BuTools 2: a Rich Toolbox for Markovian Performance Evaluation

[Tool Paper]

Gábor Horváth
Budapest Univ. of Technology and Economics
Department of Networked Systems and Services
ghorvath@hit.bme.hu

Miklós Telek
MTA-BME Information Systems Research Group
Magyar Tudósok krt. 2, 1117 Budapest, Hungary
telek@hit.bme.hu

ABSTRACT

BuTools 2 is collection of computational methods that are useful for Markovian and non-Markovian matrix analytic performance analysis. It consists of various packages. There are packages to obtain, analyze, transform and minimize discrete and continuous time phase-type (PH) distributions and Markovian arrival processes (MAP); to fit empirical measurement data and to evaluate the result; to solve many performance measures of various Markovian queueing systems; and to solve block-structured Markov chains. All three major mathematical frameworks are supported: BuTools is released for MATLAB, Mathematica and NumPy/IPython as well, with the same features, with the same call interfaces. Every function is documented, the documentation is supplemented by many examples and the related citations. BuTools uses the state-of-the art algorithms and apart of the basic functionalities it contains several unique, difficult to implement procedures as well.

1. INTRODUCTION

Most researchers have their own set of tools that they use for the every day research activity. Collaboration between researchers can sometimes be difficult because everybody uses his/her own set of tools, and because everybody has his/her own preference of mathematical framework or programming language.

The authors of this paper had faced the same problem several years ago. They were working on very similar area, but the collaboration was difficult because everybody was sticking to his own set of tools. To address this issue, the first version of BuTools has been released in 2012 with the contribution of many colleges and students[1]. BuTools turned out to be very useful, but had some drawbacks: the source code quality was hot homogeneous, and the feature parity between the three supported mathematical environments was only partial.

The aim of the second version was to address these issues. Almost every function has been rewritten from the ground up with efficiency and usability being the first priorities, and has been supplemented by unit tests. A special framework has been developed to generate the documentation, the examples, and the test scripts for the three supported environments automatically from a common source.

The second version has been finalized in September, 2015, and only very small changes were made since then. The homepage of BuTools is http://webspn.hit.bme.hu/~butools, and the source code repository is located at https://github.com/ghorvath78/butools. BuTools V2 is being used by our research group in the every day work with satisfaction. The goal of this paper is to introduce this toolbox and demonstrate its capabilities in the hope that others find it useful as well and make the results presented in the related literature easily accessible for practical computations.

2. INSTALLATION AND BASIC CONCEPTS

BuTools is portable, no installation is needed. The packages of BuTools can be loaded individually, but there are convenient functions available to load every package in a single step as well. If BuTools is located in directory `<BTDir>`, all BuTools packages can be loaded by

- `run('<BTDir>/Matlab/BuToolsInit.m')` in Matlab,
- `%run "<BTDir>/Python/BuToolsInit"` in an IPython console,
- `AppendTo[$Path,"<BTDir>/Mathematica"];` `<<BuTools`` in Mathematica.

There are three global variables used by BuTools, summarized by Table 1.

Setting `verbose` to True allows the functions to print as many useful messages to the output console as possible. Turning it off avoids bloating the console. The default value is False, but for the examples of the reference documentation we have set it to True.

[1] This version of BuTools was availabel on the internet and announced through some professional mailing lists, but never got published as a tool paper. A non exhaustive list of contributors include: Levente Bodrog, Peter Buchholz, Armin Heindl, András Horváth, István Kolossváry, András Mészáros, Zoltán Németh, János Papp Philipp Reinecke, Miklós Vécsei.

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Name in MATLAB	Name in Mathematica	Name in Python	Default value
`BuToolsVerbose`	`BuTools‘Verbose`	`butools.verbose`	False
`BuToolsCheckInput`	`BuTools‘CheckInput`	`butools.checkInput`	True
`BuToolsCheckPrecision`	`BuTools‘CheckPrecision`	`butools.checkPrecision`	10^{-12}

Table 1: Global variables in BuTools

If `checkInput` is set to True, the functions of BuTools perform as many checks on the input parameters as possible. This can be very useful to recognize typos as soon as possible, but can be a waste of computational effort in case of a computationally demanding application.

