



# PH-graphs for analyzing shortest path problems with correlated traveling times



Peter Buchholz\*, Iryna Felko

Informatik IV, TU Dortmund, D-44221 Dortmund, Germany

## ARTICLE INFO

Available online 21 January 2015

### Keywords:

Stochastic shortest path problems  
Markov decision processes  
Phase type distributions  
Correlated edge weights

## ABSTRACT

This paper presents a new approach to model weighted graphs with correlated weights at the edges. Such models are important to describe many real world problems like routing in computer networks or finding shortest paths in traffic models under realistic assumptions. Edge weights are modeled by phase type distributions (PHDs), a versatile class of distributions based on continuous time Markov chains (CTMCs). Correlations between edge weights are introduced by adding dependencies between the PHDs of adjacent edges using transfer matrices. The new model class, denoted as PH graphs (PHGs), allows one to formulate many shortest path problems as the computation of an optimal policy in a continuous time Markov decision process (CTMDP). The basic model class is defined, methods to parameterize the required PHDs and transfer matrices based on measured data are introduced and the formulation of basic shortest path problems as solutions of CTMDPs with the corresponding solution algorithms are also provided. Numerical examples for some typical stochastic shortest path problems demonstrate the usability of the new approach.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

The computation of shortest paths in a weighted graph is a well-known problem that appears in many application areas like abstract graph problems, route finding in traffic or computer networks or reliability analysis to mention only a few examples. Basically, the problem has been defined on graphs with constant lengths or weights of the edges where the shortest path between a source node and a destination node is computed efficiently with the well-known Dijkstra algorithm [17] or its variants. However in many practical applications, weights are stochastic rather than deterministic. This is for example the case in vehicle routing where the traffic on a street can only be estimated or in computer networks where the utilization of links is statistically fluctuating. This kind of problems results in a variant of the so-called stochastic shortest path problems [45]. The stochastic shortest path problem is often much more complex than the deterministic version since a variety of questions arises according to the optimality of a path and the assumptions which are necessary to specify the model. One can consider a path as optimal if its expected weight is minimal which results in a relatively simple problem if the weights of the edges are independent and

completely specified random variables. However, often weights are dependent. Dependencies may occur for example due to time-dependent traveling times or they may occur between the weights of adjacent edges in the graph. Additionally, the expected traveling time is often not the only and not even the major result, the variance or the probability of meeting a deadline are also important resulting in different problem formulations. Depending on the concrete assumptions and requirements, the problem can become complex and hard to solve such that solution algorithms often apply approximations or heuristics [47].

In this paper, we propose a new model to analyze stochastic shortest path problems with dependent weights of the edges which is denoted as *PH graphs* (PHGs). It is assumed that weights are defined by phase type distributions (PHDs) [32], a general class of distributions that allow one to approximate distributions with non-zero densities on the positive half-plane arbitrarily close [35]. Correlation between the weights of adjacent edges is introduced by making the initial phase of the PHD for the second edge dependent on the last phase of the PHD of the first edge. This is done by introducing the so-called transfer matrices. The description of dependencies is similar to the description of dependencies in Markovian arrival processes (MAPs) [31], a well-established process model in computational probability. A PHG includes all edges of the graph and for each edge the set of states defined for the corresponding PHD. We will show that each PHG can be mapped onto a *Markov decision process* (MDP) [40] in continuous

\* Corresponding author.

E-mail addresses: [peter.buchholz@cs.tu-dortmund.de](mailto:peter.buchholz@cs.tu-dortmund.de) (P. Buchholz), [iryna.felko@cs.tu-dortmund.de](mailto:iryna.felko@cs.tu-dortmund.de) (I. Felko).

time (CTMDP) such that the shortest path problem is equivalent to the computation of an optimal policy for the CTMDP. The focus of this paper is on the definition of the class of PHGs and on methods for the parameterization of models based on measurements resulting from real systems or adequate simulation models. It is not so much on specific approaches to compute the shortest path. However, those algorithms are available for MDPs and can be used for MDPs resulting from PHGs as well, as we will outline.

