Moment Matching-Based Distribution Fitting with Generalized Hyper-Erlang Distributions

Gábor Horváth[1,2,3]

[1] Budapest University of Technology and Economics
Department of Networked Systems and Services
[2] MTA-BME Information systems research group
Magyar Tudósok krt. 2, 1117 Budapest, Hungary
[3] Inter-University Center of Telecommunications and Informatics
ghorvath@hit.bme.hu

Abstract. This paper describes a novel moment matching based fitting method for phase-type (PH) distributions. A special sub-class of phase-type distributions is introduced for the fitting, called generalized hyper-Erlang distributions. The user has to provide only two parameters: the number of moments to match, and the upper bound for the sum of the multiplicities of the eigenvalues of the distribution, which is related to the maximal size of the resulting PH distribution. Given these two parameters, our method obtains *all* PH distributions that match the target moments and have a Markovian representation up to the given size. From this set of PH distributions the best one can be selected according to any distance function.

1 Introduction

Since their introduction, phase-type (PH) distributions have played an important role in performance and reliability modeling. PH distributions are simple, numerically tractable and easy to integrate into complex stochastic models.

However, the applicability of PH distributions for modeling real systems relies on efficient *fitting procedures*. A fitting procedure constructs a PH distribution based on empirical samples or based on an other known distribution.

A large number of PH fitting procedures have been published in the literature. This paper presents a fitting procedure that is based on moment matching. The moment matching problem of PH distributions can be formulated as the solution of a system of polynomial equations. It is possible, however, that none of the solutions is a valid PH distribution. To overcome this problem we introduce a sub-class of PH distributions, called generalized hyper-Erlang distributions, that can grow in size if the set of moments can not be realized with a given size. It is guaranteed that above a given size the generalized hyper-Erlang distributions can realize any valid moment set, thus any moment set that belongs to a positive distribution.

The rest of the paper is organized as follows. Section 2 introduces phase-type distributions, and shows how a Markovian representation is obtained from

a non-Markovian one. Section 3 provides an overview on the PH fitting methods published in the literature, with an emphasis on the moment matching-based solutions. Section 4 describes generalized hyper-Erlang distributions, and how they are used to fit distributions. Some numerical examples are given in Section 5, and Section 6 concludes the paper.

2 Phase-type distributions

Phase-type distributions are given by two parameters: an initial vector $\alpha = \{\alpha_i, i = 1, \ldots, K\}, \alpha \mathbb{1} = 1$, and a matrix $\boldsymbol{A} = \{q_{ij}, i, j = 1, \ldots, K\}$. The vector-matrix pair $(\alpha, \boldsymbol{A})$ is called the *representation* of the PH distribution, and K is the size of the representation. The probability density function (pdf, denoted by $f(x)$), the cumulative distribution function (cdf, denoted by $F(x)$) and the kth moment of a PH($\alpha, \boldsymbol{A}$) distributed random variable $\mathcal{X}$ are

$$f(x) = \alpha e^{\boldsymbol{A}x}(-\boldsymbol{A})\mathbb{1}, \tag{1}$$

$$F(x) = P(\mathcal{X} < x) = 1 - \alpha e^{\boldsymbol{A}x}\mathbb{1}, \tag{2}$$

$$\mu_k = E(\mathcal{X}^k) = \int_0^\infty x^k f(x)dx = k!\alpha(-\boldsymbol{A})^{-k}\mathbb{1}, \tag{3}$$

where $\mathbb{1}$ is a column vector of appropriate size.

A representation is called a *Markovian representation*, if the entries of the initial vector are probabilities ($0 \leq \alpha_i \leq 1, i = 1, \ldots, K$) and matrix $\boldsymbol{A}$ is a generator of a transient continuous time Markov chain, thus, $q_{ii} < 0$, $q_{ij} \geq 0, \forall i \neq j$, and for the row sum we have that $\sum_{j=1}^K q_{ij} \leq 0, i = 1, \ldots, K$, with at least one state where the row sum is strictly negative. If $(\alpha, \boldsymbol{A})$ is a Markovian representation, then the corresponding PH distribution has a probabilistic interpretation as well: $\mathcal{X}$ is the absorption time of the transient Markov chain with sub-generator $\boldsymbol{A}$ and initial state probability vector α of the non-absorbing states.

The pdf can be expressed in a spectral form as well. Suppose the number of distinct eigenvalues is n_d. Let us denote the eigenvalues by $-\lambda_i$, and their multiplicity by r_i ($\sum_{i=1}^{n_d} r_i = N \leq K$). From (1) we have

$$f(x) = \sum_{i=1}^{n_d} \sum_{j=1}^{r_i} b_{ij} \frac{(\lambda_i x)^{j-1}}{(j-1)!} \lambda_i e^{-\lambda_i x}. \tag{4}$$

Note that if $\lambda_i \in \mathbb{C}\backslash\mathbb{R}$ then $\exists j \neq i : \lambda_j = \bar{\lambda}_i$ ($\bar{\lambda}_i$ denotes the complex conjugate of λ_i). To define a valid distribution $\text{re}\langle\lambda_i\rangle > 0, i = 1, \ldots, n_r$ must hold. Furthermore, as a consequence of the Perron-Frobenius theorem, the dominant eigenvalue (i.e., the eigenvalue with the largest real part) must be real.

