An EM-Algorithm for MAP Fitting from Real Traffic Data*

Peter Buchholz[1]**

Fakultät für Informatik, TU Dresden, D-01062 Dresden, German
p.buchholz@inf.tu-dresden.de

Abstract. For model based analysis of computer and telecommunication systems an appropriate representation of arrival and service processes is very important. Especially representations that can be used in analytical or numerical solution approaches like phase type (PH) distributions or Markovian arrival processes (MAPs) are useful. This paper presents an algorithm to fit the parameters of a MAP according to measured data. The proposed algorithm is of the expectation-maximization (EM-) type and extends known approaches for the parameter fitting of PH-distributions and hidden Markov chains. It is shown that the algorithm generates MAPs which approximate traces very well and especially capture the autocorrelation in the trace. Furthermore the approach can be combined with other more efficient but less accurate fitting techniques by computing initial MAPs with those techniques and improving these MAPs with the approach presented in this paper.
Keywords: Markovian Arrival Process, EM-Algorithm, Data Fitting

1 Introduction

Traffic measurements show that real network traffic has a high variability and a non-negligible autocovariance over long distances. Different studies observed properties like *self-similarity*, *fractality* and *long range dependency* combined with *heavy tailed distributions* [11, 13]. The adequate capturing of these properties is one of the key aspects in modeling telecommunication systems since the approximation of real traffic by a Poisson process can result in a significant underestimation of response times and blocking probabilities. Consequently, a large number of traffic models has been developed in the last decade. Of particular interest are Markovian models for traffic description since they can be used in analytical models and they are easy to integrate in simulation models. Although Markovian models cannot describe non-exponential asymptotic behavior which sometimes has been observed, it seems that they can approximate these behaviors arbitrarily well over huge time scales and are therefore often an adequate model for the description of interarrival or service times. Furthermore, Markovian Arrival Processes (MAPs) can be used to describe and model correlated data

* In: P. Kemper, W. H. Sanders (eds.), Computer Performance Evaluation Modelling Techniques and Tools, Springer LNCS 2794 (2003), pp. 218-236.
** This research is partially supported by DFG, SFB 358

streams which often appear in practice. The major problem in the practical use of phase type distributions and MAPs is the appropriate parameterization of the distributions to match some measured traffic stream. Unfortunately, parameterization of general phase type distributions results in a nonlinear optimization problem such that usually only a few traffic characteristics are approximated by the phase type distribution. Especially the autocorrelation structure which is present in many real traffic streams is not considered in phase type approximations. In this paper we present a new heuristic approach to fit the parameters of a MAP according to some measured data stream. The proposed approach uses an EM-algorithm and is an extension of an approach that has been published recently for the fitting of parameters of hidden Markov chains [19].

The paper is structured as follows: In the next section basic definitions and notations are introduced. In this section also MAPs and their analysis are considered and a brief overview of basic fitting approaches for phase type distributions is given. Afterwards, in section 3, the new fitting procedure is introduced and its effort is analyzed. In section 4, the method is validated by different examples.

2 Basic Definitions and Notations

2.1 Basic Notation

Vectors and matrices are denoted as bold-faced small and capital letters. Elements are described using brackets. Vectors are row vectors, if not explicitly mentioned. $\mathbf{a}^T$ and $\mathbf{A}^T$ are the transposed vector $\mathbf{a}$ and matrix $\mathbf{A}$, respectively. $\mathbf{I}$ and $\mathbf{e}$ are the identity matrix and identity vector and $\mathbf{e}_x$ is a vector with 1 in position x and 0 elsewhere. Sets are denoted by calligraphic letters except the sets of (non-negative) real numbers $\mathbb{R}$ ($\mathbb{R}_+$) and natural numbers $\mathbb{N}$. $|\mathcal{S}|$ denotes the number of elements in set $\mathcal{S}$.

2.2 Data Sets and their Characteristics

We consider the fitting of traces resulting from measured interarrival or service times. A trace $\mathcal{T}$ is defined as a sequence of m times $t_i > 0$ $(i = 1, \ldots, m)$. The i-th moment of the trace and the variance are estimated as

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m}(t_j)^i \quad \text{and} \quad \sigma^2 = \frac{1}{m-1}\sum_{j=1}^{m}(t_j - \mu_1)^2 \ .$$

The autocorrelation of lag k is estimated from

$$\rho_k = \frac{1}{(m-k-1)\sigma^2}\sum_{j=1}^{m-k}(t_j - \mu_1)(t_{j+k} - \mu_1) \ .$$

The distribution function of a trace is a step function with m steps whose values are defined as

$$FT_{\mathcal{T}}(t) = \frac{\sum_{j=1}^{m}\delta(t_j \leq t)}{m}$$

where $\delta(b)$ for a boolean expression b equals 1 if the expression is true and 0 otherwise.

2.3 Markovian Arrival Processes (MAPs)

Let $Z(t)$ be an irreducible Markov chain with generator matrix $\mathbf{Q}$ and state space $\mathcal{S} = \{0, \ldots, n-1\}$. Furthermore let $\mathbf{D_0}$ and $\mathbf{D_1}$ be two matrices such that $\mathbf{D_1}$ is non-negative, $\mathbf{Q}(x,y) - \mathbf{D_1}(x,y) \geq 0$ for $x \neq y$, $\mathbf{D_1}(x,x) \geq 0$ and $\mathbf{D_0} = \mathbf{Q} - \mathbf{D_1}$. $\mathbf{D_0}$ is the generator matrix of an absorbing Markov chain. Matrix $\mathbf{D_0}$ includes as off-diagonal elements transition rates of the Markov chain that do not correspond to an arrival and in the main diagonal the negative sum of all arrival rates out of a state are collected. Matrix $\mathbf{D_1}$ contains transition rates that are accompanied by an arrival. Consequently, we define a MAP as $\mathcal{M} = (\mathbf{D_0}, \mathbf{D_1})$. The class of MAPs contains as subclasses all the different distributions of the phase type like hyperexponential-, Erlang-, Cox- or PH-distributions and it also contains MMPP which are MAPs where $\mathbf{D_1}$ is a diagonal matrix.

The stationary distribution of a MAP is defined as the solution of $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ with $\boldsymbol{\pi}\mathbf{e}^T = 1.0$. From the stationary distribution the distribution immediately after an arrival can be computed as $\boldsymbol{\pi}' = \boldsymbol{\pi}\mathbf{D_1}/(\boldsymbol{\pi}\mathbf{D_1}\mathbf{e}^T)$. The column vector of the i-th conditional moments is given as the solution of the set of equations $\mathbf{D_1}\mathbf{m}^{(i)} = -i \cdot \mathbf{m}^{(i-1)}$ where $\mathbf{m}^{(0)} = \mathbf{e}^T$ and $\mathbf{m}^{(i)}$ is the column vector of conditional i-th moments. I.e., if the MAP is in state x, then $\mathbf{m}^{(i)}(x)$ is the i-th moment of the time to the next arrival. The absolute moments are then given by $E[T^{(i)}] = \boldsymbol{\pi}'\mathbf{m}^{(i)}$. The autocorrelation of order or lag k ($k \geq 1$) is computed from [12]

$$E[\rho_k] = \frac{E[T^{(1)}]\boldsymbol{\pi}\left(\left(-\mathbf{D_0}^{-1}\mathbf{D_1}\right)^k \mathbf{m}^{(1)} - \mathbf{e}^T E[T^{(1)}]\right)}{E[T^{(2)}] - (E[T^{(1)}])^2}.$$