The `checkPrecision` serves as the tolerance when the validity of the input parameters are checked.

3. WORKING WITH PH DISTRIBUTIONS

Continuous time phase-type (PH) distributions [15] are characterized by two parameters, the initial probability vector α and the transient generator matrix of a continuous time (transient) Markov chain. The PH distribution represents the absorption time of this transient Markov chain starting from α. The cumulative distribution function (cdf) is $F_{PH}(t) = 1 - \alpha e^{\mathbf{A}t}\mathbb{1}$, where $\mathbb{1}$ is the column vector of ones.

Matrix exponential (ME) distributions [2] are the generalizations of PH distributions. Formally, the cdf is $F_{ME}(t) = 1 - be^{\mathbf{B}t}\, e$ and all further formulas for the statistical quantities are very similar to the ones of PH distributions, however, $b, \mathbf{B}$ and e can hold general numbers, the entries do not have to be valid probabilities or transition rates. ME distributions therefore lack the simple stochastic interpretation that PH distributions have. Vector b is called ”starting operator”, matrix $\mathbf{B}$ is the ”process rate operator” and vector e is the ”summing operator”. BuTools uses a special form of ME distributions, where the summing operator is a vector of ones, thus $e = \mathbb{1}$. This is not a restriction, as all ME distributions defined with general summing operator can be easily transformed to this representation [16]. Assume we have an ME distribution in the general form with parameters $b, \mathbf{B}, e$. The necessary similarity transform is obtained by calling the `T = TransformToOnes(e)` procedure of the BuTools reptrans package. The parameters of the ME distribution used by all related BuTools tools can be calculated by $b' = b \cdot \mathbf{T}^{-1}$ and $\mathbf{B}' = \mathbf{T}\mathbf{B}\mathbf{T}^{-1}$.

BuTools provides several tools for both distribution classes in the `ph` package. Of course, functions for obtaining the cdf, the pdf, moments are available as well as functions to check the validity of the representations.

With the inverse characterization tools BuTools can create APH distributions from any 2 moments and from any 3 moments by the `APHFrom2Moments` and `APHFrom3Moments` functions, the size of the necessary representation is determined automatically ([4]). Furthermore, by the `PH2From3Moments` and `PH3From5Moments` functions order 2 and order 3 PH distributions can be obtained from 3 and 5 moments, respectively, if the given moments are feasible with PH(2) and PH(3) distributions ([23],[12]). An interesting procedure is `MEFromMoments` ([26]), that returns an order N ME distribution from any $2N-1$ moments, however, there is no guarantee that the result is a valid ME distribution. To check that the density is non-negative for all points it is possible to call the `CheckMEPositiveDensity` function afterwards (this is a very non-trivial procedure, which relies on the transformation to mono-cyclic representation) [22].

An other category of functions are able to transform between various PH and ME representations. `CanonicalFromPH2` creates an order 2, `CanonicalFromPH3` an order 3 canonical representation from any PH(2) and PH(3) distributions (potentially given by non-Markovian representation). The `PHFromME` function tries to find a PH(N) representation for the given ME(N) one by applying elementary similarity transformations iteratively (note however, that this function is not able to increase the order in the hope for a Markovian representation).

One of the most valuable tool in the `ph` package is the `MonocyclicPHFromME` function, which transforms *any* ME distributions (that fulfil the eigenvalue constraint (eigenvalues with maximal real part are real) and do not touch the x axis apart from the origin) to a PH distribution ([22], [19]). The required size of the representation is determined automatically. The resulting PH distribution is returned in a monocyclic representation.