When $N = K$, the $(\alpha, \boldsymbol{A})$ representation is called *minimal*. If $N < K$ then matrix $\boldsymbol{A}$ has at least one eigenvalue that does not play a role in the pdf, because the corresponding coefficients are zero.

2.1 Similarity transformation of the representations

The $(\alpha, \boldsymbol{A})$ representation of PH distributions is not unique. According to the following Theorem, different similarity transformations generate different representations of the same distribution.

Theorem 1. *(From [5], Theorem 3.) If there exists a matrix $\boldsymbol{W} \in \mathbb{R}^{N,M}, M \geq N$, such that $\boldsymbol{W}\mathbb{1} = \mathbb{1}$, $\boldsymbol{AW} = \boldsymbol{WB}$ and $\alpha\boldsymbol{W} = \beta$, then $(\alpha, \boldsymbol{A})$ and $(\beta, \boldsymbol{B})$ define the same distribution.*

Transforming a representation $(\alpha, \boldsymbol{A})$ with matrix $\boldsymbol{W}$ can destroy the Markovian property of the representation. In this paper, however, we are using similarity transforms to achieve the opposite effect, thus to find a Markovian representation starting from a non-Markovian one, which may require to inflate to size of the representation $(M > N)$.

2.2 Obtaining a Markovian representation from a non-Markovian one

In [12] a special representation, called *monocyclic representation* is defined, which has an important feature phrased by the following Theorem.

Theorem 2. *[12] Every PH distribution has a Markovian monocyclic representation.*

A monocyclic representation consists of Feedback-Erlang Blocks (FEB) arranged in a row. FEB_i is characterized by a rate parameter ν_i, a size parameter k_i and a feedback probability z_i (see Figure 1, where $k_1 = 1, k_2 = 4, k_3 = 1$).
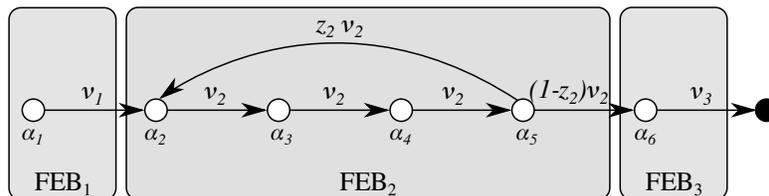

Fig. 1. Monocyclic representation of a PH distribution

[12] provides a constructive algorithm to obtain the Markovian monocyclic representation. We will rely on this result heavily in this paper, as the equations solved to match the moments typically result in a non-Markovian representation.

The transformation of a non-Markovian representation $(\alpha, \boldsymbol{A})$ to a Markovian (monocyclic) representation $(\beta, \boldsymbol{B})$ consists of the following steps ([12]).

1. In the first step matrix $\boldsymbol{B}$ is constructed. Each FEB implements one real eigenvalue or a conjugate complex eigenvalue pair of $\boldsymbol{A}$. Let us denote the jth eigenvalue of $\boldsymbol{A}$ by $-\lambda_j$, or, if it is a complex conjugate eigenvalue pair, by $-\lambda_j = a_j + b_j\mathbb{i}$ and $-\bar{\lambda}_j = a_j - b_j\mathbb{i}$.

- If λ_j is real, the parameters of the jth FEB are $\nu_j = \lambda_j, k_j = 1, z_j = 0$.
- If λ_j is complex, the parameters of the corresponding FEB are determined as

$$k_j = \text{the smallest integer for which } a_j/b_j > \tan(\pi/k_j), \qquad (5)$$

$$\nu_j = \frac{1}{2}\left(2a_j - b_j \tan\frac{\pi}{k_j} + b_j \cot\frac{\pi}{k_j}\right), \qquad (6)$$

$$z_j = \left(1 - \left(a_j - b_j \tan\frac{\pi}{k_j}\right)\right)^{k_j}. \qquad (7)$$

With these parameters matrix $\boldsymbol{B}$ is Markovian by construction and contains all eigenvalues of $\boldsymbol{A}$ with the proper multiplicities. However, the size of matrix $\boldsymbol{B}$ can be larger than the size of matrix $\boldsymbol{A}$, meaning that new eigenvalues are introduced. These extra eigenvalues do not play a role in the pdf, as vector β will be such that they have zero coefficients.

2. The second step of the procedure is calculating the initial vector β. To this end, we need to obtain the transformation matrix $\boldsymbol{W}$ that transforms matrix $\boldsymbol{A}$ to matrix $\boldsymbol{B}$.

