Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

AN EFFICIENT MCMC ALGORITHM FOR CONTINUOUS PH DISTRIBUTIONS

Ryo Watanabe
Hiroyuki Okamura
Tadashi Dohi

Department of Information Engineering,
Graduate School of Engineering, Hiroshima University
1–4–1 Kagamiyama, Higashi-Hiroshima 739–8527 JAPAN

ABSTRACT

This paper proposes an MCMC (Markov chain Monte Carlo) algorithm for estimating continuous phase-type distributions (CPHs). In Bayes estimation, it is well known that MCMC is one of the most useful and practical methods. The concrete MCMC algorithm for CPHs was developed by using Markov jump processes by Bladt et al. (2003). However, the existing MCMC algorithm spends much computation time in some cases. In this paper, we propose a new sampling algorithm which is based on uniformization technique and backward likelihood computation. The proposed algorithm is easier to implement and is more efficient in terms of computation time than the existing method.

1 INTRODUCTION

In the traffic modeling, discrete-time and continuous-time Markov chains (D/CTMCs) are popular approaches to evaluate and estimate system performance indexes. In particular, since time-homogeneous DTMC and CTMC are mathematically tractable, they are widely applied to model-based queueing and reliability analysis in practice (Bolch et al. 2006).

Phase-type (PH) distribution provides a non-negative random variable which is defined as a time-homogeneous DTMC or CTMC process, and can approximate any distribution with any precision (Asmussen and Koole 1993). Therefore, it is frequently used to represent service and failure time distributions in queueing and reliability analysis. On the other hand, when we deal with the performance evaluation of practical systems which is described by PH distributions, it becomes a problem how to determine parameters of PH distributions so that they fit to read data. Since PH distributions generally have the large number of parameters, the parameter estimation of PH distributions causes the computational difficulty compared to the parameter estimation for well-known statistical distributions such as normal distribution.

In the past literature, there are two major approaches to estimate PH parameters; moment matching and maximum likelihood (ML) estimation. Moment matching determines PH parameters so that the theoretical moments equate to the empirical ones. The accuracy of moment matching depends on how many moments are used for the PH fitting. Many papers concern moment matching methods using only the first few moments (van der Heijden 1988; Johnson and Taaffe 1989; Johnson and Taaffe 1990; Johnson 1993; Bosch et al. 2000; Telek and Heindl 2002; Bobbio et al. 2003; Bobbio et al. 2005; Osogami and Harchol-Balter 2006). van de Liefvoort (1990) and Telek and Horváth (2007) discussed a general method to use up to n moments for the fitting. On the other hand, ML estimation is to find the parameters maximizing the probability that observed data is drawn from the original model. The ML estimation for PH distributions was also discussed in several papers (Bobbio and Cumani 1992; Bobbio and Telek 1994; Asmussen et al. 1996; Thümmler et al. 2006; Panchenko and Thümmler 2007; Okamura et al. 2011).

The above two approaches give point estimates of PH parameters, and thus it is difficult to evaluate the uncertainty of estimated parameters in these schemes. The performance evaluation of systems with

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

uncertainty is often required in practical situations. For instance, in the area of reliability and traffic modeling, failure probabilities or loss probabilities of packets should be estimated with extremely high accuracy. In such situation, the uncertainty may considerably affect the system performance indexes.

This paper focuses on Bayes estimation for continuous PH distributions, which are defined by CTMCs, to take account of the uncertainty of estimated parameters. The idea behind Bayes estimation is to regard model parameters as random variables and it provides probability mass or density functions of model parameters, called posterior distributions, as estimation results. Thus the uncertainty of estimated parameters is evaluated by the variance of posterior distributions in Bayes estimation. However, it is well known that computation cost to obtain posterior distributions is comparatively higher than point estimation such as moment matching and ML estimation. To overcome this problem, Markov chain Monte Carlo (MCMC) is often applied to getting posterior distributions approximately with low computation cost.

MCMC is a versatile approach to approximate the posterior distribution in Bayes estimation. MCMC uses random samples drawn from posterior distributions, instead of analytical forms of them. To obtain such samples, MCMC builds a Markov chain whose stationary distribution becomes the posterior distribution. Bladt et al. (2003) proposed a concrete MCMC algorithm in Bayes estimation for continuous PH distributions. However, since their algorithm is based on an acceptance-rejection sampling method (Ross 2000), it spends much computation time to generate many samples as candidates. It degrades the computational efficiency in some cases.

This paper presents another MCMC algorithm to obtain parameter samples from posterior distribution in Bayes estimation for continuous PH distributions. Our idea is to use the uniformization and backward likelihood computation. The uniformization is known as an efficient method to compute transient probability vectors of CTMCs numerically, and it provides a DTMC representation from a CTMC. Moreover, the backward likelihood computation is used in some of statistical models such as the forward-backward algorithm of hidden Markov models (Baum et al. 1970). Specifically, after applying the uniformization to the original continuous PH distribution, we compute the backward likelihood computation using the DTMC obtained from the uniformization. Since we can avoid the acceptance-rejection method by applying the uniformization and backward likelihood computation, we construct an efficient MCMC algorithm in terms of computation time.

The paper is organized as follows. In Section 2, we introduce continuous PH distributions with their definition. Section 3 presents Bayes estimation and general MCMC algorithms. In Section 4, we first introduce the existing MCMC algorithm by Bladt et al. (2003), and propose our MCMC algorithm using the uniformization and backward likelihood computation in Section 4.2. Section 5 is devoted to numerical experiments to compare the proposed MCMC algorithm with the existing one in terms of computation time. In Section 6, we give some remarks on the proposed method and point to future research directions.

