

KPC-Toolbox: Best Recipes for Automatic Trace Fitting Using Markovian Arrival Processes

Giuliano Casale¹ Eddy Z. Zhang² Evgenia Smirni²

¹ *Department of Computing, Imperial College London, London SW7 2BZ, UK.
email: g.casale@imperial.ac.uk*

² *Department of Computer Science, College of William & Mary, Williamsburg, VA
23187-8795, USA. email: {eddy, esmirni}@cs.wm.edu*

Abstract

We present the KPC-Toolbox, a library of MATLAB scripts for fitting workload traces into Markovian Arrival Processes (MAPs) in an automatic way based on the recently proposed Kronecker Product Composition (KPC) method. We first present detailed sensitivity analysis that builds intuition on which trace descriptors are the most important for queueing performance, stressing the advantages of matching higher-order correlations of the process rather than higher-order moments of the distribution.

Given that the MAP parameterization space can be very large, we focus on first determining the order of the smallest MAP that can fit the trace well using the Bayesian Information Criterion (BIC). The KPC-Toolbox then automatically derives a MAP that captures accurately the most essential features of the trace. Extensive experimentation illustrates the effectiveness of the KPC-Toolbox in fitting traces that are well-documented in the literature as very challenging to fit, showing that the KPC-Toolbox offers a simple and powerful solution to fitting accurately trace data into MAPs. We provide a characterization of moments and correlations that can be fitted exactly by KPC, thus proving the wider applicability of the method compared to small order MAPs.

We also consider the fitting of phase-type (PH-type) distributions, which are an important specialization of MAPs that are useful for describing traces without correlations in their time series. We illustrate that the KPC methodology can be easily adapted to PH-type fitting and present experimental results on networking and disk drive traces showing that the KPC-Toolbox can also match accurately higher-order moments of the inter-arrival times in place of correlations.

Key words: Markovian Arrival Process, time series modeling, automatic fitting, temporal dependence, phase-type distribution, MAP characterization

1 Introduction

Markovian Arrival Processes (MAP) are a general class of Markov-modulated processes [19] used for fitting real workload traces with time-varying characteristics, e.g., for approximating workloads with short-range or long-range dependent behavior [1,11]. Traces of this type are commonly found in networks and computer systems, such as disk drives or e-commerce applications [18,22].

The main advantage of MAPs over other workload models is that, being based on Markov chains, they can be easily integrated within queueing systems to describe arrival or service processes. These queueing systems are then used in the computation of performance metrics such as mean response time or server utilization [18]. However, it is often prohibitive to derive MAPs that can reproduce the characteristics of real workloads with temporal dependence. The main reason for this difficulty is the large parameterization space of MAPs. Matching accurately traces with time-varying characteristics may require assigning the jumping rates between several tens of states, a task that must be supported by proper software tools.

In this paper, we propose the KPC-Toolbox, a library of MATLAB scripts for automatic fitting of real workload traces using MAPs. The KPC-Toolbox takes as input a trace of inter-arrival times, typically describing arrivals or service times, automatically searches for the best order of the MAP that can fit the trace accurately, and then derives a MAP which captures the most essential statistical features of the real workload. The underlying technology is the recently-proposed Kronecker Product Composition (KPC) fitting method for MAPs [6].

KPC reduces fitting problems to determining the characteristics of small MAPs composed by no more than two phases. These MAPs can be easily fitted with closed-form formulas and are later composed into a large MAP by Kronecker products. A similar compositional approach based on process superposition defined by Kronecker sums has been widely-used in the MAP fitting literature [1]. The novelty of KPC is that this method is able to impose moments or correlations *of any order* to the resulting MAP, while superposition methods are mostly limited to first-order and second-order statistical descriptors (e.g., mean arrival intensity, variance-time curve) that can be insufficient for accurate queueing prediction [2].

Compared to [6], the present paper further elaborates the design and the implementation of the toolbox for automatic trace fitting. This raises a number of fundamental problems including automatic MAP order selection and choice of the most important statistical descriptors to be fitted. We illustrate how these decisions can be taken based on quantitative criteria rather than on using

the intuition that often guides manual fitting approaches. We also develop characterization results for KPC processes that extend the theoretical work in [6].

A first fundamental problem in automatic fitting is determining which trace descriptors to fit in the target MAP. To give intuition on this problem, we perform a sensitivity analysis of a $MAP/M/1$ queue in order to determine the best moments and correlations to be matched by KPC. We observe that higher-order statistical descriptors that are changed by a modification of the higher-order properties of inter-arrival times such as skewness, tail of the distribution, or higher-order correlations, can result in dramatic performance changes. Instead, we find cases where performance can be quite insensitive to the coefficient-of-variation and to the autocorrelation values of the inter-arrival times. We also give evidence that it is the higher-order correlations (i.e., joint moments [24]), rather than the tail or the higher-order moments of the distribution, which drives the performance differences between these cases. Guided by the above observations, the KPC algorithm focuses on matching higher-order correlations rather than higher-order moments of the trace.

An important innovation of the KPC-Toolbox is to determine automatically the order of the MAPs used in fitting (i.e., the number of phases to be used in the underlying CTMC). Order selection in MAPs is an important issue because the MAP order can significantly affect the running times of fitting. To the best of our knowledge, no criteria have been proposed in previous work for determining the size of a MAP. The KPC-Toolbox tackles this issue with an order-selection technique based on the Bayesian Information Criterion (*BIC*) [23], that is a widely-used method for determining the best order-accuracy trade-off in ARIMA processes. We use the recursive characterization of MAP autocorrelations given in [6] as an input to the *BIC* to evaluate the best MAP order to use.

To complement our analysis on the applicability of the KPC-Toolbox, we give a theoretical characterization of processes obtained by KPC of two $MAP(2)$ s. We find KPC processes to be much less restrictive, with respect to $MAP(2)$ s, on conditions on third-order moments and to support the exact matching of autocorrelation coefficients with values up to $\frac{2}{3}$. The latter exceeds the maximum threshold of $\frac{1}{2}$ for the autocorrelation in a $MAP(2)$.

Finally, to illustrate the versatility of the KPC-Toolbox, we present experimentation on a set of processes with different temporal characteristics and illustrate that the tool provides a robust solution to fitting traces into MAPs. To further illustrate the generality of the methodology, we also present results on fitting processes that do not exhibit any temporal dependence, i.e., phase-type (PH-type) distribution [20]. We show that the KPC-Toolbox can be used for fitting several moments into a PH-type renewal process.

The paper is organized as follows. Background on MAPs and KPC is given in Section 2. Section 3 presents a sensitivity analysis on the *MAP/M/1* queue that guides KPC to decide which moments and autocorrelations are the most important and should be matched in fitting studies. Section 4 introduces the KPC-Toolbox and the new *BIC*-based order selection method. We illustrate the effectiveness of the tool using the case studies in Section 5. Section 6 discusses how the KPC-Toolbox can be used for PH-type fitting of processes that do not exhibit any autocorrelation. Section 7 provides a characterization of KPC processes with the aim of illustrating applicability of the methodology. Finally, Section 8 concludes the paper and outlines future work. The KPC-Toolbox, together with related papers and technical reports, are available for download at <http://www.cs.wm.edu/MAPQN/kpctoolbox.html>.

2 Markovian Arrival Processes

This section provides a simple introduction to Markovian Arrival Processes (MAPs) and explains some fundamental difficulties of MAP fitting. We provide background on the Kronecker Product Composition (KPC) approach to defining large MAPs, while keeping the focus of the paper on the tool implementation of the KPC fitting methodology as well as on new results that aid the tool user in finding a best MAP fitting in an automatic way.

2.1 A Primer on MAPs

Markovian Arrival Processes (MAPs) [19] may be seen as a generalization of continuous-time Markov chains (CTMC) used for fitting workload traces. A MAP differs from a CTMC in that transitions between states are classified by the user either as *background* transitions, which only change the active state as in CTMCs, or as *completion* transitions, which both change the active state and are conventionally associated to an arrival event. An inter-arrival time sample ΔT_k of a measured trace is modeled in a MAP as the time elapsed between successive activation of any two completion transitions.

A MAP can be specified by the (D_0, D_1) representation [17] which provides the exponential rates of each state of the underlying CTMC together with the jumping probabilities of background and completion transitions. For instance, a two phase MAP may be specified as

$$D_0 = \begin{bmatrix} -\lambda_{1,1} & \lambda_{1,2} \\ \lambda_{2,1} & -\lambda_{2,2} \end{bmatrix}, D_1 = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} \\ \mu_{2,1} & \mu_{2,2} \end{bmatrix},$$

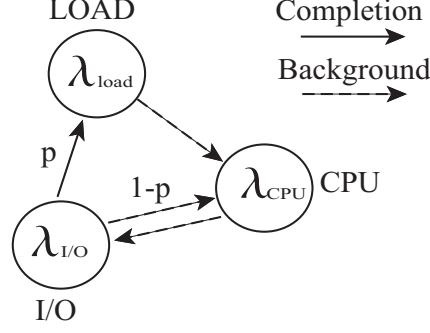


Fig. 1. Example of Markovian Arrival Process (MAP)

where the inverse of $\lambda_{1,1} = \lambda_{1,2} + \mu_{1,1} + \mu_{1,2}$ (resp. $\lambda_{2,2} = \lambda_{2,1} + \mu_{2,2} + \mu_{2,1}$) is the mean time spent in phase 1 (resp. 2) before a jump; the off-diagonal elements $\lambda_{i,j}$, $i \neq j$, are such that $\lambda_{i,j}/\lambda_{i,i}$ is the probability that the MAP will jump from the active state i to state j using a background transition. Similarly, $\mu_{i,j}/\lambda_{i,i}$ is the probability of jump from i to j following a completion transition. This also allows to specify the probability $\mu_{i,i}/\lambda_{i,i}$ of having a completion and then returning instantaneously to state i .

Example 1 *The MAP in Figure 1 gives an example of a possible job execution in a computer system: after a loading phase that lasts for a time exponentially distributed with mean λ_{load}^{-1} , the job alternates CPU-intensive phases and I/O intensive phases, each having exponentially distributed lengths with mean λ_{CPU}^{-1} and $\lambda_{I/O}^{-1}$, respectively. With probability p the MAP jumps from the I/O-intensive phase to the loading phase through the completion transition and we may associate this event with a job completion, i.e., the departure of the previous job and the arrival of a new job into service. This semantically defines the inter-arrival time ΔT_k between two jobs as the time elapsed between returns to the loading phase.*

For the model in this example, the (D_0, D_1) description would be

$$D_0 = \begin{bmatrix} -\lambda_{load} & \lambda_{load} & 0 \\ 0 & -\lambda_{CPU} & \lambda_{CPU} \\ 0 & (1-p)\lambda_{I/O} & -\lambda_{I/O} \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ p\lambda_{I/O} & 0 & 0 \end{bmatrix},$$

Example 2 *Consider a trace of inter-arrival times following an Erlang-2 distribution with mean $2\lambda^{-1}$. This can be modeled as a MAP*

$$D_0 = \begin{bmatrix} -\lambda & \lambda \\ 0 & -\lambda \end{bmatrix}, \quad D_1 = \begin{bmatrix} 0 & 0 \\ \lambda & 0 \end{bmatrix},$$

where D_1 specifies that inter-arrival times are taken when the active state changes from state 2 to state 1. Note that after each completion the MAP

restarts from state 1, which models correctly the initialization behavior of the Erlang-2 process.

Example 3 Consider a two-state hyper-exponential distribution with rates λ'_1 and λ'_2 and probability p of selecting the first phase. This can be fitted by a MAP with the following representation

$$D_0 = \begin{bmatrix} -\lambda'_1 & 0 \\ 0 & -\lambda'_2 \end{bmatrix}, D_1 = \begin{bmatrix} p\lambda'_1 & (1-p)\lambda'_1 \\ p\lambda'_2 & (1-p)\lambda'_2 \end{bmatrix}$$

This MAP specifies that each visit to a state of an hyper-exponential process is followed by a completion (no background transitions in D_0) and, in order to generate the next inter-arrival time sample, the MAP needs to be re-initialized in state 1 with probability p and in state 2 with probability $1 - p$.

In the (D_0, D_1) notation, $Q = D_0 + D_1$ is the infinitesimal generator of the CTMC that describes the evolution of the active phase over time. The discrete-time Markov chain (DTMC) with probability matrix

$$P = (-D_0)^{-1}D_1, \quad (1)$$

and equilibrium state probabilities $\vec{\pi}_e = \vec{\pi}_e P$ describes the probabilities $p_{i,j}$ of activating a completion transition that jumps to state j conditioned on the MAP being initialized in state i . Thus, phase j is the state in which the MAP restarts from in order to generate the next inter-arrival time sample. According to this property, P implicitly describes the temporal dependence between consecutive inter-arrival times by probabilistically relating the initialization conditions of successive samples. For instance, a process

$$D_0 = \begin{bmatrix} -\lambda_1 & 0 \\ 0 & -\lambda_2 \end{bmatrix}, D_1 = \begin{bmatrix} 0 & \lambda_1 \\ \lambda_2 & 0 \end{bmatrix},$$

has inter-arrival times that are taken alternatively from phase 1 and phase 2 in a cyclic manner because of the structure of D_1 . This introduces a regular temporal dependent structure that cannot be represented by models of uncorrelated (i.e., renewal) processes, such as PH-type renewal processes where the order of inter-arrival times is completely random. The support for temporal dependence is the main advantage of MAPs over other classes of workload fitting models such as PH-type distributions.

An useful property of the (D_0, D_1) representation is the simplicity of evaluating MAP feasibility [6]. This amounts to assuring that $Q = D_0 + D_1$ is a valid irreducible infinitesimal generator and that the entries of D_1 are all non-negative with at least one entry being strictly positive to imply a non-zero arrival rate. These conditions can be easily checked. Other representations that

admit a meaningful probabilistic interpretation are possible, e.g., (Q, P) , but it is hard with these descriptions to assess MAP feasibility without computing explicitly D_0 and D_1 . This is because there exist pairs (Q, P) matrices that define valid Markov chains, but upon computing the related D_0 and D_1 matrices these include one or more negative off-diagonal rates which make the MAP unfeasible. Therefore, we focus throughout the paper on the (D_0, D_1) representation. We also indicate with $\text{MAP}(N)$ a process composed by N phases, e.g., a $\text{MAP}(2)$ is a two state process.