   According to Theorem 1 matrix $\boldsymbol{W}$ is the solution of $\boldsymbol{AW} = \boldsymbol{WB}, \boldsymbol{W}\mathbb{1} = \mathbb{1}$, which is a linear system of equations with regards to the entries of $\boldsymbol{W}$. With the presented construction of $\boldsymbol{B}$, this linear system always has a unique solution. If the size of $\boldsymbol{A}$ is N, and the size of $\boldsymbol{B}$ is M, equation $\boldsymbol{AW} = \boldsymbol{WB}$ has $N \times M$ unknowns and defines $N \times M$ equations. However, only $N \times M - N$ equations will be independent, since N eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$ are the same. With $\boldsymbol{W}\mathbb{1} = \mathbb{1}$, however, we get $N \times M$ independent equations and obtain a unique solution.

   The initial vector is then given by $\beta = \alpha\boldsymbol{W}$, which may not be a probability vector.

3. The third, last step is necessary only if vector β is not a proper probability vector. In this case, an Erlang tail needs to be appended to the row of FEBs. This Erlang tail is added to matrix $\boldsymbol{B}$, the corresponding transformation matrix $\boldsymbol{W}$ is calculated, and we get a new initial vector. It is proven in [12] that an appropriate Erlang tail always makes the representation Markovian, if $(\alpha, \boldsymbol{A})$ defines a PH distribution. Unfortunately there is no explicit way to obtain the size and the rate parameter of the Erlang tail. We apply a simple heuristic that increases the size of the Erlang tail successively and applies the secant method to find its rate parameter such that β is a probability vector.

3   Fitting methods for Phase-type distributions

In this section we give a short overview on some PH fitting procedures published in the past. This overview will not be exhaustive, and concentrates on the moment matching based solutions, as the fitting method presented in this paper belongs to this category as well.

Generally speaking, PH distribution fitting means to find a vector α and a matrix A such that the corresponding PH distribution is as close to the target distribution as possible. The most popular methods can be classified as follows.

- *Expectation-maximization (EM) based methods.* These iterative methods provide a PH distribution from a measurement trace (given by samples) by increasing the likelihood in each iteration. Some variants operate on the whole class of PH distributions ([1] and [13]), some others restrict themselves to a specific sub-class like hyper-exponential distributions ([7]) or hyper-Erlang distributions ([15]). The EM algorithm based methods have the advantage that they are very user friendly, the only input parameter required for the fitting is usually only the number of states of the resulting PH distribution. The execution time, however, increases with the length of the measurement trace, furthermore, these methods perform a local maximization and possibly can not find the global optimum.
- *Other optimization based methods.* Some PH fitting methods formulate and solve the PH fitting as a non-linear optimization problem. For instance, [9] applies sequential linear programming to optimize the parameters of an acyclic PH distribution according to a user-selectable distance function (some possible distance functions can be found in [3]). The optimization can take a long time, and there is no guarantee that the global optimum is found.
- *Matching based methods.* Matching based methods intend to match some statistical quantities of the target distribution exactly. In [8] a hyper-exponential distribution is constructed such that its pdf matches the pdf of the target distribution at selected points. The point selection needs to be done by the user by manual tuning, and the resulting distribution might not be a valid distribution in some cases. The moment matching based methods fall into this category as well, that we discuss in more details soon.

In the practice, the selection between the large number of PH fitting methods depends on many factors. Typically several methods are used and the one that gives the best result is selected.

3.1 PH distribution fitting based on moment matching

Fitting distributions based on the moments has some distinct advantages:

- The moments of random variables are easy to obtain. This holds for measurement and for simulation as well. When only the moments are required, there is no need to collect and store all the samples of the observed variable, moments can be calculated in a progressive way with negligible memory and computation effort.
- Some performance measures depend only on the moments of some random variables, thus it makes sense to capture them accurately. For instance, the mean waiting time in an M/G/1 queue depends only on the first two moments of the service time distribution.

On the other hand, there are some drawbacks as well:

– Solving the equations to match the moments explicitly can be difficult if the number of moments to match is high, or if the structure of the PH is not simple enough.
– There are heavy tailed distributions with infinite moments. Moment matching can not be applied to fit such distributions.

There are a number of moment matching methods available in the literature. Based on how flexible the structure of the PH distribution is, we can distinguish between *fixed structure* and *flexible structure* methods.

