum_{i=1}^{n} \int_{-\infty}^{\infty} f_i(x) \prod_{j=1,j\ne i}^{n} [F_j(x+t) - F_j(x)]dx \tag{9}$$

This is a useful result, which requires a sum of only n terms. It will form the basis for range-optimisation in split-merge systems with heterogeneous service time distributions as considered in the next section.

4 Controlling Variability in Split-Merge systems

Introducing Delays

Our aim is to control the variability of subtask completion (equivalently merge buffer arrival) times by introducing a vector of delays:

$$\mathbf{d} = (d_1, d_2, \ldots, d_i, \ldots, d_{n-1}, d_n) \tag{10}$$

Here, element d_i of the vector $\mathbf{d}$ denotes the deterministic delay that will be applied before a subtask is sent to server i for processing. We note that in order to avoid unnecessarily delaying all subtasks we require that the subtask delay for at least one server (the "bottleneck" server) be set to 0.

After applying the delays from Eq. 10, the distribution of the range from Eq. 9 becomes:

$$F_{range}(t, \mathbf{d}) = \sum_{i=1}^{n} \int_{-\infty}^{\infty} f_i(x - d_i) \prod_{j=1,j\ne i}^{n} [F_j(x + t - d_j) - F_j(x - d_j)]dx \tag{11}$$

We assume that, $\forall i,\ f_i(t - d_i) = 0,\ \forall t < d_i$. Similarly, $\forall j,\ F_j(t - d_j) = 0,\ \forall t < d_j$.

Optimisation Procedure

In this section we move away from our previous mean-based technique [16] towards a more sophisticated framework for finding delay vectors which provide soft (probabilistic) guarantees on variability. More specifically, for a given probability α, we aim to minimise the 100αth percentile of variability with respect to $\mathbf{d}$. That is, we aim to solve for $\mathbf{d}$ in:

$$\min_{\mathbf{d}} F^{-1}_{range}(\alpha, \mathbf{d}) \tag{12}$$

Put another way, we aim to find that vector $\mathbf{d}$ which yields the lowest value for the 100αth percentile of the difference in the completion times of the first and the last subtasks (belonging to each task).

Practically, we developed a numerical optimisation procedure by prototyping it in Mathematica and subsequently implementing a full version of it in C++ for efficiency reasons. Evaluation of Eq. 11 for a given α and $\mathbf{d}$ is performed by means of straightforward numerical integration using the trapezoidal rule. While this is adequate and accurate for almost all continuous service time density and distribution functions, complications arise in the case of the pdf of deterministic service time density functions because of their infinitely thin, infinitely high impulse. We choose to resolve this by replacing the deterministic pdf with delay parameter a by the Gaussian approximation:

$$f_{Det(a)}(x) \approx \frac{1}{c\sqrt{\pi}} e^{-\frac{(x-a)^2}{c^2}}$$

which becomes exact as $c \to 0$; in practice we set $c = 0.01$.

In order to invert Eq. 11 for a given α and $\mathbf{d}$, we make use of the well-known Bisection method [4] which in turn exploits Bolzano's Intermediate Value Theorem. Although it is more computationally expensive than the Newton-Rhapson method, we choose the Bisection method because its gradient-free nature makes it considerably more robust. In circumstances where computational efficiency is a critical concern, we note that it is possible to apply more efficient gradient-free algorithms such as Brent's method [3].

Finally, we explore the optimisation surface of $F^{-1}_{range}(\alpha, \mathbf{d})$ with the initial $\mathbf{d} = \{0, \ldots, 0\}$ using a numerical optimization procedure. We constrain the search such that $d_i \geq 0$ for all i and $\prod_i d_i = 0$ (that is, the "bottleneck" server(s) should have no unnecessary additional delay). In our implementation, we have used a simple Nelder-Mead optimisation technique [12], although we note that a range of more sophisticated (and correspondingly considerably more complex to implement) gradient-free optimisation techniques are also available, e.g. [1, 11].

5 Case Study

Consider a split-merge system with 3 parallel servers having heterogeneous service time density functions:

$f_1(t) = \text{Pareto}(\alpha = 3, b = 3.5)$ $(E[X_1] = 5.25,\ \text{Med}[X_1] = 4.40972,\ \text{Var}[X_1] = 9.1875)$
$f_2(t) = \text{Erlang}(n = 2, \lambda = 1)$ $(E[X_2] = 2,\ \text{Med}[X_2] = 1.67835,\ \text{Var}[X_2] = 2)$
$f_3(t) = \text{Det}(5)$ $(E[X_3] = 5,\ \text{Med}[X_3] = 5,\ \text{Var}[X_3] = 0)$

Without adding any extra delays, it is straightforward to apply Eq. 9 in a simple root finding algorithm (e.g. the Bisection method) to compute the 50th ($\alpha = 0.5$) and 90th ($\alpha = 0.9$) percentile of the range of subtask arrival times as $t = 3.629$ time units and $t = 5.52998$ time units respectively.

Incorporating delays into the distribution of the range of subtask merge buffer arrival times as per Eq. 11, and executing a Nelder-Mead optimisation (suitably constrained so that $\prod_i d_i = 0$) to solve Eq. 12 given $\alpha = 0.5$ for $\mathbf{d}$ yields

$$\mathbf{d} = (0.79335, 3.47083, 0)$$

as shown in Figure 2. We note that in this case the "bottleneck" server is server 3, despite the fact that the server 1 has a higher mean service time than server 3. With the incorporation of the optimal delays, the 50th percentile of the range of subtask arrival times becomes $t = 1.32592$, representing an improvement of 63.4% over the original system configuration without delays.

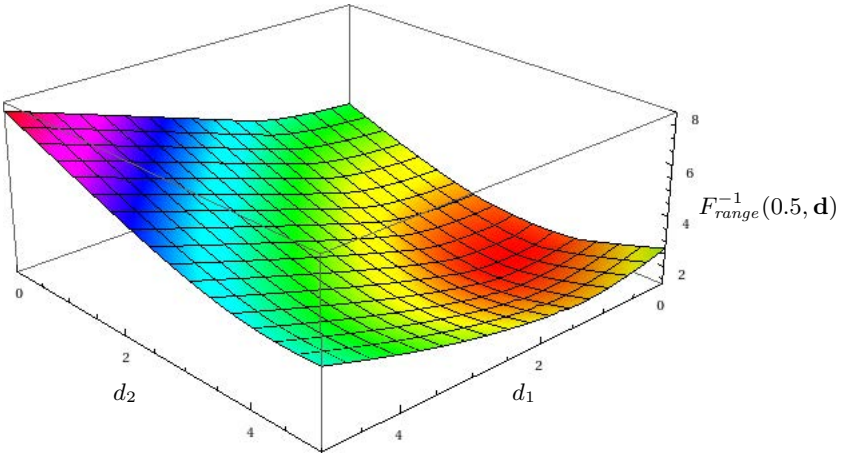


Fig. 2. 50th percentile of the range of subtask merge buffer arrival times for various deterministic processing delays. The optimal delay vector is $\mathbf{d} = (0.79335, 3.47083, 0)$.

For $\alpha = 0.9$ we obtain

$$\mathbf{d} = (0, 2.68176, 1.45705)$$

as shown in Figure 3. We note that for this percentile the "bottleneck" switched from server 3 to server 1. With the incorporation of the optimal delays, the 90th percentile of the range of subtask arrival times becomes $t = 3.77626$, representing an improvement of 31.7% over the original system configuration without delays.

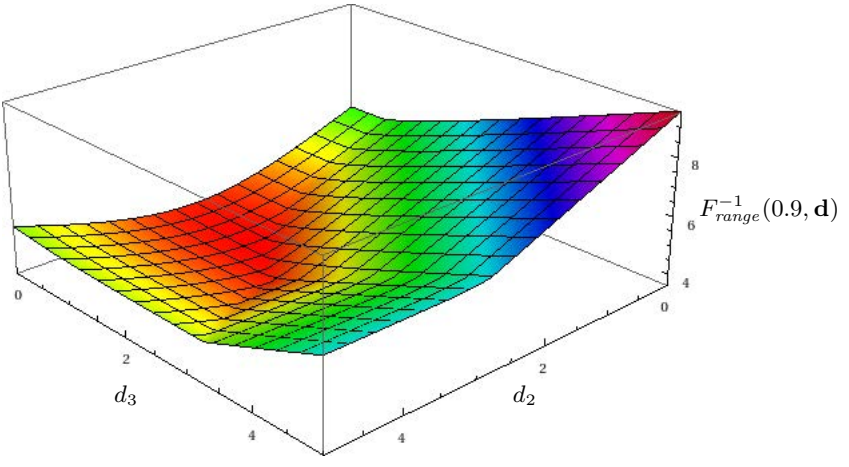


Fig. 3. 90th percentile of the range of subtask merge buffer arrival times for various deterministic processing delays. The optimal delay vector is $\mathbf{d} = (0, 2.68176, 1.45705)$.

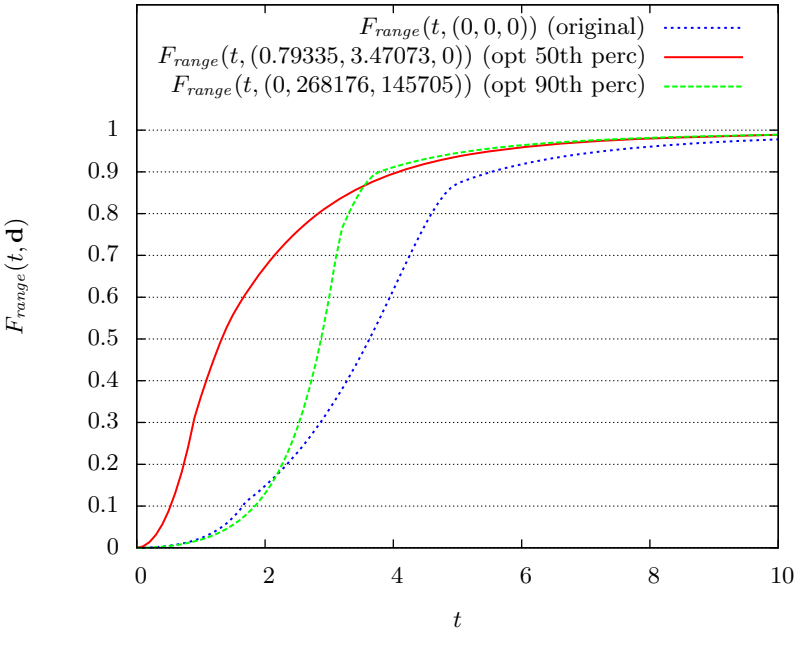


Fig. 4. Distributions of the range of subtask merge buffer arrival times given subtask delays optimised for various percentiles

Figure 4 shows how the distribution of the range of subtask merge buffer arrival times changes according to the optimised percentile. We note that a change of the optimised percentile can have a significant impact on the quantiles of $F_{range}(t, \mathbf{d})$, according to how the "bottleneck" server shifts.

Although it is not our focus, it is interesting to consider the effect of the subtask delays on the expected task completion time. For a system without delays, the expected task completion time is $E[X_{(n)}] = 5.75712$ time units. After introducing subtask delays in order to minimise the 50th and 90th percentile of the range of subtask processing times, the expected task completion time becomes 6.57628 time units (14% increase) and 7.00894 time units (26%) increase respectively.

6 Conclusions and Future Work

This paper has presented a methodology for controlling variability in split-merge systems. Here variability is defined in terms of a given percentile of the range of arrival times of subtasks in the merge buffers, and is controlled through the application of judiciously chosen deterministic delays to subtask service times. The methodology has three main building blocks. The first is an exact analytical expression for the distribution of the range of subtask merge buffer arrival times over n heterogeneous servers in a split-merge system. This is a natural generalisation of the well-known order statistics result for the distribution of the range taken over n homogeneous servers. The second is the introduction of deterministic subtask delays into the aforementioned expression. The third is a optimisation procedure which yields the vector of subtask delays which minimises a given percentile of the range of subtask merge buffer arrival times. We presented a case study which showed that the choice of percentile can have a significant impact on the optimal delay vector and the "bottleneck" server.

As previously mentioned fork-join systems are significantly less analytically tractable than split-merge systems. However, they are more realistic abstractions of many real world systems on account of their less-constrained task synchronisation. Consequently a natural future direction of this work is to try and generalise our results to fork-join systems. In line with previous research we believe we are unlikely to find an exact analytical expression for the distribution of the range of join buffer arrival times. However, a numerical approach and/or an analytical approximation may be possible.

Finally, the scalability of our methodology to very large split-merge systems with 100+ service nodes is currently an open question. However, large-scale problems are sometimes encountered when modelling real-life systems. Consequently we will conduct experiments to assess the scaling behavior of our methodology. It may be beneficial to devise an approach that makes use of parallel computations using MPI (Message Passing Interface).

References

[1] Ali, M.M., Gabere, M.N.: A simulated annealing driven multi-start algorithm for bound constrained global optimization. Journal of Computational and Applied Mathematics 223(10), 2661–2674 (2010)

[2] Bolch, G., et al.: Queueing Networks and Markov Chains. John Wiley (2006)

[3] Brent, R.P.: Algorithms for Minimization Without Derivatives. Dover Books on Mathematics, ch. 4. Dover Publications (2002)

[4] Burden, E.F., Burden, R.L.: Numerical Methods, 3rd edn. Cram101 Textbook Outlines. Academic Internet Publishers (2006)

[5] Cao, G., West, M.: Computing distributions of order statistics. Communications in Statistics – Theory and Methods 26(3), 755–764 (1997)

[6] David, H.A.: Order Statistics. Wiley Series in Probability and Mathematical Statistics. John Wiley (1980)

[7] David, H.A., Nagaraja, H.N.: The non-IID case. In: Order Statistics, ch. 5, 3rd edn., pp. 95–120. John Wiley & Sons, Inc. (2005)

[8] Harrison, P.G., Zertal, S.: Queueing models of RAID systems with maxima of waiting times. Perf. Evaluation 64(7-8), 664–689 (2007)

[9] Lebrecht, A., Knottenbelt, W.J.: Response Time Approximations in Fork-Join Queues. In: 23rd Annual UK Performance Engineering Workshop (UKPEW) (July 2007)

[10] Lebrecht, A.S., Dingle, N.J., Knottenbelt, W.J.: Analytical and Simulation Modelling of Zoned RAID Systems. The Computer Journal 54(5), 691–707 (2011)

[11] Lewis, R.M., Shepherd, A., Torczon, V.: Implementing generating set search methods for linearly constrained minimization. SIAM Journal on Scientfic Computing 29(6), 2507–2530 (2007)

[12] Nelder, J.A., Mead, R.: A simplex method for function minimization. The Computer Journal 7(4), 308–313 (1965)

[13] Nelson, R., Tantawi, A.N.: Approximate analysis of fork/join synchronization in parallel queues. IEEE Trans. on Computers 37(6), 739–743 (1988)

[14] Sen, P.K.: A note on order statistics for heterogeneous distributions. The Annals of Mathematical Statistics 41(6), 2137–2139 (1970)

[15] Serfozo, R.: Basics of Applied Stochastic Processes. Springer (2009)

[16] Tsimashenka, I., Knottenbelt, W.J.: Reduction of Variability in Split–Merge Systems. In: Imperial College Computing Student Workshop (ICCSW 2011), pp. 101–107 (2011)

[17] Varki, E., Merchant, A., Chen, H.: The M/M/1 fork-join queue with variable sub-tasks

[18] Varma, S., Makowski, A.M.: Interpolation approximations for symmetric fork-join queues. Performance Evaluation 20(1-3), 245–265 (1994)

[19] Zaharia, M., et al.: Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In: Proc. 5th European Conference on Computer Systems (EuroSys 2010), pp. 265–278 (2010)

Some Improvements for the Computation of the Steady-State Distribution of a Markov Chain by Monotone Sequences of Vectors

Jean-Michel Fourneau and Franck Quessette

PRiSM, Université de Versailles-Saint-Quentin, CNRS UMR 8144, UniverSud, Versailles, France

Abstract. We present several new improvements for a recently published algorithm [5] for computing the steady-state distribution of a finite ergodic Markov chain, which has a proved monotone convergence under some structural constraints on the matrix. We show how to accommodate infinite state space and that the structural constraints of the algorithm are consistent with Pagerank matrix. We present how to combine this algorithm with stochastic comparison theory to numerically obtain bounds and we prove a pre-processing of the matrix which allows to alleviate the structural constraints. The approaches are illustrated through several small examples.

1 Introduction

One of us has recently presented in [5] a new numerical algorithm to compute the steady state distribution of a discrete time Markov chain (DTMC in the following). Continuous-time Markov chains can be accommodated after an optimal uniformization to make the matrix row diagonal dominant [6]. This algorithm relies on a completely new iterative scheme based on operators "max", "min" and "+". At each step of the iteration, the algorithms provide upper and lower bounds for each component of the steady-state distribution of a DTMC. Note that we assumed in [5] that the chain is finite and that it has a stationary distribution. Here, we consider ergodic chains and we explain how we can handle infinite ergodic chains.

It is important to notice that the upper and lower bounding vectors considered here are not probability distribution vectors (as it is the case with the bounds obtained using stochastic orders), this will be clearer after some examples in Section 2 and 3. But all the elements of the vectors are non negative and are smaller than 1.

These iterative algorithms all provide component-wise upper or lower bounds at each iteration and some of them converge to the true solution under some technical conditions which are easy to check. More precisely we compute a quantity denoted as Nabla (i.e. ∇, which is more precisely defined in the next section). A sufficient condition for convergence is $||\nabla|| > 0$. In this paper we denote by $||x||$ the sum of the elements of vector x. The speed of convergence is also related to $||\nabla||$. The larger the norm $||\nabla||$, the faster the convergence.

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 178–192, 2012.

When $||\nabla|| > 0$, these algorithms share very nice properties: first as it is proved that the solution is within an interval, which decreases at each iteration and leads to zero, we can state the convergence of numerical computation. Second, we have a clear tradeoff between computation efforts and the tightness of the numerical results. Third, under some technical conditions, when we apply the lower bounding algorithm to an element-wise lower bound of the matrix, we obtain a proved lower bound of the steady-state solution. Thus we are able to obtain guarantee on the steady-state distribution or on the expectation of a reward even if we do not completely compute all the entries of the matrix.

To the best of our knowledge the iterative equation used in [5] gives a new interpretation of the steady-state distribution of DTMCs. Monotone sequences converging to the the solution of non-negative linear systems have been previously studied by Berman and Plemmons [2] and extended to prove some algorithms and to propose some heuristics for Markov chains by Semal in [11] and for singular matrices by Song [13]. These approaches are not related to our method (even if they obtain bounds for steady-state distributions) as they consider some classical splittings of the linear system describing the equilibrium equation, while we propose an iterative scheme which is original. The previous approaches have some drawbacks, which are inherent to the computation of monotone sequences of probabilities, and which implies a normalization of the vector. Therefore these approaches are extremely hard to use in practice. Unlike Semal's approach, our algorithms provide bounds at each iteration without normalisation when $||\nabla|| > 0$, and are simple to use.

But when $||\nabla|| = 0$, both theoretical and practical results are much less interesting. We do not have a proof of convergence to the steady-state distribution, but we still have component-wise bounds. We can also obtain accurate approximations but we do not know if they provide bounds for some steady-state probabilities. Thus we investigate how we can improve the ideas proposed in [5] to extend the applicability of the method for general stochastic matrices.

The paper is organised as follows. First in Section 2, we present a brief introduction to our previous algorithms published in [5]. We also show that the condition $||\nabla|| > 0$ is true for positive matrices and this opens new techniques to solve the ranking problem [10]. We also show how we can deal with denumerable state space under some technical conditions. Then, in Section 3 we combine stochastic bounding techniques and our algorithms to compute efficiently bounds. Section 4 is devoted to a preprocessing technique, which fills the matrix to satisfy the constraint $||\nabla|| > 0$. The paper is illustrated with several small examples to illustrate the approach. Much larger matrices can be solved as well, as a software tool is already available [8,4]. We will try to combine, in the future, these algorithms with the polynomial approach developed in [7], because some polynoms of a stochastic matrix increase the filling, while keeping the steady-state distribution unchanged.

2 A Brief Introduction to $I\nabla U$ and $I\nabla L$ Algorithms

Let $\mathbf{P}$ be a transition matrix of a finite homogenous irreducible and aperiodic discrete-time Markov chain (DTMC) with steady-state distribution π. In the following, all the vectors are row vectors. The norm $||x||$ of a vector is the sum of its elements (they are non negative), and $|\mathcal{A}|$ is the size of set $\mathcal{A}$.

We define $\nabla_{\mathbf{P}}[j] = \min_i \mathbf{P}[i,j]$ and $\Delta_{\mathbf{P}}[j] = \max_i \mathbf{P}[i,j]$. These two quantities are clearly associated to bounds on the steady state distribution. Indeed, we have the following trivial upper and lower component-wise bounding vectors:

Lemma 1

$$\nabla_{\mathbf{P}}[j] = \min_i \mathbf{P}[i,j] \leq \pi[j] = \sum_i \pi[i]\mathbf{P}[i,j] \leq \max_i \mathbf{P}[i,j] = \Delta_{\mathbf{P}}[j].$$

Proof: Remember that $\pi = \pi\mathbf{P}$, and $\pi[j]$ is between 0 and 1 for all j.

Remark that vector $\nabla_{\mathbf{P}}$ may be equal to $\mathbf{0}$, but $\Delta_{\mathbf{P}}$ is positive as the chain is irreducible.

The algorithms in [5] are based on this lemma and they try to improve the bound at each iteration. The key quantity to prove the convergence of the algorithms is the norm $||\nabla_{\mathbf{P}}||$. One can find in [5] a proof that Algorithm $I\nabla L$ (Algorithm 1) provides at each iteration a new lower bound $x^{(k)}$. It is worthy to remark that we can initialize the algorithm with $a = b = \nabla_{\mathbf{P}}$. Note that the "max" operator is applied component-wise. One can check that the conditions on the initialisation part of the algorithm require that $||\nabla_{\mathbf{P}}|| > 0$.

Algorithm 1. Algorithm Iterate ∇ Lower Bound (I∇L)

Require: $a \preceq \pi$, $b \preceq \nabla_{\mathbf{P}}$ and $b \neq \mathbf{0}$.
Ensure: Successive values of $x^{(k)}$.

1: $x^{(0)} = a$.
2: **repeat**
3: $\quad x^{(k+1)} = \max\left\{x^{(k)}, x^{(k)}\mathbf{P} + b(1 - ||x^{(k)}||)\right\}$.
4: **until** $1 - ||x^{(k)}|| < \epsilon$.

Similarly, one can derive a slightly different algorithm for the computation of an upper bound $y^{(k)}$ (see Algorithm $I\nabla U$ below). The only two differences with Algorithm $I\nabla L$ are the initialization step and the "min" operator in the iteration. We can use $c = \Delta_{\mathbf{P}}$ in the initialization step of algorithm (I∇U).

We gather in the following theorem the convergence results stated by one of us in [5] (see this reference for the proofs):

Theorem 1. *Let $\mathbf{P}$ be an irreducible and aperiodic stochastic matrix with steady-state probability distribution π. If $\nabla_{\mathbf{P}} \neq \mathbf{0}$, Algorithm $I\nabla L$ provides lower bounds*

Algorithm 2. Algorithm Iterate ∇ Upper Bound (I∇U)

Require: $c \succeq \pi$, $b \preceq \nabla_{\mathbf{P}}$ and $b \neq \mathbf{0}$.
Ensure: Successive values of $y^{(k)}$.

1: $y^{(0)} = c$.
2: **repeat**
3: $\quad y^{(k+1)} = \min\left\{y^{(k)}, y^{(k)}\mathbf{P} + b(1 - ||y^{(k)}||)\right\}$.
4: **until** $||y^{(k)}|| - 1 < \epsilon$.

for all components of π and converges to π for any value of the parameters a and b such that $a \preceq \pi$, $b \preceq \nabla_{\mathbf{P}}$, and $b \neq \mathbf{0}$. Furthermore, Algorithm I∇L converges faster than a geometric series with rate $1 - ||b||$.

Similarly, if $\nabla_{\mathbf{P}} \neq \mathbf{0}$, Algorithm I$\nabla$U gives a sequence of non-increasing upper bounds for all the components of π, and it converges to π for any parameters c and b which satisfy the constraints: $c \succeq \pi$, $b \preceq \nabla_{\mathbf{P}}$ and $b \neq \mathbf{0}$.

It is worthy to remark that the best initialization to speed up the convergence is $b = \nabla$. Let us first provide an example to illustrate some features of these algorithms.

Example 1. *Consider matrix* $\mathbf{P}1 = \begin{bmatrix} 0.1 & 0.3 & 0.2 & 0.4 \\ 0.3 & 0. & 0.3 & 0.4 \\ 0.2 & 0.4 & 0.4 & 0. \\ 0.8 & 0. & 0.2 & 0. \end{bmatrix}$.
Clearly, $\nabla_{\mathbf{P}1} = [0.1, 0., 0.2, 0.]$. *Algorithm (I$\nabla$L) is initialized with* $a = b = \nabla_{\mathbf{P}1}$ *and it gives the following sequence of lower bounds for the probabilities. The first column in the table is the iteration index. The last column is the residual (i.e. the amount of probability which has not been assigned at this iteration).*

k	1	2	3	4	$1 - \|\|x^{(k)}\|\|$
1	0.120000	0.110000	0.240000	0.040000	0.490000
2	0.174000	0.132000	0.259000	0.092000	0.343000
10	0.305143	0.200259	0.274736	0.200088	0.019773
15	0.311918	0.203534	0.275392	0.205833	0.003323
20	0.313057	0.204085	0.275502	0.206798	0.000559
25	0.313248	0.204177	0.275521	0.206960	0.000094
30	0.313280	0.204193	0.275524	0.206987	0.000016

Property 1. *Finally, note that one can derive a upper bound from the lower bound and the residual. We clearly have:*

$$x^{(k)} \preceq \pi \preceq x^{(k)} + (1 - ||x^{(k)}||)e,$$

where e is a row vector whose entries are all equal to 1.

When $\nabla = \mathbf{0}$, slightly different versions of these algorithms are given in [5] but they converge to an unknown multiple of π. We will propose in the next sections

some theoretical results and some algorithms to improve the usability of these algorithms. First we want to emphasis that a very important family of matrices satisfy the constraint on the norm of ∇.

2.1 Positive Matrices

Remember that a positive matrix $\mathbf{P}$ is such that all the entries are positive (i.e. $\mathbf{P}[i,j] > 0$).

Property 2. *Let* $\mathbf{P}$ *be a finite aperiodic positive matrix, all the entries of vector* ∇ *are positive.*

Indeed; as the state space is finite and all the entries of $\mathbf{P}$ are positive, the minimum of the entries over an arbitrary column is positive. This is a trivial result but it may have a large application area. It is very unlikely that a transition matrix of a Markov chain used for a performance evaluation model is positive. However, the ranking matrix used by Google is positive [10]. Remember that the ranking of pages is based on their relative steady-state probabilities.

Definition 1. *The Google matrix for page ranking is based on two arguments: the hyperlink structure of the web page which is reflected by matrix* S *in the following equation and the random surfer model. More precisely we define:*

$$\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\frac{e^T e}{n},$$

where n *is the size of the state space,* α *is a positive value smaller than* 1 *and* e *is a vector whose entries are all equal to* 1. $\mathbf{S}$ *and* $\mathbf{G}$ *are stochastic matrices.* $\mathbf{S}$ *is sparse and most likely reducible. Clearly* $\mathbf{G}$ *is positive.*

Property 3. *We have:* $\nabla(\mathbf{G}) \geq (1-\alpha)/n\ e$ *and* $||\nabla_{\mathbf{G}}|| \geq 1-\alpha$.

The proof is trivial and is omitted.

Note that it is not necessary to perform the multiplication of the vector $x^{(k)}$ by $\mathbf{G}$. It is sufficient to do the multiplication by matrix $\mathbf{S}$. Indeed, we have:

$$x^{(k)}\mathbf{G} = \alpha x^{(k)}\mathbf{S} + \frac{(1-\alpha)}{n}x^{(k)}e^T e = \alpha x^{(k)}\mathbf{S} + \frac{(1-\alpha)||x^{(k)}||}{n}e.$$

Property 4. *The complexity of the modified version of the vector matrix product is the number of non zero elements of* $\mathbf{S}$ *plus the number of states* n. *Furthermore, one can stop the iterations when it is clear that the ranking will not be changed: we just have to check if the intervals* $[x^{(k)}, y^{(k)}]$ *do not overlap anymore for the entries we want to rank. Note that this proof of ranking cannot be provided by the Power algorithm typically used.*

2.2 Denumerable Markov Chains

Assume now that the state space is infinite but denumerable and that there exist a positive δ and a non empty and finite set of states $\mathcal{A}_0$ such that:

- the number of transitions out of an arbitrary state is finite,
- for all state i in $\mathcal{A}_0$, the entries $\mathbf{P}[j,i]$ are larger than δ for every state j,
- for all state i not in $\mathcal{A}_0$, there exists a state j such that entry $\mathbf{P}[j,i]$ is zero.

As a consequence, $\nabla[i] > 0$ if and only if $i \in \mathcal{A}_0$. We also assume that the transitions in the matrix are described by a finite set of rules associated with probabilities provided by a high level description of the DTMC.

Clearly, these assumptions are extremely strong. But we cannot expect to numerically solve the steady-state distribution of an infinite Markov chain without further assumptions on the structure of the chain. The most interesting point is that the approach is completely different of the analysis of structured Markov chains such as QBD processes, and it may be used for chains that we do not know how to solve.

The following property states that we can analyze the chain with $I\nabla L$ Algorithm even if the state space is infinite. However one must take into account a graph representation of the problem to perform the vector-matrix product to avoid the explicit representation of the infinite state space. We will denote this framework for the computation as an ultra sparse representation. Let us begin with the ultra sparse representation of the vector and the matrix before we explain how we perform the successive operations of $I\nabla L$ algorithm.

Definition 2. *In an ultra sparse representation of the vector, one only stores the non zero entries (for instance as a linked list).*

Definition 3. *An ultra sparse representation of the matrix consists in an implicit representation of the transitions in the matrix described by the set of rules previously mentioned and an explicit representation of a finite subset of the rows of the matrix. Due to the assumption on the finiteness of the number of non zero entries per row, this explicit part is a finite rectangular matrix. If we assume that $\mathcal{A}_k$ is the set of rows we consider, we denote by $[P_k]$ this rectangular block from $\mathbf{P}$. Let m_k be the set of non zero elements of $[P_k]$. We assume that $[P_k]$ is stored in a sparse format. This representation changes at each iteration (k is the iteration index) because the number of considered rows increase at each iteration. Matrix $[P_k]$ is built by induction from the high level specification of the DTMC and previously computed block matrix $[P_{k-1}]$.*

Consider transition matrix as a directed graph. Let $\mathcal{A}$ be an arbitrary set of nodes, $\Gamma^+(\mathcal{A})$ is the set of nodes j in $\mathcal{S}$ such that there exists a node i in $\mathcal{A}$ with $P[i,j] > 0$. We assume that computing $\Gamma^+(\mathcal{A})$ is proportional to the number of non zero transitions out of states in $\mathcal{A}$. This is typically the case for many high level formalisms used to represent the transitions. This is also true if the matrix is already stored in an efficient sparse representation.

In a product of a dense vector by a sparse matrix, we need to know all non zero entries of the matrix. And they are infinite. In a multiplication of an ultra sparse vector by an ultra sparse matrix, we only need to build the rows of the matrix associated to non zero entries of the vector. Assume that, at step k, only the entries in $\mathcal{A}_k$ of vector $x^{(k)}$ are positive. The rows whose index are not in $\mathcal{A}_k$ will be multiplied by 0 during the vector-matrix product. Therefore, we do not have to build them, to store them and to multiply them. With an ultra sparse representation, we avoid to store an infinite number of non zero entries, and to multiply them by 0.

Property 5. *We assume that Algorithm $I\nabla L$ is initialized with $a = \nabla$ and $b = \nabla$. At iteration k, the indices of the non zero entries of $x^{(k)}$ are in set $\mathcal{A}_k$ given by the induction: $\mathcal{A}_k = \mathcal{A}_{k-1} \cup \Gamma^+(\mathcal{A}_{k-1})$, where $\mathcal{A}_0$ (previously defined) is such that $\nabla[i] > 0$ if and only if $i \in \mathcal{A}_0$. Furthermore, the computation of $x^{(k)}$ requires $O(m_k)$ steps.*

Proof: By construction $x^{(0)} = \nabla$. Therefore one can build the ultra sparse version of $x^{(0)}$ with a number of operations linear in the size of $\mathcal{A}_0$. Now consider the computation of $x^{(k)}$ for an arbitrary k. It consists in five operations once block $[P_k]$ has been built: the vector matrix product, the computation of the norm, the scalar multiplication of the vector, the additions of the vector and the maximum of the vectors.

- Update $[P_k]$: add the new rows, which have not been obtained before. This is equivalent to a BFS visit of the graph. The complexity is smaller than m_k due to the assumptions.
- The product of $x^{(k-1)}$ by $[P_k]$ is a product of dense vector of size $|\mathcal{A}_{k-1}|$ by a sparse matrix of size $|\mathcal{A}_{k-1}| \times |\Gamma^+(\mathcal{A}_{k-1})|$ with m_k non zero entries. The complexity is $O(m_k)$.
- The computation of the norm of $x^{(k-1)}$ requires $O(|\mathcal{A}_{k-1}|)$ computation steps.
- The product of a scalar by the vector needs $O(|\mathcal{A}_0|)$ computation steps.
- The addition needs $O(|\mathcal{A}_0|) + O(|\Gamma^+(\mathcal{A}_{k-1})|)$ steps.
- Finally, we compute the entry-wise maximum of a vector of size $|\mathcal{A}_0 \cup \Gamma^+(\mathcal{A}_{k-1})|$ and a vector of size $|\mathcal{A}_{k-1}|$. As these sets are not necessarily disjoint, one can get the entry-wise maximum vector after $|\mathcal{A}_0 \cup \Gamma^+(\mathcal{A}_{k-1}) \cup \mathcal{A}_{k-1}|$ computation steps. By induction, $A_0 \subset \mathcal{A}_k$ for all k. Therefore we obtain $A_k = \Gamma^+(\mathcal{A}_{k-1}) \cup \mathcal{A}_{k-1}$, and $x^{(k)}[i] > 0$ if and only if $i \in \mathcal{A}_k$.

□

The main computation step is the vector-matrix product but due to the ultra sparse product, we only consider a finite block of $\mathbf{P}$ at each iteration. The convergence results obtained in [5] are still valid. We only avoid to multiply the infinite part of the matrix by a null vector at each iteration. And the positive part of the vector finitely increases at each step because each row of the matrix has a finite non zero number of elements. We now consider the convergence time and give an example.

Property 6. *The theorems in [5] are proved for finite matrices. However due to the assumptions, the matrix is infinite but at each step we only consider a finite matrix. Thus the previous results on convergence hold. Using Property 2 in [5], one can easily state that the remaining mass of probability to be assigned after iteration k is upper bounded by $(1 - ||\nabla||)^{k+1}$. Therefore for any precision ϵ, we need less than $k0 = \lceil \frac{log(\epsilon)}{log(1-||\nabla||)} \rceil$ iterations to reach a residual smaller than ϵ. This is not related to the size of the matrix. After $k0$ iterations we have observed the convergence of the algorithm and all the entries of $x^{(k)}$ which have not been computed, are proved to be smaller than ϵ as their sum is smaller than ϵ.*

Finally, note that the assumption on the set of indices i such that $\nabla(i) > 0$ to be finite, is not necessary. If this set is infinite, we consider a value $b \leq \nabla$ which is not zero on a finite set of indices and which is used to define $\mathcal{A}_0$.

Example 2. *Consider the following DTMC (a toy example). The state space is* $\mathbb{N}$. *The transitions out of state i are described by:*

destination	*probability*
0	*1/10 + 1/(10(i+1))*
i	*1/5*
i+1	*2/5*
i+2	*3/10 - 1/(10(i+1))*

*Clearly this example satisfies the constraints. The state space is denumerable and the output degree of any node is finite (*3 *or* 4*). One can easily check that $\mathcal{A}_0 = \{0\}$, $\nabla[0] = 1/10$ and that $\nabla[i] = 0$ for all positive index i. We report in the next table the value of the lower bound for probability of state* 0*, the residual and the number of non zero entries of the lower bound vector computed after k iterations.*

k	$\pi(0)$	*Number of positive entries*	$1 - ||x^{(k)}||$
1	*0.13*	*3*	*0.81*
10	*0.159696*	*21*	*0.313811*
20	*0.162658*	*41*	*0.109419*
50	*0.163466*	*101*	*0.004638*
100	*0.163483*	*201*	*0.000024*

We use a precision of 10^{-5} *to stop the algorithm. We reach this precision after* 109 *iterations. And all the states which have not been computed have a probability smaller than* 10^{-5}*. We can also continue for a better precision:* 10^{-6} *is obtained at iteration* 131 *and* 10^{-8} *at iteration* 174*. For this last iteration, only* 349 *positive probabilities were computed. All the other entries of vector π are proved to be smaller than* 10^{-8}*. It it worthy to remark that some computed probabilities are also smaller than this threshold.*

Finally, using Property 1, we get from the computation at step 100*:*

$$0.163483 \leq \pi(0) \leq 0.163483 + 0.000024 = 0.163507$$

3 Stochastic Monotonicity

We present in this section how to combine stochastic comparison of matrices and our algorithms. We first give a short introduction to the stochastic comparison of DTMC. We then present two new results to efficiently obtain a bound. Indeed we do not compute the exact results but stochastic bounds or element wise bounds. The main objective of this section is to show that the algorithms can be used in conjunction with some stochastic monotonicity arguments. It must be clear that we impose that only the bounding matrix satisfy the constraints of our algorithms. Therefore, we can apply this technique to any matrix.

3.1 A Brief Presentation of Stochastic Comparison of DTMC

We refer to the books [14,12] for the theoretical issues for comparison of random variables and Markov chains. We assume that the state space is finite and endowed with a total ordering. Let $\mathcal{S}$ be the state space.

Definition 4. *Let X and Y be random variables. X is said to be less than Y in the strong stochastic sense, ($X \leq_{st} Y$) if and only if $E[f(X)] \leq E[f(Y)]$ for all non decreasing functions $f : S \to R$, whenever the expectations exist.*

This ordering provides the comparison of the underlying probability distribution functions: $X \leq_{st} Y \leftrightarrow Prob(X > a) \leq Prob(Y > a) \quad \forall a \in S$. Thus, it is more probable for Y to take larger values than for X. Since $\leq_{st}$ ordering yields the comparison of sample-paths, it is also known as sample-path ordering. We give in the next proposition the $\leq_{st}$ comparison in the case of finite state space which is more suitable for an algorithmic verification.

Property 1. Let X, Y be random variables taking values on $\{1, 2, \cdots, n\}$ and p, q be probability vectors which are respectively denoting distributions of X and Y, $X \leq_{st} Y$ iff $\sum_{j=i}^{n} p[j] \leq \sum_{j=i}^{n} q[j] \quad \forall i = \{n, n-1, \cdots, 1\}$. It is worthy to remark that $X = Y$ implies that $X \leq_{st} Y$.

Example 3. *Consider $p = [0.4, 0.2, 0.3, 0.1]$ and $q = [0, 1, 0.5, 0.2, 0.2]$. One can check easily that $p \leq_{st} q$ as:*

$$\begin{bmatrix} 0.1 & \leq 0.2, \\ 0.3 + 0.1 & \leq 0.2 + 0.2, \\ 0.2 + 0.3 + 0.1 & \leq 0.5 + 0.2 + 0.2, \\ 0.4 + 0.2 + 0.3 + 0.1 & \leq 0.1 + 0.5 + 0.2 + 0.2. \end{bmatrix}$$

The stochastic comparison of random variables has been extended to the comparison of Markov chains. It is shown in Theorem 5.2.11 of [14, p.186] that monotonicity and comparability of the probability transition matrices of time-homogeneous Markov chains yield sufficient conditions to compare stochastically the chains. We first define the monotonicity and comparability of stochastic matrices and then, we present Vincent's algorithm.

Definition 5. *Let* $\mathbf{P}$ *and* $\mathbf{Q}$ *be two stochastic matrices.* $\mathbf{Q}$ *is said to be an upper bounding matrix of* $\mathbf{P}$ *in the sense of the strong stochastic order* ($\mathbf{P} \leq_{st} \mathbf{Q}$) *if*

$$\mathbf{P}_{i,*} \leq_{st} \mathbf{Q}_{i,*}, \quad \forall i$$

where $\mathbf{P}_{i,*}$ *denotes the* i^{th} *row of matrix* $\mathbf{P}$.

Definition 6. *Let* $\mathbf{P}$ *be a stochastic matrix.* $\mathbf{P}$ *is said to be stochastically st-monotone if for any probability vectors* p *and* q,

$$p \leq_{st} q \implies p\,\mathbf{P} \leq_{st} q\,\mathbf{P}.$$

This is a rather general definition of monotonicity. In the case of finite state space with a total ordering, we obtain a much easier characterization of st-monotone stochastic matrices. $\mathbf{P}$ *is st-monotone if and only if* $\mathbf{P}_{i,*} \leq_{st} \mathbf{P}_{i+1,*}$ *for all state* i *such that state* $i+1$ *exists.*

One can easily verify in Example 4 that matrix $\mathbf{P}4$ is not monotone while both bounds (upper and lower) are st-monotone.

The following corollary allows us compare the steady-state distributions of Markov chains when they exist.

Corollary 1. *Let* $\mathbf{Q}$ *be a monotone, upper bounding matrix for* $\mathbf{P}$ *for the st-ordering. If the steady-state distributions (*$\mathbf{\Pi_P}$ *and* $\mathbf{\Pi_Q}$*) exist, then* $\mathbf{\Pi_P} \leq_{st} \mathbf{\Pi_Q}$.

Stochastic comparison and monotonicity can be represented by linear inequalities. Once we have derived a set of equalities instead of inequalities, and once we have ordered them, we obtain a constructive way to design a monotone upper bounding stochastic matrix $\mathbf{Q}$ for an arbitrary stochastic matrix $\mathbf{P}$. This algorithm is known as Vincent's algorithm [1,9].

$$\begin{cases} \sum_{k=j}^{n} \mathbf{Q}[1,k] & = \sum_{k=j}^{n} \mathbf{P}[1,k] \\ \sum_{k=j}^{n} \mathbf{Q}[i+1,k] & = max(\sum_{k=j}^{n} \mathbf{Q}[i,k], \sum_{k=j}^{n} \mathbf{P}[i+1,k]) \quad \forall\, i,j \end{cases} \tag{1}$$

A slightly different version is used to compute a lower bounding monotone matrix:

$$\begin{cases} \sum_{k=j}^{n} \mathbf{Q}[n,k] & = \sum_{k=j}^{n} \mathbf{P}[n,k] \\ \sum_{k=j}^{n} \mathbf{Q}[i-1,k] & = min(\sum_{k=j}^{n} \mathbf{Q}[i,k], \sum_{k=j}^{n} \mathbf{P}[i-1,k]) \quad \forall\, i,j \end{cases} \tag{2}$$

Due to the fact that matrices $\mathbf{P}$ and $\mathbf{Q}$ are stochastic, one can use the following version of the constraints for a lower bound. This version is more convenient for the extensions we propose in the next sections.

$$\begin{cases} \sum_{k=j}^{n} \mathbf{Q}[n,k] & = \sum_{k=j}^{n} \mathbf{P}[n,k] \\ \sum_{k=1}^{j} \mathbf{Q}[i-1,k] & = max(\sum_{k=1}^{j} \mathbf{Q}[i,k], \sum_{k=1}^{j} \mathbf{P}[i-1,k]) \quad \forall\, i,j \end{cases} \tag{3}$$

Example 4. *Consider the following stochastic matrix* $\mathbf{P}2$*. Vincent's algorithm gives:*

$$\mathbf{P}2 = \begin{bmatrix} 0.2 & 0.4 & 0. & 0.3 & 0.1 \\ 0.2 & 0.1 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0. & 0.4 & 0.4 & 0. \\ 0. & 0.5 & 0.2 & 0. & 0.3 \\ 0.2 & 0.1 & 0. & 0.5 & 0.2 \end{bmatrix}, \mathbf{LB}2 = \begin{bmatrix} 0.2 & 0.4 & 0.1 & 0.3 & 0. \\ 0.2 & 0.3 & 0.2 & 0.3 & 0. \\ 0.2 & 0.3 & 0.2 & 0.3 & 0. \\ 0.2 & 0.3 & 0.2 & 0.1 & 0.2 \\ 0.2 & 0.1 & 0. & 0.5 & 0.2 \end{bmatrix}, \mathbf{UB}2 = \begin{bmatrix} 0.2 & 0.4 & 0. & 0.3 & 0.1 \\ 0.2 & 0.1 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0. & 0.4 & 0.2 & 0.2 \\ 0. & 0.2 & 0.4 & 0.1 & 0.3 \\ 0. & 0.2 & 0.1 & 0.4 & 0.3 \end{bmatrix}.$$

Note that $\nabla(\mathbf{P}2) = \mathbf{0}$ *and the norm of* $\nabla(\mathbf{LB}2)$*and* $\nabla(\mathbf{UB}2)$ *are positive. Unfortunately this is not true in general as it can be seen with* $\mathbf{P}3$*:*

$$\mathbf{P}3 = \begin{bmatrix} 0.2 & 0.4 & 0. & 0.4 & 0. \\ 0.2 & 0.1 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0. & 0.4 & 0.4 & 0. \\ 0.5 & 0. & 0. & 0. & 0.5 \\ 0. & 0.3 & 0.2 & 0. & 0.5 \end{bmatrix}, \mathbf{LB}3 = \begin{bmatrix} 0.5 & 0.1 & 0 & 0.4 & 0. \\ 0.5 & 0. & 0.1 & 0.4 & 0. \\ 0.5 & 0. & 0.1 & 0.4 & 0. \\ 0.5 & 0. & 0.0 & 0. & 0.5 \\ 0. & 0.3 & 0.2 & 0. & 0.5 \end{bmatrix}, \mathbf{UB}3 = \begin{bmatrix} 0.2 & 0.4 & 0. & 0.4 & 0.0 \\ 0.2 & 0.1 & 0.3 & 0.2 & 0.2 \\ 0.2 & 0. & 0.4 & 0.2 & 0.2 \\ 0.2 & 0. & 0.3 & 0. & 0.5 \\ 0. & 0.2 & 0.3 & 0. & 0.5 \end{bmatrix}.$$

3.2 Using a Monotone Bound of a Non Monotone Matrix

We suppose that $\mathbf{P}$ is not monotone and we build a monotone lower bound of $\mathbf{P}$. Vincent's algorithm [1,9] gives a lower bound monotone matrix whose steady-state if it exists is a lower bound of $\pi_{\mathbf{P}}$. Unfortunately, the matrix we obtain with Vincent's algorithm is as hard to solve as the original one. Thus, we propose to use the degree of freedom in the constraints for the stochastic bounds to compute a monotone lower bound matrix $\mathbf{M}$ such that $\nabla_{\mathbf{M}} > \mathbf{0}$.