2.2 Moment Matching

Fitting a trace requires to capture the properties of a time series in terms of distribution and correlations between samples, which jointly summarize the observed patterns. That is, the correlations capture probabilistically the relative ordering of the samples in the trace, while the distribution describes the actual values assumed by each sample irrespectively of its position in the time series.

Because of the difficulty in obtaining robust estimates of the probability density function, moment matching is largely used in fitting distributions [13, 24]. The distribution of inter-arrival times in a MAP is always PH-type, thus moments are obtained from basic theory of PH-type distributions as

$$E[X^k] = k! \vec{\pi}_e (-D_0)^{-k} \vec{e}, \quad k = 1, 2, \dots, \quad (2)$$

where $\vec{\pi}_e P = \vec{\pi}_e$. For this reason, fitting algorithms for PH-type distributions can be also applied to MAP distribution fitting [13, 21], although they need to be complemented by techniques for correlation matching in order to fit traces with temporal dependence.

A popular approach to match the temporal dependence structure is to use second-order properties of the trace [1], such as the correlations

$$E[X_j X_{j+k}] = \vec{\pi}_e (-D_0)^{-1} P^k (-D_0)^{-1} \vec{e}, \quad k = 1, 2, \dots, \quad (3)$$

where X_j and X_{j+k} are inter-arrival times spaced by $k - 1$ arrivals. The autocorrelation function is a normalization of these values, i.e.,

$$\rho_k = \frac{\vec{\pi}_e (-D_0)^{-1} P^k (-D_0)^{-1} \vec{e} - E[X]^2}{E[X^2] - E[X]^2}, \quad k = 0, 1, \dots$$

A related descriptor of particular interest for two-state MAP is the decay rate of the autocorrelation, which is fixed and equal to $\gamma = \rho_{k+1}/\rho_k$.

Note that equation (3) is more difficult to fit to real data than (2) because of

the P^k term that accounts for the temporal dependence. In addition, imposing a certain distribution by (2) reduces the degrees of freedom for manipulating the matrices in (3), because D_0 and $\bar{\pi}_e$ have been already constrained to fit a set of moments $E[X^k]$ according to (2). This builds intuition on the main issue of fitting workload traces with MAPs: we need to control the properties of moments and correlations by simultaneously manipulating products of matrices appearing in (2)-(3), which is a difficult nonlinear problem. Separate fitting of moments and correlations has been recently attempted [13] and can work successfully on small and medium sized processes, but the underlying optimization can become harder on larger MAPs. The KPC method presented in the next section provides instead a simple divide-and-conquer approach for fitting both (2)-(3) in large MAPs.

2.3 Fitting Large MAPs: Kronecker Product Composition (KPC)

This section reviews the Kronecker Product Composition (KPC) method proposed in [6]. KPC is a new technique for imposing moments and correlations of arbitrary order to a MAP. The idea is to use fundamental properties of the Kronecker (or tensor) product operator [4] to simplify the control of the matrix products and inversions appearing in (2)-(3). The aim of KPC is to provide a simple way to match a set of moments and autocorrelations by composing together small MAPs, typically MAP(2)s that can be fitted easily by closed-form formulas [10]. This approach has been deeply investigated in process superposition based on the Kronecker sum operator [1], which is effective for manipulating the counting process of a MAP, however the extension to fitting inter-arrival times has been difficult because of the different structure of equations (2)-(3). KPC overcomes this difficulty by generalizing the superposition approach to inter-arrival times based on a Kronecker product operator that is constrained to generate a feasible MAP.

Given $MAP^a = \{D_0^a, D_1^a\}$ and $MAP^b = \{D_0^b, D_1^b\}$, we define the KPC of the two processes as the process

$$MAP^a \otimes MAP^b = \{D_0, D_1\} = \{-D_0^a \otimes D_0^b, D_1^a \otimes D_1^b\}, \quad (4)$$

where $\otimes$ denotes the Kronecker product operator. If MAP^a has K_a phases and MAP^b has K_b phases, then the process $MAP^a \otimes MAP^b$ has $K_a K_b$ phases. This suggests that the KPC operator should be used parsimoniously to avoid generating models that have too many states. As an example of Kronecker product, if the original MAPs have D_0 matrices

$$D_0^a = \begin{bmatrix} -a_{1,1} & 0 \\ 0 & -a_{2,2} \end{bmatrix}, D_0^b = \begin{bmatrix} -b_{1,1} & b_{1,2} \\ b_{2,1} & -b_{2,2} \end{bmatrix},$$

where $a_{i,j} \geq 0$ and $b_{i,j} \geq 0$ are nonnegative real numbers, then KPC yields a process with $D_0 = -D_0^a \otimes D_0^b$ where

$$D_0 = \begin{bmatrix} -a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & 0 & 0 \\ a_{1,1}b_{2,1} & -a_{1,1}b_{2,2} & 0 & 0 \\ 0 & 0 & -a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ 0 & 0 & a_{2,2}b_{2,1} & -a_{2,2}b_{2,2} \end{bmatrix} \quad (5)$$

is a feasible D_0 matrix, having negative elements on the main diagonal only. The last feasibility condition is always enforced by KPC if at least one matrix between D_0^a and D_0^b is a diagonal matrix (e.g., D_0^a in the previous example), otherwise the zero entries in $-D_0^a \otimes D_0^b$ are replaced by negative values that make the MAP infeasible. Note that this condition does not place strong constraints on the generality of the KPC method since one of the two matrices can always be arbitrary. The composition $D_1^a \otimes D_1^b$ instead generates always a valid D_1 matrix structured according to the general properties of the Kronecker product. We remark that if KPC is used to compose M MAPs, $M - 1$ of these MAPs must have a diagonal D_0 matrix.

The main motivation behind (4) is that the Kronecker product satisfies the algebraic relations [4]

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

$$(A \otimes B)^{-k} = A^{-k} \otimes B^{-k}$$

which allow to decompose matrix products and inversions in terms of similar operations on smaller matrices. For instance, in a KPC process it is

$$(-D_0)^{-k} = (-D_0^a)^{-k} \otimes (-D_0^b)^{-k}$$

and from this relation it can be shown with simple passages that similarly

$$P = P^a \otimes P^b, \quad \vec{\pi}_e = \vec{\pi}_e^a \otimes \vec{\pi}_e^b,$$

where the indexes a and b refer to MAP^a and MAP^b . We point to [6] for proof of the above formulas. It is also possible to show in a similar way that moments and correlations are decomposed as well. For example, the mean of the KPC process $MAP^a \otimes MAP^b$ is immediately decomposed as

$$\begin{aligned} E[X] &= \vec{\pi}_e (-D_0)^{-1} \vec{e} = (\vec{\pi}_e^a \otimes \vec{\pi}_e^b) (-D_0^a \otimes -D_0^b)^{-1} (\vec{e}_a \otimes \vec{e}_b) \\ &= (\vec{\pi}_e^a (-D_0^a)^{-1} \vec{e}_a) (\vec{\pi}_e^b (-D_0^b)^{-1} \vec{e}_b) = E[Y]E[Z], \end{aligned} \quad (6)$$

where $\vec{e}_a$ and $\vec{e}_b$ are column vectors of all ones of the same length of $\vec{\pi}_e^a$ and $\vec{\pi}_e^b$, while Y and Z are random variables denoting the inter-arrival times of the

processes MAP^a and MAP^b , respectively. Based on passages similar to the one shown above, it is easy to derive the following decomposition formulas [6]

$$E[X^k] = \frac{E[Y^k]E[Z^k]}{k!}, \quad (7)$$

$$E[X_j X_{j+k}] = E[Y_j Y_{j+k}] E[Z_j Z_{j+k}]. \quad (8)$$

which generalize to joint moments as

$$E[X_j^u X_{j+k}^v X_{j+k+h}^z] = \frac{E[Y_j^u Y_{j+k}^v Y_{j+k+h}^z] E_b[Z_j^u Z_{j+k}^v Z_{j+k+h}^z]}{u!v!z!}. \quad (9)$$

Since joint moments are insufficient to uniquely specify a MAP [24], we conclude that (9) provides a complete characterization of the theoretical properties of the KPC operator. We also remark that from (7)-(9) it has been derived in [6] that the squared coefficient-of-variation of the KPC process, henceforth denoted SCV , is related to SCV_a and SCV_b by

$$SCV = (1 + SCV_a)(1 + SCV_b)/2 - 1 \quad (10)$$

and that the autocorrelation coefficients are

$$\rho_k = \frac{SCV_a}{SCV} \rho_k^a + \frac{SCV_b}{SCV} \rho_k^b + \frac{SCV_a SCV_b}{SCV} \rho_k^a \rho_k^b. \quad (11)$$

The KPC technique, reviewed above for the case of two MAPs, generalizes in a recursive fashion to KPC of several processes. For example, the mean of the composition $MAP^a \otimes MAP^b \otimes MAP^c$ is given by $E[X] = E[Y]E[Z]E[W]$, being W the random variable denoting the inter-arrival times of MAP^c . A similar extension holds also for all other formulas [6].

Relations (7)-(9) explain that KPC creates processes with statistical descriptors that are in simple relation with those of the original MAPs used in the composition. This allows to define arbitrarily large MAPs without losing control on their moments and correlations. These descriptors can be assigned to desired values by properly adjusting the characteristics of the MAPs composed by KPC. These MAPs have much smaller order than the KPC process, hence they can be manipulated easily; specifically, in this paper we focus on the case where all MAPs used in the composition are MAP(2)s. However, the general approach works also with other processes such as MAP(3)s [6].

Summarizing, (7)-(9) state that the MAP fitting of real traces can be essentially reduced to the following divide-and-conquer problem: *assign moments and correlations of small MAPs such that their KPC composition by (4) defines a MAP with the desired values of the moments and the correlations in (7)-(9)*. The KPC-Toolbox presented in Section 4 defines an automatic fitting scheme, based on the general KPC fitting approach of [6], to achieve this result.

2.4 Kronecker Product Composition and Process Superposition

The definition of large MAPs by composition of smaller processes has been extensively explored in the literature of Markov-modulated processes using the process superposition technique [1, 19]. In this subsection, we introduce a probabilistic interpretation of KPC that allows a comparison between KPC and superposition.

Given two processes $MAP^a = \{D_0^a, D_1^a\}$ and $MAP^b = \{D_0^b, D_1^b\}$, the superposition $MAP^a \oplus MAP^b = \{D_0^a \oplus D_0^b, D_1^a \oplus D_1^b\}$ describes the inter-arrival times between activation of completion transitions in any of the processes MAP^a and MAP^b , where $\oplus$ denotes the Kronecker (tensor) sum operator. This is particularly appealing for network traffic modeling, where completion transitions represent the arrivals of new packets over a channel and therefore the superposition $MAP^a \oplus MAP^b$ represents inter-arrival times of packets originating from two independent traffic sources MAP^a and MAP^b . Thus, the counting process specified by the number of arrivals n_t of $MAP^a \oplus MAP^b$ at a timescale t is immediately defined as the sum of the number of arrivals due to MAP^a and those due to MAP^b , i.e., $n_t = n_t^a + n_t^b$. This implies that the counting process of $MAP^a \oplus MAP^b$ has mean $E[n_t] = E[n_t^a] + E[n_t^b]$, and variance $Var[n_t] = Var[n_t^a] + Var[n_t^b]$, being $Cov[n_t^a, n_t^b] = 0$ thanks to the independence of MAP^a and MAP^b . This allows to control first-order and second-order properties of the counting process of $MAP^a \oplus MAP^b$ in a straightforward manner by simply changing the mean and variance of the counting process of the superposed MAPs, which can be done by simple analytical formulas if the superposed MAPs have two states [1]. Nevertheless, no explicit formulas exist for skewness or higher-order moments of the counting process of a MAP and this makes it difficult to fully control the properties of the superposed process.

The solution proposed by KPC to control MAP properties is to use a *hierarchical* Markov modulation in which the composition $MAP^a \otimes MAP^b$ is interpreted as MAP^a modulating the rate of transitions of MAP^b . For example, suppose that MAP^a is the hyper-exponential process of Example 3 and MAP^b is the Erlang-2 process of Example 2. Then their KPC $MAP^a \otimes MAP^b$ has representation

$$D_0 = \begin{bmatrix} -\lambda \cdot \lambda'_1 & \lambda \cdot \lambda'_1 & 0 & 0 \\ 0 & -\lambda \cdot \lambda'_1 & 0 & 0 \\ 0 & 0 & -\lambda \cdot \lambda'_2 & \lambda \cdot \lambda'_2 \\ 0 & 0 & 0 & -\lambda \cdot \lambda'_2 \end{bmatrix}, D_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ p\lambda \cdot \lambda'_1 & 0 & (1-p)\lambda \cdot \lambda'_1 & 0 \\ 0 & 0 & 0 & 0 \\ p\lambda \cdot \lambda'_2 & 0 & (1-p)\lambda \cdot \lambda'_2 & 0 \end{bmatrix}.$$

According to the active state of MAP^a , the transition rates of MAP^b are

scaled by different multipliers λ'_1 or λ'_2 . Thus, MAP^a essentially modulates the frequency of arrivals from MAP^b .

Since KPC allows to compose together an arbitrary number of MAPs, we immediately conclude that the approach described above generalizes to the definition of a hierarchy of different arrival intensities. Thus, the flexibility of KPC lies in the creation of a hierarchy of Markov modulations where one can easily control the properties of distribution and burstiness at each level of the hierarchy by simply operating on the corresponding MAP that modulates that level.

Summarizing, the superposition approach allows the definition of complex processes by summing independent MAPs and this is effective to control mean and variance of the counting process. Conversely, KPC uses a hierarchical composition of MAPs and formulas (7)-(9) describe the simple effects of this modulation on moments and correlations of inter-arrival times. In the following sections, we examine how to best use this flexibility of KPC for fitting traces with complex temporal dependence properties.

