Markovian Arrival Process Parameter Estimation With Group Data

Hiroyuki Okamura, Member, IEEE, Tadashi Dohi, Member, IEEE, and Kishor S. Trivedi, Fellow, IEEE

Abstract—This paper addresses a parameter estimation problem of Markovian arrival process (MAP). In network traffic measurement experiments, one often encounters the group data where arrival times for a group are collected as one bin. Although the group data are observed in many situations, nearly all existing estimation methods for MAP are based on non-group data. This paper proposes a numerical procedure for fitting a MAP and a Markov-modulated Poisson process (MMPP) to group data. The proposed algorithm is based on the expectation-maximization (EM) approach and is a natural but significant extension of the existing EM algorithms to estimate parameters of the MAP and MMPP. Specifically for the MMPP estimation, we provide an efficient approximation based on the proposed EM algorithm. We examine the performance of proposed algorithms via numerical experiments and present an example of traffic analysis with real traffic data.

Index Terms—Expectation-maximization (EM) algorithm, group data, Markov-modulated Poisson process (MMPP), Markovian arrival process (MAP), maximum-likelihood (ML) estimation, network traffic.

I. INTRODUCTION

MARKOVIAN arrival process (MAP) was proposed in [1] and is widely used for probabilistic analysis of communication network traffic. When the MAP is used to represent an input stream in a queuing system, matrix-geometric methods [2], [3] can be applied to derive quantitative performance measures on waiting time and number of customers. For finite population systems, stochastic Petri nets can be used for the same purpose [4]–[6].

MAP has attractive properties from the viewpoint of stochastic point processes. It is one of the most general classes of stochastic counting processes that contains most of the commonly used arrival processes such as the Poisson process, the phase-type (PH) renewal process, and the Markov-modulated Poisson process (MMPP). Moreover, MAP is known to be dense [7], so it can approximate an arbitrary stochastic point process to a given degree of accuracy.

Another advantage of MAP is its ability to represent time correlation in arrival streams, as is commonly observed in the Internet traffic. Long-range dependence has received considerable attention in the area of communication networks since self-similar processes causing a long-range dependence have empirically been observed in the actual Ethernet traffic [8]. MAP is able to approximate such self-similar and long-range dependent traffic. Chaotic maps [9], fractional Brownian motion [10], and fractional auto-regressive integrated moving average [11] can also be utilized for modeling self-similar traffic with long-range dependence. Since MAP is a natural extension of the Poisson process, unlike other stochastic point processes mentioned above, it has the advantages of analytical tractability. Therefore, MAP-based modeling approaches are commonly used for traffic analysis [12], [13].

An important problem in MAP-based traffic modeling is the estimation of model parameters to fit observed traffic data. A general MAP with \( m \) phases has up to \( 2m^2 + m \) parameters. Clearly, parameter estimation of the MAP is more complex than that of the Poisson process, which has only one parameter.

In general, there are two approaches for estimating parameters of MAP: moment-based approach and likelihood-based approach. In the moment-based approach, one determines the model parameters so as to fit theoretical moments to empirical ones from observed data. Heffes and Lucantoni [14] provided an explicit formula for estimating the parameters of a two-state MMPP by using empirical moments of the number of arrivals. Anderson and Nielsen [15] proposed a fitting method for a superposition of 2-state MAPs using the Hurst parameter besides the moments. Yoshihara et al. [16] developed a moment-based estimation procedure for an MMPP with any number of states in order to represent self-similar traffic. Mitchell and Liefvoort [17] proposed a two-step method that deals with interarrival time data and lag correlation separately. The main advantage of such moment-based approaches over likelihood-based approaches is their low computational cost.

Maximum-likelihood (ML) estimation for MAP has posed difficulties until the mid 1990s. The principle of ML estimation is to find the parameters (ML estimates, or MLEs) that maximize the likelihood that the observed data occurs. Direct approaches are based on general-purpose maximization methods such as Newton’s method. The direct approach for MAP estimation with a large number of phases requires large-scale matrix computations. Expectation-maximization (EM) algorithm [18], [19] is one of the methods of computing MLEs effectively.

EM algorithm is a statistical framework to compute MLEs under incomplete data and is particularly useful for stochastic models with many parameters, such as a Gaussian mixture model. The first published EM algorithm for a family of MAPs was the forward-backward algorithm via a hidden Markov model (HMM) [20]. Deng and Mark [21] proposed an ML estimation for an MMPP by first converting it to

1063-6692/09/$26.00 © 2009 IEEE

Authorized licensed use limited to: Barbara Lange. Downloaded on August 22, 2009 at 00:56 from IEEE Xplore. Restrictions apply.
a Markov-modulated Bernoulli process (MMBP) and then using the forward-backward algorithm. Asmussen et al. [22] proposed an EM algorithm to estimate parameters of a PH distribution; their idea can be applied for the estimation of MMPP and MAP parameters with continuous-time data. Rydén [23] derived an EM algorithm to provide the exact MLEs of MMPP. Note that the EM algorithm by Rydén [23] is analogous to the forward-backward algorithm for HMM [20] as well as the EM algorithm for PH distribution [22].

Based on Rydén’s work, two enhancements of the EM algorithm are possible. One direction is to develop EM algorithms for the parameter estimation of a wider class of stochastic processes. Batch MAP (BMAP) [24] is one super-class of MMPP and MAP. Breuer [25] and Klemm et al. [13] discussed EM algorithms to estimate the parameters of BMAPs. The other direction is to reduce the computation time of the EM algorithm for MAP. Roberts et al. [26] proposed a scaling method and a computation method for matrix exponential to speed up Rydén’s EM algorithm. Klemm et al. [13] and Buchholz [27] reduced the time complexity of the EM algorithm by applying uniformization technique. Buchholz and Panchenko [28] and Horváth et al. [29] developed two-step fitting methods by combining the EM algorithm for PH distribution and the moment-based two-step method [17].

Many times in practice, only group data is available, as the exact arrival times may be unknown but are grouped into bins. All of the existing EM algorithms assume that exact arrival time data (that is, nongroup data) are available. From the statistical point of view, the estimation methods for nongroup data are completely different from those for group data because the likelihood function for the group data is expressed differently from that for the nongroup data. To the best of our knowledge, no estimation procedure for the group data has been discussed in the context of the EM algorithm for MAP, although we frequently encounter group data in practice.

Grouping data implies a statistical loss of information about exact arrival times. The missing data naturally causes increased computation time and loss of accuracy. Perhaps for this reason, existing papers have considered the estimation assuming exact arrival time data.

However, there are many situations where observable data are grouped; for example, network traffic data aggregated from some audit tools are represented as group data. Most performance-monitoring applications such as Windows Performance Monitor and SYSSTAT\(^1\) report observed statistics as group data for saving system resources. This paper considers such a situation where only group data are available and proposes two EM algorithms to provide exact and approximate MLEs for group data. Although the proposed EM algorithms, especially the algorithm providing exact MLEs, are relatively expensive in terms of computation cost than those using arrival time data, our algorithm has the attractive property that it reveals the exact estimates in the strict statistical sense. In fact, measuring the dependency of MLEs on the type of data is significant since the exact MLEs could be used as benchmarks for any (approximate) traffic estimation. Development of feasible estimation algorithms for MAP with group data is useful both from a practical perspective as well as from a statistical one. This motivates us to develop statistical estimation procedures for MAP with group data.

Moreover, this paper copes with generalized group data that are an extension of group data so that exact arrival time data can be accommodated as well. That is, the EM algorithm proposed here subsumes the existing EM algorithm [23] as a special case. This means that the proposed algorithm can handle both arrival time data and group data in the same framework.

This paper is organized as follows. Section II describes overviews of MAP and MMPP and their associated traffic analysis. In Section III, we mention the definition of generalized group data and propose an EM algorithm for MAP with the generalized group data. The proposed algorithm provides exact MLEs to MAP and MMPP. Section IV is devoted to approximating the MLEs of MMPP in the case where the group size is large. Finally, in Section V, we carry out numerical experiments to investigate the scalability of the proposed EM algorithm and to examine the accuracy of the approximate EM algorithm. Moreover, we present an example of traffic analysis based on real traffic data.

II. MARKOVIAN ARRIVAL PROCESS (MAP)

Definition

MAP is a counting process whose arrival rate is governed by a continuous-time Markov chain (CTMC). Let \( D_0 \) denote the infinitesimal generator of the underlying CTMC in the case of no arrivals, and let \( D_1 \) be the rate matrix in the case of an arrival leading to a (possible) state change of the CTMC. Consider \( m \) distinct states, called phases, in the CTMC with corresponding matrices \( D_0 \) and \( D_1 \).

\[
D_0 = \begin{pmatrix}
-\mu_{11} & \mu_{12} & \cdots & \mu_{1m} \\
\mu_{21} & -\mu_{22} & \cdots & \mu_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\mu_{m1} & \mu_{m2} & \cdots & -\mu_{mm}
\end{pmatrix}
\]

\[
D_1 = \begin{pmatrix}
\lambda_{11} & \lambda_{12} & \cdots & \lambda_{1m} \\
\lambda_{21} & \lambda_{22} & \cdots & \lambda_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{m1} & \lambda_{m2} & \cdots & \lambda_{mm}
\end{pmatrix}
\]

where \( \mu_{ij} = \sum_{j=1,j \neq i}^{m} \mu_{ij} + \sum_{j=1}^{m} \lambda_{ij} \). The phase process, including phase transitions caused by arrivals, is thus a CTMC with an infinitesimal generator \( D_0 + D_1 \).