With a very useful set of functions it is possible to analyze the redundancy of PH distributions and to obtain the minimal representation. `MEOrder` can return the order of the PH/ME based on the analysis of the parameters of the distribution, while `MEOrderFromMoments` returns the ME order necessary to realize the moments given. Several properties of various systems can be characterized through Laplace transform expressions, from which the moments are easy to obtain. From these moments, `MEOrderFromMoments` can tell if there is a matrix-exponential-like (ME-like) behavior in the background, and if the answer is yes, what is the order of that ME distribution. Function `MinimalRepFromME` gives the minimal representation of the given ME distribution, thus the ME distribution returned is the same as the input, but can be smaller.

The `dph` package provides similar tools for the discrete counterparts of PH and ME distributions, the discrete PH (DPH) distributions and matrix-geometric (MG) distributions. The basic set of functions to obtain the moments, the probability mass function and the cdf are available, of course, however, the set of inverse characterization and representation transformation tools are less comprehensive than in the continuous case due to the lack of research results. BuTools can create order-2 and order-3 DPH distributions from 3 and 5 moments (`DPH2From3Moments` and `DPH3From5Moments` based on [21] and [14]), and the moment matching method of [26] has also been adapted to the discrete case (`MGFromMoments`). Unfortunately, flexible order procedures like `APHFrom3Moments` for DPH are not available in the literature yet.

Transforming DPHs to canonical forms for the order-2 and order-3 cases are possible (`CanonicalFromDPH2` and `CanonicalFromDPH3`), and the iterative transformation based `DPHFromMG` is also included, but the discrete equivalent to monocyclic representation is unfortunately unknown, hence we can not transform any MG distribution to DPH yet.

Finally, both the ph and dph packages contain functions to generate random PH and DPH distributions (RandomPH and RandomDPH), and functions to generate random samples from PH and DPH distributions (SamplesFromPH and SamplesFromDPH) for simulation purposes.

4. TOOLS FOR MAPS

Continuous time Markovian arrival processes (MAPs, [15]) are commonly characterized by two matrices, $\mathbf{D_0}$ and $\mathbf{D_1}$. Arrivals by a MAP are modulated by a background continuous time Markov chain with generator $\mathbf{D} = \mathbf{D_0} + \mathbf{D_1}$. Markov chain transitions in $\mathbf{D_0}$ ($\mathbf{D_1}$) do not generate (generate) arrival events. As a result MAPs are capable of generating correlated arrivals.

Rational arrival processes (RAPs, also known as matrix-exponential processes, MEPs) are generalizations of MAPs [1]. Formally, all formulas for the statistical quantities are the same to the ones of MAPs. However, both $\mathbf{D_0}$ and $\mathbf{D_1}$ can hold negative real numbers, the entries do not have to be valid transition rates. RAPs therefore lack the simple stochastic interpretation that MAPs have.

Both MAPs and RAPs can be generalized to multi-type arrival processes. If there are K different arrival types, marked MAPs (MMAPs) and marked RAPs (MRAPs) defined by matrices $\mathbf{D_0}, \dots, \mathbf{D_K}$ are able to describe the multi-type arrival process.

BuTools provides several tools for MAPs, RAPs and their marked variants in the map package.

With the appropriate function BuTools can return basic properties like the marginal distribution (the parameters of the corresponding PH distribution), the marginal moments, the lag auto-correlations, and the lag-k joint moments ([24]) of MAPs and MMAPs.

With the set of inverse characterization tools it is possible to obtain order N RAPs or MAPs from $2N-1$ marginal moments and $(N-1)^2$ lag-1 joint moments (RAPFromMoments and MRAPFromMoments, using the method of [24]); or from $2N-1$ marginal moments and $2N-3$ auto-correlations (RAPFromMomentsAndCorrelations, based on [18]). The method for creating an order-2 MAP from 3 moments and 1 correlation parameter published in [5] is implemented by the MAP2FromMoments function. The only flexible matching procedure (that can adjust the order of the result automatically, based on the input parameters) is MAPFromFewMomentsAndCorrelations, that implements [9].