The paper is structured as follows. In the next section, a brief overview of related work is given. Afterwards, Section 3 introduces the model class of PHGs. In Section 4 the problem of parameterizing PHDs and transfer matrices is considered which is denoted as fitting in the area of PHDs and MAPs. Then the analysis of the weights of a given path and two different versions of the shortest path problem in PHGs are defined and it is outlined how these problems can be solved using common algorithms for MDPs. Section 7 analyzes the complexity of the proposed approach theoretically and experimentally. In Section 8 an example of a vehicular traffic network is presented. Finally, the paper ends with the conclusions which include a brief outlook on possible extensions of the model.

## 2. Related work

*Stochastic shortest path problems* are the most studied problems in random graphs. In stochastic shortest path problems the weights at the edges are modeled by random variables and one looks for a path with minimal weight. Bonet and Geffner proposed in [9] a real-time dynamic programming algorithm for minimizing the expected weight using Markov decision theory. The variant I-SSPPR was introduced by Andreatta and Romeo [2] describing deterministic networks with stochastic topology where random links can be either active or inactive. Then the system state containing information about active and inactive edges is known by a decision maker and the decision to reroute can be made each time a node with an inactive edge has been reached. This work was subsequently studied by Bertsekas and Tsikiklis [7]. Their model includes the joint probability distribution of random variables describing edge weights. Two other variants of the problem have been introduced, namely networks with dependent and independent edge weights. As the decision maker traverses the network from the source to the sink, the actual edge weights are learned and their realizations remain afterwards constant. Proposed dynamic programming algorithms have exponential run times in the number of realizations of the network, and the algorithm for solving the variant with independent edge travel times is exponential in the number of edges. It has been shown that the problem with dependent edge weights and the one with independent edge weights are NP-hard [38]. In [7] optimality results for the general case of the problem, where at each node the probability distribution over all possible successor nodes is available, have been provided. Complexity results and heuristic algorithms are given in [38,39,34,33,52].

Stochastic route planning problems often have been considered in the context of online adaptive algorithms, thus making Markov decision theory most suitable for constructing policies based on all weights realized in a route to the current location. Boyan and Mitzenmacher studied in [10] the bus network problem, where the goal is to compute an optimal plan within a city minimizing the expected traveling time. The bus network problem has been formulated as a time-dependent Markov decision process where the actions in each state are whether or not to take a bus when it arrives. Boyan and Littman address in [11] extension of the CTMDP model in which stochastic state transitions as well as stochastic, time-dependent action durations are included. Stochastic time-

dependent variants of the problem have also been extensively studied in [22,20,51]. These problems have a wide range of applications in robotics as the basis for mobile robot navigation. Simmons and Koenig [46] use partially observable Markov models for autonomous office navigation. Their settings maintain topological environment information as well as distance, sensor and actuator data. Then the position of an autonomous robot can be estimated from the Markov model and a navigation decision can be made with respect to temporary uncertainty in position and sensor data. Briggs et al. [12] solve expected shortest paths in context of robot navigation in stochastic environments. The expected shortest path problem is then defined as a Markov model, where it is assumed that the location of the robot is always known but the state of an edge can change as the robot traverses the graph. Relevant work based on partially observable MDPs has been done in the field of motion planning [46,49].

However, there has been little work on decision methods which directly incorporate correlations between edges and compute an optimal path on this basis. Relevant work most closely related to our graph model is [18] where a variant of a congested network with correlated edge weights has been studied. Each edge has been considered to be in one of the two possible conditions, either congested or not. The conditional probability density functions for edge weights are assumed to be known and the exact solution of the multistage adaptive problem has been developed.

The approach presented here has its roots in matrix analytic methods based on Neuts' fundamental work [32] which has been further developed by Asmussen [3]. A detailed overview of the related work in the field of phase-type distributions, MAPs and matrix analytic methods would exceed the scope of this paper and thus we mention only major results which are used to parameterize models. The finding of adequate parameters for a PHD is denoted as fitting. Most of the techniques for fitting PHDs are based on expectation maximization (EM) algorithms which maximize the likelihood according to measured data, e.g., [4,50] where variants of the algorithm have been used to fit a PHD to a trace. Our work on parameter fitting is mainly based on [14,26] which describes an efficient method for fitting the parameters of the PHD in one step and obtains a MAP or MMPP in a second step. This approach is used to estimate the parameters of the transfer matrix to describe correlated weights on consecutive edges. In [50] a method for fitting Hyper-Erlang distributions to a trace is represented. It has been shown that any probability density function of a non-negative random variable can be approximated arbitrarily close by a Hyper-Erlang distribution. We apply software tools like *gfit* from [50] to obtain the parameters of PHDs modeling the weights of edges in our graph.