Let \( \{N(t); t \geq 0\} \) and \( \{J(t); t \geq 0\} \) denote the number of arrivals during the time interval \([0,t]\) and the phase at time \( t \), respectively. Define matrix \( P_k(t) \) whose \((i,j)\)-element is given by

\[
[P_k(t)]_{i,j} = P(N(t) = k, J(t) = j \mid N(0) = 0, J(0) = i).
\]

This evolves as per the following differential-difference equations [2]:

\[
\frac{d}{dt} P_0(t) = P_0(t)D_0
\]

\[
\frac{d}{dt} P_k(t) = P_k(t)D_0 + P_{k-1}(t)D_1, \quad k = 1, 2, \ldots
\]
The initial phase is determined by an initial probability vector \( \pi = (\pi_1, \pi_2, \ldots, \pi_m) \) with \( \pi e = 1 \), where \( e \) is a column vector with all elements equal to 1.

MMPP is a Poisson process with its arrival rate modulated by a CTMC. In a mathematical sense, MMPP is a subclass of MAP where the matrix \( D_1 \) is a diagonal matrix. The essential difference between MAP and MMPP is whether or not the phase is potentially changed just after an arrival.

### A. Performance Measures

In the family of MAPs, lag \( k \) correlation is one of the most important measures related to the time dependency of MAPs. The lag \( k \) correlation captures the correlation between interarrival times. Let \( \mathbf{x}_n \) denote the steady-state probability of phases at arrival time instants, i.e., \( \mathbf{x}_n(-D_1)^{-1}D_1 = \mathbf{x}_n \) [29]. Then, the lag \( k \) correlation \( p_k \) is given by [29]

\[
p_k := \frac{x_n(-D_1)^{-k}D_1^{-1}(-D_0)^{-1}e - 1/\lambda_n^2}{2x_n(-D_1)^{-2}e - 1/\lambda_n^2}
\]

(6)

where \( \lambda_n = 1/(\mathbf{x}_n(-D_0)^{-1}e) \) is the average arrival rate. The MAP with a large value of lag \( k \) correlation for large \( k \) is known to approximate the long-range dependence well.

Consider a queueing system with a MAP arrival stream, generally distributed service time \( S \), and a single server [3], i.e., a MAP/G/1 queueing system. The traffic intensity \( \rho \) is defined as the expected number of arrivals during a single service time. Let \( \mathbf{x}_n \) be the steady-state probability vector at an arbitrary instant, i.e., \( \mathbf{x}_n(D_1 + D_0) = 0 \). Then, the traffic intensity is given by \( \rho = \mathbf{x}_n D_1 e[I[S]] \). Now, if the system has a finite capacity of size \( K \)—that is, the MAP/G/1/K queueing system—the effective traffic intensity is given by \( \rho_{\text{eff}} = \rho(1 - P_{\text{loss}}) \), where \( P_{\text{loss}} \) is the loss probability that an arrival encounters a full buffer. The effective traffic intensity can be used to compute the probability that the system is busy at an arbitrary instant. Define \( \mathbf{y}_k \), \( 0 \leq k \leq K \), as the probability vector of the phases where the queue length is \( k \) at an arbitrary instant. Since the effective traffic intensity can also be written as \( \rho_{\text{eff}} = 1 - \mathbf{y}_1 e / \rho \), the loss probability is obtained as \( P_{\text{loss}} = 1 - (1 - \mathbf{y}_1 e) / \rho \). Then, the probability that the buffer is full at an arbitrary instant is \( P_{\text{full}} = \mathbf{y}_K e \).

### III. Parameter Estimation for MAP

#### A. Brief Overview of the EM Algorithm

EM algorithm is an iterative method for the ML estimation with incomplete data [18], [19]. Let \( \mathcal{D} \) and \( \mathcal{U} \) respectively denote the observed and unobserved data. Assume that we wish to estimate a set of model parameters \( \theta \) given the observed data \( \mathcal{D} \). EM algorithm consists of two parts: 1) computing the expected log-likelihood function (LLF) for the complete data pair \( (\mathcal{D}, \mathcal{U}) \) given that only the data \( \mathcal{D} \) is observed, and 2) finding a parameter set \( \theta \) that maximizes the expected LLF. These two parts are mathematically formulated as

\[
\theta := \arg \max_{\theta} \mathbb{E}_{\mathcal{U}} \left[ \text{LLF}(\theta | \mathcal{D}, \mathcal{U}) | \mathcal{D} \right]
\]

(7)

where \( \mathbb{E}_{\mathcal{U}} \) is the expectation operator for the unobserved data \( \mathcal{U} \). In order to compute this expected LLF, we need a provisional set of parameters. That is, (7) implicitly gives an update formula of the parameters, and the parameters are updated until they converge to within a given tolerance. The E-step and the M-step in the EM algorithm correspond to the two parts: 1) computing the expected LLF, and 2) finding a set of parameters that maximizes the expected LLF.

#### B. Data Format

We introduce a generalized data format that can represent both the arrival time data and the usual group data. For this purpose, to the usual group data format we need to add the possibility that an arrival occurs at the end of each observation epoch. Thus, the format of generalized group data is given by

\[
\mathcal{D}_G := \{ (t_1, x_1, a_1), \ldots, (t_K, x_K, a_K) \}
\]

(8)

where \( t_k \) and \( x_k \) are time length of the \( k \)th period and the number of arrivals observed in the \( k \)th period, respectively. Also, \( a_k \) is the indicator variable for the event of an arrival at the end of the \( k \)th observation period. Let \( s_k \) denote cumulative time for the first \( k \) periods, i.e., \( s_k = \sum_{i=1}^{k} t_i \). Then, \( x_k \) is the number of arrivals during time interval \( (s_{k-1}, s_k) \), and \( a_k = 1 \) means that an arrival occurs at time \( s_k \). Otherwise, if \( a_k = 0 \), no arrival occurs at time \( s_k \). The total number of arrivals during the time interval \( (s_{k-1}, s_k) \) is thus \( x_k + a_k \).

By setting \( a_k = 0 \) for all \( k = 1, \ldots, K \), we get the data format for the usual group data. By setting \( a_k = 1 \) and \( x_k = 0 \) for all \( k \), we get the data format for the arrival time data. The generalized group data can also handle a mixture of arrival time data and group data.

#### C. M-Step Formulas

Consider an \( m \)-state MAP with generalized group data \( \mathcal{D}_G \) and their cumulative time sequence \( s_0 = 0 \) and \( s_k = \sum_{i=1}^{k} t_i \). We define the following unobserved (random) variables.

- \( B_k \) an indicator random variable for the event that the phase of the MAP is \( i \) at time \( s_0 = 0 \);
- \( t^{[k]}_{i,j} \) an indicator random variable for the event that an arrival occurs at time \( s_k \), leading to a phase transition from \( i \) to \( j \);
- \( z^{[k]}_i \) cumulative sojourn time in phase \( i \) during the time interval \( (s_{k-1}, s_k) \);
- \( M^{[k]}_{i,j} \) the number of phase transitions from \( i \) to \( j \) without arrivals during the time interval \( (s_{k-1}, s_k) \);
- \( y^{[k]}_{i,j} \) the number of arrivals leading to phase transitions from \( i \) to \( j \) during the time interval \( (s_{k-1}, s_k) \).

Define the parameter set \( \theta := \{ \pi_1, \mu, \lambda_1 \ldots, \lambda_{m} \} \) and the unobserved variables \( \mathcal{U} := \{ B_k, t_{i,j}^{[k]}, z^{[k]}_i, M^{[k]}_{i,j}, y^{[k]}_{i,j} \} \) for \( i, j = 1, \ldots, m \) and \( k = 1, \ldots, K \). Since the parameters \( \mu_{i,j} \) and \( \lambda_{i,j} \) are respective rates of phase transitions and arrivals in the MAP phase \( i \), we have the following MLEs under the complete data pair \( (\mathcal{D}_G, \mathcal{U}) \):

\[
\hat{\pi}_i = B_k, \quad \hat{\mu}_{i,j} = \frac{\sum_{k=1}^{K} M^{[k]}_{i,j}}{\sum_{k=1}^{K} z^{[k]}_i}, \quad \hat{\lambda}_{i,j} = \frac{\sum_{k=1}^{K} (y^{[k]}_{i,j} + t_{i,j}^{[k]})}{\sum_{k=1}^{K} z^{[k]}_i}.
\]

(9)
According to (7) and the aforementioned MLEs, update formulas (M-step formulas) of an EM algorithm for MAP are obtained as follows:

\[
\pi_i := \frac{E[B_i|D_G]}{\sum_k E[B_k|D_G]}, \quad i \neq j
\]

\[
\mu_{i,j} := \frac{\sum_k E[Z_{i,j}^k|D_G]}{\sum_k E[Z_i^k|D_G]}, \quad \lambda_{i,j} := \frac{\sum_k (E[Y_{i,j}^k|D_G] + E[U_{i,j}^k|D_G])}{\sum_k E[Z_i^k|D_G]}
\]

where we omit the subscripts from the above expectation operations for the sake of simplicity.

D. E-Step Formulas

Define the following events:

\[
X_k = \{N(s_k^-) - N(s_{k-1}^+) = x_k\}
\]
\[
A_k = \{N(s_k^+) - N(s_k^-) = a_k\}
\]

where \(s_k^- (s_k^+)\) represents the left (right) limit, i.e.,

\[
\{N(s_k^-) = x, N(s_k^+) = y\} = \lim_{\Delta \to 0} \{N(s_k - \Delta t) = x, N(s_k + \Delta t) = y\}.
\]

(15)

Forward segment, backward segment, and overall sequence-of-arrival events can be represented by \(F_k = X_1 A_1 \cdots X_k A_k\), \(B_k = \lambda_k A_k \cdots \lambda_k A_k\), and \(O = \lambda_1 A_1 \cdots \lambda_k A_k\), respectively. Moreover, we use a notation \(P(A)\) as an appropriate probability mass or density function for the event \(A\). Then, \(P(A_k)\) and \(P(A_k)\) correspond to probability mass and density functions, respectively.