The first algorithm receives as parameters a column index i and a positive value ϵ and it builds a monotone lower bound such that $\nabla_{\mathbf{M}}[i] \geq \epsilon$, while the vector ∇ for the original matrix is equal to $\mathbf{0}$. Note that the algorithm may fail and return an error message if it is not possible to build such a matrix. Remark that this approach is related to the patterns presented by Busic in [3].

Algorithm MLBPL (3) proceeds in three phases. During phase one, it computes column 1 to column i of the monotone lower bound of $\mathbf{P}$ using Vincent's algorithm (more precisely the set of constraints in system 3). Then, the second phase is used to modify column i: we make all the entries larger than ϵ. If an entry is already larger, we do not change it. Finally we compute columns $i+1$ to n using again Vincent's algorithm.

Theorem 2. *If* $\mathbf{P}$ *satisfies* $\sum_{l=1}^{i} \mathbf{P}[k,l] \leq 1-\epsilon$ *for all* k*, then matrix* $\mathbf{M}$*, obtained by Algorithm 3 with parameters* i *and* ϵ*, is a monotone stochastic matrix which is a lower bound of* $\mathbf{P}$ *such that* $||\nabla(\mathbf{M})|| \geq \epsilon$*.*

The proof, similar to the proof of Vincent's Algorithm, is omitted for the sake of concision. The condition $\sum_{l=1}^{i} \mathbf{P}[k,l] \leq 1-\epsilon$ is required for matrix $\mathbf{M}$ to be stochastic.

Algorithm 3. Monotone Lower Bounds with Positive Nabla (MLBPN)

Require: a state i, a value $1 > \delta > 0$, matrix $\mathbf{P}$
Ensure: matrix $\mathbf{M}$, a monotone lower of $\mathbf{P}$ such that $\nabla(\mathbf{M}) > \mathbf{0}$.

for $j = 1$ to i **do**
  $\mathbf{M}[n, j] = \mathbf{P}[n, j]$
end for
for $k = n - 1$ down to 1 **do**
  for $j = 1$ to i **do**
    $\mathbf{M}[k, j] = max\left(\sum_{l=1}^{j} \mathbf{P}[k, l], \sum_{l=1}^{j} \mathbf{M}[k + 1, l]\right) - \sum_{l=1}^{j-1} \mathbf{M}[k, l]$
  end for
end for
for $j = 1$ to n **do**
  $\mathbf{M}[j, i] = max(\mathbf{M}[j, i], \epsilon)$
end for
for $j = i + 1$ to n **do**
  $\mathbf{M}[n, j] = \sum_{l=i}^{j} \mathbf{P}[n, l] - \sum_{l=i}^{j-1} \mathbf{M}[n, l]$
end for
for $k = n - 1$ down to 1 **do**
  for $j = i + 1$ to n **do**
    $\mathbf{M}[k, j] = max\left(\sum_{l=1}^{j} \mathbf{P}[k, l], \sum_{l=1}^{j} \mathbf{M}[k + 1, l]\right) - \sum_{l=1}^{j-1} \mathbf{M}[k, l]$
  end for
end for

Example 5. *Consider again matrix* $\mathbf{P}3$*. It is not monotone (indeed row* 2 *and row* 3 *are not comparable). We apply Algorithm 3 with parameters* 3 *and* 0.1 *to obtain a lower bound:*

$$\begin{bmatrix} 0.5 & 0.1 & 0.1 & 0.3 & 0. \\ 0.5 & 0. & 0.1 & 0.4 & 0. \\ 0.5 & 0. & 0.1 & 0.4 & 0. \\ 0.5 & 0. & 0..1 & 0. & 0.4 \\ 0. & 0.3 & 0.2 & 0. & 0.5 \end{bmatrix}.$$

3.3 Using a Bound of a Monotone Matrix

Assume now that $\mathbf{P}$ is monotone. Thus, it is not necessary to build a monotone lower bounding matrix to apply the comparison theorem for DTMC. Indeed, it is sufficient that one of the two stochastic matrices we consider in the comparison is monotone. The algorithm simply needs that $\mathbf{M}$ is a stochastic lower bound of $\mathbf{P}$ and that $\nabla(\mathbf{M})$ has a positive norm. We proceed with the same three phases as in the previous approach. However, during the first phase, we simply copy the columns i to n of $\mathbf{P}$ into $\mathbf{M}$.

Theorem 3. *If* $\mathbf{P}$ *satisfies* $\sum_{l=1}^{i} \mathbf{P}[k, l] \leq 1 - \epsilon$ *for all* k*, then matrix* $\mathbf{M}$ *obtained with Algorithm 4 with parameters* i *and* ϵ *is a stochastic matrix which is a lower bound of* $\mathbf{P}$ *such that* $\nabla(\mathbf{M})(i) = \epsilon$*.*

The proof is omitted.

Algorithm 4. Lower Bounds with Positive Nabla (LBPN)

Require: a state i, a value $1 > \delta > 0$, matrix $\mathbf{P}$
Ensure: matrix $\mathbf{M}$, a lower of $\mathbf{P}$ such that $\nabla(\mathbf{M}) > \mathbf{0}$.

```
for k = n down to 1 do
    for j = 1 to i do
        M[k,j] = P[k,j]
    end for
end for
for j = 1 to n do
    M[j,i] = max(M[j,i], ε)
end for
for j = i+1 to n do
    M[n,j] = Σ_{l=i}^{j} P[n,l] − Σ_{l=i}^{j-1} M[n,l]
end for
for k = n−1 down to 1 do
    for j = i+1 to n do
        M[k,j] = max( Σ_{l=1}^{j} P[k,l], Σ_{l=1}^{j} M[k+1,l] ) − Σ_{l=1}^{j-1} M[k,l]
    end for
end for
```

Example 6. *We illustrate the approach with matrix* $\mathbf{M}4$ *which is given by Algorithm LBPN on matrix* $\mathbf{P}4$ *with parameters* 3 *and* 0.1.

$$\mathbf{P}4 = \begin{bmatrix} 0.2 & 0.5 & 0.0 & 0.3 & 0. \\ 0.1 & 0. & 0.5 & 0. & 0.4 \\ 0. & 0.15 & 0.05 & 0.3 & 0.5 \\ 0. & 0.1 & 0.. & 0.4 & 0.5 \\ 0. & 0.0 & 0.1 & 0.3 & 0.6 \end{bmatrix}, \quad \mathbf{M}4 = \begin{bmatrix} 0.2 & 0.5 & 0.1 & 0.2 & 0. \\ 0.1 & 0. & 0.5 & 0. & 0.4 \\ 0. & 0.15 & 0.1 & 0.25 & 0.5 \\ 0. & 0.1 & 0.1 & 0.3 & 0.5 \\ 0. & 0. & 0.1 & 0.3 & 0.6 \end{bmatrix}.$$

4 Filling the Matrix

We use some manipulations of the entries of the matrix to fill it, while we keep unchanged the steady-state distribution. This is a rather unusual approach as we typically avoid to fill the transition matrix to make the matrix sparse during the numerical analysis.

Definition 7. *Let* i *and* j *be two arbitrary distinct states, let* α *be positive, we define the transform* $F_{i,j,\alpha}$ *as follows:* $F_{i,j,\alpha}(\mathbf{M}) = \mathbf{P}$ *such that*

- $\mathbf{P}[k,l] = \mathbf{M}[k,l]$, *for all* k *and for all* $l \neq i$ *and* $l \neq j$.
- $\mathbf{P}[k,j] = (1-\alpha)\mathbf{M}[k,j]$, *for all* k, $k \neq j$.
- $\mathbf{P}[j,j] = (1-\alpha)\mathbf{M}[j,j] + \alpha$.
- $\mathbf{P}[k,i] = \mathbf{M}[k,i] + \alpha\mathbf{M}[k,j]$, *for all* k, $k \neq j$.
- $\mathbf{P}[j,i] = \mathbf{M}[j,i] + \alpha\mathbf{M}[j,j] - \alpha$,

Lemma 2. *If* M *is finite and ergodic and* $\alpha < min(1, \frac{\mathbf{M}[j,i]}{1-\mathbf{M}[j,j]})$ *, then* $F_{i,j,\alpha}(\mathbf{M})$ *is stochastic and ergodic.*

Proof: First, note that the summations of the elements of the rows of $F_{i,j,\alpha}(\mathbf{M})$ are all equal to 1. Furthermore, the constraint on α implies that the entry $[i,j]$ of this matrix is positive and that all the elements are non negative. Thus matrix $F_{i,j,\alpha}(\mathbf{M})$ is stochastic. Finally, as $\mathbf{M}$ is finite and ergodic, it must be irreducible and aperiodic. Clearly, all the transitions in $\mathbf{M}$ also exist in $F_{i,j,\alpha}(\mathbf{M})$. Remember that irreducibility and aperiodicity are monotone properties (i.e. the properties are still verified when we add new transitions). Therefore $F_{i,j,\alpha}(\mathbf{M})$ is finite, aperiodic and irreducible. It must be ergodic. □

Lemma 3. *Let π be the steady-state distribution of $\mathbf{M}$, consider two arbitrary distinct states i and j and an arbitrary positive value α, then $\pi F_{i,j,\alpha}(\mathbf{M}) = \pi$.*

Proof: consider an arbitrary state k, we have three cases to study:

- $k = i$. Let us compute $\sum_l \pi[l] F_{i,j,\alpha}(\mathbf{M})[l,i]$. By construction, we have after substitution:
$$\begin{aligned}\textstyle\sum_l \pi[l] F_{i,j,\alpha}(\mathbf{M})[l,i] &= -\alpha\pi[j] + \textstyle\sum_l \pi[l]\mathbf{M}[l,i] + \alpha \sum_l \pi[l]\mathbf{M}[l,j],\\ &= -\alpha\pi[j] + \pi[i] + \alpha\pi[j],\\ &= \pi[i].\end{aligned}$$
- $k = j$. Again we compute the summation and we substitute the definition of the transform to get:
$$\begin{aligned}\textstyle\sum_l \pi[l] F_{i,j,\alpha}(\mathbf{M})[l,j] &= \alpha\pi[j] + (1-\alpha)\textstyle\sum_l \pi[l]\mathbf{M}[l,j],\\ &= \alpha\pi[j] + (1-\alpha)\pi[j],\\ &= \pi[j].\end{aligned}$$
- $k \neq i$, $k \neq j$. After substitution:
$$\textstyle\sum_l \pi[l] F_{i,j,\alpha}(\mathbf{M})(l,k) = \sum_l \pi[l]\mathbf{M}(l,k) = \pi[k].$$

And the proof is complete. □

Property 7. *The transform $F_{i,j,\alpha}()$ requires at worst a number of steps linear in the size of the state space. It can be less if row j has a sparse representation. However one must remember that the goal of this approach is to fill row i due to the contribution of row j. Therefore one must consider the rows which have many non zero entries.*

Clearly, one can used the transform to fill row i of matrix $\mathbf{M}$ until the norm of ∇ becomes positive. Indeed one can apply many transforms with various columns j and factor α to fill row i.

Example 7. *Consider again matrix $\mathbf{P}2$ defined in Example 4. Clearly $\nabla(\mathbf{P}2)$ is $\mathbf{0}$. We apply $F_{1,2,0.1}$ on $\mathbf{P}2$ to get the following matrix:*

$$F_{1,2,0.1}(\mathbf{P}2) = \begin{bmatrix} 0.24 & 0.36 & 0 & 0.3 & 0.1 \\ 0.11 & 0.19 & 0.3 & 0.2 & 0.2 \\ 0.20 & 0 & 0.4 & 0.4 & 0 \\ 0.05 & 0.45 & 0.2 & 0. & 0.3 \\ 0.21 & 0.09 & 0 & 0.5 & 0.2 \end{bmatrix}.$$

And $\nabla(F_{1,2,0.1}(\mathbf{P}2)) = [0.05, 0, 0, 0, 0]$. We can now apply $I\nabla L$ and $I\nabla U$ algorithms. After 200 iterations the bounds for the probability are:

k	*1*	*2*	*3*	*4*	*5*	$1 - \|\|x^{(k)}\|\|$
lower	*0.149022*	*0.226813*	*0.198361*	*0.254866*	*0.170905*	*0.000033*
upper	*0.149028*	*0.226827*	*0.198375*	*0.254883*	*0.170917*	*-0.000030*

A careful inspection shows that $\alpha = 1/7$ *gives a better value for the norm of* ∇ *(i.e. in this case, it is the optimal value one can obtain with a transform with row* 2 *used to fill row* 1*):* $\nabla(F_{1,2,1/7}(\mathbf{P}2)) = [5/70, 0, 0, 0, 0]$.

Acknowledgement. This work was partially supported by a grant from CNRS GDR RO.

References

1. Abu-Amsha, O., Vincent, J.M.: An algorithm to bound functionals on Markov chains with large state space. In: 4th INFORMS Conference on Telecommunications, Boca Raton, Floride, E.U, Boca Raton, Floride, E.U. INFORMS (1998)
2. Berman, A., Plemmons, R.J.: Nonnegative Matrices in the Mathematical Sciences. Academic Press, NewYork (1994)
3. Busic, A., Fourneau, J.-M.: A matrix pattern compliant strong stochastic bound. In: 2005 IEEE/IPSJ International Symposium on Applications and the Internet Workshops (SAINT 2005 Workshops), Trento, Italy, pp. 260–263. IEEE Computer Society (2005)
4. Busic, A., Fourneau, J.-M.: A toolbox for component-wise bounds for steady-state distribution of a DTMC. In: QEST 2010, Seventh International Conference on the Quantitative Evaluation of Systems, W. sburg, Viginia, USA, pp. 81–82. IEEE Computer Society (2010)
5. Busic, A., Fourneau, J.-M.: Iterative component-wise bounds for the steady-state distribution of a markov chain. Numerical Linear Algebra with Applications 18(6), 1031–1049 (2011)
6. Dayar, T., Fourneau, J.-M., Pekergin, N.: Transforming stochastic matrices for stochastic comparison with the st-order. RAIRO Operations Research 37, 85–97 (2003)
7. Dayar, T., Fourneau, J.-M., Pekergin, N., Vincent, J.-M.: Polynomials of a stochastic matrix and strong stochastic bounds. In: Markov Anniversary Meeting, Charleston, pp. 211–228. Boson Books, Raleigh, North Carolina (2006)
8. Fourneau, J.-M., Le Coz, M., Pekergin, N., Quessette, F.: An open tool to compute stochastic bounds on steady-state distributions and rewards. In: 11th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2003), Orlando, FL. IEEE Computer Society (2003)
9. Fourneau, J.-M., Pekergin, N.: An Algorithmic Approach to Stochastic Bounds. In: Calzarossa, M.C., Tucci, S. (eds.) Performance 2002. LNCS, vol. 2459, pp. 64–88. Springer, Heidelberg (2002)
10. Langville, A.N., Meyer, C.D.: Google's PageRank and beyond - the science of search engine rankings. Princeton University Press (2006)
11. Semal, P.: Refinable bounds for large Markov chains. IEEE Trans. on Computers 44(10), 1216–1222 (1995)
12. Shaked, M., Shantikumar, J.G.: Stochastic Orders and their Applications. Academic Press, San Diego (1994)
13. Song, Y.: Monotone convergence of iterative methods for singular linear systems. BIT Numerical Mathematics 42, 611–624 (2002)
14. Stoyan, D.: Comparaison Methods for Queues and Other Stochastic Models. John Wiley and Sons, Berlin (1983)

Mean-Field Analysis of Markov Models with Reward Feedback

Anton Stefanek, Richard A. Hayden,
Mark Mac Gonagle, and Jeremy T. Bradley

Department of Computing,
Imperial College London, London SW7 2AZ
{as1005,rh,jb}@doc.ic.ac.uk

Abstract. We extend the *population continuous time Markov chain* formalism so that the state space is augmented with continuous variables accumulated over time as functions of component populations. System feedback can be expressed using accumulations that in turn can influence the Markov chain behaviour via functional transition rates. We show how to obtain mean-field differential equations capturing means and higher-order moments of the discrete populations and continuous accumulation variables. We also provide first- and second-order convergence results and suggest a novel normal moment closure that can greatly improve the accuracy of means and higher moments.

We demonstrate how such a framework is suitable for modelling feedback from globally-accumulated quantities such as energy consumption, cost or temperature. Finally, we present a worked example modelling a hypothetical heterogeneous computing cluster and its interaction with air conditioning units.

1 Introduction

The behaviour of large computing clusters is often controlled by feedback from various accumulated continuous quantities, such as temperature, energy consumption or total cost. For example, an air-conditioning controller in a server farm will react to the ambient temperature. At the same time, sophisticated thermally-aware schedulers [23] can use temperature sensors to regulate server operation and thus indirectly environmental temperature, creating a feedback loop.

Stochastic models of computing clusters will typically be very large and thus, due to state-space explosion, will often lie outside the capabilities of traditional performance analysis. However, the nature of these systems, consisting of many identically-behaving cooperating components is suitable for mean-field type analyses [e.g. 7, 10, 12]. Mean-field techniques have recently been extended to capture certain accumulated rewards [21]. We show how to further adapt this approach to allow modelling of feedback between the system model and the generated accumulated quantities.

In Section 2, we extend the discrete state space of a Markov population model with accumulated variables governed by integral equations. The accumulation

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 193–211, 2012.

functions can involve component populations and the discrete transition rate functions can depend on the accumulated variables, thus allowing feedback loops. In Section 2.2 we extend the mean-field techniques to analyse means and higher moments of component populations and accumulated variables. Section 3 justifies this approach by proving convergence to the solution of the mean field equations as the scale of the system increases. In Section 3.2, based on second-order convergence to a Gaussian process, we introduce a moment closure that improves the accuracy of the approximation when the rates contain occurrences of the minimum or maximum functions, common situations when modelling computer systems, for example, the process algebra *PEPA* [10] or *stochastic Petri nets* [19]. We demonstrate the techniques on a larger example of a heterogeneous computing cluster with controlled temperature in Section 4.

1.1 Related Work

Capturing the feedback interaction between process-based agents and continuously varying physical properties of a system falls in the realm of hybrid system modelling. In the field of performance analysis, an initial example of this would have been in FSPNs or fluid stochastic Petri nets [13] where fluid places are used to capture continuously varying quantities. Discrete Petri net behaviour was in turn governed by the level of a given fluid place. FSPNs could be simulated but were restricted to only one or two fluid places in practice.

A detailed comparative study of hybrid process algebras can be found in [15]. A common feature in each of these hybrid process algebras is the expression of continuous evolution via the embedding of ordinary differential equations in the process model itself. In contrast, Bortollussi *et al.* [3] have developed stochastic HYPE, a process formalism that generates both discrete and continuously varying dynamics from the semantics of the process model alone.

In this paper we present a process mechanism that expresses feedback control as a result of accumulated reward variables in Markov population models. Analysis of Markov Reward Models (MRMs) [18, 24] is if anything more computationally demanding than analysis of plain Markov models. In Stefanek *et al.* [21], we showed how a fluid approximation could be constructed for a class of MRMs, but we had no way of providing a feedback mechanism based on those reward values.

In this paper we show how accumulated reward variables can be used to influence transition guards and rates in a large Markov model. We have not endeavoured to express the continuously varying rewards and variables in a process-style language, as in [3]. Instead we have focused on showing convergence between the ODE solution of the resulting population CTMC model with reward accumulations (aPCTMC) and simulations of the underlying stochastic process. Further, we consider higher moments of rewards with feedback, something we believe has not been presented before for hybrid systems of such scale.

2 Markov Population Models with Accumulations

In this section, we define an extension of a continuous-time Markov population process (PCTMC). The state space of a PCTMC consists of vectors $\mathbf{x} \in \mathbb{Z}_+^n$ of integer-valued populations, where $\mathbf{x}_0$ is the initial configuration. Transitions of the Markov chain are defined via a set of transition classes $\mathcal{C}$. Each class $c \in \mathcal{C}$ specifies a difference vector $l_c \in \mathbb{Z}^n$ between the populations before and after such a transition occurs and a rate function $r_c \colon \mathbb{Z}_+^n \to \mathbb{R}_+$ defining the infinitesimal rate of transitions of class c as a function of populations in the given state.

We illustrate the following definitions on a PCTMC representing a simple client/system. We use the PEPA stochastic process algebra [12] to define a PCTMC:

$$\begin{array}{ll} \mathit{Client}_0 \stackrel{def}{=} (\mathit{data}, r_{\mathit{data}}).\mathit{Client}_1 & \mathit{Server}_0 \stackrel{def}{=} (\mathit{data}, r_{\mathit{data}}).\mathit{Server}_1 \\ \mathit{Client}_1 \stackrel{def}{=} (\mathit{task}, r_{\mathit{task}}).\mathit{Client}_0 & \mathit{Server}_1 \stackrel{def}{=} (\mathit{reset}, r_{\mathit{reset}}).\mathit{Server}_0 \end{array}$$

$$\mathbf{Clients}\{\mathit{Client}_0[N_C]\} \underset{\mathit{data}}{\bowtie} \mathbf{Servers}\{\mathit{Server}_0[N_S]\}$$

Here, the discrete state space consists of numerical vectors $\mathbf{x} = (C_0, C_1, S_0, S_1) \in \mathbb{Z}_+^4$ and the initial state is $(N_C, 0, N_S, 0)$, keeping track of the populations of clients and servers in their respective states. There are 3 transition classes in this model – one corresponding to the synchronised event where a client sends its data to a server and two independent events where the client and the server reset to their initial states. According to the PEPA operational semantics, the respective change vectors and rate functions are $l_1 = (-1, -1, 1, 1)$ with $r_1(\mathbf{x}) = \min(C_0, S_0) r_{\mathit{data}}$, $l_2 = (1, -1, 0, 0)$ with $r_2(\mathbf{x}) = C_1 \cdot r_{\mathit{task}}$ and $l_3 = (0, 0, 1, -1)$ with $r_3(\mathbf{x}) = S_1 \cdot r_{\mathit{reset}}$.

We augment the state space with a set of continuous variables governed by an auxiliary system of integral equations whose evolution may additionally depend on the discrete populations. The continuous variables can be used to track the evolution of associated quantities such as energy use or temperature. Furthermore, the rates of the Markovian evolution of the discrete populations may also depend on the value of these variables, thus allowing, for example, energy usage over time to feedback into the control of the system.

2.1 Definition

The state space of a *PCTMC with accumulations* (*aPCTMC*) is a subset of $\mathbb{Z}_+^n \times \mathbb{R}^m$ consisting of states $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \mathbb{Z}_+^n$ captures the discrete populations and $\mathbf{y} \in \mathbb{R}^m$ captures the continuous accumulation variables. The discrete populations evolve as in traditional PCTMCs, that is, according to a set $\mathcal{C}$ of transition classes. The associated rate functions are extended onto the full state space, that is $r_c : \mathbb{Z}_+^n \times \mathbb{R}^m \to \mathbb{R}_+$ We denote the discrete-state component of the associated stochastic process by $\mathbf{X}(t)$ with its initial state given by $\mathbf{x}_0$.

The evolution of the continuous variables $\mathbf{Y}(t)$ is given by an integral equation of the form:

$$\mathbf{Y}(t) = \mathbf{y}_0 + \int_0^t \mathbf{g}(\mathbf{X}(s), \mathbf{Y}(s))\,\mathrm{d}s \tag{1}$$

where $\mathbf{g} : \mathbb{Z}_+^n \times \mathbb{R}^m \to \mathbb{R}^m$ is an accumulation function and $\mathbf{y}_0$ is the initial state of the accumulation variables.

For example, in the client/server model we might wish to model generation of heat energy by servers when in the active state $Server_1$, resulting in an increase in the total energy in the server room. In order to model the heating–cooling process, we extend the discrete model also with air conditioning units:

$$Aircon_0 \stackrel{def}{=} (on, \lambda_{on}(t)).Aircon_1 \quad Aircon_1 \stackrel{def}{=} (off, \lambda_{off}(t)).Aircon_0$$

$$\Big(\mathbf{Clients}\{Client_0[N_C]\} \underset{data}{\bowtie} \mathbf{Servers}\{Server_0[N_S]\}\Big) \parallel \mathbf{Aircon}\{Aircon_0[N_A]\}$$

where the rates λ_{on} and λ_{off} are defined below. The active air conditioning units contribute to the cooling of the environment, by transferring heat energy out of the room. If we assume that the heat generation and cooling rates (r_{heat} and r_{cool}) are constant over time, the heat energy in the server room can be captured by an accumulated variable:

$$E(t) = E_0 + \int_0^t r_{heat} S_1(u) - r_{cool} A_1(u)\,\mathrm{d}u$$

where E_0 is the initial energy in the room.

We can introduce feedback into the system by making the air conditioning transition rates depend on the current temperature of the room. An approximate physical model for the temperature is:

$$T(t) = \frac{c}{v} E(t) \tag{2}$$

where c is a constant and v is the total volume of air in the room. One possible control policy for the air conditioning units might be: when the temperature is above a given threshold T_{thresh}, units switch on at some rate, otherwise active units switch off:

$$\lambda_{on}(t) = r_{on} \text{ if } T(t) > T_{thresh} \qquad \lambda_{off}(t) = r_{off} \text{ if } T(t) < T_{thresh}$$

and 0 otherwise.

In general, an aPCTMC process can be realised as a *piecewise deterministic Markov process (PDMP)* [5]. However, in order for the above construction to result in a uniquely well-defined PDMP on any finite interval of time, some regularity conditions are required. In particular, it is important that the possibility of infinitely many jumps of the discrete component in a finite period of time is prevented and also that the continuous component cannot grow unbounded in a finite period of time, that is, that it cannot explode. The following conditions are sufficient to achieve this, where $\mathcal{X} \subset \mathbb{Z}_+^n$ is defined to be the reachable state space of the discrete component:

1. There exist $A, B \in \mathbb{R}_+$ such that for all $\mathbf{x} \in \mathcal{X}$, $\mathbf{y} \in \mathbb{R}^m$ and $c \in \mathcal{C}$:
$$\|\mathbf{g}(\mathbf{x}, \mathbf{y})\| \leq A(\|\mathbf{y}\| + 1) \quad \text{and} \quad r_c(\mathbf{x}, \mathbf{y}) \leq B(\|\mathbf{y}\| + 1)$$
2. The function $\mathbf{g}(\mathbf{x}, \cdot)\colon \mathbb{R}^m \to \mathbb{R}^m$ satisfies a local Lipschitz condition for each $\mathbf{x} \in \mathcal{X}$.
3. For each $c \in \mathcal{C}$, the function $r_c(\mathbf{x}, \cdot)\colon \mathbb{R}^m \to \mathbb{R}_+$ is measurable for each $\mathbf{x} \in \mathcal{X}$.

Assumption 2 guarantees that, between discrete jumps, the continuous component is defined uniquely and exists as long as it does not explode. In fact, the only way that the above construction will fail is if the continuous component explodes, since, otherwise, the maximal jump rate is bounded by assumption 1. However, if the continuous component does explode, say, at time t^*, then for any $t < t^*$, we have:

$$\|\mathbf{Y}(t)\| \leq \|\mathbf{y}_0\| + \int_0^t \|\mathbf{g}(\mathbf{X}(s), \mathbf{Y}(s))\| \,\mathrm{d}s \leq \|\mathbf{y}_0\| + At^* + \int_0^t A\|\mathbf{Y}(s))\| \,\mathrm{d}s$$

Applying a version of Grönwall's lemma [e.g. 6, Page 498] yields:

$$\|\mathbf{Y}(t)\| \leq [\|\mathbf{y}_0\| + At^*] \exp(tA)$$

This implies that $\mathbf{Y}(t)$ cannot explode at time t^* since it is continuous and bounded by $[\|\mathbf{y}_0\| + At^*] \exp(t^*A)$ for any $t < t^*$. Thus we have a contradiction and have shown that, subject to the assumptions above, our construction is well-defined on finite intervals of time.

Note that transition rates can be defined using a discontinuous indicator function without breaking any of the above assumptions, such as the rates $\lambda_{on}(t)$ and $\lambda_{off}(t)$ above. Therefore the client/server model defines a valid aPCTMC model.

2.2 Mean-Field Approximations

It is straightforward to extend simulation algorithms for CTMCs to realise traces of the evolution of the discrete and continuous state components of aPCTMC models. However, in the case of large models, simulation suffers from high computational costs.

We show how to extend the efficient mean-field (a.k.a. *fluid-analysis*) approach for the analysis of massive CTMC models [e.g. 10, 12, 25] to the case of aPCTMC models. Specifically, define $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ by $\mathbf{f}(\mathbf{x}, \mathbf{y}) := \sum_{c \in \mathcal{C}} r_c(\mathbf{x}, \mathbf{y})\mathbf{l}_c$, for suitable real extensions of the functions r_c. Then an intuitive extension of the mean-field approach yields the following systems of integral equations:

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}(s), \mathbf{y}(s)) \,\mathrm{d}s \qquad \mathbf{y}(t) = \mathbf{y}_0 + \int_0^t \mathbf{g}(\mathbf{x}(s), \mathbf{y}(s)) \,\mathrm{d}s \tag{3}$$

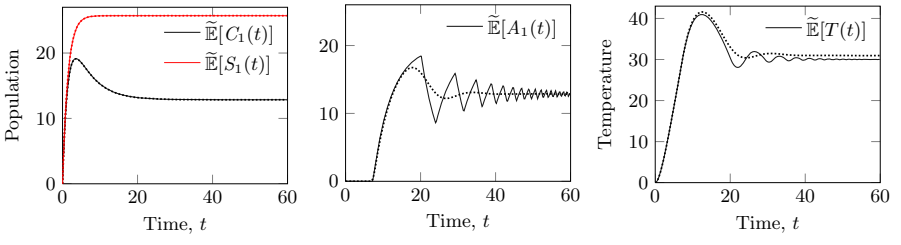
whose solutions can be interpreted as approximations to the means of the stochastic processes $\mathbf{X}(t)$ and $\mathbf{Y}(t)$, respectively, or for sufficiently large populations, as approximations to individual traces of the stochastic processes.

For example, in the client/server model, we get equations such as:

$$s_0(t) = N_S + \int_0^t r_{reset} s_1(u) - r_{data} \min(c_0(u), s_0(u)) \, \mathrm{d}u$$

where we use lower case letters for the mean-field approximations of the respective population and accumulation processes.

In Section 3 we show that, in the limit of large populations, the traces of the processes $\mathbf{X}(t)$ and $\mathbf{Y}(t)$ (and in particular the means $\mathbb{E}[\mathbf{X}(t)]$ and $\mathbb{E}[\mathbf{Y}(t)]$) converge to the mean-field solutions $\mathbf{x}(t)$ and $\mathbf{y}(t)$, respectively. Since we will usually be comparing means, we will adopt the notation $\widetilde{\mathbb{E}}[\mathbf{X}(t)]$ for $\mathbf{x}(t)$. For example, Figure 1 shows the numerical solutions to the mean-field model of Equation (3) as applied to the client/server model, compared to the estimates of the exact means sampled from 10^5 simulation runs of the stochastic process. In all figures in this paper, unless noted otherwise, the estimates from simulation are shown as dotted lines. Appendix A shows the specific values of parameters used to produce this figure and all the subsequent figures in this paper.



(a) Active clients/servers. (b) Active aircon. units. (c) Mean temperature.

Fig. 1. Approximation of mean component populations and accumulations in the client/server model

2.3 Higher-Order Moments

In addition to approximations of means of populations and the accumulation variables, systems of equations approximating higher-order moments may also be derived by extending existing approaches [e.g. 9, 10, 21] for CTMCs. The joint process $(\mathbf{X}(t), \mathbf{Y}(t))$ is clearly Markovian with infinitesimal generator $\mathcal{A}$ defined on continuous and bounded functions $h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ that are differentiable in the last m variables:

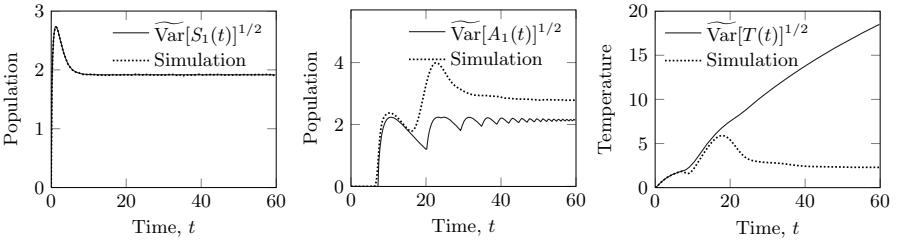
$$\begin{aligned}\mathcal{A}h(\mathbf{x}, \mathbf{y}) &:= \lim_{t \to 0} \frac{\mathbb{E}[h(\mathbf{X}(t), \mathbf{Y}(t)) | (\mathbf{X}(0), \mathbf{Y}(0)) = (\mathbf{x}, \mathbf{y})] - h(\mathbf{x}, \mathbf{y})}{t} \\ &= \sum_{i=1}^{m} g_i(\mathbf{x}, \mathbf{y}) \frac{\partial h}{\partial y_i}(\mathbf{x}, \mathbf{y}) + \sum_{c \in \mathcal{C}} r_c(\mathbf{x}, \mathbf{y}) [h(\mathbf{x} + \mathbf{l}_c, \mathbf{y}) - h(\mathbf{x}, \mathbf{y})]\end{aligned}$$

It thus follows by Dynkin's formula [e.g. 14, Lemma 17.21] that for $t \in \mathbb{R}_+$:

$$\mathbb{E}[h(\mathbf{X}(t), \mathbf{Y}(t))] = h(\mathbf{x}_0, \mathbf{y}_0) + \int_0^t \mathbb{E}[\mathcal{A}h(\mathbf{X}(s), \mathbf{Y}(s))] \, \mathrm{d}s \tag{4}$$

Equations for second-order moments can be obtained by choosing $h(\mathbf{x}, \mathbf{y}) := x_i y_j$, $x_i x_j$ and $y_i y_j$ for each appropriate i and j.[1] In fact, monomial functions of any order can be used to obtain equations for arbitrary order moments. However, if the functions $\mathbf{f}$ and $\mathbf{g}$ are non linear (as is usually the case), the term $\mathbb{E}[\mathcal{A}h(\mathbf{X}(s), \mathbf{Y}(s))]$ will involve expectations of non-linear functions of populations and will thus need to be simplified by applying some form of *moment-closure* approximation.

For example, in the client server model, the right hand side of Equation (4) will contain terms of the form $\mathbb{E}[\min(C_0(t), S_0(t))]$. In the past, and in Equation (3) above, the approximation $\min(\mathbb{E}[C_0(t)], \mathbb{E}[S_0(t)])$ has been used. This has been shown to work quite well in general for a large class of performance models [20]. However, if the process remains close to states where the arguments of the minimum function are equal, so-called *switch points*, for a long period of time, the accuracy of this approximation can decrease significantly for systems with low populations [20]. This is even more visible when the minimum function involves accumulated variables in aPCTMC models. We address this issue with a novel normal moment closure in Section 3.2.



(a) Active servers (b) Active aircon. units (c) Temperature

Fig. 2. Approximation of the evolution of standard deviation of populations and accumulations in the client/server model

Figure 2 shows approximations of standard deviations in the client/server model (we extend the $\widetilde{\mathbb{E}}[\cdot]$ notation to higher moments and expressions on them, such as variance). As demonstrated in previous work [20], this is quite accurate in the case of the client and server populations, which are not dependent on the accumulated variables, as depicted in Figure 2(a). However, in case of the population of air conditioning units (Figure 2(b)), and the temperature variable (Figure 2(c)), there are large quantitative and qualitative differences accumulated over time. Section 3.2 will discuss ways to improve the accuracy.

[1] Assuming that $\mathcal{X}$ is finite then the arguments of Section 2 guarantee that, over finite intervals of time, the process $(\mathbf{X}(t), \mathbf{Y}(t))$ is bounded to remain in some compact set, and then the boundedness requirement for the functions h need only be honoured on this set.

3 Convergence Properties

In this section of the paper we will prove that, in the limit of large populations, a suitably rescaled aPCTMC model converges to its mean-field approximation. We construct a sequence of aPCTMC models $\{(\mathbf{X}^N(t), \mathbf{Y}^N(t)) \in \mathbb{Z}_+^n \times \mathbb{R}^m\}_{N \in \mathbb{Z}_+}$. We assume that the elements $\mathcal{C}$ and $\mathbf{l}_c$ are fixed, but that the rate functions r_c^N and also $\mathbf{g}^N$ may vary with N. The initial conditions for the Nth model in the sequence are given by $(N\mathbf{x}_0, N\mathbf{y}_0)$ for some $(\mathbf{x}_0, \mathbf{y}_0) \in \mathbb{Z}_+^n \times \mathbb{R}^m$. For each model in this sequence, we assume that the assumptions of Section 2 are satisfied so that all of the processes are well defined and write $S^N \subseteq \mathbb{Z}_+^n$ for the reachable state space of the discrete component of the Nth process.

We assume further that the functions $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $f_c : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ can be defined independently of N as follows:

$$\mathbf{f}(\mathbf{x}, \mathbf{y}) := \sum_{c \in \mathcal{C}} \mathbf{l}_c f_c(\mathbf{x}, \mathbf{y}) := \sum_{c \in \mathcal{C}} (\mathbf{l}_c / N) r_c^N(N\mathbf{x}, N\mathbf{y}) \quad \mathbf{g}(\mathbf{x}, \mathbf{y}) := (1/N)\mathbf{g}^N(N\mathbf{x}, N\mathbf{y})$$

and that $\mathbf{f}$ and $\mathbf{g}$ satisfy local Lipschitz conditions on $\mathbb{R}^n \times \mathbb{R}^m$. Further, we assume that solutions to the mean-field model given by Equation (3) exist globally. Define the rescaled processes $\bar{\mathbf{X}}^N(t) := \mathbf{X}^N(t)/N$ and $\bar{\mathbf{Y}}^N(t) := \mathbf{Y}^N(t)/N$, then we require that there is some compact subset of $\mathbb{R}^n$ that contains all of the state spaces of the rescaled processes $\bar{\mathbf{X}}^N(t)$. Assume also that $\mathbf{g}^N(\mathbf{x}, \mathbf{y}) \leq C(\|\mathbf{x}\| + \|\mathbf{y}\| + 1)$ for all $\mathbf{x} \in S^N$ and $\mathbf{y} \in \mathbb{R}^m$ where $C \in \mathbb{R}_+$ is independent of N. Then by an application of Grönwall's lemma similar to that of Section 2, we have that for all $t \in [0, T]$, the rescaled stochastic processes and the mean-field approximations can be contained within a single compact set $S \subset \mathbb{R}^{n+m}$ that is independent of N.[2] Finally, we require that $r_c^N(\mathbf{x}, \mathbf{y}) \leq D(\|\mathbf{x}\| + \|\mathbf{y}\| + 1)$ for all $c \in \mathcal{C}$, $\mathbf{x} \in S^N$ and $\mathbf{y} \in \{Ns : s \in S\}$ where $D \in \mathbb{R}_+$ is independent of N.

The following theorem shows that the rescaled processes converge in probability to the mean-field approximation.

Theorem 1. *Under the assumptions and setup given above, we have, for any $T > 0$ and $\epsilon > 0$:*

$$\lim_{N \to \infty} \mathbb{P}\left\{ \sup_{t \in [0,T]} \|\bar{\mathbf{X}}(t) - \mathbf{x}(t)\| > \epsilon \right\} = 0 \quad \lim_{N \to \infty} \mathbb{P}\left\{ \sup_{t \in [0,T]} \|\bar{\mathbf{Y}}(t) - \mathbf{y}(t)\| > \epsilon \right\} = 0$$

Proof. We begin by representing each process $(\bar{\mathbf{X}}^N(t), \bar{\mathbf{Y}}^N(t))$ in terms of mutually independent rate-1 Poisson processes $\{P_c(t) : c \in \mathcal{C}\}$ by the *random-time change* approach [6]:

$$\bar{\mathbf{X}}^N(t) = \mathbf{x}_0 + \sum_{c \in \mathcal{C}} P_c\left(\int_0^t r_c^N(\mathbf{X}^N(s), \mathbf{Y}^N(s))\, \mathrm{d}s \right) \mathbf{l}_c / N$$

$$\bar{\mathbf{Y}}^N(t) = \mathbf{y}_0 + \int_0^t \mathbf{g}(\bar{\mathbf{X}}^N(s), \bar{\mathbf{Y}}^N(s))\, \mathrm{d}s$$

[2] Note that it is then only strictly necessary for this theorem that $\mathbf{f}$, f_c and $\mathbf{g}$ are defined on S rather than on the whole of $\mathbb{R}^{n+m}$.

On S, $\mathbf{f}$ and $\mathbf{g}$ are both Lipschitz continuous; let K be a Lipschitz constant for both functions. Now define:

$$D^N(t) := \sup_{s\in[0,t]} \left\| \bar{\mathbf{X}}^N(s) - \mathbf{x}_0 - \int_0^s \mathbf{f}(\bar{\mathbf{X}}^N(u), \bar{\mathbf{Y}}^N(u))\,\mathrm{d}u \right\|$$

and $\epsilon^N(t) := \|\bar{\mathbf{X}}^N(t) - \mathbf{x}(t)\| + \|\bar{\mathbf{Y}}^N(t) - \mathbf{y}(t)\|$. Then we have for $t \in [0, T]$:

$$\epsilon^N(t) \le D^N(T) + \int_0^t \|\mathbf{g}(\bar{\mathbf{X}}^N(s), \bar{\mathbf{Y}}^N(s)) - \mathbf{g}(\mathbf{x}(s), \mathbf{y}(s))\|\,\mathrm{d}s$$
$$+ \int_0^t \|\mathbf{f}(\bar{\mathbf{X}}^N(s), \bar{\mathbf{Y}}^N(s)) - \mathbf{f}(\mathbf{x}(s), \mathbf{y}(s))\|\,\mathrm{d}s \;\le D^N(T) + 2K \int_0^t \epsilon^N(s)\,\mathrm{d}s$$

and by Grönwall's inequality, we obtain $\epsilon^N(t) \le D^N(T)\exp(2KT)$. Now note that:

$$D^N(T) \le$$
$$\sup_{s\in[0,T]} \left\| \sum_{c\in\mathcal{C}} \frac{\mathbf{l}_c}{N} \left[P_c\left(\int_0^s r_c^N(\mathbf{X}^N(u), \mathbf{Y}^N(u))\,\mathrm{d}u \right) - \int_0^s r_c^N(\mathbf{X}^N(u), \mathbf{Y}^N(u))\,\mathrm{d}u \right] \right\|$$

which can be bounded above by $\sum_{c\in\mathcal{C}} \|\mathbf{l}_c\| \sup_{s\in[0,T]} |P_c(NCs)/N - Cs|$ for some $C \in \mathbb{R}_+$ independent of N. The result then follows by the strong law of large numbers for the Poisson process, which is equivalent to the functional strong law of large numbers [e.g. 26, Section 3.2], that is, for all $S \in \mathbb{R}_+$, $\sup_{s\in[0,S]} \|P_c(Ns)/N - s\| \to 0$ as $N \to \infty$ with probability 1. □

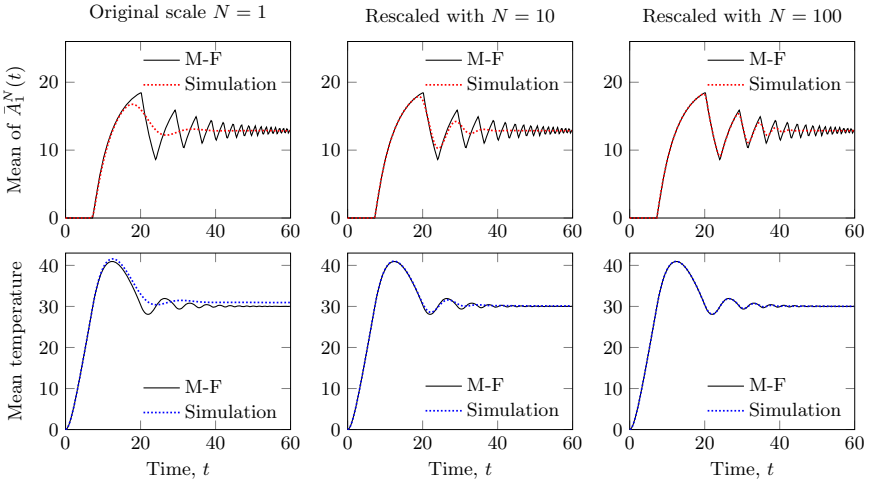


Fig. 3. Effect of rescaling on the first order mean-field approximation

In terms of the client/server model, scaling the number of components by N, and, in particular, the number of servers, can be assumed to require a room approximately N times larger in volume than that of the original system. Therefore

it makes sense if the initial heat energy content of the room E_0 is also scaled by N and the total heat energy content of the room is divided by N in order to obtain a physical model of the temperature as N increases, that is:

$$T^N(t) = \frac{c}{Nv} E^N(t) \quad \text{and} \quad E_0^N = NE_0$$

Theorem 1 requires a continuity assumption on the transition rate and accumulation functions in a aPCTMC model. The indicator functions λ_{on}^N and λ_{off}^N in the client/server model do not satisfy these requirements. However, Figure 3 does seem to suggest empirically that convergence may still occur. Indeed, extensions of Theorem 1 to discontinuous rate functions may be possible by considering mean-field models in terms of *differential inclusions* [2, 8], but we do not pursue this further in this paper.

Instead, we can replace the 0/1–valued indicator functions in λ_{on}^N and λ_{off}^N with a more smooth proportional control, setting:

$$\lambda_{on}^N(t) = (T^N(t) - T_{thresh})^+ r_{on} \tag{5}$$

where f^+ is the positive part of f, that is $\max(f, 0)$. For simplicity we also set $\lambda_{off}^N(t) = r_{off}$ for all N. With this modification, Theorem 1 then applies, and is illustrated in Figure 6.

3.1 Second-Order Convergence

In this section, we give a second-order Gaussian convergence result for the sequence of rescaled aPCTMC models, which will directly motivate the improved moment closure approach of Section 3.2. We maintain all of the notation of the previous section.

In addition to the assumptions of the previous section, we assume that we can decompose $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_i \mathbf{1}_{\{(\mathbf{x},\mathbf{y}) \in F_i\}} \mathbf{f}^i(\mathbf{x}, \mathbf{y})$ and $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \sum_j \mathbf{1}_{\{(\mathbf{x},\mathbf{y}) \in G_j\}} \mathbf{g}^j(\mathbf{x}, \mathbf{y})$ where $\{F_i\}$ and $\{G_j\}$ are finite collections of disjoint open sets in $\mathbb{R}^n \times \mathbb{R}^m$ such that for each i [resp. j], $\mathbf{f}^i$ [$\mathbf{g}^j$] is totally differentiable on $F^i \cap \text{int}(S)$ [$G^j \cap \text{int}(S)$] with uniformly continuous total derivative there. Then $\mathbf{f}$ [$\mathbf{g}$] has uniformly continuous total derivative on $\cup_i F_i \cap \text{int}(S)$ [$\cup_i G_i \cap \text{int}(S)$], which we write as $D\mathbf{f}$ [$D\mathbf{g}$].