To introduce correlation, the parameters of the transfer matrices have to be estimated based on measurements. The resulting fitting problem is related to the fitting of the  $D_1$  matrix in MAP fitting approaches [14,26,48]. We adopt methods described in these papers to obtain a non-negative least squares formulation of the problem. Algorithms for MAP generation that approximate the moments and joint moments of the observed traces are most relevant. In our case, traveling times are measured from traces and one can solve the fitting problem by a non-negative least squares problem considering joint moments of the trace and some necessary conditions resulting from the parameters of PHDs of adjacent edges.

## 3. Graphs with PH distributed weights

PH graphs (PHGs) are a weighted graph model where edge weights are random variables modeled by PHDs and correlation between arc weights is introduced by making the initial state of

the PHD of the following edge dependent on the final state of the PHD of the previous edge. We first introduce some notation before we describe the details of the model.

Consider a directed weighted graph  $(\mathcal{V}, \mathcal{E})$  where  $\mathcal{V}$  is the set of vertices and  $\mathcal{E}$  is the set of edges. Edges are numbered with the letters  $i, j$ . There is a starting vertex  $v_{ini} \in \mathcal{V}$  and a final vertex  $v_{fin} \in \mathcal{V}$  with  $v_{ini} \neq v_{fin}$ . An edge is a directed connection between two vertices. For  $i \in \mathcal{E}$  let  $ini(i) \in \mathcal{V}$  be the starting vertex of edge  $i$  and  $fin(i) \in \mathcal{V}$  the destination vertex. We say that edges  $i, j \in \mathcal{E}$  are adjacent, if  $fin(i) = ini(j)$ . The following two sets define the predecessors and successors of edges:

$$\bullet i = \begin{cases} \emptyset & \text{if } ini(i) = v_{ini} \\ \{j | fin(j) = ini(i)\} & \text{otherwise} \end{cases}$$

$$i \bullet = \begin{cases} \emptyset & \text{if } fin(i) = v_{fin} \\ \{j | ini(j) = fin(i)\} & \text{otherwise} \end{cases} \quad (1)$$

Furthermore,  $\mathcal{E}_{ini} = \{i | ini(i) = v_{ini}\}$  and  $\mathcal{E}_{fin} = \{i | fin(i) = v_{fin}\}$ . A path between  $v_{ini}$  and  $v_{fin}$  is a sequence of edges  $(i_1, \dots, i_K)$  such that  $i_1 \in \mathcal{E}_{ini}$ ,  $i_K \in \mathcal{E}_{fin}$  and  $i_{k-1} \in \bullet i_k$  for  $k=2, \dots, K$ . Let  $\mathcal{P}$  be the set of all finite paths between  $v_{ini}$  and  $v_{fin}$ . We assume that from every  $v \in \mathcal{V}$ , at least one path to  $v_{fin}$  exists. Observe that we are not restricted to acyclic structures since we allow paths of arbitrary but finite length.

The weight of edge  $i$  is a non-negative random variable  $X_i$  which has a PHD  $(\pi_i, \mathbf{D}_i)$  representation of order  $n_i$  [32]. A PHD  $(\pi_i, \mathbf{D}_i)$  of order  $n_i$  is defined by an absorbing Markov process with  $n_i + 1$  states, initial vector  $(\pi_i, \mathbf{0})$  and transition matrix

$$\begin{pmatrix} \mathbf{D}_i & -\mathbf{D}_i \mathbb{1} \\ \mathbf{0} & 0 \end{pmatrix}.$$