As for representation transformation, BuTools is able to transform a RAP(2) to a canonical form (CanonicalFromMAP2), transform a MRAP(N) to MMAP(N) (MAPFromRAP and MMAPFromMRAP, by successive similarity transformations, achieving a MAP is not guaranteed), and can minimize a RAP representation with functions MinimalRepFromRAP and MinimalRepFromMRAP ([6]).

The dmap intends to provide the same functionality for discrete time arrival processes (DMAPs and DRAPs) and their marked variants (DMMAPs and DMRAPs), however, several results present for MAPs and RAPs are not available for DMAPs and DRAPs in the literature yet.

From the inverse characterization tools only DRAPFromMoments/DMRAPFromMoments and DMAP2FromMoments are available (see [17] for the latter one).

Both the map and dmap packages contain functions to generate random MAPs, MMAPs, DMAPs, DMMAPs and to generate random samples from these processes.

5. FITTING TOOLS

5.1 The trace package

The ph, dph, map and dmap packages provide several functions to obtain PH distributions and MAPs from statistical quantities, like moments, auto-correlations and joint moments. The trace package has tools to obtain these kinds of quantities from empirical data traces.

The traces are vectors consisting of measurements. After loading them from a file, cdf, pdf, moments, joint moments, and lag-k auto-correlations can be computed by invoking the CdfFromTrace, PdfFromTrace, MarginalMomentsFromTrace, LagkJointMomentsFromTrace and LagCorrelationsFromTrace functions.

Most of these functions can cope with weighted traces as well, where each measurement data is supplemented by a weight.

5.2 Likelihood based fitting

BuTools has a fitting package, that contains two kinds of functions: procedures for likelihood (EM) based fitting, and tools to evaluate the result of the fitting.

The PHFromTrace is the implementation of the G-FIT procedure ([25]) to create a hyper-Erlang distribution by EM-algorithm. G-FIT is one of the best performing PH fitting method at the moment. While MATLAB, Mathematica and Python are known for not being efficient for such computationally demanding algorithms, BuTools has a reasonably fast, vectorized implementation capable of processing traces with millions of data.

The MAPFromTrace function implements [11], which is similar to G-FIT. The MAP it creates has Erlang components, and a switching probability matrix determining the order of these Erlang components providing the subsequent inter-arrival times. Note however, that this fitting procedure is much slower than the one for PH fitting.

The likelihood of a PH distribution or a MAP regarding a trace can be evaluated by LikelihoodFromTrace.

The functions SquaredDifference and RelativeEntropy measure the difference between two vectors (e.g., probability mass functions, lag-k autocorrelations, etc.), while EmpiricalSquaredDifference and EmpiricalRelativeEntropy are the equivalents to be used for continuous quantities (e.g., for pdf or cdf of continuous time PH variables, where they are given by a number of points only).

5.3 Application example

The usage of BuTools for trace fitting is demonstrated in Figure 1. The first line loads a trace file consisting of 1.78 million inter-arrival time samples. The next line calculates the marginal moments of the trace. Then, a PH(3) distribution is created by matching 5 moments, finally a PH(5) is obtained by fitting (G-FIT). The fitting step took 113 seconds on an average PC. After obtaining the PH distributions, the approximations are evaluated. First the moments are compared, then the likelihood.