Theorem 2. *Fix $T > 0$. Assume that the set $\{t \in [0,T] : (\mathbf{x}(t), \mathbf{y}(t)) \notin \cup_i F_i \cap \cup_j G_j \cap int(S)\}$ has Lebesgue measure zero. Then for mutually independent standard Brownian motions $\{B_c(t) : c \in \mathcal{C}\}$, the following equations have a unique strong solution [e.g. 16, Theorem 6.30] such that $(\mathbf{E}^X(t), \mathbf{E}^Y(t))$ is jointly-Gaussian:*

$$\mathbf{E}^X(t) := \int_0^t D\mathbf{f}(\mathbf{x}(s), \mathbf{y}(s)) \cdot (\mathbf{E}^X(s), \mathbf{E}^Y(s))^T \, ds + \sum_{c \in \mathcal{C}} B_c \left(\int_0^t f_c(\mathbf{x}(s), \mathbf{y}(s)) \, ds \right) \mathbf{l}_c$$

$$\mathbf{E}^Y(t) := \int_0^t D\mathbf{g}(\mathbf{x}(s), \mathbf{y}(s)) \cdot (\mathbf{E}^X(s), \mathbf{E}^Y(s))^T \, ds$$

Furthermore, $\left(\frac{\mathbf{X}^N(t)-N\mathbf{x}(t)}{\sqrt{N}}, \frac{\mathbf{Y}^N(t)-N\mathbf{y}(t)}{\sqrt{N}}\right) \Rightarrow (\mathbf{E}^X(t), \mathbf{E}^Y(t))$ *as* $N \to \infty$*, where the convergence is weak on* $D([0,T];\mathbb{R}^{n+m})$ *endowed with the uniform topology [e.g. 1].*[3]

Proof. We assume the representation of the processes $\bar{\mathbf{X}}^N(t)$ and $\bar{\mathbf{Y}}^N(t))$ given in Equation (5). Further it is possible [6, Corollary 5.5 and Remark 5.4] to construct, on the same probability space as these processes, mutually independent standard Brownian motions $\{B_c(t) : c \in \mathcal{C}\}$, such that:

$$Z_c := \sup_{t\in\mathbb{R}_+} \frac{|P_c(t) - t - B_c(t)|}{\log(2 \vee t)} < \infty \quad \text{almost surely}$$

From this it follows that as $N \to \infty$, almost surely:

$$\sqrt{N} \sup_{t\in[0,T]} \left\| \bar{\mathbf{X}}^N(t) - \mathbf{x}_0 - \int_0^t \mathbf{f}(\bar{\mathbf{X}}^N(s), \bar{\mathbf{Y}}^N(s))\,\mathrm{d}s \right.$$
$$\left. - \sum_{c\in\mathcal{C}} B_c\left(\int_0^t r_c^N(\mathbf{X}^N(s), \mathbf{Y}^N(s))\,\mathrm{d}s\right)(\mathbf{l}_c/N) \right\| \longrightarrow 0 \qquad (6)$$

A direct comparison of $\frac{\mathbf{X}^N(t)-N\mathbf{x}(t)}{\sqrt{N}}$ with $\mathbf{E}^X(t)$ and similarly for $\mathbf{E}^Y(t)$ using Equation (6) yields the result. We omit further details here for the sake of brevity. □

Theorem 2 also demands the continuity assumption on the transition rate and accumulation functions in an aPCTMC model so does not apply to the client/server model with threshold-based control. Indeed, Figure 4 shows that we do not even seem to observe convergence of the standard deviation approximation empirically in this case.

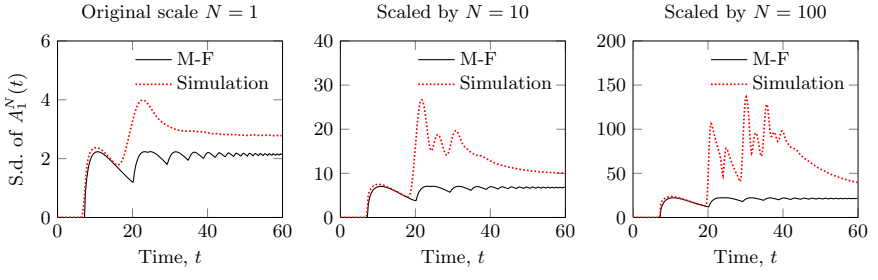


Fig. 4. Effect of scaling on the mean–field approximation of standard deviation of active air conditioning unit population

In the case of the proportionally-controlled client/server model introduced above, Theorem 2 can be applied, although, in our experiments, for populations that are of similar orders to those considered in Figure 4, the approximation of the standard deviation of the temperature variable can be very inaccurate — convergence occurs very slowly. In the next section, based on the Gaussian assumption justified by Theorem 2, we introduce a technique that can provide significant improvements.

[3] Informally, this is 'uniform convergence in distribution over $[0,T]$'.

3.2 Normal Approximations

Theorem 2 suggests that both the discrete and continuous components $\mathbf{X}(t)$ and $\mathbf{Y}(t)$ of an aPCTMC model can be approximated by a jointly Gaussian process for sufficiently large populations. The proportionally-controlled client/server example considered in this paper and, more generally, a large class of computer performance models, for example, those specified using PEPA, stochastic Petri nets or many-server queueing networks contain rates with occurrences of minimum functions. In such cases, Equation (4), when applied to extract a first moment, contains expectations of the form $\mathbb{E}[\min(\alpha, \beta)]$ where α and β are linear combinations of any of the discrete or continuous components in the model at some time t. The mean field approximation $\min(\mathbb{E}[\alpha], \mathbb{E}[\beta])$ can often be quite accurate, but Theorem 2 suggests an alternative. Because a sequence of aPCTMC processes converges to a Gaussian process, the marginal distributions at each point in time converge to multivariate normal random variables. Using a result for the moments of a minimum of two bivariate normal random variables [4], we can obtain the following approximation (where Φ and ϕ are the CDF and PDF, respectively, of a standard normal random variable):

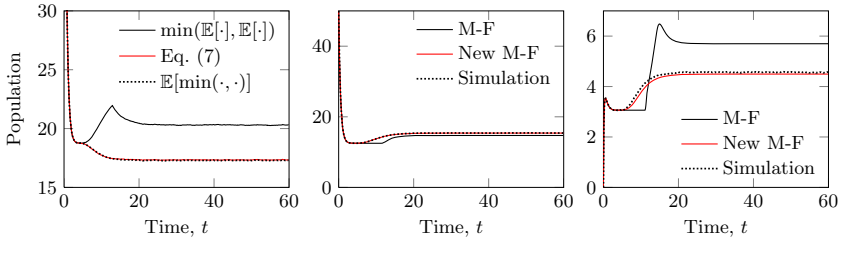
$$\mathbb{E}[\min(\alpha, \beta)] = \mathbb{E}[\alpha]\Phi(\Delta) + \mathbb{E}[\beta]\Phi(-\Delta) - \theta\phi(\Delta) \tag{7}$$

where $\theta := (\mathrm{Var}[\alpha] - 2\mathrm{Cov}[\alpha, \beta] + \mathrm{Var}[\beta])^{1/2}$ and $\Delta = (\mathbb{E}[\beta] - \mathbb{E}[\alpha])/\theta$. This uses only first- and second-order moments for which we can apply Equation (4) in order to extract equations governing their evolution.

For second-order moments, the mean field equations contain expectations of the form $\mathbb{E}[\gamma \min(\alpha, \beta)]$. Experiments suggest that the following approximation results in accurate approximations:

$$\mathbb{E}[\gamma \min(\alpha, \beta)] \approx \mathbb{E}[\gamma\alpha]\Phi(\Delta) + \mathbb{E}[\gamma\beta]\Phi(-\Delta) - \mathbb{E}[\gamma]\theta\phi(\Delta) \tag{8}$$

Figure 5 illustrates the improved accuracy for the discrete components of the model with proportional control.



(a) Expectation of $\min(\cdot, \cdot)$ (b) Mean of $S_0(t)$ (c) S.d. of $S_0(t)$

Fig. 5. Improved accuracy with normal approximations. Figure (a) compares the expected rate $\mathbb{E}[\min(C_0(t), S_0(t))r_{data}]$ with the two approximations, as obtained from simulation. Figures (b) and (c) show the effect of the different approximations when used in the mean-field equations for the mean and standard deviation of the S_0 population.

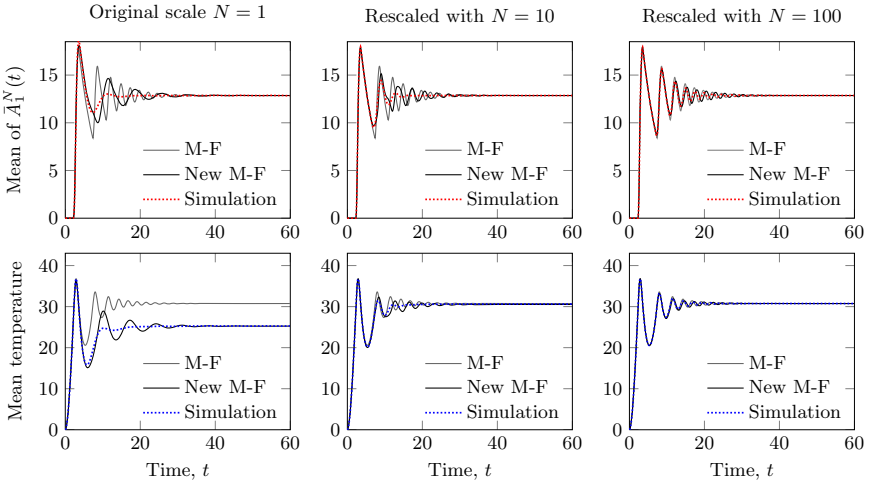


Fig. 6. Effect of scaling on the mean approximation when using the new normal approximations

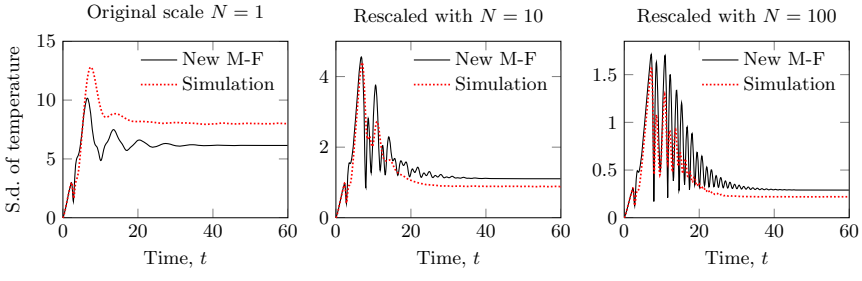


Fig. 7. Effect of scaling on standard deviation of temperature when using the new normal approximations

It is straightforward to adapt the result of [4] to obtain an expression for the maximum of bivariate normal random variables, so that an analogous approximation can be applied also to the proportional control expression (Equation (5)). Figure 6 compares simulation estimates with the numerical solution to the mean-field equations obtained from Equation (4) and with solutions to the new set of mean-field equations obtained by replacing occurrences of the minimum and maximum function according to the methods of this section. We see that this results in significant improvements in accuracy.

Figure 7 shows further that the normal moment closure can result in an accurate approximation of the standard deviation of the temperature even at relatively low scales of the system.

4 Worked Example

In this section we demonstrate the aPCTMC formalism and the efficient mean-field techniques on a larger example of a heterogeneous computing cluster. Similar to the client/server model, we consider a high level abstraction of the system. We assume that there are two types of servers in the cluster — ones with low (class A) and ones with high power consumption (class B), respectively. Clients in the system submit two types of jobs — with low (type 1) and high loads (type 2) on the servers. As in the client/server model, we include air conditioning units that maintain the ambient temperature in the room. Additionally, servers are capable of entering a sleep mode in the case that the temperature increases above a threshold. Unlike in the case of the client/server model where the client and server components of the discrete state space were unaffected by the accumulated variables, this will result in an aPCTMC with a complete dependence between the discrete components and the accumulated variables.

We use the PEPA process algebra to concisely describe the aPCTMC model ($j \in \{A, B\}$ is a server class and $i \in \{1, 2\}$ is a job type):

$$\begin{aligned}
&Client \stackrel{def}{=} \textstyle\sum_{i=1}^{2} (queue_i, r_{q,i}).Job_i \qquad Job_i \stackrel{def}{=} (service_i, r_{service_i}).Client \\
&Server^j \stackrel{def}{=} \textstyle\sum_{i=1}^{2} (service_i^j, r_{service,i}).Server_i^j + (sleep, \lambda_{sleep}(t)).Server^j_{sleep} \\
&Server_i^j \stackrel{def}{=} (reset, r_{reset}).Server^j \qquad Server^j_{sleep} \stackrel{def}{=} (wakeup, r_{wakeup}).Server^j
\end{aligned}$$

$$\begin{aligned}
&\big(\mathbf{Servers}\{Server^A[N_{SA}] | Server^B[N_{SB}]\}\big) \parallel \mathbf{Aircon}\{Aircon_0[N_A]\} \\
&\qquad \underset{\{service_i \,|\, 1 \le i \le 4\}}{\bowtie} \mathbf{Clients}\{Client[N_C]\}
\end{aligned}$$

with rates $\lambda_{off}(t) = r_{on}$ and

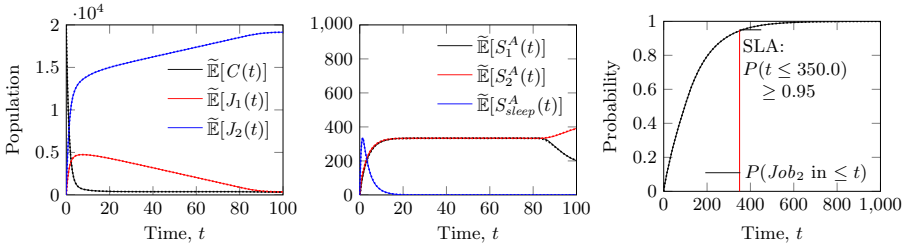
$$\lambda_{sleep}(t) = (T(t) - T_{sleep})^+ \cdot r_{j,sleep} \quad \text{and} \quad \lambda_{on}(t) = (T(t) - T_{thresh})^+ \cdot r_{on}$$

where temperature is defined as in Equation (2) and the energy variable is

$$E(t) = E_0 + \int_0^t \sum_j (S^j(u)c_{j,s} + S^j_{sleep}(u)c_{j,sl} + S^j_1(u)c_{j,1} + S^s(u)c_{j,2}) - A_1(u)c_a \mathrm{d}u$$

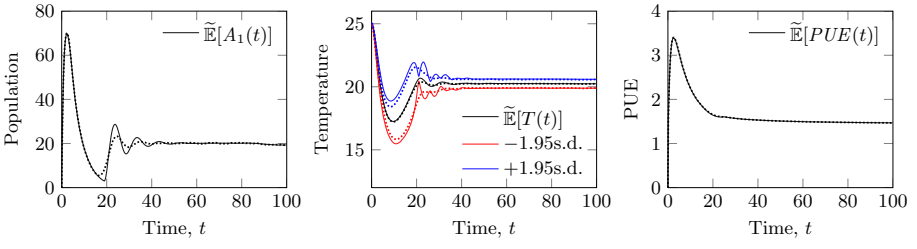
for some constants $c_{j,s}, c_{j,sl}, c_{j,1}, c_{j,2}, c_a$.

Additionally, we transform the model so that it is possible to use mean-field techniques to calculate cumulative distribution functions of various passage-time random variables [11]. We will compute the time until an individual client executes its first high load job. Such measures are often used when expressing service level agreements (SLAs). The example will show how the presented framework can be used to study the trade-off between SLA satisfaction and the energy efficiency of the system. An increasingly common metric assessing energy efficiency of data centres is the *Power Usage Efficiency (PUE)* metric [17], calculated as the ratio between the total energy consumption and the energy used by the servers. In the above model, we can model the total energy consumption as an accumulated variable:



(a) Mean client populations (b) Mean server populations (c) Passage time CDF

Fig. 8. Means of client/server populations and passage-time CDF in the computing cluster model



(a) Active aircon. units (b) Temperature (c) PUE

Fig. 9. Mean population of active airconditioning units and mean temperature in the cluster model. The approximation of standard deviation in figure (b) was obtained by applying the normal min closure from Section 3.2.

$$P(t) = \int_0^t \sum_j (p_{j,sl} S^j_{sleep}(u) + p_{j,s} S^j(u) + p_{j,1} S^j_1(u) + p_{j,2} S^j_2(u)) + p_a A_1(u) \mathrm{d}u$$

for some constants $p_{j,s}, p_{j,sl}, p_{j,1}, p_{j,2}, p_a$.

The quantity $U(t)$ represents the energy used for computation and is defined as $P(t)$, omitting the contribution of the air conditioning units and the servers in the sleeping state. To obtain an approximation of the mean PUE, we compute $\widetilde{\mathbb{E}}[P(t)]/\widetilde{\mathbb{E}}[U(t)]$ for sufficiently large t (1000 in the examples below).

Figure 8 shows the mean populations of client and server-A components and the passage-time CDF as obtained by the mean-field analysis. Figure 9 shows the mean population of air conditioning units, its effect on the mean controlled temperature and the PUE of the system.

One benefit of mean-field analysis is the relatively low computational cost of numerically integrating the mean-field equations. This allows the evaluation of a large number of system configurations in a short time. For example, we can look at the relationship between the two temperature thresholds T_{thresh} and T_{sleep} that specify when the air conditioning units start contributing to cooling and servers switch to sleep mode, respectively. We fix the server threshold at 23

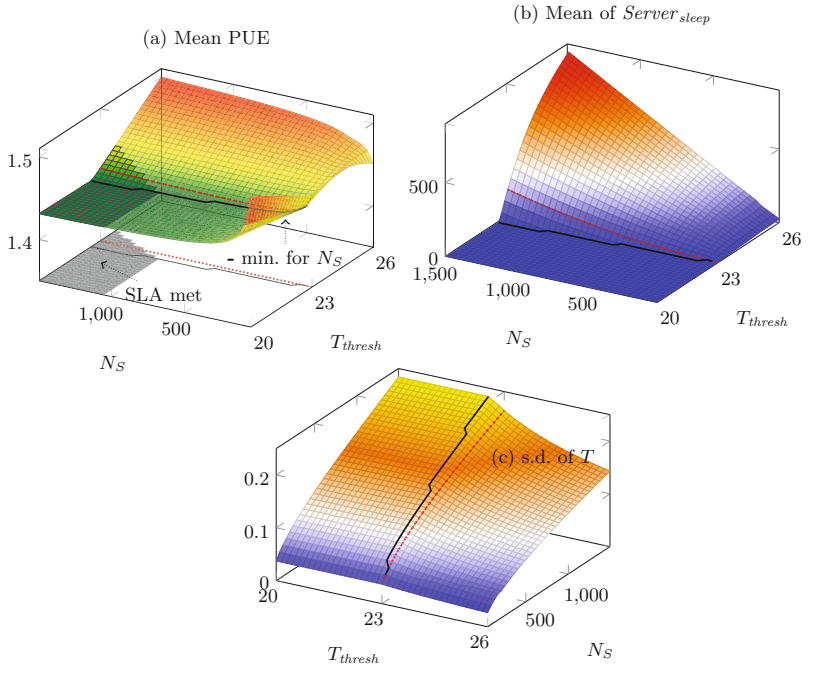


Fig. 10. The effect of cooling threshold and the number of servers on the steady state PUE metric and the number of servers in sleeping state. For each initial server population N_S, the thick black line shows the threshold under which the minimum PUE is achieved.

units and search for the best air conditioning threshold. Our target measure to minimise will be the PUE in steady state of the system and the constraints are given by requiring satisfaction of the above SLA. Figure 10 explores a range of system configurations with the number of servers of each type $N_S = N_{SA} = N_{SB}$ varying between 50 and 1500 and the threshold T_{thresh} varying between 20 and 26 units.

Figure 10(a) shows the mean steady-state PUE for each configuration. For each size of the computing cluster given by a value of N_S, there is an optimal value of T_{thresh} achieving a minimal PUE metric. These thresholds and the corresponding optimal PUE values are shown by the thick solid line.

It can be seen that this is slightly below the server threshold, shown as the red dotted line. For example, for $N_S = 850$, the value of T_{thresh} achieving the optimal PUE is 22.7. The SLA is achieved only when there are sufficiently many servers in the system, shown as the darker region on the surface plot. Figure 10(b) shows that the optimal PUE line minimises the number of sleeping servers, while keeping the air conditioning units as lightly loaded as possible. Figure 10(c) shows that line of minimum PUE separates the region with maximal standard deviation of the temperature variable.

5 Conclusion and Future Work

We have introduced the aPCTMC formalism, an extension to Markov population models that allows efficient modelling of feedback from accumulated quantities. We have extended the existing mean-field techniques to provide means and higher moments of populations and accumulations for aPCTMC models. Furthermore, we have provided convergence results justifying the mean-field approximation in the first- and second-order cases. The second-order result shows that sequences of aPCTMC models with increasing component populations converge to a jointly Gaussian process. This justifies the novel use of a normal approximation of minimum and maximum functions in the mean-field equations, resulting in significantly improved accuracy in both first- and second-order cases.

We have demonstrated the new framework on a substantial example of a computing cluster where server behaviour reacts to the ambient temperature controlled by an air conditioning system. An important advantage of the mean-field techniques is the low computational cost that can be used to explore a large number of different system configurations.

All of the numerical results in this paper were produced using a prototype implementation of the techniques in an extension to the *Grouped PEPA Analyser tool* [22].

In future, we plan to more formally investigate the accuracy improvements possible from the normal moment closures detailed in this paper. We also plan to investigate more complex accumulation mechanisms. For example, Gaussian noise might be introduced into the accumulation equations to account for error in sensor measurements of the continuous quantities.

Acknowledgment. Anton Stefanek, Richard A. Hayden and Jeremy T. Bradley are funded by EPSRC on the Analysis of Massively Parallel Stochastic Systems (AMPS) project (reference EP/G011737/1).

References

[1] Billingsley, P.: Convergence of Probability Measures. John Wiley & Sons (1968)
[2] Bortolussi, L.: Hybrid Limits of Continuous Time Markov Chains. In: QEST 2011, pp. 3–12. IEEE, Aachen (2011)
[3] Bortolussi, L., Galpin, V., Hillston, J.: HYPE with stochastic events. EPTCS 57, 120–133 (2011)
[4] Cain, M.: The moment-generating function of the minimum of bivariate normal random variables. The American Statistician 48(2), 124–125 (1994)
[5] Davis, M.H.A.: Markov models and optimization. Chapman & Hall/CRC (1993)
[6] Ethier, S.N., Kurtz, T.G.: Markov Processes: Characterization and Convergence. Wiley (2005)
[7] Gast, N., Bruno, G.: A mean field model of work stealing in large-scale systems. In: SIGMETRICS, p. 13. ACM, New York (2010)
[8] Gast, N., Gaujal, B.: Mean field limit of non-smooth systems and differential inclusions. SIGMETRICS Performance Evaluation Review 38(2), 30–32 (2010)

[9] Gillespie, C.S.: Moment-closure approximations for mass-action models. IET Systems Biology 3(1), 52–58 (2009)
[10] Hayden, R.A., Bradley, J.T.: A fluid analysis framework for a Markovian process algebra. Theoretical Computer Science 411(22-24), 2260–2297 (2010)
[11] Hayden, R.A., Stefanek, A., Bradley, J.T.: Fluid computation of passage-time distributions in large Markov models. Theoretical Computer Science 413(1), 106–141 (2012)
[12] Hillston, J.: Fluid flow approximation of PEPA models. In: QEST, pp. 33–42. IEEE (September 2005)
[13] Horton, G., Kulkarni, V.G., Nicol, D.M., Trivedi, K.S.: Fluid stochastic Petri nets: Theory, applications, and solution techniques. European Journal of Operational Research 105(1), 184–201 (1998)
[14] Kallenberg, O.: Foundations of Modern Probability. Springer (2002)
[15] Khadim, U.: A comparative study of process algebras for hybrid systems. Computer Science Report 06–23, Technische Universiteit Eindhoven (2006)
[16] Klebaner, F.C.: Introduction to stochastic calculus with applications, 2nd edn. Imperial College Press (2006)
[17] Rawson, A., Pfleuger, J., Cader, T.: Data Center Power Efficiency Metrics: PUE and DCiE. The Green Grid (2007)
[18] de Souza e Silva, E., Gail, R.: An algorithm to calculate transient distributions of cummulative rate and impulse-based rewards. Communications in Statistics: Stochastic Models 14(3), 509–536 (1998)
[19] Silva, M., Júlvez, J., Mahulea, C., Vázquez, C.R.: On fluidization of discrete event models: observation and control of continuous Petri nets. Discrete Event Dynamic Systems 21(4), 427–497 (2011)
[20] Stefanek, A., Hayden, R.A., Bradley, J.T.: A new tool for the performance analysis of massively parallel computer systems. In: QAPL. Electronic Proceedings in Theoretical Computer Science (March 2010)
[21] Stefanek, A., Hayden, R.A., Bradley, J.T.: Fluid Analysis of Energy Consumption using Rewards in Massively Parallel Markov Models. In: Computing, p. 121. ACM Press (2011)
[22] Stefanek, A., Hayden, R.A., Bradley, J.T.: GPA – A Tool for Fluid Scalability Analysis of Massively Parallel Systems. In: QEST, pp. 147–148. IEEE (September 2011)
[23] Tang, Q., Gupta, S., Varsamopoulos, G.: Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. IEEE Transactions on Parallel and Distributed Systems 19(11), 1458–1472 (2008)
[24] Telek, M., Rácz, S.: Numerical analysis of large Markovian reward models. Performance Evaluation, 36–37, 95–114 (August 1999)
[25] Tribastone, M., Gilmore, S., Hillston, J.: Scalable Differential Analysis of Process Algebra Models. IEEE Transactions on Software Engineering 38(1), 205–219 (2012)
[26] Whitt, W.: Internet supplement to Stochastic-Process Limits (2002), http://www.columbia.edu/~ww2040/supplement.html

A Parameters Used in the Examples

Table 1. Values of rate and initial population parameters used in the client/server example, Figures 1, 2, 3, 4, 5

$r_{data} = 0.6$	$r_{task} = 0.2$	$r_{reset} = 0.1$	$r_{on} = 0.2$	$r_{off} = 0.2$	$r_{heat} = 0.2$	$r_{cool} = 0.4$
$N_C = 40$	$N_S = 30$	$N_A = 20$	$T_{thresh} = 30$	$v = 1$	$c = 1$	

Table 2. Values of rate and initial population parameters used in the worked example, Figures 8, 9, 10. The constants $p_{B,\cdot}$ are set as the respective $p_{A,\cdot}$ constants multiplied by 1.7 and the heat constants $c_{\cdot,\cdot}$ are set as the corresponding $p_{\cdot,\cdot}$ constants multiplied by a conversion factor 7.71×10^{-6}.

$r_{q,1} = 0.2$	$r_{q,2} = 0.5$	$r_{s,1} = 0.2$	$r_{s,2} = 0.2$	$r_{reset} = 0.2$	$r_{wakeup} = 0.3$
$r_{on} = 0.2$	$r_{off} = 0.2$	$T_0 = 25$	$T_{thresh} = 20$	$T_{sleep} = 23$	
$p_{A,s} = 10$	$p_{A,sl} = 1$	$p_{A,1} = 30$	$p_{A,2} = 37.5$	$r_{cool} = 0.026$	
$N_C = 20000$	$N_S = 1000$	$N_A = 100$			

Lumping and Reversed Processes in Cooperating Automata

Simonetta Balsamo, Gian-Luca Dei Rossi, and Andrea Marin

Università Ca' Foscari di Venezia
Dipartimento di Scienze Ambientali, Informatica e Statistica
via Torino 155, Venezia
{balsamo,deirossi,marin}@dais.unive.it

Abstract. Performance evaluation of computer software or hardware architectures may rely on the analysis of a complex stochastic model whose specification is usually given in terms of a high level formalism such as queueing networks, stochastic Petri nets, stochastic Automata or Markovian process algebras. Compositionality is a key-feature of many of these formalisms and allows the modeller to combine several (simple) components to form a complex architecture. However, although these formalisms allow for relative compact specifications of possibly complex models, the derivation of the interested performance indices may be very time and space consuming since the set of possible states of the model tends to grow exponentially with the number of components.

In this paper we focus on models with underlying continuous time Markov chains and we show sufficient conditions under which exact lumping of the forward or the reversed process can be derived, allowing the exact computation of marginal stationary probabilities of the cooperating components. The peculiarity of our method relies on the fact that lumping is applied at component-level rather than to the CTMC of the joint process, thus reducing both the memory requirement and the computational cost of the subsequent solution of the model.

1 Introduction

System performance evaluation of computer software or hardware architectures can rely on the analysis of stochastic models that can provide prediction and comparison of design alternatives. Models are usually specified in terms of high level formalisms such as queueing networks (see e.g., [19]), stochastic Petri nets [20], stochastic Automata [21] or Markovian process algebra (see, e.g., the Performance Evaluation Process Algebra -PEPA- [16]). Performance evaluation and analysis of complex models can lead to the definition of large and complex stochastic performance models whose direct solution can become unfeasible (practically intractable) due to the computational complexity. Hence the analysis of complex systems often requires hierarchical modeling or the composition of sub-models. Compositionality is a key-feature of most of the performance formalisms and allows the modeller to combine several (possibly simple) components to form a complex architecture. However, although these formalisms allow

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 212–226, 2012.

for relative compact specifications of possibly complex models, the derivation of the interested performance indices may be very time and space consuming since the state space cardinality of the model tends to grow exponentially with the number of system components. When the model state space becomes too large, decomposition and aggregation techniques allow one to reduce the solution of a large problem to that of several much smaller ones.

The application of aggregation and lumpability techniques has been proposed to cope with large state space model solution, and it has been widely applied for the various formalisms, e.g. exact and approximate aggregation in queueing networks [1,4], decomposability and lumpablity for Markov chains [18,23], aggregation of stochastic Petri nets [2], stochastic Automata or Markovian process algebra [16,13], where the references should be considered just as examples of remarkable works in the corresponding field.

As concerns lumpability, under certain conditions the state space of a Markov chain can be partitioned into subsets of states, each of which can be seen as a single state of a smaller Markov chain. Such a chain is said to be *lumpable*. The process of lumping states in a Markov chain [18] defines a state space partition of the Markov chain and a corresponding new lumped process with a reduced state space. Specifically, consider a continuous-time homogeneous Markov chain (CTMC) with state space S with n states and transition rate matrix $\mathbf{Q}$. Let $\tilde{s}_1, \ldots, \tilde{s}_N$ be a partition of space S, where usually $N << n$. The CTMC is lumpable with respect to the partition if for any subset $\tilde{s}_i$ and states $s, s' \in \tilde{s}_i$, $\sum_{s'' \in \tilde{s}_k} \mathbf{Q}(s, s'') = \sum_{s'' \in \tilde{s}_k} \mathbf{Q}(s', s'')$ for $0 \leq k \leq N$. That is, for any two states in a given subset the cumulative transition rate to any other partition is equal. It is worthwhile to note that in [24] the authors give an algorithm that computes the optimal lumping of a Markov chain (i.e., that with lowest number of clusters) with a very efficient computational cost, $\mathcal{O}(m \log n)$ where m is the number of transitions. As concerns aggregation, several approaches to analyse complex systems consider hierarchical decomposition of the model into a set of submodels. Such a decomposition-aggregation approach defines three steps: 1) partition of the original model into a set of sub-models, and analysis of each submodel in isolation; 2) definition of a new and smaller aggregated model where each component represents an aggregated sub-model; 3) analysis of the aggregated model. Exact aggregation defines the new aggregated model equivalent to the original one, i.e., with the same solution for a set of performance indices, usually the aggregated stationary state distribution. Unfortunately, exact aggregation algorithms on the Markov chain have a computation complexity that is comparable to that of the solution of the entire model [7]. However, some exact aggregation methods have been defined directly in terms of model components at a higher level of abstraction formalism, and not at the Markov chain level. Moreover, under special constraints, conditions for exact aggregation have been defined for various classes of Markov models and for product-form models, such as product-form queueing networks [5,1]. Several approximate methods based on decomposition and aggregation, such as those described in [7,22,6,10], have also been proposed in the literature in the last decades. In this paper, we are

interested in studying those formalisms that allow the modeller to describe a system in a modular way. These formalisms can be widely applied in practice because they conform to good engineering principles. For instance PEPA or stochastic automata networks [21] represent important examples of formalisms that yield a high modularity. In [16,13] the authors present an exact technique to improve performance evaluation in PEPA models based on a relation called *strong equivalence*. The idea is to exploit the intrinsic modular nature of PEPA models. In [16] the author discusses the relations of the strong equivalence with the previous results on Markov chain lumping. Following this idea, in considering a model defined in terms of a set of cooperating components, we aim to apply the notion of lumping at the component level rather than at the CTMC level of the joint process. As in [16,13], we focus on stochastic performance models with underlying CTMCs. With respect to the cited papers we give a notion of lumpability which is more general and we present and prove two theorems on lumping in cooperating stochastic models both for the original model and for the time-reversed automata. Reversed processes have been known to be related to model decompositions especially in case of product-form models [17,14]. Here, we show that time-reversed processes can be used also in lumpability. In particular, we observe that a class of product-form models can be seen as a special case of the results we present.

The paper is organized as follows. Next section introduces some theoretical background and notation of the synchronization of the cooperating components. Section 3 presents two theorems that provide an efficient computation of the marginal steady-state probability distribution of the models obtained by exact lumping of both the forward and reversed automata. Finally, Section 4 presents some important final remarks.

2 Theoretical Background and Notation

In this paper we present the results in terms of cooperating automata -in a similar fashion of what is done in [4]- however, we restrict our analysis to pairwise cooperations (as, e.g., in [9]). We use bold letters to denote matrices and vectors (which must be considered row-vectors unless differently stated). $\mathbf{e}_n$ is the $n-$dimension vector whose components are all 1, $\mathbf{I}_n$ is the identity matrix of size $n \times n$. Sizes are omitted when they can be implicitly assumed. In what follows, we first introduce the semantics of the synchronisation between two components and then give the restrictions assumed in this paper.

Let us consider a pair of components M_1 and M_2 which synchronise on a set of transition types $\mathcal{T} = \{1, 2, \ldots, T\}$. The rate of a transition type is a positive real number λ_t, $t \in \mathcal{T}$. For each label $t \in \mathcal{T}$ we define two matrices $\mathbf{E}_{1t}$ and $\mathbf{E}_{2t}$ that describe the behaviour of component M_1 and M_2, respectively, with respect to synchronisation t and whose dimensions are $N_k \times N_k$ for $\mathbf{E}_{kt}$, with $k = 1, 2$ and N_k representing the number of states of component M_k. Matrix element $\mathbf{E}_{kt}(s, s')$ denotes the probability that automaton M_k moves from state s to state s' joint with a transition with the same type t performed by the other automaton; hence $1 \leq s, s' \leq N_j$, and $0 \leq \mathbf{E}_{kt}(s, s') \leq 1$. Moreover, the sum

$R_{kt}(s)$ of any row s of matrix $\mathbf{E}_{kt}$ is in the interval $[0,1]$ and can be interpreted as the probability that component M_k accepts to synchronise on t given that its actual state is s. The infinitesimal generator $\mathbf{Q}$ of the CTMC underlying the synchronisation of the two automata is defined as [23]:

$$\mathbf{Q} = \sum_{t=1}^{T} \lambda_t (\mathbf{E}_{1t} \otimes \mathbf{E}_{2t}) - \sum_{t=1}^{T} \lambda_t (\mathbf{D}_{1t} \otimes \mathbf{D}_{2t}), \tag{1}$$

where $\mathbf{D}_{kt} = \mathrm{diag}(\mathbf{E}_{kt}\mathbf{e}^\top)$, and $\mathrm{diag}(\mathbf{v})$ (with $\mathbf{v}$ a n-dimension row-vector) is defined as the $n \times n$ matrix:

$$\mathrm{diag}(\mathbf{v})(s,s') = \begin{cases} \mathbf{v}(s) & \text{if } s = s' \\ 0 & \text{otherwise} \end{cases},$$

and $\otimes$ denotes the Kronecker's product operator.

2.1 Feed-Forward Synchronisations

The main restriction we consider in this work concerns the class of synchronisations that we admit in our model.

Definition 1 (Non-blocking synchronisation). *We say that type $t \in \mathcal{T}$ is non-blocking if for at least one of the cooperating automata M_1 and M_2 it holds that $R_{kt}(s) = 1$ for all $s = 1, \ldots, N_k$. In this case we say that t is active in M_h, with $h \neq k$, and passive in M_k, $k, h \in \{1,2\}$.*

Informally, we can say that in a non-blocking synchronisation, one of the two cooperating automata (the active with respect to t) can carry out its activity of type t independently of the current state of the other automaton. As an instance, if we consider a tandem of two queues, and let t be the synchronisation between the customer departures from the first queue and the arrivals at the second, then a sufficient condition for the synchronisation to be non-blocking is that the second queue has infinite buffer size. The following definition is needed to avoid cycles among model synchronisation. In queueing theory this corresponds to the possibility of defining queueing networks with a feed-forward structure.

Definition 2 (Feed-forward synchronisation). *The model defined by the cooperation of M_1 and M_2 on transition types $\mathcal{T}$ is feed-forward if it is possible to identify a model M_k and M_h, $h \neq k$, $h, k \in \{1,2\}$ such that for all $t \in \mathcal{T}$ one of the following holds:*

1. *t is active in M_k and passive in M_h,*
2. *t is active in M_h and passive in M_k and $\mathbf{E}_{kt} = \mathbf{I}$.*

We call M_k and M_h the active and passive model, respectively.

Without loss of generality, we henceforth order the model labels such that in M_1 is active and M_2 is passive.

Remark 1. Observe that in a feed-forward synchronisation the infinitesimal generator underlying M_1 is well-defined by:

$$\mathbf{Q}_1 = \sum_{t=1}^{T} \lambda_t \mathbf{E}_{1t} - \sum_{t=1}^{T} \lambda_t \mathbf{D}_{1t}, \tag{2}$$

Hence, if $\mathbf{Q}_1$ is associated with an ergodic CTMC, we can compute the marginal distribution of M_1 in the cooperation. In fact, Condition 2 of Definition 2 states that if M_1 is passive with respect to a synchronisation type t, then it must not have a synchronised state-change in case the other automaton performs a transition of type t.

Note that if model M_1 is such that $\mathbf{E}_{1t_1} = \mathbf{E}_{1t_2} = \mathbf{I}$ with $t_1 \neq t_2$, then we can replace transition types t_1 and t_2 with a new type t^* for which $\mathbf{E}_{1t^*} = \mathbf{I}$ and $\mathbf{E}_{2t^*} = (\lambda_{t_1}\mathbf{E}_{2t_1} + \lambda_{t_2}\mathbf{E}_{2t_2})\lambda_{t^*}^{-1}$, where $\lambda_{t^*} = \max_s(\sum_{s'}(\lambda_{t_1}\mathbf{E}_{2t_1} + \lambda_{t_2}\mathbf{E}_{2t_2})(s, s'))$. As a consequence, in considering feed-forward synchronisations, we order the transition types such that: for $t = 1$, we have $\mathbf{E}_{21} = \mathbf{I}$, for $t = 2$ we have $\mathbf{E}_{12} = \mathbf{I}$ and for $2 < t \leq T$ we have that t is passive in M_2. Moreover, we write $q_2^2(s_2, s_2') = \lambda_2 \mathbf{E}_{22}(s_2, s_2')$, with $1 \leq s_2, s_2' \leq N_2$ and $q_1^t(s_1, s_1') = \lambda_t \mathbf{E}_{1t}(s_1, s_1')$ for $1 \leq t \leq N_1$, $t \neq 2$ and $1 \leq s_1, s_1' \leq N_1$.

3 Exact Computation of Marginal Distribution

In this section we prove two theorems that can be applied to define efficient algorithms for computing the marginal steady-state probability distribution for the passive model based on the exact lumping of the forward or reversed processes underlying the active component. Observe that, in this context, the concept of lumpability as introduced in [18] is extended in order to take into account the synchronising transition types. Roughly speaking, we aim to replace the active component M_1 by a smaller one denoted by $\tilde{M}_1$ such that the marginal distribution of M_2 in the cooperation $\tilde{M}_1 \otimes M_2$ is identical to that of M_2 in the cooperation $M_1 \otimes M_2$. In queueing theory, this idea has previously been applied for defining algorithms for approximate analysis of queueing networks (see, among others, [16,15,3] and the reference therein). However Theorem 1 and 2 give sufficient conditions for deriving exact lumping similarly to what is done in [16,13].

The following definition plays a pivotal role in what follows and extends the concept of lumpability in order to deal with synchronising transition types.

Definition 3 (Exact lumped automata). *Given active automaton M_1, a set of transition types $\mathcal{T}$, and a partition of the states of M_1 into $\tilde{N}_1$ clusters $\mathcal{S} = \{\tilde{1}, \tilde{2}, \ldots, \tilde{N}_1\}$, we say that $\mathcal{S}$ is an exact lumping for M_1 if:*

1. $\forall \tilde{s}_1, \tilde{s}_1' \in \mathcal{S}$, $\tilde{s}_1' \neq \tilde{s}_1$, $\forall s_1 \in \tilde{s}_1 \quad \varphi_1^1(s_1, \tilde{s}_1') = \tilde{q}_1^1(\tilde{s}_1, \tilde{s}_1')$
2. $\forall t > 2, \forall \tilde{s}_1, \tilde{s}_1' \in \mathcal{S}$, $\forall s_1 \in \tilde{s}_1 \quad \varphi_1^t(s_1, \tilde{s}_1') = \tilde{q}_1^t(\tilde{s}_1, \tilde{s}_1')$,

where $\varphi_1^t(s_1, \tilde{s}_1') = \sum_{s_1' \in \tilde{s}_1'} q_1^t(s_1, s_1')$. *If* M_1 *is lumpable with respect to* $\mathcal{S}$, *we define the automaton* $\tilde{M}_1$ *with* $\tilde{N}_1$ *states as follows:*

$$\tilde{\mathbf{E}}_{11}(\tilde{s}_1, \tilde{s}_1') = \begin{cases} \tilde{q}_1^1(\tilde{s}_1, \tilde{s}_1')\tilde{\lambda}_1^{-1} & \text{if } \tilde{s}_1 \neq \tilde{s}_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\tilde{\mathbf{E}}_{12} = \mathbf{I}, \quad \tilde{\mathbf{E}}_{1t}(\tilde{s}_1, \tilde{s}_1') = \tilde{q}_1^t(\tilde{s}_1, \tilde{s}_1')\tilde{\lambda}_t^{-1} \; t > 2$$

where:

$$\tilde{\lambda}_t = \max_{\tilde{s}_1 = 1, \ldots, \tilde{N}_1} \left(\sum_{\tilde{s}_1'=1}^{\tilde{N}_1} \tilde{q}_1^t(\tilde{s}_1, \tilde{s}_1') \right) \text{ for } t \neq 2, \tilde{\lambda}_2 = \lambda_2$$

are the rates associated with the transition types in the cooperation between $\tilde{M}_1$ *and* M_2.

Remark 2 (Exact lumping and strong equivalence). The reader familiar with process algebra can observe that Definition 3 is closely related to the definition of *strong equivalence* between PEPA processes given in [16]. The author exploits this approach with the same aims that we have here. The difference between Definition 3 and the concept of *strong equivalence* concerns the conditions about the non-synchronising transitions, i.e., those that in PEPA are called τ-actions (in our framework these correspond to transitions with type $t = 1$). In fact, Definition 3 distinguishes between non-synchronising ($t = 1$) and synchronising ($t > 2$) transition types, as the former does not need to have constant outgoing rate from a state of a cluster to other states of the same cluster. Therefore, we can say that strong equivalence implies lumping in the sense of Definition 3 but not vice versa, as illustrated in Example 2.

As one may expect, if $\tilde{M}_1$ is an exact lumped automaton of M_1, then the CTMC underlying $\tilde{M}_1$ is an exact lumping of that of M_1 in the standard sense of [18]. Proposition 1 trivially follows from Definition 3.

Proposition 1. *Given active model* M_1, *if model* $\tilde{M}_1$ *is a lumping of* M_1 *with respect to partition* $\mathcal{S} = \{\tilde{1}, \ldots, \tilde{N}_1\}$, *then if two states* s_1 *and* s_1' *belong to the same cluster, then we have* $\phi_1^t(s_1) = \phi_1^t(s_1')$, *for all* $t > 2$, *where:*

$$\phi_1^t(s_1) = \sum_{s_1'=1}^{N_1} q_1^t(s_1, s_1').$$

In what follows we assume that M_1, M_2 and their cooperation to have ergodic underlying CTMCs.

Theorem 1. *Given the model* $M_1 \otimes M_2$, *let* $\tilde{M}_1$ *be an exact lumping of* M_1 *whose clusters are* $\mathcal{S} = \{\tilde{1}, \ldots, \tilde{N}_1\}$. *Then, the marginal steady-state distribution* π_2 *of* M_2 *is given by:*

$$\forall s_2 = 1, \ldots, N_2, \quad \pi_2(s_2) = \sum_{s_1=1}^{N_1} \pi(s_1, s_2) = \sum_{\tilde{s}_1=1}^{\tilde{N}_1} \pi^*(\tilde{s}_1, s_2), \tag{3}$$

where π *is the steady-state distribution of the cooperation between* M_1 *and* M_2 *and* π^* *that of the cooperation between* $\tilde{M}_1$ *and* M_2.

Theorem 1 appears in [16] considering the notion of strong-equivalence between PEPA agents. The following proof is needed since Definition 3 is slightly different as pointed out by Remark 2 and because M_2 allows for probabilistic synchronisations.