$\pi_i$  is an  $n_i$  dimensional (row) distribution vector,  $\mathbf{D}_i$  is an  $n_i \times n_i$  matrix with negative diagonal elements, non-negative non-diagonal elements and row sum less or equal to zero. We assume  $\mathbf{D}_i$  is non-singular which implies that  $\mathbf{M}_i = (-\mathbf{D}_i)^{-1}$  is non-negative and  $\lim_{t \rightarrow \infty} e^{\mathbf{D}_i t} \rightarrow \mathbf{0}$ . The distribution function and moments of the PHD are given by

$$F_i(t) = 1 - \pi_i e^{\mathbf{D}_i t} \mathbb{1} \quad \text{and} \quad E(X_i^k) = k! \pi_i \mathbf{M}_i^k \mathbb{1}. \quad (2)$$

The literature on PHDs is comprehensive and contains a large number of results (see e.g., [32,35,36]). If the edge weights are defined by PHDs, then the different weights are independent. To introduce dependencies, we extend the model by introducing a dependence between the state where one PHD is left and the next one is entered. For  $i, j \in \mathcal{E}$  with  $i \in \bullet j$  define a  $n_i \times n_j$  matrix  $\mathbf{H}_{ij}$  with  $\mathbf{H}_{ij} \geq \mathbf{0}$  and  $\mathbf{H}_{ij} \mathbb{1} = -\mathbf{D}_i \mathbb{1}$ . We will denote matrix  $\mathbf{H}_{ij}$  as the transfer matrix. Then let  $\mathbf{P}_{ij} = \mathbf{M}_i \mathbf{H}_{ij}$ . Since  $\mathbf{M}_i, \mathbf{H}_{ij} \geq \mathbf{0}$  also  $\mathbf{P}_{ij} \geq \mathbf{0}$  and

$$\mathbf{P}_{ij} \mathbb{1} = \mathbf{M}_i \mathbf{H}_{ij} \mathbb{1} = (-\mathbf{D}_i)^{-1} (-\mathbf{D}_i) \mathbb{1} = \mathbb{1}.$$

Thus,  $\mathbf{P}_{ij}$  is a stochastic matrix. The interpretation of matrix  $\mathbf{P}_{ij}$  is as follows: if the model begins in phase  $x$  at edge  $i$  and chooses  $j$  as next edge, it will begin with probability  $\mathbf{P}_{ij}(x, y)$  in phase  $y$  at edge  $j$ . To keep the weight distribution of edge  $j$  invariant, we additionally need

$$\pi_i \mathbf{P}_{ij} = \pi_j \Rightarrow \pi_i \mathbf{M}_i \mathbf{H}_{ij} = \pi_j. \quad (3)$$

If we choose  $\mathbf{P}_{ij}$  as above, then this implies that if the system starts with probability distribution  $\pi_i$  at edge  $i$ , then it starts in the average with probability distribution  $\pi_j$  at edge  $j$ . Thus, the distribution function of the weight of  $j$  remains unchanged. However, the weights are no longer independent since we obtain

$$E(X_i^k, X_j^l) = k! l! \pi_i \mathbf{M}_i^k \mathbf{P}_{ij} \mathbf{M}_j^l \mathbb{1} \quad (4)$$

for the joint moments of order  $k, l$ . The computation of joint moments follows from the computation of joint moments in MAPs as given for example in [14]. In particular, the covariance of the weights is given by

$\text{Cov}(X_i, X_j) = E(X_i^1, X_j^1) - E(X_i^1) \cdot E(X_j^1) = \pi_i \mathbf{M}_i \mathbf{P}_{ij} \mathbf{M}_j \mathbb{1} - (\pi_i \mathbf{M}_i \mathbb{1}) \cdot (\pi_j \mathbf{M}_j \mathbb{1})$ . Observe that the correlation is defined for subsequent edges but over a path of length greater than two the effects cumulate. Weights may have different interpretations in different application areas. Often they are interpreted as time, e.g., as traveling time on a road or a link of a computer network. However, other interpretations like monetary cost or failure rates of components are also possible.

The analysis of paths in a PHG is considered in Section 5 and the computation of shortest paths in Section 6. Before these results are presented, we consider the parameterization of the PHDs and the matrices  $\mathbf{H}_{ij}$ .