3 What is Important for MAP Fitting?

In the design of the KPC-Toolbox, we have focused our attention on a challenging, but fundamental, question: which statistical descriptors are the most important for fitting traces using MAPs? A common approach in the current literature is to match the most important moments and correlation coefficients using the simplest available model, often a MAP(2). For example, the first three moments $E[X]$, $E[X^2]$, $E[X^3]$, and the lag-1 correlation $E[X_j X_{j+1}]$, which determines the lag-1 autocorrelation ρ_1 , are sufficient to fully parameterize a MAP(2) [8, 10]. Matching these four parameters is often considered a viable approach to fit a trace; however, we argue that this frequently results in models with poor predictive capabilities. For example, Figure 2 shows the simulated queue-length probabilities for a *Trace/M/1* queue driven by the Bellcore pAug89 (BC-pAug89) trace inter-arrival times¹ [16] and compares them with the probabilities of a *MAP(2)/M/1* queue, where the MAP(2) matches exactly the first three moments and the lag-1 autocorrelation of the Bellcore trace. The results clearly show the poor modeling accuracy of the MAP(2) fitted with this approach.

The experiment in Figure 2 motivates the investigation in this section: we study numerically the sensitivity of the *MAP/M/1* queue-length distribution

¹ The Bellcore pAug89 trace is often used in the MAP literature for accuracy evaluation of fitting techniques, for example in [1, 11].

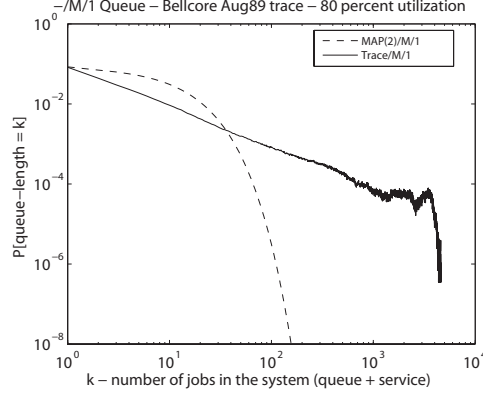


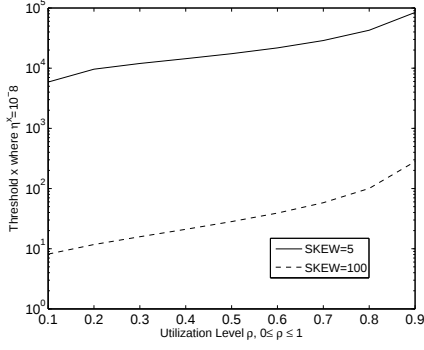
Fig. 2. Inaccurate queue-length predictions of a MAP(2) fitted by exact matching of the trace’s most important moments and correlations

to the parameters used in MAP fitting. The aim is to derive qualitative recommendations for the best moments and correlations to be matched in MAP fitting.

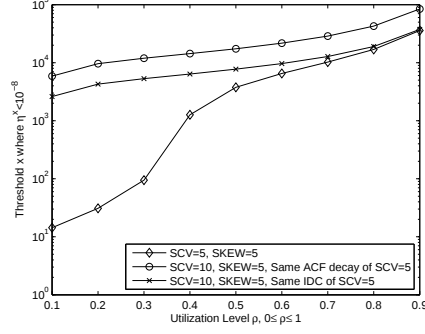
3.1 Evaluation Methodology

We perform a sensitivity analysis in two phases. We first evaluate the *MAP(2)/M/1* queue sensitivity and later confirm our observations using a larger process defined by the KPC of two MAP(2)s, henceforth called a KPC(4) process. In the first analysis on the MAP(2), queue performance is studied as a function of its first three normalized moments (mean inter-arrival time *MEAN*, squared coefficient-of-variation *SCV*, and skewness *SKEW*) and the lag-1 autocorrelation coefficient ρ_1 of inter-arrival times. A fundamental difficulty in the analysis is that a variation of a single parameter results in changes to the process, e.g., any variation of *SKEW* results in a simultaneous change of the skewness of the distribution, of the tail of the distribution, of higher-order moments, and of higher-order correlations (e.g., the bicornrelations $E[X_j X_{j+k} X_{j+k+z}]$ in (9)). This is a consequence of linear dependencies that relate moments and correlations in a MAP(*N*) [6]. Only first and second-order moments and correlations are unaffected by changes of *SKEW*. Our conjecture is that higher-order correlations, such as the bicornrelations (9), rather than the tail of the distribution or the skewness, are the main determinant of queueing performance under correlated workloads. We provide evidence of this claim in Section 3.3; before, we generically call “higher-order properties” the moments and correlations changed in a *MAP(2)* by a variation of *SKEW* for fixed *MEAN*, *SCV*, and ρ_1 .

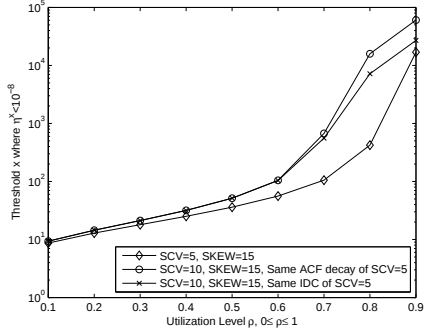
The sensitivity analysis is performed as follows. Recall that for a *MAP/M/1* queue, the queue-length probabilities decay asymptotically as $P(n = k) \sim c_0 \eta^k$, where η is a decay rate called caudal characteristic [19] and c_0 is a



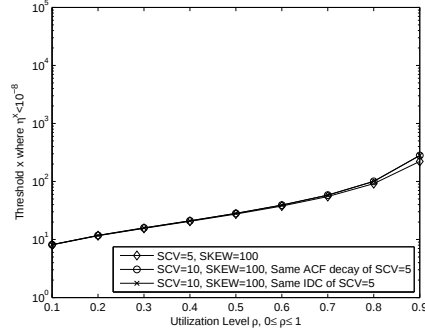
(a) Sensitivity to $SKEW$; $SCV = 10$



(b) Sensitivity to SCV ; $SKEW = 5$



(c) Sensitivity to SCV ; $SKEW = 15$



(d) Sensitivity to SCV ; $SKEW = 100$

Fig. 3. Impact of $SKEW$ and SCV on the decay rate of the $MAP/M/1$ queue-length probabilities for different utilization values

positive constant [19]. We investigate the sensitivity of the $MAP/M/1$ results by determining the queue-length value x such that $\eta^x < 10^{-8}$. Results for other values of the threshold are qualitatively similar. Intuitively, x represents a position of the queue-length distribution after which the probability values should be too small to affect performance. If the $MAP/M/1$ queue is nearly-insensitive to a MAP parameter such as a moment, we expect x to change slightly under variations of that parameter and the shape of $P(n = x)$ should not be altered in a dramatic way. Therefore, we check sensitivity to a fitting parameter by evaluating changes of the “threshold” x ; the analysis is done for different server utilization levels $\rho \in [0.1, 0.9]$ by varying the rate of the exponential service process.

3.2 $MAP(2)$ Fitting

We first investigate the sensitivity of the $MAP(2)/M/1$ queue to SCV , $SKEW$, and ρ_1 . We focus on the cases $SCV > 1$ and autocorrelations $\rho_k > 0$, which are among the most common in real traces.

Sensitivity to *SKEW* and higher-order properties. We consider a MAP(2) with mean $MEAN = 1$, $SCV = 10$, and $\rho_1 = 0.446$ which corresponds to an autocorrelation decay rate $\gamma = \rho_k/\rho_{k-1} = 0.99$. We evaluate the cases $SKEW = 5$ and $SKEW = 100$. The value $SKEW = 100$ represents a distribution with light tail, since the large asymmetry places most of the probability mass around small values; conversely, $SKEW = 5$ has a fat tail. Similar considerations hold for the higher-order correlations. It is easy to verify numerically that $SKEW = 5$ has a considerably larger temporal dependence than $SKEW = 100$.

Figure 3(a) shows the impact of $SKEW$ on the threshold x . For all utilization levels, the threshold is about 1000 times larger if $SKEW = 5$ instead of $SKEW = 100$. That is, performance degrades dramatically for $SKEW = 5$ with the tail of the queue-length probabilities becoming orders of magnitude longer than for $SKEW = 100$. It is striking to see that the impact of $SKEW$ is considerable also at utilizations as low as 0.1, where the variation of x appears maximal.

The result indicates that the higher-order properties affected by a change of $SKEW$ can have a remarkable influence on queueing predictions. We have noticed that if the same experiment is performed after setting all autocorrelations to zero, the variations of x under changes of $SKEW$ are not nearly as dramatic. This is an indication that either a long tail in the distribution or higher-order moments may be insufficient to capture alone the $MAP/M/1$ queueing performance. This is consistent with our claim that higher-order correlations are the main determinants of queueing performance.

Sensitivity to SCV . We have performed several experiments for different values of SCV , $SKEW$, and ρ_1 , and found that for large skewness (e.g., $SKEW = 100$), the queue-length probabilities are weakly sensitive to SCV , see 3(d). From now on, we focus on the more difficult cases $SKEW = 5$ and $SKEW = 15$. The 99th percentile of $SKEW = 5$ is in the range $[11.7, 16.2]$, while for $SKEW = 15$ it is in $[5.6, 9.6]$; in the previous subsection, for $SKEW = 100$ the 99th percentile is in $[4.4, 4.5]$.

The sensitivity to the SCV is evaluated by setting $SCV = 5$ or $SCV = 10$ for each choice of $SKEW$. Since in general a change of SCV implies also a change in the autocorrelations, we illustrate the effects of keeping the same autocorrelation decay rate of the model or the same index of dispersion $IDC = SCV(1 + 2 \sum_{k=1}^{\infty} \rho_k)$. The results of the experiments are shown in Figure 3(b-d). We find that for each choice of $SKEW$, the value of this parameter implies fundamental performance differences: if $SKEW = 5$, the impact of SCV is mostly at low utilization, otherwise for $SKEW = 15$ only the high utilization is affected by a change of SCV , for $SKEW = 100$ the impact of SCV is instead negligible at all utilizations. The choice of fixing the autocorrelation

decay rate or *IDC* impacts only at larger loads and suggests that the differences at heavy-load between the two figures cannot be attributed to the autocorrelations only. Explaining the different results in Figure 3(b) and Figure 3(c) is difficult and the tail of the distribution does not give any clear intuition behind these effects. Instead, a more detailed analysis of temporal dependence reveals that the *distance* between bursty arrival periods is dramatically changed by *SCV* and *SKEW*, e.g. for *SKEW* = 5 the autocorrelation in counts [7] between consecutive bursty periods of length $T = \text{MEAN}$ is $\rho_1^c = 0.91$ for *SCV* = 10 and $\rho_1^c = 0.57$ for *SCV* = 5, while for *SKEW* = 15 the two cases are similar. This suggests that the busy period of the queue may be substantially affected by the *SCV* and *SKEW* changes and this could reasonably explain the very different results in Figure 3(b)-3(c). Once again, this stresses the importance of matching correlations to fit processes that can be used in queueing modeling.

Sensitivity to ρ_1 . We have also evaluated MAP(2)s for *SCV* = 10, *SKEW* = 5, and $\rho_1 \in [0.0, 0.446]$. The results indicate the following properties: if *SKEW* = 15, then ρ_1 mostly impacts for utilization values larger than 70% with a gap of two orders of magnitude for x between the extreme cases $\rho_1 = 0.0$ ($x \approx 10^3$) and $\rho_1 = 0.446$ ($x \approx 10^5$). For utilization values smaller than 30% and *SKEW* = 15 the queue is almost insensitive to changes in ρ_1 . Conversely, for *SKEW* = 5 the threshold x varies up to two order of magnitude under changes of ρ_1 and for all utilization values. This is consistent with our previous findings that the higher-order properties are critical for a good fitting, but also stresses that for medium-high utilizations a good match of the autocorrelations is fundamental.

Summary. The experiments performed in this section indicate that it may be difficult to fit real traces by relying on first and second order properties of the trace only. The higher-order properties controlled by *SKEW* affect dramatically the performance of a *MAP(2)/M/1* queue. Also *SCV* and autocorrelations remain very important, although we have found some insensitivity for certain combinations of utilization and *SKEW* values; in light of this last observation, it also appears quite difficult to discriminate which is more important between *SCV* and ρ_1 , as their relative impact changes with the utilization and the *SKEW* values. In the next section, we complete our analysis on higher-order properties by investigating which between higher-order moments and higher-order correlations is the most important for queueing performance.

<i>Process</i>	<i>MEAN</i>	<i>SCV</i>	<i>SKEW</i>	ρ_1
MAP^a	1	19	19.82	0.468
MAP^b	1	4	85.88	0.371
MAP^c	1	19	77.51	0.468
MAP^d	1	4	210.8	0.371
$MAP^a \otimes MAP^b$	1	49	57.77	0.482
$MAP^c \otimes MAP^d$	1	49	57.77	0.482

Table 1

MAP input parameters used in the KPC experiment in Section 3.3.

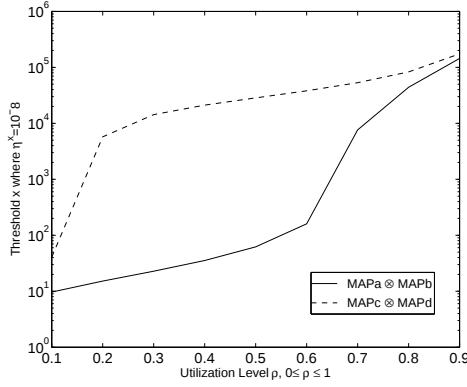


Fig. 4. Effects of higher-order temporal dependence on the $KPC(4)/M/1$ queue performance

3.3 General MAP Fitting

We use KPC to try to discriminate whether higher-order correlations are more important than higher-order moments in fitting traces with temporal dependence. We consider two $MAP(2)$ processes MAP^a and MAP^b such that the resulting KPC $MAP^a \otimes MAP^b$ has fat tail and temporal dependence, and we study the resulting threshold x as a function of the utilization ρ . Then, we repeat the experiment using two different $MAP(2)$ processes MAP^c and MAP^d such that the resulting $MAP^c \otimes MAP^d$ has same *MEAN*, *SCV*, *SKEW*, and approximately the same tail of the distribution and higher-order moments as $MAP^a \otimes MAP^b$. Table 1 gives the parameters of the MAPs used. The key difference is that $MAP^a \otimes MAP^b$ and $MAP^c \otimes MAP^d$ have considerably different higher-order correlations.

Since the two distributions are virtually identical in terms of tail decay, we would expect reasonably similar performance if the distribution would be the main determinant of the $MAP/M/1$ queue performance. However, Figure 4 shows that the variation of the threshold x in the two cases is extreme. Except for the high-utilization case, where the results must converge because

of the identical value of the index of dispersion IDC of the two MAPs, the performance of $MAP^c \otimes MAP^d$ is orders of magnitude worse than for $MAP^a \otimes MAP^b$. This provides a strong example in favor of our claim that fitting higher-order correlations is of critical importance.