The value of the distribution function at time $t \geq 0$ results from $FT(t) = \boldsymbol{\pi}'\mathbf{D_0}[t]\mathbf{e}^T$ with $\mathbf{D_0}[t] = exp(\mathbf{D_0}t)$ which is usually computed using the randomization technique [18] with the relation

$$\mathbf{D_0}[t] = \sum_{k=0}^{\infty} \beta(k, \alpha t)(\mathbf{P_0})^k$$

where $\mathbf{P_0} = \mathbf{D_0}/\alpha + \mathbf{I}$, $\mathbf{P_1} = \mathbf{D_1}/\alpha$, $\alpha \geq max_{x \in \mathcal{S}}|\mathbf{D_0}(x,x)|$ and $\beta(k, \alpha t) = exp(-\alpha t)(\alpha t)^k/k!$, the probability of k jumps of a Poisson process with rate α in the interval $[0, t)$. For a practical implementation, the infinite sum is truncated from the left by starting the summation with value $l \geq 0$ and to the right by ending the summation with value $r < \infty$ to compute the Poisson probabilities up to machine precision [5]. The number of required iteration is in $O(\alpha t)$ for large values of αt. Thus, in randomization, probabilities are computed from a discrete time Markov chain and a Poisson process. This feature will be exploited for the development of an EM-algorithm for MAP-parameter fitting.

2.4 Fitting Procedures for phase type distributions and MAPs

A large number of methods exist to fit PH-distributions according to measured data [6, 10]. Methods can be distinguished whether they perform the fitting based

on the complete trace or by using some information extracted from the trace. Most known methods belong to the second class and approximate some quantities of the data by an appropriate phase type distribution. However, if only some quantities are matched, it is not clear which quantities of a trace are important and which are not. Some methods exist to use the whole trace for parameter fitting, most of these methods belong to the class of EM-methods like the approach of this paper. Examples of those approaches are [3] where hyperexponential distributions are fitted to approximate Pareto or Weibull distributions and [8] where the former approach is extended by fitting hyperexponential distributions directly from measured data. Another EM-approach which fits phase type distributions by minimizing the Kullback-Leibler distance of the observed empirical distribution and a phase type distribution is given in [2]. In [14] an EM-algorithm for PH fitting is presented which decomposes traces first into subsets to improve efficiency.

The fitting of MAPs in principle can be performed similarly to the fitting of PH-distributions. Unfortunately, MAP-fitting seems to be much more complex than PH-fitting. Nevertheless some older methods are available to fit specific MAPs or MMPPs with few phases [4, 16]. More recently EM-algorithms for the use in performance analysis have been developed. The approaches published in [20, 17] describe packet losses in communication systems by DTMCs and estimate the parameters of the DTMC by an EM-algorithms. In [9, 15] EM-algorithms are presented to fit a batch Markovian arrival process (BMAP) from measured data. The approach presented in [9] is similar to our approach because it also uses randomization in conjunction with an EM-algorithm, but the used EM-algorithm differs slightly from our approach. Our technique is mainly influenced by [19] where an approach is described to model delays in communication systems by continuous time hidden Markov chains. In this approach first the parameters of a discrete time hidden Markov chain are estimated with an EM-algorithm and in a second step the discrete time chain is transformed into a continuous time hidden Markov process. However, in contrast to [19] we directly estimate the parameters of the continuous time process which avoids the often crucial step of transforming a discrete in a continuous time model. Furthermore, the result of our fitting procedure is a MAP and not a hidden Markov chain which is much more natural if times have to be described.

3 EM-Fitting of MAPs

3.1 The likelihood of an observed sequence

In [19] conditional state probabilities for being in a specific state after observing an initial part of the trace and probabilities of observing the remaining part of a trace depending on the current state are given for discrete time hidden Markov chains where elements of the trace are defined by observable states. Our approach has to consider transitions instead of observable states and we apply the approach to the matrices of the continuous time Markov chain after using randomization. Therefore we compute some form of joint densities or likelihoods rather than probabilities. Let $\mathbf{a}^{(i)}$ be a row vector including in position x the

likelihood of state x immediately after observing $t_1, \ldots, t_i$ $(i \leq m)$

$$\mathbf{a}^{(i)} = \mathbf{a}^{(i-1)}\mathbf{D_0}[t_i]\mathbf{P_1} \text{ with } \mathbf{a}^{(0)} = \boldsymbol{\pi}' \ . \tag{1}$$

If vector $\mathbf{a}^{(i)}$ is normalized to 1.0, then it includes the conditional state probabilities after the i-th observation. Similarly we define the backward likelihood in a column vector $\mathbf{b}^{(i)}$.

$$\mathbf{b}^{(i)} = \mathbf{D_0}[t_i]\mathbf{P_1}\mathbf{b}^{(i+1)} \text{ with } \mathbf{b}^{(m)} = \mathbf{e}^T \ . \tag{2}$$

The normalized vector $\mathbf{b}^{(i)}$ includes in position $\mathbf{b}^{(i)}(x)$ the probability of observing sequence $t_i, \ldots, t_m$ starting in state x.

The quality of a MAP $\mathcal{M}$ for the approximation of a trace $\mathcal{T}$ can be measured by the value of the likelihood

$$d(\mathcal{T}, \mathcal{M}) = \alpha^m \mathbf{a}^{(m)}\mathbf{e}^T \tag{3}$$

where $\mathbf{a}^{(m)}$ is computed from $\mathcal{M}$ as introduced above and α is the rate used for randomization. Observe that the measure is independent of α for a given MAP $\mathcal{M}$. MAP $\mathcal{M}$ is a better approximation for trace $\mathcal{T}$ than MAP $\mathcal{M}'$ if $d(\mathcal{T}, \mathcal{M}) > d(\mathcal{T}, \mathcal{M}')$. Thus, a procedure for MAP fitting has to maximize $d(\mathcal{T}, \mathcal{M})$ which, unfortunately, is a complex, highly non-linear optimization problem.

3.2 An EM-algorithm for parameter fitting

The basic idea of our EM-algorithm is to compute the transition likelihoods between states according to transitions with and without arrivals and use the normalized likelihoods as estimates for the transition probabilities in the matrices $\mathbf{P_0}$ and $\mathbf{P_1}$. According to the jumps of the Poisson process we define a row vector $\mathbf{v}^{(i),k} = \mathbf{a}^{(i)}\left(\mathbf{P_0}\right)^k$ describing the *forward likelihood* after k internal transitions in the i-th interval. Similarly we can define a *backward likelihood* as a column vector $\mathbf{w}^{(i),k} = \left(\mathbf{P_0}\right)^k \mathbf{P_1}\mathbf{b}^{(i+1)}$. From these vectors, two $n \times n$ matrices of transition likelihoods can be defined elementwhise as

$$\mathbf{X_0}^{(i)}(x,y) = \sum_{k=l_i}^{r_i-1} \beta(k, \alpha t_i) \sum_{l=0}^{k-1} \mathbf{v}^{(i),l}(x)\mathbf{P_0}(x,y)\mathbf{w}^{(i),k-l-1}(y) \tag{4}$$

and

$$\mathbf{X_1}^{(i)}(x,y) = \sum_{k=l_i}^{r_i} \beta(k, \alpha t_i)\mathbf{v}^{(i),k}(x)\mathbf{P_1}(x,y)\mathbf{b}^{(i+1)}(y) \tag{5}$$

where l_i and r_i are the left and right truncation point for the computation of the Poisson probabilities in the interval $[0, t_i)$. $\mathbf{X_0}^{(i)}$ contains estimates for the likelihood of transitions without arrivals and $\mathbf{X_1}^{(i)}$ estimates for the likelihood of transitions with arrivals. The likelihood values are collected in matrices $\mathbf{Y_0}$ and $\mathbf{Y_1}$ such that

$$\mathbf{Y_0} = \sum_{i=1}^{m} \mathbf{X_0}^{(i)} \quad \text{and} \quad \mathbf{Y_1} = \sum_{i=1}^{m} \mathbf{X_1}^{(i)}$$