– *Fixed structure* methods use a PH representation in which the number of parameters equals the number of moments to match. In [14] a PH of size 2 is used to match 3 moments, while [10] is able to match 5 moments with 3 states (it is known that the canonical form of 2-state PH distributions have 3, the one of 3-state PH distributions have 5 parameters). In [6] a hyper-exponential distribution of size N is obtained based on $2N - 1$ moments. Methods belonging to this category have the drawback that only those moments can be matched that fall into the region of feasible moments of the applied PH distribution. The region of feasible moments can be quite restrictive in the practice. Suppose we have a distribution whose first 3 moments can not be matched with a 2-state PH distribution. One would think that increasing the size of the PH can help, and there will be a better chance that a 3-state PH distribution will match the required moments. Actually the opposite is true. The contradiction is, that more (namely 5) moments need to be matched to obtain a 3-state PH distribution. The more moments we have to match, the larger the chance is that at least one of them lies outside of the feasible region.
– *Flexible structure* methods use PH representations that have *more* parameters than the number of moments to match. A set of parameters are set to ensure the matching of moments, while the remaining parameters add some extra degrees of freedom to obtain a valid PH distribution to match any set of moments. Such a method is described in [11], which uses a representation called *mixture of Erlang distributions of common order* (MECO). Mixing N Erlang distributions of common order has $2N$ parameters: $N - 1$ initial probabilities (which Erlang to choose, sums up to 1), the intensity parameters of the Erlang distributions (there are N of them), and the common order of the Erlang distributions (1 parameter). With these $2N$ parameters this procedure can match $2N - 1$ moments. The free parameter not involved in moment matching is the order parameter. The moment matching is performed with order=1, order=2, etc., the order is increased till a Markovian solution is found. It is guaranteed that, with appropriately large order, this procedure is able to match any set of $2N - 1$ moments belonging to a positive distribution. The other method operating on a flexible structure has been published in [2]. It matches 3 moments with an exponential and an Erlang distribution connected after each other in a row. The degree of freedom is the order of the Erlang component again. It is proven that, by choosing the

order of the Erlang component appropriately large, it is possible to match any 3 moments with this structure.

As this quick summary suggests, moment matching methods having a *flexible structure* are very convenient to use in the practice. The user just has to enter the moments, and these procedures find the appropriate PH distribution of appropriate size automatically. The method presented in this paper falls into this category as well.

4 Moment matching with generalized hyper-Erlang distributions

In this section we propose a new sub-class of PH distributions for matching moments.

Definition 1. *Random variable $\mathcal{X}$ has order-N generalized hyper-Erlang distribution iff its density function is*

$$f(x) = \sum_{i=1}^{N} \alpha_i \frac{(\lambda_i x)^{r_i - 1}}{(r_i - 1)!} \lambda_i e^{-\lambda_i x}, \tag{8}$$

with $f(x) \geq 0$ and $\int_0^\infty f(x)dx = 1$. For the parameters we have that $\lambda_i \in \mathbb{C}, re\langle \lambda_j \rangle \geq 0, \alpha_j \in \mathbb{C}, \sum_{i=1}^{N} \alpha_i = 1$ and $r_i \in \mathbb{N}$ for $i = 1, \ldots, N$.

Thus, generalized hyper-Erlang distributions (GHErD) are similar to hyper-Erlang distributions, the difference is that coefficients α_i do not need to be valid probabilities, and that λ_i can be complex as well. The kth moment of generalized hyper-Erlang distributions is calculated as

$$\mu_k = \int_0^\infty x^k f(x)dx = \sum_{i=1}^{N} \alpha_i \frac{(k + r_i - 1)!}{(r_i - 1)!} \frac{1}{\lambda_i^k}, \quad k \geq 0. \tag{9}$$

(Note that $\mu_0 = 1$).

4.1 Solution of the moment matching problem when the r_i parameters are fixed

For matching moments $\mu_1, \ldots, \mu_{2N-1}$ with order-N GHErD having the r_i parameters fixed we have to solve a system of polynomial equations defined by (9) for $k = 0, \ldots, 2N-1$, such that the unknown variables are $\lambda = \{\lambda_i, i = 1, \ldots, N\}$ and $\sigma = \{\alpha_i, i = 1, \ldots, N\}$, which give $2N$ unknowns in total.

Due to the structure of the system of polynomial equations we were not able to derive an explicit solution for arbitrary N. However, there are excellent tools available that are able to solve polynomial systems numerically. For this purpose, we are using PHCpack (see [16]), which is a multi-platform open-source

tool and is under continuous development and refinement. We would emphasize that this does not mean that we are applying a non-linear programming or other optimization methods to find the solution of the moment matching problem (as it is done in [4]). What we are doing is the numerical solution of the polynomial system, that is able to provide *all* the solutions of the system of polynomial equations.

This polynomial system has typically several solutions, and it is also possible that it has no solutions at all (it is inconsistent). If it does have solutions, each solution either defines a valid PH distribution, or it does not. To decide if a solution is valid, we try to obtain a Markovian representation by using the method described in Section 2.2. From a solution given by vectors λ and σ the initial (non-Markovian) representation is obtained in a direct way as

$$\alpha = \begin{bmatrix} \alpha_1 & \underbrace{0 \ \ldots \ 0}_{r_1-1} & \alpha_2 & \underbrace{0 \ \ldots \ 0}_{r_2-1} & \alpha_N & \underbrace{0 \ \ldots \ 0}_{r_N-1} \end{bmatrix},$$

$$\boldsymbol{A} = \begin{bmatrix} \begin{array}{ccc|cc|c|cc} -\lambda_1 & \lambda_1 & & & & & & \\ & \ddots & \ddots & & & & & \\ & & -\lambda_1 & & & & & \\ \hline & & & -\lambda_2 & \lambda_2 & & & \\ & & & & \ddots & \ddots & & \\ & & & & & -\lambda_2 & & \\ \hline & & & & & \ddots & & \\ \hline & & & & & & -\lambda_N & \lambda_N \\ & & & & & & & \ddots & \ddots \\ & & & & & & & & -\lambda_N \end{array} \end{bmatrix} \cdot \left.\begin{array}{c}\\\\\\\end{array}\right\} r_1 \quad \left.\begin{array}{c}\\\\\\\end{array}\right\} r_2 \quad \left.\begin{array}{c}\\\\\\\end{array}\right\} r_N \qquad (10)$$

If the output $(\beta, \boldsymbol{B})$ of the algorithm of Section 2.2 is Markovian, then we found a valid solution. Given vector r it may happen that several different PH distributions are found, but it is also possible that no solutions exist. In the latter case the entries of vector r need to be increased to obtain a valid solution.