Proof of Theorem 1. The proof is based on verifying the global balance equations (GBEs) of model M_2 when it cooperates with $\tilde{M}_1$, where the transition rates are conditioned on the state of $\tilde{M}_1$ (see, e.g., [8]). Their solution gives π_2^*, i.e., the marginal steady-state distribution of M_2 when it cooperates with $\tilde{M}_1$. Then, we show that π_2^* satisfies also the GBEs derived by the analysis of M_2 cooperating with M_1 and hence, by the uniqueness of the stationary distribution we conclude the theorem, i.e., $\pi_2 = \pi_2^*$. Let us consider a generic state $(\tilde{s}_1, s_2)$, $\tilde{s}_1 = 1, \ldots, \tilde{N}_1$ and $s_2 = 1, \ldots, N_2$. Recall that M_1 and the cooperation between M_1 and M_2 are ergodic, hence trivially also the cooperation between $\tilde{M}_1$ and M_2 is ergodic as well as $\tilde{M}_1$ itself. Let $\tilde{\pi}_1$ be the steady-state distribution of $\tilde{M}_1$. The following GBE can be written as:

$$\pi_2^*(s_2)\left(\sum_{s_2'=1}^{N_2} q_2^2(s_2,s_2') + \sum_{t=3}^{T}\sum_{\tilde{s}_1=1}^{\tilde{N}_1}\sum_{\tilde{s}_1'=1}^{\tilde{N}_1} \tilde{q}_1^t(\tilde{s}_1,\tilde{s}_1')\tilde{\pi}_1(\tilde{s}_1)\right)$$
$$= \sum_{s_2'=1}^{N_2} \pi_2^*(s_2')\left(q_2^2(s_2',s_2) + \sum_{t=3}^{T}\sum_{\tilde{s}_1=1}^{\tilde{N}_1}\sum_{\tilde{s}_1'=1}^{\tilde{N}_1} \tilde{q}_1^t(\tilde{s}_1',\tilde{s}_1)\tilde{\pi}_1(\tilde{s}_1')\mathbf{E}_{2t}(s_2',s_2)\right)$$

We now apply the conditions given in Definition 3. In particular, observe that for $t > 2$, $\sum_{s_1' \in \tilde{s}_1'} q_1^t(s_1, s_1')$ is independent of the choice of $s_1 \in \tilde{s}_1$ and hence we can write that:

$$\left(\sum_{x \in \tilde{s}_1} \pi_1(x)\right)\left(\sum_{s_1' \in \tilde{s}_1'} q_1^t(s_1,s_1')\right) = \sum_{s_1 \in \tilde{s}_1}\sum_{s_1' \in \tilde{s}_1'} \pi_1(s_1) q_1^t(s_1,s_1'),$$

where π_1 is the marginal distribution of M_1. The CTMC underlying M_1 is lumpable with respect to partition $\mathcal{S}$ and hence $\forall \tilde{s}_1 = 1, \ldots, \tilde{N}_1$ $\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1) = \tilde{\pi}_1(\tilde{s}_1)$:

$$\pi_2^*(s_2)\left(\sum_{s_2'=1}^{N_2} q_2^2(s_2,s_2') + \sum_{t=3}^{T}\sum_{\tilde{s}_1=1}^{\tilde{N}_1}\sum_{\tilde{s}_1'=1}^{\tilde{N}_1}\Big(\sum_{s_1 \in \tilde{s}_1}\sum_{s_1' \in \tilde{s}_1'} \pi_1(s_1) q_1^t(s_1,s_1')\Big)\right)$$
$$= \sum_{s_2'=1}^{N_2} \pi_2^*(s_2')\left(q_2^2(s_2',s_2) + \sum_{t=3}^{T}\sum_{\tilde{s}_1=1}^{\tilde{N}_1}\sum_{\tilde{s}_1'=1}^{\tilde{N}_1}\Big(\sum_{s_1' \in \tilde{s}_1'}\sum_{s_1 \in \tilde{s}_1} \pi_1(s_1') q_1^t(s_1',s_1)\Big)\mathbf{E}_{2t}(s_2',s_2)\right)$$

Finally, we rewrite the indices of the sums and obtain:

$$\pi_2^*(s_2)\left(\sum_{s_2'=1}^{N_2} q_2^2(s_2,s_2') + \sum_{t=3}^{T}\sum_{s_1=1}^{N_1}\sum_{s_1'=1}^{N_1} \pi_1(s_1) q_1^t(s_1,s_1')\right)$$
$$= \sum_{s_2'=1}^{N_2} \pi_2^*(s_2')\left(q_2^2(s_2',s_2) + \sum_{t=3}^{T}\mathbf{E}_{2t}(s_2',s_2)\sum_{s_1=1}^{N_1}\sum_{s_1'=1}^{N_1} \pi_1(s_1') q_1^t(s_1',s_1)\Big)\right), \quad (4)$$

that is the GBE for state s_2 derived from the cooperation between M_1 and M_2 considering the conditional transition rates. Therefore, by uniqueness of the steady-state distribution, we conclude $\pi_2(s_2) = \pi_2^*(s_2)$ for $1 \leq s_2 \leq N_2$. □

Timed-Reversed Automata. Theorem 2 relies on the theory of reversed Markov processes as studied in [17] and successively in [14]. Before stating the second theorem, we briefly review some results about reversed Markov processes [17]. Given a continuous time stochastic process $X(t)$ we say that it is *stationary* if $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same joint-distribution of $(X(t_1 + \tau), X(t_2 + \tau), \ldots X(t_n + \tau))$ and we say that it is *reversible* if $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same joint-distribution of $(X(\tau - t_1), X(\tau - t_2), \ldots, X(\tau - t_n))$ for all $t_1, \ldots, t_n, \tau \in \mathbb{R}$ and $n \in \mathbf{N}$. It is easy to prove that a reversible process is also stationary. For a reversible CTMC, the following relation holds:

$$\pi_1(s_1)q_1(s_1, s_1') = \pi_1(s_1')q_1(s_1', s_1),$$

where $\pi_1(s_1)$ is the stationary probability of s_1 and s_1, s_1' two arbitrary states of the CTMC, and $q_1(s_1, s_1')$ denotes the transition rate from s_1 to s_1'. Obviously, a stationary process may be not reversible. In this case, it is still possible to define the reversed process, but the joint-distribution of $(X(t_1), X(t_2), \ldots, X(t_n))$ is in general different form that of process $(X(\tau - t_1), X(\tau - t_2), \ldots, X(\tau - t_n))$. Assume that the forward chain admits the steady-state distribution π_1 and has a transition from state s_1 to s_1' with rate $q(s_1, s_1')$, then it can be proved that the reversed process is still a Markov process with the same state space that has a transition from s_1' to s_1 whose rate $q_1^R(s_1', s_1)$ is given by:

$$q_1^R(s_1', s_1) = \frac{\pi(s_1)}{\pi(s_1')} q_1(s_1, s_1'). \tag{5}$$

Observe that from the transition rates of the reversed CTMC we can efficiently compute the unnormalised steady-state distribution and vice versa. Based on Equation (5) we give the following definition:

Definition 4 (Timed-reversed automata). *Given the active automaton M_1 synchronising on transition type $\mathcal{T}$ with rates $\lambda_1, \ldots, \lambda_T$, we define the timed-reversed automaton M_1^R as follows:*

$$\mathbf{E}_{1t}^R(s_1, s_1') = \frac{\pi_1(s_1')}{\pi_1(s_1)} q_1^t(s_1', s_1) \frac{1}{\lambda_t^R} \quad t \neq 2$$

$$\mathbf{E}_{12}^R = \mathbf{I}$$

where:

$$\lambda_t^R = \max_{s_1=1,\ldots,N_1} \left(\sum_{s_1'=1}^{N_1} q_1^{tR}(s_1, s_1') \right)$$

and $q_1^{tR}(s_1, s_1') = (\pi_1(s_1')/\pi_1(s_1))q_1^t(s_1', s_1)$, for all $1 \leq s_1, s_1' \leq N_1$.

Theorem 2. *Given the model $M_1 \otimes M_2$, let M_1^R be the reversed automaton of M_1 and let $\tilde{M}_1^R$ be an exact lumping of M_1^R whose clusters are $\mathcal{S} = \{\tilde{1}, \ldots, \tilde{N}_1\}$. Then, the marginal steady-state distribution π_2 of M_2 is given by:*

$$\forall s_2 = 1, \ldots, N_2, \pi_2(s_2) = \sum_{s_1=1}^{N_1} \pi(s_1, s_2) = \sum_{\tilde{s}_1=1}^{\tilde{N}_1} \tilde{\pi}^R(\tilde{s}_1, s_2), \tag{6}$$

where π is the steady-state distribution of the cooperation between M_1 and M_2 and $\tilde{\pi}^R$ that of the cooperation between $\tilde{M}_1^R$ and M_2.

Proof of Theorem 2. The first part of the proof resembles that of Theorem 1. Here we rewrite Equation (4) for this case:

$$\pi_2^*(s_2)\left(\sum_{s_2'=1}^{N_2} q_2^2(s_2,s_2') + \sum_{t=3}^{T}\sum_{s_1=1}^{N_1}\sum_{s_1'=1}^{N_1} \pi_1(s_1) q_1^{tR}(s_1,s_1')\right)$$
$$= \sum_{s_2'=1}^{N_2} \pi_2^*(s_2')\left(q_2^2(s_2',s_2) + \sum_{t=3}^{T} \mathbf{E}_{2t}(s_2',s_2) \sum_{s_1=1}^{N_1}\sum_{s_1'=1}^{N_1} \pi_1(s_1') q_1^{tR}(s_1',s_1)\right)\Bigg).$$

Applying Definition 4 we obtain:

$$\pi_2^*(s_2)\left(\sum_{s_2'=1}^{N_2} q_2^2(s_2,s_2') + \sum_{t=3}^{T}\sum_{s_1=1}^{N_1}\sum_{s_1'=1}^{N_1} \pi_1(s_1) q_1^{t}(s_1',s_1)\frac{\pi_1(s_1')}{\pi_1(s_1)}\right)$$
$$= \sum_{s_2'=1}^{N_2} \pi_2^*(s_2')\left(q_2^2(s_2',s_2) + \sum_{t=3}^{T} \mathbf{E}_{2t}(s_2',s_2) \sum_{s_1=1}^{N_1}\sum_{s_1'=1}^{N_1} \pi_1(s_1') q_1^{t}(s_1,s_1')\frac{\pi_1(s_1)}{\pi_1(s_1')}\right)\Bigg),$$

that can be trivially simplified to observe that it is the GBE corresponding to state s_2 of M_2 cooperating with M_1. Hence, by uniqueness of the steady-state probability we conclude that $\pi_2(s_2) = \pi_2^*(s_2)$ for $1 \le s_2 \le N_2$. □

3.1 Running Examples

Before discussing some implications of Theorem 1 and 2 we apply them to some examples.

Example 1 (Toy Example). We consider the automaton depicted by Figure 1 in which M_1 has 4 states and M_2 has 2 states. Synchronisation types are $T = 3$. The figure depicts the behaviour of the automata in which the arcs are labelled by the transition type t and the rate $q_i^t(s_i, s_i')$. For the sake of simplicity, transitions with type 1 and 2 are omitted in automaton M_2 and M_1, respectively. Observe that, according to Proposition 1 a necessary (but not sufficient) condition for M_1 to be lumpable with respect to a partition is that the sum of the rates of type 3 transitions outgoing from states belonging to the same cluster should be the same. In this case we have: $\phi_1^3(1) = 0$. $\phi_1^3(2) = 7/12\lambda$, $\phi_1^3(3) = 3\gamma - 5/12\lambda$, $\phi_1^3(4) = \lambda/2 + 2\mu$. Therefore, we immediately conclude that Theorem 1 is applicable only to the trivial case of $\tilde{M}_1 = M_1$. However, we can still try to apply Theorem 2, but we need to compute the time-reversed automaton of M_1 according to Definition 4. M_1^R is depicted in Figure 2. Let $\phi_1^{tR}(s_1) = \sum_{s_1'} q_1^{tR}(s_1, s_1')$, then we have: $\phi_1^{3R}(1) = \phi_4^{3R}(4) = \lambda + \mu$ and $\phi_1^{3R}(2) = \phi_1^{3R}(3) = \gamma$. This *suggests* the possible lumping $\tilde{1} = \{1, 4\}$ and $\tilde{2} = \{2, 3\}$ that can be easily shown to be exact according to Definition 3. Therefore, we can derive the lumped automaton $\tilde{M}_1^R$ of Figure 3. It is worthwhile to point out that it is *not* the case that the

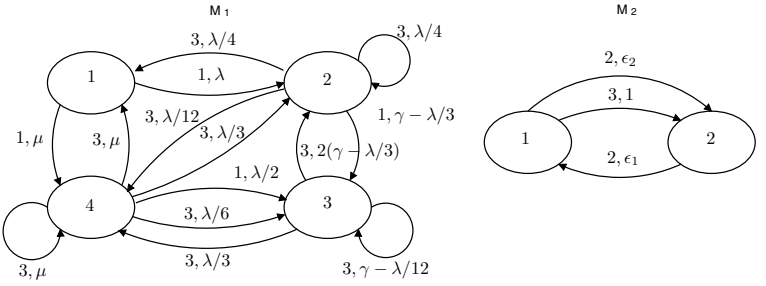


Fig. 1. Example of cooperation between two automata

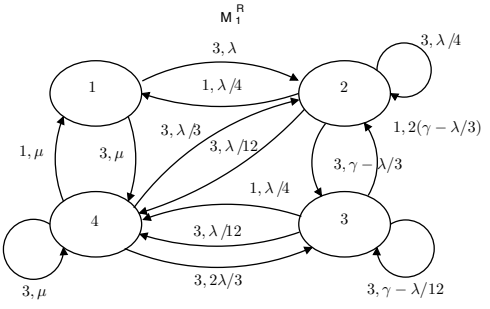


Fig. 2. Timed-reversed automaton of M_1 depicted by Figure 1

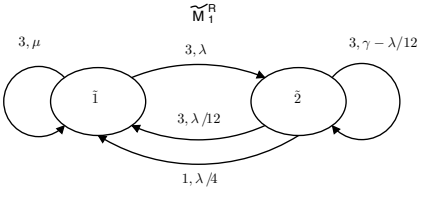


Fig. 3. Lumping of automaton M_1^R depicted in Figure 2

steady-state distribution of $M_1 \otimes M_2$ is identical to that of $M_1^R \otimes M_2$, but we can only compute the marginal distribution of M_1 (by the solution of its underlying CTMC) and of M_2 by solving $\tilde{M}_1^R \otimes M_2$ and aggregating the state probabilities as specified by Theorem 2. Finally, we spot that $\phi_1^{1R}(1) = 0$, while $\phi_1^{1R}(4) = \mu$ even if 1 and 4 belongs to the same cluster $\tilde{1}$. This can happen because the the transition of type 1 outgoing from 4 is directed to a state of the same cluster, i.e., 1.

Example 2 (Producer-Consumer). Consider two identical software processes sharing a common memory buffer. Each of those processes can alternate between

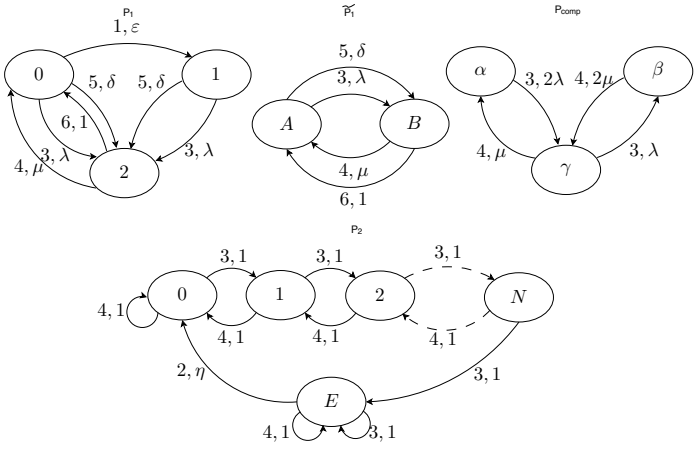


Fig. 4. A model for a producer-consumer process and the relative buffer

producing data (with rate λ) which is written in the buffer and consuming them with rate μ. Sometimes the process, while producing data (state 0), switches to its internal state 1 with rate ϵ, however this transition does not stop the production. If, for some reason, the process cannot produce data, it moves back to the state 2 where it can consume data in the buffer with rate δ, forcing the other process to skip its consuming phase and to return to the producing one. Each of those processes can be modelled as in automaton P_1 of Figure 4. In the same figure the automaton P_2 models the shared buffer, which has N memory slots. Whenever there is an overflow, the buffer enters in an error state E until it is emptied within an exponentially-distributed time with parameter η. We are interested in computing the marginal steady state distribution of P_2, e.g., to analyse the frequency of buffer overflows given a set of parameters. According to the conditions of Definition 3, P_1 allows for the lumping shown in $\tilde{P}_1$. Notice that this lumping doesn't respect the strong equivalence relation as defined in [16]. Moreover, when we combine a process P_1' that behaves as P_1 with another one $P_1^{''}$ identically modelled, imposing that whenever a transition labelled with 5 happens in one process a transition labelled with 6 is forced on the other one, we are able to further aggregate the joint process, as shown in the lumping P_{comp}. Transitions of type 5 and 6 do not appear in the lumped process, because joint transitions of those types are not synchronised with further processes. The cardinality of the joint state space among the two instances of P_1 and the buffer model P_2 can be thus reduced from $3 \times 3 \times (N+2)$ to $3 \times (N+2)$. Notice that this aggregation cannot be found using state-of-the-art techniques such as in [13], since we exploit Definition 3 to determine sufficient conditions for aggregation and Theorem 1 to compute marginal steady state distributions.

3.2 Theoretical Considerations about Theorem 1 and 2

In this section we have presented two theorems about lumping in cooperating stochastic models. We already pointed out the connections between Theorem 1 and the notion of strong equivalence (and its consequences) presented in [16]. Now, we compare Theorem 2 with other relatively recent results that appeared in literature, in particular for what concerns the analysis of product-form stochastic models. A model is in product-form if the joint stationary distribution of an ergodic joint state can be expressed in terms of the product of the marginal distributions of its components considered in isolation and opportunely parametrised. A very general theory about product-form models has been developed in [14] where the author, based on process algebra analysis, gives sufficient conditions for the cooperation of two models to be in product-form (Reversed Compound Agent Theorem -RCAT-). If we reformulate those conditions in terms on cooperating stochastic automata, we have:

C1 If transition $t > 2$ is passive with respect to M_k, then each state of M_k has exactly one outgoing transition of type t (and its weight is 1);
C2 If transition $t > 2$ is active with respect to M_k, then each state of M_k has exactly one incoming transition of type t;
C3 Let $t > 2$ be an active type with respect to M_k, then the reversed rate associated with each transition of type t in M_k is the same.

Observe that Conditions C2 and C3 imply that the timed-reversed automaton M_1^R associated with M_1 admits a lumping of one single state as illustrated by the following example. From this we observe that when Theorem 2 is applied and an automaton can be lumped into a single state, then we can also say that the joint steady-state probability is given by the product of the marginal distributions of the single automata.

Example 3 (G-network analysis). G-networks [11] are very powerful and versatile class of models developed in queueing theory and they can be efficiently studied because they yield a product-form stationary distribution. Let us consider the model of Figure 5 that consists of two G-queues. Customers arrive at $Q1$ and $Q2$ according to independent Poisson processes with rates λ_1 and λ_2, respectively. Service time is exponentially distributed with mean μ_1^{-1} and μ_2^{-1} for $Q1$ and $Q2$, respectively. At a service completion epoch at $Q1$, the customer can move to $Q2$ as an ordinary customer with probability p_{12}^+, while it can enter $Q2$ as a negative customer with probability p_{12}^-. The effect of a negative customer arrival at $Q2$ is to delete a positive one if the queue is non-empty (otherwise the negative customer simply vanishes). In [14] it is shown that the reversed rates of the transitions with types 3 and 4 in $Q1$ are constant and equal to $\lambda_1 p_{12}^+$ and $\lambda_1 p_{12}^-$, respectively. According to RCAT we can obtain the marginal distribution of $Q2$ by setting the rates of transitions with type 3 to $\lambda_1 p_{12}^+$ and those of transitions with type 4 to $\lambda_1 p_{12}^-$. Analogously, the application of Theorem 2 leads to a lumping of M_1^R consisting of one single state with two self-loops: one with type 3 and rate $\lambda_1 p_{12}^+$ and the other with type 4 and rate $\lambda_1 p_{12}^-$.

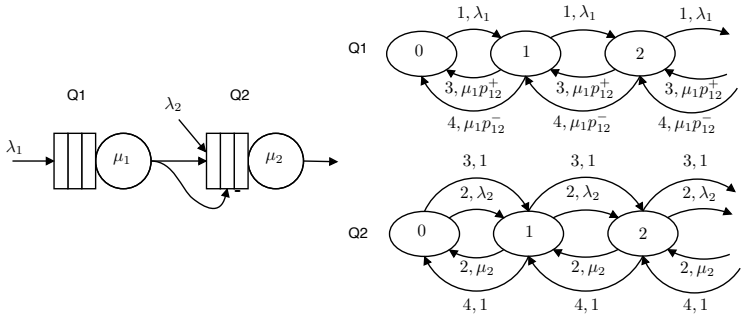


Fig. 5. Graphical representation of a G-network and the corresponding model using automata

Does the product-form property yield in case of a lumping to a single state applying Theorem 1? The answer is negative as shown by the following counterexample.

Example 4. This example is intentionally very simple in order to spot the difference between performing a lumping in the forward process or in the reversed. We consider two exponential queues with synchronised arrivals modelled by a Poisson process with rate λ as shown in Figure 6. Service rates are μ_1 and μ_2. In this case we can apply Theorem 1, and queue $Q1$ is lumped into a model

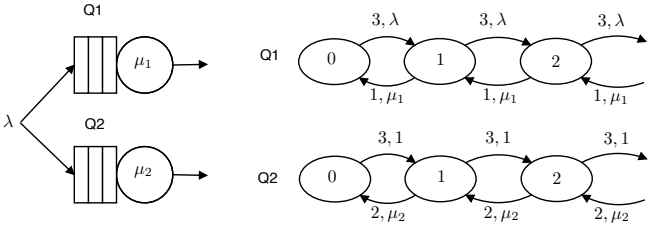


Fig. 6. Exponential queues with synchronised arrivals and their representation by stochastic automata

$\tilde{M}_1$ with a single state with one self-loop with type 3 and rate λ. Observe that although the marginal distribution obtained for Q_2 by the analysis of $\tilde{M}_1 \otimes Q_2$ is trivially correct, the model is not in product-form since the stationary probability is different from the product of the marginal distributions of Q_1 and Q_2 (although they can be derived without the derivation of the joint state space).

3.3 Practical Implementation of Exact Lumping

The problem of lumping Markov chains, PEPA models, queueing networks and stochastic Petri nets has been widely addressed. In this context we mention the

important algorithm based on a notion of isomorphism among PEPA components presented in [13]. The isomorphism relation is stricter than the strong equivalence (which is itself stricter than the conditions of Definition 3) and hence, as the same authors point out, the resulting lumping may be not optimal. However, the algorithm is very efficient, already implemented in the PEPA Workbench [12] and the results it provides may be straightforwardly used to apply Theorem 1. The same algorithm may still be applied once the time-reversed automaton has been derived and hence apply Theorem 2. The problem of time-reversing PEPA agents has been addressed in [14].

4 Final Remarks

In this paper we have proposed some theoretical results concerning the lumpability of cooperating stochastic automata, allowing to reduce the computational cost of the derivation of the marginal distributions of two cooperating Markovian agents. We use a notion of lumpability that is similar, but different, to that proposed in [16] where the definition of the *strong equivalence* is applied with the same purposes that we have here. In particular, the author aims to reduce the cardinality of the joint state space of the cooperation of two stochastic models by replacing one component by a simpler (but strongly equivalent) one. Here, we show that the notion of lumpabiliy given by Definition 3 is different in terms of how the non-synchronising transitions are treated. From a theoretical point of view, we proved two theorems about lumping. In particular, to the best of our knowledge, the connections between the lumping of the reversed process of a stochastic automaton and the properties enjoyed by the joint process have been not explored before. A remarkable exception is given by the product-form theorem presented in [14] whose conditions could be reformulated in terms of lumping of a timed reversed automaton: RCAT conditions are satisfied if the cooperating processes admit a lumping of the time-reversed automata with a single state. As concerns future directions of research, our aim is to extend Theorems 1 and 2 in order to overcome their limitations on the class of automata on which those results can be applied. Moreover, we are developing a methodology for approximating the marginal distributions in the cooperations of two stochastic automata using results derived from the aforementioned theorems whenever an exact lumping with a desired state space cardinality cannot be found.

References

1. Balsamo, S., Iazeolla, G.: An extension of Norton's theorem for queueing networks. IEEE Trans. on Software Eng. SE-8, 298–305 (1982)
2. Buchholz, P.: Adaptive decomposition and approximation for the analysis of stochastic Petri nets. Perf. Eval. 56, 23–52 (2004)
3. Buchholz, P.: Bounding stationary results of tandem networks with MAP input and MAP service time distributions. In: Proc. of ACM SIGMETRICS/PERFORMANCE, Saint Malo, FR, pp. 191–202 (2006)

4. Buchholz, P.: Product form approximations for communicating Markov processes. Perf. Eval. 67(9), 797–815 (2010), special Issue: QEST 2008
5. Chandy, K.M., Herzog, U., Woo, L.: Parametric analysis of queueing networks. IBM Journal of Res. and Dev. 1(1), 36–42 (1975)
6. Courtois, P.J., Semal, P.: Bounds for the positive eigenvectors of nonnegative matrices and for their approximations by decomposition. J. of the ACM 31(4), 804–825 (1984)
7. Courtois, P.: Decomposability. Academic Press, New York (1977)
8. Economou, A.: Generalized product-form stationary distributions for Markov chains in random environment with queueing application. Adv. in Appl. Prob. 37, 185–211 (2005)
9. Fourneau, J.M., Plateau, B., Stewart, W.J.: Product form for stochastic automata networks. In: ValueTools 2007: Proc. of the 2nd International Conference on Performance Evaluation Methodologies and Tools, pp. 1–10. ICST, Brussels (2007)
10. Franceschinis, G., Muntz, R.R.: Bounds for quasi-lumpable markov chains. Elsevier Perf. Eval. 20(1-3), 223–243 (1994)
11. Gelenbe, E.: Product form networks with negative and positive customers. J. of Appl. Prob. 28(3), 656–663 (1991)
12. Gilmore, S., Hillston, J.: The PEPA Workbench: A Tool to Support a Process Algebra Based Approach to Performance Modelling. In: Haring, G., Kotsis, G. (eds.) TOOLS 1994. LNCS, vol. 794, pp. 353–368. Springer, Heidelberg (1994)
13. Gilmore, S., Hillston, J., Ribaudo, M.: An Efficient Algorithm for Aggregating PEPA Models. IEEE Trans. on Software Eng. 27(5), 449–464 (2001)
14. Harrison, P.G.: Turning back time in Markovian process algebra. Theoretical Computer Science 290(3), 1947–1986 (2003)
15. Heindl, A.: Decomposition of general queueing networks with mmpp inputs and customer losses. Perf. Eval. 51(1-4), 117–136 (2003)
16. Hillston, J.: A Compositional Approach to Performance Modelling. Ph.D. thesis, Department of Computer Science, University of Edinburgh (1994)
17. Kelly, F.: Reversibility and stochastic networks. Wiley, New York (1979)
18. Kemeny, J.G., Snell, J.L.: Finite Markov Chains, ch. II. D. Van Nostrand Company, Inc. (1960)
19. Lavenberg, S.S.: Computer Performance Modeling Handbook. Academic Press, New York (1983)
20. Molloy, M.K.: Performance analysis using stochastic petri nets. IEEE Trans. on Comput. 31(9), 913–917 (1982)
21. Plateau, B.: On the stochastic structure of parallelism and synchronization models for distributed algorithms. SIGMETRICS Perform. Eval. Rev. 13(2), 147–154 (1985)
22. Stewart, G.: Computable error bounds for aggregated Markov chains. J. of the ACM 30(2), 271–285 (1983)
23. Stewart, W.J.: Introduction to the Numerical Solution of Markov Chains. Princeton University Press, UK (1994)
24. Valmari, A., Franceschinis, G.: Simple $O(m \log n)$ Time Markov Chain Lumping. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 38–52. Springer, Heidelberg (2010)

Transform-Domain Solutions of Poisson's Equation with Applications to the Asymptotic Variance

Koen De Turck, Sofian De Clercq, Sabine Wittevrongel,
Herwig Bruneel, and Dieter Fiems

Department of Telecommunications and Information Processing,
Ghent University
Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
{kdeturck,sdclercq,sw,hb,df}@telin.UGent.be

Abstract. Recent years have seen a considerable increase of attention devoted to Poisson's equation for Markov chains, which now has attained a central place in Markov chain theory, due to the extensive list of areas where Poisson's equation pops up: perturbation analysis, Markov decision processes, limit theorems of Markov chains, etc. all find natural expression when viewed from the vantage point of Poisson's equation.

We describe how the use of generating functions helps solve Poisson's equation for different types of structured Markov chains and for driving functions, and point out some applications. In particular, we solve Poisson's equation in the transform domain for skip-free Markov chains and Markov chains with linear displacement. Closed-form solutions are obtained for a class of driving functions encompassing polynomial functions and functions with finite support.

1 Introduction

In this paper, we investigate the use of transform-domain techniques to solve Poisson's equations associated with various classes of Markov chains. Poisson's equation (PE) has in recent years been acknowledged to be one of the central concepts of Markov chains, almost equal in importance to the steady-state equation. In order to advance the exposition, we introduce some notation. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $\{X_n\}, n \in \mathbb{Z}$, be a time-homogeneous Markov chain, taking values in a countable state space $\mathcal{X}$ (this is not a strict necessity, but as our example Markov chains will all have a countable state space, we have opted to avoid the technicalities of dealing with more general state spaces). The transition probabilities $p_{ij} = \mathbb{P}[X_{n+1} = j | X_n = i], i, j \in \mathcal{X}$ are recorded in a (possibly countably infinite) transition matrix P.

Two linear systems of equations are commonly associated with such a Markov chain. The first is the *invariance equation* that says that the stationary distribution recorded in row vector $\boldsymbol{\pi} = [\pi_i]_{i \in \mathcal{X}}$, provided it exists, satisfies:

$$\pi_j = \sum_{i \in \mathcal{X}} \pi_i p_{ij} \quad \text{, or in matrix notation: } \boldsymbol{\pi} = \boldsymbol{\pi} P. \tag{1}$$

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 227–239, 2012.

The importance of this equation is such, that the expression 'solving a Markov chain' is colloquially taken to mean almost unambiguously solving equation 1.

The second equation, *Poisson's equation* (PE) – perhaps somewhat surprisingly – has only quite recently been put on a comparable level of importance. Its formulation requires a function on the state space $f : \mathcal{X} \to \mathbb{R}$, which is depending on the application referred to as the driving function, cost function or reward function. Given this function f, we aim to assign a value h_i to each state i of the chain, which records the relative long-term cost of starting from state i as compared to others. Indeed, Poisson's equation is equal to:

$$h_i = f_i - \sum_{j \in \mathcal{X}} \pi_j f_j + \sum_{j \in \mathcal{X}} p_{ij} h_j. \tag{2}$$

In words, the relative value of state i is equal to the cost of state i, minus the average cost in stationarity, plus the expected relative value of the Markov chain one step in the future. Readers may notice the similarity with Bellman's equation [3] or with dynamic programming equations in general. Note that the term Poisson's equation is not native to probability theory, but hails from the theory of partial differential equations and was chosen due to certain similarities between the two. A slight rearrangement of terms results in the following perhaps more familiar form:

$$(I - P)\mathbf{h} = (I - \mathbf{1}\boldsymbol{\pi})\mathbf{f}, \tag{3}$$

where I denotes the identity matrix of appropriate dimension, $\mathbf{1}$ is a column vector with element 1 on each row and $\mathbf{f}$ and $\mathbf{h}$ are the value and cost column vectors respectively.

Solutions for the PE typically proceed from the fundamental kernel or deviation matrix which will be introduced in the Sec. 2.

A variant equation that has shown to be relevant for applications is the *discounted Poisson's equation*:

$$h_i - \gamma \sum_j p_{ij} h_j = f_i, \tag{4}$$

or in matrix notation:

$$(I - \gamma P)\mathbf{h} = \mathbf{f}. \tag{5}$$

where γ is referred to as the discount factor. The idea behind this equation is that costs further in the future should have a smaller impact on the value than short-term costs. In this paper, we will concentrate on the original Poisson equation.

Neveu [9] seems to have been the first to coin the term Poisson's equation in a Markov context. Over the years, many new results were discovered, both from a practitioner's point of view, as more applications were discovered, such as perturbation [5], limit theorems [10], constructing approximative models [7], as well as from a theoretical point of view, which has resulted in a better, and perhaps almost complete understanding as to when solutions of PE exist. A key concept in this regard is V-uniform ergodicity [6], which provides sufficient

conditions that seem close to necessary. Its importance was first noticed by Hordijk and Spieksma [14], and taken to full generality by Meyn and Tweedie [6].

It is to be expected that the importance of Poisson's equation will further increase, as control of queueing systems gets a more pronounced place next to mere performance analysis. In this paper, we tackle the Poisson's equation with transform-domain tools, which are since long popular in applied probability and in queueing theory in particular [1,2]. Related work includes the results of Koole [13], who derived closed-form expressions for the deviation matrix of birth-and-death-processes. Related work also includes [17,18], which uses the matrix-analytic paradigm but, in contrast to this paper, is restricted to finite state spaces.

The outline of the rest of this paper is as follows. In Sec. 2 we outline the general recipe and work out in detail a solution for the PE for some often encountered types of Markov chains, and in Sec. 3, we show an application to the computation of the asymptotic variance. Finally, we shed a light on further applications and offer some concluding remarks in Sec. 4.

2 Main Results

In this section, we outline a general recipe for solving Poisson's equation with transform-domain techniques. We will in the current paper generally take the stance that 'the spirit is more important than the letter', and will only briefly address such technical issues as the convergence, existence and uniqueness of solutions for the occurring expressions. As tools for proving such existence exist [6], a rigourous application of the results of this paper thus consists of proving the existence of a PE solution with such means, calculating the solution with the computational method of this paper, and verifying that it indeed satisfies PE afterwards.

Let us assume for a moment that $\mathcal{X} = \mathbb{N}$ and introduce some notation. For a function on the state space recorded in a (either row or column) vector $\mathbf{v} = [v_i]_{,i\in\mathbb{N}}$, let $\mathcal{G}_{\mathbf{v}}$ denote the corresponding generating function, i.e.

$$\mathcal{G}_{\mathbf{v}}(z) := \sum_i z^i v_i. \tag{6}$$

This kind of generating function is common enough in probability theory, as it forms the basis of a quite successful probabilistic method [1,2]. A bit more unusual however is the generating function of a matrix P, which is defined as

$$\mathcal{G}_P(x,y) := \sum_i \sum_j x^i p_{ij} y^j. \tag{7}$$

where we use the convention, here and in the rest of the paper, that sums run over the state space $\mathcal{X}$, unless specified otherwise.

Note: Although all countably infinite sets are isomorphic to $\mathbb{N}$, and hence the above definitions should be sufficient, it is sometimes unnatural to invoke

such isomorphism. For example, when considering a random walk on the quarter plane, it is unnatural to consider anything else than bivariate generating functions, of the type $\mathcal{G}_{\mathbf{v}}(z_1, z_2) := \sum_{i\in\mathcal{X}} \sum_{j\in\mathcal{X}} z_1^i z_2^j v_{ij}$. For the transform of the corresponding transition matrix, the same line of reasoning results in a four-dimensional generating function.

We shall now be concerned with finding the fundamental kernel Z, which has the property that if a solution to Poisson's equation exists, then $\mathbf{h} = Z\mathbf{f}$ is the solution for which $\boldsymbol{\pi}\mathbf{h} = \boldsymbol{\pi}\mathbf{f}$.[1] The matrix Z satisfies (see e.g. [6]):

$$Z = (I - P + \mathbf{1}\boldsymbol{\pi})^{-1}, \tag{8}$$

whenever the inverse exists as a bounded linear operator.

The main starting point is the equation

$$Z(I - P + \mathbf{1}\boldsymbol{\pi}) = I, \tag{9}$$

which in the transform domain translates to

$$\mathcal{G}_Z(x, y) - \mathcal{G}_{ZP}(x, y) + \mathcal{G}_Z(x, 1)\mathcal{G}_{\boldsymbol{\pi}}(y) = \frac{1}{1 - xy}. \tag{10}$$

Note that $\mathcal{G}_Z(x, 1) = 1/(1 - x)$, as $Z\mathbf{1} = \mathbf{1}$, hence we have:

$$\mathcal{G}_Z(x, y) - \mathcal{G}_{ZP}(x, y) = \frac{1}{1 - xy} - \frac{\mathcal{G}_{\boldsymbol{\pi}}(y)}{1 - x}, \tag{11}$$

which is altogether not too different from the transform version of the invariance equation:

$$\mathcal{G}_{\boldsymbol{\pi}}(z) - \mathcal{G}_{\boldsymbol{\pi}P}(z) = 0. \tag{12}$$

Indeed, in both cases, success of the transform-domain recipe largely depends on whether the expression for $\mathcal{G}_{\boldsymbol{\pi}P}(z)$ (resp. $\mathcal{G}_{ZP}(x, y)$) can be conveniently rewritten in terms of $\mathcal{G}_{\boldsymbol{\pi}}(z)$ (resp. $\mathcal{G}_Z(x, y)$). The examples below seem to indicate that if the transform-domain solution can be found for the invariance equation, the corresponding PE derivation is a bit more tedious, but not by much. In the following subsections, we shall explicitly derive expressions for $\mathcal{G}_Z(x, y)$ for a variety of models.

2.1 Reflected Random Walks, Skip-Free to the Left

Let us consider Markov chains with the following often-encountered transition matrix:

$$P = \begin{pmatrix} b_0 & b_1 & b_2 & b_3 & \cdots \\ a_0 & a_1 & a_2 & a_3 & \cdots \\ & a_0 & a_1 & a_2 & a_3 & \cdots \\ & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{13}$$

[1] It is easily checked that if $\mathbf{h}$ is a solution, then also $\tilde{\mathbf{h}} = \mathbf{h} + c\mathbf{1}$, for any c. In this respect, a popular alternative ([8]) for Z is the deviation matrix $D = Z - \mathbf{1}\boldsymbol{\pi}$, so that $\tilde{\mathbf{h}} = D\mathbf{f}$ is the PE solution for which $\boldsymbol{\pi}\tilde{\mathbf{h}} = \mathbf{0}$.

This Markov chain, which is a random walk on $\mathbb{N}$, reflected at 0, has found important applications in queueing theory. Firstly, it appears in the study of the continuous-time $M/G/1$ queue at embedded points, but also in discrete-time queues, it plays an important role. The qualification *skip-free to the left* refers to the fact that while jumps to the right may be arbitrarily large, jumps to the left are at most of size 1.

The invariance equation for this Markov chain has been derived many times. We rederive it in the notation of the paper in order to be able to show the similarities and differences with the PE. Let $\mathcal{G}_{\mathbf{a}}(z) = \sum_i z^i a_i$, $\mathcal{G}_{\mathbf{b}}(z) = \sum_i z^i b_i$ and let $z^{\mathcal{X}}$ stand for the row-vector $(1, z, z^2, \dots)$. The invariance equation in the transform domain (12), simplifies in this case to:

$$\begin{aligned}
\mathcal{G}_{\boldsymbol{\pi}}(z) &= \mathcal{G}_{\boldsymbol{\pi}P}(z) = \boldsymbol{\pi}P\left(z^{\mathcal{X}}\right)^T \\
&= \boldsymbol{\pi}(\mathcal{G}_{\mathbf{b}}(z), \mathcal{G}_{\mathbf{a}}(z), z\mathcal{G}_{\mathbf{a}}(z), z^2\mathcal{G}_{\mathbf{a}}(z), \dots)^T \\
&= \pi_0 \mathcal{G}_{\mathbf{b}}(z) + \sum_{j>0} \pi_j z^{j-1} \mathcal{G}_{\mathbf{a}}(z) \\
&= \mathcal{G}_{\boldsymbol{\pi}}(0)\mathcal{G}_{\mathbf{b}}(z) + (\mathcal{G}_{\boldsymbol{\pi}}(z) - \mathcal{G}_{\boldsymbol{\pi}}(0))\frac{\mathcal{G}_{\mathbf{a}}(z)}{z}. \qquad (14)
\end{aligned}$$

The only unknown in this equation is $\mathcal{G}_{\boldsymbol{\pi}}(0)$, which we can find by differentiating the above equation with respect to z and substituting $z = 1$, which gives after some manipulations:

$$\mathcal{G}_{\boldsymbol{\pi}}(0) = \frac{1 - \mathcal{G}'_{\mathbf{a}}(1)}{1 - \mathcal{G}'_{\mathbf{a}}(1) + \mathcal{G}'_{\mathbf{b}}(1)}. \qquad (15)$$

This leads to the following final expression for $\mathcal{G}_{\boldsymbol{\pi}}(z)$:

$$\mathcal{G}_{\boldsymbol{\pi}}(z) = \frac{1 - \mathcal{G}'_{\mathbf{a}}(1)}{1 - \mathcal{G}'_{\mathbf{a}}(1) + \mathcal{G}'_{\mathbf{b}}(1)} \frac{z\mathcal{G}_{\mathbf{b}}(z) - \mathcal{G}_{\mathbf{a}}(z)}{z - \mathcal{G}_{\mathbf{a}}(z)}. \qquad (16)$$

For Poisson's equation, we need an expression for $\mathcal{G}_{ZP}(x, y)$:

$$\begin{aligned}
\mathcal{G}_{ZP}(x, y) &= x^{\mathcal{X}} ZP \left(y^{\mathcal{X}}\right)^T \\
&= x^{\mathcal{X}} Z(\mathcal{G}_{\mathbf{b}}(y), \mathcal{G}_{\mathbf{a}}(y), y\mathcal{G}_{\mathbf{a}}(y), y^2\mathcal{G}_{\mathbf{a}}(y), \dots)^T \\
&= \left(x^{\mathcal{X}} Z\right)_0 \mathcal{G}_{\mathbf{b}}(y) + \sum_{j>0} \left(x^{\mathcal{X}} Z\right)_j y^{j-1} \mathcal{G}_{\mathbf{a}}(y) \\
&= \mathcal{G}_Z(x, 0)\mathcal{G}_{\mathbf{b}}(y) + (\mathcal{G}_Z(x, y) - \mathcal{G}_Z(x, 0))\frac{\mathcal{G}_{\mathbf{a}}(y)}{y}. \qquad (17)
\end{aligned}$$

which leads to

$$\mathcal{G}_Z(x, y)[y - \mathcal{G}_{\mathbf{a}}(y)] - \mathcal{G}_Z(x, 0)[y\mathcal{G}_{\mathbf{b}}(y) - \mathcal{G}_{\mathbf{a}}(y)] = \frac{y}{1 - xy} - \frac{y\mathcal{G}_{\boldsymbol{\pi}}(y)}{1 - x}. \qquad (18)$$

Still unknown in this equation is the function $\mathcal{G}_Z(x,0)$, which we can recover with almost exactly the same trick as before: we differentiate to y and then substitute $y=1$.

$$\mathcal{G}_Z(x,0) = \frac{1-\mathcal{G}'_{\mathbf{a}}(1)+\mathcal{G}'_{\pi}(1)-x/(1-x)}{(1-\mathcal{G}'_{\mathbf{a}}(1)+\mathcal{G}'_{\mathbf{b}}(1))(1-x)} \tag{19}$$

2.2 Reflected Random Walks, Limited Displacement

We now consider a generalization of the previous section that allows jumps of maximally size c to the left (the standard term *limited displacement* refers to exactly this restriction in leftward jumps):

$$P = \begin{pmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & \cdots \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \\ b_{c-1,0} & b_{c-1,1} & b_{c-1,2} & b_{c-1,3} & \cdots \\ a_0 & a_1 & a_2 & a_3 & \cdots \\ & a_0 & a_1 & a_2 & a_3 & \cdots \\ & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{20}$$

This type of Markov chain finds applications in queues with multiple servers, or in queues with batch servers. Let $\mathcal{G}_{\mathbf{a}}(z) = \sum_i z^i a_i$ and $\mathcal{G}_{\mathbf{b}_i}(z) = \sum_j z^j b_{i,j}$.

We rederive the stationary distribution $\boldsymbol{\pi}$ which leads to

$$\begin{aligned} \mathcal{G}_{\boldsymbol{\pi}}(z) &= \boldsymbol{\pi}(\mathcal{G}_{\mathbf{b}_0}(z), \ldots, \mathcal{G}_{\mathbf{b}_{c-1}}(z), \mathcal{G}_{\mathbf{a}}(z), z\mathcal{G}_{\mathbf{a}}(z), \ldots)^T \\ &= \sum_{i=0}^{c-1} \pi_i \mathcal{G}_{\mathbf{b}_i}(z) + \sum_{i=c}^{\infty} \pi_i z^{i-c} \mathcal{G}_{\mathbf{a}}(z) \\ &= \sum_{i=0}^{c-1} \pi_i \mathcal{G}_{\mathbf{b}_i}(z) + [\mathcal{G}_{\boldsymbol{\pi}}(z) - \sum_{i=0}^{c-1} \pi_i z^i] \frac{\mathcal{G}_{\mathbf{a}}(z)}{z^c}. \end{aligned} \tag{21}$$

After some elementary manipulations, this gives rise to

$$\mathcal{G}_{\boldsymbol{\pi}}(z)[z^c - \mathcal{G}_{\mathbf{a}}(z)] = \sum_{i=0}^{c-1} \pi_i [z^c \mathcal{G}_{\mathbf{b}_i}(z) - z^i \mathcal{G}_{\mathbf{a}}(z)], \tag{22}$$

where π_i is the ith component of the stationary vector $\boldsymbol{\pi}$. We can determine the first c components of this vector by computing the zeros inside the unit circle of the equation $\mathcal{G}_{\mathbf{a}}(z) = z^c$. Rouché's theorem ensures the existence of exactly $c-1$ zeros ζ_j, $|\zeta_j| < 1$, $0 < j < c$, in addition to the zero $\zeta_0 = 1$. We thus find a system of linear equations

$$\sum_{i=0}^{c-1} \pi_i [\zeta_j^c \mathcal{G}_{\mathbf{b}_i}(\zeta_j) - \zeta_j^i \mathcal{G}_{\mathbf{a}}(\zeta_j)] = 0, \text{ for } 0 < j < c. \tag{23}$$

For the PE, we likewise get:

$$\begin{aligned}\mathcal{G}_{ZP}(x,y) &= x^{\mathcal{X}} Z(\mathcal{G}_{\mathbf{b}_0}(y),\ldots,\mathcal{G}_{\mathbf{b}_{c-1}}(y),\mathcal{G}_{\mathbf{a}}(y),y\mathcal{G}_{\mathbf{a}}(y),\ldots)^T \\ &= \sum_{i=0}^{c-1} Z_i(x)\mathcal{G}_{\mathbf{b}_i}(y) + \sum_{i=c}^{\infty} Z_i(x)y^{i-c}\mathcal{G}_{\mathbf{a}}(y) \\ &= \sum_{i=0}^{c-1} Z_i(x)\mathcal{G}_{\mathbf{b}_i}(y) + [\mathcal{G}_Z(x,y) - \sum_{i=0}^{c-1} Z_i(x)y^i]\frac{\mathcal{G}_{\mathbf{a}}(y)}{y^c}, \end{aligned} \tag{24}$$

where $Z_j(x) = \sum_i x^i Z_{ij}$. Substituting this into (11), we obtain:

$$\mathcal{G}_Z(x,y)[y^c - \mathcal{G}_{\mathbf{a}}(y)] - \sum_{i=0}^{c-1} Z_i(x)[z^c\mathcal{G}_{\mathbf{b}_i}(z) - z^i\mathcal{G}_{\mathbf{a}}(z)] = \frac{y^c}{1-xy} - \frac{y^c\mathcal{G}_{\boldsymbol{\pi}}(y)}{1-x}, \tag{25}$$

As with the invariance equation we equally get a system of linear equations by substituting y by ζ_k, $k = 0,...,c-1$. The equation for $k = 0$ is replaced by the equation obtained by deriving (25) w.r.t. y and substituting y by 1, since the former does not provide any additional information.

$$\sum_{i=0}^{c-1} Z_j(x)(j - c + \mathcal{G}'_{\mathbf{a}}(1) - \mathcal{G}'_{\mathbf{b}_i}(1)) = \frac{x/(1-x) - \mathcal{G}'_{\boldsymbol{\pi}}(1) + \mathcal{G}'_{\mathbf{a}}(1) - c}{1-x}. \tag{26}$$

Hence we see that all $Z_j(x)$ are linear combinations of the following form:

$$Z_j(x) = \sum_{k=0}^{c-1} \frac{d_{kj}}{1-x\zeta_k} + \frac{d_{cj}}{(1-x)^2}, \tag{27}$$

so that $Z_j(x)$ can have no other poles than ζ_k^{-1}, and the problem is reduced to finding the different d_{kj}. Introduce matrices A and B where A is $c \times c$ and B is $(c+1) \times c$, and there elements are given as follows:

$$\begin{aligned} A_{jk} &= j - c + \mathcal{G}'_{\mathbf{a}}(1) - \mathcal{G}'_{\mathbf{b}_j}(1),\ k = 0 \\ &= \zeta_k^j - \mathcal{G}_{\mathbf{b}_j}(\zeta_k),\ k > 0 \end{aligned} \tag{28}$$

$$B = \begin{pmatrix} [\mathcal{G}'_{\mathbf{a}}(1) - \mathcal{G}'_{\boldsymbol{\pi}}(1) - c] & -\mathcal{G}_{\boldsymbol{\pi}}(\zeta_1) & -\mathcal{G}_{\boldsymbol{\pi}}(\zeta_2) & \cdots & -\mathcal{G}_{\boldsymbol{\pi}}(\zeta_{c-1}) \\ 0 & 1 & & \cdots & 0 \\ 0 & & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & & & \cdots & 1 \\ x & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{29}$$

Then the (i,j)'th element of BA^{-1} is exactly $d_{i,j}$. Note that apart from the first row, A can also be used for the invariance equation above (for obtaining π_j, $j = 0,...,c-1$). Therefore in essence solving Poisson's equation is numerically not harder than solving the invariance equation.