The normalized matrices

$$\hat{\mathbf{Y}}_\mathbf{0}(x,y) = \frac{\mathbf{Y_0}(x,y)}{\sum_{z=0}^{n-1} \mathbf{Y_0}(x,z) + \sum_{z=0}^{n-1} \mathbf{Y_1}(x,z)} \;,\; \hat{\mathbf{Y}}_\mathbf{1}(x,y) = \frac{\mathbf{Y_1}(x,y)}{\sum_{z=0}^{n-1} \mathbf{Y_0}(x,z) + \sum_{z=0}^{n-1} \mathbf{Y_1}(x,z)} \tag{6}$$

can afterwards be used as new estimates for $\mathbf{P_0}$ and $\mathbf{P_1}$. Observe that $\hat{\mathbf{Y}}_\mathbf{0} + \hat{\mathbf{Y}}_\mathbf{1}$ is a stochastic matrix.

By combination of the introduced steps we obtain an iterative algorithm to fit the parameters of a MAP according to some trace $\mathcal{T}$.

1. Input: trace $\mathcal{T} = (t_1, \ldots, t_m)$
2. set basic rate α, e.g., $\alpha = \min_{i=1,\ldots,m}((t_i)^{-1})$;
3. choose randomly $\mathbf{P_0} \geq \mathbf{0}$, $\mathbf{P_1} \geq \mathbf{0}$ such that $\mathbf{P_0} + \mathbf{P_1}$ is stochastic ;
4. repeat
5. compute $\boldsymbol{\pi}'$ and set $\mathbf{a}^{(0)} = \boldsymbol{\pi}'$;
6. for $i = 1$ to $m - 1$ do
7. compute $\mathbf{a}^{(i)}$ via (1) ;
8. $\mathbf{b}^{(m)} = \mathbf{e}^T$;
9. for $i = m - 1$ downto 1 do
10. compute $\mathbf{b}^{(i+1)}$ via (2) ;
11. compute $\mathbf{X_0}^{(i)}$ via (4) and $\mathbf{X_1}^{(i)}$ via (5) ;
12. $\mathbf{Y_0} = \mathbf{Y_0} + \mathbf{X_0}^{(i)}$ and $\mathbf{Y_1} = \mathbf{Y_1} + \mathbf{X_1}^{(i)}$;
13. ${\mathbf{P_0}}^{old} = \mathbf{P_0}$ and ${\mathbf{P_1}}^{old} = \mathbf{P_1}$;
14. $\mathbf{P_0} = \hat{\mathbf{Y}}_\mathbf{0}$ and $\mathbf{P_1} = \hat{\mathbf{Y}}_\mathbf{1}$;
15. until $\max\left(\|\mathbf{P_0} - {\mathbf{P_0}}^{old}\|, \|\mathbf{P_1} - {\mathbf{P_1}}^{old}\|\right) \leq \epsilon$;
16. $\mathbf{D_0} = \alpha\left(\mathbf{P_0} - \left(diag(\mathbf{P_0}\mathbf{e}^T + \mathbf{P_1}\mathbf{e}^T)\right)\right)$ and $\mathbf{D_1} = \alpha\mathbf{P_1}$;
17. Output: $\mathcal{M} = (\mathbf{D_0}, \mathbf{D_1})$;

3.3 Improvements and Extensions

Since the problem of parameter fitting of MAPs is complicated and results in a nonlinear optimization problem, we cannot expect to find the global optimum in a single run of the algorithm. However, the approach is very flexible, since it can be started with an arbitrary MAP as input and the quality of the approximation can be measured by computing $d(\mathcal{T}, \mathcal{M})$. Thus, it is usually a good idea to start the algorithm with different MAPs and perform the optimization for the whole trace or a subsequence of the trace to observe the values $d(\mathcal{T}, \mathcal{M})$ for the resulting MAPs. The best MAP, or if different MAPs with similar d-values are generated these MAPs, can afterwards by used as input for further steps of the algorithm. Alternatively one may start with some MAP resulting from any other method to approximate the trace. E.g., one may use one of the methods presented in [8, 3, 6, 15, 21] to fit first some characteristics of the trace and use the algorithm to improve the MAP or PH-distribution resulting from this fitting. With the choice of the initial MAP, also the number of states is fixed. Since the number of states determines the complexity of the algorithm and the complexity of the optimization problem, a small dimension of the state space reduces the effort,

but might yield a bad approximation. It is very hard to determine a priori the appropriate dimension of the MAP and it is not even clear that an increased number of states always yields a better approximation. This has already been noticed in [19].

Additionally the complexity of the algorithm and also the complexity of the optimization problem depends also the number of non-zero elements in the matrices $\mathbf{P_0}$ and $\mathbf{P_1}$. If one element $\mathbf{P_0}(x,y)$ or $\mathbf{P_1}(x,y)$ becomes 0 during the algorithm or is set to 0 right from the beginning, then it will remain 0, because the corresponding values ${\mathbf{X_0}}^{(i)}(x,y)$ computed via (4) and ${\mathbf{X_1}}^{(i)}(x,y)$ computed via (5) will remain 0. If the algorithm is started with sparse matrices, the structure is preserved or additional zero elements are introduced. Thus, the approach may as well be used to fit MMPPs instead of MAPs by starting with an MMPP.

The effort of the approach and also the quality of the fitting depend, among other parameters, on the choice of α. In the presented algorithm α is chosen equal to the inverse of the minimum value in the trace. This choice results in a MAP which matches the first moment of the trace, but may yield to long run times. A smaller value of α improves the runtime, but may yield a bad approximation of the trace. However, the first moment can alway be exactly matched by rescaling α. Assume that the procedure yields a MAP with matrices $\mathbf{P_0}$ and $\mathbf{P_1}$. Let $\boldsymbol{\pi}$ the stationary distribution of the MAP, then the first moment equals $E[T^{(1)}] = \boldsymbol{\pi} \cdot \mathbf{P_1}\mathbf{e}^T/\alpha$. If μ_1 is the first moment of the trace, then setting α to $\alpha \cdot E[T^{(1)}]/\mu_1$ yields a new MAP with the same matrices $\mathbf{P_0}$ and $\mathbf{P_1}$, but different matrices $\mathbf{D_0}$ and $\mathbf{D_1}$ which matches exactly the first moment of the trace.