#### 4. Fitting of distributions and correlation

PHDs describing the weights of the edges and matrices  $\mathbf{H}_{ij}$  defining the dependencies have to be parameterized. Usually this is done according to values resulting from real measurements or simulations. We assume that a set of measurements of entities that pass through the graph is available. A single measurement is given by a sequence  $\mathbf{s}_k = ((i_k^1, w_k^1), \dots, (i_k^{c_k}, w_k^{c_k}))$  where  $i_k^j \in \mathcal{V}$ ,  $i_k^{j-1} \in \bullet i_k^j$  and  $0 < w_k^j < \infty$  ( $1 < j \leq c_k$ ). Observe that the sequences need not start in  $v_{ini}$  or end in  $v_{fin}$ .  $\mathcal{M}$  is the set of all measured sequences and  $K$  is the number of measured sequences. For edge  $i$ ,  $\mathcal{W}_i$  is the multiset of all  $w_k^j$  from  $\mathbf{s}_k \in \mathcal{M}$ , such that  $\mathbf{s}_k$  contains an element  $(i_k^j, w_k^j)$  with  $i_k^j = i$ . If the value  $w_k^j$  is part of several sequences or occurs several times in one sequence, then it appears several times in  $\mathcal{W}_i$  such that  $\mathcal{W}_i$  is a multiset and not a set. Similarly,  $\mathcal{W}_{ij}$  is the multiset of tuples  $(w_i, w_j)$  which appear in some  $\mathbf{s}_k \in \mathcal{M}$  such that  $(i_k^h, w_i), (i_k^{h+1}, w_j)$  with  $i_k^h = i$  and  $i_k^{h+1} = j$ . Again, if the same pair appears in several sequences or several times in one sequence, then it appears several times in the multiset. Let  $d_i$  and  $d_{ij}$  be the number of elements in  $\mathcal{W}_i$  and  $\mathcal{W}_{ij}$ , respectively. In summary,  $\mathcal{W}_i$  contains all measured weights for edge  $i$  and  $\mathcal{W}_{ij}$  all measured value pairs of adjacent edges  $i$  and  $j$ .

We begin with the computation of parameters for the PHDs  $(\pi_i, \mathbf{D}_i)$  related to the edges. There are two possible ways of fitting a PHD to a set of measured values. One can use the measured values directly, alternatively, first results like moments are derived from the measurements and then the parameters are fitted according to these results. If the whole set of measured values is used, then usually the likelihood, which is defined as

$$L_{\pi_i, \mathbf{D}_i}(\mathcal{W}_i) = \prod_{w \in \mathcal{W}_i} \pi_i e^{w \mathbf{D}_i} (-\mathbf{D}_i) \mathbb{1} \quad (5)$$

is maximized using an expectation maximization (EM) algorithm. EM algorithms have a guaranteed convergence towards a local maximum but convergence can be very slow. The first version of an EM algorithm for PHDs has been published in [4], later extensions [28,50] use restricted classes of PHDs which makes the parameter fitting much more efficient without reducing the fitting quality significantly. For example, with the *gfit* algorithm presented in [50] Hyper-Erlang distributions, a subclass of PHDs, can be parameterized fairly quickly as shown experimentally in Section 7.

Alternatively, it is possible to first derive some quantities from the measurements which are subsequently used for fitting. Typically moments are estimated

$$\nu_i^k = \frac{1}{d_i} \sum_{w \in \mathcal{W}_i} w^k \quad (6)$$

is an estimate for  $\mu_i^k = E(X_i^k)$ . Methods for fitting parameters of a PH distribution according to the moments are presented in [8,14,25]. Fitting algorithms for PHDs using maximum likelihood or moment

fitting approaches are implemented in several freely available tools [5,24].

After the PHDs are parameterized to model the weight distributions at the edges, correlation is added by considering values in the sets  $\mathcal{W}_{ij}$ . Parameters in the matrices  $\mathbf{H}_{ij}$  can also be fitted by maximizing the likelihood function or by an approximation of derived values, namely the joint moments. We first introduce the latter approach.