4.2 Optimizing the r_i parameters

Finding the appropriate vector r can be made automatic as well. In this case the user just has to enter a single parameter, R, and the algorithm repeats the moment matching with all vectors r satisfying $\sum_{i=1}^{N} r_i \leq R$.

As the MECO is a sub-class of GHErD, furthermore, according to [11] it is always possible to find a MECO for any set of moments, our procedure always finds a Markovian solution (with an appropriately large R parameter).

Our algorithm is depicted in Figure 2. First, the algorithm solves the moment matching problem with different r vectors up to $\sum_{i=1}^{N} r_i \leq R$ and collects all solutions that have a Markovian representation in set *res*. Notice that all solutions in *res* match the first $2N - 1$ moments of the target distribution. In

```
 1: procedure FITGHERD($\mu_1, \ldots, \mu_{2N-1}, R, D(\cdot)$)
 2:     $res \leftarrow \{\}$
 3:     for all $r = \begin{bmatrix} r_1 \ \ldots \ r_N \end{bmatrix}$ with $\sum_{i=1}^{N} r_i \leq R$ do
 4:         $\{(\lambda, \sigma)_i\} \leftarrow$ Solve polynomial equations of (9) for $k = 0, \ldots, 2N - 1$
 5:         for each solution $(\lambda, \sigma)$ do
 6:             $(\alpha, \boldsymbol{A}) \leftarrow$ Create non-Markovian representation based on (10)
 7:             $(\beta, \boldsymbol{B}) \leftarrow$ Transform $(\alpha, \boldsymbol{A})$ to a Markovian representation (Section 2.2)
 8:             if $(\beta, \boldsymbol{B})$ is Markovian then
 9:                 Add $(\beta, \boldsymbol{B})$ to set $res$
10:             end if
11:         end for
12:     end for
13:     if $res$ is $\{\}$ then
14:         error "No solutions found up to size $R$. Parameter $R$ has to be increased."
15:     end if
16:     $(\gamma, \boldsymbol{G}) \leftarrow \arg\min_{(\beta, \boldsymbol{B}) \in res} D(\beta, \boldsymbol{B})$
17:     return $(\gamma, \boldsymbol{G})$
18: end procedure
```

Fig. 2. GHErD fitting algorithm based on moment matching

the second step (last line of the algorithm) the best solution is selected by a distance function $D(\cdot)$. Any distance function can be used that quantifies the distance between two distributions. Two possible distance functions are:

– Moment distance (MD): the sum of the squared relative difference of the moments up to moment K. Denoting the kth moment of the target distribution by $\hat{\mu}_k$ this means

$$D(\beta, \boldsymbol{B}) = \sum_{k=1}^{K} \left(\frac{\mu_k^{(\beta, \boldsymbol{B})} - \hat{\mu}_k}{\hat{\mu}_k} \right)^2. \tag{11}$$

  (Note that for $k \leq 2N - 1$ we have $\mu_k^{(\beta, \boldsymbol{B})} = \hat{\mu}_k$).
– Relative entropy (RE): the relative entropy has been introduced in [3] as a measure of goodness of the approximation. By denoting the pdf of the target distribution by $\hat{f}(x)$ it is

$$D(\beta, \boldsymbol{B}) = \int_0^\infty f(x) |\log \frac{f(x)}{\hat{f}(x)}| dx, \tag{12}$$

  where $f(x) = \beta e^{\boldsymbol{B}x}(-\boldsymbol{B})\mathbb{1}$.

Both distance functions have their advantage. The moment distance still relies only on the moments, thus the exact shape of the target distribution (eg. the pdf) is not required. The relative entropy, however, may quantify the similarity of the shape of the density functions better. Both distance functions will be evaluated in the subsequent numerical examples.

5 Numerical examples

In this section we apply the presented fitting method on two well known traffic measurement traces, the BC-pAug89 and the LBL-TCP-3 traces[4]. The former one records one million packet arrivals on an Ethernet network, while the latter one captures two hours of wide-area TCP traffic. These traces are too old to be representative for the network traffic in these days, the only reason we selected them is that they both are frequently used as demonstration of various traffic fitting algorithms, thus they serve as a kind of benchmark.

Our method has been implemented in MATLAB. To solve the system of polynomial equations we used PHCpack v2.3.76[5], that has a MATLAB interface as well. All the results have been calculated on an average PC with a CPU clocked at 3.4 GHz and 4 GB of memory.