2 CONTINUOUS PH DISTRIBUTIONS

A continuous PH distribution (CPH) is defined as the time to absorption in a CTMC with one absorbing state. Let Q denote an infinitesimal generator matrix of the CTMC. Without loss of generality, Q is assumed to be partitioned as follows:

$$Q = \left(\begin{array}{c|c} T & \xi \\ \hline 0 & 0 \end{array} \right), \tag{1}$$

where T and ξ correspond to transition rates among transient states and exit rates from transient states to the absorbing state, respectively. Let a row vector π be an initial probability vector over the transient states. The probability density function (p.d.f.) of the CPH is given by

$$f_{PH}(t) = \pi \exp(Tt)\xi, \tag{2}$$

where the column vector ξ is often called an exit vector which represents transition rates to the absorbing state, and is given by $\xi = -Te$. We assume that π, T and ξ are of finite size and that the number of

transient states is m. The transient states are called phases in this paper. Moreover, we define π_i, ξ_i and $\lambda_{i,j}$ as the i-th elements of π, ξ and the (i,j)-element of T, respectively.

The CPH is classified into several probability distributions according to its structure of the matrix T (Thümmler et al. 2006). In particular, acyclic CPH (ACPH) is an important class which includes well-known distributions such as Erlang, hypoexponential and hyper Erlang distributions. In addition, ACPH is mathematically more tractable than general CPH. Cumani (1982) derived three canonical forms of ACPH. All of the canonical forms have $2m-1$ free parameters and any of ACPH can be reconfigured to one of the canonical forms. Specifically, CF1 (canonical form 1) is defined by

$$\pi = (\ \pi_1 \quad \pi_2 \quad \cdots \quad \pi_m\ ), \quad T = \begin{pmatrix} -\zeta_1 & \zeta_1 & & & \\ & -\zeta_2 & \zeta_2 & & \\ & & \ddots & \ddots & \\ & & & -\zeta_{m-1} & \zeta_{m-1} \\ & & & & -\zeta_m \end{pmatrix}, \quad \xi = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \zeta_m \end{pmatrix}. \tag{3}$$

where $\pi_i \geq 0$, $\sum_{i=1}^{m} \pi_i = 1$, and $\zeta_1, \ldots, \zeta_m$ are transition rates with constraints $0 < \zeta_1 \leq \cdots \leq \zeta_m$. When the parameter constraint does not hold, the resulting PH distribution is reduced to a bidiagonal CPH (He and Zhang 2008). Thus CF1 is also called the ordered bidiagonal CPH.

As shown in Eqs. (2), the p.d.f. of CPH includes a matrix exponential, i.e., it requires the transient analysis of CTMC. The uniformization (Reibman and Trivedi 1988) is one of the most effective methods to compute transient probability vectors of CTMCs. Let μ be a value that is greater than the maximum value of absolute diagonal elements of T, i.e., $\mu > \max_i |\lambda_{i,i}|$. Then the p.d.f. of CPH can be rewritten in the form:

$$f_{PH}(t) = \sum_{n=0}^{\infty} \frac{\mu^{n+1} t^n}{n!} e^{-\mu t} \pi P^n \nu, \tag{4}$$

where the matrix P is given by $P = I + T/\mu$ using the identity matrix I, and $\nu = \xi/\mu$. In Eq. (4), $\pi P^n \nu$ is the probability mass function (p.m.f.) of a discrete PH distribution (DPH) which represents an absorption time in a DTMC. Also, $\mu^{n+1} t^n e^{-\mu t}/n!$ corresponds to a p.d.f. of Erlang distribution with shape and scale parameters $n+1$ and μ, respectively. Hence the uniformization form can be regarded as a mixture of DPH and Erlang distribution (Kijima 1997).

3 BAYES ESTIMATION

Bayes estimation is a popular method to obtain the model parameters from empirical data. The key idea behind Bayes estimation is to regard model parameters as random variables. Then, under given prior information on model parameters, we provide the update formula of the information by using observed information. Concretely, let θ and $\mathscr{D}$ denote a parameter vector to be estimated and observed data, respectively. Then the update formula can be derived from Bayes theorem,

$$p(\theta|\mathscr{D}) = \frac{p(\mathscr{D}|\theta)p(\theta)}{C} \propto p(\mathscr{D}|\theta)p(\theta), \tag{5}$$

where $p(\cdot)$ is a probability mass or density function. Thus $p(\mathscr{D}|\theta)$ corresponds to the likelihood that the data $\mathscr{D}$ is observed under the given parameter vector θ, and $p(\theta)$ is the prior information (knowledge) on the parameter vector. Also $\propto$ denotes an operator representing a proportional relationship, which is used to omit a normalizing constant C. Equation (5) means that the prior information on parameter vector $p(\theta)$ is updated to the posterior distribution $p(\theta|\mathscr{D})$ by observing the data $\mathscr{D}$. In general, the estimator of parameters is given by the posterior distribution $p(\theta|\mathscr{D})$. Hence the variation of $p(\theta|\mathscr{D})$ means the uncertainty of estimated parameters.