Let  $i, j$  be two edges,  $i \in \bullet j$  and let  $\mathcal{W}_{ij}$  contain more than two elements, then

$$\nu_{ij}^{k,l} = \frac{1}{d_{ij} - 2} \sum_{(w_i, w_j) \in \mathcal{W}_{ij}} w_i^k w_j^l \quad (7)$$

is an estimate for  $\mu_{ij}^{k,l} = E(X_i^k, X_j^l)$ . Assume that joint moments  $(k, l)$  with  $1 \leq k \leq K, 1 \leq l \leq L$  should be considered. Since  $\mu_{ij}^{k,l} = k!l!\pi_i \mathbf{M}_i^{k-1} \mathbf{H}_{ij} \mathbf{M}_j^{l-1}$  is linear in the elements of matrix  $\mathbf{H}_{ij}$

$$\mathbf{H}_{ij} \cdot \mathbf{H}_{ij} \geq \mathbf{0} \wedge \mathbf{H}_{ij} \mathbf{1} = -\mathbf{D}_i \mathbf{1} \wedge \pi_i \mathbf{M}_i \mathbf{H}_{ij} = \pi_j \left( \sum_{k=1}^K \sum_{l=1}^L \left( \frac{\mu_{ij}^{k,l}}{\nu_{ij}^{k,l}} - 1 \right)^2 \right) \quad (8)$$

is a non-negative least squares problem with equality constraints which can be solved with standard methods [29] (see also [14]). The algorithm for solving the least squares problem is efficient for problems with up to a few hundred variables. The number of variables corresponds to the number of non-zero elements in matrix  $\mathbf{H}_{ij}$  which equals the product of the number of non-zero elements in  $\pi_j$  and  $\mathbf{D}_i \mathbf{1}$ . If the minimum in (8) becomes zero, the estimated joint moments are exactly fitted by the composition of the two phase type distributions. The equality constraint  $\pi_i \mathbf{M}_i \mathbf{H}_{ij} = \pi_j$  assures that the initial distribution of the PHD for edge  $j$  remains  $\pi_j$ .

To maximize the likelihood, the maximum of

$$L_{\mathbf{H}_{ij}} = \prod_{(w_i, w_j) \in \mathcal{W}_{ij}} \pi_i e^{w_i \mathbf{D}_i} \mathbf{H}_{ij} e^{w_j \mathbf{D}_j} (-\mathbf{D}_j) \mathbf{1} \quad (9)$$

has to be computed for  $\mathbf{H}_{ij} \geq \mathbf{0}$ ,  $\mathbf{H}_{ij} \mathbf{1} = -\mathbf{D}_i \mathbf{1}$  and  $\pi_i \mathbf{M}_i \mathbf{H}_{ij} = \pi_j$ . We present a combination of an EM algorithm and a non-negative least squares solution to compute an appropriate matrix.

First, matrix  $\mathbf{X}^{(0)} = -\mathbf{D}_i \mathbf{1} \pi_j$  is generated which is a matrix describing uncorrelated weights at  $i$  and  $j$ . Other more sophisticated initializations of  $\mathbf{X}^{(0)}$  may be used as well, but the conditions  $\mathbf{X}^{(0)} \mathbf{1} = -\mathbf{D}_i \mathbf{1}$  and  $\pi_i \mathbf{M}_i \mathbf{X}^{(0)} = \pi_j$  have to hold. Then iterations are performed and in each iteration the following three steps are computed to obtain matrix  $\mathbf{X}^{(k)}$  from  $\mathbf{X}^{(k-1)}$ . First, an EM step is made to compute a matrix  $\mathbf{Y}^{(k)}$  as follows:

$$\mathbf{Y}^{(k)}(x, y) = \sum_{(w_i, w_j) \in \mathcal{W}_{ij}} \pi_i e^{w_i \mathbf{D}_i} \mathbf{X}^{(k-1)}(x, y) e^{w_j \mathbf{D}_j} (-\mathbf{D}_j) \mathbf{1} \quad (10)$$

for  $1 \leq x \leq n_i$  and  $1 \leq y \leq n_j$ . Observe that  $\mathbf{X}^{(k-1)}(x, y) = 0$  implies  $\mathbf{Y}^{(k)}(x, y) = 0$  which implies that zero elements in  $\mathbf{X}^{(k-1)}$  remain zero in  $\mathbf{Y}^{(k)}$ . In the second step, the rows of  $\mathbf{Y}^{(k)}$  are normalized such  $\mathbf{X}^{(k-1)}$  and  $\mathbf{Y}^{(k)}$  have the same row sums. Thus, all non-zero entries are transformed to

$$\mathbf{Y}^{(k)}(x, y) = \frac{\sum_{z=1}^{n_j} \mathbf{X}^{(k-1)}(x, z) \mathbf{Y}^{(k)}(x, z)}{\sum_{z=1}^{n_j} \mathbf{Y}^{(k)}(x, z)} \quad (11)$$

