Correlated Phase-type distributed Random Numbers as Input Models for Simulations

Jan Kriege, Peter Buchholz
Department of Computer Science, TU Dortmund, D-44221 Dortmund, Germany
{jan.kriege,peter.buchholz}@tu-dortmund.de

Author prepared version of a paper published in

Performance Evaluation

Volume 68, Issue 11, November 2011, Pages 1247-1260

Special Issue: Performance 2011

http://dx.doi.org/10.1016/j.peva.2011.07.014

Correlated Phase-type distributed Random Numbers as Input Models for Simulations

Jan Kriege and Peter Buchholz

{jan.kriege,peter.buchholz}@tu-dortmund.de

Department of Computer Science, TU Dortmund, D-44221 Dortmund, Germany

Abstract

The adequate modeling of correlated input processes is an important step in building simulation models. Modeling independent identically distributed data is well established in simulation whereas the integration of correlation is still a challenge. In this paper, ARTA processes which have been used several times for describing correlated input processes in simulation are extended by using ARMA instead of AR processes to realize the correlation and Acyclic Phase Type distributions to model the marginal distribution. For this new process type a fitting algorithm is presented. By means of some real network traces it is shown that the extended model allows a better fitting of the marginal distribution as well as the correlation structure and results in a compact process description that can be used in simulation models.

Keywords: Simulation, Input Modeling, Stochastic Processes

1 Introduction

Many real life processes show a significant correlation between the occurrence of events. Examples for correlated events are the traffic in computer networks [30], access times and locations at a disk in a computer [32] and also failure times of software [20]. Usually these systems are analyzed using simulation models which have to capture the real behavior closely to produce reliable results. This implies that correlations are modeled using stochastic processes rather than distributions. Most simulation environments or tools for input modeling assume only *independent identically distributed* (iid) input data and do not support the modeling of correlations [23]. However, neglecting correlation in models like queueing networks often results in huge approximation errors as shown several times for example in [14] and [25]. Thus, an adequate consideration of autocorrelations in simulation models is necessary but still a challenge.

Although, in principle, the modeling of correlated processes is established [8], the use of time series models as arrival or service processes in stochastic simulation models is complicated and an active research field. There are only very few approaches that are applicable to model dependent processes from trace data. Among these approaches the *autoregressive to anything* (ARTA) model and its corresponding fitting method [7, 12] are very prominent and have been used to model various dependent processes [6]. In this paper, we slightly extend the ARTA approach in two directions. First, we consider

Autoregressive Moving Average (ARMA) processes instead of *Autoregressive Processes* (AR) to model the autocorrelation. Second, we show that the class of *Acyclic Phase Type* (APH) distributions can be integrated as marginal distributions in ARTA like processes, allowing an efficient generation of the process specification and a very good fitting of the marginal distribution. Our primary goal is to find a mathematical model that can be used to capture real processes sufficiently accurately and can be used to generate realizations of these processes in a simulation efficiently. The major application are traffic processes in computer networks which are often characterized by a strong autocorrelation over many lags such that long traces are necessary for a reliable estimation of measures related to the processes. Typical network traces include more than a million entries and fitting methods have to deal with this number of events. Our examples which use real network traces show that APH distributions allow a much better approximation of the marginal distribution than other commonly used distributions which can be measured in terms of the likelihood value or the difference in the fitted moments and still have the necessary flexibility to be integrated into ARTA models.

The outline of the paper is as follows: In the next section we give a brief overview of related work and introduce the basic definitions. Afterwards, in Section 3 we compare different models for the marginal distribution of three different network traces. Then the model of correlated APH distributions using ARMA processes is introduced. In Section 5 we show how the processes model the three example traces. The paper ends with the conclusions where we also present a brief outline of future extensions.

2 Related Work

We begin with a very brief overview of input modeling for simulation before we introduce the ingredients of our approach, namely phase type distributions, ARMA and ARTA processes.

2.1 Input Modeling for Simulation

We consider the modeling of stochastic processes Y_t according to some trace observed in a real system and use the notation X for an iid random variable. The use of iid random variables to model data is well established in stochastic simulation, methods for estimating the parameters of different distributions according to trace data are presented in textbooks on simulation (e.g., [23]). The quality of the fitting is often measured in terms of the likelihood, i.e., if $f_X(x)$ is the probability density of X and $T = (t_1, t_2, \ldots, t_l)$ are the values of the trace which are in our case interdeparture times of packets in a computer network, then the likelihood function is given by $L(T, X) = \prod_{i=1}^{l} f_X(t_i)$. For numerical reasons often the log-likelihood $l(T) = \log L(T, X)$ rather than the likelihood is used. Of course, a larger value of the (log-) likelihood indicates a better fitting. Other measures to compare the fitting of different distributions are the comparison of moments, statistical tests and QQ plots [23].

Less well established in simulation input modeling is the fitting of stochastic processes to capture apart from the marginal distribution also the autocorrelation. Let Y_t ($t = 1, 2, \ldots$) be a stationary time series with lag k autocorrelation $\rho_k = Corr(Y_t, Y_{t+k})$ ($k = 1, \ldots, K$). There are two basic approaches to model such processes [23, 13]. First, one can exploit specific properties of the marginal distribution F_Y which has been done for various marginal distributions like Normal, Lognormal, Johnson distributions but has to be specifically developed for every distribution. The second approach first constructs a process U_t with uniformly $[0, 1]$ distributed marginals and uses the transformation $Y_t = F_Y^{-1}(U_t)$ which has to assure that the generated Y_t possess the required autocorrelation structures. TES and ARTA processes belong to the second class of models [23]. We consider here an extension of ARTA processes which are very general and will be introduced in Section 2.4. Of course, $Corr(Y_t, Y_{t+k})$ does not characterize the complete process but it is usually seen as a measure that adequately describes dependencies and can be captured by input models.

2.2 Phase-type distributions and Markovian Arrival Processes

Phase Type (PH) distributions describe iid random variables as absorption times of a finite Markov chain [28, 29]. A PH distribution is defined by an $n \times n$ matrix D_0 and an initial distribution π. Matrix D_0 is the generator of an absorbing continuous time Markov chain. Events are generated whenever an absorption occurs and the process starts afterwards immediately as defined by π. The moments and probability density of PH distributions are given as

$$\mu_i = E(X^i) = i!\pi M^i e^T \text{ and } f_X(t) = \pi e^{D_0 t} e^T \quad (1)$$

where $M = -(D_0)^{-1}$ and e is a row vector of ones of length n. Some subtypes of PH distributions like hyperexponential or Erlang distributions are commonly used in simulation, their parameters are usually fitted according to low order moments. PH distributions are very flexible since every distribution with a strictly positive density in $(0, \infty)$ can be represented by a PH distribution [2] and they can be used in Markov models which apart from simulation allow also an analysis using numerical techniques.

Several approaches exist to fit the parameters of a PH distribution according to available trace data like EM algorithms to maximize the likelihood [1]. However, it turns out that parameter fitting for arbitrary PH distributions is hard since no canonical representation is known for this class and the matrix representation is redundant. For *Acyclic Phase Type* (APH) distributions which are characterized by an upper triangular matrix D_0 parameter fitting becomes easier and can be done by moment fitting as for in example in [11] or likelihood maximization via specific EM algorithms as in [33]. Examples show that the restriction to APH distributions does not seriously restrict the modeling power (see e.g. [33]).

PH distributions can be extended to Markovian Arrival Processes (MAPs) [28] for modeling correlated arrival streams. However, the parameter fitting for MAPs to capture a larger number lag k correlations is hard, although some progress has been made recently [15, 21]. We do not consider parameter fitting for MAPs but present for comparison some results in Section 5.

Although PH distributions and MAPs are mainly applied for analytical modeling, they can be used in simulation models as well. Event generation requires then the simulation of the Markov chain and the generation of an event whenever a transition triggering an event occurs. This can be done efficiently as long as the fraction of transitions triggering events is not too small. For an APH distribution at most $n + 1$ random numbers are necessary to generate an event as shown below.