After the comparison the density functions are obtained and plotted. (In case of the PH distributions the IntervalPdfFromPH function is used, which, instead of evaluating the pdf at the given points, returns the probability of falling

```
>> trace = dlmread('lbltcp3_iat.txt');
>> trmoms = MarginalMomentsFromTrace (trace,5);
>> [alpha3,A3] = PH3From5Moments(trmoms(1:5));
>> [alpha5,A5] = PHFromTrace (trace, 5);
>> disp(trmoms);
            1 2.942 16.84 150.73 1876.8
>> disp(MomentsFromPH(alpha3,A3));
            1 2.942 16.84 150.73 1876.8
>> disp(MomentsFromPH(alpha5,A5,5));
            1 2.8827 15.074 112.2 1062.3
>> disp(LikelihoodFromTrace(trace,alpha3,A3));
     -0.94802
>> disp(LikelihoodFromTrace(trace,alpha5,A5));
      -0.9343
>> [xt,yt] = PdfFromTrace (trace, (1:0.1:3));
>> [xp3,yp3] = IntervalPdfFromPH (alpha3, A3, (1:0.1:3));
>> [xp5,yp5] = IntervalPdfFromPH (alpha5, A5, (1:0.1:3));
>> plot (xt,yt,xp2,yp2,xp3,yp3,xp5,yp5);
```

Figure 1: Application example for fitting

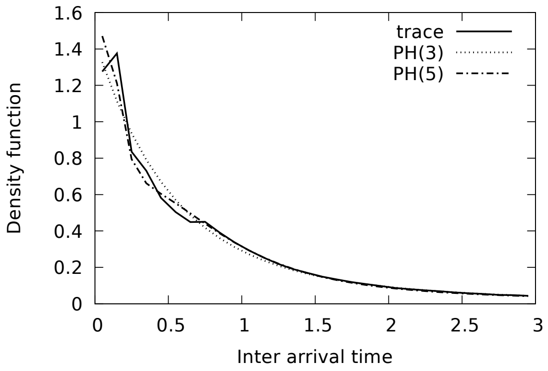

Figure 2: The output of the example of Figure 1

into intervals divided by the interval lengths. This is the correct way to compare it with the empirical pdf of the trace.) The result is depicted in Figure 2.

6. ANALYSIS OF QUEUES

The queues package heavily relies on the matrix-analytic methods.

6.1 Support for Matrix-Analytic Methods

The mam package of BuTools provides solution methods for problems based on non-linear matrix equations.

Functions solving the three diagonal block-structured Markov chains, namely the quasi birth-death processes (QBDs), Markov chains skip-free to the left, M/G/1 type Markov chains, and Markov chains skip-free to the right, G/M/1 type Markov chains ([15]) are included. The underlying algorithms are from the SMCSolver toolbox ([3]). The MATLAB version of BuTools requires the presence of SMCSolver, while the Mathematica and the NumPy/IPython versions include the necessary parts of SMCSolver ported to these environments.

The fundamental matrices for QBDs are returned by the QBDFundamentalMatrices function (matrices $\mathbf{R}, \mathbf{G}$ and $\mathbf{U}$ [15]). Matrix $\mathbf{G}$, the fundamental matrix of M/G/1 type Markov chains is provided by MG1FundamentalMatrix, and the $\mathbf{R}$ matrix of G/M/1 type systems is given by GM1FundamentalMatrix. Based on the fundamental matrices the stationary solutions are provided by functions QBDStationaryDistr, MG1StationaryDistr and GM1StationaryDistr, respectively, that return the stationary probabilities themselves. For QBDs the ingredients of the stationary matrix-geometric solution (the initial vector and the coefficient matrix) are returned by QBDSolve.

BuTools also supports the solution of continuous queueing systems, Markovian fluid flows ([7]). A large part of the related literature considers *canonical* Markovian fluid flows, where the rate at which the fluid level increases or decreases is always 1. For these systems FluidFundamentalMatrices returns the most important matrices, $\mathbf{\Psi}, \mathbf{K}$ and $\mathbf{U}$ [7]. The stationary solution for the fluid level at the requested points is given by FluidStationaryDistr, and the components of the matrix-exponential solution (intial vector, matrix exponent) can be obtained by FluidSolve. For fluid systems, however, we find it more useful to use GeneralFluidSolve, which, based on the generator of the background Markov chain and the diagonal matrix of fluid rates returns the probability mass at level 0, and for positive levels the initial vector, matrix exponent and closing matrix of the matrix-exponential solution.