Let \(f_{k}(n, u)\) and \(b_{k}(n, u)\) be row and column vectors representing probabilities (likelihoods) for the forward and backward events in the \(k\)th time period \((s_{k-1}, s_k)\). Specifically, the \(i\)th elements of both vectors are defined by

\[
[f_{k}(n, u)]_i = P(F_{k-1}, N((s_{k-1} + u^-) - N((s_{k-1}) = n, J((s_{k-1} + u^-) = i)
\]

\[
b_{k}(n, u)]_i = P(N(s_k^-) - N((s_k - u^+) = n, A_k, E_{k+1}, J((s_k - u^+) = i).
\]

(16)

(17)

Here, we derive expressions of the expected values in (10)–(12) involving \(f_{k}(n, u)\) and \(b_{k}(n, u)\). By using the indicator random variable, \(O\), (10) is rewritten as

\[
\pi_i := \frac{E[B_i|D_G]}{P(O)} = \frac{\pi_i [b_{i}(x_1, t_1)]}{\pi_i [b_{i}(x_1, t_1)]}.
\]

(18)

Since \(E\left[M_{i,j}^k|D_G\right] = E\left[M_{i,j}^k|\mathcal{O}\right]/P(O)\), we treat only \(E\left[M_{i,j}^k|\mathcal{O}\right]\) in the subsequent analysis. According to the conditional stationary independent increments of \(N(t)\), provided that the Markov chain \(J(t)\) is known, we get

\[
E\left[M_{i,j}^k|\mathcal{O}\right] = \int_{s_{k-1}}^{s_k} P(J(\tau^-) = i, J(\tau^+) = j, N(\tau^+) - N(\tau^-) = 0, O) d\tau
\]

\[
= \int_{s_{k-1}}^{s_k} \sum_{l=0}^{x_k} P(F_{k-1}, N(\tau^-) - N(s_{k-1}^+) = l, J(\tau^-) = i) \times P(J(\tau^+) = j, N(\tau^+) - N(\tau^-) = 0|J(\tau^-) = i) \times P(N(s_k^-) - N(\tau^+) = x_k - l, A_k, E_{k+1}, J(\tau^+) = j) d\tau.
\]

(19)

Using \(f_{k}(n, u)\) and \(b_{k}(n, u)\), (19) can be reduced to

\[
E\left[M_{i,j}^k|\mathcal{O}\right] = \sum_{l=0}^{x_k} \int_{0}^{t_k} f_{k}(l, \tau) [b_{k}(x_k - l, t_k - \tau)] d\tau.
\]

(20)

The expected value of \(Z_{i}^k\) is obtained from (20). That is, substituting \(i\) into \(j\) in (19) and (20) yields

\[
E\left[Z_{i}^k|\mathcal{O}\right] = \sum_{l=0}^{x_k} \int_{0}^{t_k} f_{k}(l, \tau) [b_{k}(x_k - l, t_k - \tau)] d\tau.
\]

(21)

The expected value of \(Y_{i,j}^k\) is similarly derived. In this case, since at least one arrival occurs during the time interval \((s_{k-1}, s_k)\), we have

\[
E\left[Y_{i,j}^k|\mathcal{O}\right] = \int_{s_{k-1}}^{s_k} \sum_{l=0}^{x_{k-1}} P(F_{k-1}, N(\tau^-) - N(s_{k-1}^+) = l, J(\tau^-) = i) \times P(J(\tau^+) = j, N(\tau^+) - N(\tau^-) = 1|J(\tau^-) = i) \times P(N(s_k^-) - N(\tau^+) = x_k - l - 1, A_k, E_{k+1}, J(\tau^+) = j) d\tau.
\]

(22)

The main difference between (19) and (22) is whether an arrival occurs at time \(\sigma\) or not. Thus, the upper limit of the summation has to be changed from \(x_k\) to \(x_k - 1\). Using the indicator function, we represent the expected value as follows:

\[
E\left[Y_{i,j}^k|\mathcal{O}\right] = \left\{ x_k > 0 \right\} \sum_{l=0}^{x_{k-1}} \int_{0}^{t_k} f_{k}(l, \tau) [b_{k}(x_k - l - 1, t_k - \tau)] d\tau
\]

(23)
where \( I(\cdot) \) denotes an indicator function. Finally, the expected value of \( U_{ik}^{[k]} \) is computed only when \( a_k = 1 \):

\[
E\left[U_{ik}^{[k]} \right] = P(\mathcal{F}_{k-1}, x_k, J(s_k^-) = i) \times P(J(s_k^+) = j, N(s_k^+) = n(s_k^+) = 1) \times P(\mathcal{B}_{k+1}, s_{k+1}^+) = j) = I(a_k = 1) [f_k(x_k, t_k), \lambda_{ij}[b_{k+1}(x_{k+1}, t_{k+1})]]_j.
\]

(24)

E. Computation of E-Step Formulas

The E-step of MAP estimation with group data requires computation of the vectors \( f_k(n, u) \) and \( b_k(n, u) \) and their convolutions. This paper gives concrete computation algorithms for the expected values.

Define the following block vectors:

\[
\tilde{f}_k(u) = (f_k(0, u), \cdots, f_k(x_k, u))
\]

\[
\tilde{b}_k(u) = (b_k(x_k, u), \cdots, b_k(0, u)) \tag{25}
\]

From the definitions of \( f_k(n, u) \) and \( b_k(n, u) \), these block vectors can be expressed as

\[
\tilde{f}_k(u) = \tilde{x} \cdot D_0^{[k]} \cdot \tilde{D}_1^{[k]} \cdot \cdots \cdot D_0^{[k+1]} \cdot \tilde{D}_1^{[k+1]} \cdot \cdots \cdot D_0^{[k+m]} \cdot \tilde{u}
\]

(26)

and

\[
\tilde{b}_k(u) = \tilde{e} \cdot D_0^{[k]} \cdot \tilde{D}_1^{[k]} \cdot \cdots \cdot D_0^{[k+1]} \cdot \tilde{D}_1^{[k+1]} \cdot \cdots \cdot D_0^{[k+m]} \cdot \tilde{D}_1^{[k+m]} \cdot \tilde{e}
\]

(27)

where \( D_0^{[k]} \) is an \((mx_k + m)\)-(\(mx_k + m\)) block upper triangular matrix

\[
D_0^{[k]} = \begin{pmatrix} D_0 & D_1 & \cdots & D_1 & \cdots & D_1 & \cdots & D_1 \\ D_0 & D_0 & \cdots & D_0 & \cdots & D_0 & \cdots & D_0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ D_0 & D_0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \end{pmatrix}
\]

(28)

and \( D_1^{[k]} \) is an \((mx_k + m)\)-(\(mx_k + m\)) block matrix

\[
D_1^{[k]} = \begin{pmatrix} O & O & \cdots & O \\ \end{pmatrix} \begin{pmatrix} I & O & \cdots & O \\ D_0^{[k]} \end{pmatrix}
\]

(29)

The matrices \( I \) and \( O \) represent the identity and zero matrices, respectively. In (29), when \( a_k = 0 \), \( D_1^{[k]} \) becomes the identity matrix. The row and column vectors \( \tilde{x} \) and \( \tilde{e} \) are defined by

\[
\tilde{x} = (x_0, 0, \cdots, 0), \hspace{1cm} \tilde{e} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ e \end{pmatrix}.
\]

(30)

Equations (26) and (27) imply that the block vectors \( \tilde{f}_k(u) \) and \( \tilde{b}_k(u) \) can be computed in a forward–backward manner as is common in almost all the EM algorithms for the family of MAPs. Also, an \( m \)-by-\( m \) matrix \( H_k(n, u) \) representing the convolution of \( f_k(n, u) \) and \( b_k(n, u) \) is needed to derive computation methods for the expected values. Define

\[
H_k(n, u) = \sum_{l=0}^{n} \int_{0}^{u} b_k(n-l, u-\tau)f_k(l, \tau) \, d\tau.
\]

(31)

In this paper, we propose two algorithms for the above computation: one based on differential equations and the other based on uniformization. We first describe the differential equation approach. The differential equations for \( f_k(n, u) \) and \( b_k(n, u) \) are simply obtained from the block matrices of \( D_0^{[k]} \) and \( D_1^{[k]} \) and are analogous to the MAP formulation of (4) and (5). For instance, the differential equations for \( f_k(n, u) \) are given by

\[
\frac{d}{du} f_k(0, u) = f_k(0, u)D_0
\]

(32)

\[
\frac{d}{du} f_k(n, u) = f_k(n, u)D_0 + f_k(n-1, u)D_1, \hspace{1cm} n = 1, \ldots, x_k
\]

(33)

where the initial conditions are \( f_k(0, 0) = \pi, f_k(0, 0) = f_{k-1}(x_k, t_k)D_1^{[k-1]} \) and \( f_k(n, 0) = 0, n = 1, \ldots, x_k \). The differential equations of \( b_k(n, u) \) are analogous to those of \( f_k(n, u) \). The differential equations for \( H_k(n, u) \) are given by

\[
\frac{d}{du} H_k(0, u) = D_0 H_k(0, u) + b_k(0, 0)f_k(0, u)
\]

(34)

\[
\frac{d}{du} H_k(n, u) = D_0 H_k(n, u) + D_1 H_k(n-1, u)
\]

(35)

where the initial conditions are \( H_k(n, 0) = 0, n = 0, \ldots, x_k \). These differential equations can be dealt with even for the case where the underlying CTMC is stiff [30]. In the stiff case, these differential equations are solved by the implicit Runge–Kutta method.