For the resulting matrix  $\mathbf{Y}^{(k)} \mathbf{1} = -\mathbf{D}_i \mathbf{1}$  and  $\mathbf{Y}^{(k)} \geq \mathbf{0}$  holds but the third condition assuring initial distribution  $\pi_j$  usually will not hold. Therefore, a non-negative least squares problem with equality constraints is solved to compute the new matrix  $\mathbf{X}^{(k)}$

$$\mathbf{X}^{(k)} \cdot \mathbf{X}^{(k)} \geq \mathbf{0} \wedge \mathbf{X}^{(k)} \mathbf{1} = \mathbf{Y}^{(k)} \mathbf{1} \wedge \pi_i \mathbf{M}_i \mathbf{X}^{(k)} = \pi_j \left( \|\mathbf{X}^{(k)} - \mathbf{Y}^{(k)}\|^2 \right) \quad (12)$$

$\|\cdot\|$  is the Frobenius norm. Observe that (12) and (8) are identical problems. The resulting matrix  $\mathbf{X}^{(k)}$  is used as  $\mathbf{H}_{ij}$  and assures often a good approximation of the measured correlation.

**Example 1.** We use a small queueing example to show the fitting of a PHD to measured data. The basic model is shown in Fig. 1a.

*Model description and parameterization:* The model consists of three queues. Arrivals to  $Q_1$  are generated by a *Markov modulated Poisson process* (MMPP) with two states. In an MMPP, arrival rates of a Poisson process depend on the state of an independent Markov process [19]. The MMPP is a special case of the more general MAP. The 2-state MMPP for our model is given by the following matrices:

$$\mathbf{D}_0^{MMPP} = \begin{pmatrix} -0.005 & 0.001 \\ 2 & -13 \end{pmatrix}, \quad \mathbf{D}_1^{MMPP} = \begin{pmatrix} 0.004 & 0 \\ 0 & 11 \end{pmatrix},$$

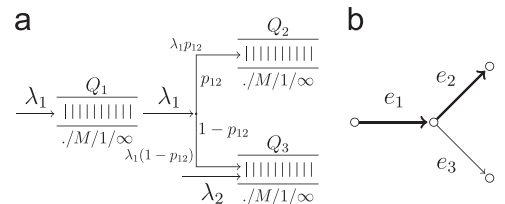
where  $\mathbf{D}_0^{MMPP}$  is the infinitesimal generator of the continuous time Markov chain and  $\mathbf{D}_1^{MMPP}$  is the matrix containing two Poisson arrival rates on the diagonal. If the Markov chain is in state  $i$ , arrivals occur according to a Poisson process with rate  $\mathbf{D}_1^{MMPP}(i, i)$ .

The service time at  $Q_1$  is exponentially distributed. After leaving  $Q_1$  an entity enters with probability  $p_{12}$ ,  $Q_2$  and with probability  $p_{13} = 1 - p_{12}$ ,  $Q_3$ . Service times at  $Q_2$  and  $Q_3$  are also exponentially distributed. We assume that  $p_{12} = 0.99$ .  $Q_3$  receives an additional Poisson arrival stream with rate  $\lambda_2$ .

The MMPP generating the arrivals has a high arrival rate in state 2 and a low arrival rate in state 1. It stays a long time in state 1 with a low arrival rate which implies that the load of  $Q_1$  and subsequently  $Q_2$  is low. In state 2,  $Q_1$  fills quickly such that customers are backlogged and waiting times increase. Additionally, many customers leave towards  $Q_2$  which also fills up. The effect of the high arrival rate on  $Q_3$  is marginal because the routing probability from  $Q_1$  to  $Q_3$  is small. Thus, the sojourn times in  $Q_1$  and  $Q_2$  are highly correlated whereas the sojourn times in  $Q_1$  and  $Q_3$  are almost independent.