6.2 Queueing models

Building upon the mam package, the queues package provides functions to obtain many performance measures of several queueing systems. The following queues are supported:

- The MAP/MAP/1 queue (MAPMAP1). Special cases of this queue are the PH/MAP/1, the MAP/PH/1, the PH/PH/1, etc. (see [15] and [8]).
- The QBD queue (QBDQueue). In this queue the arrival and the service process are not independent of each other, they share the same background process. Marked transitions of this background process are accompanied by a level increase, other marked transitions by a level decrease event. The MAP/MAP/1 queue is the special case of the QBD queue, however, several performance measures are more demanding to compute for the QBD queue. The implementation is based on [15] and [20].
- The MMAP[K]/PH[K]/1-FCFS queue (MMAPPH1FCFS). This is a multi-type queue with K types of customers. Each customer type can have a different (PH) service time distribution. The solution is based on [8].
- The MMAP[K]/PH[K]/1 queue with non-preemptive and preemptive resume priority service (MMAPPH1NPPR and MMAPPH1PRPR). This is a multi-type queue with preemptive service, the efficient solution is based on a recent result [10].
- The fluid queue (FluidQueue). In this queue there is a common background Markov chain, a diagonal matrix of fluid arrival rates and a diagonal matrix of fluid service rates in each state of the background process. The queue length and the sojourn time of the fluid drops are the two most interesting performance measures (based on [7] and [13]). This is the continuous counterpart of the QBD queues.
- The Flu/Flu queue (FluFluQueue), which is similar to the ordinary fluid queue, but the fluid input and

output processes are independent, they are modulated by two separate background Markov chains. This independence is exploited in the solution, thus Flu/Flu queues are easier and faster to solve than the general fluid queues. This is the continuous counterpart of the MAP/MAP/1 queues.

The performance measures that can be obtained from these queues are summarized by Table 2. The abbreviation of the performance measures are

- `ncMoms/flMoms`: Stationary moments of the number of customers (in case of discrete queues) / fluid level (in case of fluid queues).
- `ncDistr/flDistr`: Stationary distribution of the number of customers / the fluid level.
- `ncDistrMG/flDistrME`: The parameters of the MG/ME distribution of the number of customers / fluid level.
- `ncDistrDPH/flDistrPH`: The parameters of the DPH/PH distribution of the number of customers / fluid level. The DPH/PH representation is obtained from the MG/ME one, which is always possible. However, in some rare cases (when a phase has a very low probability) this transformation can introduce numerical errors, hence `ncDistrMG/flDistrME` are safer to use.
- `stMoms`, `stDistr`, `stDistrME`, `stDistrPH`: the same as above for the sojourn time of the customers/fluid drops. Again, `stDistrME` behaves better numerically than `stDistrPH`.

When calling these functions, the performance measures to compute are listed in the function arguments. Several performance measures can be computed at the same function call, and BuTools will save as much computational effort as possible by avoiding repeated re-computation of some demanding steps.

6.3 Application example

The example in Figure 3 demonstrates how well the different packages of BuTools play nicely together. First the arrival process is created based on three moments and the lag-1 auto-correlation, then the service time distribution is obtained by matching two moments. The next two lines calculate the $\mathbf{S_0}$ and $\mathbf{S_1}$ matrices representing the service MAP of the queue. All the performance measures are obtained by the same function call. According to the function arguments, 3 moments of the number of customers, the distribution of the number of customers up to 20, and the PH representation of the sojourn time are requested, and the `MAPMAP1` function returns them in the same order. After that, the solution is displayed either on the screen or in a plot (the mean number of customers and the mean sojourn times are equal in this example since the mean arrival rate is 1).