Although the intention of the algorithm is the fitting of processes with some autocorrelation structure and not the fitting of distributions, the technique can also be used to fit distributions. In this case, a PH distribution rather than a MAP is generated. This means that $\mathbf{D_1} = {\mathbf{d_1}}^T\boldsymbol{\pi'}$ where $\mathbf{d_1}$ is a column vector of length n such that $\mathbf{D_0}\mathbf{e}^T + \mathbf{d_1} = \mathbf{0}$. Let $\mathbf{p_1} = \mathbf{d_1}/\alpha$ such that $\mathbf{P_0} + \mathbf{p_1} \cdot \boldsymbol{\pi'}$ is a stochastic matrix. Instead of $\mathbf{P_1}$ now $\mathbf{p_1}$ and $\boldsymbol{\pi'}$ have to be fitted by the algorithm. Since values from a distribution are assumed to be independent, it is not necessary to consider a sequence of times in a trace, instead one can initialize each $\mathbf{a}^{(i)}$ with $\boldsymbol{\pi'}$ and $\mathbf{b}^{(i)}$ with $\mathbf{e}^T$ and perform with these values the computation of the matrix ${\mathbf{X_0}}^{(i)}$ and of the vectors ${\mathbf{x_1}}^{(i)}$ and $\mathbf{z}^{(i)}$ which are used to compute new values for $\mathbf{p_1}$ and $\boldsymbol{\pi'}$. Values are computed via

$$ {\mathbf{x_1}}^{(i)}(x) = \sum_{k=l_i}^{r_i} \beta(k, \alpha t_i)\mathbf{v}^{(i),k}(x)\mathbf{p_1}(x) \tag{7} $$

and vector $\mathbf{z}^{(i)}$ is computed via

$$ \mathbf{z}^{(i)} = \sum_{k=l_i}^{r_i} \beta(k, \alpha t_i)\mathbf{w}^{(i),k} \text{ where } \mathbf{w}^{(i),k} = {(\mathbf{P_0})}^k {\mathbf{p_1}}^T . \tag{8} $$

Define the weight of the i-th transition as

$$ \xi^{(i)} = \left(\sum_{k=l_i}^{r_i} {\mathbf{x_1}}^{(i)}(x) \right)^{-1} . \tag{9} $$

From the weights $\xi^{(i)}$, the vectors $\mathbf{x_1}^{(i)}$, $\mathbf{z}^{(i)}$ and the matrix $\mathbf{X_0}^{(i)}$, which is generated as shown in (4) using vectors $\mathbf{w}^{(i),k}$ from (8), a new vector $\mathbf{p_1}$ and a new matrix $\mathbf{P_0}$ are generated.

$$\mathbf{P_0}(x,y) = \frac{\sum_{i=1}^{m} \xi^{(i)} \mathbf{X_0}^{(i)}(x,y)}{\sum_{i=1}^{m} \sum_{z=0}^{n-1} \xi^{(i)} \mathbf{X_0}^{(i)}(x,z) + \sum_{i=1}^{m} \sum_{z=0}^{n-1} \xi^{(i)} \mathbf{x_1}^{(i)}(z)} \tag{10}$$

and

$$\mathbf{p_1}(x) = \frac{\sum_{i=1}^{m} \xi^{(i)} \mathbf{x_1}^{(i)}(x)}{\sum_{i=1}^{m} \sum_{z=0}^{n-1} \xi^{(i)} \mathbf{X_0}^{(i)}(x,z) + \sum_{i=1}^{m} \sum_{z=0}^{n-1} \xi^{(i)} \mathbf{x_1}^{(i)}(z)} \tag{11}$$

The new vector $\boldsymbol{\pi}'$ is chosen in a way that the likelihood is maximized.

$$\boldsymbol{\pi}'(x) = \begin{cases} 1 \text{ if } \mathbf{z}^{(i)}(x) > \mathbf{z}^{(i)}(y) \text{ for all } y = 0, \ldots, n-1 \\ 0 \text{ otherwise} \end{cases} \tag{12}$$

If $\mathbf{z}^{(i)}$ contains more than one maximal element, one of these elements is chosen. Alternatively one my fix the vector right from the beginning of the algorithm and generate the PH-distribution according to the fixed vector.

3.4 Effort of the Fitting Procedure

In this subsection we briefly analyze the asymptotic effort per iteration and the memory requirements of the algorithm. We assume that $m \gg n$ which is a realistic assumption. Since the vectors $\mathbf{a}^{(i)}$ are computed starting with $i = 0$ and the vectors $\mathbf{b}^{(i)}$ are computed starting with $i = m$, one of both series of vectors has to be precomputed and stored which requires memory in $O(nm)$. We assume that vectors $\mathbf{a}^{(i)}$ are first computed. Thus, each iteration starts with the computation of $\boldsymbol{\pi}'$ with an effort in $O(n^3)$. For the computation of the vectors $\mathbf{a}^{(i)}$ the randomization approach has to be applied and $O(t_i \alpha)$ vector matrix products have to be computed to derive $\mathbf{a}^{(i)}$ from $\mathbf{a}^{(i-1)}$. Let $t_{av} = 1/m \cdot \sum_{i=1}^{m} t_i$, then all vectors $\mathbf{a}^{(i)}$ are computed in $O(m t_{av} \alpha n^2)$. Starting with $i = m$ vectors $\mathbf{b}^{(i)}$ are computed and for each $\mathbf{b}^{(i)}$ matrix elements $\mathbf{X_0}^{(i)}$ and $\mathbf{X_1}^{(i)}$ are computed. Vector $\mathbf{b}^{(i)}$ is computed from $\mathbf{b}^{(i+1)}$ with $O(t_i \alpha)$ iterations such that the computation of all vectors $\mathbf{b}^{(i)}$ requires the same effort than the computation of all vectors $\mathbf{a}^{(i)}$. Computation of the matrix elements in step i requires an effort in $O(t_i \alpha n^2)$ if $r_i = 0$. For values $t_i \geq \alpha$, the lower index for summation grows (i.e., $r_i > 0$) and the effort shrinks. The normalization of matrix elements requires an effort in $O(n^2)$ which is negligible. Consequently, the overall effort for the computation of the matrix elements is in $O(m t_{av} \alpha n^2)$ which is also the effort for one iteration of the complete algorithm. Vectors $\mathbf{b}^{(i)}$ can be stored in the same locations as $\mathbf{a}^{(i)}$ and apart from the vectors only a few matrices need to be stored such that memory requirements for the algorithm remains in $O(mn)$.

4 Experimental Validation

Since the proposed approach is a heuristic which is obviously not guaranteed to find the optimal solution, it is important to validate the results of the approach by means of examples. We consider three different classes of problems for validation. First independent data drawn from different distributions is used as input. For these examples PH-distributions are generated using (10)-(12). Afterwards data is drawn from MAPs, resulting in correlated samples. From these traces MAPs are generated. In a last series of experiments, traces resulting from measurements are used as input. Afterwards we compare the quality of the generated MAPs by analyzing the performance measures for a simple queueing network which is fed with original and the fitted MAP resulting from the EM-algorithm.

We do not present the effort which is required for fitting the different distributions because the number of iterations is very sensitive for the initial MAP, the choice of α and the stopping criterion. Small changes in these parameters result in different runtimes of the algorithm. However, it is interesting to note that the resulting MAP seems to be relatively robust concerning the parameters. Often the resulting MAPs are identical up to the ordering of states. A good strategy is to choose for the first steps of the EM-algorithm a small value of α since this results in less iterations in the randomization steps. If convergence is observed such that the value $d(\mathcal{T}, \mathcal{M})$ does not change during some iterations, α is increased for a better approximation. This approach often yields a faster convergence and converges towards a MAP which is identical to the MAP resulting from the procedure with a large value for α right from the beginning. However, this observation is not always true. For the presented examples with $10^3 - 10^4$ samples in the trace the runtimes of the EM-algorithm for $\epsilon = 10^{-6}$ are in the range of 5 minutes to 5 hours on a standard PC.