Next, we present the formulas using uniformization for \( f_k(n, u) \), \( b_k(n, u) \), and \( H_k(n, u) \). From (26), the following relationship between \( f_k(n, u) \) and \( f_k(0, 0) \) holds:

\[
f_k(n, u) = f_k(0, 0) \sum_{l=0}^{n} e^{-ru_l} \left[ \frac{ru_l}{l!} \right]^l \prod_{i=0}^{l-1} P_{0}^{z_i} P_{1}^{z_i} \cdots \prod_{j=0}^{n-1} P_{0}^{z_j}
\]

(36)

where \( r > \max_{i} \left[ |D_0^{[k]}|, \hspace{1cm} P_{0} = I + D_0/r, \hspace{1cm} P_{1} = D_1/r \right. \) and \( z_l = l - n - \sum_{j=0}^{l} z_j \). Intuitively, the convolution term in (36) indicates a phase transition probability matrix at the time instants when \( l - n \) phase transitions and \( n \) arrivals occur. Let \( \tilde{f}(n, y) \) be the probability vector in (36)

\[
\tilde{f}(n, y) = f_k(0, 0) \sum_{l=0}^{n} e^{-ru_l} \left[ \frac{ru_l}{l!} \right]^l \tilde{f}(n, l - n),
\]

(37)

Then, (36) is reduced to

\[
f_k(n, u) = \sum_{l=0}^{\infty} e^{-ru_l} \left[ \frac{ru_l}{l!} \right]^l \tilde{f}(n, l - n),
\]

(38)
Moreover, we utilize the following recurrent formula to compute $\tilde{f}(n,y)$:

$$
\tilde{f}(n,y) = \tilde{f}(n,y-1)P_0 + \tilde{f}(n-1,y)P_1.
$$

(39)

Similarly, we have the following uniformization formulas for $b_k(n,u)$:

$$
b_k(n,u) = \sum_{l=0}^{\infty} e^{-ru} \frac{(ru)^l}{l!} b(n,l-n)
$$

(40)

$$
\tilde{b}(n,y) = P_0 \tilde{b}(n,y-1) + P_1 \tilde{b}(n-1,y)
$$

(41)

$$
\tilde{b}(0,0) = b_k(0,0).
$$

(42)

Note that the probability vectors $\tilde{f}(n,u)$ and $\tilde{b}(n,u)$ are defined to be zero vectors when either $n$ or $y$ is negative. Now, the matrix $H_k(n,u)$ is given by

$$
H_k(n,u) = \sum_{l=0}^{\infty} \frac{1}{l!} \left( \sum_{z_0=0}^{\min(n-1,\bar{z}_0)} \sum_{z_1=0}^{\min(z_0-1,\bar{z}_1)} \cdots \sum_{z_{n-1}=0}^{\min(z_{n-2},\bar{z}_{n-1})} \sum_{z_n=0}^{\bar{z}_n} \frac{(ru)^l}{l!} \tilde{f}(n,l-n) \prod_{i=0}^{n} P_i \right)
$$

(43)

where $\Lambda = b_k(0,0)f_k(0,0)$. Similar to $f_k(n,u)$ and $b_k(n,u)$, we obtain the following uniformization formula for $H_k(n,u)$:

$$
H_k(n,u) = \sum_{l=0}^{\infty} \frac{1}{l!} \left( \sum_{z_0=0}^{\min(n-1,\bar{z}_0)} \sum_{z_1=0}^{\min(z_0-1,\bar{z}_1)} \cdots \sum_{z_{n-1}=0}^{\min(z_{n-2},\bar{z}_{n-1})} \sum_{z_n=0}^{\bar{z}_n} \frac{(ru)^l}{l!} \tilde{f}(n,l-n) \right)
$$

(44)

$$
\tilde{H}(n,y) = P_0 \tilde{H}(n,y-1) + P_1 \tilde{H}(n-1,y) + b_k(0,0) \tilde{f}(n,y)
$$

(45)

where $\tilde{H}(n,y) = O$ when either $n$ or $y$ is negative. Specifically, letting $n = 0$, the above equation reduces to the uniformization formula for arrival time data discussed in [13]. In (43), the convolution terms correspond to respective summations in (31).

Now, we summarize the EM algorithm to estimate the parameters of MAP with (generalized) group data.

Algorithm 1

- **Step 0:** Determine a set of initial guesses $\theta := \{\pi_{i\ell}, \mu_{i\ell}, \lambda_{i\ell}\}$.
- **Step 1:** Compute $b_k(n,u)$ by either the differential equation or the uniformization method.
- **Step 2:** Compute $f_k(n,u)$ and $H_k(n,u)$ simultaneously by either the differential equation or the uniformization method.
- **Step 3:** Compute the expected values in (18), (20), (21), (23), and (24) by using $f_k(n,u)$, $b_k(n,u)$, and $H_k(n,u)$.
- **Step 4:** Update the parameters according to (10)–(12).
- **Step 5:** If a termination condition is satisfied (see Remark 3.4), the algorithm stops. Otherwise, go to Step 1.

**Remark 3.1 (Time Complexity):** The time complexity of one EM-step with group data is given by $O(RKWm^3)$ for the differential equation method and is $O(UKWm^3)$ for the uniformization method, where $W$ is the maximum number of arrivals for one time interval, $R$ is the number of computational steps for solving differential equations, and $U$ is the number of steps before truncation of the infinite sum in the uniformization. Note that we treat the case where $D_1$ and $D_2$ are full matrices. For instance, if the differential equation is solved by Runge–Kutta method, $R$ is given by $r_i$, where $r_i$ is the number of steps required per unit time and $t$ is length of the time interval. On the other hand, $U$ is given by $\sqrt{rt}$ (see, e.g., [31]). When the underlying CTMC is nonstiff, the number of steps in the uniformization method is less than that in the differential equation method. The storage requirement of the proposed algorithm is proportional to $Wm^2$. Therefore, the EM algorithm with group data may not function well when $W$ is extremely large.

**Remark 3.2 (Estimation for MMPP):** Using the proposed EM algorithm with group data, we can also perform parameter estimation for MMPP. Modification to realize the estimation for MMPP is that an initial guess of the matrix $D_k$ is set as a diagonal matrix. As seen in (12), (23), and (24), the updated estimates of off-diagonals of $D_k$ are given by $0$ at every EM-step if the initial guess is a diagonal matrix.

**Remark 3.3 (Convergence Property and Initial Guesess):** In general, EM algorithm has a global convergence property. The global convergence guarantees that the algorithm converges to the global maximum, to a local maximum, or to a saddle point for any initial guess. Therefore, compared to other numerical methods having local convergence like Newton’s method, it is not hard to choose an initial guess. However, the global convergence property does not always guarantee the global optimization of LLF, and thus estimation results do depend on initial guesses. In the case of MAP parameter estimation, random variables are assigned to initial guesses, or the $k$-means algorithm is used to determine initial guesses. For example, using $k$-means, we divide all the arrivals into $k$ different classes based on their interarrival times where we assume that the classified arrivals occur in the same phase. That is, the classification gives the state of MAP at arrival instants. Moreover, making the assumption that there is at most one phase transition between arrivals, we can count the number of phase transitions and compute initial guesses from the classified arrivals.

**Remark 3.4 (Termination Condition):** The convergence speed of EM algorithm is generally lower than other numerical methods. The termination condition in Algorithm 1 should carefully be determined since it would affect the accuracy of the resulting estimates.

Intuitively a reasonable condition to stop the algorithm is based on the difference in successive likelihood values or on the relative difference in estimated parameter values. Let $\theta$ and $\theta'$ respectively denote provisional parameters and their updated values after an EM-step. Then, typical termination conditions are given by

$$
||\text{LLF}(\theta') - \text{LLF}(\theta)|| < \epsilon_1 \quad \text{and} \quad \frac{||\theta' - \theta||}{||\theta||} < \epsilon_p.
$$

(46)

**Remark 3.5 (Finding the Optimal Number of Phases):** One of the most important issues in MAP parameter estimation is determining the optimal number of phases. Although accuracy of fitting gets better as the number of phases increases, it often
leads to an overfitting problem. The simplest yet valid method to avoid the overfitting problem is to use information criteria such as Akaike’s information criterion (AIC) [32]. The AIC is defined by

$$
\text{AIC} = -2(\text{maximum log-likelihood}) + 2p
$$

(47)

where $p$ is the degrees of free parameters. In the case of a general MAP with $m$ phases, the number of free parameters is given by $2m^2 - 1$ or $m^2$. Using the above equation, we can determine the number of phases with the smallest AIC.

IV. APPROXIMATE EM ALGORITHM FOR MMPP WITH GROUP DATA

The proposed EM algorithm for MAP with group data needs a large amount of memory and computation time as the maximum number of arrivals $W$ per bin increases. This section presents an approximate method to reduce the computation cost even in the case where $W$ is large. In order to achieve efficiency, we sacrifice generality and only consider estimation of MMPP with group data. The idea behind our approximation is to use an MMPP-like, but somehow restrictive, stochastic process rather than the exact MMPP. Similar approaches have been applied by others; batch MMPP (BMMPP) has been used to fit to group data. This implicitly makes the assumption that arrivals occur in a batch and do not spread over the observed time intervals.