2.3 Extracting Information from $\mathcal{G}_Z(x, y)$

In this section, we explore how to get information on the infinite matrix Z by means of generating function $\mathcal{G}_Z(x, y)$. Specifically, we focus on results of the form $\boldsymbol{\phi} Z \mathbf{f}$ for a given row vector $\boldsymbol{\phi}$ and a column vector $\mathbf{f}$. First, note that $Z(x, y)$ is in itself a result of this form, with 'geometric' vectors $\boldsymbol{\phi} = [x^i]_{i\in\mathbb{N}}$ and $\mathbf{f} = [y^i]_{i\in\mathbb{N}}$. We first consider the special case for which $\boldsymbol{\phi}$ is geometric but $\mathbf{f}$ has a more general form, so that $\boldsymbol{\phi} Z \mathbf{f} = \mathcal{G}_{\mathbf{h}}(x)$, the generating function of the value function corresponding with cost function $\mathbf{f}$. We assume that all but finitely many f_i have the same sign, so that we can indeed swap sum and integral in the following derivation:

$$\begin{aligned}\mathcal{G}_{\mathbf{h}}(x) &= \sum_{i,j} x^i Z_{ij} f_j \\ &= \frac{1}{2\pi\imath} \sum_j f_j \oint \mathcal{G}_Z(x, y) y^{-j-1} dy \\ &= \frac{1}{2\pi\imath} \oint \mathcal{G}_Z(x, y) \mathcal{G}_{\mathbf{f}}(y^{-1}) y^{-1} dy \,, \end{aligned} \tag{30}$$

where the integrals are taken along a suitable contour. Evaluation can proceed numerically by approximating the integral by a sum, which in fact amounts to the application of fast Fourier transform techniques (FFT). Such approximation techniques are not required if $\mathbf{f}$ has one of the two fundamental forms:

$$f_j^{(1)} = j^n \alpha^j \quad \text{or} \quad f_j^{(2)} = \delta_j^k. \tag{31}$$

We have that $\mathcal{G}_{\mathbf{h}^{(1)}}(x) = \mathcal{G}_{Z\mathbf{f}^{(1)}}(x)$ reduces to

$$\mathcal{G}_{\mathbf{h}^{(1)}}(x) = \left(y \frac{\partial}{\partial y} \right)^n \mathcal{G}_Z(x, y) \Big|_{y=\alpha}, \tag{32}$$

while $\mathcal{G}_{\mathbf{h}^{(2)}}(x) = \mathcal{G}_{Z\mathbf{f}^{(2)}}(x)$ can be written as:

$$\mathcal{G}_{\mathbf{h}^{(2)}}(x) = \frac{\partial^k}{\partial y^k} \mathcal{G}_Z(x, y) \Big|_{y=0} \tag{33}$$

Note that we have closed-form expressions for the generating function of the value function if $\mathbf{f}$ is a (finite) linear combination of such forms as well. We denote the set of such functions as C, which is closed under both addition and multiplication. It contains all constant, polynomial, exponential, functions and all linear combinations and products of such functions. As such, it is a dense set on the set of functions $f : \mathcal{X} \to \mathbb{R}$, meaning that any function can be arbitrarily closely approximated.

We now derive integral expressions for $\boldsymbol{\phi} Z \mathbf{f}$:

$$\begin{aligned}\boldsymbol{\phi} Z \mathbf{f} &= \sum_{i,j} \phi_i Z_{ij} f_j \\ &= -\sum_{ij} \phi_i f_j \frac{1}{4\pi^2} \oint \oint dx dy x^{-i-1} \mathcal{G}_Z(x,y) y^{-j-1} \\ &= -\frac{1}{4\pi^2} \oint \oint dx dy \mathcal{G}_{\boldsymbol{\phi}}(x^{-1}) x^{-1} \mathcal{G}_Z(x,y) \mathcal{G}_{\mathbf{f}}(y^{-1}) y^{-1}.\end{aligned} \tag{34}$$

In case $\boldsymbol{\phi}$ is also of a fundamental form:

$$\phi_j^{(1)} = j^m \beta^j \quad \text{or} \quad \phi_j^{(2)} = \delta_j^\ell, \tag{35}$$

we obtain for each of the four options a different expression for $\boldsymbol{\phi} Z \mathbf{f}$:

$$\boldsymbol{\phi}^{(1)} Z \mathbf{f}^{(1)} = \left(x \frac{\partial}{\partial x} \right)^m \left(y \frac{\partial}{\partial y} \right)^n \mathcal{G}_Z(x,y) \Big|_{(x,y)=(\beta,\alpha)}, \tag{36}$$

$$\boldsymbol{\phi}^{(1)} Z \mathbf{f}^{(2)} = \left(x \frac{\partial}{\partial x} \right)^m \frac{1}{k!} \frac{\partial^k}{\partial y^k} \mathcal{G}_Z(x,y) \Big|_{(x,y)=(\beta,0)}, \tag{37}$$

$$\boldsymbol{\phi}^{(2)} Z \mathbf{f}^{(1)} = \frac{1}{\ell!} \frac{\partial^\ell}{\partial x^\ell} \left(y \frac{\partial}{\partial y} \right)^n \mathcal{G}_Z(x,y) \Big|_{(x,y)=(0,\alpha)}, \tag{38}$$

$$\boldsymbol{\phi}^{(2)} Z \mathbf{f}^{(2)} = \frac{1}{\ell!} \frac{\partial^\ell}{\partial x^\ell} \frac{1}{k!} \frac{\partial^k}{\partial y^k} \mathcal{G}_Z(x,y) \Big|_{(x,y)=(0,0)}. \tag{39}$$

3 Application to Asymptotic Variance

The asymptotic variance (see e.g. [6]) of a functional f on a Markov chain $\{X_n\}$ is defined as follows:

$$\gamma_f^2 = \lim_{N \to \infty} \frac{1}{N} \mathbb{E} : \left[\sum_{n=0}^{N-1} (f(X_n) - \bar{f})^2 \right], \tag{40}$$

where $\bar{f} = \boldsymbol{\pi} \mathbf{f}$, i.e. the expectation of f under stationarity.

This concept plays a key role in establishing a central limit theorem (CLT) for Markov chains, indeed, under some fairly broad conditions [6], we have that the sequence

$$(N \gamma_f^2)^{-1/2} \sum_{n=0}^{N-1} (f(X_n) - \bar{f}),$$

converges to a normal distribution with zero mean and unit variance as $N \to \infty$.

This property has many important applications, for example in planning simulations, but also in heavy-traffic theory, which we will illustrate in Sec. 3.2.

3.1 Calculation of γ_f^2

Calculating γ_f^2 directly from the formula (40) often leads to long-winded derivations, which can be alleviated if we carry it out in the following way: for $\mathbf{h} = Z\mathbf{f}$, we have (see [6]):

$$\begin{aligned}\gamma_f^2 &= \sum_i \pi_i((h_i)^2 - (h_i - f_i + \bar{f})^2) \\ &= 2\sum_i \pi_i h_i f_i - \sum_i \pi_i (f_i)^2 - (\bar{f})^2. \end{aligned} \tag{41}$$

where we made use of the fact that $\boldsymbol{\pi}\mathbf{h} = \boldsymbol{\pi}\mathbf{f}$. Note that only the first term involves $\mathbf{h}$. We show how to compute this term by using the techniques of Sec. 2.3. Note that

$$\sum_i \pi_i h_i f_i = \sum_{i,j} \pi_i f_i Z_{ij} f_j. \tag{42}$$

If both π_i and f_i belong to the set C as defined in Sec. 2.3, then we obtain a closed-form expression for the asymptotic variance. This implies that $\mathcal{G}_{\boldsymbol{\pi}}(z)$ must be a rational function.

3.2 An Application to Heavy Traffic

Since Kingman [16], heavy-traffic theory is an important topic in queueing theory. In the following, we give a very short account of the results for a single queue (i.e. the $G/G/1$ model). In keeping with the conventions of rest of the paper, we stick to a discrete-time scenario, although extensions to continuous time are fairly straight-forward. Consider a queue with a stationary but not necessarily independent input process $\{A_n\}$ and a capacity c. The virtual work W_n at time instant n is given by the recursive equation

$$W_{n+1} = (W_n + A_n - c)^+ \tag{43}$$

Heavy-traffic theory is concerned with the limit as $\mathbb{E} : [A_0] \uparrow c$ and more specifically states that if the process $\{A_n\}$ admits a CLT with mean ρ and asymptotic variance γ^2, then the mean virtual work in stationarity is equal to

$$E[W_\infty] = \frac{\gamma^2}{2(c-\rho)} \tag{44}$$

Moreover the stationary distribution converges to an exponential distribution with a mean given by the above display.

In order to apply the results of this paper to a concrete queueing scenario. We consider a two-queue model where the content of the first queue serves as the input for the second queue:

$$(U_{n+1}^{(1)}, U_{n+1}^{(2)}) = ([U_n^{(1)} + A_n - 1]^+, [U_n^{(2)} + U_n^{(1)} - 1]^+) \tag{45}$$

This may seem like an artificial model, but it might be used in the following scenario. People are waiting in line for a service (for example, buying a ticket at the movies), and while doing so, they make use of an internet hotspot, thus creating internet traffic proportional to the number of waiting customers. A more pragmatic reason is that it allows us to make direct use of the results for the class of Markov chains of Sec. 2.1. With some extra work, e.g. by applying our recipe to the two-dimensional Markov chain $\{(U_n^{(1)}, D_n^{(1)})\}$, where $D_n^{(1)}$ denotes the number of departures from the first queue, we may also consider more traditional scenarios such as tandem queues etc., which in fact leads to the asymptotic variance of departure processes as considered eg. in [15].

If we assume the second queue to be operating close to its maximal capacity, then we can apply the aforementioned heavy-traffic results, as the sequence $\{U_n^{(1)}\}$ forms a (dependent) input process for the second queue. We compute the value of γ_f^2 in closed form. Note that in this particular case, $f_i = i$. Let us apply the results of Sec. 2.1 with $\mathcal{G}_{\mathbf{a}}(z) = \mathcal{G}_{\mathbf{b}}(z) = a_0 + a_1 z + a_2 z^2$. Note that we can derive closed-form solutions under broader conditions that we do here, but we opt for a simple, tractable example. Let $\rho_1 = a_1 + 2a_2$. We have that $\mathcal{G}_{\boldsymbol{\pi}}(z) = (a_0 - a_2)/(a_0 - a_2 z)$, such that $\pi_i = (1-r)r^i$, with $r = a_2/a_0$. After some (fairly straightforward) algebraic manipulations, we find that

$$\mathcal{G}_Z(x,y) = \frac{1}{a_0 - a_2 y}\Big(\frac{y\,(a_0 x - a_2)}{(1-x)(a_0 - a_2 y)(1-xy)} + \frac{a_0 - a_2 + a_0/(a_0 - a_2)}{1-x} - \frac{x}{(1-x)^2}\Big). \quad (46)$$

Eq. (41) specializes in this case to

$$\gamma^2 = 2(1-r)\boldsymbol{\phi} Z \mathbf{f} - \frac{a_0 a_2 + 2a_2^2}{(a_0 - a_2)^2}. \quad (47)$$

Note that $\phi_i = ir^i$ and $f_i = i$, so that

$$\boldsymbol{\phi} Z \mathbf{f} = \frac{\partial^2}{\partial x \partial y} \mathcal{G}_Z(x,y)\Big|_{(x,y)=(r,1)}, \quad (48)$$

which leads to a long but closed-form expression for γ^2.

In Figure 1, we illustrate the results of this section with a numerical example. Note that there are only two degrees of freedom for the the parameters of the first queue. When we fix the load ρ_2 as seen by the second queue (which is equal to $a_2/(a_0 - a_2)$, the mean content of the first queue), then a plausible choice for the other parameter is given by a_1, which is a measure for how quickly the queue content process varies. We see that γ^2 increases as a_1 gets larger.

4 Further Applications and Concluding Remarks

As there are plenty of applications for Poisson's equation next to asymptotic variance, we presume that the methods of this paper may be useful in other

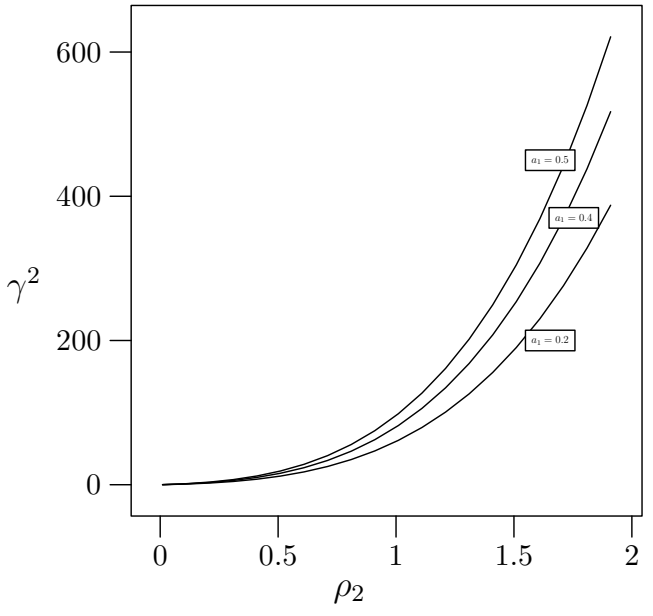


Fig. 1. Plot of the asymptotic variance γ^2 of the input stream against load of the second queue for different values of a_1

areas than in asymptotic variance. We give some initial findings on a few of them.

Controlled Markov chains, or Markov decision processes, are perhaps the most obvious application. As this work may help find efficient solution methods for the value function (often a rather costly step), it may lead to better algorithms for some classes of MDP. It seems of utmost importance however that every policy leads to a Markov chain that is sufficiently structured.

Another application is perturbation of Markov chains. Consider a family of Markov chains depending on a parameter α with transition matrices $P^{(\alpha)}$. For small values of α, The central formula for the stationary vector of a perturbed system is as follows:

$$\boldsymbol{\pi}^{(\alpha)} = \boldsymbol{\pi}^{(0)} \sum_k ((P^{(\alpha)} - P^{(0)})(Z^{(0)} - \mathbf{1}\boldsymbol{\pi}^{(0)}))^k.$$

Although perturbation problems can sometimes be tackled in the transform domain without a detour to PE (see eg. [19]), the approach of this paper may lead to a higher genericity, and potentially to more insight.

Summing up, we have derived transform-based solutions of Poisson's equation for some frequently encountered types of Markov chains and have pointed out some applications. We show that the transform domain may form an attractive

tool for researchers working with Poisson's equation, and also that Poisson's equation may offer new results and applications for models that allow transform-based solutions.

References

1. Cohen, J.: The single-server queue. North-Holland Series in Appl Math. and Mech. (1969)
2. Takagi, H.: Queueing analysis, a foundation of performance evaluation. Discrete-time systems, vol. 3. Elsevier Science Publishers BV, Amsterdam (1993)
3. Bellman, R.: On the Theory of Dynamic Programming. Proceedings of the National Academy of Sciences (1952)
4. Glynn, P.W., Meyn, S.P.: A Liapounov bound for solutions of the Poisson equation. Ann. Probab. 24(2), 916–931 (1996)
5. Schweitzer, P.J.: Perturbation theory and finite Markov chains. J. Appl. Prob. 5, 401–403 (1968)
6. Meyn, S.P., Tweedie, R.L.: Markov chains and stochastic stability. Springer, London (2003)
7. Meyn, S.P.: Control Techniques for Complex Networks. Cambridge University Press (2007)
8. Asmussen, S.: Applied probability and queues, 2nd edn. Springer (2003)
9. Neveu, J.: Potentiel Markovien récurrent des chaînes de Harris. Ann. Inst. Fourier 22(7), 130 (1972)
10. Jones, G.L.: On the Markov chain central limit theorem. Probab. Surv. 1, 299–320 (2004)
11. Nummelin, E.: On the Poisson equation in the potential theory of a single kernel. Math. Scand. 68, 59–82 (1991)
12. Shwartz, A., Makowski, A.: On the Poisson equation for Markov chains: existence of solutions and parameter dependence. Technical report, Dept. Electrical Engineering, Technion – Israel Institute of Technology (1991)
13. Koole, G., Spieksma, F.: On deviation matrices for birth-death processes. Probability in the Engineering and Informational Sciences 15, 239–258 (2001)
14. Hordijk, A., Spieksma, F.M.: On ergodicity and recurrence properties of a Markov chain with an application. Adv. Appl. Probab. 24, 343–376 (1992)
15. Nazarathy, Y.: The Variance of Departure Processes: Puzzling Behavior and Open Problems. Queueing Systems 68, 385–394 (2011)
16. Kingman, J.F.C.: On Queues in Heavy Traffic. Journal of the Royal Statistical Society. Series B (Methodological) 24(2), 383–392 (1962)
17. White, L.B.: A new policy evaluation algorithm for Markov decision processes with quasi birth-death structure. Stochastic Models 21, 785–797 (2005)
18. Lambert, J., van Houdt, B., Blondia, C.: A policy iteration algorithm for Markov decision processes skip-free in one direction. Numerical Methods for Structured Markov Chains (2007)
19. Fiems, D., Prabhu, B., De Turck, K.: Analytic approximations of queues with lightly- and heavily-correlated autoregressive service times. Annals of Operations Research

Analytical and Stochastic Modelling of Battery Cell Dynamics

Ingemar Kaj[1] and Victorien Konané[2,3]

[1] Department of Mathematics, Uppsala University
ikaj@math.uu.se
[2] International Science Program, Uppsala University
[3] Department of Mathematics, University of Ouagadougou, Burkina Faso

Abstract. In this work we present and discuss a modelling framework for the basic discharge process which occurs in simple electrochemical battery cells. The main purpose is to provide a setting for analyzing delivered capacity, battery life expectancy and other measures of performance. This includes a number of deterministic and stochastic variations of kinetic battery models. The primary tool is a novel phase plane analysis of the balance of nominal and theoretical capacity. In particular, we study spatial versions of such models which lead to a linear diffusion equation with Robin type boundary conditions under scaling. Explicit solutions are obtained by considering reflected Brownian motion.

1 Introduction

This work concerns mathematical modeling of the state of charge and the voltage level dynamics in simple battery cells under discharge. The purpose is to provide an efficient framework for predicting battery life, delivered capacity and other measures of performance, which takes into account that batteries are commonly subject to considerable variation in performance. Such variations occur not only because of variable usage patterns or variable disload mechanisms of the electrochemically stored energy, but also as a result of recovery mechanisms in the electrolyte. The type of battery we have in mind primarily is a non-rechargeable and non-costly unit expected to last several years, such as a 3 Volts lithium-ion coin battery to be deployed in large numbers for low-energy applications in communication networks, sensor networks, etc.

A battery is made of one or several electrochemical cells. The modeling discussed here relies on the simplified view that a cell essentially consists of an anode-cathode pair of electrodes connected by electrolyte, which may be liquid as in lead-acid batteries or solid as in Li-ion batteries. In the cell, stored chemical energy is converted into electrical energy through an oxidation reaction at the anode. By Faraday's first law the mass of active material altered at an electrode is directly proportional to the quantity of electrical charge which is transfered at the electrode in the battery reaction. The Nernst equation in electrochemistry then states that the logarithm of electric charge determines the terminal voltage

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 240–254, 2012.

that exists between the pair of electrode terminals. It is the terminal voltage that measures the ability of the battery to drive electric current.

The terminal voltage for a battery in a state of rest is typically larger in magnitude than the terminal voltage under discharging due to effects of internal resistance. Batteries for digital applications would often be expected to deliver power spikes, either periodically in time or at random time points. Such pulsed discharge patterns may have a different effect on terminal voltage to continuous discharge loads of constant current.

The theoretical capacity of the battery is a measure of the maximal charge which in principle could be obtained were the battery discharged arbitrarily slowly, allowing the chemical reaction to equilibriate over time restricted only by the total amount of active material contained in the cell. The nominal capacity of a battery is typically a manufacturers specification of the amount of electric charge which is delivered if the cell is put under constant load and drained of its energy over a certain time interval. Normally the discharge process occurs on some intermediate time scale that allows for recovery mechanisms to take place. This may slow down the decrease of the state of charge or even cause the state of charge to increase. A further mechanism known to affect the performance of a battery and which we will take into account is the balance between migration and solid state diffusion. The battery stops functioning if the terminal voltage passes below a minimal acceptable level or if the battery runs out of theoretical capacity.

In addition to introducing new modeling variations our study provides a survey of a number of battery models discussed in the literature. While the simplest kinetic battery model introduced by Manwell and McGowan, [6,7], is essential for our approach as a reference and background, we also consider the spatial extension of this model, [10]. We do not discuss, however, another modeling approach based on discrete Markov chains, see e.g. [1].

We summarize the novel contributions in this work as follows. Based on an approach focusing on the interplay between remaining nominal capacity and remaining theoretical capacity during discharge evolution we perform what appears to be a novel phase plane analysis of battery capacities. This allows us to obtain battery life, gained capacity and delivered capacity as functions of the basic model parameters and in some cases to optimize performance over such parameters. The setting begins with the two-well kinetic battery ODE model of constant current discharge but includes general situations such as regular pulsed discharge or stochastic pointwise discharge. The unified approach to general workload patterns and comparison of these appears to be new. We also propose a new kinetic-diffusive battery model designed to describe the balance between migration and drift diffusion. Finally, we extend the modeling approach and generality of the models to a version where the bound charge is supposed to be distributed over a spatial reservoir.

2 Some Principles of Battery Cell Dynamics

To introduce the main ideas of our approach, we consider a battery which is initially fully loaded with nominal capacity N and which has the theoretical

capacity T at time $t = 0$. Realistically, $N \leq T$. For $t \geq 0$ let $x(t)$ denote the level of available charge and $v(t)$ the level of remaining theoretical capacity of the battery at time t, so $(v(0), x(0)) = (T, N)$. Charge is drawn from the battery either continuously, or such that the charge level drops instantly from one discrete level to a lower level. The discharge process acts randomly or in a deterministic fashion and it acts continuously in time or at discrete time epochs.

The normalized and dimensionless quantity $\widetilde{x}(t) = x(t)/N \in [0, 1]$ represents the state of charge of the battery at time t. A fully charged battery has $\widetilde{x}(t) = 1$ and an empty one $\widetilde{x}(t) = 0$. In practice a battery stops functioning before it runs out of charge entirely at some level x_0 which corresponds to voltage reaching a cut-off level E_c. According to the Nernst equation the concentration C of active material at an electrode determines the potential E according to $E = E_0 - K_e \ln C$, where E_0 is the equilibrium potential and $K_e = RT_a/nF$ with R the ideal gas constant, T_a absolute temperature, n the valency of the battery reaction ($n = 1$ for Lithium), and F Faraday's constant. Faraday's law identifies the capacity Q of the cell as a multiple of C, with a proportionality constant which is F times volume. Since $\widetilde{x}(t)$ is a measure of capacity we obtain the terminal voltage E_t of the cell at time t, as

$$E_t = E_0 + K_e \ln \widetilde{x}(t), \quad 0 \leq t \leq t_0,$$

where t_0 is the battery life given by the first instance at which E_t reaches the cut-off voltage $E_c < E_0$. For example, a Li-ion battery may have $E_0 = 3$ and $E_c = 2$ Volt.

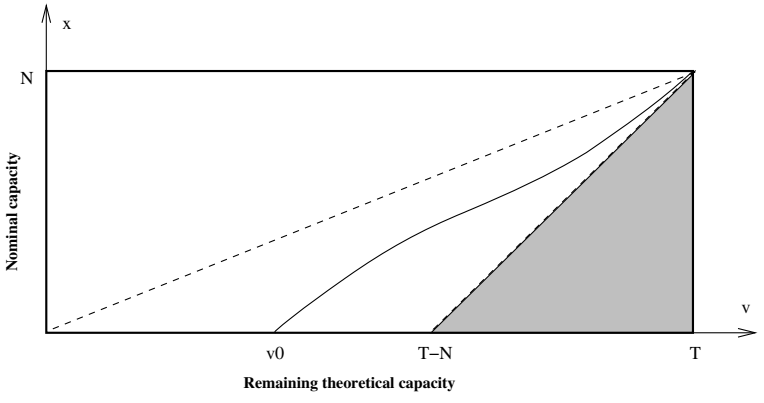


Fig. 1. Phase-plane trace of nominal and theoretical capacities

It is natural to consider the trajectory of the system $(v(t), x(t))$, $t \geq 0$, as a path in the (v, x) phase plane starting in (T, N) at $t = 0$. Initially, the path moves downwards and to the left in the (v, x) plane as the nominal and hence the remaining theoretical capacity decreases. While the remaining theoretical capacity continues its descent with the same average rate as the discharge process, it is reasonable that the battery recovers some nominal charge capacity.

This is because chemical transport in the electrolytes enables previously stored material to become available, at least if λ is not too large. This effect is likely to be less effective at lower levels of nominal charge.

If there exists a solution $v_0 > 0$ of $x(v) = x_0$ then v_0 is the remaining capacity at the battery charge expiration time. For this case we note that $D = T - v_0$ is the delivered capacity of the battery. We expect, based on the brief discussion above, that D tends to T if $\lambda \to 0$. Also, we normalize the nominal capacity such that D tends to N if $\lambda \to \infty$, c.f. [9], Figure 6. A closely related quantity is the gain of the battery, $G = T - N - v_0$. This is the capacity which is gained during the life of the battery and measures the amount of bound charge that the battery was able to convert into available charge and deplete during its time of operation. Figure 1 indicates a typical trace in the phase-plane starting from (T, N) and ending in $(v_0, 0)$.

3 The Kinetic Battery Model

The Kinetic Battery Model, [10], is a deterministic modeling approach which assumes that charge is drawn continuously over time according to a given discharge current $i(t) \geq 0$, $t \geq 0$. The average discharge rate is $\bar{\lambda} = \lim_{t\to\infty} \frac{1}{t} \int_0^t i(s)\, ds$, if this limit exists. The most basic case is a battery subject to constant discharge over time, $i(t) = \lambda$. The total theoretical capacity of the battery is split in two components called available charge and bound charge. As above, for $t \geq 0$ let $v(t)$ denote the total capacity and $x(t)$ the available capacity of the battery. Call $y(t) = v(t) - x(t)$ the bound charge. Suppose $x(0) = N$, $v(0) = T > N$.

The kinetic battery model involves a parameter $c \in (0, 1)$ which allows the ratios $x(t)/c$ and $y(t)/(1-c)$ to be interpreted as the current heights of an available charge well and a bound charge well, respectively. During operation of the battery, bound charge is supposed to be made available according to a rate which is proportional to the height difference $y(t)/(1-c) - x(t)/c$ between the bound and available charge wells. The matching with initial conditions then dictates one should take $c = N/T \in (0, 1)$ to be the fraction of total theoretical capacity which is initially made available. Then at time $t = 0$, the wells have equal height T and the charge flow gradient which builds up between the two wells represents recovery of the battery, in the sense of its positive growth effect on the nominal charge level x. As a result we obtain for $(x(t), y(t))$ the linear system of differential equations

$$\begin{cases} x'(t) = -i(t) + k\Big(\frac{y(t)}{1-c} - \frac{x(t)}{c}\Big), & x(0) = N \\ y'(t) = -k\Big(\frac{y(t)}{1-c} - \frac{x(t)}{c}\Big), & y(0) = T - N, \end{cases}$$

where k is a conductance parameter. Hence the total discharge process is independent of the charge flow gradient and we have

$$v(t) = x(t) + y(t) = T - \int_0^t i(s)\, ds, \quad t \geq 0.$$

The linear system is readily solved in terms of the parameters k, T and $c = N/T$, as

$$\begin{cases} x(t) = cv(t) - (1-c)\int_0^t e^{-k(t-s)/c(1-c)}\, i(s)\, ds \\ y(t) = (1-c)v(t) + \int_0^t e^{-k(t-s)/c(1-c)}\, i(s)\, ds. \end{cases}$$

Since $cv(t) - x(t) \geq 0$ the system life equals

$$t_0 = \inf\{t > 0 : x(t) = x_0 \text{ or } v(t) = 0\} = \inf\{t > 0 : x(t) = x_0\}.$$

The model in this form is discussed in e.g. [5].

Phase Plane Analysis and General Workload Discharge. It is straightforward to extend the kinetic battery model and incorporate general discharge patterns by replacing $i(t)\, dt$ with some measure $\Lambda(dt)$, and consider the differential system

$$\begin{cases} dx(t) = -\Lambda(dt) + k\Big(\frac{y(t)}{1-c} - \frac{x(t)}{c}\Big)\, dt, & x(0) = N \\ dy(t) = -k\Big(\frac{y(t)}{1-c} - \frac{x(t)}{c}\Big)\, dt, & y(0) = T - N. \end{cases}$$

Then $v(t) = x(t) + y(t) = T - \Lambda(t)$ and

$$\begin{cases} x(t) = cv(t) - (1-c)\int_0^t e^{-k(t-s)/c(1-c)}\, \Lambda(ds) \\ y(t) = (1-c)v(t) + \int_0^t e^{-k(t-s)/c(1-c)}\, \Lambda(ds). \end{cases}$$

For example, if a cell is subject to successive periods of low, medium and high loads then $\Lambda(dt)$ would be a continuous measure with discharge rates varying accordingly from one time interval to the next.

Next we analyze in some detail and compare three types of discharge patterns for the kinetic battery model in its extended form. First of all, for the constant discharge case $\Lambda(t) = \lambda t$ we obtain the solution $(v_\lambda(t), x_\lambda(t))$, where $v_\lambda(t) = T - \lambda t$ and

$$x_\lambda(t) = cv_\lambda(t) - \lambda c(1-c)^2 k^{-1}(1 - e^{-kt/c(1-c)}). \tag{1}$$

In line with the phase plane view point in Figure 1, the corresponding trajectory of the bivariate dynamical system $(v_\lambda(t), x_\lambda(t))$ in the (v, x)-plane with terminal condition $x(T) = N$ is

$$x(v) = cv - C_\lambda(1 - e^{-k(T-v)/\lambda c(1-c)}), \quad v \leq T, \quad C_\lambda = \lambda c(1-c)^2/k. \tag{2}$$

This system starts in $(v, x) = (T, N)$ at time $t = 0$ and traces out a path below the diagonal $x = cv$ but above the line $x = -(T-N) + v$ which exits at time t_0 in (v_0, x_0), where $0 \leq v_0 \leq T - N$. The phase plane path depends on the parameters λ and k only through the ratio λ/k. Figure 2 shows the drop of voltage according to $E_t = 3 + 0.2\ln(x(t)/N)$ as a function of normalized capacity $1 - v(t)/T$ (upper panel) and phase plane curves (lower panel) until the cut-off voltage of 2 Volts is reached, which occurs close to the time of complete discharge where $x(t) \approx 0$.

Three discharge rates, $\lambda/k = 500, 1000, 2500$ are indicated and compared to the ideal case $x = cv$ for a very small discharge current, which corresponds to 100% utilization, and the worst case of highest loads where 40% of the available capacity is utilized.

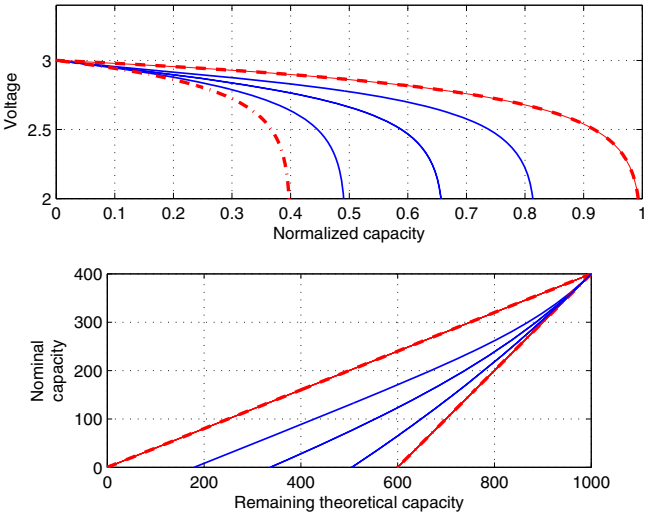


Fig. 2. Upper panel: Voltage versus (normalized) capacity; Lower panel: Discharge profiles for the kinetic battery model; Parameters: constant current load, $N = 400$, $T = 1000$, $\lambda/k = 500, 1000, 2500$

Another case of interest is regularly spaced pulsed discharge. This is a relevant model for batteries in sensor nodes programmed to carry out a fixed task once per day, say. Here we let $r > 0$ be the time between any two consecutive pulses each releasing the charge λr. The corresponding discharge measure is $\Lambda(t) = \lambda r \sum_{j=1}^{\lfloor t/r \rfloor} \delta_{jr}$, for which the average discharge rate is kept at (approximately) λt. The solution $(x^{(r)}(t), v^{(r)}(t))$ for this case is

$$v^{(r)}(t) = T - \Lambda(t) \approx T - \lambda t$$

$$x^{(r)}(t) = cv^{(r)}(t) - (1-c)\lambda r \sum_{j=1}^{\lfloor t/r \rfloor} e^{-k(t-jr)/c(1-c)},$$

where evaluation of the geometric sum $\sum_{j=1}^{\lfloor t/r \rfloor} (e^{kr/c(1-c)})^j$ yields

$$x^{(r)} \approx cv^{(r)} - C_\lambda^{(r)}(1 - e^{-k(T-v^{(r)})/\lambda c(1-c)}), \quad C_\lambda^{(r)} = \frac{\lambda r(1-c)}{1 - e^{-kr/c(1-c)}}. \tag{3}$$

The previous case $\Lambda(t) = \lambda t$ is recovered by taking an informal limit $r \to 0$. Strictly speaking, the curves with constants $C_\lambda^{(r)}$ indicate the lower jump points

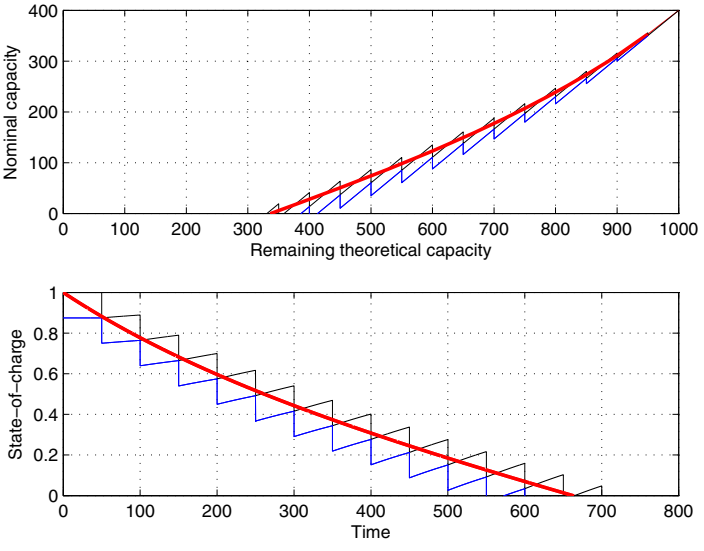


Fig. 3. Discharge of pulsed kinetic battery model, $r = 50$, $N = 400$, $T = 1000$, $\lambda/k = 1000$

of the discharge profile. This is appropriate as the battery is considered empty at the first instance when the voltage drops below E_c. Figure 3 shows phase plane and the change of state with time for the kinetic battery model under pulsed discharge with $r = 50$ and additional parameters $N = 400$, $T = 1000$, $\bar{\lambda}/k = 1000$. Two curves are shown for the case when the first pulse occurs at $t = 0$ (blue) or $t = r$ (black) and compared with the constant discharge case of same average (red).

As a third example we let $(\Lambda(t))_{t\geq 0}$ be a Poisson process with intensity measure $\lambda\, dt$. This is a battery released of charge pulsewise at random times uniformly scattered over the time interval of operation with an average of λ per time unit. The result is a system of random processes $(V(t), X(t))$ with $V(t) = T - \Lambda(t)$ and

$$X(t) = cV(t) - (1-c)Z(t), \quad Z(t) = \int_0^t e^{-k(t-s)/c(1-c)}\, d\Lambda(s),$$

and $Z(t)$ is known as a so called Poisson shot-noise process. Clearly, the expected value $EX(t) = x_\lambda(t)$ is given by (1). Moreover, the shot-noise process has a steady-state Z_∞, such that asymptotically

$$Z(t) \Rightarrow Z_\infty, \quad EZ_\infty = \lambda c(1-c)/k, \quad \mathrm{Var} Z_\infty = \lambda c(1-c)/2k. \tag{4}$$

Kinetic-Diffusive Battery Model. The kinetic battery model was primarily framed for lead-acid batteries. Shortcomings of the model have been discussed in e.g. [3], and attempts have been made to incorporate other designs. In the context of Ni-MH batteries, [9] proposed a modified, non-linear, factor in the flow charge between the two wells.

For Li-ion cells a shortcoming of the kinetic battery model appears to be that solid state diffusion is not taken into account. In solid phase the application of an external driving force makes the diffusing particles experience a drift motion in addition to random diffusion. This effect of diffusion drift of charge carriers is discussed in detail in the specialized electrochemical litterature on all-solid batteries and is known to hamper performance of the units, see e.g. [2]. We propose the following modification of the dynamics of the two-well kinetic battery model as a means of introducing in a simplistic but meaningful way a negative drift in the flux of charge:

$$\begin{cases} dx(t) = -\Lambda(dt) + k\Big((1-p)\Big(\frac{y(t)}{1-c} - \frac{x(t)}{c}\Big) - p\frac{1}{c}\Big(\frac{N}{c} - \frac{y(t)}{1-c}\Big)\Big)\,dt, & x(0) = N \\ dy(t) = -k\Big((1-p)\Big(\frac{y(t)}{1-c} - \frac{x(t)}{c}\Big) - p\frac{1}{c}\Big(\frac{N}{c} - \frac{y(t)}{1-c}\Big)\Big)\,dt, & y(0) = T - N. \end{cases}$$

Here, p, $0 \le p \le 1$, signifies a fraction of the current flow of charge which is removed and sent back to the bound well. The solution in this case is given by

$$\begin{cases} x(t) = (c + (1-c)p)v(t) - (1-c)pT - (1-p)(1-c)\int_0^t e^{-k(t-s)/c(1-c)}\,\Lambda(ds) \\ \qquad = (1-p)\Big(cv(t) - (1-c)\int_0^t e^{-k(t-s)/c(1-c)}\,\Lambda(ds)\Big) + p(v(t) - (T-N)) \\ v(t) = x(t) + y(t) = T - \Lambda(t) \end{cases}$$

Now choose a discharge measure $\Lambda(dt)$. It is then straightforward to derive results such as (2, 3) for the more general model that involves the drift parameter p.

Performance of the Kinetic Battery Model. Here we compare briefly delivered capacity and battery life for the kinetic battery model. For simplicity we consider the standard model $p = 0$. All formulas listed in this section may also be derived for the case $0 < p < 1$ of the kinetic-diffusive modification discussed above. Indeed, we conclude this section with some comments on delivered capacity for the general model.

We begin with the non-random models. The unused capacity that remains after depletion of all available charge is the unique solution $v_0 > 0$ of $x(v) = x_0$. The delivered capacity is $D = T - v_0$ and the gained capacity $G = D - N$. By (3) and (2), which we include as the case $r = 0$,

$$x_0 = cv_0 - C_\lambda^{(r)}(1 - e^{-k(T-v_0)/\lambda c(1-c)}), \quad C_\lambda^{(0)} = C_\lambda. \tag{5}$$

Equation (5) may be solved explicitly in terms of the so called Lambert W function, the principal branch of which we denote by W_0. Then

$$v_0 = \frac{x_0 + C_\lambda^{(r)}}{c} - \frac{\lambda c(1-c)}{k}\,W_0\Big(\frac{kC_\lambda^{(r)}}{\lambda c^2(1-c)}\exp\Big\{-\frac{k(T - x_0/c - C_\lambda^{(r)})}{\lambda c(1-c)}\Big\}\Big)$$

However, for typical parameters the exponential term in (5) may be ignored for v close to v_0, and hence $v_0 \approx (x_0 + C_\lambda^{(r)})/c$. In conclusion, the delivered capacity $D_\lambda^{(r)}$ for the deterministic version of the kinetic battery model is approximately

$$D_\lambda^{(r)} \approx T - x_0/c - C_\lambda^{(r)}/c.$$

The lifelength t_0 of the battery is directly proportional to the delivered capacity. Indeed, since $(v(t_0), x(t_0)) = (v_0, x_0)$ and $v(t) = T - \lambda t$ we have $\lambda t_0 = D$. As an example, the lifelength for the continuous model (2) is obtained as the solution $t_0 \in [T/\lambda - 1/k, T/\lambda]$ of

$$t_0 = \frac{T}{\lambda} - \frac{x_0}{c\lambda} - \frac{1}{k}(1-c)^2(1 - e^{-kt_0/c(1-c)}).$$

Again this equation may be solved in terms of W_0, as

$$t_0 = \frac{T}{\lambda} - \frac{x_0}{c\lambda} - \frac{1-c}{k} + \frac{c(1-c)}{k} W_0\Big(\frac{1}{c}\exp\Big\{-\frac{k(T - x_0/c)}{\lambda c(1-c)} + \frac{1}{c}\Big\}\Big).$$

Turning to the random model driven by Poisson discharge events, if we stop at the random time $t_0 = \min\{t : X(t) = x_0\}$, then by (4),

$$cV(t_0) \approx x_0 + (1-c)Z_\infty \approx x_0 + (1-c)EZ_\infty = x_0 + C_\lambda.$$

Hence

$$\Lambda(t_0) \approx T - \frac{x_0 + C_\lambda}{c} = T(1 - \frac{x_0 + C_\lambda}{N})$$

and so the distribution of the lifelength t_0 is comparable to a Gamma distribution $\Gamma(m, \lambda)$ where m is an integer approximation of $T(1 - (x_0 + C_\lambda)/N)$. Also, the average delivered capacity for the Poisson model is given by $D_\lambda = T(1 - (x_0 + C_\lambda)/N)$.

As mentioned above it is straightforward to include the kinetic-diffusive version. The delivered capacity will decrease with increasing p as illustrated in Figure 4, with $p = 0$, $p = 0.2$ and $p = 0.4$. For the basic model $p = 0$, the delivered capacity D_λ descends from its maximal value T, or $T(1 - x_0/N)$ in case $x_0 > 0$, to the asymptotic value N as $\lambda \to \infty$. For $p > 0$ however there is a maximal $T_p \approx (N - x_0)/((1-p)c + p)$ which is attained for vanishing load, $\lambda \to 0$. The interpretation is that solid phase diffusion puts a principal restriction on the amount of chemical energy which can be drawn from the cell.

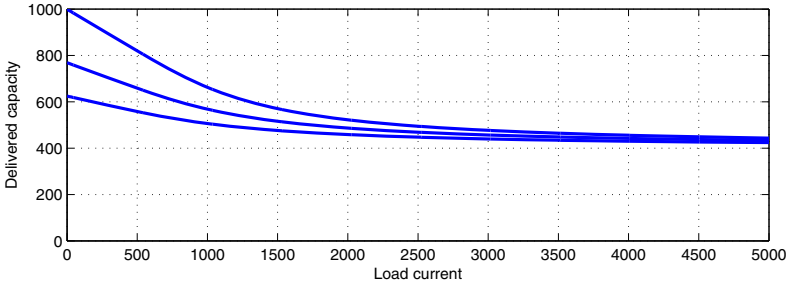


Fig. 4. Delivered capacity D as function of current load λ/k for $p = 0$ (upper), $p = 0.2$ (mid) and $p = 0.4$ (lower); $T = 1000$, $N = 400$

4 Spatial Diffusion Models

This approach extends the mechanism of the kinetic battery model to act pairwise on adjacent fluid compartments spread out on a one-dimensional spatial range. We begin with a discretized version and split the charges in m components $u(t) = (u_1(t), \dots, u_m(t))$. Here u_1 is the available charge, u_2 is a bound well charge for u_1 and so on until u_m, which is a bound well charge for u_{m-1}. As charge is drawn from u_1 electrochemical material continuously flows downwards from each bound well to help recover the charge level at the next lower well. By considering a limit of many small wells we obtain a limiting PDE for the charge concentration profile, which may be solved explicitly and analyzed by phase plane methods just as for the two-well case.