In general, our experience with PHCpack is positive. We got some warnings that the polynomial system is ill-conditioned initially, but normalizing the trace (such that the mean value is one) solved this issue. The solution time of the polynomial system depends on N heavily. We got prompt results in case of $N = 2$, it took about $0.5s$ in case of $N = 3$, and $25s$ in case of $N = 4$. As our method relies on the iterative solution of a large number of polynomial equations, we did not increase parameter N above 4, which means that we are going to match at most 7 moments in the numerical examples. When N is large, the number of solutions is large as well, however trying to create the corresponding Markovian representations is such fast that is has negligible effect on the execution time. The total execution time of the algorithm depends on parameter R as well. If R is large, a large number of r vectors are considered, and a large number of polynomial systems are solved.

In the subsequent case studies we compare the following 4 moment matching-based PH fitting methods:

1. Our method, where the PH distribution with the smallest moment distance is selected from the set of all PH distributions matching the moments of the trace;
2. Our method, where the PH distribution with the smallest relative entropy is selected from the set of all PH distributions matching the moments of the trace;
3. Moment matching with a mixture of Erlang distributions of common order (MECO, [11]);
4. The method of [2], which is able to match the first three moments only.

5.1 The LBL trace

In this example the R parameter (the sum of the multiplicities of the eigenvalues) is set to 20. According to the algorithm in Figure 2 this means that the moment

---

[4] Downloaded from http://ita.ee.lbl.gov/html/contrib/BC.html and from http://ita.ee.lbl.gov/html/contrib/LBL-TCP-3.html, respectively

[5] It can be obtained from http://homepages.math.uic.edu/ jan/download.html

matching is performed with all vectors r satisfying $\sum_{i=1}^{N} r_i \leq R$. With a given vector r, the moment matching problem can result in several different valid PH distributions. At the end we have a large number of PH distributions from which we can select the best according to some distance function. Table 1 shows how many r vectors and valid solutions there are, and how long the execution time of the algorithm is.

Table 1. The number of different r vectors, the number of valid solutions, and the total execution time in the LBL example

# of moments to match	# of different r vectors	# of valid solutions	Execution speed
3	100	88	28 sec
5	237	688	337 sec
7	408	3920	810 min

The numerical results are shown in Table 2. When the first three moments are matched, all methods found the same solution. Even if a large number of r vectors have been checked by our method, the best solution has been found to be the same according to both distance functions.

When 5 moments are matched, the MECO matching method returned a hyper-exponential distribution that was found to be optimal by our method as well according to the distance of moments. However, our method was able to find a PH distribution that has lower relative entropy. This PH distribution has 5 states and the corresponding r vector is $r = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix}$. The sum of the elements of vector r is only 4, which means that a new eigenvalue has been introduced and the size of the PH has been increased by 1 to obtain a Markovian representation (this new eigenvalue cancels out, it has no effect on the pdf).

The advantage and the flexibility of our method can be seen the best when 7 moments are matched. Our method was able to find a PH distribution with significantly lower moment distance, and an other one with significantly lower relative entropy as well.

Figures 3, 4 and 5 plot the density functions belonging to the methods discussed, both on linear and on logarithmic scale. While the tail of the pdf is fitted well by all methods, the plots differ significantly in the body of the pdf. Based on a visual comparison, the solution found by our method by matching 7 moments and selecting the best according to the RE distance seems to capture the shape of the pdf best.

5.2 The BC trace

The numerical results corresponding to the fitting of the BC trace are summarized in Table 3. When fitting the first 3 moments, the same hyper-exponential