This paper considers an MMPP-like process based on the assumption that there is at most one phase transition in each time interval. This assumption might be acceptable if the length of the observed time period is sufficiently small compared to the average time interval between phase transitions. Therefore, the approximate estimates are expected to get closer to those produced by the exact EM algorithm proposed here as the length of observed time interval becomes smaller.

A. EM-Step Formulas

Consider the same unobserved variables as in Section III. Then, M-step formulas in the approximation are the same as in (10)–(12). The essential difference between the approximate EM algorithm and the exact one is in E-step formulas.

By the assumption of at most one phase transition for each observed time interval, there are at most two different phases during each time interval. An important step in the approximation is to separate each of the observed time intervals into two subintervals: the one before and the other after the time instant of the phase transition. Hence, the continuous time processes $N(t)$ and $J(t)$ of MMPP are reduced to discrete time processes on time points of the phase transition instants in our approximation. The following random variables are defined:

- $J_{1[k]}^k, J_{2[k]}^k$: phase indices before and after the time instant of a phase transition in the time interval $(s_{k-1}, s_k)$;
- $T_{1[k]}^k, T_{2[k]}^k$: time durations before and after the time instant of a phase transition in the time interval $(s_{k-1}, s_k)$.

Here, $2m^2 - 1$ indicates the number of free parameters in $\mathbf{r}$, $\mathbf{D}_{0}$, and $\mathbf{D}_{1}$. On the other hand, $m^2$ is the number of free parameters in minimal representation for MAP [33]. The definition of AIC for MAP has not been discussed yet in the literature. This paper uses $m^2$ free parameters for MMPP in Section V.

If a phase transition does not occur during the time interval $(s_{k-1}, s_k)$, we set $J_{1[k]}^k = J_{2[k]}^k$ and $T_{1[k]}^k = 0$. The unobserved variables of the MMPP are formulated as follows:

$$
B_k = J_{1[k]}^k, \quad U_{i,k}^k = I(q_k = 1)I(J_{2[k]}^k = i),
\quad Z_{i,k}^k = I(J_{1[k]}^k = j)I(J_{2[k]}^k = i)T_{2[k]}^k
$$

(48)

$$
M_{i,k}^k = I(J_{1[k]}^k = i, J_{2[k]}^k = j), \quad i \neq j
$$

(49)

$$
\gamma_{i,k}^k = I(J_{1[k]}^k = i)X_{1[k]}^k + I(J_{2[k]}^k = i)X_{2[k]}^k
$$

(50)

Define the following probability:

$$
\gamma_{i,k}^k(n, u) = \sum_{l=0}^{n} P(X_{i}^k = l, X_{2}^k = n-l)
$$

$$
= \sum_{l=0}^{n} \int_{u}^{\infty} \binom{n}{l} \lambda_{i}^{l}(\tau)^{l} e^{-\lambda_{i} \tau} e^{-\mu_{i} \tau} \mu_{i,j}^{l} \times \frac{\lambda_{j}(u-\tau)^{(n-l)}}{(n-l)!} e^{-\lambda_{j} (u-\tau)} e^{-\mu_{j} (u-\tau)} dt
$$

(51)

Let $G_{k}$ be a matrix whose $(i,j)$-element is $\gamma_{i,j}^k(x_{k}, t_{k})$. Then, the vectors $f_{k}(x_{k}, t_{k})$ and $b_{k}(x_{k}, t_{k})$ can be rewritten as

$$
f_{k}(x_{k}, t_{k}) = \pi G_{k} D_{0}^{1} G_{2} \cdots D_{k}^{k-1} G_{k}
$$

$$
b_{k}(x_{k}, t_{k}) = G_{k} D_{0}^{k} \cdots G_{k} D_{1}^{1} e.
$$

(52)

Also, we define two expected values representing time lengths before and after the time instant of the phase transition

$$
\psi_{i,j}^k(n, u, z) = \sum_{l=0}^{n} E[T_{1}[k]I(X_{i}^k = l, X_{2}^k = n-l)]
$$

$$
= \sum_{l=0}^{n} \int_{0}^{u} \frac{\tau}{l!} \lambda_{i}^{l}(\tau)^{l} e^{-\lambda_{i} \tau} e^{-\mu_{i} \tau} \mu_{i,j}^{l} \times \frac{\lambda_{j}(u-\tau)^{(n-l)}}{(n-l)!} e^{-\lambda_{j} (u-\tau)} e^{-\mu_{j} (u-\tau)} d\tau
$$

(54)

In the case that $\lambda_{i} \neq \lambda_{j}$, we have

$$
\psi_{i,j}^k(n, u, 1) = \sum_{l=0}^{n} \int_{0}^{u} \frac{\tau}{l!} \lambda_{i}^{l}(\tau)^{l} e^{-\lambda_{i} \tau} e^{-\mu_{i} \tau} \mu_{i,j}^{l} \times \frac{\lambda_{j}(u-\tau)^{(n-l)}}{(n-l)!} e^{-\lambda_{j} (u-\tau)} e^{-\mu_{j} (u-\tau)} d\tau
$$

$$
= \frac{n+1}{\lambda_{i}} - \frac{\lambda_{j}}{\lambda_{i}} \gamma_{i,j}^k(n+1, u)
$$

(55)

Equation (55) is derived using integration by parts. An explicit form of $\psi_{i,j}^k(n, u, 2)$ is obtained by interchanging $\lambda_{i}$ and $\lambda_{j}$ in (55). This can be verified using the relation

$$
\sum_{z=1}^{\infty} \psi_{i,j}^k(n, u, z) = \psi_{j,i}^k(n, u),
$$

which is easily obtained from the definition of $\psi_{i,j}^k(n, u, z)$.

In the case that $\lambda_{i} = \lambda_{j}$, $\mu_{i} = \mu_{j}$, and $\mu_{i} = \mu_{j}$, the expressions of $\psi_{i,j}^k(n, u, z)$ are simple. For instance, $\psi_{i,j}^k(n, u, 1)$ is given by the probability that $n$ arrivals
occur at phase $i$ during the time interval of duration $u$, i.e., $u(\lambda_{i,ut})^u/\mu_{i,ut}$.

Similar to the analysis in Section III, the expected value $E \left[ M_{ik}^{[k]}(n) \right]$ in the approximation is given by

$$E \left[ M_{ik}^{[k]}(n) \right] = \sum_{t=0}^{n} P(F_{k-1}, X_{k}^{[k]} = i, J_{k}^{[k]} = i, \lambda_{kt}^{[k]} = x_k - l, T_{k}^{[k]} + T_{k+1}^{[k]} = t, k_j^{[k]} = j, A_k, B_{k+1})$$

$$= f_{k-1}(x_k - t, k-1)D_{k}^{[k-1]} \times \gamma_{ij}^{[k]}(x_k, t_k)D_{k+1}^{[k+1]}b_{k+1}(x_{k+1}, t_{k+1})]_j,$$  

(56)

The expected value $E \left[ Z_{ik}^{[k]}(n) \right]$ can be derived as follows:

$$E \left[ Z_{ik}^{[k]}(n) \right] = \sum_{j=1}^{m} \left[ f_{k-1}(x_k - t, k-1)D_{k}^{[k-1]} \times \psi_{[k]}(x_k, t_k; 1)D_{k+1}^{[k+1]}b_{k+1}(x_{k+1}, t_{k+1})]_jight]$$

$$+ \sum_{j=i}^{m} \left[ f_{k-1}(x_k - t, k-1)D_{k}^{[k-1]} \times \psi_{[k]}(x_k, t_k; 2)D_{k+1}^{[k+1]}b_{k+1}(x_{k+1}, t_{k+1})]_jight]$$

$$= f_{k-1}(x_k - t, k-1)D_{k}^{[k-1]} - \Psi_k(x_k, t_k; i) \times D_{k+1}^{[k+1]}b_{k+1}(x_{k+1}, t_{k+1}),$$  

(57)

where $\Psi_k(n, u, t)$ is the matrix whose $i$th row and $j$th column are nonzero and the other elements are zero. The $i$th row and the $j$th column values are $\psi_{[k]}(n, u, 1)$, $j = 1, \ldots, m$, and $\psi_{[k]}(n, u, 2)$, $j = 1, \ldots, m$, respectively.