Spatial Version of the Kinetic Battery Model. By treating each pair of adjacent components as available and bound wells, we obtain the coupled system of linear equations

$$\begin{cases} du_1(t) = -\Lambda(dt) + k_c(cu_2(t) - (1-c)u_1(t))\,dt \\ du_2(t) = -k_c(cu_2(t) - (1-c)u_1(t))\,dt + k_c(cu_3(t) - (1-c)u_2(t))\,dt \\ \qquad \vdots \\ du_{m-1}(t) = -k_c(cu_{m-1}(t) - (1-c)u_{m-2}(t))\,dt \\ \qquad\qquad\qquad +k_c(cu_m(t) - (1-c)u_{m-1}(t))\,dt \\ du_m(t) = -k_c(cu_m(t) - (1-c)u_{m-1}(t))\,dt, \end{cases}$$

where we have put $k_c = k/c(1-c)$ as a temporary notation. Rewriting,

$$\begin{cases} du_1(t) = -\Lambda(dt) + \frac{k_c}{2}(u_2(t) - u_1(t))\,dt + k_c(c-1/2)(u_1(t) + u_2(t))\,dt \\ du_2(t) = \frac{k_c}{2}(u_1(t) - 2u_2(t) + u_3(t))\,dt + k_c(c-1/2)(u_3(t) - u_1(t))\,dt \\ \qquad \vdots \\ du_{m-1}(t) = \frac{k_c}{2}(u_{m-2}(t) - 2u_{m-1}(t) + u_m(t))\,dt \\ \qquad\qquad\qquad +k_c(c-1/2)(u_m(t) - u_{m-2}(t))\,dt \\ du_m(t) = -\frac{k_c}{2}(u_m(t) - u_{m-1}(t)) - k_c(c-1/2)(u_{m-1}(t) + u_m(t))\,dt \end{cases}$$

To prepare for studying the limit as $m \to \infty$ we introduce a new parameter ℓ, that will be tuned to the initial condition $u(0)$ and N and T later, and think of each well as occupying intervals of length ℓ/m positioned uniformly on the strip $0 \le x \le \ell$. To this end, put $\varepsilon = \ell/m$, and for $x = j\varepsilon$, $j = 1, \dots, m$, let $u_\varepsilon(t, x) = u_j(t)$. We also adapt conductivity by introducing the scaled parameter $\kappa = k/m^2$. Moreover, put $\kappa_c = \kappa/c(1-c)$. Then, for $x \in \{2/m, \dots, (\ell-1)/m\}$,

$$\begin{aligned} du_\varepsilon(t,x) = \frac{\kappa_c \ell^2}{2} \frac{u_\varepsilon(t, x-\varepsilon) - 2u_\varepsilon(t,x) + u_\varepsilon(t, x+\varepsilon)}{\varepsilon^2}\,dt \\ +\kappa_c \ell m(2c-1)\frac{u(t, x+\varepsilon) - u(t, x-\varepsilon)}{2\varepsilon}\,dt. \end{aligned}$$

The boundary equations attain the form

$$\frac{du_\varepsilon(t,\varepsilon)}{m} = -\frac{\Lambda(dt)}{m} + \frac{\kappa_c}{2}\left\{\ell\,\frac{u_\varepsilon(t,2\varepsilon)-u_\varepsilon(t,\varepsilon)}{\varepsilon} + m(2c-1)(u_\varepsilon(t,\varepsilon)+u_\varepsilon(t,2\varepsilon))\right\}dt$$

and

$$\frac{du_\varepsilon(t,\ell)}{m} = -\frac{\kappa_c}{2}\left\{\ell\,\frac{u_\varepsilon(t,\ell)-u_\varepsilon(t,\ell-\varepsilon)}{\varepsilon} + m(2c-1)(u_\varepsilon(t,\ell-\varepsilon)+u_\varepsilon(t,\ell))\right\}dt$$

We now consider the case of scaling the height parameter $c = c_m$ with the number of spatial compartments by putting $c_m = (1+\mu/m)/2$. With $\mu \neq 0$ and large m, this will keep the system close to the symmetric situation $c_m \sim 1/2$ but with a flux of charge at each adjacent pair of wells with magnitude of order μ/m. This gives the approximative system

$$du_\varepsilon(t,x) = -\Lambda(dt)\delta_\varepsilon(dx) + 2\kappa\ell^2\frac{u_\varepsilon(t,x-\varepsilon)-2x_\varepsilon(t,x)+u_\varepsilon(t,x+\varepsilon)}{\varepsilon^2}\,dt + 4\kappa\ell\mu\frac{u(t,x+\varepsilon)-u(t,x-\varepsilon)}{2\varepsilon}\,dt$$

with Robin type boundary conditions

$$\ell\,\frac{u_\varepsilon(t,2\varepsilon)-u_\varepsilon(t,\varepsilon)}{\varepsilon} = -2\mu u_\varepsilon(t,\varepsilon), \quad \ell\,\frac{u_\varepsilon(t,\ell-\varepsilon)-u_\varepsilon(t,\ell)}{\varepsilon} = 2\mu u_\varepsilon(t,\ell-\varepsilon).$$

Taking an informal limit as $m \to \infty$, we conclude that the relevant limiting equation is

$$du(t,x) = -\Lambda(dt)\delta_0(dx) + 2\kappa\ell^2\frac{\partial^2 u}{\partial x\partial x}(t,x)\,dt + 4\kappa\ell\mu\frac{\partial u}{\partial x}(t,x)\,dt, \quad 0 \le x \le \ell$$
$$\ell\,\frac{\partial u}{\partial x}(t,0+) = -2\mu u(t,0), \quad \ell\,\frac{\partial u}{\partial x}(t,\ell-) = 2\mu u(t,\ell), \quad u(0,x) = u_0(x).$$

Our interpretation is that $u(t,0)_{t\ge 0}$ represents the available charge of the battery and $\{u(t,x), 0 < x < \ell\}_{t\ge 0}$ represents the fluid level of a reservoir of bound charge such that $\int_{(0,\ell)} u(t,x)\,dx$ is what remains in the reservoir at time t. For simplicity we will consider the case $u_0(y) = u_0$, $y \in [0,\ell]$, for which initial charge is uniformly located on the strip $[0,\ell]$. Hence

$$v(t) = u(t,0) + \int_0^\ell u(t,x)\,dx, \quad t \ge 0, \quad v(0) = u_0 + u_0\ell,$$

defines the remaining capacity in the system as function of time.

Now we are in position to relate the model parameters u_0 and ℓ to the battery parameters N and T. For this we take the initial level of available charge to be $u_0 = N$ and the initial content of the reservoir to be $u_0\ell = T - N$. Then the total potentially available charge is $v(0) = T$ and we have $\ell = T/N - 1$.

Consider the parameters $\sigma^2 = 4\kappa\ell^2$ and $\beta = -4\kappa\ell\mu$. Let $(\xi_t)_{t\ge 0}$ denote a Brownian motion with variance parameter σ^2 and constant drift β. The initial

condition is $\xi_0 = x \in (0, \ell)$ and the paths are subject to reflecting boundaries at both end points 0 and ℓ with no loss of probability mass. Denote by $p_\ell(t, y, x)$ the corresponding transition probability density function, such that $P(\xi_t \in dx | \xi_0 = y) = p_\ell(t, y, x)\, dx$. Then the solution $u(t, x)$ of the above PDE has the representation

$$u(t,x) = \int_0^\ell u_0(y) p_\ell(t,y,x)\, dy - \int_0^t p_\ell(t-s,0,x)\, \Lambda(ds). \tag{6}$$

The reflected Brownian motion (ξ_t) arises also as a model in economics and other fields, see e.g. [11] and references in there. The transition density is given by

$$p_\ell(t,y,x) = \frac{2\mu}{\ell} \frac{e^{-2\mu x/\ell}}{1-e^{-2\mu}} + \frac{2e^{-\mu(x-y)/\ell}}{\ell} \times$$
$$\sum_{n=1}^{\infty} (\cos(\frac{n\pi x}{\ell}) - \frac{\mu}{n\pi}\sin(\frac{n\pi x}{\ell}))(\cos(\frac{n\pi y}{\ell}) - \frac{\mu}{n\pi}\sin(\frac{n\pi y}{\ell})) \frac{e^{-2\kappa(\mu^2+n^2\pi^2)t}}{1+(\mu/n\pi)^2}.$$

In particular, for the symmetric case, taking $\mu \to 0$,

$$p_\ell(t,y,x) = \frac{1}{\ell} + \frac{2}{\ell} \sum_{n=1}^{\infty} \cos(n\pi x/\ell) \cos(n\pi y/\ell)\, e^{-2\kappa n^2\pi^2 t}.$$

By (6),

$$u(t,x) = (T - N - \lambda t) \frac{2\mu}{\ell} \frac{e^{-2\mu x/\ell}}{1-e^{-2\mu}}$$
$$+4N\mu e^{-\mu x/\ell} \sum_{n=1}^{\infty} (\cos(\frac{n\pi x}{\ell}) - \frac{\mu}{n\pi}\sin(\frac{n\pi x}{\ell})) \frac{(-1)^n e^{\mu} - 1}{\mu^2 + n^2\pi^2} \frac{e^{-2\kappa(\mu^2+n^2\pi^2)t}}{1+(\mu/n\pi)^2}$$
$$-\frac{\lambda e^{-\mu x/\ell}}{\kappa\ell} \sum_{n=1}^{\infty} (\cos(\frac{n\pi x}{\ell}) - \frac{\mu}{n\pi}\sin(\frac{n\pi x}{\ell})) \frac{n^2\pi^2(1-e^{-2\kappa(\mu^2+n^2\pi^2)t})}{(\mu^2+n^2\pi^2)^2}.$$

The remaining capacity is

$$v(t) = u(t,0) + \int_0^\ell u(t,x)\, dx = u(t,0) + T - N - \lambda t.$$

Furthermore,

$$u(t,0) = (T - N - \lambda t) \frac{1}{\ell} \frac{2\mu}{1-e^{-2\mu}} - \frac{\lambda}{\kappa\ell} \sum_{n=1}^{\infty} \frac{n^2\pi^2(1-e^{-2\kappa(\mu^2+n^2\pi^2)t})}{(\mu^2+n^2\pi^2)^2}$$
$$+4N\mu \sum_{n=1}^{\infty} ((-1)^n e^{\mu} - 1) \frac{n^2\pi^2\, e^{-2\kappa(\mu^2+n^2\pi^2)t}}{(\mu^2+n^2\pi^2)^2}.$$

This shows that the quantities $(v, u) = (v(t), u(t, 0))$ form an autonomous system such that the relation between v and $u = u(v)$ is given by

$$u = (v-u)\frac{1}{\ell}\frac{2\mu}{1-e^{-2\mu}} - \frac{\lambda}{\kappa\ell}\sum_{n=1}^{\infty}\frac{n^2\pi^2(1-e^{-2\kappa(\mu^2+n^2\pi^2)(T-N-v+u)/\lambda})}{(\mu^2+n^2\pi^2)^2}$$
$$+4N\mu\sum_{n=1}^{\infty}((-1)^n e^{\mu}-1)\frac{n^2\pi^2\, e^{-2\kappa(\mu^2+n^2\pi^2)(T-N-v+u)/\lambda}}{(\mu^2+n^2\pi^2)^2}.$$

As $\mu \to 0$,

$$u = \frac{1}{\ell}(v-u) - \frac{\lambda}{\kappa\ell}\sum_{n=1}^{\infty}\frac{1}{n^2\pi^2}(1-e^{-2\kappa n^2\pi^2(T-N-v+u)/\lambda}).$$

The important conclusion now is that we have obtained closed phase plane representations of nominal and theoretical capacity also for the spatial model, at least for constant load. Thus, performance can be studied just as for the two-well model. Figure 5 displays typical discharge profiles of the driftless spatial version of the kinetic battery model. The graphs are very similar to those for the basic model in Figure 2. Figure 6 shows the effect of adding drift μ to the model. With the same N and T as previously and for $\lambda/\kappa = 2000$, three discharge profile curves are plotted with $\mu = -0.5$, $\mu = 0$ and $\mu = 0.5$. Clearly, negative drift lowers the delivered capacity whereas positive drift $\mu > 0$ improves the utilization of bound charge.

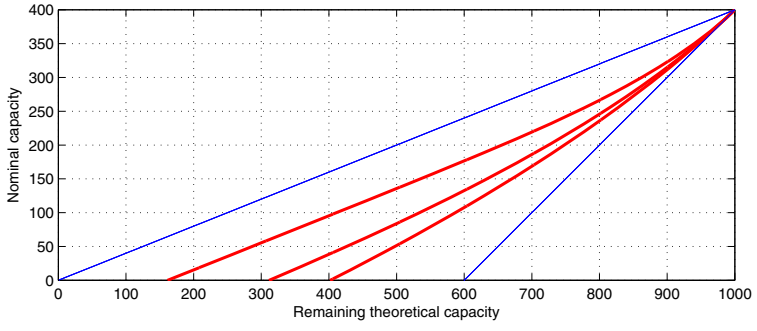


Fig. 5. Discharge profiles of the spatial kinetic battery model, $N = 400$, $T = 1000$, from left to right: $\lambda/\kappa = 1000, 2000, 3000$

Spatial Version of the Kinetic-Diffusive Battery Model. Finally we consider a spatial extension of the proposed kinetic-diffusive model with symmetric kinetic dynamics $c = 1/2$ but diffusive effect governed by $p \geq 0$. Here, the total conductivity has been partitioned into two mechanisms, see [8]. First the basic flow of charge caused by the height difference between wells, secondly a drift

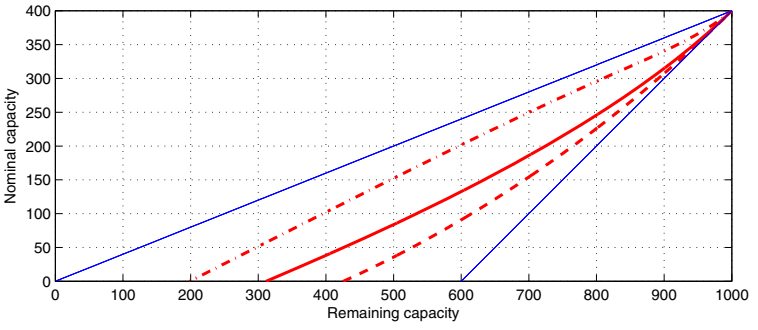


Fig. 6. Discharge profiles of the spatial kinetic battery model with drift, $N = 400$, $T = 1000$, $\kappa = 1$, $\lambda = 2000$, $\mu = -0.5, 0, 0.5$

under the action of the discharge load which affects the bound charge. The parameter p controls the balance of these two contributions to the overall flux. Putting $q = 1 - p$,

$$\begin{cases} du_1(t) = -\Lambda(dt) + 2kq(u_2(t) - u_1(t))\,dt - 4kp(N - u_2(t))\,dt \\ du_2(t) = 2kq(u_1(t) - 2u_2(t) + u_3(t))\,dt + 4kp(u_3(t) - u_2(t))\,dt \\ \qquad \vdots \\ du_{m-1}(t) = 2kq(u_{m-2}(t) - 2u_{m-1}(t) + u_m(t))\,dt + 4kp(u_m(t) - u_{m-1}(t))\,dt \\ du_m(t) = -2kq(u_m(t) - u_{m-1}(t)) + 4kp(N - u_m(t)) \end{cases}$$

Again, we place m wells of width ℓ/m on the interval $0 \le x \le \ell$. With $\varepsilon = \ell/m$ and $u_\varepsilon(t,x) = u_j(t)$ for $x = j\varepsilon$, $j = 1, \dots, m$, and with scaled parameters $\kappa = k/m^2 > 0$ and $\rho = mp > 0$, we find for for $x \in \{2/m, \dots, (\ell-1)/m\}$,

$$du_\varepsilon(t,x) = 2\kappa\ell^2(1-\rho/m)\frac{u_\varepsilon(t,x-\varepsilon) - 2u_\varepsilon(t,x) + u_\varepsilon(t,x+\varepsilon)}{\varepsilon^2}\,dt$$
$$+4\kappa\ell\rho\frac{u(t,x+\varepsilon) - u(t,x)}{\varepsilon}\,dt.$$

The boundary equations attain the form

$$\frac{du_\varepsilon(t,\varepsilon)}{m} = -\frac{\Lambda(dt)}{m} + 2\kappa\left\{\ell(1-\rho/m)\,\frac{u_\varepsilon(t,2\varepsilon) - u_\varepsilon(t,\varepsilon)}{\varepsilon} - 2\rho(N - u_\varepsilon(t,2\varepsilon))\right\}dt$$

and

$$\frac{du_\varepsilon(t,\ell)}{m} = -2\kappa\left\{\ell(1-\rho/m)\,\frac{u_\varepsilon(t,\ell) - u_\varepsilon(t,\ell-\varepsilon)}{\varepsilon} - 2\rho(N - u_\varepsilon(t,\ell-\varepsilon))\right\}dt.$$

Considering a limit for large m, this yields

$$du(t,x) = -\Lambda(dt)\delta_0(dx) + 2\kappa\ell^2\frac{\partial^2 u}{\partial x \partial x}(t,x)\,dt + 4\kappa\ell\rho\frac{\partial u}{\partial x}(t,x)\,dt, \quad 0 \le x \le \ell$$
$$\ell\frac{\partial u}{\partial x}(t,0+) = 2\rho(N - u(t,0)), \quad \ell\frac{\partial u}{\partial x}(t,\ell-) = -2\rho(N - u(t,\ell)),$$

with initial condition $u(0,x) = u_0(x)$. In comparison to the previous case where a drift μ was created by keeping the height parameter c asymptotically of the order $c = 1/2 + \mu/2m$, in this case we obtain the same equation with ρ replacing μ but other boundary conditions still of the generalized Robin type.

References

1. Chiasserini, C.F., Rao, R.R.: Energy efficient battery management. IEEE J. Sel. Areas in Communication 19(7) (2001)
2. Danilov, D., Niessen, R.A.H., Notten, P.H.L.: Modeling All-Solid-State Li-Ion Batteries. Journal of the Electrochemical Society 158(3), A215–A222 (2011)
3. Jongerden, M.R., Haverkort, B.R., Bohnenkamp, H.C., Katoen, J.P.: Maximizing System Lifetime by Battery Scheduling. In: Proceedings of the 39th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE Computer Society Press, Los Alamitos (2009)
4. Jongerden, M.R., Haverkort, B.R.: Which battery model to use? IEEE/IET Software 3(6), 445–457 (2009)
5. Jongerden, M.R., Haverkort, B.R.: Lifetime improvement by battery scheduling. In: 27th Annual UK Performance Engineering Workshop, UKPEW (2011)
6. Manwell, J., McGowen, J.: Lead acid battery storage model for hybrid energy systems. Solar Energy 50, 399–405 (1993)
7. Manwell, J., McGowen, J.: Extension of the kinetic battery model for wind/hybrid power systems. In: Proc. fifth European Wind Energy Association Conf., pp. 284–289 (1994)
8. Mehrer, H.: Diffusion in Solids: Fundamentals, Methods, Materials. In: Diffusion-Controlled Processes. Springer, Heidelberg (2007)
9. Rao, V., Singhal, G., Kumar, A., Navet, N.: Battery model for embedded systems. In: 18th International Conference on VLSI Design (2005)
10. Rakhmatov, D., Vrudhula, S.: Energy Management for Battery-Powered Embedded Systems. ACM Transactions on Embedded Computing Systems 2(3), 277–324 (2003)
11. Veestraeten, D.: The conditional probability density function for a reflected Brownian motion. Computational Economics 24, 185–207 (2004)

Branching Processes, the Max-Plus Algebra and Network Calculus

Eitan Altman[1] and Dieter Fiems[2]

[1] Maestro group, INRIA
2004 Route des Lucioles, 06902 Sophia Antipolis Cedex, France
[2] Department TELIN, Ghent University
St-Pietersnieuwstraat 41, 9000 Gent, Belgium

Abstract. Branching processes can describe the dynamics of various queueing systems, peer-to-peer systems, delay tolerant networks, etc. In this paper we study the basic stochastic recursion of multitype branching processes, but in two non-standard contexts. First, we consider this recursion in the max-plus algebra where branching corresponds to finding the maximal offspring of the current generation. Secondly, we consider network-calculus-type deterministic bounds as introduced by Cruz, which we extend to handle branching-type processes. The paper provides both qualitative and quantitative results and introduces various applications of (max-plus) branching processes in queueing theory.

Keywords: Stochastic recursive equations, Branching processes, Max-plus algebra, Network calculus.

1 Introduction

Branching processes model the dynamics of populations over successive generations, each member of some generation independently producing offspring in the next generation in accordance with a given probability distribution. Originating from the nobility and family extinction problem, branching process theory has been applied in diverse fields including computer science and networking.

Branching processes have been frequently identified in queueing theory and the connection between branching processes and queueing theory is well established. Already in 1942, Borel showed that the numbers of customers in a busy period of an M/G/1 queue corresponds to the number of generations till extinction of a Galton-Watson branching process [8,19]. Similarly, Crump-Mode-Jagers branching processes describe the dynamics of processor sharing queues [16], whereas multitype branching processes with immigration have been used to study retrial queues [17] and polling systems [23].

Applications of branching processes in networking are not limited to queueing theory. In [22], a multitype branching process is studied to determine the maximum stable throughput of tree algorithms with free access. Stability of the tree algorithm corresponds to criticality of the associated multitype Galton-Watson

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 255–270, 2012.

branching process. Multitype branching processes with migration can also capture the dynamics of distributing a file into a delay tolerant network with the single-hop forwarding paradigm [11]. In addition, peer-to-peer networks also provide many interesting applications of branching processes [12,13,21]. For example, Leskala et Al. [21] study interacting branching processes in the context of file sharing networks.

In this paper we study the basic stochastic recursion of multitype branching processes, but in two non-standard contexts. The first part of the paper introduces problems that yield similar recursions but where summation and multiplication are understood as being in another possible algebra than the standard one. In particular, we define and characterize branching processes in the max-plus algebra, for both discrete and continuous state spaces. As shown further, we obtain general characterizations of the stationary behaviour of the single-type branching process with independent migration in the max-plus algebra. In addition, multitype branching processes in the max-plus algebra are introduced and sufficient conditions are proven for stability of these processes in the presence of stationary ergodic migration.

We then introduce a deterministic framework for studying branching processes. A deterministic view on branching allows for focussing on worst-case behaviour rather then average behaviour, as advocated by literature on network calculus. When designing a network so as to meet strict bounds on quality of service, then standard probabilistic descriptions of input and output processes are no longer relevant; one has to come up with a design adapted to the worst case of the input process. Much attention has been given to producing a network calculus in which each network element has a transfer function: it maps a given description of its input process to a similar description of the output process. A complete mapping of this type allows one to dimension buffers in the network that guarantee that there are no losses [10,20]. A more complex situation arises when the network includes feedback. The outputs are no longer functions of exogenous traffic. Computing tight bounds for feedback systems proves to be much harder. A well known example of such a system is given in part ii of [10]; the bound given there is indeed not tight and has later been improved. Other examples of explicit or implicit feedback for which obtaining tight bounds is not a simple task are presented in [2,3] in the context of polling systems.

The deterministic framework for branching processes closely relates to arrival processes in network calculus. It concerns processes that are shaped at the entrance of the network using RED (Random Early Discard) or leaky bucket mechanisms. These processes are characterized by a bound ρ on the average rate as well as some bounded σ on the burstiness. More precisely, the output process is "$\sigma - \rho$ constrained", i.e. for any interval $[s, t)$, the output process from such a buffer is bounded by $\rho(t - s) + \sigma$. R. L. Cruz showed in [10] that standard network elements preserve this type of bound. Moreover, they imply uniform bounds on the amount of workload in the network, which allows one to dimension buffer sizes so as to guarantee no overflow as long as the input processes are $\sigma - \rho$ constrained. In this paper we study some form of feedback in which

the arrival process itself depends on the output process. We show that if this dependence can be described using $\sigma - \rho$ type bounds, one still obtains uniform bounds on the workload in the system. We show that the feedback mechanism is of the same type that is used to define branching processes. Our results thus provide a motivation for investigating a deterministic type of branching processes. Finally, as for ordinary network calculus, we relax the deterministic bounds by introducing elements of stochastic network calculus [18].

2 Branching in the Standard Algebra

We start by recalling some basic characterization of branching in the standard algebra. The standard branching process is defined as follows. Let Y_n be the number of individuals in generation n. Starting with a fixed Y_0, we define recursively

$$Y_{n+1} = \sum_{i=1}^{Y_n} \xi_n^{(i)}$$

where $\xi_n^{(i)}$ are independent and identically distributed (i.i.d.) random variables taking non-negative integer values. Define $A_n(m) := \sum_{i=1}^{m} \xi_n^{(i)}$, we can then rewrite the above as

$$Y_{n+1} = A_n(Y_n)\,. \tag{1}$$

Given the definition of the branching process above, branching processes with immigration are defined through the recursion,

$$Y_{n+1} = A_n(Y_n) + B_n\,, \tag{2}$$

B_n being referred to as the immigration component.

Next we recall the definition of branching process on a continuous state space. We note that A_n is nonnegative and has a divisibility property: for any nonnegative integers m, m_1 and m_2 such that $m_1 + m_2 = m$, and for any n,

$$A_n(m) = A_n^{(1)}(m_1) + A_n^{(2)}(m_2)$$

where for each n, $A_n^{(1)}$ and $A_n^{(2)}$ are independent random processes, both with the same distribution as A_n. We take this property, together with the nonnegativity of A_n as the basis to define the continuous state branching processes. Noting that these properties are satisfied by Lévy processes, we define a continuous state branching process as one satisfying (1) where A_n is a nonnegative Lévy process (a subordinator).

Example 1. Consider an $M/G/1$ queue with gated repeated vacations: the arrivals are modeled by a Poisson process and the service and vacation times constitute sequences of i.i.d. random variables. Each time the server returns from vacation, it closes a gate, and the next busy period starts. The busy period consists of the service times requested by all those that are present when the gate

is closed. When the busy period ends then the server leaves on vacation. The period that consists of a busy period followed by a vacation is called a cycle. Let Y_n be the number of customers present at the beginning of the nth cycle. Let $\xi_n^{(i)}$ denote the number of customer arrivals during the service time of the ith customer among those present in the queue at the beginning of cycle n and let B_n be the number of arrivals during the nth vacation. With these definitions it easily follows that Y_n satisfies (2).

Example 2. Consider the model of the previous example and Let C_n be the nth cycle time. Moreover, let $A_n(C_n)$ denote the workload that arrives during C_n (i.e. the time to serve all those that arrive during the nth cycle time) and B_n denote the length of the $n+1$st vacation. Then again (2) holds (thereby replacing Y_n by C_n).

3 Branching in the Max-Plus Algebra

In the max-plus algebra, summation corresponds to max, and multiplication to summation. Hence, we define the (single-type) branching process in the max-plus as follows,

$$Y_{n+1} = \bigoplus_{i=1}^{Y_n} \xi_n^{(i)} ,$$

where $\bigoplus$ stands for maximization and where $\{\xi_n^{(i)}\}$ constitutes a doubly indexed sequence of i.i.d. random variables taking non-negative integer values. Thus (1) still holds but this time with,

$$A_n(m) := \bigoplus_{i=1}^{m} \xi_n^{(i)} . \tag{3}$$

When considering immigration we shall consider two forms. The first form is,

$$Y_{n+1} = A_n(Y_n) \otimes B_n , \tag{4}$$

with $\otimes$ denoting summation in the standard algebra, such that the former expression can be written in the standard algebra as,

$$\max_{i=1,\ldots,Y_n} \xi_n^{(i)} + B_n .$$

Notice that we here replaced only the branching part by its max-plus version. The second form of immigration we consider is,

$$Y_{n+1} = A_n(Y_n) \oplus B_n , \tag{5}$$

which can be written in the standard algebra as,

$$Y_{n+1} = \max\left(\max_{i=1}^{Y_n} \xi_n^{(i)}, B_n \right) .$$

To define continuous branching in the max-plus algebra, we relate max-plus branching with a continuous state space to Lévy processes, just as is done for ordinary branching. The max-plus equivalent of branching in continuous state space is defined as the maximum step of a (non-decreasing) Lévy process $L_n(t)$ over an interval of length y,

$$A_n(y) = \sup_{0 \le t < y} dL_n(t)\,. \tag{6}$$

Notice that the divisibility of the branching process now holds in the max-plus algebra. For any non-negative real values y, y_1 and y_2 such that $y_1 + y_2 = y$, and for any n, we now have,

$$A_n(y) = A_n^{(1)}(y_1) \oplus A_n^{(2)}(y_2)\,,$$

A_n being defined in either (3) or (6) (in the former case, y_1 and y_2 are positive integers).

We now consider some queueing systems whose dynamics can be described by the max-plus branching processes introduced above.

Example 3. Consider a discrete-time infinite-server queue with exactly one arrival at each time slot. We consider gated service and general vacations: when the nth vacation ends, a gate is closed and the $n+1st$ busy period starts. Let Y_n denote the number of customers present when the nth busy period starts. All customers that are present are served in parallel, their service times being i.i.d. and the next vacation starts when all services are completed. Let $\xi_n^{(i)}$ be the service time of the ith customer served during the nth busy period and let B_n denote the length of the nth vacation. As there is a single arrival in each slot, the sequence Y_n satisfies (4) with A_n as defined in (3).

Example 4. Consider the same model as in previous example and let $\widehat{Y}_n$ be the number of customers at the end of the nth busy period. Retaining the notation introduced in example 3, $\widehat{Y}_n$ relates to Y_n as

$$\widehat{Y}_n = A_n(Y_n) = Y_{n+1} - B_n$$

such that,

$$\widehat{Y}_{n+1} = A_{n+1}(\widehat{Y}_n) \oplus \widehat{A}_{n+1}(B_n)\,,$$

by the divisibility of the max-plus branching process. Here $\widehat{A}_n$ is an independent copy of A_n such that $\widehat{A}_{n+1}(B_n)$ represents the maximal service time of a customer that arrives during the nth vacation. This is a branching process in the max-plus algebra of the same type as (5), the migration process being $Q_n \doteq \widehat{A}_{n+1}(B_n)$.

Example 5. We consider the same setting of the previous examples except for the arrival process. The ith arrival occurs at time τ^i and brings a service requirement of $\xi^{(i)}$ which need not be integer valued. The arriving workload is then given by

$$V(t) = \sum_i \xi^{(i)} 1\{0 \le \tau^i \le t\}\,.$$

If the arrival process is Poisson, and the service times are i.i.d. and independent of the arrival times, then $V(t)$ is a non-decreasing Lévy process. The independent increments property of Lévy processes allows us to introduce a sequence of i.i.d. Lévy processes $V_n(t)$, distributed as $V(t)$ which denote the arriving workload during the nth cycle. As before, the $n+1$st busy period is the maximum service time of all those that arrived during the nth cycle,

$$A_n(C_n) := \sup_{0 \le t \le C_n} dV_n(t)\,.$$

Hence A_n is a max-branching process, see (6). As the $n+1$st cycle time equals the sum of the largest service time of a customer that arrived during the preceding cycle and the vacation length B_n, consecutive cycle times relate as in (4), B_n being the length of the nth vacation as before.

4 Solution

4.1 Discrete State Space

We first consider max-plus branching with a discrete state space. For a discrete random variable r, its distribution function and probability mass function are denoted by $F_r(i) = \Pr[r \le i]$ and $p_r(i) = \Pr[r = i]$, respectively, whereas $r^*(z) = \mathrm{E}[z^r]$ denotes its probability generating function.

We first solve (5). In this case, $Y_{n+1} \le i$ if $B_n \le i$ as well as all $\xi_n^{(j)}$ for $j = 1, \ldots, Y_n$; see (3). Hence, we have,

$$\Pr(Y_{n+1} \le i) = \mathrm{E}[\Pr(Y_{n+1} \le i | Y_n, B_n)] = \mathrm{E}([F_\xi(i)]^{Y_n} 1\{B_n \le i\})\,.$$

Here $1\{\cdot\}$ denotes the indicator function. As the consecutive B_n are i.i.d. and independent of A_n, this gives

$$\Pr(Y_{n+1} \le i) = Y_n^*(F_\xi(i)) \Pr(B_n \le i)\,.$$

Let π be the steady state probability vector of Y, $\pi(j) = \Pr[Y = j]$, then we get the following set of equations for π:

$$\sum_{j=0}^{i} \pi(j) = \sum_{j=0}^{\infty} \pi(j)[F_\xi(i)]^j F_B(i)\,. \tag{7}$$

Now assume that ξ_n and B_n have finite support, they are both bounded by an integer L. This implies that (7) consists of a set of at most $L+1$ linear equations which allows us to solve for the unknowns (together with the equation $\sum_i \pi(i) = 1$).

We now solve (4) for the discrete state space. In this case, $Y_{n+1} \le i$ if all $\xi_n^{(j)} \le i - B_n$ for $j = 1, \ldots, Y_n$; see (3). Hence, we have,

$$\Pr(Y_{n+1} \le i) = \mathrm{E}[\Pr(Y_{n+1} \le i | Y_n, B_n)] = \sum_{\ell=0}^{i} \mathrm{E}([F_\xi(i-\ell)]^{Y_n} 1\{B_n = \ell\})$$
$$= \sum_{\ell=0}^{i} Y_n^*(F_\xi(i-\ell))\, p_B(\ell)\,.$$

As before, let π be the steady state probability vector of Y, $\pi(j) = \Pr[Y = j]$, then we get the following set of equations for π,

$$\sum_{j=0}^{i} \pi(j) = \sum_{j=0}^{\infty} \sum_{\ell=0}^{i} \pi(j)[F_\xi(i-\ell)]^j\, p_B(\ell)$$

Again assuming that ξ_n and B_n have finite support, let L denote the common upper bound, the former set (7) consists of at most L linear equations which allows us to solve for the unknowns (together with the normalization condition $\sum_i \pi(i) = 1$).

4.2 Continuous State Space

We now consider max-plus branching with a continuous state space. For a continuous random variable r, let $F_r(x) = \Pr[r \le x]$ denote its distribution function and, with some abuse of notation, let $r^*(\zeta) = \mathrm{E}[\exp(-\zeta r)]$ denote its Laplace-Stieltjes transform.

We first consider (5). By conditioning on the Y_n and B_n, we find,

$$\Pr(Y_{n+1} \le x) = \mathrm{E}[\Pr(Y_{n+1} \le x | Y_n, B_n)] = \mathrm{E}(\exp(\lambda Y_n(\sigma(x) - 1)) 1\{B_n \le x\})$$
$$= Y_n^*(\lambda(1 - \sigma(x)))F_B(x)\,,$$

where $\lambda = \Pi[0, \infty)$ and $\sigma(x) = \Pi[0, x)/\lambda$ relate to the Lévy measure Π of L_n. In view of the former expression, we then obtain the following integral equation,

$$Y_{n+1}^*(\zeta) = \int_0^\infty \exp(-\zeta x) d(Y_n^*(\lambda(1 - \sigma(x)))F_B(x))\,.$$

Therefore, the Laplace-Stieltjes transform of the steady state distribution of Y satisfies,

$$Y^*(\zeta) = \zeta \int_0^\infty Y^*(\lambda(1 - \sigma(x)))F_B(x)e^{-\zeta x} dx - Y^*(\lambda)F_B(0)\,.$$

We now consider (4). By conditioning on the Y_n and B_n, we find,

$$\Pr(Y_{n+1} \le x) = \mathrm{E}[\Pr(Y_{n+1} \le x | Y_n, B_n)]$$
$$= \int_0^x \mathrm{E}(\exp(\lambda Y_n(\sigma(x-y) - 1)))F_B(dy)$$
$$= \int_0^x Y_n^*(\lambda(1 - \sigma(x-y)))F_B(dy)\,,$$

such that the Laplace-Stieltjes transform of the steady-state distribution satisfies,

$$Y^*(\zeta) = \zeta \int_0^\infty \int_0^x Y^*(\lambda(1-\sigma(x-y)))e^{-\zeta x} F_B(dy)dx\,.$$

In general, no easy solution for these integral equations is available. One can nevertheless resort to numerical solution techniques for integral equations, see e.g. [15].

5 The Multitype Branching

We now turn to stability conditions for max-plus branching processes. We do this in a more general setting: (i) we consider vector-valued processes and (ii) we consider all types of processes which have the same divisibility property as branching processes. In particular, consider the $\mathbb{R}_+^K$ valued process $\{Y_n\}$ and denote the ith entry of Y_n by Y_n^i, $i = 1, ..., K$. The process Y_n satisfies the following equation in vector form:

$$Y_{n+1} = A_n(Y_n) + B_n. \tag{8}$$

The K-dimensional column vector B_n is a stationary ergodic stochastic process whose entries B_n^i, $i = 1, ..., K$ are in subsets of the nonnegative real numbers.

For each n, A_n are non-negative vector valued random fields that are non-decreasing in their arguments. A_n are i.i.d. with respect to n, and $A_n(0) = 0$.

We characterize max-branching processes by their divisibility property. That is, we assume that A_n satisfies the following. If for some k, $y = y^0 + y^1 + ... + y^k$ where y^m are vectors, then $A_n(y)$ can be represented as

$$A_n(y) = \bigoplus_{i=0}^{k} \widehat{A}_n^{(i)}(y^i) \tag{9}$$

where $\{\widehat{A}_n^{(i)}\}_{i=0,1,2,...,k}$ are identically distributed with the same distribution as $A_n(\cdot)$. In particular, for any sequence $k(n)$, $\{\widehat{A}_n^{(k(n))}\}_n$ are independent.

Remark 1. For a given n, we do not assume independence of the random variables $\{\widehat{A}_n^{(i)}\}_{i=0,1,2,...}$. In the case of ordinary multitype branching processes, this leads to a unified framework of linear difference equations and branching processes. In the case of max-branching considered here, the correspondence with max-plus-linear difference equations does not hold. Nevertheless, independence of $\{\widehat{A}_n^{(i)}\}_{i=0,1,2,...}$ is not required for proving stability and is therefore not assumed.

5.1 Examples

We first introduce some processes that satisfy the divisibility property.

Example 6. We define the discrete multitype branching process $A(y)$ as follows. Let $\xi^{(k)}(n)$, $k = 1, 2, 3, ...,$ $n = 1, 2, 3, ...$ be a doubly-indexed sequence of i.i.d. random $K \times K$ matrices. The elements of these matrices take values in the nonnegative integers. Moreover, assume that for any $\ell = 1, 2, 3, ...,$ $\ell' = 1, 2, 3, ...,$ $k = 1, ..., K$, $i = 1, ..., K$, $m = 1, ..., K$, $j = 1, ..., K$ and $m \neq k$, $\xi_{ki}^{(\ell)}$ and $\xi_{mj}^{(\ell')}$ are independent.

Let y_j be the jth element of the vector y, the ith element of the column vector $A(y)$ is given by

$$[A(y)]_i = \bigoplus_{j=1}^{K} \bigoplus_{k=1}^{y_j} \xi_{ji}^{(k)} \,. \tag{10}$$

One easily verifies that the divisibility property holds for this process.

Example 7. As for the single-type max-branching, we express the continuous multitype max-branching in terms of Lévy processes. To this end, let $L_n(y), y \in \mathbb{R}_+^K$ be an additive Lévy field. That is, we assume that $L(y)$ can be decomposed into the sum of K independent $\mathbb{R}_+^K$ valued Lévy processes,

$$L(y) = \sum_{i=1}^{K} L_i(y_i) \,,$$

y_i being the ith element of the vector y as before. The jth element of the continuous multitype max-branching process $A(y)$ is then defined as follows,

$$[A(y)]_j = \bigoplus_{i=1}^{K} d[L_i(y_i)]_j \,,$$

where $[L_i(y_i)]_j$ is the jth element of $L_i(y_i)$. Again, one easily verifies that the divisibility property holds for this process.

5.2 Stability Conditions

We shall understand below $\bigotimes_{i=n}^{k} A_i(x) = x$ whenever $k < n$, and $\bigotimes_{i=n}^{k} A_i(x) = A_k(A_{k-1}(...(A_n(x))..))$ whenever $k > n$.

In the remainder, let $\|x\|$ denote the max-norm in $\mathbb{R}^K$ and, with some abuse of notation, let $\|A_n\|$ denote the corresponding operator norm,

$$\|A_n\| = \inf\{c \geq 0 : \|A_n(y)\| \leq c\|y\|, \forall y \in \mathbb{R}^K\} \,.$$

Let $\mathcal{A} \doteq \mathrm{E}[\|A_0\|]$. Then, we have $A_n(y) \leq \|A_0\|\|y\|$, almost surely such that $\mathrm{E}[A_n(y)] \leq \mathcal{A}\|y\|$. By the independence of the consecutive branching processes, this further implies for $j > 1$,

$$\mathrm{E}\left[\left\|\left(\bigotimes_{i=1}^{j} A_i\right)(y)\right\|\right] \leq \mathcal{A}^j \|y\| \,. \tag{11}$$

We now introduce our stability conditions.

Theorem 1. *Let Y_n satisfy (8), with A_n satisfying the divisibility property (9) and B_n stationary ergodic. We then have the following.*

(i) For $n > 0$, Y_n can be written in the form

$$Y_n = \widetilde{Y}_n + \left(\bigotimes_{i=0}^{n-1} \widehat{A}_i^{(0)} \right) (Y_0) \tag{12}$$

where

$$\widetilde{Y}_n = \sum_{j=0}^{n-1} \left(\bigotimes_{i=n-j}^{n-1} \widehat{A}_i^{(n-j)} \right) (B_{n-j-1}) \tag{13}$$

is the solution of (8) with initial condition $Y_0 = 0$.

(ii) For $\mathcal{A} < 1$ and $\mathrm{E}[\|B_0\|] < \infty$, there is a unique stationary solution Y_n^* of (8), distributed like,

$$Y_n^* =_d \sum_{j=0}^{\infty} \left(\bigotimes_{i=n-j}^{n-1} \widehat{A}_i^{(n-j)} \right) (B_{n-j-1}), \qquad n \in Z. \tag{14}$$

The sum on the right side of (14) converges absolutely almost surely. Furthermore, one can construct a probability space such that $\lim_{n\to\infty} \|Y_n - Y_n^*\| = 0$, almost surely, for any initial value Y_0.

Proof. (13) is obtained by iterating (8). Now, define the following set of stochastic recursions on the same probability space as Y_n:

$$Y_{n+1}^{[\ell]} = A_n(Y_n^{[\ell]}) + B_n, \quad m \geq -\ell, \qquad Y_{-\ell}^{[\ell]} = 0. \tag{15}$$

For each $n \geq 0$, $Y_n^{[\ell]}$ is monotonically non-decreasing in ℓ so that the limit $Y_n^* = \lim_{n\to\infty} Y_n^{[\ell]}$ is well defined. Since this is measurable on the tail σ-algebra generated by the stationary ergodic sequence $\{A_n, B_n\}$, it is either finite almost surely or infinite almost surely. The last possibility is excluded since it follows by induction that for every $\ell \geq 0$ and $n \geq -\ell$ that $\mathrm{E}[\|Y_n^{[\ell]}\|] \leq (1-\mathcal{A})^{-1}\,\mathrm{E}[\|B_0\|]$, and hence $\mathrm{E}[\|Y_n^*\|] \leq (1-\mathcal{A})^{-1}\,\mathrm{E}[\|B_0\|]$, which is finite.

By the definition of $\widehat{A}_n^{(i)}$ and by (11), we have

$$E\left[\left\| \left(\bigotimes_{i=1}^{j} \widehat{A}_i^{(0)} \right) (y) \right\| \right] = \mathcal{A}^j \|y\|,$$

which converges to zero since $\mathcal{A} < 1$. Since

$$\left\| \left(\bigotimes_{i=1}^{j} \widehat{A}_i^{(0)} \right) (y) \right\|$$

is non-negative, it then follows from Fatou's Lemma that it converges to zero almost surely. Finally, this implies that the difference

$$Y_n - Y_n^* = \left(\bigotimes_{i=1}^{j} \widehat{A}_i^{(0)}\right)(Y_0) - \left(\bigotimes_{i=1}^{j} \widehat{A}_i^{(0)}\right)(Y_0^*)$$

converges to 0 almost surely. This implies also the uniqueness of the stationary regime.

Remark 2. Recall that two forms of immigration were studied in section 3. The stability conditions of Theorem 1 also hold in the case:

$$Y_{n+1} = A_n(Y_n) \oplus B_n \,,$$

A_n and B_n as defined in the current section. To verify this, note that the inequality $\mathrm{E}[\|Y_n^{[\ell]}\|] \leq (1-\mathcal{A})^{-1}\,\mathrm{E}[\|B_0\|]$ is also valid for this modified recursion. The rest of the proof remains unaltered.

6 Deterministic Cruz Type Branching

We now return to ordinary branching processes and study these by means of a Cruz-type network calculus. Recall the following definition of an arrival curve in (deterministic) network calculus.

Definition 1. *An arrival process is said to satisfy the (σ, ρ) constraints for some constant ρ and σ, if it satisfies for any interval $[s,t]$, $t \geq s$:*

$$A[s,t] \doteq A(t) - A(s) \leq \rho(t-s) + \sigma \,.$$

In order to apply network calculus for branching processes, we first show how a single arrival process can be identified for a standard discrete branching processes. That is, the whole branching process can be derived from this single arrival process. We shall apply the same type of derivation to an arrival process that satisfies Cruz-type constraints and obtain a new recursive characterization of the branching process. We then study the properties of the resulting process.

Consider a discrete-time, one-dimensional branching process given by

$$y_{n+1} = \sum_{i=1}^{y_n} \xi_i^{(n)} + B_n$$

where $\xi_i^{(n)}$ are i.i.d. random variables taking values nonnegative integer numbers.

This branching process is driven by an immigration process B_n and by an infinite set $\xi^{(n)}$ of driving sequences. In making the relation between Cruz-type processes and a branching type structure, the immigration term will not play an important role, and we shall replace it for simplicity by a constant $B_n = B$. Our extension of the Cruz framework is to replace the driving processes $\xi^{(n)}$ by a single $\sigma - \rho$ constrained arrival process.

More generally, we shall define below the arrival processes for processes that satisfy the recursion

$$y_{n+1} = A_n(y_n) + B_n$$

where, in the case of standard discrete branching, we have

$$A_n(y_n) = \sum_{i=1}^{y_n} \xi_i^{(n)} \tag{16}$$

where $\xi_i^{(n)}$ are i.i.d. random variables taking values in the nonnegative integers.

Definition 2. *Let A be a monotone nonnegative random function from $\mathbb{R}$ to $\mathbb{R}$. We call it an arrival generator process (AGP).*

Given an AGP A and some t_0, we define $A_1(y)$ as $A_1(y) = A[t_0, t_0 + y]$, for $0 \le y \le y_1$ and define $t_1 = t_0 + y_1$. We then recursively define $t_n = y_n + t_{n-1}$ and for $n > 1$,

$$A_n(y) = A[t_n, t_n + y]. \tag{17}$$

where $0 \le y \le y_{n+1}$. Thus for a given AGP A, we obtain a unique sequence A_n of arrival processes. Conversely, assume that the sequence A_n is given, then (17) defines uniquely the AGP A.

Example 8. In the case of standard discrete branching, y is discrete and the AGP A is the counting function of a single infinite i.i.d. sequence ζ_n,

$$A(y) = \sum_{n=1}^{y} \zeta_n \,,$$

where ζ_n have the same distribution as $\xi_i^{(n)}$. It is now easy to check that with the definition (17), $A_n(y)$ have the same distribution as those given by (16), and in particular, the consecutive A_n are i.i.d.

Example 9. This way of describing a branching process easily extends to branching processes with a continuous state space. In particular, the AGP is now a subordinator A. The construction above then ensures that consecutive A_n are i.i.d. by the independent increment property of Lévy processes.