Algorithm 1 MH algorithm

Step 1. Generate a candidate θ' from a proposal distribution $q(\theta'|\theta)$ with the current parameter vector θ, where $q(\theta'|\theta)$ is a p.m.f. or p.d.f. of θ' provided that θ is given.
Step 2. Compute the acceptance probability $a(\theta'|\theta)$

$$a(\theta'|\theta) = \min\left(\frac{p(\theta'|\mathscr{D})q(\theta|\theta')}{p(\theta|\mathscr{D})q(\theta'|\theta)}, 1\right).$$

Step 3. Generate a uniform random number over $[0,1]$.
Step 4. If $U \geq a(\theta'|\theta)$, then θ' is a new sample. Otherwise, θ is a new sample.

Algorithm 2 Gibbs sampling

for $i = 1 : |\theta|$ **do**
Generate θ_i' from a marginal distribution of $p(\theta_i'|\theta_{-i}, \mathscr{D})$ provided that θ_{-i} is given, where θ_{-i} means the parameter samples except for θ_i.
Replace an element θ_i of the current parameter vector with θ_i'.
end for

The difficulty of Bayes estimation comes from the computation of normalizing constant C. The normalizing constant C is needed to ensure the posterior distribution becomes a proper mass or density function, i.e., the normalizing constant is generally given by

$$C = \int p(\mathscr{D}|\theta)p(\theta)d\theta. \tag{6}$$

A straightforward approach for computing the normalizing constant is to use numerical integration such as trapezoidal rule and Gaussian quadrature. However, the integral of Eq. (6) is a multiple integral with respect to the parameter vector θ. It is quite difficult to compute the normalizing constant when the number of parameters is large. To overcome this computational problem, several approximation methods have been proposed in Bayes estimation.

Markov chain Monte Carlo (MCMC) is a powerful approach to obtain the posterior distribution in Bayes statistics. MCMC provides a set of parameter samples drawn from the posterior distribution, instead of computing the exact posterior distribution analytically. In MCMC, we build a Markov chain whose stationary distribution is consistent with the prior distribution, and iteratively simulate the Markov chain to get samples drawn from the stationary distribution.

Metropolis-Hastings (MH) method and Gibbs sampling are two of the most popular algorithms that build the Markov chain whose stationary distribution is the prior distribution. In the MH method, samples (candidates) are generated from a proposal distribution. The candidates are accepted as new samples with an acceptance probability which is computed from the likelihood and the proposal distribution. The Gibbs sampling is a special case of the MH method. In the Gibbs sampling, the proposal distributions are derived from marginal distributions of the posterior distribution, and then the acceptance probability is always 1. Algorithms 1 and 2 present one-step execution of the MH method and the Gibbs sampling, respectively. In Algorithm 2, $|\theta|$ denotes the number of parameters to be estimated.

Although the MH method can be applied to any types of models, the efficiency of the algorithm strongly depends on the selected proposal distribution. If the proposal distribution is far from the posterior distribution, many candidates are rejected at Step 4. Therefore it needs many iterations of one-step execution until the Markov chain converges to the stationary distribution. On the other hand, the Gibbs sampling is more effective to generate samples. However, it is rare to obtain analytically simple forms of the marginal distributions, and the marginal distributions becomes complex forms in many cases. In such cases, we should carefully consider how to generate samples drawn from complicated forms on a case-by-case basis.

In the MCMC scheme, one-step executions of Algorithms 1 and 2 are performed iteratively to collect parameter samples as MCMC sequences. When $\theta^{(1)},\ldots,\theta^{(M)}$ are parameter samples from M steps MCMC sequences, a point estimate and accuracy (uncertainty) of estimator are calculated as an arithmetic mean and variance of parameter samples, i.e.,

$$\mathrm{E}[\hat{\theta}] \approx \frac{1}{M}\sum_{l=1}^{M}\theta^{(l)}, \quad \mathrm{Var}(\hat{\theta}) \approx \frac{1}{M-1}\sum_{l=1}^{M}(\theta^{(l)}-\mathrm{E}[\hat{\theta}])^2. \tag{7}$$

Also, the predictive distribution is obtained as the estimated p.d.f. namely,

$$\hat{f}(t) = \int f(t;\theta)p(\theta|\mathscr{D})d\theta \approx \frac{1}{M}\sum_{l=1}^{M} f(t;\theta^{(l)}). \tag{8}$$

4 MCMC ALGORITHMS FOR CPH

4.1 MJP-Based MCMC Algorithm

This section presents concrete MCMC algorithms for CPH. We first introduce an existing MCMC algorithm based on Markov jump process (MJP) by Bladt et al. (2003).

Consider IID (independent and identically distributed) samples $\mathscr{D} = \{t_1,\ldots,t_K\}$ drawn from a CPH. Then the likelihood function $p(\mathscr{D}|\theta)$ is given by

$$p(\mathscr{D}|\theta) = \prod_{k=1}^{K}\pi\exp(Tt_k)\xi, \tag{9}$$

where $\theta = (\pi, T, \xi)$. As seen in the above equation, the likelihood includes the matrix exponential. Thus it is not practical to make parameter samples from the above likelihood function directly. The idea behind the MJP-based MCMC algorithm is to generate unobserved samples regarding the underlying CTMC (phase process) which dominates the CPH.

Define the following unobserved variables $\mathscr{Z} = (B_i, S_i, N_{i,j}, A_i)$ under the IID samples $\mathscr{D}$:

B_i: the number of times that the phase process begins with the phase i.
S_i: total sojourn time of phase i.
$N_{i,j}$: the number of times that the phase process changes from the phase i to j.
A_i: the number of times that the phase process changes from the phase i to the absorbing state.