2.3 Autoregressive Moving Average (ARMA) Processes

Autoregressive Processes (AR) and *Moving Average Processes* (MA) are well established in time series modeling (see [8]). An AR process of order p ($AR(p)$) is given by

$$Z_t = \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + ... + \alpha_p Z_{t-p} + \epsilon_t \quad (2)$$

and MA processes of order q ($MA(q)$) are defined as

$$Z_t = \beta_1 \epsilon_{t-1} + \beta_2 \epsilon_{t-2} + ... + \beta_q \epsilon_{t-q} + \epsilon_t \quad (3)$$

with the values ϵ_t that are denoted as *innovations* and are normally distributed with mean zero and variance σ_ϵ^2. A combination of autoregressive and moving average processes results in $ARMA(p, q)$ models defined as

$$Z_t = \alpha_1 Z_{t-1} + \alpha_2 Z_{t-2} + ... + \alpha_p Z_{t-p} + \beta_1 \epsilon_{t-1} + \beta_2 \epsilon_{t-2} + ... + \beta_q \epsilon_{t-q} + \epsilon_t \quad (4)$$

ARMA models are the most flexible of the three process types, methods for fitting the parameters according to given lag k ($k = 1, \ldots, K$) autocorrelation coefficients are available and are implemented in standard statistical software tools [8]. However, the marginal distribution of the models is always given as a weighted sum of $N(0, \sigma_\epsilon^2)$ random variables which implies that only marginal distributions that can be derived from a normal distribution by simple transformations, like the $\log$-normal distribution, can be adequately captured by ARMA processes.

2.4 Autoregressive to Anything (ARTA) Processes

ARTA processes [6, 12] combine an $AR(p)$ base process with an arbitrary marginal distribution F_Y and are defined as a sequence $Y_t = F_Y^{-1}[\Phi(Z_t)]$ ($t = 1, 2, ...$) where F_Y is the required marginal distribution, Φ is the standard normal cumulative distribution function and $\{Z_t; t = 1, 2, ...\}$ is a stationary Gaussian $AR(p)$ process as described in Eq. 2 that is constructed such that the distribution of the $\{Z_t\}$ is $N(0, 1)$ (cf. [12]). Then the probability-integral transformation $U_t = \Phi(Z_t)$ ensures that $U(t)$ is uniformly distributed on $(0, 1)$ (cf. [18]) and the application of $Y_t = F_Y^{-1}[U_t]$ yields a time series $\{Y_t, t = 1, 2, ...\}$ with the desired marginal distribution F_Y. [12, 13] established a relation between the autocorrelation structures of the $ARTA$ process and the base process, which are related by $Corr(Y_t, Y_{t+h}) = Corr\left(F_Y^{-1}(\Phi(Z_t)), F_Y^{-1}(\Phi(Z_{t+h}))\right)$, and gave an efficient numerical procedure to construct an $AR(p)$ base process such that the $ARTA$ process has the desired autocorrelations that are e.g. estimated from a trace.

For fitting of ARTA models two approaches exist that are described in [13] and [6]. The first approach only determines the $AR(p)$ base process for a given marginal distribution and a trace, while the second fits a Johnson distribution and the base process. In both approaches marginal distribution and autocorrelation are fitted separately. This is also the approach we use. Of course, this approach implicitly assumes that the process is stationary (i.e., the marginal distribution does not change over time and the base process is stationary).

3 Motivation

Since ARTA processes rely on the inversion of the cumulative distribution function, they are suitable for distributions for which a closed-form expression for the inverse cdf exists. They may also be applied for distribution where F_Y^{-1} can be approximated numerically, although this can be cumbersome if many random numbers have to be generated from the ARTA process since for every random number the value of F_Y^{-1} has to be computed numerically. For the general class of PH distributions the ARTA approach is not feasible since in general the inverse cdf cannot be computed efficiently. But for APH distributions, which can be characterized by a mixture of finite sequences of exponential distributions as shown below, we can find a different way to combine the marginal distribution with a base process that does not rely on the inversion of the cdf. In particular, the class of Hyper-Erlang distributions is very promising since it allows an easy integration in the ARTA approach as shown below and is flexible enough to capture adequately many marginal distributions occurring in real processes including network traces.

To clarify the benefits of using APH distributions we fitted APH distributions, in particular Hyper-Erlang distributions with r states ($HErD(r)$), with an increasing number of states and several other distributions, for which the ARTA approach has been used, to three different network traces. Two of the traces are from the Internet Traffic Archive[1]. The trace *BC-pAug89* [24] contains a million packet arrivals observed at the Bellcore Morristown Research and Engineering facility in August 1989. The trace *LBL-TCP-3* [30] contains two hours of TCP traffic from the Lawrence Berkeley Laboratory and was recorded in January 1994. The third trace *TUDo* [22] contains interarrival times of one million packets that have been measured from the Squid proxy server at the Computer Science Department of TU Dortmund in 2006. For our experiments all traces have been scaled to have a mean value of 1. The distributions considered for fitting are the exponential and lognormal distributions using the maximum likelihood estimators that are available in textbooks [23], the Johnson system of distributions using quantile estimation from [35], the Weibull distribution using a general purpose optimization algorithm for the maximization of the likelihood function and, as already mentioned, Hyper-Erlang distributions using the EM approach from [33]. Table 1 shows the likelihood values of the fitted distributions according to the three traces, which are computed as described in Section 2 neglecting the correlation. In addition we list the moments of the fitted distributions in conjunction with the relative errors $(|\mu_i - \hat{\mu}_i|/\hat{\mu}_i) \cdot 100$ in percent with μ_i and $\hat{\mu}_i$ being the ith moment of the distribution and the trace, respectively. The results in Table 1 clearly indicate that APH distributions usually yield better approximations in terms of both likelihood and moments than the other considered distributions, especially the likelihood increases significantly when increasing the number of phases. Fitting times for APH distributions using the tool ProFiDo [3] are slightly larger than parameter fitting for standard distributions like for example exponential, Weibull, lognormal but require usually at most a few minutes which is acceptable. This is a strong argument to use APH distributions in ARTA processes for a better fitting of the marginal distribution.

Another limitation of ARTA processes results from the use of $AR(p)$ base processes, that may result in large model descriptions if a large number of autocorrelation coefficients has to be matched which is often the case in application areas like computer networks with strong dependencies over several lags. Usually, an $AR(p)$ process can match the first p autocorrelation coefficients exactly but falls short of fitting higher lags, while $ARMA(p, q)$ processes can provide a close approximation

[1] http://ita.ee.lbl.gov

	Distribution	Log-Likelihood	Moment 1	Moment 2	Moment 3
BC-pAug89	Exponential	-999999	1.0 (0.0%)	2.0 (52.7%)	6.0 (90.7%)
	Johnson SU	-959863	0.89 (11.0%)	1.8 (57.4%)	8.1 (87.5%)
	Weibull	-990007	0.95 (5.0%)	2.1 (50.3%)	7.3 (88.7%)
	Lognormal	-953799	1.05 (5.0%)	4.1 (2.9%)	56.9 (12.1%)
	HErD(2)	-911558	1.0 (0.0%)	4.5 (5.9%)	66.3 (2.4%)
	HErD(3)	-911135	1.0 (0.0%)	3.5 (16.9%)	34.4 (46.9%)
	HErD(4)	-874270	1.0 (0.0%)	4.1 (2.9%)	51.2 (20.9%)
	HErD(5)	-847551	1.0 (0.0%)	4.1 (2.9%)	50.8 (21.5%)
LBL3-TCP	Exponential	$-1.78999e+06$	1.0 (0.0%)	2.0 (32.0%)	6.0 (64.4%)
	Johnson SB	$-\infty$	0.96 (4.0%)	2.3 (21.8%)	8.6 (48.9%)
	Weibull	$-1.72494e+06$	0.95 (5.0%)	2.4 (18.4%)	9.8 (41.8%)
	Lognormal	$-1.69839e+06$	1.14 (14.0%)	8.1 (175.3%)	345.6 (1952%)
	HErD(2)	$-1.69793e+06$	1.0 (0.0%)	2.8 (4.8%)	14.4 (14.5%)
	HErD(3)	$-1.69581e+06$	1.0 (0.0%)	2.9 (1.4%)	16.1 (4.4%)
	HErD(4)	$-1.69509e+06$	1.0 (0.0%)	2.9 (1.4%)	15.5 (8.0%)
	HErD(5)	$-1.67248e+06$	1.0 (0.0%)	2.9 (1.4%)	14.9 (11.5%)
TUDo	Exponential	$-\infty$	1.0 (0.0%)	2.0 (96.9%)	6.0 (100%)
	Johnson SL	398952	0.64 (36.0%)	49.0 (24.2%)	426400.6 (3206%)
	Weibull	455191	0.48 (52.0%)	2.0 (96.9%)	24.6 (99.8%)
	Lognormal	603905	0.55 (45.0%)	21.9 (66.1%)	64009.4 (396%)
	HErD(2)	384320	1.0 (0.0%)	12.0 (81.4%)	236.8 (98.1%)
	HErD(3)	571616	1.0 (0.0%)	24.0 (62.9%)	1022.5 (92.1%)
	HErD(4)	595607	1.0 (0.0%)	31.1 (51.9%)	1818.7 (85.9%)
	HErD(5)	609339	1.0 (0.0%)	48.0 (25.8%)	4938.6 (61.7%)

Table 1: Likelihood and moments for the fitted distributions

for a larger number of lags (often much larger than $p+q$) and still keep the model size small. These observations motivate an extension of the ARTA approach that allows for a combination of APH distributions with an $ARMA(p,q)$ base process which will be presented in the following section.