Summary. Matching higher-order correlations in the KPC-Toolbox has priority over matching higher-order moments or the tail of the distribution. In practice, it may be reasonable to assume that matching the first two-three moments of the distribution of temporal dependent traces may be sufficient for capturing the impact of the distribution on queueing performance under temporal dependence. All residual degrees of freedom of the model should be spent into matching correlations of different orders. Due to the cost of computing on a real trace a large number of correlations with more than three terms, the correlations $E[X_j X_{j+k}]$ and $E[X_j X_{j+k} X_{j+k+z}]$ are often the most practical descriptors to be matched. Note also that higher-order correlations of the type $E[X_j^2 X_{j+k}]$ are included as special cases of $E[X_j X_{j+k} X_{j+k+z}]$.

4 The KPC-Toolbox

The KPC-Toolbox is a set of MATLAB scripts for fitting real traces into MAPs. The toolbox implements the algorithmic solution of the divide-and-conquer fitting problem defined by KPC based on MAP(2)s presented in [6]. Several design choices have been made to strike a good balance between simplicity of use and accuracy of fitting. Noticeably, the underlying fitting paradigm is based on the sensitivity analysis results in Section 3. In the next subsections we outline the key ideas behind the design of the KPC-Toolbox.

4.1 Order Selection

The KPC-Toolbox approach to MAP order selection is based on the recursive characterization of MAP autocorrelation coefficients presented in [6]. We consider the trace autocorrelations and use this characterization to bound from above the goodness-of-fit of a MAP(N) model for different choices of the order N . From these values, we select the target MAP order for the KPC process using an information-theoretic approach.

We begin by recalling that MAP moments and correlations satisfy simple linear recurrence expressions [6]. For example, the autocorrelations ρ_k of a MAP with N phases satisfy

$$\rho_k = a_1 \rho_{k-1} + a_2 \rho_{k-2} + \dots + a_N \rho_{k-N}, \quad k \geq N, \quad (12)$$

where the a_k coefficients are computed from the eigenvalues of $P = (-D_0)^{-1}D_1$. Therefore, a *necessary* condition for a good matching of the measured trace autocorrelations $\hat{\rho}_k$ is that they can be fitted accurately by (12) for a choice of the coefficients a_k . This is not a *sufficient* condition, unless the a_k 's are constrained in sign and magnitude to be generated from a feasible MAP(N); unfortunately, feasibility expressions for the a_k are prohibitive to obtain for large MAPs because of the high-order of the nonlinear equations involved. For this reason, our order selection approach relies only on necessary conditions.

We select the target MAP order N^* by (12) as follows. For a trace where we consider m autocorrelation coefficients $\hat{\rho}_k$, we give a preliminary evaluation of a MAP(N) fitting by defining a linear system of equations (12)

$$\begin{cases} \hat{\rho}_{k+1} = a_1\hat{\rho}_k & + a_2\hat{\rho}_{k-1} & + \dots + a_N\hat{\rho}_{k-N+1}, \\ \hat{\rho}_{k+2} = a_1\hat{\rho}_{k+1} & + a_2\hat{\rho}_k & + \dots + a_N\hat{\rho}_{k-N+2}, \\ \vdots & \vdots & \vdots \\ \hat{\rho}_{k+m} = a_1\hat{\rho}_{k+m-1} & + a_2\hat{\rho}_{k+m-2} & + \dots + a_N\hat{\rho}_{k-N+m}. \end{cases} \quad (13)$$

The linear system is solved efficiently by linear regression to compute the residual sum of squares $RSS = \sum_k (\hat{\rho}_k - \rho_k)^2$. Linear regression of hundreds of equations (12) can be solved with small time and space requirements, usually in a few seconds. In particular, the KPC-Toolbox uses $m = 500$ logarithmically-spaced autocorrelations $\hat{\rho}_k$, $k \in [1, L/10]$ being L the trace length, to parameterize the linear system (13). If the trace has low RSS , then a MAP(N) is a suitable candidate for matching the trace correlation structure.

After this first evaluation, we use the computed RSS values for different choices of the model size $N \in \{2, 4, 8, 16, 32, 64\}$ to select the best trade-off between accuracy and model size as follows. We adopt the Bayesian Information Criterion (BIC) [23] as a quantitative method to estimate the best MAP order. This is defined as

$$BIC(N) = m \log \left(\frac{RSS}{m} \right) + N \log m, \quad (14)$$

where, in our application, m is the number of autocorrelations $\hat{\rho}_k$ used in the regression. According to its definition, the BIC should be intended as a cost function, i.e., lower values of BIC denote better trade-offs. Thus, the best order is immediately selected as

$$N^* = \{N : \min_N BIC(N)\},$$

and it is used by the KPC-Toolbox as the best choice for the MAP order. Indeed, similar criteria may be defined by replacing BIC with similar cost-accuracy objective functions, such as the Akaike Information Criteria (AIC).

However, the *BIC* is known to be better than *AIC* as the number of available observations becomes asymptotically large [23].

4.2 KPC Degrees of Freedom for MAP(2)s

The KPC-Toolbox defines MAPs of large order by composition of MAP(2)s. The advantage of this approach is that each composing MAP(2) can be generated efficiently using closed-form fitting formulas and enforced to be feasible using simple analytical constraints² [10].

The composition of J MAP(2)s by KPC defines a MAP(2^J) that has at least $4J$ degrees of freedom, because the original MAP(2)s are each defined by four parameters. Starting from the results of the sensitivity analysis in Section 3, the KPC-Toolbox uses 3 degrees of freedom to match the basic moments of the trace (mean exactly, second and third-order moments approximately), while the residual $4J - 3$ degrees are used for temporal dependence fitting of autocorrelations and bicorrelations. For example, in order to fit a MAP(16) the KPC-Toolbox uses no less than 13 degrees of freedom for autocorrelations and bicorrelations.

In addition to these $4J - 3$ degrees of freedom, the KPC-Toolbox leverages on the following property of KPC that further increases fitting flexibility. The concept is introduced with an example. Consider two processes MAP^a and MAP^b , and let us impose the $4J = 8$ degrees of freedom such that: $E_a[X] = 1$, $E_b[X] = 1$, $E_a[X^2] = 20$, $E_b[X^2] = 30$, $E_a[X^3] = 2000$, $E_b[X^3] = 3000$, $\rho_1^a = \rho_1^b = 0$. Then the KPC $MAP^a \otimes MAP^b$ has fourth moment

$$E[X^4] = 4.\underline{7876} \times 10^9.$$

Let us now swap the values of the second moments between MAP^a and MAP^b , i.e., $E_a[X^2] = 30$, $E_b[X^2] = 20$, without altering the other parameters. Then, the first three moments of the KPC process are unchanged because of (7), but the fourth moment becomes

$$E[X^4] = 4.\underline{7964} \times 10^9,$$

since the fourth moments of MAP^a and MAP^b are affected by the swapping of the second moments because of the linear dependencies between moments found in [6]. If the MAPs have temporal dependence, the change also affects the correlations at all orders. This experiment tells us that a permutation of the assignment of the MAP(2) parameters, rather than of their values,

² On the other hand, restricting to MAP(2)s reduces generality. For example, fitting variable bit-rate (VBR) traffic traces requires imposing complex eigenvalues in the spectrum of P , but this cannot be done using MAP(2)s or their KPC.

provides additional flexibility to improve fitting accuracy. A combinatorial analysis reveals that the above property provides up to

$$4 \binom{J}{J-1} = 4J$$

degrees of freedom for temporal dependence fitting in addition to the $4J - 3$ left after matching the first three trace moments. The KPC-Toolbox is able to use implicitly such additional degrees of freedom thanks to the optimization approach discussed in the next section. This brings the maximum number of degrees of freedom available to the analysis to $8J$, however in practice the number may be less since certain combinations of second and third moments are not feasible for a MAP(2). We also remark that, in general, the number of degrees of freedom offered by a KPC process is much smaller than the maximum number of degrees of freedom offered by a general MAP with identical number of states. However, fitting methods are lacking for general MAPs in order to exploit such extra degrees of freedom.

4.3 KPC-Toolbox Fitting Algorithm

The pseudo-code in Figure 5 provides an algorithmic overview of the KPC-Toolbox execution. The KPC-Toolbox first performs the *BIC* order selection and determines the optimal number $J = \log_2 N^*$ of MAP(2)s to be composed by KPC. Then, the tool searches for an actual set of J MAP(2)s that can match accurately the first three moments, the autocorrelations, and the bicorrelations of the trace. This search is performed by the KPC fitting algorithm presented in [6]. The KPC-Toolbox uses by default up to 500 logarithmically-spaced autocorrelation coefficients and up to 25 bicorrelation values, the latter obtained from a grid of 5×5 logarithmically-spaced points. The range of sampling is set by default in $[1, L/10]$ for the autocorrelations and to lags in $[1, L/10] \times [1, L/10]$ for the bicorrelations, where L is the trace length. Lags where the measured autocorrelations decay below 10^{-6} are ignored. These parameters can be changed by the user if needed. Other parameters of interest that can be modified by the user are numerical tolerance of the optimization program, expected size of the final MAP thus overriding the *BIC* order selection, and maximum number of optimization programs to be carried out and their iterations.

The KPC-Toolbox fitting algorithm is organized around three stages: *SCV* and autocorrelation KPC fitting (*stage 1*); mean, skewness and bicorrelation KPC fitting (*stage 2*); generation of the final MAP (*stage 3*).

In the first stage, the KPC-Toolbox fits the measured autocorrelations $\hat{\rho}_k$ by searching the values of *SCV* and autocorrelation decay rate γ for each MAP(2)

BIC order selection

$K_1 = \text{set of } m = \min\{n, 10^4\} \text{ logarithmically spaced lags in } [1, \hat{\rho}_{min}]$

For all candidate MAP orders $N = \{2, 4, 8, 16, 32, 64\}$

$r_k = \text{set of residuals of linear regression of measured autocorrelations}$

$\{\hat{\rho}_k, \hat{\rho}_{k+1}, \dots, \hat{\rho}_{k+n}\}$ against lags $\{k, k+1, \dots, k+n\} \in K_1$

Compute $BIC(N) = m \log(\sum_k r_k^2/m) + N \log(m)$

Return as best order the smallest $N^* = \min_N BIC(N)$

Set number of MAP(2)s to compose to $J = \log_2 N^*$

For $i = 1, \dots, I_1$ times (default: $I_1 = 50$)

1. *KPC-based nonlinear least square fitting of measured autocorrelations $\hat{\rho}_k$ [6]*

Objective: $f_1^i = \min \sum_{k \in K_1} (\rho_k - \hat{\rho}_k)^2$

Unknowns: squared coefficient of variation of the j th composed

MAP(2) SCV_j , $1 \leq j \leq J$; autocorrelation decay rates γ_j , $1 \leq j \leq J$

End For

$I_1^* = \text{set of solutions } \{(SCV_j, \gamma_j), 1 \leq j \leq J\} \text{ for } q \text{ (default: } q = 10) \text{ best minimal}$
solutions f_1^i found at step 1

$K_2 = \text{grid of } p \times p \text{ (default: } p = 10) \text{ logarithmically-spaced lags in } [1, \hat{\rho}_{min}]$

For each solution $\{(SCV_j, \gamma_j), 1 \leq j \leq J\}$ in I_1^*

For $i = 1, \dots, I_2$ times (default: $I_2 = 30$)

2. *KPC-based nonlinear least square fitting of bicomrelations [6]*

Objective: $f_2^i = \min \sum_{(k_1, k_2) \in K_2} (E[X_0 X_{k_1} X_{k_2}] - E[\hat{X}_0 \hat{X}_{k_1} \hat{X}_{k_2}])^2$

Unknowns: MAP(2) means $E[X_j]$ and third moments $E[X_j^3]$, $1 \leq j \leq J$

End For

End For

$I_2^* = \text{solution } \{(E[X_j], SCV_j, E[X_j^3], \gamma_j), 1 \leq j \leq J\} \text{ for the best objective}$
function value f_2^i found at step 2

3. Generate J MAP(2)s from best solution $(E[X_j], SCV_j, E[X_j^3], \gamma_j)$, $1 \leq j \leq J$
Compose J MAP(2)s by KPC and return the resulting MAP(N^*)

Fig. 5. KPC-Toolbox Pseudocode

with the aim of minimizing the residual sum-of-squares (RSS) between the $\hat{\rho}_k$ set and the autocorrelation ρ_k of the KPC process. From (11) it follows that the autocorrelation ρ_k of the KPC process can be computed directly from the values of SCV and autocorrelation decay rate of each composing MAP(2) without the need of imposing at this stage mean and skewness for each of the MAP(2)s. This is a fundamental result because it allows to decouple fitting of second order properties from fitting of all other statistical descriptors and thus significantly reduces computational costs. The optimization program used in stage 1 also includes a constraint which imposes that the maximum error between the measured SCV and the KPC process SCV must be less than 10%. Further, it uses constraints that assure MAP feasibility based on the feasibility ranges of SCV_j and γ_j found in [9, 10].

Mean and skewness of the J MAP(2)s are determined during stage 2 of the fitting algorithm, where we match with a nonlinear least-squares approach

bicorrelations, mean and third moment of the trace. The only significant difference with respect to the previous stage is that we also impose constraints on the feasible mean and skewness values for the composing MAP(2)s. This is because, upon fixing the SCV and autocorrelation decay values found in stage 1, not all combinations of mean and skewness evaluated in stage 2 would result in feasible MAP(2)s. This issue is tackled by adding to the optimization program the feasibility constraints found in [10]. Note also that, throughout the entire stage 2, each output couple $(E[X_j], E[X_j^3])$ has a one-to-one mapping with a specific couple (SCV_j, γ_j) determined in the previous stage and thus allow to uniquely identify the underlying MAP(2) processes.

In the final stage 3, we use the values of the first three moments and autocorrelation decay rate found in the first two stages to determine the (D_0, D_1) representation of each MAP(2) using the closed-form formulas in [8, 10]. The resulting MAP(2)s are finally composed by KPC into a MAP(N^*) that is returned to the user.