Let $\{J_k(t), t \geq 0\}$, $k = 1,\ldots,K$, be a phase process of CPH for the k-th sample. Then the unobserved values can be defined as follows.

$$B_i = \sum_{k=1}^{K}\chi(J_k(0)=i), \quad S_i = \sum_{k=1}^{K}\int_0^{t_k}\chi(J_k(t)=i)dt, \tag{10}$$

$$N_{i,j} = \sum_{k=1}^{K}\int_0^{t_k}\chi(J_k(t-)=i, J_k(t)=j)dt, \quad A_i = \sum_{k=1}^{K}\chi(J_k(t_k-)=i), \tag{11}$$

where $\chi(A)$ is an indicator function of the event A. The likelihood of the data $(\mathscr{D},\mathscr{Z})$ is given by

$$p(\mathscr{D},\mathscr{Z}|\theta) = \prod_{i=1}^{m}\pi_i^{B_i}\prod_{i=1}^{m}e^{-\lambda_{i,i}S_i}\prod_{i=1}^{m}\prod_{j=1,j\neq i}^{m}\lambda_{i,j}^{N_{i,j}}\prod_{i=1}^{m}\xi_i^{A_i}. \tag{12}$$

Algorithm 3 Gibbs sampling with data augmentation

Generate an unobserved sample $\mathscr{Z}$ from $p(\mathscr{Z}|\mathscr{D},\theta)$.
for $i = 1 : |\theta|$ **do**
  Generate θ_i' from a marginal distribution of $p(\theta_i'|\theta_{-i},\mathscr{D})$ provided that θ_{-i} is given, where θ_{-i} means the parameter samples except for θ_i.
  Replace an element θ_i of the current parameter vector with θ_i'.
end for

Algorithm 4 MJP-based MCMC algorithm

Step 0. Generate a sample path of CTMC, $\{J(s), 0 \le s < t\}$, which is not absorbed before time t.
Step 1. Generate a sample path of CTMC, $\{J'(s), 0 \le s < t\}$, which is not absorbed before time t.
Step 2. Generate a uniform random number U over $[0,1]$.
Step 3. If $U < \xi_{J(t-)}/\xi_{J'(t-)}$, then $\{J'(s), 0 \le s < t\}$ is accepted as a sample path during $[0,t)$. Otherwise, $\{J(s), 0 \le s < t\}$ is used as a sample path during $[0,t)$.

Suppose that the prior distributions of π, $\lambda_{i,j}$ and ξ_i are Dirichlet and gamma distributions, namely,

$$p(\pi) = \text{Dirichlet}(\pi; u_1, \ldots, u_m) = \frac{\Gamma(\sum_{i=1}^m u_i)}{\prod_{i=1}^m \Gamma(u_i)} \prod_{i=1}^m \pi_i^{u_i - 1}, \tag{13}$$

$$p(\lambda_{i,j}) = \text{Gamma}(\lambda_{i,j}; \alpha_{i,j}, \beta_i) = \frac{\beta_i^{\alpha_{i,j}} \lambda_{i,j}^{\alpha_{i,j}-1} e^{-\beta_i \lambda_{i,j}}}{\Gamma(\alpha_{i,j})}, \tag{14}$$

$$p(\xi_i) = \text{Gamma}(\xi_i; \alpha_{i,0}, \beta_i), \tag{15}$$

where u_i, $\alpha_{i,j}$, β_i are hyper parameters of the prior distributions. From Eq. (12), the posterior distributions are explicitly given by

$$p(\pi|\mathscr{D},\mathscr{Z}) = \text{Dirichlet}(\pi; u_1 + B_1, \ldots, u_m + B_m), \tag{16}$$

$$p(\lambda_{i,j}|\mathscr{D},\mathscr{Z}) = \text{Gamma}(\lambda_{i,j}; \alpha_{i,j} + N_{i,j}, \beta_i + S_i), \tag{17}$$

$$p(\xi_i|\mathscr{D},\mathscr{Z}) = \text{Gamma}(\xi_i; \alpha_{i,0} + A_i, \beta_i + S_i). \tag{18}$$

Since $\mathscr{Z}$ is unobservable, Bladt et al. (2003) applied the data augmentation technique (Tanner and Wong 1987). The data augmentation is a kind of Gibbs sampling where the unobservable variables are regarded as model parameters. Algorithm 3 presents one-step execution of the Gibbs sampling with data augmentation. In the algorithm, $p(\mathscr{Z}|\mathscr{D},\theta)$ is a p.d.f. or p.m.f. of unobserved sample provided that the data $\mathscr{D}$ and model parameter vector θ are given:

$$p(\mathscr{Z}|\mathscr{D},\theta) = \frac{p(\mathscr{D},\mathscr{Z}|\theta)}{\int p(\mathscr{D},\mathscr{Z}|\theta) d\mathscr{Z}}. \tag{19}$$

In the case of CPH estimation, sampling from the above p.d.f. is equivalent to making K CTMC sample paths which are absorbed at $t_1, \ldots, t_K$. It is not easy to make such sample paths from the CTMC directly.

Bladt et al. (2003) also proposed a sampling algorithm based on MJP shown in Algorithm 4. Algorithm 4 presents concrete procedures to generate a CTMC sample path which goes to the absorbing state at time t, provided that the CTMC parameters are given. In Step 0 and Step 1, we should make sample paths which are not absorbed before time t. Bladt et al. (2003) suggested the acceptance-rejection method to make such sample paths in Step 0 and 1. That is, we make a CTMC sample path without any constraints from the starting time. If the path is absorbed before time t, we retry to make a path from the starting time.