4 Correlated Acyclic Phase-type Processes

In the following we will present an approach that combines an APH distribution of order n and an autoregressive moving average process $ARMA(p,q)$ into one process to describe correlated phase-type distributed data. These novel processes will be denoted $CAPP(n,p,q)$ (*Correlated Acyclic Phase-type Processes*). First, we describe how the APH and the ARMA process are combined into one model. The APH and the base ARMA process can be fitted separately in two steps. For fitting

APH distributions various approaches exist that can be used and we refer to the literature mentioned in Section 2.2. For our examples we used the moment fitting technique of [11] and the EM algorithm of [33] which are both implemented in the freely available tool ProFiDo [3].

The construction of the ARMA process will be described in Section 4.1. Section 4.2 deals with a special class of Correlated Acyclic Phase-type Processes that restricts the APH distribution to Hyper-Erlang distribution and results in simpler notations. In Section 4.3 we will outline some requirements for the APH distributions and show the limits of this approach. The complete algorithm for constructing CAPPs is given in Section 4.4.

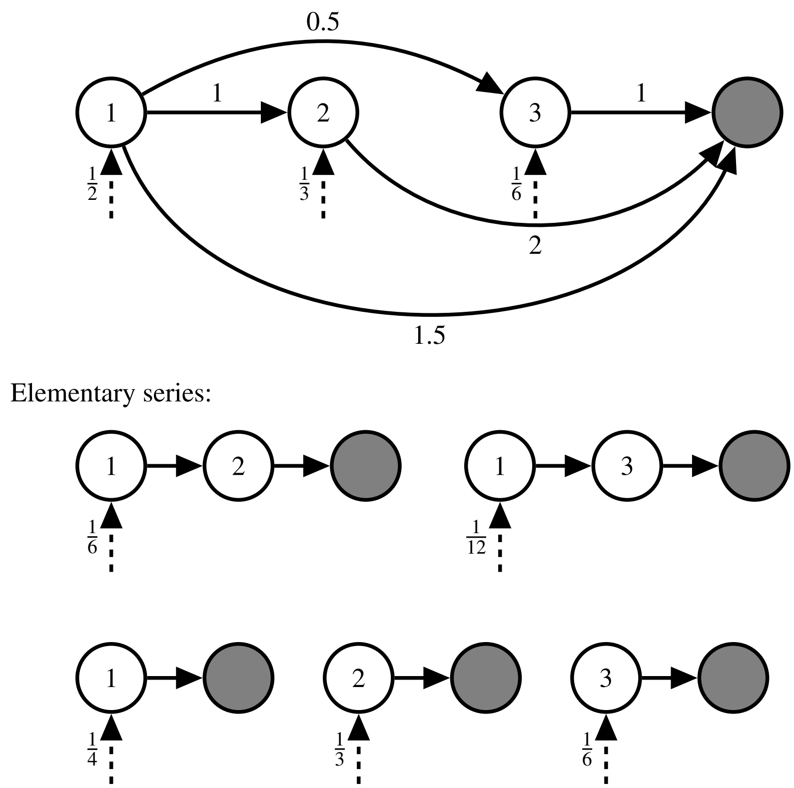


Figure 1: An APH and its elementary series

For the combination of APH distributions and ARMA processes we need a different representation for the APH distributions than the usual matrix notation. An APH distribution can be represented as a set of elementary series [16]. Each series describes one path from an initial state to the absorbing state of the APH and has a probability that is computed from the the transition rates along the path and the initial probability of the first state of the path. Let $i_1, i_2, \ldots, i_k$ be the states of an elementary series. Then the initial probability of this series is given by

$$\pi_{i_1} \frac{D_0(i_1, i_2)}{-D_0(i_1, i_1)} \frac{D_0(i_2, i_3)}{-D_0(i_2, i_2)} \cdots \frac{t(i_k)}{-D_0(i_k, i_k)}$$

where $t(i_k)$ denotes the transition rate from state i_k to the absorbing state.

Furthermore, each path describes an Hypo-Exponential distribution. An example of an APH and its elementary series is shown in Fig. 1.

Let τ_i be the probability of the i-th path ($i = 1, ..., m$). Define

$$\begin{aligned} \underline{b}_1 &= 0 \\ \bar{b}_i &= \underline{b}_i + \tau_i \qquad i = 1, ..., m \\ \underline{b}_i &= \bar{b}_{i-1} \qquad i = 2, ..., m \end{aligned}$$

and

$$\delta(U, i) = \begin{cases} 1, & U \in [\underline{b}_i, \bar{b}_i) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Let $\{X_t^{(\Lambda_i)}\}$ be sequences of iid random variables with Hypo-Exponential distribution described by a vector of rates Λ_i with length S_i that contains the transition rates of the i-th series.

Assume, that $\{U_t\}$ is a sequence of uniform $(0, 1)$ distributed random numbers. If the $\{U_t\}$ are independent the process

$$Y_t = \sum_{i=1}^{m} \delta(U_t, i) X_t^{(\Lambda_i)} \tag{6}$$

uses the elementary series to describe a sequence of iid random variables with the same acyclic Phase-type distribution that the elementary series have been computed from. In a CAPP, the ARMA process is used to determine an elementary series which generates the value of the next realization such that the marginal probability of the ith series is τ_i but the probabilities are correlated to introduce correlation in the generated random variables.

Now let $\{Z_t\}$ be an $ARMA(p, q)$ process as defined in Eq. 4 and assume that σ_ϵ^2 is set such that the $\{Z_t\}$ have a standard normal distribution. The construction of this process will be explained in the following section. Setting $U_t = \Phi(Z_t)$, where Φ is again the standard normal cdf, ensures that the $\{U_t\}$ still have uniform distribution on $(0, 1)$ (cf. [18]). Then, since the $\{Z_t\}$ are correlated, the use of U_t generated from Z_t in Eq. 6 will generate correlated random numbers with the same APH distribution as before.

To make use of this correlation a relation between $Corr[Y_t, Y_{t+h}]$ (i.e., the correlation of the generated values) and $Corr[Z_t, Z_{t+h}]$ (i.e., the correlation of the values from the ARMA process) has to be established. We are looking for a process $\{Z_t\}$ with autocorrelations $Corr[Z_t, Z_{t+h}]$ such that $\{Y_t\}$ has the desired autocorrelations $Corr[Y_t, Y_{t+h}]$.