4.1 Independent Samples from Known Distributions

In this subsection we consider the approximation of four different distributions by phase type distributions with varying numbers of phases. In all cases the PH-distributions are generated from a randomly generated PH-distribution with dense matrix $\mathbf{D_0}$ and a vector $\mathbf{d_1}$ where all elements are non-zero. All initial PH-distributions result from the same setting of the random number generator.

The first two distributions we approximate are a lognormal distribution with mean 1.0 and standard deviation 1.5 and uniform distribution on the interval $[0.5, 1.5]$. From both distributions we draw a sample of size 10^4 and generate different PH-distributions from these samples. Table 1 contains the first three moments of the samples and the moments of the generated PH-distributions. Furthermore $\ln(d(\mathcal{T}, \mathcal{M}))$ is given as measure of the difference between sample and approximating PH-distribution. Observe that a larger value of $d(\mathcal{T}, \mathcal{M})$ indicates a better approximation.

In figure 1 the densities of the traces and the corresponding PH-distributions are shown. Densities of traces are approximated by histograms with intervals of width 0.1. For the lognormal distribution, the approximations are good. By

	Lognormal distribution			Uniform distribution			
	Trace	PH 2	PH 5	Trace	PH 5	PH 10	PH 20
$E[T^1]$	1.004e+0	1.004e+0	1.004e+0	9.960e-1	9.960e-1	9.960e-1	9.960e-1
$E[T^2]$	4.154e+0	3.308e+0	3.055e+0	1.076e+0	1.488e+0	1.191e+0	1.136e+0
$E[T^3]$	1.075e+2	2.722e+1	2.027e+1	1.240e+0	2.964e+0	1.664e+0	1.461e+0
$\ln(d(\mathcal{T},\mathcal{M})$		-9.491e+3	-9.113e+3		-6.556e+3	-3.121e+3	-2.332e+3

Table 1. Moments and d-values for the lognormal and uniform distribution.

inspection of the different densities it becomes clear that a better approximation of the moments does not necessarily yield a better approximation of the distribution. For the uniform distribution the approximations are less good, although the values for $d(\mathcal{T},\mathcal{M})$ are relatively small for this distribution. The reason is that continuous phase type distributions always have an infinite support and can therefor not exactly approximate distributions with a finite support like the uniform distribution.

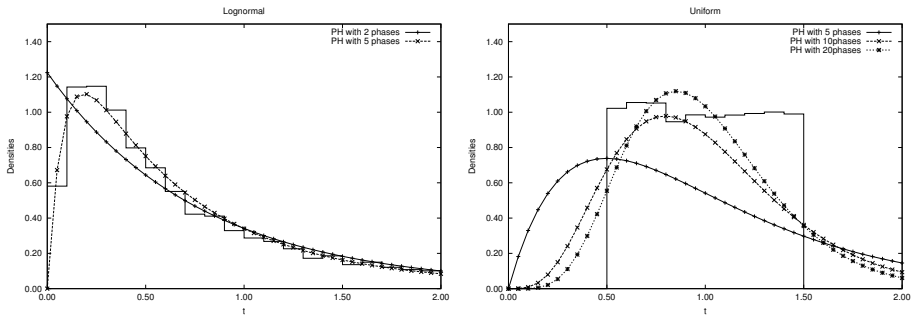

Fig. 1. Densities of the traces and the PH-approximations for the first two examples

Next we consider the fitting of two different Weibull-distributions. The first has a shape of 5 and a scale parameter of 1 and the second has a shape of 0.66464 and a scale parameter of 0.6. The second distribution, which is also used as an example in [3], has a decreasing density whereas the density of the first distribution is first increasing and then decreasing. For both distributions a trace with 10^4 elements is generated and afterwards the EM-algorithm is applied.

For the first version of the Weibull distribution we start with randomly generated phase type distributions with 2, 5, 10 and 20 phases. The results for this distribution are shown in the upper half of table 2 and the resulting densities are printed on the left side of figure 2. The PH-approximation becomes better with an increasing number of phases which can be seen by comparing the measures $d(\mathcal{T},\mathcal{M})$ and by comparing the resulting densities of the phase type distributions with the empirical density of the trace.

The second Weibull distribution has decreasing density function such that the efficient fitting methods of [3, 8] can be applied. As shown in [3] the Weibull distribution can be matched very well with a 6 hyperexponential distribution with 6 phases. This hyperexponential distribution can be transformed into an

	Weibull distribution $(5.0, 1.0)$				
	Trace	PH 2	PH 5	PH 10	PH 20
$E[T^1]$	9.381e-1	9.381e-1	9.381e-1	9.381e-1	9.381e-1
$E[T^2]$	9.261e-1	1.320e+0	1.056e+0	9.688e-1	9.437e-1
$E[T^3]$	9.519e-1	2.476e+0	1.387e+0	1.092e+0	1.011e+0
$\ln(d(\mathcal{T}, \mathcal{M})$		-5.806e+2	-1.902e+2	1.166e+1	7.534e+1
	Weibull distribution $(0.66464, 0.6)$				
	Trace	Cox_FW	Cox_1	Cox_2	PH
$E[T^1]$	9.498e-1	9.498e-1	9.498e-1	9.498e-1	9.498e-1
$E[T^2]$	3.227e+0	3.658e+0	3.260e+0	3.221e+0	3.097e+0
$E[T^3]$	1.920e+1	3.036e+1	2.048e+1	1.922e+1	1.684e+1
$ft_{\mathcal{M}}(0)$		2.651e+1	2.551e+1	1.467e+1	1.346e+1
$\ln(d(\mathcal{T}, \mathcal{M})$		-6.823e+2	-6.801e+2	-6.818e+2	-6.793e+2

Table 2. Moments and d-values for the Weibull distributions.

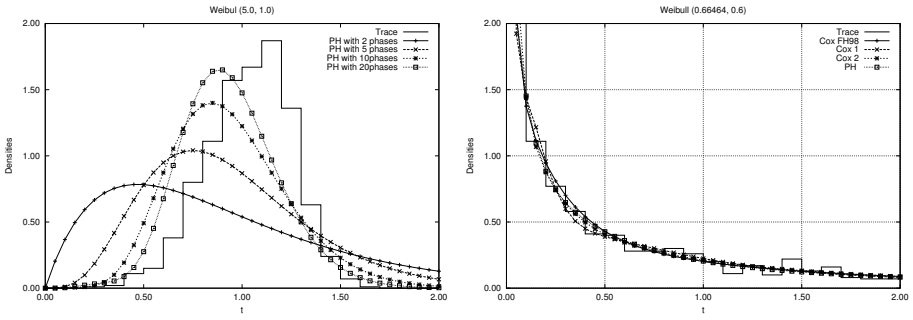

Fig. 2. Densities of the traces for the Weibull distributions and PH-approximations

equivalent Coxian distribution with 6 phases [7][1]. The corresponding Coxian distribution where the value of α is scaled to fit exactly the first moment of the trace is denoted as *Cox_FW*. Distribution *Cox_1* results from the use of the EM-algorithm with *Cox_FW* as initial distribution. *Cox_2* is a Coxian distribution which is computed from a randomly generated initial Coxian distribution with 6 phases. About 10 times more iterations are required to generate *Cox_2* compared to *Cox_1*. The last distribution, denoted as *PH* results from a PH-distribution with 6 phases. The densities of the different distributions are shown on the right side of figure 2. Obviously all densities are very similar and match the trace very well. The remaining quantities of the distributions are shown in table 2. The value $d(\mathcal{T}, \mathcal{M})$ is minimal for the *PH* and *Cox_FW* yields the largest value. However, the differences are small which shows that the method of [3] gives excellent results for this kind of distributions.