Algorithm 4 comes from the MH method for two MJPs. In fact, the unobserved data $\mathscr{Z}$ can be computed from the CTMC sample paths by executing Algorithm 4 for each $t_1, \ldots, t_K$. However, it is pointed out that the sampling algorithm in Step 0 and Step 1 is ineffective if the absorbing time t is large.

4.2 Uniformization-Based MCMC Algorithm

Next we propose an efficient MCMC algorithm for CPH based on the uniformization technique and backward likelihood computation. As mentioned before, the uniformization provides a mixture of DPH and Erlang distributions as another representation of CPH. This implies that the unobserved data of CPH can be replaced by sample paths of the DPH. This is a basic idea of our algorithm.

Consider IID samples $\mathscr{D} = \{t_1, \ldots, t_K\}$ drawn from a CPH, and define the following unobserved data in the uniformized CPH:

R_k: the number of steps until the DTMC process goes to the absorbing state for the sample t_k.
B_i: the number of times that the DTMC process begins with the phase i.
$N_{i,j}$: the number of times that the DTMC process changes from the phase i to j.
A_i: the number of times that the DTMC process changes from the phase i to the absorbing state.

For the notational convenience, we define two unobserved data $\mathscr{R} = (R_1, \ldots, R_K)$ and $\mathscr{Z} = (B_i, N_{i,j}, A_i)$. Then the likelihood for $(\mathscr{D}, \mathscr{Z}, \mathscr{R})$ is given by

$$p(\mathscr{D}, \mathscr{Z}, \mathscr{R}|\theta) = \prod_{k=1}^{K} \frac{\mu^{R_k+1} t_k^{R_k}}{R_k!} e^{-\mu t_k} \prod_{i=1}^{m} \pi_i^{B_i} \prod_{i=1}^{m} \prod_{j=1}^{m} \eta_{i,j}^{N_{i,j}} \prod_{i=1}^{m} \nu_i^{A_i}, \tag{20}$$

where $\eta_{i,j}$ and ν_i are the (i,j)-element of P and the i-th element of ν, respectively. It should be noted that the uniformization factor μ becomes a model parameter. Similar to Section 4.1, we suppose that the prior distributions of μ, π, $\eta_{i,j}$ and ν_i are given as follows.

$$p(\mu) = \text{Gamma}(\mu; \alpha_\mu, \beta_\mu), \tag{21}$$
$$p(\pi) = \text{Dirichlet}(\pi; u_1, \ldots, u_m), \tag{22}$$
$$p(\eta_i) = \text{Dirichlet}(\eta_i; \alpha_{i,1}, \ldots, \alpha_{i,m}, \alpha_{i,0}), \tag{23}$$

where $\eta_i = (\eta_{i,1}, \ldots, \eta_{i,m}, \nu_i)$ and α_μ, β_μ, u_i and $\alpha_{i,j}$ are hyper parameters of the prior distributions. Thus we have the following closed forms of the posterior distributions:

$$p(\mu|\mathscr{D}, \mathscr{Z}, \mathscr{R}) = \text{Gamma}(\mu; \alpha_\mu + K + \textstyle\sum_{k=1}^{K} R_k, \beta_\mu + \sum_{k=1}^{K} t_k), \tag{24}$$
$$p(\pi|\mathscr{D}, \mathscr{Z}, \mathscr{R}) = \text{Dirichlet}(\pi; u_1 + B_1, \ldots, u_m + B_m), \tag{25}$$
$$p(\eta_i|\mathscr{D}, \mathscr{Z}, \mathscr{R}) = \text{Dirichlet}(\eta_i; \alpha_{i,1} + N_{i,1}, \ldots, \alpha_{i,m} + N_{i,m}, \alpha_{i,0} + A_i). \tag{26}$$

Using the above unobserved data, we can also develop the MCMC algorithm with data augmentation for the uniformized CPH, which is the same scheme as Algorithm 3. Then the problem is how to make unobserved samples $\mathscr{R}$ and $\mathscr{Z}$.

First we discuss the sampling of R_k. Since the marginal distribution with respect to R_k is given by

$$p(R_k|\mathscr{D}, \mathscr{Z}, \theta) \propto \frac{\mu^{R_k+1} t_k^{R_k} e^{-\mu t_k}}{R_k!} \pi P^{R_k} \nu, \tag{27}$$

it is not easy to draw a sample from the above distribution. Thus we apply the MH method to generate R_k. Concretely, we use the following proposal distribution which is a Poisson p.m.f.

$$q(R_k) = \frac{\mu^{R_k+1} t_k^{R_k} e^{-\mu t_k}}{R_k!}. \tag{28}$$

Algorithm 5 Sampling of an absorbing DTMC path

Step 1 Compute the backward likelihoods for each $k = R, \ldots, 1$

$$b_k = Pb_{k+1}, \quad b_{R+1} = \nu.$$

Step 2. Generate an initial phase J_0 which is drawn from a multivariate Bernoulli distribution with $(\pi_1 b_{1,1}, \ldots, \pi_m b_{1,m})$.
Step 3. Generate a path $\{J_k, k = 1, \ldots, R\}$ iteratively by using Step 3-1.
Step 3-1. Generate a phase at the k-th step, J_k, which is drawn from a multivariate Bernoulli distribution with $(\eta_{J_{k-1},1} b_{k+1,1}, \ldots, \eta_{J_{k-1},m} b_{k+1,m})$.