The autocorrelations of $\{Y_t\}$ can be expressed as

$$Corr[Y_t, Y_{t+h}] = \frac{E[Y_t Y_{t+h}] - E[Y]^2}{Var[Y]} \tag{7}$$

Note, that $E[Y]$ and $Var[Y]$ are known, they can be computed using Eq. 1. Hence, we can restrict ourselves to $E[Y_t Y_{t+h}]$.

$$\begin{aligned} E[Y_t Y_{t+h}] &= E\left[\left(\sum_{i=1}^{m} \delta(U_t, i) X_t^{(\Lambda_i)}\right)\left(\sum_{j=1}^{m} \delta(U_{t+h}, j) X_{t+h}^{(\Lambda_j)}\right)\right] \qquad (8) \\ &= E\left[\sum_{i,j} \delta(U_t, i) X_t^{(\Lambda_i)} \delta(U_{t+h}, j) X_{t+h}^{(\Lambda_j)}\right], i, j = 1, ..., m \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i,j} E\left[\delta(U_t,i)X_t^{(\Lambda_i)}\delta(U_{t+h},j)X_{t+h}^{(\Lambda_j)}\right] \\
&= \sum_{i,j} \left(E\left[\delta(U_t,i)\delta(U_{t+h},j)\right]E[X_t^{(\Lambda_i)}]E[X_{t+h}^{(\Lambda_j)}]\right) \\
&= \sum_{i,j} \left(\left(\sum_{s=1}^{S_i}\frac{1}{\Lambda_i(s)}\right)\left(\sum_{s=1}^{S_j}\frac{1}{\Lambda_j(s)}\right)E\left[\delta(U_t,i)\delta(U_{t+h},j)\right]\right) \\
&= \sum_{i,j} \left(\left(\sum_{s=1}^{S_i}\frac{1}{\Lambda_i(s)}\right)\left(\sum_{s=1}^{S_j}\frac{1}{\Lambda_j(s)}\right)E\left[\delta(\Phi(Z_t),i)\delta(\Phi(Z_{t+h}),j)\right]\right) \\
&= \sum_{i,j} \left(\left(\sum_{s=1}^{S_i}\frac{1}{\Lambda_i(s)}\right)\left(\sum_{s=1}^{S_j}\frac{1}{\Lambda_j(s)}\right)\right. \\
&\qquad \left.\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\delta(\Phi(z_t),i)\delta(\Phi(z_{t+h}),j)\varphi_{\rho_h}(z_t,z_{t+h})dz_t dz_{t+h}\right)
\end{aligned}
$$

where $\varphi_{\rho_h}(z_t, z_{t+h})$ is the bivariate standard normal density function with correlation $\rho_h = Corr[Z_t, Z_{t+h}]$.

Using our knowledge of the function $\delta(\cdot)$ Eq. 8 can be simplified. Note from Eq. 5 that $\delta(u, i)$ is 1 for $u \in [\underline{b}_i, \bar{b}_i)$ and 0 otherwise. We can use this information to determine the integration bounds in Eq. 8 and omit $\delta(\cdot)$ in the double integral:

$$
E[Y_tY_{t+h}] = \sum_{i,j}\left(\left(\sum_{s=1}^{S_i}\frac{1}{\Lambda_i(s)}\right)\left(\sum_{s=1}^{S_j}\frac{1}{\Lambda_j(s)}\right)\int_{\Phi^{-1}(\underline{b}_j)}^{\Phi^{-1}(\bar{b}_j)}\int_{\Phi^{-1}(\underline{b}_i)}^{\Phi^{-1}(\bar{b}_i)}\varphi_{\rho_h}(z_t,z_{t+h})dz_t dz_{t+h}\right) \tag{9}
$$

Thus, to determine $E[Y_tY_{t+h}]$ for each combination of elementary series the product of the mean value of the first series, the mean value of the second series and the bivariate normal integral, where the integration bounds are given by the probabilities of the two series, have to be computed. For the computation of the bivariate normal integral fast numerical procedures exist [19]. The complete approach is outlined in Fig. 2.

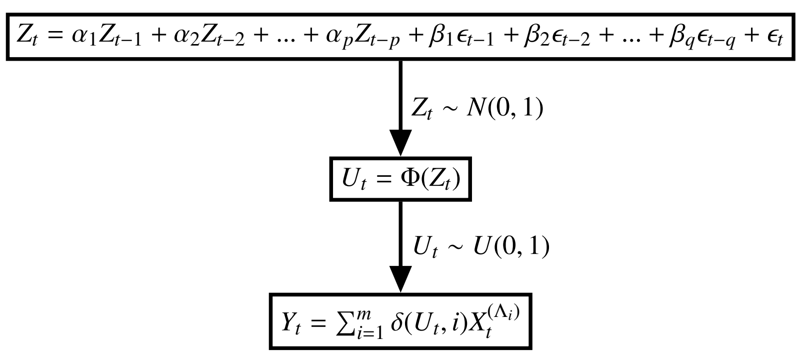


Figure 2: Steps for Constructing a Correlated Acyclic Phase-type Processes

4.1 Constructing the ARMA Base Process

The $ARMA(p,q)$ base process has to observe two requirements. First, it has to exhibit an autocorrelation structure such that the $CAPP(n,p,q)$ approximates a given set of lag-k autocorrelation coefficients that have been estimated from a trace.

Second, in the previous section we required the $ARMA(p,q)$ base process from Eq. 4 to have a standard normal distribution ($Z_t \sim N(0,1)$), i.e. the variance (or autocovariance at lag 0) of the $ARMA(p,q)$ process has to be 1. Hence, the construction of the ARMA base process consists of three parts. We have to determine the autocorrelation structure of the base process depending on the desired autocorrelation structure of the CAPP model, we have to fit an ARMA model according to that autocorrelation structure and we have to modify the ARMA model to have a standard normal distribution.

The autocorrelation structure can be determined numerically up to an arbitrary accuracy using Eq. 9. Let $\hat{\rho_h}$ be the lag h autocorrelation the resulting CAPP process should have. Then we are looking for a base process autocorrelation ρ_h such that the CAPP has correlation $\hat{\rho_h}$ according to Eq. 9. Observe that Eq. 9 defines how to compute $\hat{\rho_h}$ from a given ρ_h, but not the other way round. Since in our case $\hat{\rho_h}$ is given (i.e. estimated from the trace) and we have to determine the corresponding ρ_h, this has to be done numerically by a simple line search algorithm [31]. The simple approach can be applied since the transformation is monotonic in the autocorrelations of the base process (see Section 4.3). It should be observed that also in the basic ARTA approach a numerical technique has to be applied to compute the lag k autocorrelation coefficients of the base process [7, 13].

If the base process autocorrelation structure $\rho = (\rho_1, \rho_2, ..., \rho_K)$ has been determined we need to construct an $ARMA(p,q)$ that exhibits this structure. This can be done by using a general purpose optimization algorithm to minimize

$$min \sum_{k=1}^{K} \left(\frac{\rho_k^*}{\rho_k} - 1 \right)^2 \tag{10}$$

where the ρ_k^* are the autocorrelations of the $ARMA(p,q)$ model that is constructed during the minimization process and the ρ_k are the autocorrelations to be achieved. One has to find a compromise between a moderate size of the parameters p and q to keep the model simple and an adequate fitting of the lag k ($k = 1, \ldots, K$) autocorrelations.

The last step in the construction of the base process consists in adjusting the variance such that the process has a standard normal marginal distribution. We use the following representation from [9] for the autocovariance that is appropriate for actually computing the autocovariances:

$$\gamma(k) - \alpha_1\gamma(k-1) - ... - \alpha_p\gamma(k-p) = \sigma_\epsilon^2 \sum_{k \le j \le q} \beta_j \psi_{j-k}, \qquad 0 \le k < max(p, q+1) \tag{11}$$

and

$$\gamma(k) - \alpha_1\gamma(k-1) - ... - \alpha_p\gamma(k-p) = 0, \qquad k \ge max(p, q+1) \tag{12}$$

Defining $\beta_0 = 1$, $\beta_j = 0, j > q$ and $\alpha_j = 0, j > p$ the ψ_i can be computed from

$$\psi_j - \sum_{0<k\le j} \alpha_k \psi_{j-k} = \beta_j, \qquad 0 \le j < max(p, q+1)$$

and

$$\psi_j - \sum_{0<k\le p} \alpha_k \psi_{j-k} = 0, \qquad j \ge max(p, q+1)$$

The autocorrelation of a stationary $ARMA(p,q)$ process is given by $\rho_k = \gamma(k)/\gamma(0)$, the ρ_k are independent of σ_ϵ^2 and we can set σ_ϵ^2 such that the $\{Z_t; t = 1, 2, ...\}$ have a standard normal distribution without modifying the autocorrelation structure of the base process. Thus, for a given $ARMA(p, q)$ process with $\tilde{\sigma}_\epsilon^2$ being the variance of the white noise and $\tilde{\gamma}(0)$ being the autocovariance at lag 0 we can solve Eqs. 11 and 12 and then set the new variance to

$$\sigma_\epsilon^2 = \tilde{\sigma}_\epsilon^2 / \tilde{\gamma}(0) \tag{13}$$

resulting in a new $N(0, 1)$ process with the same autocorrelations as the old process.