7. SOME FURTHER, SMALL PACKAGES

7.1 The `moments` package

Several moment expressions are being used in various publications related to PH distributions. Most works rely on the ordinary (raw) moments, while others are able to arrive to simpler formulas by introducing some alternative moment expressions, like reduced moments, normalized moments, Hankel moments. Furthermore, in case of discrete systems transform domain techniques make it easy to obtain the factorial moments. The `moments` package provides conversion routines between these moment expressions. Additionally, it provides the `CheckMoments` function to determine if a sequence of real numbers is a valid moment sequence (there exists a distribution with the given moments), or not.

7.2 The `mc` package

A couple of basic functions to obtain the stationary distribution of Markov chains is mandatory in a tool devoted to Markovian performance analysis. The `CTMCSolve` and `DTMCSolve` functions return the stationary distribution of continuous and discrete time Markov chains, respectively, based on the direct (non-iterative) solution of the corresponding set of linear equations.

Two further functions, `CRPSolve` and `DRPSolve` provide the same functionality on rational processes, which are similar to Markov chains without the restrictions on the sign of the elements of the generator matrices. These functions are mostly used internally by the procedures operating on ME distributions and RAPs.

8. CONCLUSION

BuTools collects the implementations of many research results related to stochastic models with Markov background process. This way it makes several complex research results of the field easily accessible for practical application. BuTools is heavily used by our research group and found to be efficient for practical computations. The reader is encouraged to check and use BuTools, which is facilitated with online documentation and application demo.

The authors would be glad to receive any related ideas, comments, feature requests or bug fixes.

9. REFERENCES

[1] S. Asmussen and M. Bladt. Point processes with finite-dimensional probabilities. *Stochastic Processes and their Applications*, 82(1):127Ű142, 1999.

[2] Soren Asmussen and Colm Art OŠcinneide. *Matrix-Exponential Distributions*. John Wiley Sons, Inc., 2004.

[3] D Bini, B Meini, S Steffé, and B Van Houdt. Structured markov chains solver: software tools. In *Proceeding from the 2006 workshop on Tools for solving structured Markov chains*, page 14. ACM, 2006.

[4] Andrea Bobbio, András Horváth, and Miklós Telek. Matching three moments with minimal acyclic phase type distributions. *Stochastic models*, 21(2-3):303–326, 2005.

[5] Levente Bodrog, Armin Heindl, Gábor Horváth, and Miklós Telek. A markovian canonical form of second-order matrix-exponential processes. *European Journal of Operational Research*, 190(2):459–477, 2008.

Perf. meas.	QBDQueue	MAPMAP1	MMAPPH1FCFS	MMAPPH1-Prio	FluidQueue	FluFluQueue
ncMoms	✓	✓	✓	✓	✓	✓
ncDistr	✓	✓	✓	✓	✓	✓
ncDistrMG	✓	✓	–	–	✓	✓
ncDistrDPH	✓	✓	–	–	✓	✓
stMoms	✓	✓	✓	✓	✓	✓
stDistr	✓	✓	✓	✓	✓	✓
stDistrME	✓	✓	✓	–	✓	✓
stDistrPH	✓	✓	✓	–	✓	✓

Table 2: Performance measures that can be computed

```
>> trace = dlmread('lbltcp3_iat.txt');
>> trmoms = MarginalMomentsFromTrace (trace,3);
>> tracf1 = LagCorrelationsFromTrace(trace,1);
>> [D0,D1]=MAP2FromMoments(trmoms,acf1);
>> [alpha,A] = APHFrom2Moments([0.8,6.5]);
>> S0=A;
>> S1=sum(-A,2)*alpha;
>> [ncm,ncd,beta,B]=MAPMAP1(D0,D1,S0,S1,'ncMoms',3,'ncDistr',20,'stDistrPH');
>> disp(ncm);
       21.359 1245.1 1.0969e+05
>> plot(ncd);
>> disp(MomentsFromPH(beta,B,3));
       21.359 1172.3 97262
```

Figure 3: Usage example for queueing

[6] Peter Buchholz and Miklós Telek. On minimal representations of rational arrival processes. *Annals of Operations Research*, 202(1):35–58, 2013.