From the above proposal distribution, the acceptance probability for a candidate R'_k is given by

$$a(R'_k | R_k) = \min\left(\frac{\pi P^{R'_k} \nu}{\pi P^{R_k} \nu}, 1\right). \tag{29}$$

Next we consider a method to generate a sample path of DTMC which is absorbed at the step R. To generate such sample paths efficiently, we propose the method using the backward likelihood computation. Algorithm 5 shows our algorithm to generate a DTMC path which is absorbed at the R-th step with the backward likelihood computation. This is a variant of forward-backward algorithm used in some of statistical models such as hidden Markov model (Baum et al. 1970). Although this method requires the computation of backward likelihoods, it is not necessary to execute the acceptance-rejection method. Thus the computational efficiency is improved from the MJP-based method by Bladt et al. (2003).

Finally, Algorithm 6 presents pseudo code of our algorithm to generate unobserved data, provided that $(R_1, \ldots, R_K)$ are given as the outputs of the previous MCMC execution. The notation $\sim$ represents generating a random number from the left-hand side p.m.f. or p.d.f., and $\leftarrow$ means substitution from left to right. Also, Poisson(μ) is a p.m.f. of Poisson distribution with mean μ, Uniform$(0,1)$ is a p.d.f. of uniform distribution over $[0,1]$ and MBernoulli$(c_1, \ldots, c_m)$ is a p.m.f. of the multivariate Bernoulli distribution with the (non-normalized) probability vector $(c_1, \ldots, c_m)$.

5 NUMERICAL EXPERIMENTS

In numerical experiments, we compare our MCMC algorithm with MJP-based approach by Bladt et al. (2003) in terms of computation speed. We generate 1,000 samples from the following four CPHs:

$$\text{PH2STF:} \quad \pi = (0.3\ 0.7), \quad T = \begin{pmatrix} -0.01 & 0.01 \\ 0 & -0.1 \end{pmatrix}, \quad \xi = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} \tag{30}$$

$$\text{PH2NSF:} \quad \pi = (0.3\ 0.7), \quad T = \begin{pmatrix} -0.1 & 0.1 \\ 0 & -0.1 \end{pmatrix}, \quad \xi = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} \tag{31}$$

$$\text{PH2GEN:} \quad \pi = (0.3\ 0.7), \quad T = \begin{pmatrix} -1.0 & 0.2 \\ 0.8 & -1.0 \end{pmatrix}, \quad \xi = \begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix} \tag{32}$$

$$\text{PH5:} \quad \pi = (0.2\ 0.2\ 0.2\ 0.2\ 0.2), \tag{33}$$

$$T = \begin{pmatrix} -0.1 & 0.1 & & & \\ & -0.1 & 0.1 & & \\ & & -0.1 & 0.1 & \\ & & & -0.1 & 0.1 \\ & & & & -0.1 \end{pmatrix}, \quad \xi = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.1 \end{pmatrix}. \tag{34}$$

Algorithm 6 Uniformization-based MCMC algorithm

for $k = 1 : K$ **do**
  $R'_k \sim \text{Poisson}(\mu t_k)$.
end for
$\overline{R} \leftarrow \max(R_1, \ldots, R_K, R'_1, \ldots, R'_K)$
$b_{\overline{R}+1} \leftarrow \nu$
for $k = \overline{R} : 1$ **do**
  $b_k \leftarrow P b_{k+1}$
end for
for $k = 1 : K$ **do**
  $U \sim \text{Uniform}(0, 1)$
  if $\pi b_{R'_k} / \pi b_{R_k} < U$ **then**
    $R_k \leftarrow R'_k$.
  else
    $R_k \leftarrow R_k$.
  end if
  $J_{k,0} \sim \text{MBernoulli}(\pi_1 b_{\overline{R}-R_k+1,1}, \ldots, \pi_m b_{\overline{R}-R_k+1,m})$
  for $l = 1 : R_k$ **do**
    $J_{k,l} \sim \text{MBernoulli}(\eta_{J_{k,l-1},1} b_{\overline{R}-R_k+k+1,1}, \ldots, \eta_{J_{k,l-1},m} b_{\overline{R}-R_k+k+1,m})$
  end for
end for

The first three CPHs are 2-phase CPHs. PH2STF and PH2NSF are canonical forms and PH2GEN is a general form of CPH. In addition, the stiffness of the underlying CTMC of PH2STF is higher than that of PH2NSF.

For each 1,000 samples of PH2STF, PH2NSF, PH2GEN and PH5, we execute MJP-based MCMC algorithm and our MCMC algorithm until 5,000 parameter samples are obtained. Both MCMC algorithms are implemented by R (The R- Project for Statistical Computing, http://www.r-project.org/), and the initial values of MCMC algorithms are set as original CPH parameters given in Eqs. (30)– (34). Moreover, the hyper parameters are set as $u_\cdot = 1$, $\alpha_{\cdot,\cdot} = 1$ and $\beta_\cdot = 1$ in all the cases.