Overall, the three stages described above are run several times to find the most accurate MAP fitting to be returned to the user. The selection algorithm operates as follows. The KPC-Toolbox first runs several times stage 1 while keeping stored the solutions that produced the top 10 RSS values of the stage 1 optimization. Then, stage 2 is run several times for each of the solutions selected in stage 1. The KPC-Toolbox returns as best MAP the stage 2 solution associated to the overall lowest RSS for the bicorrelations. According to this selection approach, the final MAP should have a high-quality fitting of the autocorrelations (stage 1) and a good fitting of the bicorrelations (stage 2) and it is thus consistent with the sensitivity analysis conclusions presented in Section 3.

5 Numerical Experiments

We now discuss experiments on the accuracy of the KPC-Toolbox in fitting both real and synthetic traces. Consistently with the sensitivity analysis, we focus our evaluation on the queueing prediction accuracy for a $-/M/1/FCFS$ queue and fit two traces:

- *BC-pAug89 trace*: this is a benchmark case for evaluating the quality of long-range dependent trace fitting approximations [1, 11, 13]. The traffic consists of 1 million inter-arrival time samples collected in 1989 at the Bellcore Morrison Research and Engineering facility.
- *SWeb trace*: This trace is composed by 3.6×10^6 inter-arrival times of requests at the disk drive of a Web server, see [6] for a description of this trace and related analyses of its temporal dependence structure.

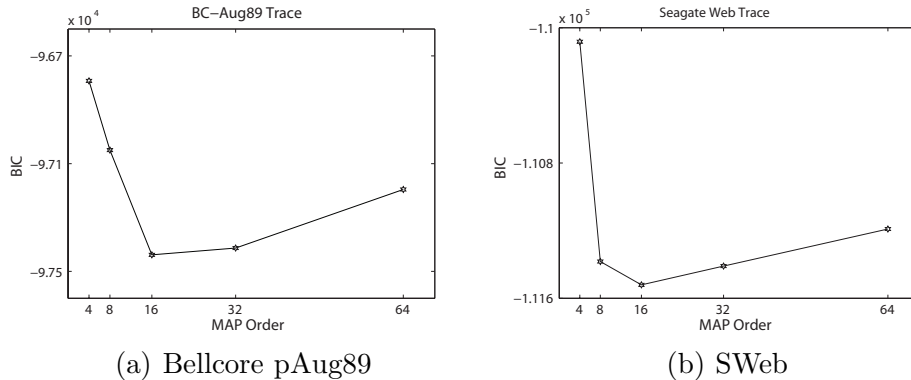


Fig. 6. Order selection for the BC-pAug89 and SWeb traces

We remark that the two traces are used also in the manually-performed fitting experiments in [6]. However, the results presented here are generated *automatically*, and not manually, using the KPC-Toolbox. In this section, we compare the results for the disk drive and the networking traces with the manually-generated results in [6] in order to prove that the automatic fitting approach provides comparable accuracy of a much more time-consuming manual fitting.

5.1 Order Selection Results

For each of the three traces, the KPC-Toolbox runs the *BIC* order selection for different orders. The results of the *BIC* order selection are shown in Figure 6. For the Bellcore pAug89 trace, the *BIC* selection in Figure 6 indicates that order 16 is the best choice, order 32 is a close candidate, while the other orders are significantly worse in terms of *BIC*. Similarly, for the SWeb trace, the *BIC* criterion indicate a best order of 16 phases. These results are consistent with our manual fitting experience, since it is difficult to fit MAPs with less than 16 phases and composed by MAP(2)s that could reproduce effectively the SWeb and Bellcore traces queueing performance. In particular, the best available MAPs fitting in the literature for the Bellcore trace have either 16 or 32 states [1, 11] which is consistent with the *BIC* results in Figure 6.

5.2 Queueing Results

In Figures 7 and 8, we present the complementary cumulative distribution function (CCDF) of the queue length probabilities for a *MAP/M/1* queue and the empirical CCDF obtained by simulating the *Trace/M/1* queue. The CCDFs of the *MAP/M/1* queue are obtained by solving the underlying quasi-birth death process using MAMSolver (<http://www.cs.wm.edu/MAMSolver/>). The service rate of the exponential server is adjusted to tune the load of

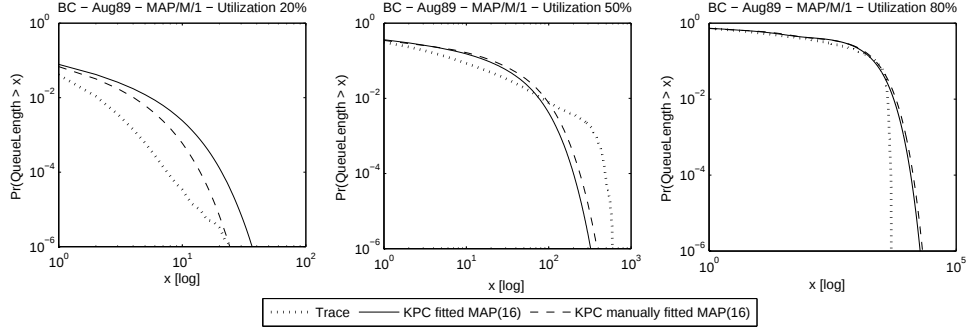


Fig. 7. Queueing results for comparison between the Bellcore pAug89 trace and the fitted MAP

the server at different utilization levels. We plot the queueing distributions at utilizations $\rho = \{0.2, 0.5, 0.8\}$ representing low, medium and high load, respectively.

The Bellcore pAug89 trace has been identified as a difficult trace to fit and thus extensively used in the literature to assess fitting accuracy. For utilization 0.2, the fitted MAP(16) overestimates the queueing probabilities. When the utilization increases, the queueing probability prediction improves. For utilization 0.5, the fitted MAP captures the small and medium queue lengths probabilities better than at the 0.2 utilization level. At 0.8 utilization, the fitted MAP(16) almost overlaps the trace up to the queue length value of 8×10^3 . Further, as shown in comparison with the manual fitting, the queueing prediction accuracy for the KPC is comparable with the best in the literature [6] with the additional benefit that KPC automatically generates the MAP while other methods require exhaustive manual tuning to obtain a good MAP. Similarly to the SWeb trace, as utilization increases, the fitted MAP(16) captures more accurately the queueing behavior of the *Trace/M/1* queue. For utilization levels 0.5 and 0.8, the fitted MAP captures the probabilities of the smallest queue lengths very well. The result is again comparable with the manually fitted MAP determined in [6] and also plotted in the figure. The manually fitted MAP in [6] is the best result of experimentation over an extended period of time with models of several sizes. Thus, it is slightly better than the automatically generated MAP that has been produced in less than an hour by the KPC-Toolbox. Still the results are quite close, and this provides intuition on the effectiveness of the automatic fitting approach we have developed.

6 PH-Type Distribution Fitting

The analysis presented in the previous sections provides fitting guidelines for traces characterized by temporal dependence and correlations between samples. Indeed, there exist many traces of practical interest where samples are

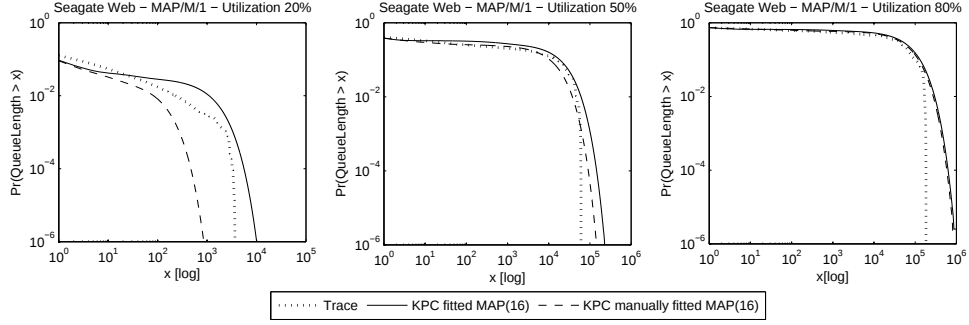


Fig. 8. Queueing results for comparison between the SWeb trace and the fitted MAP

nearly independent, i.e., *not* significantly correlated with each other, and in these cases the analysis focuses on fitting distribution or moments of inter-arrival times rather than correlations. Since inter-arrival times in MAPs are always PH-type distributed [19], we explain in this section how the KPC-Toolbox can be used to fit uncorrelated traces into PH-type distributions using a moment-matching approach³ We also present numerical results on several traces showing the effectiveness of the KPC-Toolbox in PH-type fitting.

Let J be the number of MAPs used for KPC PH-type fitting and denote by X_j , $j = 1, \dots, J$, the random variable standing for the inter-arrival times of the j th MAP that will be used in the KPC. By recursive application of (7) we can write the k th moment of the KPC process as

$$E[X^k] = k!^{1-J} \prod_{j=1}^J E[X_j^k],$$

or equivalently, by taking the logarithm of both sides,

$$\log E[X^k] = (1 - J) \log k! + \sum_{j=1}^J \log E[X_j^k],$$

where $E[X^k]$ denotes the k th moment of the KPC process obtained by composition of the J MAPs.

The problem under consideration is to impose the first M moments of the KPC process such that they closely match the trace moments $M_k = E[\widehat{X}^k]$, $k = 1, \dots, M$, where $\widehat{X}$ is the random variable denoting a measured inter-arrival time in the trace. Since the J MAPs used in KPC-Toolbox are all MAP(2)s, it is straightforward to impose the first three moments of the KPC

³ Since we use the (D_0, D_1) notation, we still refer in this section to the PH-type distributions as MAPs, but we implicitly assume that all correlations in the defining MAP(2)s are set to zero.

process by setting for the arbitrary MAP(2)

$$\log E[X_1] = \log M_1 - \sum_{j=2,\dots,J} \log E[X_j] \quad (15)$$

$$\log E[X_1^2] = \log M_2 - (1 - J) \log 2! - \sum_{j=2,\dots,J} \log E[X_j^2] \quad (16)$$

$$\log E[X_1^3] = \log M_3 - (1 - J) \log 3! - \sum_{j=2,\dots,J} \log E[X_j^3] \quad (17)$$

which can be done easily since the first three moments of a MAP(2) can be imposed analytically without the need of specialized algorithms. As we show in Section 7, even in the basic case where only $J = 2$ MAPs are composed, KPC fitting is more flexible than two-phase PH-type fitting, since KPC imposes much less restrictive requirements on the feasibility of the third moment $E[\widehat{X}^3]$. Additionally, the first $J - 1$ MAPs can be used to match higher-order moments of the distribution. In particular, since a MAP(2) has 3 degrees of freedom for fitting moments, the KPC process used in PH-type fitting has no less than $3J$ degrees of freedom, among which $3(J - 1)$ can be spent for fitting higher-order moments, e.g., $E[X^4]$, $E[X^5]$, $E[X^6]$. Figure 9 summarizes the PH-type fitting algorithm implemented in the KPC-Toolbox. This a nonlinear optimization program in the variables $E[X_j^k]$, $j = 1, \dots, J$, $k = 1, \dots, 3$, which are the first three moments that uniquely define the distribution of the J MAP(2)s used in KPC. The correlations of these MAP(2)s are all set to zero. At each iteration of the nonlinear program, the MAP(2)s are generated using the closed-form fitting formulas of [9] and, whenever an infeasible MAP(2) is obtained, this is immediately corrected to restore feasibility by setting the second and third moments to the closest feasible values according to the MAP(2) characterizations in [9, 10]. This allows to have feasible MAPs throughout all stages of the PH-type fitting, thus if the program does not find an optimal solution it can still return the best feasible MAP obtained throughout the iterations. For each iteration, given the set of J feasible MAP(2)s, the KPC process is assembled and its first M moments computed according to (2). The PH-type fitting program seeks with the constraint (20) to obtain a KPC process which has the first m moments identical to those of the trace and further uses as objective function the difference in moments of order between $m + 1$ and M . An experimental validation of the PH-type approach used in the KPC-Toolbox is reported below.

6.1 Experimental Validation

We present an experimental validation of the KPC-Toolbox PH-type fitting capabilities. We have considered a set of workload traces with the characteristics reported in Table 2. The SWeb* and BC-pAug89* traces used in the

Fig. 9. KPC PH-type Fitting

$$\text{minimize } \sum_{k=m+1}^M |\log M_k - \log E[X^k]| \quad (18)$$

$$\text{subject to} \quad (19)$$

$$\log E[X^k] = \log M_k, \quad k = 3, \dots, m \quad (20)$$

$$\log E[X_1] = \log M_1 - \sum_{j=2, \dots, J} \log E[X_j] \quad (21)$$

$$\log E[X_1^2] = \log M_2 - (1 - J) \log 2! - \sum_{j=2, \dots, J} \log E[X_j^2] \quad (22)$$

$$\log E[X_1^3] = \log M_3 - (1 - J) \log 3! - \sum_{j=2, \dots, J} \log E[X_j^3] \quad (23)$$

queueing experiments are obtained by randomly shuffling the SWeb and BC-pAug89 traces to removing temporal dependence.

For each trace, the KPC-Toolbox returns a MAP with PH-type distribution fitting the inter-arrival time samples and that is generated as follows. Since the time for solving the optimization program in Figure 9 is typically less than 1 minute, the toolbox is able to evaluate alternatives much faster than in the case of correlated traces. Therefore, a whole sequence of experiments for increasing values of J can be considered without the need of a *BIC* order selection. The toolbox selects as optimal order of the PH-type process the smallest order such that the fourth moment $E[X^4]$ is matched by the KPC process within a tolerance of 1%; this is the first moment that cannot be matched exactly by a MAP(2). After performing 30 trials on the current value J , the KPC process increases J by one unit. On all experiments considered in this section, we have found that using $J = 2$ MAPs was always sufficient to fit the fourth moment within the 1% tolerance constraint.