[1] We use Coxian instead of hyperexponential distributions because they can be described more easily using the matrices $\mathbf{D_0}$ and $\mathbf{D_1}$.

4.2 Fitting of Data Generated from MAPs

If traces are generated from MAPs or MMPPs, then a significant autocorrelation often exists and cannot be neglected. The same holds for many real data sets. In this subsection we fit one MAP and one MMPP using the proposed method, before the approach is applied to real data in the following subsection. For both examples one MAP and one MMPP are fitted starting from a randomly generated MAP or MMPP, respectively. First we consider a MAP with 3 states and the following matrices.

$$\mathbf{D_0} = \begin{pmatrix} -3.721 & 0.500 & 0.020 \\ 0.100 & -1.206 & 0.005 \\ 0.001 & 0.002 & -0.031 \end{pmatrix} \quad \text{and} \quad \mathbf{D_1} = \begin{pmatrix} 0.200 & 3.000 & 0.001 \\ 1.000 & 0.100 & 0.001 \\ 0.005 & 0.003 & 0.020 \end{pmatrix}$$

From this MAP we generate a sample with $5 \cdot 10^3$ elements which is used for fitting a MMPP and a MAP of order 3. Table 3 includes the results of the fitting procedure. The d-value in the column trace describes the d-value of the original MAP computed for the trace. It is interesting to note that the d-value of the fitted MAP is slightly larger than the d-value of the original MAP, whereas the d-value for the fitted MMPP is smaller. The autocorrelation of lag 2 of the trace overestimates the true value, whereas the lag 1 autocorrelation is underestimated.

	orig. MAP	Trace	MAP 3	MMPP 3
$E[T^1]$	1.020e+0	1.068e+0	1.068e+0	1.068e+0
$E[T^2]$	2.717e+1	2.729e+1	2.594e+1	2.642e+1
$E[T^3]$	2.544e+3	2.288e+3	2.014e+3	2.098e+3
ρ_1	0.30409	0.24200	0.35475	0.35648
ρ_2	0.19824	0.26829	0.27334	0.26806
$\ln(d(\mathcal{T}, \mathcal{M})$	-2.850e-3		-2.835e+3	-2.924e+3

Table 3. Moments and d-value for the 3 state MAP.

The fitted MAP is described by the matrices

$$\mathbf{D_0}^{(MAP)} = \begin{pmatrix} -0.873 & 0.006 & 0.000 \\ 0.007 & -0.037 & 0.001 \\ 0.000 & 0.002 & -3.653 \end{pmatrix}, \; \mathbf{D_1}^{(MAP)} = \begin{pmatrix} 0.033 & 0.000 & 0.834 \\ 0.000 & 0.028 & 0.001 \\ 3.636 & 0.008 & 0.007 \end{pmatrix}$$

and the matrices of the fitted MMPP are

$$\mathbf{D_0}^{(MMPP)} = \begin{pmatrix} -2.826 & 0.000 & 0.645 \\ 0.000 & -0.037 & 0.009 \\ 1.839 & 0.026 & -1.865 \end{pmatrix}, \; \mathbf{D_1}^{(MMPP)} = \begin{pmatrix} 2.181 & 0.000 & 0.000 \\ 0.000 & 0.028 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{pmatrix}$$

In both cases only values ≥ 0.0005 are printed as non-zero values, but most values printed as 0 are very small and can be neglected. Figure 3 includes the densities of the original MAP, the two fitted processes and the approximated density of the trace. Both fitted processes approximate the true density of the MAP and the approximated density of the trace very well, such that not difference between true and fitted density is visible in the figure. Furthermore figure

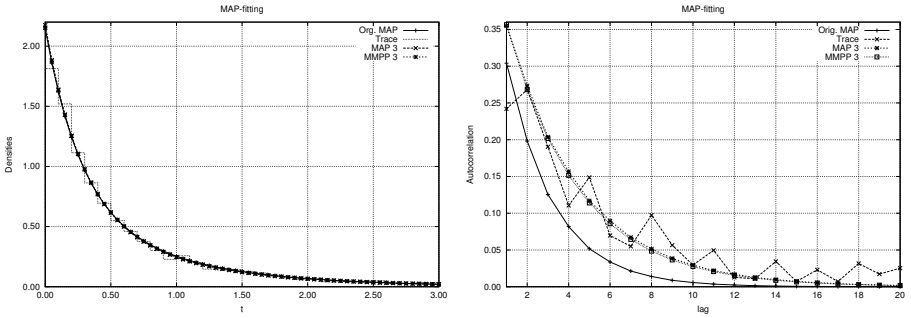

Fig. 3. Densities and autocorrelations of the 3 state MAP and of the fitted processes.

3 shows the autocorrelations of lag 1 through 20. The autocorrelations of the MAPs and the MMPP are decreasing function with a very regular structure, whereas the autocorrelations of the trace are much more irregular. Furthermore there is a significant difference between the autocorrelations of the MAP and the autocorrelations of the trace generated from this MAP. Consequently, for an accurate representation of the autocorrelation structure a much larger sample size seems to be necessary.

As a second example we consider the fitting of a MMPP with 6 states. The MMPP is described by the following matrices.

$$\mathbf{D_0} = \begin{pmatrix} -0.1 & 0.1 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.2 & -1.3 & 0.1 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.2 & -2.3 & 0.1 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & -3.3 & 0.1 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.2 & -4.3 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.2 & -5.2 \end{pmatrix}, \mathbf{D_1} = \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 2.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 3.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 4.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 5.0 \end{pmatrix}$$

For the fitting procedure we generate a trace with 10^4 elements from the MMPP and fit the parameters of a MAP and a MMPP, both with 6 states. The resulting MAP is characterized by the following matrices.

$$\mathbf{D_0}^{(MAP)} = \begin{pmatrix} -2.13 & 0.00 & 0.00 & 0.12 & 0.00 & 0.08 \\ 0.00 & -0.10 & 0.00 & 0.10 & 0.00 & 0.00 \\ 0.20 & 0.00 & -2.89 & 0.00 & 0.00 & 0.00 \\ 0.02 & 0.18 & 0.00 & -1.17 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -4.99 & 0.13 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.09 & -3.64 \end{pmatrix} \text{ and}$$

$$\mathbf{D_1}^{(MAP)} = \begin{pmatrix} 1.86 & 0.00 & 0.00 & 0.07 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.03 & 0.00 & 0.00 & 0.00 & 2.66 \\ 0.00 & 0.00 & 0.00 & 0.88 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 4.86 & 0.00 \\ 0.00 & 0.00 & 3.55 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

Although the resulting MAP differs from the original MMPP, one can notice that the resulting structure similar. For the MMPP, the following matrices are computed.