4.2 Correlated Hyper-Erlang Processes

Using Hyper-Erlang distributions instead of the more general acyclic phase-type distributions has several advantages. For this class of distributions an efficient EM algorithm exists for fitting [33]. Furthermore, the ideas presented in the previous sections can be simplified for Hyper-Erlang distributions. The elementary series of the Hyper-Erlang distribution correspond to the branches of the distribution. Additionally, each series has an Erlang distribution instead of the Hypo-Exponential distribution which simplifies Eq. 9. Let S_i be the number of phases and λ_i the rate of the i-th Erlang branch. Then Eq. 9 becomes

$$E[Y_t Y_{t+h}] = \sum_{i,j} \left(\frac{S_i}{\lambda_i} \frac{S_j}{\lambda_j} \int_{\Phi^{-1}(\underline{b}_j)}^{\Phi^{-1}(\bar{b}_j)} \int_{\Phi^{-1}(\underline{b}_i)}^{\Phi^{-1}(\bar{b}_i)} \varphi_{\rho_h}(z_t, z_{t+h}) dz_t dz_{t+h} \right) \tag{14}$$

We will denote this special class $CHEP(n, p, q)$ (Correlated Hyper-Erlang Process).

4.3 Requirements and Limits of the Approach

The possible range of the autocorrelation coefficients for a CAPP depends on the APH distribution that is used. To create a correlated sequence of phase-type distributed random numbers at least two elementary series with different expected durations are necessary. For Hyper-Erlang distributions this requirement is equivalent with the distribution to have at least two distinct branches. This is not a significant restriction, because an equivalence transformation can be applied to increase the number of elementary series of an APH distribution [16, 21]. For distributions like exponential and Erlang that do not fulfill the requirement, the ARTA approach is applicable because their cumulative distribution function can be inverted.

The correlated series of uniformly distributed random numbers generated by the base process is only used for the determination of the elementary series, the exponentially distributed phases of each series are independent of this choice. In general a CAPP cannot achieve the full range from $[-1, 1]$ for the autocorrelation. The minimal and maximal possible autocorrelation of the CAPP can be computed from Eq. 9 for a base process autocorrelation of -1 or 1, respectively. In cases where this minimal or maximal autocorrelation is not sufficient to fit the estimated autocorrelation from the trace, a transformation of the APH distribution can be applied to add additional elementary series such that the range of achievable autocorrelation lags is increased. Note, that this is a rather theoretical issue. For all the experiments conducted in Section 5 no transformation was necessary at all.

We assume that the elementary series are ordered according to their mean values. Let X_i and X_j be two random variables that have a Hypo-Exponential distribution corresponding to the i-th and j-th elementary series of the APH distribution. Then $i \leq j \Rightarrow E[X_i] \leq E[X_j]$. This is, of course, not a restriction because the order of the series has no effect on the distribution but is a necessary requirement to ensure that the autocorrelation of the CAPP is nondecreasing in the autocorrelation of the base process. Observe, from Eqs. 7 and 9 that for a given APH distribution the autocorrelation of the CAPP is a function only of the base process autocorrelation ρ which appears in $\varphi_\rho(\cdot)$ in Eq. 9. We will denote this function as $\omega(\rho) = Corr[Y_t, Y_{t+h}]$ for the base process autocorrelation ρ. Then it is easy to verify that we have $\rho_1 \leq \rho_2 \Rightarrow \omega(\rho_1) \leq \omega(\rho_2)$. For positive correlations this follows directly from [34, Theorem 5.3.10], which states that for two normal variables Z_1 and Z_2 with correlation ρ $Corr[g(Z_1), g(Z_2)]$ is nondecreasing in ρ for all functions $g(\cdot)$. For

$$Y_t = g(Z_t) = \sum_{i=1}^{m} \delta(\Phi(Z_t), i) X_t^{(\Lambda_i)} \quad \text{and} \quad Y_{t+h} = g(Z_{t+h}) = \sum_{i=1}^{m} \delta(\Phi(Z_{t+h}), i) X_{t+h}^{(\Lambda_i)}$$

we obtain that $\omega(\rho_h) = Corr[Y_t, Y_{t+h}]$ is nondecreasing for $0 \leq \rho_h \leq 1$. For negative correlations a similar result has been obtained in [12, Theorem 1]. Note, that for the proof in [12] to work, $E[g(Z_t)]$ is required to be nondecreasing in Z_t, which is actually ensured in our case by sorting the elementary series of the distribution according to their mean values.

4.4 An Algorithmic Approach

In the following we will combine the ideas from the previous paragraphs into one algorithm for fitting $CAPP(n, p, q)$ processes. The algorithm is outlined in the listing below. We assume that the APH distribution has already been fitted according to a trace by one of the available approaches. The other inputs are the lag k ($k = 1, \ldots, K$) autocorrelations $\hat{\rho}_k$ that should be used for the fitting and the order (p, q) of the ARMA base process.

1. Input:
   APH or HErD $APH(n)$
   Lag k autocorrelation of the trace $(\hat{\rho}_1, \ldots, \hat{\rho}_K)$ computed as in Eq. 15
   p, q: number of AR and MA coefficients
2. determine and sort elementary series of $APH(n)$
3. determine $ARMA$ autocorrelations $\rho = (\rho_1, \rho_2, ..., \rho_K)$ using Eq. 9 or Eq. 14
4. minimize Eq. 10 to find an $ARMA(p, q)$ model for ρ
5. set variance of $ARMA(p, q)$ according to Eq. 13
6. return $CAPP(n, p, q)$ model with base $ARMA(p, q)$ process and distribution $APH(n)$

In the first step of the algorithm the elementary series of the APH distribution are computed (cf. Fig. 1) and sorted according to their mean values. As already mentioned, for Hyper-Erlang distributions the series are directly given by the branches of the distribution. The autocorrelation of lag k is estimated from the trace by

$$\hat{\rho_k} = \frac{1}{(l-k-1)\sigma^2} \sum_{j=1}^{l-k} (t_j - \mu_1)(t_{j+k} - \mu_1) \tag{15}$$

where μ_i is given by

$$\mu_i = \frac{1}{l} \sum_{j=1}^{l} (t_j)^i.$$

$\hat{\rho}$ is the desired autocorrelation that the $CAPP$ model should have.

In the third step the base process autocorrelations are determined that correspond to the desired autocorrelations for the CAPP that have been estimated from the trace as described in Section 4.1 with a simple search algorithm using Eq. 9 in the general case or Eq. 14 for Hyper-Erlang distributions.

In the next step an $ARMA(p, q)$ model is fitted to the autocorrelations determined in the previous step according to Eq. 10. For our approach we used the downhill simplex method of [27] to solve the minimization problem. A ready to use implementation is for example available in [26].

In the last step the variance of the innovations is adjusted (cf. Eq. 13) such that the base process has standard normal distribution and the $CAPP(n, p, q)$ model consisting of an $APH(n)$ distribution and a base $ARMA(p, q)$ process is returned.

The selection of the model order is important for the quality of the fitted model. If p and q are too small for the number of autocorrelation lags to be matched, the model will provide a poor approximation of the autocorrelation. If p and q are too large, the model becomes complex and one might run into the problem of overfitting [9]. However, since network traces are usually very long with a million or more entries, the problem of overfitting is usually negligible. A run of the algorithm presented above only takes a few seconds and it is possible to fit an $ARMA(p, q)$ model for all combinations of $p \in [p_{min}, p_{max}]$ and $q \in [q_{min}, q_{max}]$ for given $p_{min}, p_{max}, q_{min}, q_{max}$ and select the model with the best result according to Eq. 10.

Random numbers can be easily generated from the resulting process. First, a $N(0, \sigma_\epsilon^2)$ random number has to be generated to determine the value of Z_t which determines the elementary series of the APH distribution. Then the interevent time is generated from the elementary series which means that for an elementary series of length c, c exponentially distributed random numbers are drawn.