Table 1 presents computation time (user time) in seconds until 5,000 parameter samples are obtained in the AMD Opteron 2.3GHz processor with a single thread. From the table, it can be found that the proposed uniformization-based MCMC algorithm is much faster than the existing MJP-based method. In all the cases, the proposed method is more than 10 times faster as the MJP-based MCMC algorithm. Particularly, in the case of PH2STF, our method is 24 times faster as the MJP-based method. The MJP-based algorithm generates a candidate as a CTMC path which is not absorbed before an observation time. However, Since PH2STF is dominated by a stiff Markov chain, it has a higher coefficient of variation compared to the other cases. Therefore, the samples of PH2STF include a few large values. As mentioned in Section 4.1, the acceptance-rejection method in Step 0 and Step 1 of Algorithm 4 is ineffective in the case where the absorbing time is large, namely, many rejections occur in the acceptance-rejection method of the MJP-based algorithm. Hence the MJP-based algorithm spent much computation time in the case of PH2STF. Also we find that the computation time of MJP-based algorithm in PH2NSF is faster than that in PH2GEN. This is the reason why possible CTMC paths of PH2NSF are less than those in PH2GEN because the phase structure of PH2NSF is restricted to CF1. In this way, computation time of the MJP-based algorithm strongly depends on the efficiency of the acceptance-rejection method in Step 0 and Step 1 of Algorithm 4.

On the other hand, in our method, we generate DTMC paths by using the backward likelihoods. This ensures that a generated DTMC path is never rejected. However, the computation cost of our method depends on the backward likelihood computation. In general, it requires much computation cost in the

Table 1: Computation time for MCMC algorithms.

CPH	Computation time (seconds) MJP	Uniformization
PH2STF	22389.0	896.9
PH2NSF	7043.2	456.5
PH2GEN	6273.9	570.1
PH5	50035.4	808.9

Table 2: Z scores of MCMC parameter sequences (PH2STF).

	MJP			Uniformization		
iteration	π_1	$\lambda_{1,1}$	$\lambda_{2,2}$	π_1	$\lambda_{1,1}$	$\lambda_{2,2}$
1000	0.50	1.27	0.10	2.41	2.52	2.45
2000	0.98	1.68	0.49	2.05	1.88	2.32
3000	1.34	1.43	0.67	1.01	1.06	1.32
4000	0.76	0.41	-0.02	0.87	0.68	1.28
5000	0.00	0.05	-0.36	0.72	0.73	1.07

cases where CPH is dominated by stiff and large Markov chains. Therefore, in the table, we also find that the computation times in PH2STF and PH5 become almost double than that in PH2NSF.

Next we examine convergence of MCMC parameter sequences. In general, as the number of parameters to be estimated increases, so does the large number iterations required for MCMC to converge to the stationary distribution. As seen in Section 4.2, in our method, we regard a uniformization factor μ as a model parameter to be estimated. Thus our method requires more MCMC iterations than the MJP-based algorithm.

In this experiment, Geweke diagnostic (Geweke 1992) is used to check the convergence of MCMC parameter sequence. Geweke diagnostic is an application of Z test to time series data and computes Z scores for partitioned time spans. If the absolute Z scores are within 1.96, the sequence is considered converged (Strictly speaking, the hypothesis that the sequence does not converge is not rejected.).

In this experiment, we did not set burn-in periods because we give true parameters as initial parameters. Table 2 shows Z scores of MCMC parameter sequences for PH2STF generated by MJP-based and uniformization-based algorithms, which were computed by CODA (Convergence Diagnostic and Output Analysis software) for each 1,000, 2,000, 3,000 4,000 and 5,000 MCMC iterations. Since PH2STF consists of only three non-redundant parameters, we present Z scores for parameters π_1, $\lambda_{1,1}$ and $\lambda_{2,2}$ in the table. Moreover, MCMC sequences did not converge even for 5,000 iterations in other PH2NSF, PH2GEN and PH5 cases, and thus we discuss only the PH2STF case. In Table 2, MCMC sequence of MJP-based algorithm has already converged at 1,000 iterations. On the other hand, our method needs 3,000 iterations for the sequence to converge to the stationary distribution. It indicates that our method requires 3 times larger as MCMC sequences than the MJP-based method. However, since the computation speed of uniformization-based method is 24 times faster in the case of PH2STF, it is concluded that the proposed approach is faster even if we execute the uniformization-based MCMC algorithm until the sequence converges.

6 CONCLUSIONS

This paper has presented an improved MCMC algorithm for CPH in terms of computation speed. More precisely, we have applied the uniformization and the backward likelihood computation to enhance the computation speed of generating latent phase processes. Although the proposed method has the disadvantage of requiring longer MCMC sequences than the existing method, the computation speed of our method is still faster even if we generate the long MCMC sequences. In the numerical experiment, we have applied

both proposed and existing methods to 4 types of CPH parameter estimation. As a result, we have examined that our approach was effective to obtain the MCMC parameter sequences even if the phase process is represented by stiff-type and 5-phase CPHs. However, since our method includes the backward likelihood computation, the computational problem for stiffness and large phases has not completely been solved.

In future, we will apply generalized ensemble algorithms such as replica-exchange method to the uniformization-based MCMC algorithm to reduce MCMC sequences needed for the convergence. Also, to overcome the stiffness from the viewpoint of computation time, we develop a parallel variant of MCMC algorithms.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), Grant No. 21510167 (2009–2011), Grant No. 23500047 (2011–2013) and Grant No. .23510171 (2011–2013).

REFERENCES

Asmussen, S., and G. Koole. 1993. "Marked point processes as limits of Markovian arrival streams". *Journal of Applied Probability* 30:365–372.

Asmussen, S., O. Nerman, and M. Olsson. 1996. "Fitting phase-type distributions via the EM algorithm". *Scandinavian Journal of Statistics* 23 (4): 419–441.

Baum, L. E., T. Petrie, G. Soules, and N. Weiss. 1970. "A maximization technique occurring in the statistical analysis of probabilistic function of Markov chains". *Annals of Mathematical Statistics* 41 (1): 164–171.