Table 3 reports experimental results of the KPC PH-type fitting for the traces in Table 2. Each experiment reports the values of the first six moments of each trace, for the PH-type fitting obtained by KPC, and for the MAP(2) which fits analytically the first three moments of the trace. The results indicate that the KPC PH-type fitting matches the fourth moment very well on all traces

Table 2

Trace characteristics

	<i>Trace Length</i>	<i>Description</i>
SWeb*	3,639,838	Disk drive inter-arrival times (web server)
SDev*	425,114	Disk drive inter-arrival times (development server)
BC-pAug89*	999,999	LAN traffic

	<i>Basic moments</i>			<i>Higher-order moments</i>		
	$E[X]$	$E[X^2]$	$E[X^3]$	$E[X^4]$	$E[X^5]$	$E[X^6]$
SWeb*	1	4.329	58.42	5520.9	$9.798 \cdot 10^5$	$1.996 \cdot 10^8$
KPC-PH(4)	1	4.329	58.42	<u>5520.9</u>	<u>$1.294 \cdot 10^6$</u>	<u>$3.868 \cdot 10^8$</u>
MAP(2)	1	4.329	58.42	1294.6	37032	$1.277 \cdot 10^6$
SDev*	1	135.86	$1.813 \cdot 10^5$	$4.269 \cdot 10^8$	$1.102 \cdot 10^{12}$	$2.924 \cdot 10^{15}$
KPC-PH(4)	1	135.86	$1.813 \cdot 10^5$	<u>$4.266 \cdot 10^8$</u>	<u>$1.275 \cdot 10^{12}$</u>	$4.579 \cdot 10^{15}$
MAP(2)	1	135.86	$1.813 \cdot 10^5$	$3.262 \cdot 10^8$	$7.334 \cdot 10^{11}$	<u>$1.979 \cdot 10^{15}$</u>
BC-pAug89*	1	4.224	64.76	1862.6	82474	$5.136 \cdot 10^6$
KPC-PH(4)	1	4.224	64.76	<u>1862.6</u>	<u>82472</u>	<u>$5.742 \cdot 10^6$</u>
MAP(2)	1	4.224	64.76	1734.2	60332	$2.530 \cdot 10^6$

Table 3

Experimental results of KPC-PH method. The moment values between KPC-PH(k) and MAP(2) that are closest to the measured ones are underlined.

and this requires only the composition of $J = 2$ MAPs. The gap on higher-order moments between the KPC PH-type fitting and the MAP(2) is up to two orders of magnitude. The largest deviations appear in the SWeb* trace, where the KPC process is quite close to the trace, whereas the MAP(2) is far less accurate and two orders of magnitude smaller in the estimate of the sixth moment $E[X^6]$. This can be explained by the fact that a MAP(2) with assigned first three moments has higher-order moments which are uniquely determined as linear combinations of the first three [6]. As a result, the MAP(2) does not offer any ability of imposing higher-order moments such as $E[X^6]$. In the following subsection, we investigate the benefits of fitting higher-order moments with KPC PH-type fitting.

6.2 Impact of Higher-Order Moments on Distribution

We first evaluate how fitting higher-order moments of inter-arrival times helps in approximating the CDF and the CCDF of the traces. Figures 10-12 present fitting results of the three traces in Table 2. The plots compare the inter-arrival time distribution of the original trace with that of the KPC PH-type processes and of the MAP(2)s shown in Table 3. The results indicate that, by increasing with KPC the number of matched moments in the MAP, the largest accuracy improvements are obtained on the tail of the distribution, shown by the CCDF, rather than on the body of the distribution, shown by the CDF. For example, the CDF of the SWeb* trace in Figure 10(a) is rather insensitive to the additional moments fitted by the KPC process compared to the MAP(2); however, if one looks at the CCDF in Figure 10(b), the accuracy

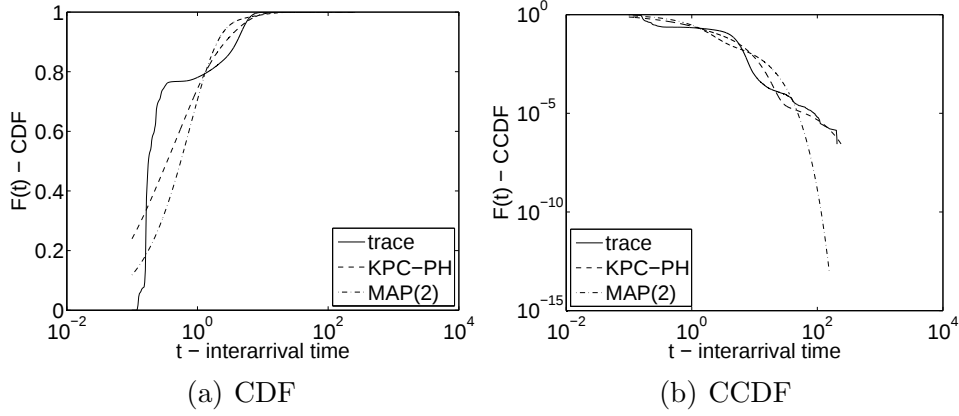


Fig. 10. PH-Type Fitting of SWeb* trace

of the KPC process is immediately visible and dramatically better compared to the one of the MAP(2).

Similar conclusions follow also from the SDev* trace in Figure 11. In the CCDF plot, MAP(2) and KPC both achieve good accuracy, with the latter being the closest to the trace; the CDF of the KPC process again does not improve significantly over the one of the MAP(2) process. The results for the BC-pAug89* trace in Figure 12 are consistent as well with the observations for the SWeb* and SDev* traces. Here, the KPC and the MAP(2) approximations of the CDF and of the CCDF are very close and the KPC is more accurate in both cases and in particular on the CCDF, where the KPC plot is indistinguishable from the trace for most of the graph.

Summarizing, fitting higher-order moments of the distribution using KPC helps in achieving a much improved approximation of the tail of the distribution. A direct approximation of the body of the CDF may instead require methods other than moment matching, which appears to be rather ineffective in improving the accuracy of the CDF body approximation. In the following subsection, we investigate to which extent a better approximation of the tail, rather than of the body, can improve the quality of the queueing prediction of $PH/M/1$ models.

6.3 Queueing Results

We evaluate the practical impact of the KPC PH-type fitting in improving the prediction accuracy of queueing models. We consider, for each trace, a FCFS queue fed by a PH-type distributed inter-arrival process and with exponential service times, i.e., a $PH/M/1$ queue. Similarly to the experiments for the correlated traces, we focus on utilization levels $\rho = 0.2, 0.5, 0.8$ and evaluate the approximation accuracy of the overflow probability for different values of

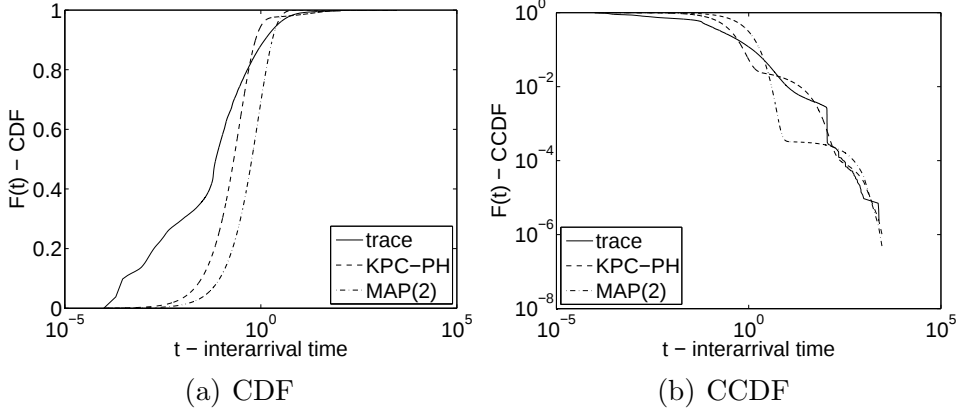


Fig. 11. PH-Type Fitting of SDev* trace

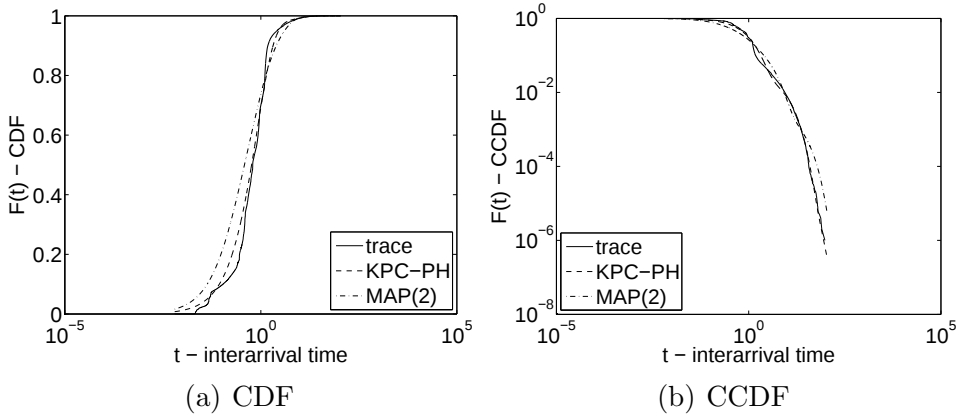


Fig. 12. PH-Type Fitting of Bellcore pAug89* trace

the overflow threshold x . Experimental results are shown in Figures 13-15.

The results strongly indicate that a better fitting of the tail leads to strong improvements in the queueing prediction accuracy for almost all utilization levels and all traces. For the SWeb* trace in Figure 13, both the KPC and the MAP(2) process provide optimistic estimates on the overflow probabilities, with the KPC approximation being much closer to the trace results than the MAP(2), especially for $\rho = 0.2$ and $\rho = 0.5$.

The SDev* trace in Figure 14 presents a case that illustrates the importance of fitting accurately moments such as $E[X^4]$ and $E[X^5]$ rather than higher-order moments such as $E[X^6]$. The dramatic difference in the results in Figure 14 suggests that it is far more important to approximate accurately medium and low-order moments rather than very high-order ones, since the MAP(2) is unable to follow the overflow probability trend for utilizations 0.5 and 0.8. Also for utilization 0.2 there is a significant deviation with respect to the KPC process, thus suggesting that $E[X^4]$ and $E[X^5]$ can be important also at low utilization.

Finally, the results in Figure 15 for the Bellcore pAug89* trace show a case where the KPC and the MAP(2) results are both pessimistic with respect to the trace overflow probability curve. At all utilization levels, there is a minor deviation between the KPC curve and the empirical one. Instead, the MAP(2) curve converges slowly to the trace results, with the tail value remaining quite apart from the empirical one at all utilization levels.

6.4 Comparison with Other PH Fitting Methods

We have compared the results presented in the previous section with other PH-type distribution fitting methods, specifically the PH(3) moment matching algorithm presented in [12] and the G-Fit tool introduced in [20]. PH(3)s are currently the only general phase-type models with more than two states that can be fitted exactly using analytical techniques [12]. We have used the fitting methods presented in [12, 15] to first fit a matrix exponential process (MEP) and then transform this model into a PH(3) distribution. This transformation may lead either to a feasible or to an infeasible process depending on the characteristics of the original MEP. We have reported in Appendix A the matrices that describe MEPs that fit exactly the first five moments of the the SWeb*, SDev*, and BC-pAug89* traces. These MEP processes have been obtained by the exact fitting algorithm described in [14]. The algorithm in [12] has been then used to transform a MEP into a PH(3) distribution.

Using the above methodology we have found that two of the three considered traces do not seem to admit a feasible PH(3) representation that matches exactly the first five moments. For instance, for the SWeb* trace, we have found the following PH(3) model in the $(\pi, \mathbf{A})$ notation of [12]

$$\pi = [-2.9346 \ -678.07 \ 682.00], \quad \mathbf{A}^{SWeb*} = \begin{bmatrix} -0.5468 & 0 & 0 \\ 0.0274 & -0.0274 & 0 \\ 0 & -4.7228 & 4.7228 \end{bmatrix},$$

which is clearly infeasible. This illustrates a case where more than three states should be used to fit accurately a sample distribution. We have instead obtained nearly identical results with the PH(3) and KPC-PH(4) models on the BC-pAug89* trace distribution.

We have also compared the KPC-PH fitting results with the G-Fit tool version 1.1 [20] on the same set of traces considered above. The BC-pAug89* trace fitted using the same log-aggregation parameters recommended in [20], we considered 100 random runs of G-Fit with different number of Erlang branches and assuming a 4 state model as in KPC-PH. The best result with

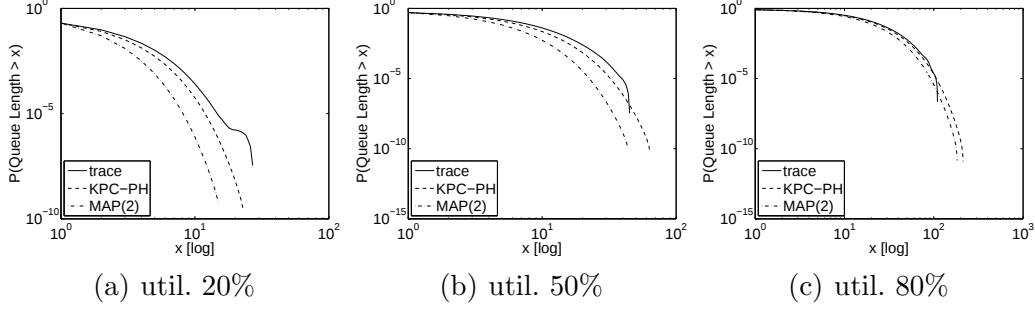


Fig. 13. PH-Type Fitting of SWeb* trace - Queueing Results

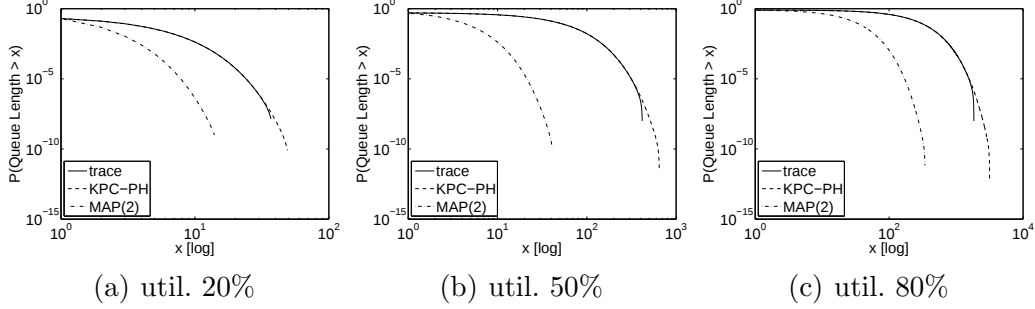


Fig. 14. PH-Type Fitting of SDev* trace - Queueing Results

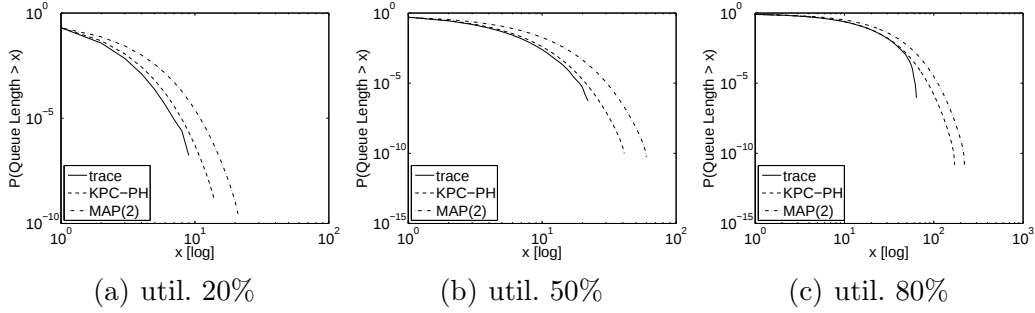


Fig. 15. PH-Type Fitting of Bellcore pAug89* trace - Queueing Results

respect to the *SCV* matching error is $E[X] = 1.0$, $E[X^2] = 4.2467$ (error -0.55%), $E[X^3] = 53.9286$ ($+16.72\%$), $E[X^4] = 1.0736 \cdot 10^3$ ($+42.35\%$), $E[X^5] = 2.6626 \cdot 10^4$ ($+67.71\%$), $E[X^6] = 7.7471 \cdot 10^5$ ($+84.91\%$). This result has higher errors than the KPC-PH(4) model. This further suggests that the KPC-PH fitting methodology provides a useful alternative to existing PH-type fitting approaches.