We now assume that the AGP is (σ, ρ) constraint which implies that all A_n are (σ, ρ) constraint as well. Before proceeding to our main results, we note that if each A_n is (σ, ρ) constraint, then the following bound is obtained by applying the recursion directly.

$$\begin{aligned} y_{n+1} = A_{n+1}(y_n) + B &\le \rho y_n + \sigma + B \le \rho^2 y_{n-1} + \rho(\sigma + B) + \sigma + B \\ &\le \ldots \le \rho^n y_1 + \frac{1 - \rho^{n+1}}{1 - \rho}(\sigma + B) \end{aligned} \tag{18}$$

We shall be mainly interested in the case $\rho < 1$ for which we get the following uniform bound,

$$y_n \le y_1 + \frac{\sigma + B}{1 - \rho} \tag{19}$$

Finding tighter bounds is the subject of the following section.

7 Bounds on Branching Process

We use the construction of the process given in the preceding section, based on a AGP A. Define

$$\sigma_{s,t} = A[s,t] - \rho(t-s)$$

We can then rewrite the branching recursion as follows. The first step is,

$$y_2 = A[0, y_1] + B = \rho y_1 + \sigma_{0,t_1} + B\,,$$

whereas the nth step is,

$$y_{n+1} = A[t_n, t_n + y_n] + B = \rho y_n + \sigma_{t_n, t_{n+1}} + B\,.$$

Solving this recursion gives the following lemma.

Lemma 1. *The branching process can be written as*

$$y_{n+1} = \rho^n y_1 + \sum_{i=0}^{n-1} \rho^i (\sigma_{t_{n-i}, t_{n-i+1}}) + \frac{1-\rho^{n+1}}{1-\rho} B\,. \tag{20}$$

We shall use the following Lemma, proved in [2].

Lemma 2. *Suppose we have two sequences of real numbers,* $\{V_i\}_{i=1}^n$ *and* $\{\zeta_i\}_{i=1}^n$*, such that* $0 \le \zeta_1 \le \cdots \le \zeta_n$*. Then*

$$\zeta_1 V_1 \,+\, \cdots \,+\, \zeta_n V_n \;\le\; \zeta_n \max\{0, V_n, V_n \,+\, V_{n-1}, \ldots, V_n \,+\, \cdots \,+\, V_1\}\,. \tag{21}$$

Proof. The proof is by induction on n. Suppose (21) holds for n. Since the right hand side is non–negative, we can replace ζ_n with ζ_{n+1} on the right hand side, and add $\zeta_{n+1} V_{n+1}$ to both sides, thus obtaining equation (22):

$$\begin{aligned}
&\zeta_1 V_1 + \cdots + \zeta_n V_n + \zeta_{n+1} V_{n+1} \\
&\quad \le \zeta_{n+1} \left(V_{n+1} + \max\{0,\, V_n, V_n + V_{n-1}, \ldots, V_n + \cdots + V_1\}\right) \\
&\quad = \zeta_{n+1} \max\{V_{n+1}, V_{n+1} + V_n, V_{n+1} + V_n + V_{n-1}, \ldots, V_{n+1} + V_n + \cdots + V_1\} \\
&\quad \le \zeta_{n+1} \max\{0, V_{n+1}, V_{n+1} + V_n, \ldots, V_{n+1} + \cdots + V_1\}\,.
\end{aligned} \tag{22}$$

which establishes (21) for $n+1$.

By combining the preceding lemmas, we now obtain a substantial improvement over (19).

Theorem 2. *Assume that* $\rho < 1$*. Then we have for all* n

$$y_n \le y_1 + \sigma + \frac{B}{1-\rho}$$

Proof. The proof of the Theorem follows by combining the last two lemmas. The sequence $\hat{\zeta}_i$ in the last Lemma corresponds to the power of ρ's: $\zeta_n = \rho^0 = 1$, $\zeta_{n-1} = \rho$, $\zeta_{n-i} = \rho^i$. Also we have $V_i = \sigma_{t_i,t_{i+1}}$. All elements of the max in (21) are given by a summation of the form

$$V_n + V_{n-1} + ... + V_{n-i} = \sigma_{t_{n-i},t_{n+1}}$$

which is bounded by σ. This proves the bound.

It has been argued that deterministic bounds yield overly pessimistic performance bounds, which gave rise to various competing stochastic network calculi [18]. We here adopt the so-called traffic-amount-centric arrival curves to the branching processes considered here.

Definition 3. *An arrival process is said to satisfy the (σ, ρ) constraints probabilistically with non-increasing bounding function $f(x)$ for some constant ρ and σ, if it satisfies for any interval $[s,t]$, $t \geq s$:*

$$\Pr[A[s,t] - \rho(t-s) > \sigma + x] \leq f(x)$$

The definition above allows that the (σ, ρ) constraint is violated by the AGP, albeit with a small probability which is bounded by $f(x)$. We then obtain the following probabilistic bound for the branching process.

Theorem 3. *Assume that $\rho < 1$ and that A_n satisfies the (σ, ρ) constraints probabilistically with bounding function $f(x)$. Then we have,*

$$\Pr[y_{n+1} - \rho^n y_1 - \frac{1-\rho^{n+1}}{1-\rho} B > \sigma + x] \leq f(x)\,.$$

Proof. Following the arguments of the proof of Theorem 2, we have,

$$y_{n+1} \leq \rho^n y_1 + \sigma_{t_1,t_{n+1}} + \frac{1-\rho^{n+1}}{1-\rho} B\,,$$

or equivalently,

$$y_{n+1} - \rho^n y_1 - \frac{1-\rho^{n+1}}{1-\rho} B \leq \sigma_{t_1,t_n+1}\,,$$

This inequality then implies,

$$\Pr[y_{n+1} - \rho^n y_1 - \frac{1-\rho^{n+1}}{1-\rho} B > \sigma + x] \leq \Pr[\sigma_{t_1,t_n+1} > \sigma + x] \leq f(x)\,.$$

Here the last inequality follows from the definition of $\sigma_{t_1,t_{n+1}}$ and definition 3.

8 Conclusions

In this paper we reconsider branching processes and their use in evaluating performance of communication systems from two non-standard perspectives. First, we introduce max-plus branching, where branching corresponds to finding the maximal offspring of a member of the current generation rather then summing all offspring of members of the current generation. We show that, as for a standard branching processes, a divisibility property holds. However, in the case of max-plus branching, dividing the current generation leads to maximizing over the respective offspring. The divisibility property also allows us to define continuous-state max-branching in terms of Lévy processes, just like for ordinary branching. All max-plus branching processes are investigated in the presence of a migration component which is either added in the ordinary sense or in the max-plus sense. Various applications in queueing theory for this type of branching processes are introduced along the way.

For the single-type discrete max-branching with i.i.d. migration, we obtain a system of equations for the stationary solution. For the continuous equivalent, a functional equation is obtained for the Laplace-Stieltjes transform of the stationary solution. Finally, for multitype max-branching, we study conditions which ensures the existence of a stationary solution.

A network calculus approach to branching processes constitutes the second non-standard perspective. We show that a branching process can be created from a single arrival process and then find bounds on the growth of this branching process in terms of the deterministic constraints on this arrival process. Finally, we relax these constraints by assuming probabilistic bounds on the arrival process.

References

1. Altman, E.: Semi-linear stochastic difference equations. Discrete Event Dynamic Systems 19, 115–136 (2008)
2. Altman, E., Foss, S., Riehl, E., Stidham, S.: Performance Bounds and Pathwise Stability for Generalized Vacation and Polling Systems. Operations Research 46(1), 137–148 (1998)
3. Altman, E., Kofman, D.: Bounds for Performance Measures of Token Rings. IEEE/ACM Transactions on Networking 4(2), 292–299 (1996)
4. Altman, E.: Stochastic recursive equations with applications to queues with dependent vacations. Annals of Operations Research 112(1), 43–61 (2002)
5. Altman, E.: On stochastic recursive equations and infinite server queues. In: Proceedings of IEEE Infocom, Miami, March 13-17 (2005)
6. Altman, E., Fiems, D.: Expected waiting time in symmetric polling systems with correlated vacations. Queueing Systems 56, 241–253 (2007)
7. Baccelli, F., Cohen, G., Olsder, G.J., Quadrat, J.P.: Synchronization and Linearity. Wiley, Chichester (1992)
8. Borel, E.: Sur l'emploi du théorème de Bernoulli pour faciliter le calcul d'une infinité de coefficients. application au problème de lattente à un guichet. Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences 214, 452–456 (1942)

9. Cohen, J.E., Kesten, H., Newman, C.M. (eds.): Random Matrices and Their Applications. Contemporary Mathematics, vol. 50. American Mathematical Society, Providence (1986)
10. Cruz, R.L.: A Calculus for Network Delay. Part I: Network Elements in Isolation and Part II: Network Analysis. IEEE Transactions on Information Theory 37(1), 114–141 (1991)
11. Fiems, D., Altman, E.: Applying branching processes to delay-tolerant networks. In: Proceedings of the 4th International Conference on Bio-Inspired Models of Network, Information, and Computer Systems, Avignon. Lecture Notes of ICST, vol. 39, pp. 117–125 (December 2009)
12. Gaeta, R., Balbo, G., Bruell, S., Gribaudo, G., Sereno, M.: A simple analytical framework to analyze search strategies in large-scale peer-to-peer networks. Performance Evaluation 62(1-4), 1–16 (2005)
13. Gaeta, R., Sereno, M.: Generalized probabilistic flooding in unstructured peer-to-peer networks. IEEE Transaction on Parallel and Distributed Systems 22(12), 2055–2062 (2011)
14. Glasserman, P., Yao, D.D.: Stochastic vector difference equations with stationary coefficients. Journal of Applied Probability 32, 851–866 (1995)
15. Goldberg, M.A.: Numerical Solution of Integral Equations. Springer (1990)
16. Grishechkin, S.A.: On a relation between processor sharing queues and Crump-Mode-Jagers branching processes. Advances in Applied Probability 24, 653–698 (1992)
17. Grishechkin, S.A.: Multiclass batch arrival retrial queues analyzed as branching processes with immigration. Queueing Systems 11, 395–418 (1992)
18. Jiang, Y.: A basic stochastic network calculus. ACM SIGCOMM Computer Communication Review 36(4), 123–134 (2006)
19. Kendall, D.G.: Some problems in the theory of queues. Journal of the Royal Statistical Society Series B-Methodological 13, 151–185 (1951)
20. Le Boudec, J.-Y., Thiran, P.: Network Calculus. LNCS, vol. 2050. Springer, Heidelberg (2001)
21. Leskela, L., Robert, P., Simatos, F.: Interacting branching processes and linear file-sharing networks. Advances in Applied Probability 42(3), 834–854 (2010)
22. Peeters, G.T., Van Houdt, B.: On the Maximum Stable Throughput of Tree Algorithms With Free Access. IEEE Transactions on Information Theory 55(11), 5087–5099 (2009)
23. Resing, J.A.C.: Polling systems and multi-type branching processes. Queueing Systems 13, 409–426 (1993)

Efficient Generation of PH-Distributed Random Variates

Gábor Horváth[2], Philipp Reinecke[1], Miklós Telek[2], and Katinka Wolter[1]

[1] Freie Universität Berlin
Institut für Informatik
Takustraße 9
14195 Berlin, Germany
{philipp.reinecke,katinka.wolter}@fu-berlin.de
[2] Budapest University of Technology and Economics
Department of Telecommunications
1521 Budapest, Hungary
{hgabor,telek}@webspn.hit.bme.hu

Abstract. Phase-type (PH) distributions are being used to model a wide range of phenomena in performance and dependability evaluation. The resulting models may be employed in analytical as well as in simulation-driven approaches. Simulations require the efficient generation of random variates from PH distributions. PH distributions have different representations and different associated computational costs for random-variate generation. In this paper we study the problem of efficient representation and efficient generation of PH distributed variates.

Keywords: PH distribution, pseudo random number generation.

1 Introduction

Phase-type (PH) distributions [1] are very useful in modelling interarrival times, failure times, and other phenomena in computer systems. They can be employed in analytical approaches as well as in simulation-based evaluations. When PH distributions are used in simulations, often large sets of random variates must be generated, and thus efficiency of random-variate generation from PH distributions is important. We consider algorithms that 'play' the underlying Markov chain. These algorithms provide high accuracy, because they represent each PH sample as a sum of exponential samples, directly following the definition of PH distributions.

PH distributions have different Markovian representations. In [2] we observed that the computational complexity of PH-distributed random-variate generation depends on the representation. This fact poses the research problem of finding the representation that is optimal for random-variate generation.

In [3] we addressed the question by considering the sub-class of Acyclic Phase-type (APH) distributions. For APH distributions the optimal representation is obtained as follows: Starting from any representation the first step is to transform

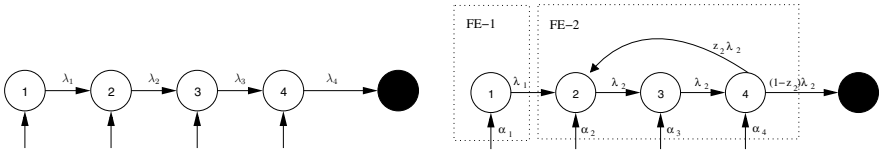
K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 271–285, 2012.

the representation to the CF-1 canonical form defined in [4]. An APH distribution is in CF-1 form if the generator matrix has a bi-diagonal structure and the transition rates are non-decreasing towards the absorbing state. Transformation to the CF-1 form is always possible because all APH distributions have a CF-1 representation [4]. The second step is to find the optimal ordering of the diagonal (and the associated sub-diagonal) elements. It is shown in [3] that for APH in bi-diagonal form the optimal representation is the reversed CF-1 form if it is Markovian. For the case when the reversed CF-1 form is not Markovian, heuristic search algorithms are proposed to find the optimal ordering of the diagonal elements.

In this paper we generalize the results obtained for the APH class to the PH class. We propose to follow a similar approach. In the first step we transform the representation to a sparse Markovian representation. To the best of our knowledge, the only representation with these properties which is available for the whole PH class is the monocyclic representation proposed by Mocanu and Commault [5]. The monocyclic representation is a natural extension of the CF-1 form, in the sense that the generator matrix remains bi-diagonal, but on the matrix block level. Due to their structures these matrix blocks are referred to as feedback Erlang (FE) blocks. The second step of the proposed procedure is to find the optimal ordering of the FE matrix blocks (and the associated sub-diagonal matrix blocks). We are going to show that in contrast to the APH case the optimal ordering of the FE blocks cannot be predicted in a simple way (e.g., by the associated dominant eigenvalue). As a result, finding the optimal representation composed by FE blocks is based on the use of exhaustive or heuristic search algorithms over the set of possible ordering of the FE blocks.

The method proposed in this paper is composed of two parts, preprocessing and random variate generation. The computational complexity of the sum of both steps should be optimized in general. There are obvious extreme solutions for the cases when very few and extremely large numbers of random samples are required. In the first case the preprocessing phase can be omitted and in the second case arbitrarily large look-up tables can be computed during the preprocessing phase. Our proposed solution is between these extremes. We assume $10^6 - 10^{10}$ samples, where the cost of the preprocessing phase of our method is negligible in case of moderate size (< 15 states) PH distributions. The cost of the preprocessing phase increases sub-linearly with the size.

The paper is structured as follows. We first describe the notation used throughout the paper and briefly recall the results from [3] in Section 2. In Section 3 we propose an algorithm for generating random variates from general PH distributions with monocyclic representation and discuss the associated computational cost. Section 4 presents the transformation of reordering the FE blocks in the monocyclic representation and discusses its properties. A counterexample is presented to show that the nice ordering rules of APH representations are not applicable in case of general PH distributions. In Section 5 we provide heuristics for efficient search of optimal representation, and Section 6 studies the efficiency of the proposed procedures.



(a) Bi-diagonal form for APH distributions.

(b) FE-diagonal form for general PH distributions.

Fig. 1. Comparison of bi-diagonal and FE-diagonal forms for PH distributions

2 Notation and Previous Results

A PH distribution of size n is described by an initial probability vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ and a sub-generator matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$ with entries q_{ij} such that $q_{ii} < 0$ and $q_{ij} \geq 0$ for $i \neq j$. In this case the cumulative distribution function (CDF) is

$$F(x) = 1 - \boldsymbol{\alpha} e^{\mathbf{Q}x} \mathbb{1},$$

where $\mathbb{1}$ is the column vector of ones of appropriate size. The representation $(\boldsymbol{\alpha}, \mathbf{Q})$ is not unique.

Definition 1. *If* $\mathbf{P}$ *is invertible and* $\mathbf{P}\mathbb{1} = \mathbb{1}$*, then the* similarity transformation $(\boldsymbol{\alpha}\mathbf{P}, \mathbf{P}^{-1}\mathbf{Q}\mathbf{P})$ *provides another representation of the same distribution, since its CDF is*

$$1 - \boldsymbol{\alpha}\mathbf{P}\mathrm{e}^{\mathbf{P}^{-1}\mathbf{Q}\mathbf{P}t}\mathbb{1} = 1 - \boldsymbol{\alpha}\mathbf{P}\mathbf{P}^{-1}\mathrm{e}^{\mathbf{Q}t}\mathbf{P}\mathbb{1} = 1 - \boldsymbol{\alpha}\mathrm{e}^{\mathbf{Q}t}\mathbb{1}.$$

The representation $(\boldsymbol{\alpha}, \mathbf{Q})$ is called *Markovian* if $\forall i : \alpha_i \geq 0$ and $\boldsymbol{\alpha}\mathbb{1} = 1$ and $\mathbf{Q}_{ii} < 0$, $\mathbf{Q}_{ij} \geq 0, i \neq j$. With a Markovian representation, we refer to $\boldsymbol{\alpha}$ as the initial probability vector and to $\mathbf{Q}$ as the sub-generator matrix.

The computational complexity of PH-distributed random-variate generation depends on the representation [2]. Our goal is to find the representation with the lowest computational complexity. The optimization of APH representations is based on bi-diagonal representations of APH distributions in [3]. In these representations, the only non-zero entries of the sub-generator matrix $\mathbf{Q}$ are on the diagonal and on the upper diagonal, with $q_{i,i+1} = -q_{ii}$. Bi-diagonal representations can be conveniently specified by the vector $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_n)$, where $\lambda_i = -q_{ii}$ for $i = 1, \ldots, n$. Figure 1(a) shows the CTMC of an APH distribution with size $n = 4$ in bi-diagonal form. All APH distributions have at least one bi-diagonal representation with Markovian initialisation vector [4]. The bi-diagonal representation with non-decreasing $\boldsymbol{\Lambda}$ is referred to as CF-1 form.

Bi-diagonal Markovian representations are only available if all eigenvalues of the sub-generator matrix $\mathbf{Q}$ (i.e. all poles of the Laplace-Stieltjes Transform of the distribution) are real. As general PH distributions may have complex poles, [5] proposed the use of Feedback-Erlang (FE) blocks to represent pairs of complex eigenvalues:

Definition 2. *[5] A* Feedback-Erlang (FE) block *with parameters* (b, q, z) *is a chain of* b *states with transition rate* q *and one transition from the* b*th state to the first state, with rate* zq*. The probability* $z \in [0, 1)$ *is called the feedback probability.*

The dominant eigenvalue of the FE block with parameters (b, q, z) is always real and given by $r = -q\left(1 - z^{1/b}\right)$ [5]. Feedback-Erlang blocks with length $b = 1$ or feedback probability $z = 0$ are called *degenerate* FE blocks. Note that an FE block (b, q, z) with length $b = 1$ corresponds to an exponential distribution with rate q, while $z = 0$ gives the Erlang-b distribution with rate q (the sum of b independent exponentially distributed random variables with parameter q). In both cases, the dominant eigenvalue is $-q$. Analogously to the approach for the APH class, we consider representations with a diagonal structure of the sub-generator:

Definition 3. *An* FE-diagonal representation *consisting of* m *Feedback-Erlang blocks* $(b_i, q_i, z_i), i = 1, \ldots, m$ *has a sub-generator matrix* $\mathbf{Q}$ *where the only non-zero entries are in the FE blocks along the diagonal and the transition rates from the last state of a Feedback-Erlang block to the first state of the next one. The size of the representation is* $n = \sum_{i=1}^{m} b_i$*.*

Where appropriate, we also use the vector notation

$$\boldsymbol{\Upsilon} = \{(b_1, q_1, z_1), \ldots, (b_m, q_m, z_m)\},$$

to describe the structure of the sub-generator matrix. Figure 1(b) shows an example of the CTMC of a general PH distribution in FE-diagonal form. In this representation there are two FE blocks, one of length $b_1 = 1$ with rate $q_1 = \lambda_1$, and one of length $b_2 = 3$ with rate $q_2 = \lambda_2$ and feedback probability z_2. The following theorem, restated from [5], ensures that every PH distribution has at least one Markovian FE-diagonal representation:

Theorem 1. *[5] Every PH distribution has a monocyclic representation with Markovian initial vector. A monocyclic representation is an FE-diagonal representation*

$$\boldsymbol{\Upsilon} = \{(b_1, q_1, z_1), \ldots, (b_m, q_m, z_m)\}$$

such that the dominant eigenvalues of the FE blocks are ordered by increasing absolute value, $|r_i| \leq |r_j|$*, for* $1 \leq i \leq j \leq m$*.*

The monocyclic form can be computed using, e.g., the implementations available in the Butools library [6]. Note that for APH distributions the feedback probabilities are zero for all blocks, i.e. FE-diagonal forms for APH distributions consist of degenerate FE blocks. For the APH class the FE diagonal form thus corresponds to the bi-diagonal form defined in [3], and the monocyclic form is equivalent to the CF-1 form. The next section discusses an algorithm for generating random variates from a general PH distribution in FE-diagonal form.

3 Random-Variate Generation from FE-Diagonal Representations

Given a PH distribution with representation $(\boldsymbol{\alpha}, \mathbf{Q})$ in FE-diagonal form with FE-block vector $\boldsymbol{\Upsilon} = \{(b_1, q_1, z_1), \ldots, (b_m, q_m, z_m)\}$, random variates can be generated by the following algorithm:

Procedure `FE-diagonal`:
Let $x := 0$
Draw an $\boldsymbol{\alpha}$-distributed discrete sample for the initial state.
The chain is in block i and has to traverse l states until the block may be left (e.g., for the left-most state of the ith block, $l = b_i$).
while $i \leq m$ **do**
  $c = \mathrm{Geo}(z_i)$
  $x+ = \mathrm{Erl}(cb_i + l, q_i)$
  $i++$
  $l = b_i$
end while
Return(x)

This algorithm was proposed as `Procedure Monocyclic` in [2], but can of course also be applied to FE-diagonal representations. The algorithm uses the

$$\mathrm{Geo}(p) = \left\lfloor \frac{\ln U}{\ln p} \right\rfloor \quad \text{and} \quad \mathrm{Erl}(b, q) = -\frac{1}{q} \ln \left(\prod_{i=1}^{b} U_i \right) \tag{1}$$

operations for drawing a random variate from the Geometric distribution with parameter p and support $0, 1, \ldots$, or the Erlang-b with rate q, respectively. In both cases, U denotes a uniformly distributed pseudo random number on $(0, 1)$.

The algorithm works as follows: An initial state is chosen according to the initial probability vector $\boldsymbol{\alpha}$. We assume that this state belongs to FE block i, and there are $1 \leq l \leq b_i$ states to traverse before the chain may enter the next block. Since all rates in the given FE block are equal (q_i), this corresponds to an Erlang-l distribution with rate q_i. When the last state of the block is reached, one may either enter the next block or follow the feedback-loop to the first state of the current block. The number of loops $c = 0, 1, \ldots$ within the ith block follows a geometric distribution with parameter z_i. The random variate corresponding to the loop is Erlang-c-distributed, again with rate q_i. Consequently, for the block entered upon initialisation the algorithm draws a random variate from an Erlang-$(cb_i + l)$ distribution. All the remaining blocks until absorption are entered at the first state, and thus the respective random variates for the jth block $(j = i+1, \ldots, m)$ are distributed according to $(\mathbf{e}_1, \mathbf{F}_j)$ distributions, where $\mathbf{e}_1$ is the row vector with 1 at position 1 and zero everywhere else and $\mathbf{F}_j$ is the sub-generator corresponding to the jth FE block. Following the argument for the initial step, random variates from these distributions are generated from Erlang-$(c_j b_j + l_j)$ distributions, where c_j is the number of loops and $l_j = b_j$ (since each block is entered at the beginning and has to be traversed at least once).

In a preliminary measurement study we observed that the computational cost of computing logarithms dominates the cost of generating PH-distributed random variates. Further Performance Measures Have Been Considered In [2]. Although the exact cost ratio depends on hardware and software specifics, optimisation for the number of logarithm operations appears to be an effective approach. The number of logarithm operations depends on the distribution of the initial probability mass over the Feedback-Erlang blocks and is independent of both the distribution within the blocks and the length of the blocks. Computation of the expected number of logarithms is straightforward:

$$n^*(\boldsymbol{\alpha}, \boldsymbol{\Upsilon}) = 3\boldsymbol{\beta}\boldsymbol{\nu}^{\mathsf{T}},$$

where $\boldsymbol{\beta} = \left(\sum_{i=1}^{b_1} \alpha_i, \sum_{i=b_1+1}^{b_1+b_2} \alpha_i, \ldots, \sum_{i=n-b_m+1}^{n} \alpha_i\right)$ is the vector of initial probabilities for each FE block, and the entries of $\boldsymbol{\nu} = (m, m-1, \ldots, 1)$ give the number of blocks to traverse when entering the ith FE block. The result is multiplied by 3, because 3 logarithm operations are required for the computation of the geometric and Erlang samples.

Usually, the `FE-diagonal` algorithm is more efficient than the algorithms that simply 'play' the CTMC by selecting a sequence of states, due to the special block bi-diagonal structure of the representation. When a general representation is allowed, random selection of the next state is required in each step. This is eliminated in the `FE-diagonal` algorithm, since the next FE block is uniquely determined by the chain structure of the representation. Therefore, we consider the optimisation of representations in FE-diagonal form, and we present numerical results about the computational gain of the `FE-diagonal` algorithm in Section 6.

4 Optimisation for FE-Diagonal Representations of the PH Class

The optimal representation of an APH for random-variate generation is the reversed CF-1 form (non-increasing $\boldsymbol{\Lambda}$), if the reversed CF-1 is Markovian [3]. This result was obtained by an analysis of the properties of the `Swap` operator, which exchanges two adjacent entries of the vector $\boldsymbol{\Lambda}$. The monocyclic representation of the PH class is the generalization of the CF-1 form used for the APH class. The obvious similarity between both forms raises the question whether the result obtained for the APH class can be generalized to the PH class in FE diagonal representation. To answer this question we introduce the similarity transformation matrix $\mathbf{P}$ that swaps two adjacent FE blocks and produces the new initial probability vector $\boldsymbol{\alpha}' = \boldsymbol{\alpha}\mathbf{P}$.

Definition 4. *The* `GSwap(`$\boldsymbol{\alpha}, \mathbf{A}, i$`)` operator *exchanges the ith FE block with the $(i+1)$th FE block ($1 \leq i \leq m-1$) on the diagonal in a block-bi-diagonal representation by swapping the ith and $(i+1)$th entry in the vector $\boldsymbol{\Upsilon}$ (or, equivalently, by swapping the block matrices associated with the FE blocks in the sub-generator).*

The associated similarity transformation matrix $\mathbf{P}$ *has the form*

$$\mathbf{P} = \begin{pmatrix} \mathbf{I}_{\nu\times\nu} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{P}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{\mu\times\mu} \end{pmatrix}, \quad \textit{where} \quad \nu = \sum_{k=1}^{i-1} b_k,\ \mu = \sum_{k=i+2}^{m} b_k.$$

ν *and* μ *are the number of states in front of and behind the two blocks, respectively, and* $\hat{\mathbf{P}} \in \mathbb{R}^{(b_i+b_{i+1})\times(b_i+b_{i+1})}$ *is the solution of*

$$\begin{pmatrix} \mathbf{F}_i & -\mathbf{F}_i \mathbb{1}\mathbf{e}_1 \\ \mathbf{0} & \mathbf{F}_{i+1} \end{pmatrix} \hat{\mathbf{P}} = \hat{\mathbf{P}} \begin{pmatrix} \mathbf{F}_{i+1} & -\mathbf{F}_{i+1}\mathbb{1}\mathbf{e}_1 \\ \mathbf{0} & \mathbf{F}_i \end{pmatrix} \tag{2}$$

$$\hat{\mathbf{P}}\mathbb{1} = \mathbb{1}. \tag{3}$$

$\hat{\mathbf{P}}$ is the similarity transformation matrix that swaps the order of the ith and $(i+1)$th FE blocks. Due to the block upper-triangular structure of the coefficient matrices in (2), $\hat{\mathbf{P}}$ is a block lower-triangular matrix. We use the $\texttt{GSwap}(\boldsymbol{\alpha}, \boldsymbol{\Upsilon}, i)$ notation to denote the swap operation of the ith and $(i+1)$th FE blocks.

We summarise the properties of the `GSwap` operator in the following remarks. While the `GSwap` operator is a generalization of the `Swap` operator, the fact that the Monocyclic form is block-bi-diagonal with block matrices potentially having size larger than 1 results in one important difference, described in Remark 2.

Remark 1. The structure of $\mathbf{P}$, which describes the `GSwap` operator (i.e. $\mathbf{Q}' = \mathbf{P}^{-1}\mathbf{Q}\mathbf{P}$), ensures that `GSwap` has only local effects on the initial probability vector, i.e. it only affects the entries of the initial probability vector that belong to the two FE blocks being swapped. We denote the vector of these probabilities by $\hat{\boldsymbol{\alpha}}$.

Remark 2. Given two Feedback-Erlang blocks (b_i, q_i, z_i) and $(b_{i+1}, q_{i+1}, z_{i+1})$ with corresponding sub-generator matrix

$$\mathbf{Q} = \begin{pmatrix} \ddots & \ddots & & & & & \\ & -q_i & q_i & & & & \\ & & \ddots & \ddots & & & \\ & z_i q_i & & -q_i & (1-z_i)q_i & & \\ & & & & -q_{i+1} & q_{i+1} & \\ & & & & & \ddots & \ddots \\ & & & & z_{i+1}q_{i+1} & & -q_{i+1} & (1-z_{i+1})q_{i+1} \\ & & & & & & & \ddots & \ddots \end{pmatrix},$$

application of the GSwap operator exchanges the entire block matrices corresponding to the FE blocks resulting in

$$\mathbf{Q}' = \begin{pmatrix} \ddots & \ddots & & & & & \\ & -q_{i+1} & q_{i+1} & & & & \\ & & \ddots & \ddots & & & \\ & z_{i+1}q_{i+1} & & -q_{i+1} & (1-z_{i+1})q_{i+1} & & \\ & & & & -q_i & q_i & \\ & & & & & \ddots & \ddots \\ & & & & z_iq_i & & -q_i \\ & & & & & & \ddots \end{pmatrix}.$$

This means that the GSwap operator maintains the proper exit rates from the FE blocks.

Remark 3. The Steinhaus-Johnson-Trotter theorem [7] ensures that all permutations of FE blocks can be obtained by repeated application of the GSwap operator.

According to Lemma 1 of [3], for APH distributions in bi-diagonal form any swap of transition rates that moves a phase with a higher rate away from the absorbing state and maintains a Markovian initialisation vector results in a more efficient representation for simulation. This means that for APH distributions the direction of search for the optimal representation is known solely from the properties of the sub-generator matrix and it is independent of the initial probability vector. Unfortunately, the corresponding statement does not hold for PH distributions in FE-diagonal form, as we demonstrate with the following counterexample:

Example 1. Let $\mathbf{Q}$ denote the Monocyclic sub-generator matrix defined by $\boldsymbol{\Upsilon} = ((1, 0.1, 0), (b_1, q_1, z_1), (b_2, q_2, z_2))$, where

$$b_1 = 3, q_1 = 1.5, z_1 = 0.5 \text{ and } b_2 = 3, q_2 = 1, z_2 = 0.$$

Exchanging the second and third block results in $\boldsymbol{\Upsilon}' = ((1, 0.1, 0), (b_2, q_2, z_2), (b_1, q_1, z_1))$ and the associated sub-generator matrix $\mathbf{Q}'$. Let

$$\mathbf{P} = \begin{pmatrix} 1 & \\ & \hat{\mathbf{P}} \end{pmatrix} \text{ such that } \mathbf{Q}' = \mathbf{P}^{-1}\mathbf{Q}\mathbf{P} \tag{4}$$

denote the similarity transformation that describes the GSwap operation for this case, i.e. $\hat{\mathbf{P}}$ is the solution of Equations 2 and 3:

$$\hat{\mathbf{P}} = \begin{pmatrix} 1 & & & & & \\ 0.3333 & 0.6667 & & & & \\ 0.1111 & 0.4444 & 0.4444 & & & \\ -0.925926 & 0.4444 & 0.8889 & 0.592593 & & \\ 0 & -0.925926 & 0.4444 & 0.592593 & 0.8889 & \\ 0 & 0 & -0.925926 & 0.148148 & 0.4444 & 1.3333 \end{pmatrix} \tag{5}$$

Now consider the two initial vectors

$$\boldsymbol{\alpha}_1 = (0.09 \mid 0.1, 0.3, 0.31 \mid 0.1, 0.1, 0),$$

and

$$\boldsymbol{\alpha}_2 = (0.09 \mid 0.1, 0.3, 0.31 \mid 0.2, 0, 0)$$

whose only difference is the distribution of the probability mass assigned to the third FE block. The costs for random-variate generation from $(\boldsymbol{\alpha}_1, \mathbf{Q})$ and $(\boldsymbol{\alpha}_2, \mathbf{Q})$, are

$$n^*(\boldsymbol{\alpha}_1, \mathbf{Q}) = n^*(\boldsymbol{\alpha}_2, \mathbf{Q}) = 3 \cdot (0.09 \cdot 3 + 0.71 \cdot 2 + 0.2) = 5.67.$$

After swapping the two blocks using $\mathbf{P}$ the resulting initial probability vectors are

$$\boldsymbol{\alpha}_1' = \boldsymbol{\alpha}_1 \mathbf{P} = (0.09 \mid 0.141852, 0.28963, 0.271111 \mid 0.118519, 0.0888889, 0)$$

and

$$\boldsymbol{\alpha}_2' = \boldsymbol{\alpha}_2 \mathbf{P} = (0.09 \mid 0.0492593, 0.426667, 0.315556 \mid 0.118519, 0, 0)$$

respectively. Note that in $\boldsymbol{\alpha}_1'$ the initial probability mass assigned to the third FE block increased from 0.2 to 0.207407, while in $\boldsymbol{\alpha}_2'$ the probability mass decreased from 0.2 to 0.118519. The costs of random-variate generation changed as follows:

$$n^*(\boldsymbol{\alpha}_1', \mathbf{Q}') = 3 \cdot 1.8825939 = 5.6477817,$$
$$n^*(\boldsymbol{\alpha}_2', \mathbf{Q}') = 3 \cdot 1.9714836 = 5.9144508.$$

That is, with $\boldsymbol{\alpha}_1$ swapping the blocks resulted in a cost decrease, while with $\boldsymbol{\alpha}_2$ costs increased.

The example illustrates that with true FE-diagonal representations (i.e. those with non-degenerate FE blocks) the effect of swapping two consecutive FE blocks may depend not only on the properties of the sub-generator of the distribution, but also on the distribution in the initial probability vector. Consequently, the procedures proposed for finding the optimal representation of APH distributions in [3] cannot be used for the PH class.

5 Algorithms for Monocyclic Optimisation

Example 1 shows that the reversed Monocyclic form of a true PH distribution is not guaranteed to be optimal, even if the initialisation vector is Markovian. Hence the efficient optimisation methods developed for the APH class cannot be applied to the PH class, since, first, there is no general optimum that only needs to be checked for non-negativity of the initialisation vector, and, second, the direction in which to search for the optimum cannot be derived from the sub-generator alone. On the other hand, Example 1 is not just bad news, since

```
Algorithm GBubbleSortOptimise(α, ϒ):
for i = 1, ..., m − 1 do
  for j = 1, ..., m − 1 do
    (α′, ϒ′) :=GSwap(α, ϒ, i)
    if ComparisonHeuristic(α, ϒ, j) = true ∧ α′ ≥ 0 then
      (α, ϒ) := (α′, ϒ′)
    else
      break
    end if
  end for
end for
return (α, ϒ)
```

Fig. 2. `GBubbleSortOptimise` attempts to re-order phases such that the global ordering imposed by `ComparisonHeuristic` is generated

it also shows that a cost reduction by swapping adjacent FE blocks is indeed possible.

The optimum for the FE-diagonal representation of a PH distribution can be found by an optimisation of the costs over the set of all permutations of the FE blocks. This exhaustive approach is guaranteed to find the optimum with respect to all permutations, but involves generating and checking $m!$ representations. The approach may be feasible if neither the number of FE blocks m nor the block lengths are too large, but running-times become prohibitive for large m or large FE blocks. Large m require a large number of permutations to be checked, while large FE blocks imply large $\hat{\mathbf{P}}$, and thus higher costs for solving (2) and (3).

Therefore, we propose extensions of the algorithms for efficient APH optimisation. We need to replace the strict ordering criterion available for the bi-diagonal form of APH distributions with efficient heuristics. The general approach of the algorithms `GBubbleSortOptimise` in Figure 2 and `GFindMarkovian` in Figure 3 are identical to those for APH optimization. In both algorithms, the comparison of two adjacent rates has been replaced by a call to the generic routine `ComparisonHeuristic`, which returns `true` or `false`, depending on whether the two Feedback-Erlang blocks given as its arguments are in the order imposed by the heuristic. The `GBubbleSortOptimise` algorithm is guaranteed to always find a Markovian representation, since it does not leave the region of orderings with Markovian initialisation vectors. The `GFindMarkovian` algorithm, on the other hand, may terminate without finding a Markovian representation. It is guaranteed to terminate with a Markovian representation only with the eigenvalue heuristic discussed below.

In the following we discuss four heuristics that can be used as `ComparisonHeuristic` in either algorithm. The heuristics presented here have been derived based on the following argument: In Lemma 1 of [3] we showed that the optimal ordering for the APH case is obtained if the ordering of the elements of the diagonal of the CF-1 form is reversed (provided that this ordering is Markovian). Due to the simplicity of the CF-1 form, this re-ordering can be seen equivalently

Algorithm `GFindMarkovian`$(\boldsymbol{\alpha}, \boldsymbol{\Upsilon})$:
Let $(\boldsymbol{\alpha}', \boldsymbol{\Upsilon}')$ be the reversed Monocyclic form of $(\boldsymbol{\alpha}, \boldsymbol{\Upsilon}')$
r:=0
while $\neg(\boldsymbol{\alpha}' \geq \mathbf{0})$ **do**
  $i := \max\{2, \text{argmin}_i\{\alpha'_i < 0\}\}$
  while $\neg(\boldsymbol{\alpha}' \geq \mathbf{0}) \wedge \exists k$: `ComparisonHeuristic`$(\boldsymbol{\Upsilon}[k], \boldsymbol{\Upsilon}[k+1]) =$ false **do**
    $k := \text{argmin}_j\{j \mid i-1 \leq j \leq m-1 \wedge \boldsymbol{\Upsilon}[j] \geq \boldsymbol{\Upsilon}[j+1]\}$
    $(\boldsymbol{\alpha}', \boldsymbol{\Upsilon}') :=$ `GSwap`$(\boldsymbol{\alpha}', \boldsymbol{\Upsilon}', k)$
    if $(\boldsymbol{\alpha}', \boldsymbol{\Upsilon}')$ is a new representation **then**
      $r++$
    end if
    if $r = m!$ **then**
      goto END
    end if
  end while
end while
END:
return $(\boldsymbol{\alpha}', \boldsymbol{\Upsilon}')$

Fig. 3. `GFindMarkovian` starts from the possibly non-Markovian reversed Monocyclic form and searches for a Markovian representation by re-ordering phases such that the reversed ordering imposed by `ComparisonHeuristic` is produced

as a re-ordering of the dominant eigenvalues, of the means, and of the exit rates. Furthermore, property (3) in [3] corresponds to the determinant of the swap matrix being larger than 1. These four criteria differ from each other if true Feedback-Erlang blocks are compared.

Eigenvalues Heuristic. The eigenvalues heuristic relates to the CF-1 case most directly. Recall that the monocyclic representation is defined such that the Feedback Erlang blocks are ordered along the diagonal according to increasing absolute value of their dominant eigenvalues. The eigenvalues heuristic directly applies the observation from the CF-1 case that swapping two blocks may move probability mass to the right iff the eigenvalue of the right phase is larger than that of the left phase. Equivalently, in the monocyclic case probability mass may be moved to the right if the dominant eigenvalue of the right FE block is larger than that of the left FE block. We define the Eigenvalues heuristic as follows:

$$\texttt{EigenvaluesHeuristic}(\boldsymbol{\alpha}, \boldsymbol{\Upsilon}, i) = \begin{cases} \text{true} & |r_i| < |r_{i+1}|, \\ \text{false} & \text{else}, \end{cases}$$

that is, the heuristic returns true if the absolute value of the dominant eigenvalue of the ith block is larger than that of the $(i+1)$th block.

Mean Heuristic. The mean heuristic stems from the following observation for the CF-1 case: Re-ordering phases directly relates to re-ordering the means of the associated distributions. I.e., swapping two phases such that the one with higher rate is moved to the left is equivalent to swapping them such that the

one with the lower mean moves to the left. This idea can be applied to the monocyclic case as follows. First, we have to assign a mean to each FE block. While this is unambiguous in the CF-1 case (where FE blocks are of length 1), in the monocyclic case probability mass may be assigned to all phases of a block, rendering the mean dependent on the distribution of the mass. The most straightforward approach is then to assign probability mass of 1 to the first entry of the block. The mean of a Feedback-Erlang block (b_i, q_i, z_i) with sub-generator matrix $\mathbf{F}_i$ and probability mass 1 at the first entry is $\hat{M}_i = \mathbf{e}_1(-\mathbf{F}_i)^{-1}\mathbb{1}$. The mean heuristic is then defined as

$$\texttt{MeanHeuristic}(\boldsymbol{\alpha}, \boldsymbol{\Upsilon}, i) = \begin{cases} \text{true} & \hat{M}_i > \hat{M}_{i+1}, \\ \text{false} & \text{else.} \end{cases}$$

Exit-Rates Heuristic. The exit-rates heuristic compares the exit rates $(1 - z_i)q_i, (1 - z_{i+1})q_{i+1}$ of neighbouring FE blocks. Based on the result for the CF-1 case, optimisation then consists in re-ordering blocks such that the highest exit rates (i.e. largest rates q_i in the CF-1 case) move to the left. The heuristic is defined as follows:

$$\texttt{ExitRatesHeuristic}(\boldsymbol{\alpha}, \boldsymbol{\Upsilon}, i) := \begin{cases} \text{true} & (1 - z_i)q_i < (1 - z_{i+1})q_{i+1}, \\ \text{false} & \text{else.} \end{cases}$$

Determinant Heuristic. For the APH case, the similarity transformation matrix $\hat{\mathbf{P}}$ has the following explicit structure [3]:

$$\hat{\mathbf{P}} = \begin{pmatrix} 1 & 0 \\ \frac{q_i - q_{i+1}}{q_i} & \frac{q_{i+1}}{q_i}, \end{pmatrix} \tag{6}$$

with determinant $|\hat{\mathbf{P}}| = q_{i+1}/q_i$. In this case, swapping two adjacent rates moves probability mass to the right iff $q_{i+1} > q_i$, or, equivalently, if $|\hat{\mathbf{P}}| > 1$. The determinant heuristic, defined as

$$\texttt{DeterminantHeuristic}(\boldsymbol{\alpha}, \boldsymbol{\Upsilon}, i) := \begin{cases} \text{true} & |\hat{\mathbf{P}}| > 1, \\ \text{false} & \text{else.} \end{cases}$$

generalizes this criterion to the general case.

5.1 Discussion

While these heuristics are exact for degenerate FE blocks, they may be misleading with non-degenerate FE blocks, as can be illustrated using Example 1. Table 1 shows the relevant properties considered by the eigenvalues, mean, and exit rates heuristics. The determinant of the swap matrix $\hat{\mathbf{P}}$ is $|\hat{\mathbf{P}}| = 0.208$. Observe that the eigenvalues, mean, and exit rates heuristics would recommend to swap the two blocks. As we saw in the counterexample, this is correct for $\boldsymbol{\alpha}_1$, but incorrect for $\boldsymbol{\alpha}_2$. Likewise, the prediction by the determinant heuristic that swapping would not move probability mass to the right is wrong for $\boldsymbol{\alpha}_1$, but correct for $\boldsymbol{\alpha}_2$.

Table 1. Properties of the FE blocks in Example 1

	$\mathbf{F}_1$	$\mathbf{F}_2$
Dominant eigenvalue	$r_2 = -0.3095$	$r_1 = -1$
Mean with mass 1 at first state	$\hat{M}_2 = 4$	$\hat{M}_1 = 3$
Mean with normalised mass ($\boldsymbol{\alpha}_1$)	$M_2 = 4$	$M_1 = 1.7042$
Mean with normalised mass ($\boldsymbol{\alpha}_2$)	$M_2 = 2.5$	$M_1 = 1.7042$
Exit rate	0.75	1

6 Application of the Algorithms

We have implemented the proposed representation optimization methods in Mathematica. To test their efficiency we also implemented a random general PH representation generator. For a given size n, we first draw uniformly distributed samples for the initial distribution, which are normalized later. We then draw uniformly distributed samples for the off-diagonal elements of the generator matrix and for the transition rates to the absorbing state of the PH distribution. The mean ratio of the off-diagonal elements of the generator matrix and the transition rates to the absorbing state has a significant effect on the cost of random-variate generation. This ratio is referred to as *termination rate* below.

Having a random general PH representation we first compute the monocyclic representation of the same distribution and then we optimize the representation using the introduced heuristic approaches and an exhaustive search method by interchanging the FE block of the representation. The exhaustive search method evaluates all permutations of the FE blocks. The computational complexity of the exhaustive search method becomes significant at $n > 6$. The heuristic search algorithms perform a negligible number of `GSwap` operations compared to the exhaustive search method and find suboptimal representations in the majority of the cases. Table 2 shows the cost of generating PH-distributed random variates based on the obtained representations. When the termination rate is equal to 1 and $n = 6$ there is a gain of $\sim$60% due to the transformation to the monocyclic representation. A further $\sim$40% gain comes from heuristic optimisation of the FE blocks. The results of the exhaustive search method indicates that the suboptimal representation of the eigenvalue, the mean and the exit rate heuristics are very close to the global optimum obtained by the exhaustive search method.

The rows of Table 2 demonstrate the effect of the termination rate. The higher the transition rate to the absorbing state, the lower the number of state transitions before absorption. In case of fast transitions to the absorbing state the simple simulation which 'plays' the transitions of the Markov chain until absorption is an efficient simulation method due to the low number of state transitions. In this case the transformation to the monocyclic representation and the additional optimization cannot reduce the cost of random-variate generation. For both order 6 (Table 2) and order 10 (Table 3) the turning point is around termination rate ~ 10: For higher termination rate the direct simulation is more efficient, and vice versa for lower termination rate.