Bladt, M., A. Gonzalez, and S. L. Lauritzen. 2003. "The estimation of phase-type related functionals using Markov chain Monte Carlo methods". *Scandinavian Actuarial Journal* 4:280–300.

Bobbio, A., and A. Cumani. 1992. "ML estimation of the parameters of a PH distribution in triangular canonical form". In *Computer Performance Evaluation*, edited by G. Balbo and G. Serazzi, 33–46. Elsevier Science Publishers.

Bobbio, A., A. Horváth, M. Scarpa, and M. Telek. 2003. "Acyclic discrete phase type distributions: Properties and a parameter estimation algorithm". *Performance Evaluation* 54:1–32.

Bobbio, A., A. Horváth, and M. Telek. 2005. "Matching Three Moments with Minimal Acyclic Phase Type Distributions". *Stochastic Models* 21 (2/3): 303–326.

Bobbio, A., and M. Telek. 1994. "A benchmark for PH estimation algorithms: Result for acyclic-PH". *Stochastic Models* 10:661–677.

Bolch, G., S. Greiner, H. de Meer, and K. S. Trivedi. 2006. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. 2nd ed. New York: John Wiley & Sons.

Bosch, P. M. V., C. Dietz, and E. A. Pohl. 2000. "Moment matching using a family of phase-type distributions". *Stochastic Models* 16:391–398.

Cumani, A. 1982. "On the canonical representation of homogeneous Markov Processes modelling failure-time distributions". *Microelectronics and Reliability* 22:583–602.

Geweke, J. 1992. "Evaluating the accuracy of sampling-based approaches to calculating posterior moments". In *Bayesian Statistics*, edited by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, Volume 4, 169–194. Oxford: Clarendon Press.

He, Q.-H., and H. Zhang. 2008. "An algorithm for computing minimal Coxian representations". *INFORMS Journal on Computing* 20 (2): 179–190.

Johnson, M. A. 1993. "Selecting parameters of phase distributions: Combining nonlinear programming, heuristics, and Erlang distributions". *ORSA Journal on Computing* 5:69–83.

Johnson, M. A., and M. R. Taaffe. 1989. "Matching moments to phase distributions: Mixtures of Erlang distribution of common order". *Stochastic Models* 5:711–743.

Johnson, M. A., and M. R. Taaffe. 1990. "Matching moments to phase distributions: Density function shapes". *Stochastic Models* 6:283–306.

Kijima, M. 1997. *Markov Processes for Stochastic Modeling*. Chapman and Hall.

Okamura, H., T. Dohi, and K. S. Trivedi. 2011. "A refined EM algorithm for PH distributions". *Performance Evaluation* 68:938–954.

Osogami, T., and M. Harchol-Balter. 2006. "Closed Form Solutions for Mapping General Distributions to Minimal PH Distributions". *Performance Evaluation* 63 (6): 524–552.

Panchenko, A., and A. Thümmler. 2007. "Efficient phase-type fitting with aggregated traffic traces". *Performance Evaluation* 64:629–645.

Reibman, A. L., and K. S. Trivedi. 1988. "Numerical transient analysis of Markov models". *Computers and Operations Research* 15:19–36.

Ross, S. M. 2000. *Introduction to Probability Models*. 7th ed. Academic Press.

Tanner, M. A., and W. H. Wong. 1987. "The calculation of posterior distributions by data augmentation". *Journal of the American Statistical Association* 82 (398): 528–540.

Telek, M., and A. Heindl. 2002. "Matching Moments for Acyclic Discrete and Continuos Phase-Type Distributions of Second Order". *Int'l Journal of Simulation Systems, Science & Technology* 3 (3–4): 47–57.

Telek, M., and G. Horváth. 2007. "A minimal representation of Markov arrival processess and a moments matching method". *Performance Evaluation* 64 (9-12): 1153–1168.

Thümmler, A., P. Buchholz, and M. Telek. 2006. "A novel approach for phase-type fitting with the EM algorithm". *IEEE Transactions on Dependable and Secure Computing* 3 (3): 245–258.

van de Liefvoort, A. 1990. "The moment problem for continuous distributions". Technical report, University of Missouri, WP-CM-1990-02, Kansas City.

van der Heijden, M. C. 1988. "On the three-moment approximation of a general distribution by a Coxian distribution". *Probability in the Engineering and Informational Sciences* 2:257–261.

AUTHOR BIOGRAPHIES

RYO WATANABE received the B.S.E. of Engineering degrees from Hiroshima University, Japan, in 2012. He is a master course student in Department of Information Engineering, Graduate School of Engineering, Hiroshima University.

HIROYUKI OKAMURA received the B.S.E., M.S. and Dr. of Engineering degrees from Hiroshima University, Japan, in 1995, 1997 and 2001, respectively. In 1998 he joined the Hiroshima University as an Assistant Professor, and is now working as an Associate Professor in the Department of Information Engineering, Graduate School of Engineering from 2003. His research areas include Performance Evaluation, Dependable Computing and Applied Statistics.

TADASHI DOHI received the B.S.E., M.S. and Dr. of Eng. degrees from Hiroshima University, Japan, in 1989, 1991 and 1995, respectively. Since 2002, he has been working as a full Professor in the Department of Information Engineering, Graduate School of Engineering, Hiroshima University. His research areas include Reliability Engineering, Software Reliability, Dependable Computing and High Assurance Systems Design.