To obtain an expression for the expected value $E \left[ Y_{ik}^{[k]}(n) \right]$ in the approximation, we utilize the following relationship:

$$E\left[ X_{ik}^{[k]} \right] = n, T_{ik}^{[k]} + \gamma_{ik}^{[k]} = u, J_{ik}^{[k]} = j \right] f_{ik}^{[k]} = \delta$$

$$= \sum_{t=0}^{n} \left[ (\lambda_{i,ut})^u/\mu_{i,ut} \right] e^{-\lambda_{i,t}} e^{-\mu_{i,t} t} \frac{n!}{(n-t)!}$$

$$\times \lambda_{i,t} \psi_{[k]}(n - 1, u, 1),$$  

(58)

Thus, the expected value is derived as

$$E \left[ Y_{ik}^{[k]}(n) \right] = I(x_k > 0)f_{k-1}(x_k - t, k-1)$$

$$\times D_{k}^{[k-1]} \lambda_{i,t} \psi_k(x_k - t, k-1),$$  

(59)

Similarly, the expected value $E \left[ T_{ik}^{[k]}(n) \right]$ can be obtained starting with (24)

$$E \left[ T_{ik}^{[k]}(n) \right] = I(a_k = 1)[f_{k}(x_k, t_k), \lambda_{i,t} \psi_{k}(x_k - 1, t_k, 1)D_{k+1}^{[k+1]}b_{k+1}(x_{k+1}, t_{k+1})]_j.$$  

(60)

### B. Computation of $\gamma_{ik}^{[k]}(n, U)$

The computation cost of the E-step strongly depends on the computation cost of $\gamma_{ik}^{[k]}(n, U)$. We choose one of the following two methods based on the MMPP approach as suggested in Remark 4.2. After applying some algebraic manipulations, we have an explicit form of the function $\gamma_{ik}^{[k]}(n, U)$

$$\gamma_{ik}^{[k]}(n, U) = \frac{1}{\xi_{n+1}(\lambda_{i,d} - \lambda_{j,d})} \times \{F_G(\lambda_{i,d}u; n+1) - F_G(\lambda_{j,d}u; n+1)\}$$  

(61)

where $\xi = -\left(\sigma_{i,d} - \sigma_{j,d}\right)/(\lambda_{i,d} - \lambda_{j,d})$, $\sigma_{i,d} = \mu_{i,d} + \lambda_{i,d}$, and $F_G(x; \alpha)$ is the cumulative distribution function (cdf) of standard gamma distribution with shape parameter $\alpha$. Since this equation includes subtraction as well as addition, it does not work when $\lambda_{i,d}$ is close to $\lambda_{j,d}$ (see Remark 4.2).

The second method is based on uniformization that can be used even in the case that $\lambda_{i,d} = \lambda_{j,d}$. The probability $\gamma_{ik}^{[k]}(n, U)$ equals a state probability of the CTMC with the following block infinitesimal generator $Q$:

$$Q = \begin{pmatrix} -\sigma_{i,d} & \lambda_{i,d} & \mu_{i,j} \\ \vdots & \ddots & \vdots \\ -\sigma_{j,d} & \lambda_{j,d} & -\sigma_{j,j} \end{pmatrix}$$  

(62)

where each block is an $(n + 1)$-by-$(n + 1)$ square matrix. We apply the uniformization method using the above infinitesimal generator in order to compute $\gamma_{ik}^{[k]}(n, U)$. Specifically, $\gamma_{ik}^{[k]}(n, U)$ is the state probability of the last state in the above CTMC at time $u$, provided that the initial state of the CTMC is the first state.

We now summarize the approximate EM algorithm for MMPP with group data.

#### Algorithm 2

Step 0: Determine a set of initial guesses $\theta := \{\pi_i, \mu_{i,j}, \lambda_{i,}\}$. Via $\gamma_{ik}^{[k]}(n, U)$.

Step 1: Compute the matrices $G_k$, $k = 1, \ldots, K$, via $\gamma_{ik}^{[k]}(n, U)$.

Step 2: Compute $b_0(x_k, t_k)$ using (53).

Step 3: Compute $f_k(x_k, t_k)$ using (52).

Step 4: Compute the expected values in (56), (57), (59), and (60) using $G_k$ and $\psi_{[k]}(n, u, t)$.

Step 5: Update the parameters according to the M-step formulas (10)–(12).

Step 6: If a termination condition is satisfied (see Remark 3.4), the algorithm stops. Otherwise, go to Step 1.

#### Remark 4.1 (Time Complexity): Time complexity of the approximate EM algorithm is $O(KGm^2)$, where $G$ is the computation cost of $\gamma_{ik}^{[k]}(n, U)$, and $m$ is the number of the algorithmic steps in the parameter estimation. Equation (61) is selected in most of the cases in practice (see Remark 4.2). Moreover, time complexity of the computation of (61) does not depend on the maximum group size $W$ because well-known approximations of the standard gamma distribution can be applied to the computation.
Thus, actual computation time of the approximate method is less than that of the exact EM algorithm described in Section III.

Remark 4.2 (Selecting the Computation Method): Since (61) has less computation time than the uniformization, determining the applicable condition for (61) affects the total computation cost of the approximation. The simplest way is to use absolute distance between $\lambda_{ij}$ and $\lambda_{ij}^a$. A better strategy is to choose between the two methods based on the difference between two standard gamma distributions. Instead of the absolute difference, we measure relative difference between $\xi_{ij}$ and $\xi_{ij}^a$ over a quantile-based range $[q_{0.01}, q_{0.99}]$, where $q_\alpha$ is an $\alpha$-quantile point of the standard gamma distribution. If the relative difference is large, (61) can be used for the computation without numerical exception such as an underflow or an overflow. Otherwise, uniformization is used.

V. NUMERICAL EXPERIMENTS

A. Performance Test of the Exact EM Algorithm With Group Data

This section examines the scalability and the accuracy of the proposed EM algorithms.

Consider a 2-state MMPP with the following parameters:

$$D_0 = \begin{pmatrix} -11.0 & 1.0 \\ 0.1 & -1.1 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 10.0 & 0 \\ 0 & 1.0 \end{pmatrix}. \quad (63)$$

Noting that the choice of the initial probability vector does not affect estimation results, we set $\vec{\pi} = (1, \vec{0})$.

Three sets of arrival time data (AD) with 100, 1000, and 10 000 arrivals are generated from the 2-state MMPP. Group data (GD) used in the experiments are obtained by aggregation from the ADs with six different observation time intervals; 10.0, 5.0, 1.0, 0.5, 0.1, and 0.05. Table I summarizes the data. In this table, $K$ and $W$ denote, respectively, the number of data records and the maximum number of arrivals for one record, where the number of data records corresponds to the number of arrivals for AD and the number of observed time intervals for GD.

The proposed EM algorithms are programmed in C++ language with Basic Linear Algebra Subprograms (BLAS) and Linear Algebra Package (LAPACK). The termination condition for the EM algorithm is the LLF rule (see Remark 3.4), with $\epsilon_l = 1.0e - 3$. The initial guesses are uniform random variates with a range $(0, 1)$ for all the parameters. The same initial guesses are also used in all the later performance tests.

Tables II–IV present estimation results for the cases of 100, 1000, and 10 000 arrivals, respectively. In these tables, $N^*$ indicates the number of iterations (EM-steps) until the termination condition is satisfied. Also, the column ‘Time’ presents execution time in seconds under 2-GHz Intel Core Duo and 1-GB memory. The estimates in Table IV are closer to the actual parameters of (63) than those in Tables II and III. That is, the exact MLEs of the MMPP become more accurate as the number of arrivals increases. On the other hand, as observed time intervals in GD are longer, the estimates move away from the actual parameters. This is the evidence that the amount of information in GD is less than that in AD. In other words, GD are always inferior to AD in terms of estimation accuracy. Therefore, the step size of observed intervals strongly affects the accuracy of the estimates with GD. In this numerical experiment, we observe that the estimates with GD become more accurate when the step size is less than 1.0. Note that the lowest arrival rate of the MMPP is 1.0 in our experiments. Based on these results, we recommend that the step size of observed intervals in GD should be less than the longest average of interarrival time of the MMPP. If the step size is larger than the average time interval, the information about low arrival rates would be lost due to the arrivals with high rates.

Next, we investigate the computation time and the scalability of the exact EM algorithm. Fig. 1 shows the dependence of computation time per record on the maximum number of arrivals $W$ (left) and the number of phases (right). Although the theoretical time complexity linearly increases with $W$, the actual computation time increases polynomially with $W$ in Fig. 1. Similarly,
Fig. 1. Computation time of the exact EM algorithm.

Fig. 2. The lag $k$ correlations (Example 1: 3-state MAP).

Fig. 3. The lag $k$ correlations (Example 2: 6-state MMPP).

the computation time increases much faster with the number of phases, $m$, than with the theoretical time complexity that is proportional to $p^3$. The two results on computation time tell us that the exact EM algorithm is expensive in terms of computation cost in the case of large $W$ and $m$, so approximation methods are needed in some applications.

Next, we investigate the fitting ability of the MAP estimation algorithm for group data to lag $k$ correlation. For this purpose, we consider the 3-state MAP and the 6-state MMPP presented in [29, Examples 1 and 2]. As in previous experiments with MMPP, arrival time data are randomly generated until 10 000 arrivals are obtained, and group data are derived by aggregation from the arrival time data. Figs. 2 and 3 present the lag $k$ correlations of the original 3-state MAP and 6-state MMPP (actual), and the estimated ones by the EM algorithm with group data. The proposed EM algorithm with group data gives both MAP and MMPP estimates such that the estimated lag $k$ correlations are close to the original ones. In addition, the parameter estimation with group data becomes more accurate, as the time range of bins for group data gets smaller. Compared to the results in [29], Figs. 2 and 3 clarify that likelihood-based fitting methods discussed here can provide more accurate estimates than those by moment-based methods in terms of lag $k$ correlation as well as likelihoods.

B. Performance of the Approximate EM Algorithm With Group Data

This section is devoted to studying the computation speed and the accuracy of the approximate EM algorithm. Tables V–VII show estimated results with the 2-state MMPP in (63) by the approximate EM algorithm for the same data in Tables II–IV. In these tables, it can be seen that computation times of the approximate method are much smaller than those of the exact EM algorithm. Also, when the step size is less than 0.1, the approximation seems to work better; namely, the approximate estimates are close to the original ones. In addition, the step size is expected to be less than the shortest average interarrival time to ensure that the approximation is accurate.