7 Characterization of MAP(2)-Based KPC Processes

In order to illustrate the range of feasible moments and autocorrelations that can be matched exactly by the KPC-Toolbox, we investigate the characterization of KPC processes obtained by composition of two MAP(2)s, i.e., KPC(4)

processes. This represents the case where the KPC process has the least degrees of freedom, hence the characterization results we obtain in this section describe the minimum possible advantages achieved by the KPC processes over MAP(2)s.

Throughout this section, we denote by X the inter-arrival times of the KPC process $MAP^a \otimes MAP^b$. Based on the KPC process requirements, we always assume that MAP^a is an arbitrary MAP(2), while MAP^b has diagonal D_0 such that its distribution is hyper-exponential. All statistical descriptors of MAP^a and MAP^b are marked with indexes a and b .

The main results we obtain in this section are as follows:

- By studying analytically the first three moments of the KPC(4) process we find it much more flexible than a MAP(2). Specifically, we show that the range of variability of $E[X^3]$ is significantly larger than in a MAP(2) both in the hypo-exponential and hyper-exponential cases. We stress that these results apply both to MAP fitting and PH-type fitting.
- We study numerically the feasibility range of the lag-1 autocorrelation coefficient ρ_1 of the KPC(4) process and find that it can achieve autocorrelation values upper bounded by $\frac{2}{3}$, whereas the MAP(2) autocorrelation is upper bounded by $\frac{1}{2}$. The lower bounds on ρ_1 of KPC(4) and MAP(2) are often identical. Since bursty traces typically have positive autocorrelations, we believe the extension of the upper bound of ρ_1 to be useful for practical fitting of such traces.

7.1 Characterization of Moments

We first focus on the derivation of the feasibility ranges for $E[X^k]$, $k = 1, 2, 3$, based on the equivalent Hankel determinant notation h_k , $k = \{1, 2, 3\}$, [9, 14]. Since the first three h_k values uniquely identify a triplet of moments $E[X^k]$, $k = \{1, 2, 3\}$, studying the feasibility ranges of Hankel determinants immediately characterizes also the feasibility range of moments by simple algebra.

Lemma 1 *Consider the first three normalized Hankel determinants*

$$h_1 = E[X], \quad h_2 = \frac{E[X^2]}{2E[X]^2} - 1, \quad h_3 = \frac{E[X^3]}{6E[X]^3} - h_2^2.$$

Then, any KPC process $MAP^a \otimes MAP^b$ composed by two MAPs of arbitrary size has

$$h_1 = h_1^a h_1^b, \tag{24}$$

$$h_2 = h_2^a h_2^b + h_2^a + h_2^b, \tag{25}$$

$$h_3 = h_3^a h_3^b + h_3^a(1 + h_2^b)^2 + h_3^b(1 + h_2^a)^2, \tag{26}$$

where h_k^a and h_k^b , $k = \{1, 2, 3\}$, are the Hankel determinants of the processes MAP^a and MAP^b , respectively. In particular, h_2 satisfies the relation

$$(1 + h_2) = (1 + h_2^a)(1 + h_2^b), \quad (27)$$

which is equivalent to (25).

Proof 1 Define by X_a and X_b the random variables denoting the inter-arrival times in MAP^a and MAP^b , respectively. By the given definitions it is

$$h_2^a h_2^b = \left(\frac{E[X_a^2]}{2E[X_a]^2} - 1 \right) \left(\frac{E[X_b^2]}{2E[X_b]^2} - 1 \right) = \frac{E[X_a^2]E[X_b^2]}{4E[X_a]^2E[X_b]^2} - h_2^a - h_2^b - 1,$$

and using (7) for $E[X^2]$ and $E[X]$ we get the formula for h_2 . Equation (27) is verified by expanding the products and noting that the terms simplify to (25).

The proof of the formula for h_3 is similar. Consider the following derivation:

$$\begin{aligned} h_3^a h_3^b &= \left(\frac{E[X_a^3]}{6E[X_a]^3} - (h_2^a)^2 \right) \left(\frac{E[X_b^3]}{6E[X_b]^3} - (h_2^b)^2 \right) \\ &= h_3 - h_3^a(1 + h_2^b)^2 - h_3^b(1 + h_2^a)^2 - (1 + h_2)^2 + (1 + h_2^a)^2(1 + h_2^b)^2 \end{aligned}$$

where we used again (7) for $E[X^k]$, $k = \{1, 2, 3\}$. Noting that the last two terms are identical by (27), the final expression for h_3 is obtained immediately. $\square$

The advantage of working with Hankel determinants is that the MAP(2) feasibility ranges for the first three moments have much simpler analytic expressions with this notation than with the $E[X^k]$ moment notation. In particular, the sign of h_2 immediately discriminates between the MAP(2) distribution being exponential ($h_2 = 0$), hyper-exponential ($h_2 > 0$), or hypo-exponential ($h_2 < 0$). Feasibility ranges for AMAP(2)s have been characterized in [9] and the stationary behavior of AMAP(2)s and MAP(2)s has been shown to be identical in [3]. Thus, in a MAP(2) it is always $-\frac{1}{4} \leq h_2$ and for h_3 [9] is

$$\begin{cases} h_2(1 - h_2 - 2\sqrt{-h_2}) \leq h_3 \leq -(h_2)^2, & \text{for } -\frac{1}{4} \leq h_2 < 0 \quad (\text{hypo}) \\ h_3 = 0, & \text{for } h_2 = 0 \quad (\text{exp}) \\ 0 \leq h_3 < +\infty, & \text{for } h_2 > 0 \quad (\text{hyper}) \end{cases}$$

We now obtain a similar characterization for the KPC(4) process.

Theorem 1 In a KPC(4) process, the feasibility ranges of the first two moments are characterized as $h_1 > 0$ and $h_2 \geq \frac{1}{4}$, for the third moment it is

$$\begin{cases} -\frac{1}{9}(1 + h_2)^2 \leq h_3 < +\infty, & \text{for } -\frac{1}{4} \leq h_2 < 0 \quad (\text{hypo}) \\ h_3 = 0, & \text{for } h_2 = 0 \quad (\text{exp}) \\ -\frac{1}{9}(1 + h_2)^2 \leq h_3 < +\infty, & \text{for } h_2 > 0 \quad (\text{hyper}) \end{cases}$$

In particular, the minimum h_3 value is obtained when the arbitrary MAP(2) describes an Erlang-2 distribution.

Proof 2 For the KPC(4) process $MAP^a \otimes MAP^b$, MAP^a is an arbitrary MAP(2), while MAP^b is either exponential or hyper-exponential. Thus, h_1^a , h_2^a , and h_3^a ranges are identical to a general MAP(2), while it is $h_2^b \geq 0$ and $h_3^b \geq 0$ for MAP^b and these are equalities if and only if MAP^b is exponential.

The ranges for h_2 are proved easily by observing that h_2 in (25) is monotonically increasing with h_2^a and h_2^b , thus h_2 is unbounded since h_2^b is always unbounded. Conversely, it is trivial to see that the lower bound is achieved for the minimum values $h_2^a = -\frac{1}{4}$ and $h_2^b = 0$ which give $h_2 = -\frac{1}{4}$.

The ranges for h_3 are characterized by assuming h_1 and h_2 as constant and then studying the feasibility range of h_3 as a function of the other determinants. We begin by noting that for the hyper-exponential MAP^b the values of h_2^b and h_3^b are independent since they need only to be nonnegative for MAP^b feasibility, hence the derivative of h_3 with respect to h_3^b simplifies

$$\left. \frac{\partial h_3}{\partial h_3^b} \right|_{\substack{h_1=\text{const} \\ h_2=\text{const}}} = h_3^a + (1 + h_2^a)^2 \geq h_2^a(1 - h_2^a - 2\sqrt{-h_2^a}) + (1 + h_2^a)^2 \geq \frac{1}{2} > 0$$

where we have used that in the arbitrary two-phase MAP^a it is $h_3^a \geq h_2^a(1 - h_2^a - 2\sqrt{-h_2^a})$ and that $h_2^a \geq -\frac{1}{4}$. Thus, the minimum of h_3 is obtained in the limit $h_3^b \rightarrow 0$ and we have that

$$\min h_3|_{\substack{h_1=\text{const} \\ h_2=\text{const}}} = \lim_{h_3^b \rightarrow 0} h_3 = h_3^a(1 + h_2^b)^2 = \frac{h_3^a}{(1 + h_2^a)^2}(1 + h_2)^2$$

where the last passage follows by (27). Now, observe that a feasible $h_2 \geq -\frac{1}{4}$ can always be obtained by first setting $h_2^a = -\frac{1}{4}$ and then setting

$$h_2^b = \frac{(1 + h_2)^2}{(1 + h_2^a)^2} - 1 > \frac{16}{9}(1 + h_2)^2 - 1 > \frac{16}{9}(1 - \frac{1}{4})^2 - 1 \geq 0$$

which is always feasible. Therefore, regardless of the h_1 and h_2 values, it is always possible to have h_3^a set to its lower bound $h_3^a = h_2^a(1 - h_2^a - 2\sqrt{-h_2^a})$. Without loss of generality, this implies that the worst case

$$\min h_3|_{\substack{h_1=\text{const} \\ h_2=\text{const} \\ h_2^a=-1/4}} = \frac{h_3^a}{(1 + h_2^a)^2}(1 + h_2)^2 = \frac{h_2^a(1 - h_2^a - 2\sqrt{-h_2^a})}{(1 + h_2^a)^2}(1 + h_2)^2$$

is always feasible and this provides a lower bound on $\min h_3|_{\substack{h_1=\text{const} \\ h_2=\text{const}}}$. Noting

that the right hand side is minimum for $h_2^a = -\frac{1}{4}$, this gives the final result

$$\min_{h_1=const, h_2=const} h_3 \geq \frac{h_2^a(1 - h_2^a - 2\sqrt{-h_2^a})}{(1 + h_2^a)^2} (1 + h_2)^2 \geq -\frac{1}{9}(1 + h_2)^2.$$

Note in particular that in the hypo-exponential case $h_2 = -\frac{1}{4}$ the minimum converges to the MAP(2) minimum $\min h_3 = -\frac{1}{4}$, however for $h_2 \rightarrow 0$ it is $\min h_3 \rightarrow -1/9$ which is a lower bound on h_3 for the KPC(4) process. Conversely, for $SCV \rightarrow +\infty$ it is $h_2 \rightarrow +\infty$ and h_3 becomes unbounded.

The last result immediately implies that a KPC(4) supports a wider range of feasibility values for the third moment $E[X^3]$ compared to the MAP(2) process. In particular, also hypo-exponential processes can have positive h_3 values, and for both hyper-exponential and hypo-exponential cases h_3 is lower bounded by $-\frac{1}{9}(1 + h^2)$ while no upper bound exists. This means that for sufficiently large h_2 any positive or negative h_3 value can always be matched by the KPC(4) model.

7.2 Characterization of Autocorrelations

We now focus on the characterization of the autocorrelation function ρ_k for the KPC(4) process $MAP^a \otimes MAP^b$. Due to the properties of KPC, we have that the relation between the autocorrelations of MAP^a and MAP^b is summarized by (11), where for MAP(2)s we have, e.g., $\rho_k^a = \frac{1}{2}(1 - 1/SCV_a)\gamma_a^k$, being γ_a the smallest eigenvalue of $P^a = (-D_0^a)^{-1}D_1^a$. This immediately implies that, differently from a MAP(2), in a KPC(4) process the autocorrelation function decays according to the rates γ_a , γ_b , and $\gamma_a\gamma_b$. Indeed, this gives greater flexibility in fitting measured autocorrelation coefficients since, by properly selecting the values of γ_a and γ_b , one can represent different decaying trends at low and high lags. This is impossible in MAP(2)s where the decay rate is constant throughout all the trace whereas in KPC(4) the KPC-Toolbox can help in automatically selecting the desired γ_a and γ_b values that match a given trace.

In practice, since the absolute value of the decay rates γ_a and γ_b is strictly less than 1, ρ_k always decays⁴ with k and the characterization of which autocorrelation coefficients can be fitted is mostly determined by the value of ρ_1 . This is also the case of MAP(2)s, where studying ρ_1 as a function of SCV

⁴ This is always true when k is not too small and γ_a and γ_b are not both equal to -1 . For very small values such as $k = 2, 3$ some effects due to degenerate Jordan form of the P matrix may result in increasing autocorrelation values [6], however these are usually degenerate cases that affect only low lag autocorrelation coefficients.

fully characterizes the MAP(2) autocorrelation function. Thus, in this subsection we provide a numerical study on the achievable ρ_1 values for the KPC(4) process in comparison to a MAP(2).