[7] A. da Silva Soares. *Fluid Queues: Building Upon the Analogy with QBD Processes*. PhD thesis, Université Libre de Bruxelles, 2005.

[8] Qi-Ming He. Age process, workload process, sojourn times, and waiting times in a discrete time SM[K]/PH[K]/1/FCFS queue. *Queueing Systems*, 49(3-4):363–403, 2005.

[9] Gábor Horváth. Matching marginal moments and lag autocorrelations with MAPs. In *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*, pages 59–68, 2013.

[10] Gábor Horváth. Efficient analysis of the MMAP[K]/PH[K]/1 priority queue. *European Journal of Operational Research*, 246(1):128–139, 2015.

[11] Gábor Horváth and Hiroyuki Okamura. A fast EM algorithm for fitting marked markovian arrival processes with a new special structure. In *European Workshop on Performance Engineering*, pages 119–133. Springer, 2013.

[12] Gábor Horváth and Miklós Telek. On the canonical representation of phase type distributions. *Performance Evaluation*, 66(8):396–409, 2009.

[13] Gábor Horváth and Miklós Telek. Sojourn times in fluid queues with independent and dependent input and output processes. *Performance Evaluation*, 79:160–181, 2014.

[14] Illés Horváth, János Papp, and Miklós Telek. On the canonical representation of order 3 discrete phase type distributions. *Electronic Notes in Theoretical Computer Science*, 318:143–158, 2015.

[15] Guy Latouche and Vaidyanathan Ramaswami. *Introduction to matrix analytic methods in stochastic modeling*, volume 5. Siam, 1999.

[16] Lester Lipsky. *Queueing Theory: A linear algebraic approach*. Springer Science & Business Media, 2008.

[17] András Mészáros and Miklós Telek. Canonical representation of discrete order 2 MAP and RAP. In *European Workshop on Performance Engineering*, pages 89–103. Springer, 2013.

[18] Kenneth Mitchell and Appie van de Liefvoort. Approximation models of feed-forward G/G/1/N queueing networks with correlated arrivals. *Performance Evaluation*, 51(2):137–152, 2003.

[19] Ştefăniţă Mocanu and Christian Commault. Sparse representations of phase-type distributions. *Stochastic Models*, 15(4):759–778, 1999.

[20] Toshihisa Ozawa. Sojourn time distributions in the queue defined by a general QBD process. *Queueing Systems*, 53(4):203–211, 2006.

[21] János Papp and Miklós Telek. Canonical representation of discrete phase type distributions of order 2 and 3. In *Proc. of UK Performance Evaluation Workshop, UKPEW*, volume 2013, 2013.

[22] Philipp Reinecke and Miklós Telek. Does a given vector–matrix pair correspond to a PH distribution? *Performance Evaluation*, 81:40–51, 2014.

[23] Miklós Telek and Armin Heindl. Moment bounds for acyclic discrete and continuous phase type distributions of second order. In *In Proceedings of UK Performance Evaluation Workshop, UKPEW*. Citeseer, 2002.

A VONATKOZO ABRAT BERAKNI !!!!!!

Figure 4: The output of the example of Figure 3

[24] Miklós Telek and Gábor Horváth. A minimal representation of markov arrival processes and a moments matching method. *Performance Evaluation*, 64(9):1153–1168, 2007.

[25] Axel Thummler, Peter Buchholz, and Miklós Telek. A novel approach for fitting probability distributions to real trace data with the EM algorithm. In *2005 International Conference on Dependable Systems and Networks (DSN'05)*, pages 712–721. IEEE, 2005.

[26] A Van de Liefvoort. The moment problem for continuous distributions. *Unpublished technical report, University of Missouri, WP-CM-1990-02, Kansas City*, 1990.