<table>
<thead>
<tr>
<th>Data</th>
<th>N*</th>
<th>Time</th>
<th>$\lambda_{1,2}$</th>
<th>$\lambda_{2,1}$</th>
<th>$\lambda_{1,1}$</th>
<th>$\lambda_{2,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>8</td>
<td>0.03</td>
<td>0.670</td>
<td>0.091</td>
<td>10.315</td>
<td>0.873</td>
</tr>
<tr>
<td>GD10</td>
<td>16</td>
<td>0.00</td>
<td>0.156</td>
<td>0.065</td>
<td>4.444</td>
<td>0.601</td>
</tr>
<tr>
<td>GD5</td>
<td>18</td>
<td>0.00</td>
<td>0.333</td>
<td>0.057</td>
<td>7.374</td>
<td>0.950</td>
</tr>
<tr>
<td>GD1</td>
<td>11</td>
<td>0.02</td>
<td>0.693</td>
<td>0.081</td>
<td>10.975</td>
<td>0.918</td>
</tr>
<tr>
<td>GD05</td>
<td>9</td>
<td>0.04</td>
<td>0.692</td>
<td>0.084</td>
<td>10.886</td>
<td>0.884</td>
</tr>
<tr>
<td>GD01</td>
<td>8</td>
<td>0.12</td>
<td>0.733</td>
<td>0.101</td>
<td>10.374</td>
<td>0.871</td>
</tr>
<tr>
<td>GD005</td>
<td>8</td>
<td>0.22</td>
<td>0.726</td>
<td>0.102</td>
<td>10.256</td>
<td>0.874</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>N*</th>
<th>Time</th>
<th>$\lambda_{1,2}$</th>
<th>$\lambda_{2,1}$</th>
<th>$\lambda_{1,1}$</th>
<th>$\lambda_{2,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>14</td>
<td>0.47</td>
<td>0.859</td>
<td>0.061</td>
<td>9.799</td>
<td>1.014</td>
</tr>
<tr>
<td>GD10</td>
<td>14</td>
<td>0.04</td>
<td>0.075</td>
<td>0.033</td>
<td>3.116</td>
<td>0.966</td>
</tr>
<tr>
<td>GD5</td>
<td>23</td>
<td>0.12</td>
<td>0.207</td>
<td>0.033</td>
<td>4.985</td>
<td>1.060</td>
</tr>
<tr>
<td>GD1</td>
<td>16</td>
<td>0.40</td>
<td>0.636</td>
<td>0.048</td>
<td>8.944</td>
<td>1.023</td>
</tr>
<tr>
<td>GD05</td>
<td>14</td>
<td>0.63</td>
<td>0.740</td>
<td>0.056</td>
<td>9.233</td>
<td>1.011</td>
</tr>
<tr>
<td>GD01</td>
<td>15</td>
<td>2.78</td>
<td>0.936</td>
<td>0.072</td>
<td>9.407</td>
<td>0.999</td>
</tr>
<tr>
<td>GD005</td>
<td>15</td>
<td>5.35</td>
<td>0.991</td>
<td>0.075</td>
<td>9.501</td>
<td>0.999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>N*</th>
<th>Time</th>
<th>$\lambda_{1,2}$</th>
<th>$\lambda_{2,1}$</th>
<th>$\lambda_{1,1}$</th>
<th>$\lambda_{2,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>12</td>
<td>3.96</td>
<td>0.785</td>
<td>0.074</td>
<td>10.342</td>
<td>1.029</td>
</tr>
<tr>
<td>GD10</td>
<td>24</td>
<td>0.55</td>
<td>0.109</td>
<td>0.025</td>
<td>4.659</td>
<td>1.184</td>
</tr>
<tr>
<td>GD5</td>
<td>19</td>
<td>0.83</td>
<td>0.204</td>
<td>0.035</td>
<td>5.970</td>
<td>1.129</td>
</tr>
<tr>
<td>GD1</td>
<td>14</td>
<td>2.93</td>
<td>0.583</td>
<td>0.057</td>
<td>9.630</td>
<td>1.069</td>
</tr>
<tr>
<td>GD05</td>
<td>13</td>
<td>4.97</td>
<td>0.699</td>
<td>0.067</td>
<td>10.090</td>
<td>1.041</td>
</tr>
<tr>
<td>GD01</td>
<td>13</td>
<td>21.17</td>
<td>0.886</td>
<td>0.087</td>
<td>10.219</td>
<td>1.010</td>
</tr>
<tr>
<td>GD005</td>
<td>13</td>
<td>41.19</td>
<td>0.928</td>
<td>0.092</td>
<td>10.203</td>
<td>1.005</td>
</tr>
</tbody>
</table>
Fig. 4 presents lag $k$ correlation by the approximate EM algorithm for Example 2. Note that the proposed approximation could be applied to only estimation of MMPP. Although the lag $k$ correlations by the approximation are slightly different from the actual one compared to Fig. 3, the lag $k$ correlation gets close to the actual one as the range of bins gets smaller. This observation is consistent with the theoretical insight of the approximation; the approximation becomes better as observed time intervals get smaller. In addition, numerical results show that the behavior of lag $k$ correlation with large $k$ is well approximated even when the range of bins is large.

The approximation might not do well when off-diagonals of $D$ are relatively larger than arrival rates of $D$ since the assumption that phase transition rates are lower than arrival rates is then violated. Thus, we examine another 2-state MMPP where two off-diagonals of $D$ in (63) are 10.0. Table VIII presents estimates by the exact EM algorithm (upper) and the approximate EM algorithm (lower). Compared to the estimation results in Table VII, the estimates in Table VIII are far from the original MMPP parameters. That is, the approximation tends to fail to estimate the parameters when off-diagonals of an infinitesimal generator are large.

Finally, we discuss the fitting ability of the approximation. Table IX presents relative errors of maximum log-likelihoods (MLLs) between the exact EM algorithm and the approximate one. In the table, the upper (2-MMPP-1) indicates the relative errors of MLLs for the 2-state MMPP in (63) with 10,000 arrivals, and the lower (2-MMPP-2) shows those for the 2-state MMPP whose off-diagonals are 10.0. Since the exact and approximate EM algorithms are performed on the same data, the difference of MLL can measure the fitting accuracy of the approximation. Although the approximate estimates are not close to the original parameters as in Table VIII, the relative errors reduce as the range of bins becomes smaller in both cases. This result implies that the accuracy of the approximation is improved by reducing the step size even in the case of 2-MMPP-2.

C. Traffic Analysis

In this section, the EM algorithm is applied to group data of real traffic measured in Hiroshima University, Japan. The traffic data are aggregated from an IP transaction audit tool Argus. A targeted host is installed in the Department of Information Engineering, Hiroshima University. The audit records are collected for two weeks. Although this audit tool originally reports group data for inbound and outbound packets, we further aggregate the group data at 10.0-s time intervals so as to reduce the number of total records. The number of total records for two weeks is 129,881. The targeted host mainly provides Web service for external domain and file service for internal PC clients. In this paper, we separate HTTP traffic (port 80) and SMB traffic (port 135) to clarify their statistical characteristics. Fig. 5 shows time sequences of the number of packets for two protocols, where the maximum numbers of arrivals in port 80 and port 135 are 1169 packets and 59 packets, respectively.

We apply the proposed EM algorithms for MMPP, exact and approximate methods for these packet data. We first discuss the optimal number of phases based on the AIC (see Remark 3.5). Tables X and XI present the MLL and the corresponding AIC for each number of phases. When the number of phases is 1, the corresponding stochastic process is Poisson. Moreover, the number of free parameters of the $m$-state MMPP with a fixed initial probability vector is given by $m^2$. Although the MLL should increase as the number of phases increases, the result of the approximate method with 6 phases in port 135 does not increase. This is caused by the local maximum of LLF (see Remark 3.3). In this example, we use 10 different initial guesses that are randomly generated for each MMPP parameter estimation in the traffic analysis (The tables give the best among 10 initial guesses).
TABLE X  
ESTIMATION RESULTS (Port 80)

<table>
<thead>
<tr>
<th>Phase</th>
<th>exact MLL</th>
<th>exact AIC</th>
<th>approximate MLL</th>
<th>approximate AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-448955.0</td>
<td>897912.0</td>
<td>-24506.4</td>
<td>49020.8</td>
</tr>
<tr>
<td>2</td>
<td>-18175.7</td>
<td>36359.4</td>
<td>-20646.6</td>
<td>41311.2</td>
</tr>
<tr>
<td>3</td>
<td>-16736.3</td>
<td>33490.6</td>
<td>-20111.1</td>
<td>40254.2</td>
</tr>
<tr>
<td>4</td>
<td>-16171.2</td>
<td>32374.3</td>
<td>-19669.7</td>
<td>39389.4</td>
</tr>
<tr>
<td>5</td>
<td>-15573.8</td>
<td>31197.7</td>
<td>-19261.6</td>
<td>38595.2</td>
</tr>
<tr>
<td>6</td>
<td>-15394.1</td>
<td>30860.2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE XI  
SUMMARY OF ESTIMATED PARAMETERS (Port 135)

<table>
<thead>
<tr>
<th>Phase</th>
<th>exact MLL</th>
<th>exact AIC</th>
<th>approximate MLL</th>
<th>approximate AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-50397.6</td>
<td>100797.2</td>
<td>-39803.9</td>
<td>79615.8</td>
</tr>
<tr>
<td>2</td>
<td>-34974.9</td>
<td>69957.9</td>
<td>-38731.2</td>
<td>77480.4</td>
</tr>
<tr>
<td>3</td>
<td>-33671.1</td>
<td>67360.3</td>
<td>-37918.1</td>
<td>75868.2</td>
</tr>
<tr>
<td>4</td>
<td>-33650.1</td>
<td>67332.2</td>
<td>-37825.9</td>
<td>75701.8</td>
</tr>
<tr>
<td>5</td>
<td>-33402.9</td>
<td>66855.8</td>
<td>-37856.3</td>
<td>75784.6</td>
</tr>
<tr>
<td>6</td>
<td>-33292.4</td>
<td>66656.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE XII  
SUMMARY OF ESTIMATED PARAMETERS (Port 80)