$$\mathbf{D_0}^{(MMPP)} = \begin{pmatrix} -1.19 & 0.08 & 0.07 & 0.00 & 0.00 & 0.10 \\ 0.12 & -2.51 & 0.00 & 0.03 & 0.00 & 0.00 \\ 0.20 & 0.00 & -1.23 & 0.00 & 0.88 & 0.13 \\ 0.00 & 0.10 & 0.00 & -4.29 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.07 & 0.00 & -0.79 & 0.72 \\ 0.14 & 0.00 & 0.26 & 0.00 & 0.33 & -0.73 \end{pmatrix} \text{ and}$$

$$\mathbf{D_1}^{(MMPP)} = \begin{pmatrix} 1.04 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 2.36 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.00 \\ 0.09 & 0.00 & 0.00 & 4.28 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

	orig. MMPP	Trace	MAP 6	MMPP 6
$E[T^1]$	1.105e+0	1.049e+0	1.049e+0	1.049e+0
$E[T^2]$	1.620e+1	1.449e+1	1.471e+1	1.467e+1
$E[T^3]$	5.646e+2	4.884e+2	5.133e+2	5.086e+2
ρ_1	0.07595	0.08706	0.07807	0.07755
ρ_2	0.06884	0.05952	0.06802	0.06154
$\ln(d(\mathcal{T}, \mathcal{M})$	-4.839e-3		-4.824e+3	-4.841e+3

Table 4. Moments and d-value for the 6 state MMPP.

The difference between the fitted MMPP and the original MMPP is in some sense larger than the difference between the fitted MAP and the original MMPP because the resulting MMPP has 3 states with arrival rates and 3 states with very small arrival rates. Table 4 contains the measures for the original process, the trace and the fitted processes. Again one can see that the fitted MAP has a smaller distance to the trace than the original MMPP, whereas the distance of the fitted MMPP is slightly larger. However, the difference between the moments and first autocorrelations of the two fitted processes and the trace is smaller than the difference between these measures of the trace and the original process which indicates that the fitting procedure reaches a reasonable accuracy. Figure 4 shows the densities and the autocorrelations for the MMPP fitting.

4.3 Fitting of Measured Sequences

For fitting real data we use the LBL-TCP-2 trace from the internet archive [1]. From this trace we consider the first 10^6 interarrival times. For the fitting of a MAP to this sequence we consider a subset of 10^3 consecutive interarrival times. To find an appropriate subsequence for fitting, the trace is divided into 10^3 subsequences of length 10^3. For all subsequences the first two moments and the autocorrelation of lag 1 and 2 are computed. Afterwards the subsequence

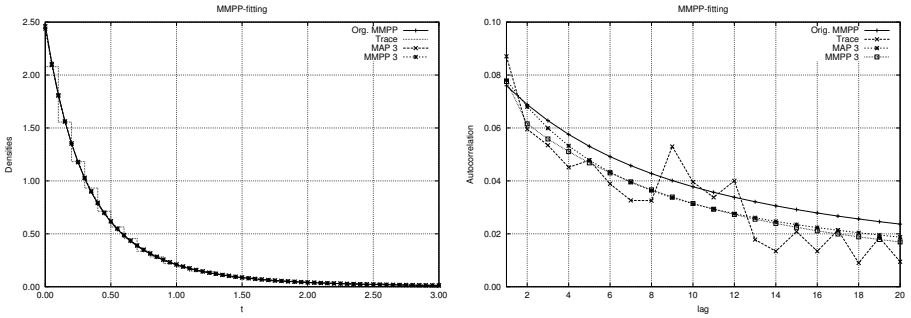

Fig. 4. Densities and autocorrelations of the 6 state MMPP and of the fitted processes.

which is nearest to the complete trace according to those measures is chosen for fitting.

	LBL-TCP-2 trace					
	Trace	Subtrace	MMPP 3	MMPP 5	MAP 3	MAP 5
$E[T^1]$	4.198e-3	4.153e-3	4.153e-3	4.153e-3	4.153e-3	4.153e-3
$E[T^2]$	5.274e-5	4.994e-5	4.863e-5	4.862e-5	4.968e-5	4.673e-5
$E[T^3]$	1.283e-6	1.044e-6	9.840e-7	9.833e-7	1.060e-6	8.541e-7
ρ_1	0.15831	0.14931	0.16066	0.15837	0.16532	0.11369
ρ_2	0.12172	0.12273	0.11835	0.11578	0.11547	0.07226
$\ln(d(\mathcal{T}, \mathcal{M})$			4.580e+3	4.580e+3	4.581e+3	4.597e+3

Table 5. Moments and d-value for the LBL-TCP-2 trace.

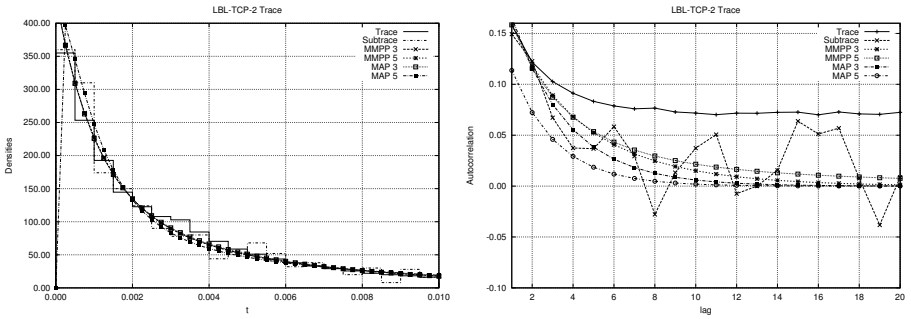

Fig. 5. Densities/autocorrelations of the LBL trace/subtrace and the fitted processes.

Four different processes are fitted to the subsequence, namely MAPs of order 3 and 5 and two MMPPs of order 3 and 5. Table 5 contains the moments, the first two autocorrelations and the difference measures for the complete trace, the subsequence used for fitting and the four fitted processes. The results for the two MMPPs and the small MAP are similar with respect to all quantities shown in table 5. The results for the MAP with 5 states are rather different from

the other processes. First of all, the d-value is larger for this MAP. However, the higher order moments and also the first two autocorrelations are less good approximations for the quantities of the subtrace. Figure 5 shows the approximated densities of the traces and the densities of the computes MAPs and MMPPs. Furthermore the figure shows the first 20 autocorrelations for the trace and the fitted processes. The densities show a significant difference between the 5 state MAP and the remaining processes. The MMPPs and the 3 state MAP have all densities which are decreasing, whereas the density of the 5 state MAP starts at 0 and reach a value of 396.7 at 0.00025. This behavior is not visible in the trace if intervals of width 0.0005 are used for the approximation of the densities as done for the representation in the figure. However, if this value is reduced the value to 0.00025, then the density of the subtrace is no longer decreasing which explains the better approximation of the subtrace by the 5 state MAP, even with a less good approximation of the moments or the autocorrelations. The plot of the autocorrelations of lag 1 through 20 in figure 5 shows that the subtrace is only a good representation for the original trace if autocorrelations of lag 1 and 2 are considered, for higher order autocorrelations trace and subtrace differ significantly.

4.4 Comparison of Queueing Performance Measures

To evaluate the quality of the approximation of some process by a MAP, apart from measures directly related to the process, also performance measures of some queueing system which is fed with the process may be used. We consider the MAP and MMPP which have been presented section 4.2 as input processes for a queue with finite capacity and exponential service time distribution. Thus, a MMPP/M/1/K and a MAP/M/1/K system are analyzed.