Table 2. Results of the fitting of the LBL trace

Num. of moments	Method	MD	RE	Num. of states
	Our method (MD)	1.786	0.3024	$2\ (r = \begin{bmatrix} 1 & 1 \end{bmatrix})$
3	Our method (RE)	1.786	0.3024	$2\ (r = \begin{bmatrix} 1 & 1 \end{bmatrix})$
	MECO [11]	1.786	0.3024	$2\ (r = \begin{bmatrix} 1 & 1 \end{bmatrix})$
	ErlExp [2]	1.786	0.3024	2
	Our method (MD)	0.0072	0.0984	$3\ (r = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix})$
5	Our method (RE)	0.0386	0.0953	$5\ (r = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix})$
	MECO [11]	0.0072	0.0984	$3\ (r = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix})$
	Our method (MD)	8.26×10^{-6}	3.9727	$20\ (r = \begin{bmatrix} 2 & 2 & 3 & 13 \end{bmatrix})$
7	Our method (RE)	0.00499	0.0727	$8\ (r = \begin{bmatrix} 1 & 1 & 3 & 3 \end{bmatrix})$
	MECO [11]	0.00475	0.1339	$8\ (r = \begin{bmatrix} 2 & 2 & 2 & 2 \end{bmatrix})$
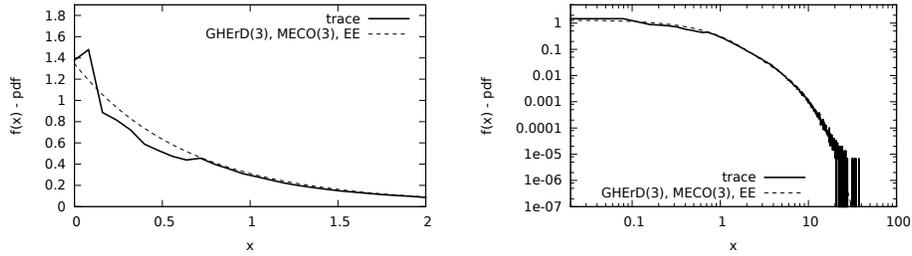
Fig. 3. Comparison of the density functions matching 3 moments of the LBL trace
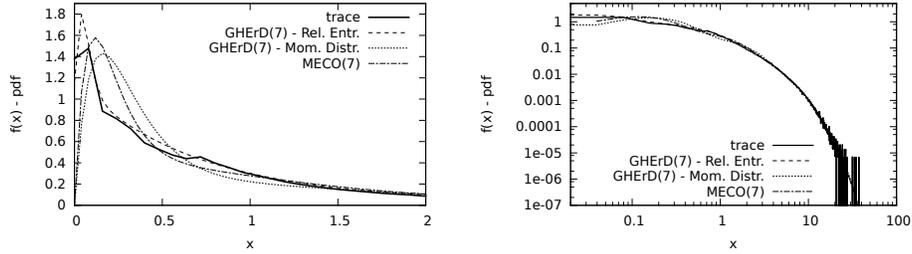
Fig. 4. Comparison of the density functions matching 7 moments of the LBL trace
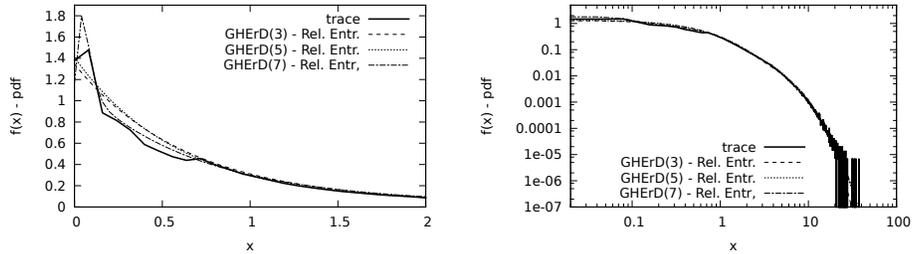

Fig. 5. Comparison of the results of our method, matching 3, 5 and 7 moments

distribution turned out to be optimal by all the methods involved into the comparison. When fitting 5 phases, our method has found a PH distribution with slightly lower relative entropy. In the case when 7 moments are matched, we have a PH distribution with better moment distance, and an other one with significantly better relative entropy than the MECO based method.

Table 3. Results of the fitting of the BC trace

Num. of moments	Method	MD	RE	Num. of states
3	Our method (MD)	3.0509	0.30244	$2\ (r = \begin{bmatrix} 1 & 1 \end{bmatrix})$
	Our method (RE)	3.0509	0.30244	$2\ (r = \begin{bmatrix} 1 & 1 \end{bmatrix})$
	MECO [11]	3.0509	0.30244	$2\ (r = \begin{bmatrix} 1 & 1 \end{bmatrix})$
	ErlExp [2]	3.0509	0.30244	2
5	Our method (MD)	0.00198	0.30521	$14\ (r = \begin{bmatrix} 1 & 4 & 8 \end{bmatrix})$
	Our method (RE)	55.4699	0.30212	$5\ (r = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix})$
	MECO [11]	55.4699	0.30212	$3\ (r = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix})$
7	Our method (MD)	0.0056	0.48178	$20\ (r = \begin{bmatrix} 1 & 2 & 2 & 15 \end{bmatrix})$
	Our method (RE)	0.0072	0.185	$19\ (r = \begin{bmatrix} 1 & 1 & 2 & 15 \end{bmatrix})$
	MECO [11]	0.0391	0.3536	$16\ (r = \begin{bmatrix} 4 & 4 & 4 & 4 \end{bmatrix})$

The density functions corresponding to the investigated cases are depicted in Figure 6, 7 and 8. Figure 7 demonstrates how different the shapes of the density functions can be even if the first 7 moments are the same. The MECO-based method looks to be the least successful in this example, while our method with the relative entropy based selection managed to capture the characteristics of the density function reasonably well.

6   Conclusion

This paper describes a unique approach to PH distribution fitting. The presented method is based on moment matching, and allows to use an arbitrary distance function to select the best solution from the ones matching the target moments.

Two case studies are presented to demonstrate the behavior and the capabilities of the procedure when fitting real measurement traces.

The weak point of the procedure is the numerical solution of the polynomial system, which limits the number of moments to match.