<table>
<thead>
<tr>
<th>Phase</th>
<th>In</th>
<th>Out</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.084e-02</td>
<td>2.986e-01</td>
<td>8.457e+00</td>
</tr>
<tr>
<td>2</td>
<td>3.399e-03</td>
<td>7.517e-01</td>
<td>1.214e+02</td>
</tr>
<tr>
<td>3</td>
<td>2.810e-02</td>
<td>2.113e-02</td>
<td>9.638e-01</td>
</tr>
<tr>
<td>4</td>
<td>5.427e-01</td>
<td>3.692e-02</td>
<td>2.338e-27</td>
</tr>
<tr>
<td>5</td>
<td>2.973e-03</td>
<td>3.196e+00</td>
<td>2.636e+01</td>
</tr>
<tr>
<td>6</td>
<td>3.687e+00</td>
<td>7.478e-04</td>
<td>3.982e-16</td>
</tr>
</tbody>
</table>

TABLE XIII  
SUMMARY OF ESTIMATED PARAMETERS (Port 135)

<table>
<thead>
<tr>
<th>Phase</th>
<th>In</th>
<th>Out</th>
<th>Arrival</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.254e+00</td>
<td>1.664e-02</td>
<td>3.119e+13</td>
</tr>
<tr>
<td>2</td>
<td>3.832e-06</td>
<td>1.234e-05</td>
<td>3.607e-22</td>
</tr>
<tr>
<td>3</td>
<td>6.292e-03</td>
<td>7.742e-01</td>
<td>8.117e-01</td>
</tr>
<tr>
<td>4</td>
<td>9.101e-03</td>
<td>3.178e+00</td>
<td>2.954e+00</td>
</tr>
<tr>
<td>5</td>
<td>1.222e-03</td>
<td>3.137e-03</td>
<td>6.095e-59</td>
</tr>
<tr>
<td>6</td>
<td>1.341e-04</td>
<td>2.993e-01</td>
<td>4.953e+00</td>
</tr>
</tbody>
</table>

Fig. 6. The lag $k$ correlation for estimated MMPPs.

different results). However, all the initial guesses might not converge to the global maximum of LLF. In other words, we could not avoid the local maximum problem completely. This is an open problem for future research. Also, in ports 80 and 135, the AICs for the 6-state MMPPs still decrease. This indicates that there might be some better models among MMPPs with seven or more phases. Anyhow, from the viewpoint of AIC, we select the best models as the 6-state MMPPs estimated by the exact EM algorithms for ports 80 and 135. Tables XII and XIII show the summaries of the estimated MMPPs, where the columns of ‘In’ and ‘Out’ indicate total inbound and outbound rates from/to the other phases, respectively. According to these estimates, we expect that phase 2 in port 80 and phase 6 in port 135 represent burst arrivals and that phase 2 in port 135 indicates the long-term idling time that probably corresponds to nonworking days in the laboratory.

In Tables XII and XIII, it seems that the approximate method does not work well in the real traffic experiments. As discussed before, the accuracy of the approximate method depends on the length of time interval. In this real traffic example, the time interval is not sufficiently small to provide accurate estimates. However, the approximate method can also catch some statistical characteristics of arrival processes. Fig. 6, for example, depicts lag $k$ correlation of the estimated 6-state MMPPs. In both ports, the first several lag correlations rapidly decrease. On the other hand, the correlations after the first several ones keep high values. This shows long-range dependency in network traffic, and in particular, we find that the correlation of port 80 is much higher than that of port 135. In this figure, the MMPPs estimated by the approximate method also catch the similar lag $k$ correlation to the MMPPs estimated by the exact method.

Next, we perform traffic analysis based on a queueing model. We consider separate MMPP/D/1 queueing systems for respective ports. Figs. 7 and 8 illustrate complementary cumulative distribution functions of the queue length in buffer. Here, we draw empirical queue length distributions when applying the trace data to the input stream. Since group data do not contain exact interarrival times, we compute the empirical queue length distributions under which the group data form a sample path of a BMMPP. The other queue length distributions are analytically derived from the estimated 6-state MMPPs based on the exact EM algorithm. The traffic intensity is set as $0.1, 0.5,$ or $0.9$. For example, in the case of $0.1$, the service times at ports 80 and 135 are given by 1.74 and 10.93, respectively. In all the results, the analytical tails of queue length distributions are different from the empirical ones. Also, there are large differences between the analytical and empirical queue length distributions in the case of $0.1$ in port 135. In these examples, the queue lengths of trace data tend to be longer than those of MMPPs because the trace data are regarded as BMMPPs. However, even if we take account of the difference between BMMPP and MMPP, the difference of queue length distributions might be large. These imply that the accuracy of MMPP approximation is insufficient for the real traffic data and suggest that we need more phases for the MAP/MMPP approximation.

5We cannot draw the lag $k$ correlation from the trace data because group data do not have information about interarrival times.
Tables XIV and XV present buffer full probabilities and loss probabilities for the MMPP/D/1/K queueing systems with varying traffic intensities $\rho = 0.1, 0.5, 0.9$ for both ports. Similar to the queue length distribution, we present the empirical buffer full and loss probabilities based on the trace data where the input stream is assumed to be a BMMPP. The other results are derived from matrix-geometric analysis for MAP/D/1/K queueing system [2], [3]. We find that the buffer full and loss probabilities of the MMPPs further diverge from those of the trace data as the capacity $K$ increases. On the other hand, as a common feature of buffer full and loss probabilities of both MMPPs and trace data, the loss probability is much larger than the buffer full probability. That is, the buffer is rarely full, but most of the arriving packets are lost. This phenomenon is caused by burst arrivals. For example, in port 80, the ratio of the highest arrival rate over the lowest arrival rate is $5.192e + 28$. Such a large difference causes the phenomenon that most of the arriving packets with high arrival rate are lost. This also points to the difficulty of buffer size design in the Internet traffic.

**VI. CONCLUSION**

In this paper, we have proposed two EM algorithms for fitting MAP and MMPP with generalized group data. The proposed EM algorithm can perform the ML estimation when exact arrival times are not known. Moreover, in order to deal with the data consisting of many arrival observations in one bin, we have proposed the approximate EM algorithm for the special case of MMPP. In numerical experiments, we have shown the performance of the proposed EM algorithms. The numerical results have pointed out that the maximum number of arrivals strongly affects the computation time of the EM algorithm with group data and that length of step size for group data is a significant factor in determining the accuracy and the computation time in both exact and approximate EM algorithms. In addition, we have presented an application example of the proposed EM algorithm to real traffic data. We frequently encounter the situation where the number of arrivals is observed as a bin. Thus, the two methods proposed in this paper would be useful to estimate time series data arising in a variety of situations including the Internet traffic.

Although our method provides ML estimates of the MAP with group data effectively, the size of MAP is limited due to the computational effort needed. As shown in the traffic analysis example, we need a large number of phases to fit the MAP to the trace data with long-range dependence accurately. In the future, we plan to develop a faster estimation algorithm of the MAP using parallel computing in the context of ML estimation.

**REFERENCES**


Hiroiyuki Okamura (M’03) received the B.S.E., M.Eng., and Dr.Eng. degrees from Hiroshima University, Higashi-Hiroshima, Japan, in 1995, 1997, and 2001, respectively.

In 1998, he joined the Department of Industrial and Systems Engineering, Hiroshima University, as an Assistant Professor. Since 2003, he has been an Associate Professor in the Department of Information Engineering of the university’s Graduate School of Engineering.

Prof. Okamura served in the program committee of ValueTools2007 and ISSRE’08.

Tadashi Dohi (M’00) received the B.S.E., M.S., and Dr. of Engineering degrees from Hiroshima University, Higashi-Hiroshima, Japan, in 1989, 1991, and 1995, respectively.

In 1992, he joined the Department of Industrial and Systems Engineering, Hiroshima University, as an Assistant Professor. He has been working as a Full Professor in the Department of Information Engineering of the university’s Graduate School of Engineering since 2002. In 1992 and 2000, he was a Visiting Research Scholar at the University of British Columbia, Vancouver, BC, Canada, and Duke University, Durham, NC.

Dohi is serving as an Associate Editor of *IEEE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* (A) and *Asia-Pacific Journal of Operational Research* and as an Editorial Board Member of *Journal of Autonomic and Trusted Computing and Journal of Risk and Reliability*.

Kishor S. Trivedi (M’86–SM’87–F’92) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, in 1968, and the M. S. and Ph.D. degrees from the University of Illinois, Urbana, IL, in 1972 and 1974, respectively.

He holds the Hudson Chair in the Department of Electrical and Computer Engineering at Duke University, Durham, NC. He also holds a joint appointment in the Department of Computer Science at Duke. He is a co-designer of HARPS, SAVE, HARPE, SPNP, and SREPT modeling packages and is the author of a well-known text, *Probability and Statistics with Reliability, Queueing and Computer Science Applications* (thoroughly revised 2nd edition, New York: Wiley, 2002). His research interests are in reliability and performance assessment of computer and communication systems.

Dr. Trivedi was an Editor of the *IEEE Transactions on Computers* from 1983 to 1987, and he is on the Editorial Board of the *IEEE Transactions on Dependable and Secure Computing*. He is a Golden Core Member of the IEEE Computer Society and was the recipient of the society’s 2008 Technical Achievement Award.