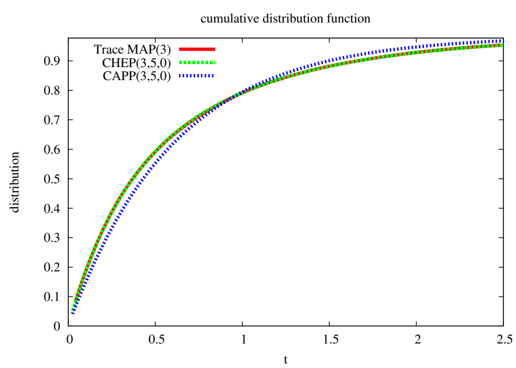
5 Experimental Results

To assess the quality of our fitting approach we used several traces that have been synthetically generated by other models and traces that have been measured in real systems and have already been introduced in Section 3. An empirical investigation of the fitting requires some form of evaluation to compare different fitted processes which is not trivial for real traces which contain correlation and a huge number of events. As in [7] we consider separately the fitting of the marginal distribution and the correlation structure. For evaluation of the marginal distribution one can compare the fitted and the empirical moments or the values of the likelihood function. Furthermore, QQ plots [23] can be applied to compare empirical and fitted distribution functions. These measures usually allow a well-founded comparison of different models. In statistics additionally different tests are used to decide whether a trace might be drawn from a distribution. However, these tests are developed for moderate sample sizes with at most a few thousand samples. If they are applied to network traces with a million or more entries they tend to reject the hypothesis that the sample is drawn from the distribution even if it has been drawn from exactly this distribution [17]. Therefore, we do not use statistical tests to evaluate the marginal distribution.

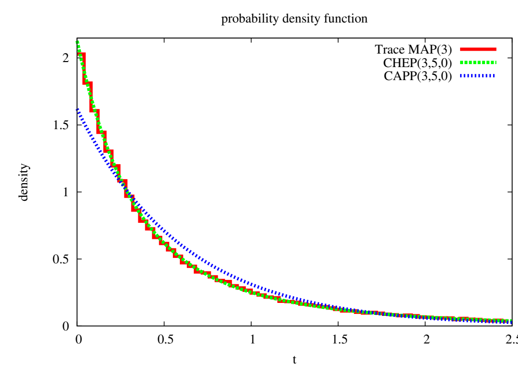
Evaluation of the correlation structure is harder. It is, of course, possible to compare the lag k autocorrelation for the trace and the fitted process. Similarly, higher order joint moments can be compared. However, these measures describe only part of the correlation. Unfortunately, other measures like the joint density or the likelihood function of the correlated trace are hard to compute for the models that are used here. In [7] the two-dimensional KS test is used to evaluate values one lag apart but the KS test in two dimensions suffers like in the one dimensional case from the huge sample size and therefore we did not use it. Instead we compare the different processes by means of a derived quantity, namely the queueing performance when used as input process for a simple *./D/1/K* queue. Thus, we use the original trace and the fitted processes as input for a queue with a capacity of 10 and deterministic service time and compare afterwards the distribution of the population in the queue for different levels of utilization.

5.1 Synthetically Generated Traces

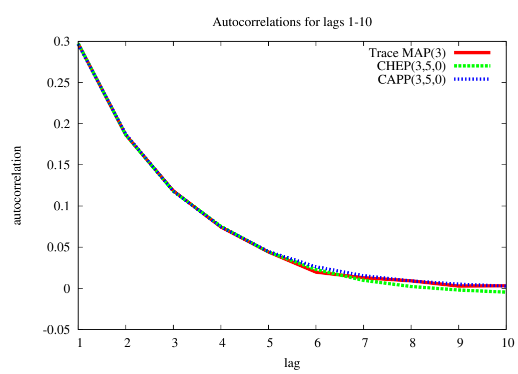
As a first example we generated a trace with 200.000 observations from a $MAP(3)$ taken from [10]. We fitted Hyper-Erlang distributions using GFIT [33] and general APH distributions using a moments matching approach from [11] and expanded them into CHEPs or CAPPs, respectively, using the algorithm from Sect. 4.4. Since the MAP that was used for



a) cumulative distribution function



b) probability density function



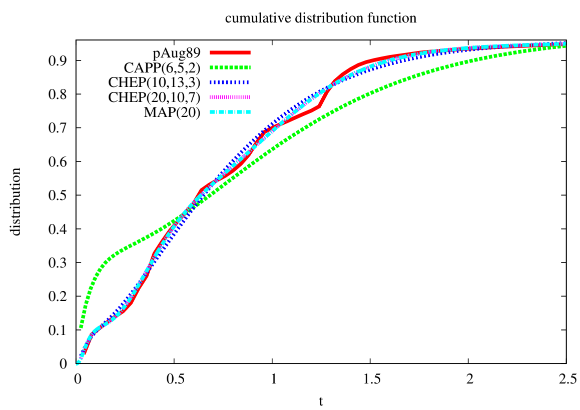
c) autocorrelations

Figure 3: Fitting results for the Trace generated by a $MAP(3)$

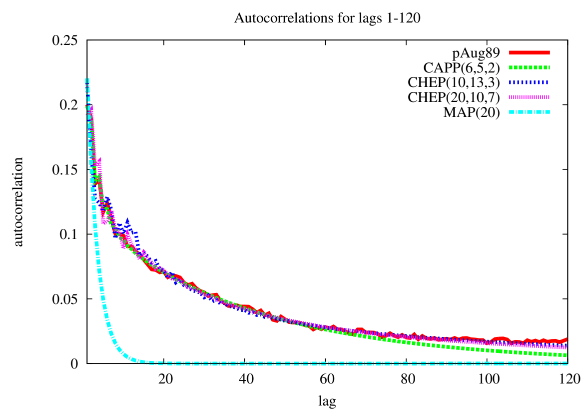
trace generation exhibits autocorrelations only for lower lags, we use an $AR(p)$ base process, that allows an exact matching of the first p lags. Fitting result are shown in Fig. 3. As one can see GFIT that uses an EM algorithm provided a better approximation of the distribution than the moments matching approach, but in both cases an exact fitting of the first 5 lags was possible.

5.2 Real Traces

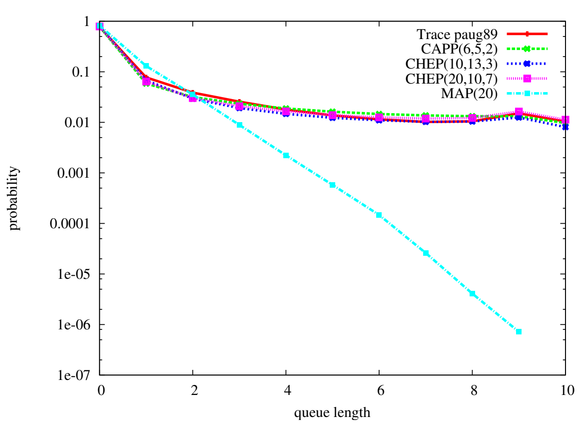
As real traces we used two common benchmark traces from the Internet Traffic Archive, namely *BC-pAug89* and *LBL-TCP-3*, and the trace *TUDo*. Again, we used GFIT for fitting a Hyper-Erlang distribution to construct a CHEP and moments matching to generate an APH distribution for CAPP fitting. For comparison we also fitted MAPs to the trace using the autocorrelation fitting approach from [22]. The approach takes the same APH distribution as input that is used for CHEP/CAPP fitting and expands it into a MAP by fitting a matrix D_1 according to the empirical autocorrelation coefficients of the trace.



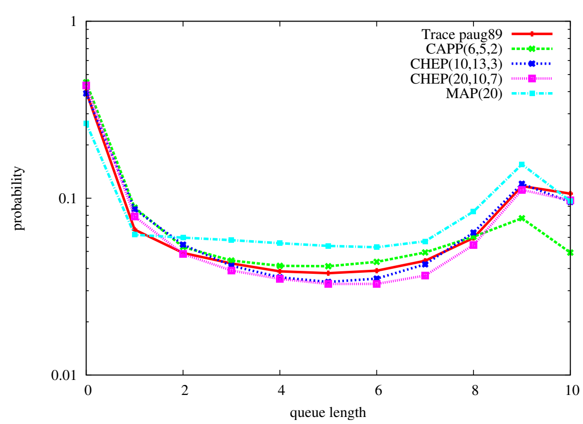
a) cumulative distribution function



b) autocorrelations



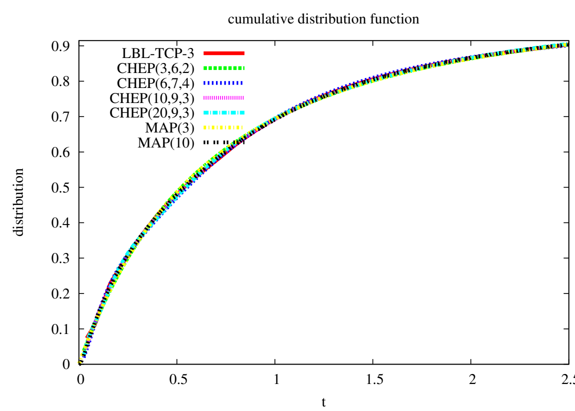
c) queue length distribution for utilization 0.5