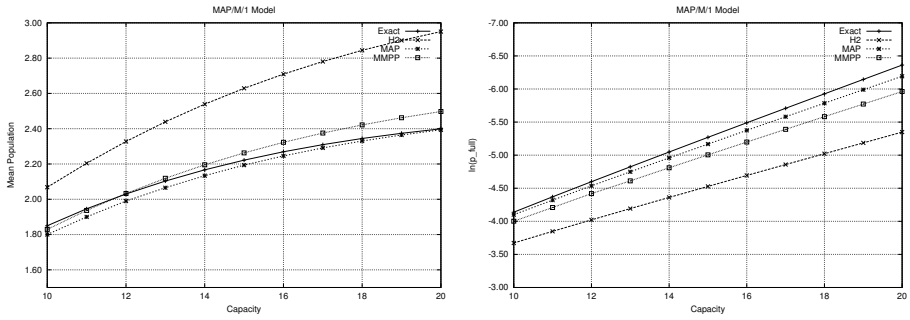

Fig. 6. Results for the MAP/M/1/K system and its approximations.

First we use the 3 state MAP as an input process for a queue with mean service time 0.5 and determine the mean population and the probability that the queue is completely filled (p_{full}) for capacities varying from 10 through 20. The exact results are compared with the results for a system where the original MAP is substituted by the fitted MAP or MMPP or by an hyperexponential distribution which is derived by fitting the first two moments. Results are shown

in figure 6. If the hyperexponentail distribution is used as input process, then the mean population and p_{full} are overestimated. The MAP and MMPP approximation are much more accurate, p_{full} is slightly overestimated, whereas the true value for the buffer population lies between both approximations. The approximation by the MAP is slightly better than the MMPP-approximation.

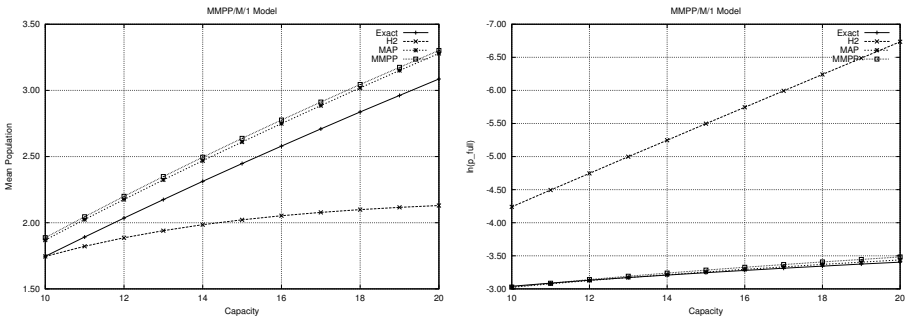

Fig. 7. Results for the MMPP/M/1/K system and its approximations.

The results for the approximation of the 6 state MMPP are shown in figure 7. They are similar to the results of the previous example.

5 Conclusions

In this paper we present an algorithm of the expectation-maximization type to fit the parameters of Markovian arrival processes according to some measured data. The approach is an extension of known approaches from the area of hidden Markov chains and uses the randomization technique to transform the continuous time Markov arrival process into a discrete time process. The presented algorithm is applied to several example traces and it is shown that the resulting Markovian arrival processes capture the trace behavior very well. The presented approach is very flexible because it allows one to generate Markovian arrival processes of arbitrary structures including Markov modulated Poisson processes or phase type distributions, if the correlation is not relevant. The limiting aspect of the algorithm, like most other algorithms of this type, is the relatively high effort. As shown one iteration of the algorithm requires an effort which is proportional to the effort of the randomization method applied to a Markov chain of the size of the fitted process over the time intervals of the trace. The algorithm can be applied to fit traces with a few thousand entries, but not with a million entries. However, to capture heavy tailed distributions or long range dependencies, very long traces have to be used. Consequently, the major aspect for future research is to increase the efficiency of the approach. This can be done by using better initial guesses of the process to reduce the number of iterations or by representing relevant information of a long trace in a more condensed form.

References

1. The internet traffic archive. http://ita.ee.lbl.gov/index.html.

2. S. Asmussen, O. Nerman, and M. Olsson. Fitting phase type distributions via the EM algorithm. *Scand. J. Statist.*, 23:419–441, 1996.
3. A. Feldmann and W. Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 31:245–258, 1998.
4. W. Fischer and K. Meier-Hellstern. The Markov-modulated Poisson process (MMPP) cookbook. *Performance Evaluation*, 18:149–171, 1992.
5. B. L. Fox and P. W. Glynn. Computing Poisson probabilities. *Communications of the ACM*, 31(4):440–445, 1986.
6. A. Horvath and M. Telek. Markovian modeling of real data traffic: Heuristic phase type and MAP fitting of heavy tailed and fractal like samples. In M. C. Calzarossa and S. Tucci, editors, *Performance 2002*, volume 2459 of *LNCS*, pages 405–434. Springer, 2002.
7. V. B. Iversen and F. Nielsen. Some properties of Coxian distributions with applications. In N. Abu el Ata, editor, *Modeling Techniques and Tools for Performance Analysis*, pages 61–66. Elsevier, 1986.
8. R. El Abdouni Khayari, R. Sadre, and B. Haverkort. Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation*, 52:175–191, 2003.
9. A. Klemm, C. Lindemann, and M. Lohmann. Modeling IP traffic using the batch Markovian arrival process. *Performance Evaluation*, to appear, 2003.
10. A. Lang and J. L. Arthur. Parameter approximation for phase-type distributions. In S. R. Chakravarty and A. S. Alfa, editors, *Matrix-analytic methods in stochastic models*, Lecture Notes in Pure and Applied Mathematics, pages 151–206. Marcel Dekker, 1996.
11. W. E. Leland, M. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions in Networking*, 2:1–15, 1994.
12. M. Neuts. *Algorithmic Probability: A Collection of Problems.* Chapman & Hall, 1995.
13. V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3:226–244, 1995.
14. A. Riska, V. Diev, and E. Smirni. An EM-based technique for approximating long-tailed data sets with PH distributions. *Performance Evaluation*, to appear, 2003.
15. A. Riska, M. S. Squillante, S. Z. Yu, Z. Liu, and L. Zhang. Matrix-analytic analysis of a MAP/PH/1 queue fitted to web server data. In G. Latouche and P. Taylor, editors, *Matrix-Analytic Methods: Theory and Applications*, pages 335–356. World Scientific, 2002.
16. T. Ryden. Parameter estimation for Markov modulated Poisson processes. *Stochastic Models*, 10(4):795–829, 1994.
17. K. Salamatian and S. Vaton. Hidden Markov modelling for network communication channels. In *Proc. ACM Sigmetrics*, 2001.
18. W. J. Stewart. *Introduction to the numerical solution of Markov chains.* Princeton University Press, 1994.
19. W. Wei, B. Wang, and D. Towsley. Continuous-time hidden Markov models for network performance evaluation. *Performance Evaluation*, 49(1-4):129–146, 2002.
20. M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modelling of the temporal dependence in packet loss. In *Proc. IEEE Infocom.* IEEE CS-Press, 1999.
21. T. Yoshihara, S. Kasahara, and Y. Takashi. Pratical time-scale fitting of self-similar traffic with markov-modulated poisson process. *Telecommunication Systems*, 17:185–211, 2001.