Table 2. Average simulation costs (number of logarithms) for order 6 PH distribution based on 100 samples

termination rate	random PH	mono cyclic	eigenvalue	mean	exitrate	determinant	exhaustive search
			heuristic				
0.033	332.008	5.07253	3.06384	3.06234	3.12738	5.07174	3.02422
0.1	102.696	5.03166	3.02877	3.05235	3.11337	5.02784	3.01783
0.33	33.308	4.98996	3.08534	3.09713	3.15197	4.98222	3.01082
1	11.6818	4.24592	2.6476	2.61502	2.77029	4.2132	2.53818
3.3	4.53882	3.38355	2.23615	2.18118	2.25545	3.37198	2.11419
10	2.2624	2.7238	1.9441	1.92582	1.9441	2.72091	1.85605
33	1.50076	2.39407	1.91512	1.91488	1.91512	2.39407	1.83054

Table 3. Average simulation costs (number of logarithms) for order 10 PH distribution based on 100 samples

termination rate	random PH	mono cyclic	eigenvalue	mean	exitrate	determinant
			heuristic			
0.033	561.077	8.64065	8.62078	8.62078	8.62078	8.64065
0.1	186.376	8.18222	8.17221	8.17221	8.17221	8.18222
0.33	58.8492	7.97052	7.9417	7.94135	7.94203	7.97014
1	19.4483	6.52973	6.47238	6.47238	6.47238	6.52973
3.3	7.07673	5.45818	5.38195	5.38257	5.38397	5.45661
10	3.15833	4.98105	4.96477	4.96336	4.96477	4.98105
33	1.77828	3.06237	3.05572	3.05572	3.05572	3.06237

It is interesting to see how the proposed transformation reduces the dynamics of the cost. In the evaluated range of termination rates between 0.033 and 33, the cost of random-variate generation with direct simulation varies from 1.5 to 332, while the cost of random-variate generation with optimized representation varies from 1.8 to 3, and a bit larger dynamics reduction appears in Table 3.

Comparing the performance of heuristic optimization methods we obtain that the eigenvalue and mean heuristics perform better than the exit-rate and determinant heuristics. Based on the average performance in Tables 2 and 3, the order of the heuristics is eigenvalue, mean, exit rate and determinant, and there are only a few cases where the mean heuristic performs better than the eigenvalue heuristic. The results of this section are computed by the `GBubbleSortOptimise` procedure, which always terminates with Markovian initial vector.

7 Conclusion

In this paper we considered the optimization of phase-type distributions for random-variate generation. We propose to use the FE-diagonal representations. We studied optimization of the costs involved with random-variate generation and showed by a counterexample that the nice ordering property of the APH

class does not generalize to the PH class with FE-diagonal form. We developed different heuristic algorithms for optimisation of the PH representation and studied the quality of the heuristics compared to the exhaustive search. The structure of the original PH representation affects the gain of representation optimisation. In a wide range of cases (termination rate < 10) the proposed procedure reduces the cost of random-variate generation. Additionally, the proposed procedure significantly reduces the dependence of the cost on the structure of the original PH representation.

Acknowledgements. This work was supported by DFG grants Wo 898/3-1 and Wo 898/5-1, and OTKA grant no. K-101150.

References

1. Neuts, M.: Probability distributions of phase type. In: Liber Amicorum Prof. Emeritus H. Florin, University of Louvain, pp. 173–206 (1975)
2. Reinecke, P., Wolter, K., Bodrog, L., Telek, M.: On the Cost of Generating PH-distributed Random Numbers. In: Horváth, G., Joshi, K., Heindl, A. (eds.) Proceedings of the Ninth International Workshop on Performability Modeling of Computer and Communication Systems (PMCCS-9), Eger, Hungary, September 17-18, pp. 16–20 (2009)
3. Reinecke, P., Telek, M., Wolter, K.: Reducing the Cost of Generating APH-Distributed Random Numbers. In: Müller-Clostermann, B., Echtle, K., Rathgeb, E.P. (eds.) MMB&DFT 2010. LNCS, vol. 5987, pp. 274–286. Springer, Heidelberg (2010)
4. Cumani, A.: On the Canonical Representation of Homogeneous Markov Processes Modelling Failure-time Distributions. Microelectronics and Reliability 22, 583–602 (1982)
5. Mocanu, S., Commault, C.: Sparse Representations of Phase-type Distributions. Commun. Stat., Stochastic Models 15(4), 759–778 (1999)
6. Bodrog, L., Buchholz, P., Heindl, A., Horváth, A., Horváth, G., Kolossváry, I., Németh, Z., Reinecke, P., Telek, M., Vécsei, M.: Butools: Program packages for computations with PH, ME distributions and MAP, RAP processes (October 2011), http://webspn.hit.bme.hu/~butools
7. Johnson, S.M.: Generation of Permutations by Adjacent Transposition. Mathematics of Computation 17(83), 282–285 (1963)

Finding Prediction Limits for a Future Number of Failures in the Prescribed Time Interval under Parametric Uncertainty

Nicholas Nechval[1], Maris Purgailis[1], Uldis Rozevskis[1], Inta Bruna[1], and Konstantin Nechval[2]

[1] University of Latvia, EVF Research Institute, Statistics Department, Raina Blvd 19, LV-1050 Riga, Latvia
Nicholas Nechval, Maris Purgailis, Uldis Rozevskis, Inta Bruna
nechval@junik.lv
[2] Transport and Telecommunication Institute, Applied Mathematics Department, Lomonosov Street 1, LV-1019 Riga, Latvia
konstan@tsi.lv

Abstract. Computing prediction intervals is an important part of the forecasting process intended to indicate the likely uncertainty in point forecasts. Prediction intervals for future order statistics are widely used for reliability problems and other related problems. In this paper, we present an accurate procedure, called 'within-sample prediction of order statistics', to obtain prediction limits for the number of failures that will be observed in a future inspection of a sample of units, based only on the results of the first in-service inspection of the same sample. The failure-time of such units is modeled with a two-parameter Weibull distribution indexed by scale and shape parameters β and δ, respectively. It will be noted that in the literature only the case is considered when the scale parameter β is unknown, but the shape parameter δ is known. As a rule, in practice the Weibull shape parameter δ. is not known. Instead it is estimated subjectively or from relevant data. Thus its value is uncertain. This δ uncertainty may contribute greater uncertainty to the construction of prediction limits for a future number of failures. In this paper, we consider the case when both parameters β and δ are unknown. In literature, for this situation, usually a Bayesian approach is used. Bayesian methods are not considered here. We note, however, that although subjective Bayesian prediction has a clear personal probability interpretation, it is not generally clear how this should be applied to non-personal prediction or decisions. Objective Bayesian methods, on the other hand, do not have clear probability interpretations in finite samples. The technique proposed here for constructing prediction limits emphasizes pivotal quantities relevant for obtaining ancillary statistics and represents a special case of the method of invariant embedding of sample statistics into a performance index applicable whenever the statistical problem is invariant under a group of transformations, which acts transitively on the parameter space. This technique represents a simple and computationally attractive statistical method based on the constructive use of the invariance principle in mathematical statistics. Frequentist probability interpretation of the technique considered here is clear. Application to other distributions could follow directly. An example is given.

Keywords: Number of failures, Weibull distribution, prediction limits.

K. Al-Begain, D. Fiems, and J.-M. Vincent (Eds.): ASMTA 2012, LNCS 7314, pp. 286–301, 2012.

1 Introduction

Prediction of an unobserved random variable is a fundamental problem in statistics. Hahn and Nelson [1], Patel [2], and Hahn and Meeker [3] provided surveys of methods for statistical prediction for a variety of situations on this topic. In the areas of reliability and life-testing, this problem translates to obtaining prediction intervals for lifetime distributions. One of the earlier works on prediction for the Weibull distribution is by Mann and Saunders [4]. They considered prediction intervals for the smallest of a set of future observations, based on a small (two or three) preliminary sample of past observations. An expression for the warranty period (time before the failure of the first ordered observation from a set of future observations or a lot) was derived as a function of the ordered past observations. Mann [3] extended the results for lot sizes n = 10 (5) 25 and sample sizes m = 2 (1) n −3 for a specified assurance level of 0.95. This method requires numerical integration. In addition, the tables provided are limited to sample sizes less than 25 and are given only for the assurance level of 0.95. Antle and Rademaker [5] provided a method of obtaining a prediction bound for the largest observation from a future sample of the Type I extreme value distribution, based on the maximum likelihood estimates of the parameters. They used Monte Carlo simulations to obtain the prediction intervals. Using the well-known relationship between the Weibull distribution and the Type I extreme value distribution one can use their method to construct an upper prediction limit for the largest among a set of future Weibull observations. However this method is valid only for complete samples and limited to constructing an upper prediction limit for the largest among a set of future observations. Lawless [6] proposed a method for constructing prediction intervals for the smallest ordered observation among a set of k future observations based on a Type II censored sample of past observations. These results are based on the conditional distribution of the maximum likelihood estimates given a set of ancillary statistics. This procedure is exact, but it requires numerical integration, for each new sample obtained, to determine the prediction bounds. Mee and Kushary [7] provided a simulation based procedure for constructing prediction intervals for Weibull populations for Type II censored case. This procedure is based on maximum likelihood estimation and requires an iterative process to determine the percentile points. Meeker and Escobar [8] developed a method to determine the prediction limits (upper and lower) for the future number of fails (Y) in the time interval $[t_c, t_w]$. Such procedure is based upon the conditional binomial distribution of Y given that X components have failed in the time interval $[0, t_c]$. Rostum [9] developed statistical models to predict the state of the pipelines in a network of water distribution. Nelson [10] provided simple prediction limits for the number of failures that will be observed in a future inspection of a sample of units. The past data consist of the cumulative number of failures in a previous inspection of the same sample of units. Life of such units is modeled with a Weibull distribution with a given shape parameter value. Nelson's prediction limits were motivated by the following application. Nuclear power plants contain large heat exchangers that transfer energy from the reactor to steam turbines. Such exchangers typically have 10,000 to 20,000 stainless steel tubes that conduct the flow of steam. Due to stress and corrosion, the tubes develop cracks over time. Cracks are detected during planned inspections. The cracked tubes are subsequently plugged to remove them from service. To develop

efficient inspection and plugging strategies, plant management can use a prediction of the added number of tubes that will need plugging by a specified future time. A prediction expressed as an interval indicates the magnitude of the possible prediction error and quantifies the confidence in the prediction. Nelson [10] has established three procedures for the prediction intervals of the number of tubes to fail in components of heat exchangers, namely (I) the procedure of ratio of probabilities, (II) the procedure of ratio of probabilities simplified and (III) the likelihood ratio procedure. Nordman and Meeker [11] compared probability ratio, simplified probability ratio and likelihood ratio methods proposed by Nelson [10], assuming known the Weibull shape parameter δ. Nechval et al. [12] described a technique for using censored life data from extreme value distributions to construct prediction limits or intervals for future outcomes. Cox [13] presented a general approximate analytical approach to prediction based on the asymptotic distribution of maximum likelihood estimators. Atwood [14] used a similar approach. Efron and Tibshirani [15] described an approximate simulation/pivotal based approach. Beran [16] gave theoretical results on the properties of prediction statements computed with simulated (bootstrap) samples. Kalbfleisch [17] described a likelihood-based method, Thatcher [18] described the relationship between Bayesian and frequentist prediction for the binomial distribution, and Geisser [19] presented a more general overview of the Bayesian approach.

In this paper, we use a frequentist procedure, which is called 'within-sample prediction of future order statistics', when the time-to-failure follows the two-parameter Weibull distribution indexed by scale and shape parameters β and δ. We consider the case when both parameters β and δ are unknown. The technique proposed here for constructing prediction limits emphasizes pivotal quantities relevant for obtaining ancillary statistics and represent a special case of the method of invariant embedding of sample statistics into a performance index applicable whenever the statistical problem is invariant under a group of transformations, which acts transitively on the parameter space [12, 20–28].

Conceptually, it is useful to distinguish between "new-sample" prediction, "within-sample" prediction, and "new-within-sample" prediction. Some mathematical preliminaries for the within-sample prediction are given below.

2 Mathematical Preliminaries for Within-Sample Prediction

2.1 Prediction Limits for Future Order Statistics

For within-sample prediction, the problem is to predict future events in a sample or process based on early data from that sample or process. For example, if m units are followed until t_k and there are k observed failures, $t_1, \ldots, t_k$, one could be interested in predicting the time of the next failure t_{k+1}; time until r additional failures, t_{k+r}; number of additional failures in a future interval.

Theorem 1 (Lower (upper) one-sided prediction limit h on the lth order statistic Y_l in a sample of m observations from the two-parameter Weibull distribution on the basis of the early-failure data $Y_1 \le ... \le Y_k$ from the same sample). Let $Y_1 \le ... \le Y_k$ be the first k ordered early-failure observations from a sample of size m from the two-parameter Weibull distribution

$$f(y \mid \beta, \delta) = \frac{\delta}{\beta}\left(\frac{y}{\beta}\right)^{\delta-1} \exp\left[-\left(\frac{y}{\beta}\right)^{\delta}\right] \quad (y > 0), \tag{1}$$

where δ>0 and β>0 are the shape and scale parameters, respectively. Then a lower one-sided conditional $(1-\alpha)$ prediction limit h on the lth order statistic Y_l $(l > k)$ in the same sample is given by

$$h = w_h^{1/\hat{\delta}} Y_k, \tag{2}$$

where w_h satisfies the equation

$$\Pr\{Y_l > h \mid \mathbf{z}^{(k)}\} = \Pr\left\{\left(Y_l / Y_k\right)^{\hat{\delta}} > \left(h / Y_k\right)^{\hat{\delta}} \,\middle|\, \mathbf{z}^{(k)}\right\} = \Pr\{W > w_h \mid \mathbf{z}^{(k)}\}$$

$$= \left[\int_0^{\infty} v_2^{k-2} \prod_{i=1}^{k} z_i^{v_2} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^{l-k-1-j}}{m-k-j} \left((m-k-j)(w_h z_k)^{v_2} + j z_k^{v_2} + \sum_{i=1}^{k} z_i^{v_2}\right)^{-k} dv_2\right]$$

$$\times \left[\int_0^{\infty} v_2^{k-2} \prod_{i=1}^{k} z_i^{v_2} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^{l-k-1-j}}{m-k-j} \left((m-k) z_k^{v_2} + \sum_{i=1}^{k} z_i^{v_2}\right)^{-k} dv_2\right]^{-1} = 1-\alpha, \tag{3}$$

$$\mathbf{z}^{(k)} = (z_1, ..., z_k), \tag{4}$$

$$Z_i = \left(\frac{Y_i}{\hat{\beta}}\right)^{\hat{\delta}}, \quad i = 1, ..., k, \tag{5}$$

$$w_h = \left(\frac{h}{Y_k}\right)^{\hat{\delta}}, \tag{6}$$

where $\hat{\beta}$ and $\hat{\delta}$ are the maximum likelihood estimates of β and δ based on the first k ordered past observations $Y_1 \le ... \le Y_k$ from a sample of size m from the two-parameter Weibull distribution (1), which can be found from solution of

$$\hat{\beta} = \left(\left[\sum_{i=1}^{k} y_i^{\hat{\delta}} + (m-k) y_k^{\hat{\delta}}\right] \Big/ k\right)^{1/\hat{\delta}}, \tag{7}$$

and

$$\hat{\delta} = \left[\left(\sum_{i=1}^{k} y_i^{\hat{\delta}} \ln y_i + (m-k) y_k^{\hat{\delta}} \ln y_k\right)\left(\sum_{i=1}^{k} y_i^{\hat{\delta}} + (m-k) y_k^{\hat{\delta}}\right)^{-1} - \frac{1}{k}\sum_{i=1}^{k} \ln y_i\right]^{-1}, \tag{8}$$

(Observe that an upper one-sided conditional α prediction limit h on the lth order statistic Y_l based on the first k ordered early-failure observations $Y_1 \le ... \le Y_k$, where

$l > k$, from the same sample may be obtained from a lower one-sided conditional $(1-\alpha)$ prediction limit by replacing $1-\alpha$ by α (α<0.5))

Proof. The joint density of $Y_1 \le ... \le Y_k$ and Y_l is given by

$$f(y_1,...,y_k,y_l \mid \beta,\delta) = \frac{m!}{(l-k-1)!(m-l)!}\prod_{i=1}^{k}\frac{\delta}{\beta}\left(\frac{y_i}{\beta}\right)^{\delta-1}\exp\left(-\left(\frac{y_i}{\beta}\right)^{\delta}\right)$$

$$\times\left[\exp\left(-\left(\frac{y_k}{\beta}\right)^{\delta}\right)-\exp\left(-\left(\frac{y_l}{\beta}\right)^{\delta}\right)\right]^{l-k-1}\frac{\delta}{\beta}\left(\frac{y_l}{\beta}\right)^{\delta-1}\exp\left(-\left(\frac{y_l}{\beta}\right)^{\delta}\right)$$

$$\times\exp\left(-(m-l)\left(\frac{y_l}{\beta}\right)^{\delta}\right). \tag{9}$$

Let $\hat{\beta}$, $\hat{\delta}$ be the maximum likelihood estimates of β, δ, respectively, based on $Y_1 \le ... \le Y_k$ from a complete sample of size m, and let

$$V_1 = \left(\frac{\hat{\beta}}{\beta}\right)^{\delta}, \quad V_2 = \frac{\delta}{\hat{\delta}}, \quad W = \left(\frac{Y_l}{Y_k}\right)^{\hat{\delta}}, \tag{10}$$

and $Z_i = (Y_i/\hat{\beta})^{\hat{\delta}}$, i=1(1)k. Using the invariant embedding technique [12, 20–28], we then find in a straightforward manner, that the joint density of V_1, V_2, W, conditional on fixed $\mathbf{z}^{(k)} = (z_1,...,z_k)$, is

$$f(v_1,v_2,w \mid \mathbf{z}^{(k)}) = \vartheta(\mathbf{z}^{(k)})v_2^{k-1}\prod_{i=1}^{k}z_i^{v_2}(wz_k)^{v_2}v_1^{k}\sum_{j=0}^{l-k-1}\binom{l-k-1}{j}(-1)^{l-k-1-j}$$

$$\times\exp\left[-v_1\left((m-k-j)(wz_k)^{v_2}+jz_k^{v_2}+\sum_{i=1}^{k}z_i^{v_2}\right)\right],$$

$$v_1\in(0,\infty), \quad v_2\in(0,\infty), \quad w\in(1,\infty), \tag{11}$$

where

$$\vartheta(\mathbf{z}^{(k)}) = \left[\int_0^{\infty}v_2^{k-2}\prod_{i=1}^{k}z_i^{v_2}\sum_{j=0}^{l-k-1}\binom{l-k-1}{j}\frac{(-1)^{l-k-1-j}\Gamma(k)}{m-k-j}\left((m-k)z_k^{v_2}+\sum_{i=1}^{k}z_i^{v_2}\right)^{-k}dv_2\right]^{-1}, \tag{12}$$

is the normalizing constant. Using (11), we have that

$$\Pr\{Y_l > h \mid \mathbf{z}^{(k)}\} = \Pr\left\{\left(\frac{Y_l}{Y_k}\right)^{\hat{\delta}} > \left(\frac{h}{Y_k}\right)^{\hat{\delta}} \middle| \mathbf{z}^{(k)}\right\}$$

$$= \Pr\{W > w_h \mid \mathbf{z}^{(k)}\} = \int_0^{\infty}\int_{w_h}^{\infty}\int_0^{\infty} f(v_1, v_2, w \mid \mathbf{z}^{(k)})\, dv_1\, dw\, dv_2$$

$$= \left[\int_0^{\infty} v_2^{k-2} \prod_{i=1}^{k} z_i^{v_2} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^{l-k-1-j}}{m-k-j} \left((m-k-j)(w_h z_k)^{v_2} + j z_k^{v_2} + \sum_{i=1}^{k} z_i^{v_2}\right)^{-k} dv_2\right]$$

$$\times \left[\int_0^{\infty} v_2^{k-2} \prod_{i=1}^{k} z_i^{v_2} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^{l-k-1-j}}{m-k-j} \left((m-k) z_k^{v_2} + \sum_{i=1}^{k} z_i^{v_2}\right)^{-k} dv_2\right]^{-1}, \quad (13)$$

and the proof is complete. □

Theorem 2. (Proof that $V_1 = (\hat{\beta}/\beta)^{\delta}$, $V_2 = (\delta/\hat{\delta})$ and $V_3 = (\hat{\beta}/\beta)^{\hat{\delta}}$) are pivotal quantities). Let $\hat{\beta}$ and $\hat{\delta}$ be the maximum likelihood estimates (MLEs) that are based on a complete random sample (Y_1, …, Y_m) of m observations from the Weibull(β,δ) distribution. Then the unconditional distributions of V_1, V_2 and V_3 do not depend on β and δ, and these are the pivotal quantities.

Proof. Let Y follow the two-parameter Weibull distribution with scale parameter β and shape parameter δ. The probability distribution function of Y is given by

$$F(y \mid \beta, \delta) = 1 - \exp\left(-\left[\frac{y}{\beta}\right]^{\delta}\right), \quad y > 0,\ \beta > 0,\ \delta > 0. \quad (14)$$

Let (Y_1, …, Y_m) be a complete random sample of observations from the Weibull(β,δ) distribution. For the complete case, the MLE $\hat{\delta}$ of δ is the solution to the equation

$$\frac{1}{\hat{\delta}} - \frac{\sum_{i=1}^{m} y_i^{\hat{\delta}} \ln y_i}{\sum_{i=1}^{m} y_i^{\hat{\delta}}} + \frac{1}{m}\sum_{i=1}^{m} \ln y_i = 0 \quad (15)$$

and the MLE $\hat{\beta}$ of β is given by

$$\hat{\beta} = \left(\sum_{i=1}^{m} y_i^{\hat{\delta}} \Big/ m\right)^{1/\hat{\delta}}. \quad (16)$$

Let U_i, i=1, …, n, be a set of independent random variables distributed uniformly over the interval (0,1). Using the inverse transformation method on (14), the variable Y_i may be expressed in terms of U_i as

$$Y_i = \beta\left[\ln\frac{1}{1-U_i}\right]^{1/\delta} \equiv \beta W_i^{1/\delta}, \tag{17}$$

where

$$W_i = \ln\frac{1}{1-U_i} \tag{18}$$

is independent on the Weibull parameters. Multiplying (15) through by $\hat{\delta}$ results in

$$1-\frac{\sum_{i=1}^{m} y_i^{\hat{\delta}} \ln y_i^{\hat{\delta}}}{\sum_{i=1}^{m} y_i^{\hat{\delta}}} + \frac{1}{m}\sum_{i=1}^{m} \ln y_i^{\hat{\delta}} = 0. \tag{19}$$

Substituting (17) into (19) gives

$$1-\frac{\sum_{i=1}^{m} w_i^{1/v_2} \ln w_i^{1/v_2}}{\sum_{i=1}^{m} w_i^{1/v_2}} + \frac{\sum_{i=1}^{m} \ln w_i^{1/v_2}}{m} = 0. \tag{20}$$

Equation (20) shows that for a given set of random numbers, there is a solution in terms of v_2. The value of $V_2 = (\delta/\hat{\delta})$ will thus vary from sample to sample in a way that depends only on sample size, and thus, we may assert that $V_2 = (\delta/\hat{\delta})$ is a pivotal quantity. Substituting (17) into (16) gives

$$v_3 = \frac{\sum_{i=1}^{m} w_i^{1/v_2}}{m} = v_1^{1/v_2}\,. \tag{21}$$

Thus, it follows from (20) and (21) that V_1, V_2 and V_3 are pivotal quantities. This ends the proof. □

It will be noted that these results are also valid for type II singly censored sample, but they are not valid if the samples are type I censored.

Theorem 3. (Lower (upper) one-sided prediction limit h on the lth order statistic Y_l in a sample of m observations from the two-parameter Weibull distribution, with δ=1, on the basis of the past kth order statistic Y_k ($k < l \le m$) from the same sample via the ancillary statistic Y_l/Y_k). Let Y_k be the kth order statistic in a sample of size m from the two-parameter Weibull distribution (1). Then a lower one-sided conditional $(1-\alpha)$ prediction limit h on the lth order statistic Y_l ($l > k$) in the same sample is given by

$$h = w_h Y_k, \tag{22}$$

where w_h satisfies the equation

$$\Pr\{Y_l > h\} = \Pr\left\{\frac{Y_l}{Y_k} > \frac{h}{Y_k}\right\} = \Pr\{W > w_h\} = \frac{1}{\mathrm{B}(k,l-k)\mathrm{B}(l,m-l+1)} \sum_{i=0}^{l-k-1} (-1)^i \binom{l-k-1}{i}$$

$$\times \frac{(k-1)!}{(m-l+1+i)\prod_{j=0}^{k-1}[w_h(m-l+1+i)+l-k-i+j]} = 1-\alpha. \tag{23}$$

$$w_h = \frac{h}{Y_k}. \tag{24}$$

Proof. The joint density of Y_k, Y_l $(k < l)$ is given by

$$f(y_k, y_l \mid \beta) = \frac{1}{\mathrm{B}(k,l-k)\mathrm{B}(l,m-l+1)} \left[1-\exp\left(-\frac{y_k}{\beta}\right)\right]^{k-1}$$

$$\times \left[\exp\left(-\frac{y_k}{\beta}\right) - \exp\left(-\frac{y_l}{\beta}\right)\right]^{l-k-1} \exp\left(-(m-l)\frac{y_l}{\beta}\right) \frac{1}{\beta}\exp\left(-\frac{y_k}{\beta}\right) \frac{1}{\beta}\exp\left(-\frac{y_l}{\beta}\right). \tag{25}$$

Let

$$W = \frac{Y_l}{Y_k}. \tag{26}$$

Using the invariant embedding technique [12, 20–28], we then find in a straightforward manner, that the joint density of W, Y_k is

$$f(w, y_k \mid \beta) = \frac{1}{\mathrm{B}(k,l-k)\mathrm{B}(l,m-l+1)}$$

$$\times \exp\left(-(m-l+1)\frac{wy_k}{\beta}\right)\left[\exp\left(-\frac{y_k}{\beta}\right) - \exp\left(-\frac{wy_k}{\beta}\right)\right]^{l-k-1}$$

$$\times \left[1-\exp\left(-\frac{y_k}{\beta}\right)\right]^{k-1} \exp\left(-\frac{y_k}{\beta}\right)\frac{y_k}{\beta^2}, \quad w \in (1, \infty), \quad y_k \in (0, \infty). \tag{27}$$

Using (27), we have that

$$\Pr\{Y_l > h\} = \Pr\left\{\frac{Y_l}{Y_k} > \frac{h}{Y_k}\right\} = \Pr\{W > w_h\} = \int_{w_h}^{\infty}\int_{0}^{\infty} f(w, y_k \mid \beta)\,dw\,dy_k$$

$$= \frac{1}{\mathrm{B}(k,l-k)\mathrm{B}(l,m-l+1)} \sum_{i=0}^{l-k-1}\sum_{j=0}^{k-1} (-1)^{i+j} \binom{l-k-1}{i}\binom{k-1}{j}$$

$$\times \frac{1}{(m-l+1+i)[w_h(m-l+1+i)+l-k-i+j]}. \tag{28}$$

Since

$$\sum_{j=0}^{k-1}(-1)^j\binom{k-1}{j}\frac{1}{w_h(m-l+1+i)+l-k-i+j}=\frac{(k-1)!}{\prod_{j=0}^{k-1}[w_h(m-l+1+i)+l-k-i+j]}, \tag{29}$$

(28) reduces to

$$\Pr\{Y_l > h\} = \Pr\left\{\frac{Y_l}{Y_k} > \frac{h}{Y_k}\right\} = \Pr\{W > w_h\} = \frac{1}{\mathrm{B}(k,l-k)\mathrm{B}(l,m-l+1)}\sum_{i=0}^{l-k-1}(-1)^i\binom{l-k-1}{i}$$

$$\times \frac{(k-1)!}{(m-l+1+i)\prod_{j=0}^{k-1}[w_h(m-l+1+i)+l-k-i+j]}. \tag{30}$$

This ends the proof. □

Theorem 4. (Lower (upper) one-sided prediction limit h on the lth order statistic Y_l in a sample of m observations from the two-parameter Weibull distribution, with δ=1, on the basis of the early-failure data $Y_1 \le ... \le Y_k$ from the same sample). Let $Y_1 \le ... \le Y_k$ be the first k ordered early-failure observations from a sample of size m from the two-parameter Weibull distribution (1). Then a lower one-sided conditional $(1-\alpha)$ prediction limit h on the lth order statistic Y_l $(l > k)$ in the same sample is given by

$$h = Y_k + w_h^{\bullet}\hat{\beta}, \tag{31}$$

where $w_h^{\bullet}$ satisfies the equation

$$\Pr\{Y_l > h\} = \Pr\left\{\frac{Y_l - Y_k}{\hat{\beta}} > \frac{h - Y_k}{\hat{\beta}}\right\} = \Pr\{W^{\bullet} > w_h^{\bullet}\}$$

$$= \frac{1}{\mathrm{B}(l-k,m-l+1)}\sum_{j=0}^{l-k-1}\frac{\binom{l-k-1}{j}(-1)^{l-k-1-j}}{(m-k-j)[1+(m-k-j)w_h^{\bullet}/k]^k} = 1-\alpha, \tag{32}$$

$$w_h^{\bullet} = \frac{h - Y_k}{\hat{\beta}}, \tag{33}$$

$\hat{\beta}$ is the maximum likelihood estimates of β based on the first k ordered past observations $Y_1 \le ... \le Y_k$ from a sample of size m from the two-parameter Weibull distribution (1), which can be found from solution of

$$\hat{\beta} = \frac{\sum_{i=1}^{k} y_i + (m-k)\, y_k}{k}. \tag{34}$$

Proof. The joint density of $Y_1 \le ... \le Y_k$ and Y_l is given by

$$f(y_1,...,y_k,y_l \mid \beta) = \frac{m!}{(l-k-1)!(m-l)!} \prod_{i=1}^{k} \frac{1}{\beta} \exp\left(-\frac{y_i}{\beta}\right) \left[\exp\left(-\frac{y_k}{\beta}\right) - \exp\left(-\frac{y_l}{\beta}\right)\right]^{l-k-1}$$

$$\times \frac{1}{\beta} \exp\left(-\frac{y_l}{\beta}\right) \exp\left(-(m-l)\frac{y_l}{\beta}\right). \tag{35}$$

Let $\hat{\beta}$ be the maximum likelihood estimate of β, based on $Y_1 \le ... \le Y_k$ from a complete sample of size m, and let

$$V_1 = \frac{\hat{\beta}}{\beta}, \quad W^{\bullet} = \frac{Y_l - Y_k}{\hat{\beta}}, \tag{36}$$

and $Z_i = Y_i / \hat{\beta}$, i=1(1)k. Using the invariant embedding technique [12, 20–28], we then find in a straightforward manner, that the joint density of V_1, $W^{\bullet}$ is

$$f(v_1, w^{\bullet}) = \vartheta v_1^k \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} (-1)^{l-k-1-j} \exp[-v_1((m-k-j)w^{\bullet} + k)],$$

$$v_1 \in (0, \infty), \quad w^{\bullet} \in (0, \infty), \tag{37}$$

where

$$\vartheta = \left[\sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^{l-k-1-j}\Gamma(k)}{(m-k-j)k^k}\right]^{-1} = [\mathrm{B}(l-k, m-l+1)\Gamma(k)/k^k]^{-1} \tag{38}$$

is the normalizing constant. Using (37), we have that

$$\Pr\{Y_l > h\} = \Pr\left\{\frac{Y_l - Y_k}{\hat{\beta}} > \frac{h - Y_k}{\hat{\beta}}\right\} = \Pr\{W^{\bullet} > w_h^{\bullet}\} = \int_{w_h}^{\infty} \int_{0}^{\infty} f(v_1, w^{\bullet}) dv_1 dw^{\bullet}$$

$$= \frac{1}{\mathrm{B}(l-k, m-l+1)} \sum_{j=0}^{l-k-1} \frac{\binom{l-k-1}{j}(-1)^{l-k-1-j}}{(m-k-j)[1+(m-k-j)w_h^{\bullet}/k]^k}$$

$$= \frac{1}{\mathrm{B}(l-k, m-l+1)} \sum_{j=0}^{l-k-1} \frac{\binom{l-k-1}{j}(-1)^{j}}{(m-l+1+j)[1+(m-l+1+j)w_h^{\bullet}/k]^k}, \tag{39}$$

and the proof is complete. □

Theorem 5. (Lower (upper) one-sided prediction limit h on the lth order statistic Y_l in a sample of m observations from the two-parameter Weibull distribution, with δ=1, on the basis of the past kth order statistic Y_k ($k < l \le m$) from the same sample via the ancillary statistic $(Y_l - Y_k)/Y_k$). Let Y_k be the kth order statistic in a sample of size m from the two-parameter Weibull distribution (1). Then a lower one-sided conditional $(1-\alpha)$ prediction limit h on the lth order statistic Y_l ($l > k$) in the same sample is given by

$$h = (1+v_h)Y_k, \tag{40}$$

where v_h satisfies the equation

$$\Pr\{Y_l > h\} = \Pr\{V > v_h\} = \frac{1}{\mathrm{B}(k, l-k)\mathrm{B}(l, m-l+1)} \sum_{i=0}^{l-k-1} (-1)^i \binom{l-k-1}{i}$$

$$\times \frac{(k-1)!}{(m-l+1+i)\prod_{j=0}^{k-1}[v_h(m-l+1+i)+m-k+1+j]} = 1-\alpha, \tag{41}$$

$$v_h = \frac{h-Y_k}{Y_k}. \tag{42}$$

Proof. The joint density of Y_k, Y_l ($k < l$) is given by

$$f(y_k, y_l \mid \beta) = \frac{1}{\mathrm{B}(k, l-k)\mathrm{B}(l, m-l+1)} \left[1-\exp\left(-\frac{y_k}{\beta}\right)\right]^{k-1}$$

$$\times \left[\exp\left(-\frac{y_k}{\beta}\right) - \exp\left(-\frac{y_l}{\beta}\right)\right]^{l-k-1} \exp\left(-(m-l)\frac{y_l}{\beta}\right) \frac{1}{\beta}\exp\left(-\frac{y_k}{\beta}\right) \frac{1}{\beta}\exp\left(-\frac{y_l}{\beta}\right). \tag{43}$$

Let

$$V = \frac{Y_l - Y_k}{Y_k}. \tag{44}$$

Using the invariant embedding technique [12, 20–28], we then find in a straightforward manner, that the joint density of V, Y_k is

$$f(v, y_k \mid \beta) = \frac{1}{\mathrm{B}(k, l-k)\mathrm{B}(l, m-l+1)} \exp\left(-(m-l+1)\frac{vy_k}{\beta}\right) \left[1-\exp\left(-\frac{vy_k}{\beta}\right)\right]^{l-k-1}$$

$$\times\left[1-\exp\left(-\frac{y_k}{\beta}\right)\right]^{k-1}\exp\left(-(m-k+1)\frac{y_k}{\beta}\right)\frac{y_k}{\beta^2},\quad v\in(0,\infty),\quad y_k\in(0,\infty). \tag{45}$$

Using (45), we have that

$$\Pr\{Y_l > h\} = \Pr\left\{\frac{Y_l - Y_k}{Y_k} > \frac{h-y_k}{y_k}\right\} = \Pr\{V > v_h\} = \int_{v_h}^{\infty}\int_{0}^{\infty} f(v, y_k \mid \beta)\,dv\,dy_k$$

$$= \frac{1}{\mathrm{B}(k,l-k)\mathrm{B}(l,m-l+1)}\sum_{i=0}^{l-k-1}\sum_{j=0}^{k-1}(-1)^{i+j}\binom{l-k-1}{i}\binom{k-1}{j}$$

$$\times\frac{1}{(m-l+1+i)[v_h(m-l+1+i)+m-k+1+j]}. \tag{46}$$

Taking into account the identity

$$\sum_{j=0}^{k-1}(-1)^j\binom{k-1}{j}\frac{1}{v_h(m-l+1+i)+m-k+1+j} = \frac{(k-1)!}{\prod_{j=0}^{k-1}[v_h(m-l+1+i)+m-k+1+j]}, \tag{47}$$

(46) reduces to

$$\Pr\{Y_l > h\} = \frac{1}{\mathrm{B}(k,l-k)\mathrm{B}(l,m-l+1)}\sum_{i=0}^{l-k-1}(-1)^i\binom{l-k-1}{i}$$

$$\times\frac{(k-1)!}{(m-l+1+i)\prod_{j=0}^{k-1}[v_h(m-l+1+i)+m-k+1+j]}. \tag{48}$$

This ends the proof. □

2.2 Prediction Limits for Future Number of Failures

Consider the situation in which m units start service at time 0 and are observed until a time t_c when the available Weibull failure data are to be analyzed. Failure times are recorded for the k units that fail in the interval $[0, t_c]$. Then the data consist of the k smallest-order statistics $Y_1 < \ldots < Y_k \le t_c$ and the information that the other $m-k$ units will have failed after t_c. With time (or Type I) censored data, t_c is prescribed and k is random. With failure (or Type II) censored data, k is prescribed and $t_c = Y_k$ is random.

The problem of interest is to use the information obtained up to t_c to construct the Weibull within-sample prediction limits (lower and upper) for the number of units

that will fail in the time interval $[t_c, t_w]$. For example, this t_w could be the end of a warranty period.

Consider the situation when $t_c = Y_k$. Under conditions of Theorem 1, the lower prediction limit for the number of units that will fail in the time interval $[t_c, t_w]$ is given by

$$L_{\text{lower}} = l_{\max} - k, \tag{49}$$

where

$$l_{\max} = \max_{k<l\le m} \arg\left(\Pr\{Y_l > t_w \mid \mathbf{z}^{(k)}\} \le \alpha\right), \tag{50}$$

$$\Pr\{Y_l > t_w \mid \mathbf{z}^{(k)}\} = \Pr\{W > w_{t_w} \mid \mathbf{z}^{(k)}\} = \int_0^{\infty} v_2^{k-2} \prod_{i=1}^{k} z_i^{v_2} \sum_{j=0}^{l-k-1} \binom{l-k-1}{j} \frac{(-1)^{l-k-1-j}}{m-k-j}$$

$$\times \left((m-k-j)(w_{t_w} z_k)^{v_2} + j z_k^{v_2} + \sum_{i=1}^{k} z_i^{v_2} \right)^{-k} dv_2, \tag{51}$$

$$w_{t_w} = (t_w / Y_k)^{\bar{\delta}}. \tag{52}$$

The upper prediction limit for the number of units that will fail in the time interval $[t_c, t_w]$ is given by

$$L_{\text{upper}} = l_{\min} - k - 1, \tag{53}$$

where

$$l_{\min} = \min_{k<l\le m} \arg\left(\Pr\{Y_l > t_w \mid \mathbf{z}^{(k)}\} \ge 1-\alpha\right). \tag{54}$$

3 Numerical Example

Consider the special case where $m = 40$ items simultaneously tested have life times following the Weibull distribution. Two items fail at times, 45 and 100 hours. Consider the situation when $t_c = Y_k = 100$ hours, where k=2. Suppose, say, t_w= 450 hours. Under conditions of Theorem 5, the lower prediction limit for the number of units that will fail in the time interval $[t_c, t_w]$ is given by

$$L_{\text{lower}} = l_{\max} - k = 3 - 2 = 1, \tag{55}$$

where

$$l_{\max} = \max_{k<l\le m} \arg\left(\Pr\{Y_l > t_w\} \le \alpha\right) = 3, \quad \alpha\text{=}0.05, \tag{56}$$

$$\Pr\{Y_l > t_w\} == \Pr\{V > v_{t_w}\} = \frac{1}{\mathrm{B}(k, l-k)\mathrm{B}(l, m-l+1)} \sum_{i=0}^{l-k-1} (-1)^i \binom{l-k-1}{i}$$

$$\times \frac{(k-1)!}{(m-l+1+i)\prod_{j=0}^{k-1}[v_{t_w}(m-l+1+i)+m-k+1+j]}, \tag{57}$$

$$v_{t_w} = \frac{t_w - y_k}{y_k} = 3.5. \tag{58}$$

The upper prediction limit for the number of units that will fail in the time interval $[t_c, t_w]$ is given by

$$L_{\text{upper}} = l_{\min} - k - 1 = 17 - 2 - 1 = 14, \tag{59}$$

where

$$l_{\min} = \min_{k<l\le m} \arg\left(\Pr\{Y_l > t_w\} \ge 1-\alpha\right) = 17. \tag{60}$$

4 Conclusion and Future Work

The aim of the present paper is to construct lower (upper) prediction limits under parametric uncertainty that are exceeded with probability $1-\alpha$ (α) by future observations or functions of observations. The prediction limits depend on early-failure data of the same sample from the two-parameter Weibull distribution, the shape and scale parameters of which are not known.

The methodology described here can be extended in several different directions to handle various problems that arise in practice.

We have illustrated the prediction methods for log-location-scale distributions (such as the Weibull distribution). Application to other distributions could follow directly.

The results obtained in this work can be used to solve the service problems of the following important engineering structures:

(1) Transportation Systems and Vehicles – aircraft, space vehicles, trains, ships;

(2) Civil Structures – bridges, dams, tunnels;

(3) Power Generation – nuclear, fossil fuel and hydroelectric plants;

(4) High-Value Manufactured Products – launch systems, satellites, semiconductor and electronic equipment;

(5) Industrial Equipment – oil and gas exploration, production and processing equipment, chemical process facilities, pulp and paper.

Acknowledgments. This research was supported in part by Grant No. 06.1936, Grant No. 07.2036, Grant No. 09.1014, and Grant No. 09.1544 from the Latvian Council of Science and the National Institute of Mathematics and Informatics of Latvia.

References

1. Hahn, G.J., Nelson, W.: A Survey of Prediction Intervals and Their Applications. Journal of Quality Technology 5, 178–188 (1973)
2. Patel, J.K.: Prediction Intervals-A Review. Communications in Statistics. Theory and Methods 18, 2393–2465 (1989)

3. Hahn, G.J., Meeker, W.Q.: Statistical Intervals: A Guide for Practitioners. Wiley, New York (1991)
4. Mann, N.R., Saunders, S.C.: On Evaluation of Warranty Assurance when Life Has a Weibull Distribution. Biometrika 56, 615–625 (1969)
5. Antle, C.E., Rademaker, F.: An Upper Confidence Limit on the Maximum of *m* Future Observations from a Type I Extreme-Value Distribution. Biometrika 59, 475–477 (1972)
6. Lawless, J.F.: On Estimation of Safe Life When the Underlying Distribution is Weibull. Technometrics 15, 857–865 (1973)
7. Mee, R.W., Kushary, D.: Prediction Limits for the Weibull Distribution Utilizing Simulation. Computational Statistics & Data Analysis 17, 327–336 (1994)
8. Meeker, W., Escobar, L.: Statistical Methods for Reliability Data. John Wiley, New York (1998)
9. Rostum, J.: Decision Support Tools for Sustainable Water Network Management. A Research Project Supported by the European Commission under the Fifth Framework Program, http://www.unife.it
10. Nelson, W.: Weibull Prediction of a Future Number of Failures. Quality Reliability Engineering International 16, 23–26 (2000)
11. Nordman, D.J., Meeker, W.Q.: Weibull Prediction for a Future Number of Failures. Technometrics 44, 15–23 (2002)
12. Nechval, N.A., Nechval, K.N., Purgailis, M.: Statistical Inferences for Future Outcomes with Applications to Maintenance and Reliability. In: Proceedings of the World Congress on Engineering (WCE 2011), London, U.K. Lecture Notes in Engineering and Computer Science, pp. 865–871 (2011)
13. Cox, D.R.: Prediction Intervals and Empirical Bayes Confidence Intervals. In: Gani, J. (ed.) Perspectives in Probability and Statistics, pp. 47–55. Academic Press, London (1975)
14. Atwood, C.L.: Approximate Tolerance Intervals Based on Maximum Likelihood Estimates. Journal of the American Statistical Association 79, 459–465 (1984)
15. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Chapman and Hall, New York (1993)
16. Beran, R.: Calibrating Prediction Regions. Journal of the American Statistical Association 85, 715–723 (1990)
17. Kalbfleisch, J.D.: Likelihood Methods of Prediction. In: Godambe, V.P., Sprott, D.A. (eds.) Proceedings of the Symposium on the Foundations of Statistical Inference, pp. 378–390. Holt, Rinehart, and Winston, Toronto (1971)
18. Thatcher, A.R.: Relationships Between Bayesian and Confidence Limits for Prediction (with discussion). Journal of the Royal Statistical Society B26, 176–210 (1964)
19. Geisser, S.: Predictive Inference: An Introduction. Chapman and Hall, New York (1993)
20. Nechval, N.A., Nechval, K.N.: Invariant Embedding Technique and Its Statistical Applications. In: Conference Volume of Contributed Papers of the 52nd Session of the International Statistical Institute. ISI-International Statistical Institute, Finland, http://www.stat.fi/isi99/proceedings/arkisto/varasto/nech0902.pdf
21. Nechval, N.A., Vasermanis, E.K.: Improved Decisions in Statistics. SIA "Izglitibas soli", Riga (2004)
22. Nechval, N.A., Berzins, G., Purgailis, M., Nechval, K.N.: Improved Estimation of State of Stochastic Systems via Invariant Embedding Technique. WSEAS Transactions on Mathematics 7, 141–159 (2008)

23. Nechval, N., Purgailis, M., Berzins, G., Cikste, K., Krasts, J., Nechval, K.: Invariant Embedding Technique and Its Applications for Improvement or Optimization of Statistical Decisions. In: Al-Begain, K., Fiems, D., Knottenbelt, W. (eds.) ASMTA 2010. LNCS, vol. 6148, pp. 306–320. Springer, Heidelberg (2010)
24. Nechval, N., Purgailis, M., Cikste, K., Berzins, G., Rozevskis, U., Nechval, K.: Prediction Model Selection and Spare Parts Ordering Policy for Efficient Support of Maintenance and Repair of Equipment. In: Al-Begain, K., Fiems, D., Knottenbelt, W. (eds.) ASMTA 2010. LNCS, vol. 6148, pp. 321–338. Springer, Heidelberg (2010)
25. Nechval, N.A., Purgailis, M., Cikste, K., Berzins, G., Nechval, K.N.: Optimization of Statistical Decisions via an Invariant Embedding Technique. In: Proceedings of the World Congress on Engineering (WCE 2010), London, U.K. Lecture Notes in Engineering and Computer Science, pp. 1776–1782 (2010)
26. Nechval, N.A., Purgailis, M., Cikste, K., Nechval, K.N.: Planning Inspections of Fatigued Aircraft Structures via Damage Tolerance Approach. In: Proceedings of the World Congress on Engineering (WCE 2010), London, U.K. Lecture Notes in Engineering and Computer Science, pp. 2470–2475 (2010)
27. Nechval, N.A., Purgailis, M.: Improved State Estimation of Stochastic Systems via a New Technique of Invariant Embedding. In: Myers, C. (ed.) Stochastic Control, pp. 167–193. Sciyo, Croatia, India (2010)
28. Nechval, N.A., Purgailis, M., Nechval, K.N., Rozevskis, U.: Optimization of Prediction Intervals for Order Statistics Based on Censored Data. In: Proceedings of the World Congress on Engineering (WCE 2011), London, U.K. Lecture Notes in Engineering and Computer Science, pp. 63–69 (2011)

Author Index