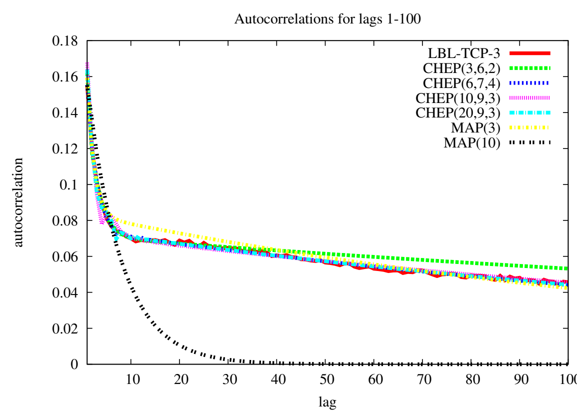
d) queue length distribution for utilization 0.9

Figure 4: Results for the Trace *BC-pAug89*

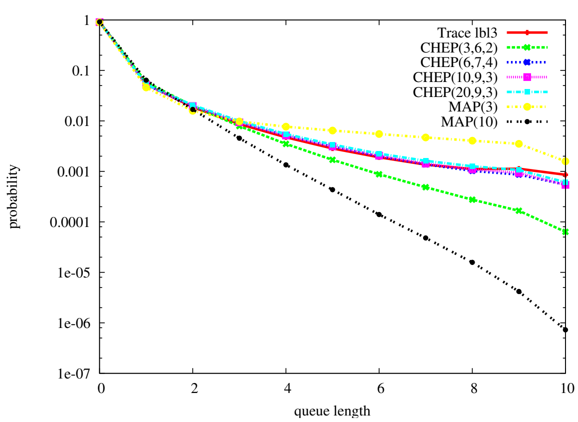
Fig. 4 shows the results for the trace *BC-pAug89*. The figure contains the curves for the cumulative distribution function, the lag-k autocorrelations and the queue length distribution for two different utilizations that have been obtained by simulating an *./D/1* queue with a buffer size of 10 for the trace and several fitted models. For the smaller $CAPP(6, 5, 2)$ we used an acyclic PH distribution that was fitted according to the empirical moments of the trace and considered the first 60 lags of autocorrelation for fitting the base process. For the $CHEP(10, 13, 3)$ and the $CHEP(20, 10, 7)$ a Hyper-Erlang distribution was fitted using GFIT with 10 and 20 states, respectively. For the construction of the base process 100 lags of autocorrelation were considered. For comparison also a $MAP(20)$ was fitted that uses the same interarrival time distribution as the $CHEP(20, 10, 7)$. As one can see from Fig. 4 a) the Hyper-Erlang distributions provided a better approximation of the distribution than the general APH that was fitted using the moments. Regarding the autocorrelation shown in Fig. 4 b) all CHEPs and CAPPs provided a good approximation. The MAP fitting algorithm was only able to fit the first lags of autocorrelation but falls short in approximating the higher lags. The better fitting of the autocorrelation structure for CHEPs and CAPPs is also reflected by the queueing results shown in Figs. 4 c) and d). As one can see the MAP underestimates the tail of the queue length for the lower utilization of 0.5, while the CHEPs and the CAPP capture the queue length distribution of the trace for both utilization levels.



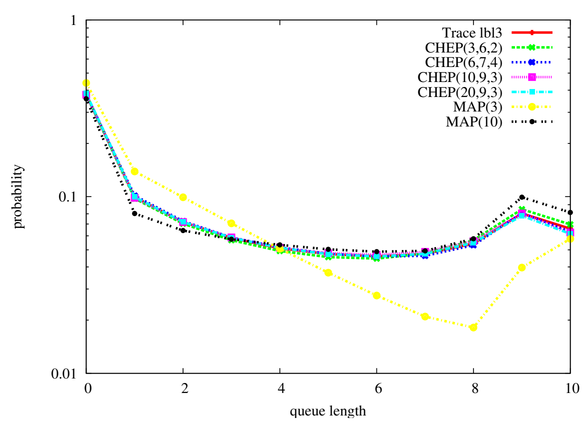
a) cumulative distribution function



b) autocorrelations



c) queue length distribution for utilization 0.3

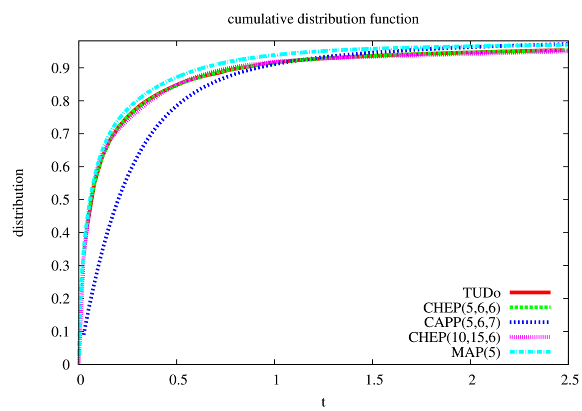


d) queue length distribution for utilization 0.9

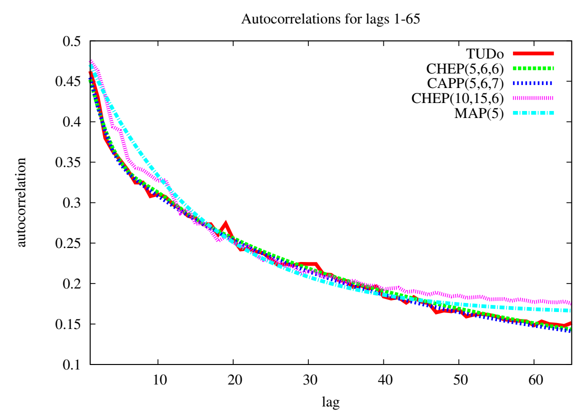
Figure 5: Results for the Trace *LBL-TCP-3*

The results for the Trace *LBL-TCP-3* are shown in Fig. 5. We fitted four different CHEPs, in particular a smaller

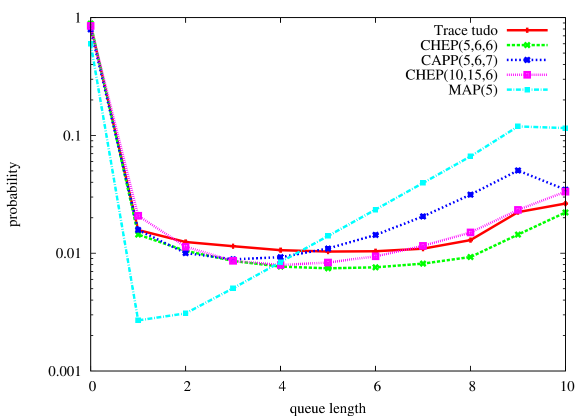
$CHEP(3,6,2)$, a $CHEP(6,7,4)$, a $CHEP(10,9,3)$ and a large $CHEP(20,9,3)$, all using a Hyper-Erlang distribution fitted with GFIT. Additionally, we used the small HErD with 3 states and larger one with 10 states to fit two MAPs. As one can see from Fig. 5 a) all HErDs were able to capture the empirical distribution of the trace. The result for the autocorrelation are shown in Fig. 5 b). For fitting the $CHEP(3,6,2)$ only the first 30 lags of autocorrelation were considered, for which the process provides a good approximation, but the larger lags that are not used for fitting are overestimated. For the other CHEPs 100 lags were fitted and consequently, these processes also captured the higher lags. The $MAP(3)$ was also able to provide a good approximation for the autocorrelation, although the lower lags are overestimated. For the larger $MAP(10)$ the fitting algorithm was only able to capture the first lags, again. As one can see from Figs. 5 c) and d) the models with the best approximation of the autocorrelation, i.e. the three larger CHEPs, also provide the best approximation of the queueing behavior. The smaller $CHEP(3,6,2)$ and the $MAP(10)$ fail to capture the tail of the queue length distribution for a small utilization. The $MAP(3)$ overestimates either the larger values of the queue length distribution for a small utilization or the smaller values for a larger utilization, respectively.