Acknowledgment
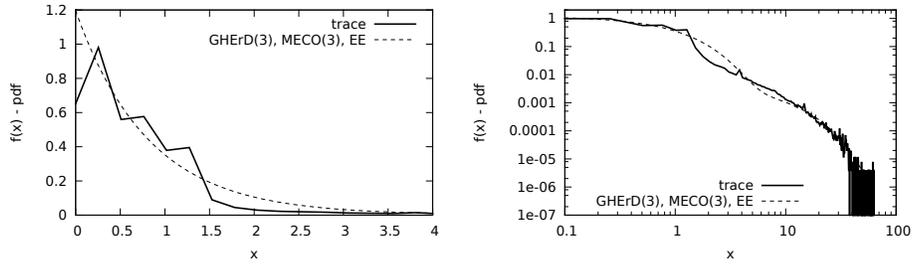
**Fig. 6.** Comparison of the density functions matching 3 moments of the BC trace
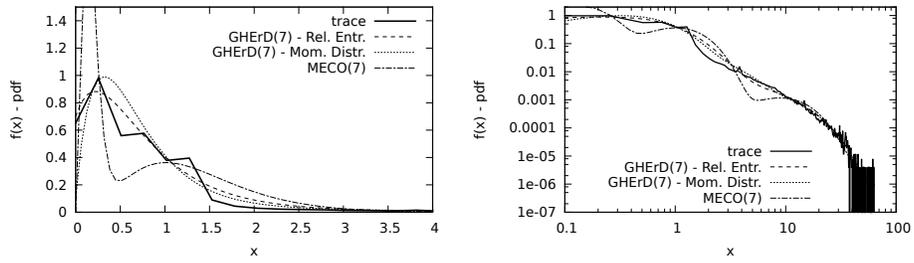
**Fig. 7.** Comparison of the density functions matching 7 moments of the BC trace
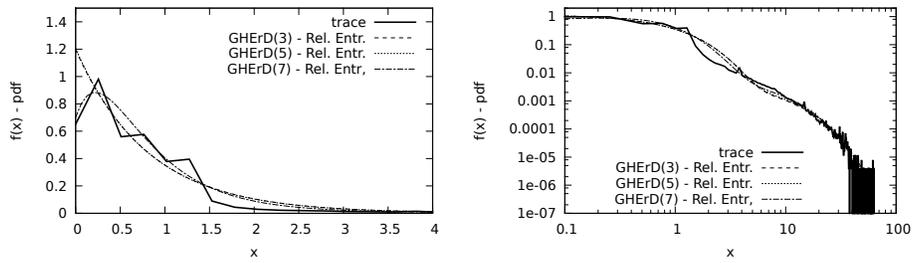
Fig. 8. Comparison of the results of our method, matching 3, 5 and 7 moments

References

1. S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, pages 419–441, 1996.
2. A. Bobbio, A. Horváth, and M. Telek. Matching three moments with minimal acyclic phase type distributions. *Stochastic Models*, 21(2-3):303–326, 2005.
3. A. Bobbio and M. Telek. A benchmark for PH estimation algorithms: results for Acyclic-PH. *Stochastic Models*, 10(3):661–677, 1994.
4. Peter Buchholz, Peter Kemper, and Jan Kriege. Multi-class Markovian arrival processes and their parameter fitting. *Performance Evaluation*, 67(11):1092–1106, 2010.
5. Peter Buchholz and Miklós Telek. Stochastic Petri nets with matrix exponentially distributed firing times. *Performance Evaluation*, 67(12):1373–1385, 2010.
6. G. Casale, E.Z. Zhang, and E. Smirni. Interarrival times characterization and fitting for markovian traffic analysis. *Numerical Methods for Structured Markov Chains*, 7461, 2008.
7. R. El Abdouni Khayari, R. Sadre, and B.R. Haverkort. Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation*, 52(2):175–191, 2003.
8. A. Feldmann and W. Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 31(3):245–279, 1998.
9. A. Horváth and M. Telek. Phfit: A general phase-type fitting tool. *Computer Performance Evaluation: Modelling Techniques and Tools*, pages 1–14, 2002.
10. G. Horváth and M. Telek. On the canonical representation of phase type distributions. *Performance Evaluation*, 66(8):396–409, 2009.
11. M.A. Johnson and M.R. Taaffe. Matching moments to phase distributions: Mixtures of Erlang distributions of common order. *Stochastic Models*, 5(4):711–743, 1989.
12. Ş. Mocanu and C. Commault. Sparse representations of phase-type distributions. *Stochastic Models*, 15(4):759–778, 1999.
13. H. Okamura, T. Dohi, and K.S. Trivedi. A refined EM algorithm for PH distributions. *Performance Evaluation*, 68(10):938–954, 2011.
14. M. Telek and A. Heindl. Matching moments for acyclic discrete and continuous phase-type distributions of second order. *International Journal of Simulation Systems, Science & Technology*, 3(3-4), 2002.
15. A. Thummler, P. Buchholz, and M. Telek. A novel approach for fitting probability distributions to real trace data with the EM algorithm. In *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, pages 712–721. IEEE, 2005.
16. J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)*, 25(2):251–276, 1999.