Figure 16 illustrates the feasibility range of ρ_1 as a function of the squared coefficient of variation of the KPC(4) and MAP(2) processes. The figure has been generated by evaluating over 50,000 pairs (MAP^a, MAP^b) spanning all range of feasible $\gamma_a, \gamma_b, SCV_a$, and SCV_b values that guarantee the considered SCV value. In these experiments, we set the third moments of both MAP^a and MAP^b to the values that provide maximal feasibility in the MAP(2) autocorrelation coefficients; these values have been studied in [9].

The results in Figure 16 indicate that the KPC(4) has much greater flexibility than the MAP(2) process in matching exactly autocorrelation coefficients, especially for positive values of ρ_1 . Remarkably, KPC(4) is able to match ρ_1 values much larger than the $\frac{1}{2}$ upper bound of the MAP(2) and which are significantly large also for the hypo-exponential case $SCV \leq 1$. This last property address a known issue of MAP(2)s, which have very limited feasible autocorrelation ranges for low SCV values [10]. Conversely, for negative autocorrelations KPC(4) and MAP(2) have very similar ranges, except in the neighborhood of $SCV = 1$ where KPC(4) is slightly more flexible.

We conclude the characterization in this section by studying the maximum asymptotic value of the autocorrelation ρ_1 under an increase of SCV which proves that a KPC(4) can reach autocorrelation values up to $\frac{2}{3}$ that are much greater than the maximum $\frac{1}{2}$ value of a MAP(2).

Theorem 2 *In the limit $SCV \rightarrow +\infty$ the maximum value assumed by the autocorrelation function is $\rho_1 = \frac{2}{3}$ obtained by setting MAP^a as an Erlang-2 distribution and MAP^b with $\gamma_b \rightarrow 1$ and $SCV_b \rightarrow +\infty$. The minimum value is $\rho_1 = 0$.*

Proof 3 *From (10), taking the limit of ρ_k for $SCV \rightarrow +\infty$ implies that either SCV_a or SCV_b or both become infinite. Since MAP^a is arbitrary, we can assume without loss of generality that $SCV_b \rightarrow +\infty$ and consider both cases where SCV_a is finite or infinite. Let us first observe that*

$$\rho_1^\infty = \lim_{SCV_b \rightarrow \infty} \rho_1 = \frac{\gamma_b(2 + \gamma_a(SCV_a - 1))}{2(SCV_a + 1)}$$

where the formula already accounts for (10). Note that since γ_b for large SCV becomes lower bounded by $\gamma_b = 0$ [9], we conclude immediately that the lower bound of the asymptotic lag-1 coefficient is $\rho_1^\infty = 0$.

In determining the upper bound, consider first the case $SCV_a \rightarrow +\infty$, we have that $\rho_k \rightarrow \gamma_b \gamma_a / 2$ which is immediately upper bounded by $\frac{1}{2}$ as in a MAP(2) [9].

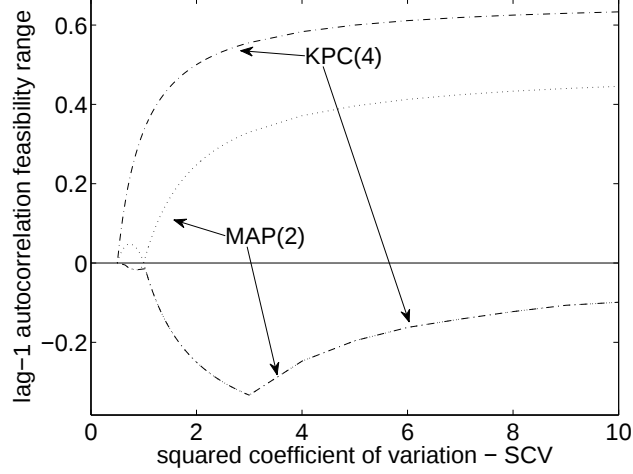


Fig. 16. **Feasibility range of the lag-1 autocorrelation coefficient ρ_1 .** Dotted lines indicate MAP(2) ranges, dash-dotted lines are for KPC(4) ranges.

Let us then focus on the case where SCV_a is finite, we distinguish between the two sub-cases $SCV_a \geq 1$ and $SCV_a < 1$. If $SCV_a \geq 1$

$$\frac{\partial \rho_1^\infty}{\partial \gamma_a} = \frac{\gamma_b(SCV_a - 1)}{2(SCV_a + 1)} \geq 0$$

since for $SCV_b \rightarrow +\infty$ it is always $0 \leq \gamma_b \leq 1$. We therefore conclude that for $SCV_a \geq 1$ the autocorrelation coefficients are maximized for $\gamma_a \rightarrow 1$ which yet yields again an upper bound of $\frac{1}{2}$. Note that the case $SCV_a = 1$ does not yield an improvement because it again corresponds to the upper bound of $\frac{1}{2}$ since the resulting autocorrelation is identical to that of MAP^b.

Conversely, for $SCV_a < 1$ it is easy to see that ρ_1^∞ is maximized by setting γ_a to its minimum feasible value that is a function of SCV_a according to the expression [9]

$$\min \gamma_a = - \left(\frac{h_3^a}{h_2^a} + h_2^a \right) \geq - \left(\frac{h_2^a(1 - h_2^a - 2\sqrt{-h_2^a})}{h_2^a} + h_2^a \right) = 1 - 2\sqrt{\frac{1 - SCV_a}{2}}$$

Inserting the last expression into ρ_1^∞ , the expression becomes a function of SCV_a and γ_b and it is found that the maximum is always achieved for $SCV_a = \frac{1}{2}$ and $\gamma_b \rightarrow 1$ which gives by simple passages $\rho_1^\infty = \frac{2}{3}$.

8 Conclusion

We have presented the KPC-Toolbox, a set of MATLAB scripts for fitting workload traces into MAPs. One of the greatest challenges in MAP fitting is

to (a) decide a proper order for the MAP that can fit the trace data accurately while limiting the dimension of the state space, (b) determine the relative importance of the various stochastic descriptors of the trace that should be matched by the MAP. The KPC-Toolbox meets the above challenges with a novel approach that uses the *BIC* criterion to determine the best order-accuracy trade-off for a MAP and by using optimization to explore a vast parameter space of alternatives such that the most important stochastic properties of the trace are captured by the resulting MAP. The KPC-Toolbox can be used to fit into MAPs traces that exhibit temporal dependence as well as the simpler cases where there is negligible temporal dependence in the successive samples. For traces with temporal dependence, detailed queueing analysis that confirms the importance of matching higher-order correlations (i.e., joint moments) rather than higher-order moments is used to guide the optimization. Instead, if there is no temporal dependence in the trace, then KPC strives to maximize the number of high-order fitted moments to guide the optimization.

The tool implements the theoretical results of [6], and guided by *BIC* and the various derived criteria for fitting, it produces good MAP fittings of challenging temporal dependent traces in an automatic way. Experimental results on real traces from both the computer systems and networking domains show the effectiveness of deriving a MAP that captures well the workload characteristics. We have also presented an extension of the KPC-Toolbox to fit traces without correlations between samples and a characterization of KPC processes that provides insights on the applicability of the toolbox. The KPC-Toolbox is available for download at <http://www.cs.wm.edu/MAPQN/kpctoolbox.html>.

Open challenges for further development of the KPC-Toolbox are at least two. First, we would like to integrate in the toolbox MAP processes different from MAP(2)s as basic building block of the KPC composition. A special class of MAP(3)s that can be integrated effectively with KPC has been presented in [6], however these models are not still supported by the KPC-Toolbox. In addition, we remark that we have performed preliminary experiments with a number of traces with autocorrelation function that clearly shows periodic oscillations of the autocorrelation coefficients. For instance, video streaming traces often show this characteristic. Such traces are impossible to fit with the present version of the KPC-Toolbox because MAP(2)s lack support for complex eigenvalues that create periodicities in the autocorrelations and this is propagated to the KPC process. As a result, the inter-arrival time based fitting of traces with oscillations in the autocorrelation function remains an open challenging problem for MAP fitting research and related tools.

A Appendix

A.1 Matrix Exponential Distribution Fitting Results

We report below the matrix exponential processes (MEPs) with representation $(\mathbf{v}, \mathbf{H})$ [12] that fit the SWeb*, SDev*, and BC-pAug89* traces according to the algorithm presented in [14]. We have focused on MEPs with three states since these can be easily converted into a PH(3) process using the canonical transformation defined in [12]. In all cases the initialization vector is found equal to $\mathbf{v} = [1/3 \ 1/3 \ 1/3]$, while the $\mathbf{H}$ matrices are

$$\mathbf{H}^{SWeb*} = \begin{bmatrix} 4.508180106857546 & 1.046851029180154 & 1.259164846086030 \\ -8.055669274608974 & -2.550582061501703 & -2.857588169939868 \\ 7.389002607942255 & 1.883915394835023 & 2.190921503273185 \end{bmatrix},$$

$$\mathbf{H}^{SDev*} = \begin{bmatrix} -0.423438168388604 & -0.421460282137257 & -0.421436744980477 \\ -11.487646469632512 & -11.239841316612891 & -11.242842474868281 \\ 10.820979802977188 & 10.573174649957339 & 10.576175808212732 \end{bmatrix},$$

$$\mathbf{H}^{BCAug89*} = \begin{bmatrix} -0.721732605907524 & -0.431194300379008 & -0.460567399256791 \\ -0.300027946513111 & -0.351562083855954 & -0.295802597267175 \\ -0.366638720153555 & -0.315104582810712 & -0.370864069399491 \end{bmatrix},$$

The first five moments $E[X^k] = k! \mathbf{v}(-\mathbf{H}^{-k})\mathbf{e}$ are identical to the measured ones for all traces; however numerical issues can arise in computing these values because the $\mathbf{H}$ matrices for the SWeb* and SDev* traces returned by the algorithm in [14] are ill conditioned. Thus, we recommend exact algebra environments such as Mathematica or Maple for the evaluation of these moments.

Acknowledgement

This work was supported by the National Science Foundation under grants CCF-0811417 and CNS-0720699, and by Seagate Research.

References

- [1] A. T. Andersen and B. F. Nielsen. A Markovian approach for modeling packet traffic with long-range dependence. *IEEE JSAC*, 16(5):719–732, 1998.
- [2] A. T. Andersen and B. F. Nielsen. On the use of second-order descriptors to predict queueing behavior of MAPs. *Naval Res. Logistics*, 49(4):391–409, 2002.
- [3] L. Bodrog and A. Heindl and G. Horvath and M. Telek. A Markovian canonical form of second-order matrix-exponential processes. *European Journal of Operational Research*, 190(2):459–477, 2008.
- [4] J. Brewer. Kronecker products and matrix calculus in system theory. *IEEE T. Circuits and Sys.*, 25(9), 1978.
- [5] G. Casale, E.Z. Zhang, E. Smirni. Interarrival times characterization and fitting for Markovian traffic analysis. College of William and Mary Technical Report, Department of Computer Science, TR WM-CS-2008-02. http://www.cs.wm.edu/MAPQN/kpc_toolRelatedPapers.html
- [6] G. Casale, E.Z. Zhang, E. Smirni. Trace Data Characterization and Fitting for Markov Modeling. *Performance Evaluation*, 67(2):61-79, Feb 2010.
- [7] D. Cox and P. Lewis. *The Statistical Analysis of Series of Events*. John Wiley and Sons, New York, 1966.
- [8] J. Diamond and A. Alfa. On approximating higher-order MAPs with MAPs of order two. *Queueing Systems*, 34:269–288, 2000.
- [9] A. Heindl, G. Horvath, and K.Gross. Explicit Inverse Characterizations of Acyclic MAPs of Second Order. Proc. of *EPEW*, Springer LNCS 4054, 108–122, 2006.
- [10] A. Heindl, K. Mitchell, and A. van de Liefvoort. Correlation bounds for second-order MAPs with application to queueing network decomposition. *Perf. Eval.*, 63(6):553–577, 2006.
- [11] A. Horváth and M. Telek. Markovian modeling of real data traffic: Heuristic phase type and MAP fitting of heavy tailed and fractal like samples. In *Performance Evaluation of Complex Systems: Techniques and Tools, IFIP Performance 2002, LNCS Tutorial Series Vol 2459*, pages 405–434, 2002.
- [12] G. Horváth and M. Telek. On the Canonical Representation of Phase Type Distributions. *Perf. Eval.*, 66:396–409, 2009.
- [13] G. Horváth, P. Buchholz, and M. Telek. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag correlation. In *Proc. of 2nd Conf. on Quantitative Evaluation of Systems (QEST)*, pages 124–133, 2005.
- [14] A. van de Liefvoort. The moment problem for continuous distributions. Unpublished technical report, University of Missouri, WP-CM-1990-02, Kansas City (1990).

- [15] L. Lipsky. *Queuing Theory: A Linear Algebraic Approach*. Pearson Publications, 2nd ed., 2009.
- [16] W. E. Leland, M. S. Taqqu, W. Willinger, D. V. Wilson. On the self-similar nature of ethernet traffic. *IEEE/ACM TON*, 2(1):1–15, 1994.
- [17] D. M. Lucantoni. New results on the single server queue with a batch Markovian arrival process. *Stochastic Models*, 7:1–46, 1991.
- [18] N. Mi, Q. Zhang, A. Riska, E. Smirni, and E. Riedel. Performance impacts of autocorrelated flows in multi-tiered systems. *Perf. Eval.*, 64(9-12):1082–1101, 2007.
- [19] M. F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Marcel Dekker, New York, 1989.
- [20] A. Panchenko and A. Thümmel. Efficient phase-type fitting with aggregated traffic traces. *Perf. Eval.*, 64(7-8):629–645, 2007.
- [21] A. Riska, V. Diev, and E. Smirni. An EM-based technique for approximating long-tailed data sets with PH distributions. *Perf. Eval.*, 55(1-2):147–164, Jan. 2004.
- [22] A. Riska and E. Riedel. Long-range dependence at the disk drive level. In *Proc. of 3rd Conf. on Quantitative Evaluation of Systems (QEST)*, pages 41–50, IEEE Press, 2006.
- [23] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- [24] M. Telek and G. Horváth. A minimal representation of Markov arrival processes and a moments matching method. *Perf. Eval.*, 64(9–12):1153–1168, 2007.