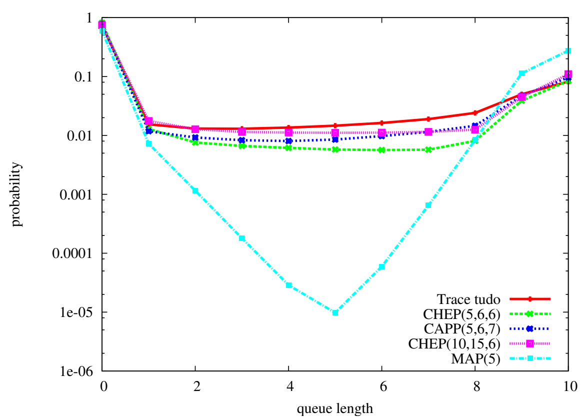
a) cumulative distribution function



b) autocorrelations



c) queue length distribution for utilization 0.3



d) queue length distribution for utilization 0.9

Figure 6: Results for the Trace *TUDo*

The last trace considered for our experimental study is the trace *TUDo*. As one can see from Fig. 6 the Hyper-Erlang distribution provided a better approximation of the empirical distribution than the APH fitted according to the moments, but in all cases a good approximation of the autocorrelation was possible, even though the trace exhibits much larger autocorrelations than the previous traces. Regarding the queue length distribution shown in Figs. 6 c) and d) the largest $CHEP(10, 15, 6)$ was able to capture the behavior of the trace best.

From the examples it is visible, that the CHEPs and CAPPs offer a larger flexibility for capturing the autocorrelations than MAP fitting algorithms. Note, that in contrast to MAP fitting the approach for constructing the base process of a CAPP is independent of the number of states of the APH distribution. Eq. 9 only depends on the number of elementary series and once the base process autocorrelation has been determined the minimization of Eq. 10 is completely independent of the APH order, but depends on the order of the base process and the number of autocorrelation lags to match. Moreover, for fitting MAPs according to the autocorrelation the given APH distribution has a large influence on the possible entries in matrix D_1 and therefore on the autocorrelation structure the MAP can exhibit. For a CAPP the distribution only determines the lower and upper bound for the autocorrelation that is possible but due to the flexibility of the ARMA base process it has little influence on the possible structure of the autocorrelation.

6 Conclusions

In this paper, we developed an approach to model correlated traffic streams by extending the established ARTA model of [12] using an ARMA instead of an AR process to model the correlation structure and *Acyclic Phase Type* (APH) distributions to describe the marginal distribution. The resulting model is denoted as *Correlated Acyclic Phase Type Process* (CAPP). An algorithm is given that allows an efficient generation of CAPPs from trace data. By fitting some real network traces it is shown that the use of APH distributions results in a significantly higher likelihood value and a better moment fitting for the marginal distribution than possible with other commonly used distributions like Weibull or lognormal. The use of ARMA instead of AR processes to describe the correlation results in a better match of a large number of lag correlations with a model of a moderate size. The resulting CAPP processes can be easily integrated in simulation models. We used the approach to fit traces from computer networks but it can be easily applied to other problems where high order correlations occur, examples are system failures or some processing times in manufacturing systems.

It is, of course, possible to extend the model. For example it is possible to consider multivariate random processes or correlated random vectors as done in the VARTA approach of [5]. Additionally, it is known that the autocorrelation is in some cases not sufficient to describe the dependencies such that additional measures have to be considered. An approach extending the VARTA approach in such a direction is given in [4], similar extensions of CAPP processes should be possible as well.

References

[1] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the EM-algorithm. *Scand. J. Stat.*, 23(4):419–441, 1996.

[2] S. Asmussen and C. A. O'Cinneide. Matrix-exponential distributions – distributions with a rational Laplace transform. In S. Kotz and C. Read, editors, *Encyclopedia of Statistical Sciences*, pages 435–440, New York, 1997. John Wiley & Sons.

[3] F. Bause, P. Buchholz, and J. Kriege. ProFiDo - The Processes Fitting Toolkit Dortmund. In *QEST*, pages 87–96. IEEE Computer Society, 2010.

[4] B. Biller. Copula-based multivariate input models for stochastic simulation. *Oper. Res.*, 57(4):878–892, 2009.

[5] B. Biller and B.L. Nelson. Modeling and generating multivariate time-series input processes using a vector autoregressive technique. *ACM Trans. Model. Comput. Simul.*, 13(3):211–237, 2003.

[6] B. Biller and B.L. Nelson. Fitting time-series input processes for simulation. *Oper. Res.*, 53(3):549–559, 2005.

[7] B. Biller and B.L. Nelson. Evaluation of the ARTAFIT method for fitting times-series input processes for simulation. *INFORMS J. on Computing*, 20(3):485–498, 2008.

[8] G.E.P. Box and G.M. Jenkins. *Time Series Analysis - forecasting and control*. Holden-Day, 1970.

[9] P.J. Brockwell and R.A. Davis. *Time Series: Theory and Methods*. Springer, 2nd edition, 1998.

[10] P. Buchholz. An EM-algorithm for MAP fitting from real traffic data. In *Computer Performance Evaluation / TOOLS*, pages 218–236, 2003.

[11] P. Buchholz and J. Kriege. A heuristic approach for fitting MAPs to moments and joint moments. In *Proc. of 6th International Conference on Quantitative Evaluation of SysTems (QEST09)*. IEEE, 2009.

[12] M.C. Cario and B.L. Nelson. Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters*, 19(2):51–58, 1996.

[13] M.C. Cario and B.L. Nelson. Numerical methods for fitting and simulating autoregressive-to-anything processes. *INFORMS J. on Computing*, 10(1):72–81, 1998.

[14] G. Casale, N. Mi, and E. Smirni. Bound analysis of closed queueing networks with workload burstiness. In Z. Liu, V. Misra, and P.J. Shenoy, editors, *SIGMETRICS*, pages 13–24. ACM, 2008.

[15] G. Casale, E.Z. Zhang, and E. Smirni. Trace data characterization and fitting for Markov modeling. *Perform. Eval.*, 67(2):61–79, 2010.

[16] A. Cumani. On the canonical representation of homogeneous Markov processes modeling failure-time distributions. *Micorelectronics and Reliability*, 22(3):583–602, 1982.

[17] J.L. Devore. *Probability and Statistics for Engineering and the Sciences*. Thomson, 7th edition, 2008.

[18] L. Devroye. *Non-Uniform Random Variate Generation*. Springer, New York, 1986.

[19] Z. Drezner and G.O. Wesolowsky. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35:101 – 107, 1990.

[20] K. Goseva-Popstojanova and K.S. Trivedi. Effects of failure correlation on software in operation. In *PRDC*, pages 69–76, 2000.

[21] G. Horváth, M. Telek, and P. Buchholz. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag-correlation. In *QEST*, pages 124–133. IEEE CS Press, 2005.

[22] J. Kriege and P. Buchholz. An empirical comparison of MAP fitting algorithms. In *Proceedings of MMB & DFT 2010*, volume 5987 of *LNCS*, pages 259–273. Springer, 2010.

[23] A.M. Law and W.D. Kelton. *Simulation modeling and analysis*. McGraw-Hill, Boston, 3rd edition, 2000.

[24] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1):1–15, 1994.

[25] M. Livny, B. Melamed, and A.K. Tsiolis. The impact of autocorrelations in queuing systems. *Management Science*, 39:322–339, 1993.

[26] J.C. Nash. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation*. Adam Hilger, 2. edition, 1990.

[27] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[28] M.F. Neuts. A versatile Markovian point process. *Journal of Applied Probability*, 16:764–779, 1979.

[29] M.F. Neuts. *Matrix-geometric solutions in stochastic models*. Johns Hopkins University Press, 1981.

[30] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions in Networking*, 3:226–244, 1995.

[31] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1993.

[32] A. Riska and E. Riedel. Long-range dependence at the disk drive level. In *QEST*, pages 41–50, 2006.

[33] A. Thümmler, P. Buchholz, and M. Telek. A novel approach for phase-type fitting with the EM algorithm. *IEEE Trans. Dependable Sec. Comput.*, 3(3):245–258, 2006.

[34] Y.L. Tong. *The multivariate normal distribution*. Springer series in statistics. Springer-Verlag, New York, 1990.

[35] R.E. Wheeler. Quantile estimators of Johnson curve parameters. *Biometrika*, 67(3), 1980.