A graph abstraction of the queueing model is shown in Fig. 1b. The sojourn time of  $Q_1$  corresponds to the traveling time along the edge  $e_1$  in Fig. 1b. Analogously, the sojourn times of  $Q_2$  and  $Q_3$  correspond to the traveling time along the adjacent edges  $e_2$  and  $e_3$ , respectively.

*Simulation results:* Samples are generated from a model which is implemented in the OMNeT++ simulator [23]. We produced a trace with  $K=14,300$  observed samples for each queue. From the trace the three sets  $\mathcal{W}_{Q_1}$ ,  $\mathcal{W}_{Q_2}$ , and  $\mathcal{W}_{Q_3}$  containing all measured sojourn times in queues and two sets  $\mathcal{W}_{Q_1, Q_2}$ ,  $\mathcal{W}_{Q_1, Q_3}$  containing all measured value pairs of consecutive queues are generated. The sets  $\mathcal{W}_{Q_1}$ ,  $\mathcal{W}_{Q_2}$ , and  $\mathcal{W}_{Q_3}$  are used for fitting three phase-type distributions of order 4 using the software *gfit* (see, e.g., [50]). The sojourn time of entities traveling through  $Q_1$  and  $Q_2$  is correlated with a correlation coefficient of  $\hat{\rho} = 0.1345$ , and the first joint moment  $\hat{\mu} = 1.2142$ . The sojourn times for each queue are summarized graphically in Fig. 2. Obviously, the sojourn times in  $Q_1$  and  $Q_2$  have large peaks and the correlation between both sojourn time becomes visible. The sojourn time in  $Q_3$  is less variable and no correlation with the sojourn time in  $Q_1$  or  $Q_2$  is visible.



**Fig. 1.** The network of  $.M/1/\infty$  queues and corresponding traffic junction graph in Example (1). The traveling times along the edges are modeled by residence times of  $Q_1$ ,  $Q_2$ , and  $Q_3$  in the open queue network in (a). In (b) the edges with correlated traveling times are highlighted.



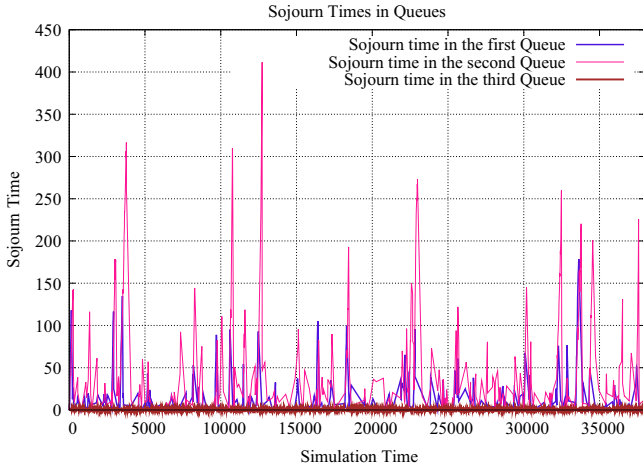


Fig. 2. Sojourn times of entities traveling through queues  $Q_1$ ,  $Q_2$ , and  $Q_3$ .

We obtain with *gfit* a hyperexponential distributions  $(\pi_{e_1}, \mathbf{D}_{e_1})$  for the sojourn time in  $Q_1$  and a Hyper-Erlang distribution  $(\pi_{e_2}, \mathbf{D}_{e_2})$  for the sojourn time in  $Q_2$

$$\mathbf{D}_{e_1} = \begin{pmatrix} -0.657 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.721 & 0.000 & 0.000 \\ 0.000 & 0.000 & -3.429 & 0.000 \\ 0.000 & 0.000 & 0.000 & -4.717 \end{pmatrix},$$

$$\mathbf{D}_{e_2} = \begin{pmatrix} -0.724 & 0.000 & 0.000 & 0.000 \\ 0.000 & -3.534 & 0.000 & 0.000 \\ 0.000 & 0.000 & -1.085 & 1.085 \\ 0.000 & 0.000 & 0.000 & -1.085 \end{pmatrix}.$$

The initial distributions are  $\pi_{e_1} = (0.4148, 0.1958, 0.1851, 0.2043)$  and  $\pi_{e_2} = (0.5206, 0.3858, 0.0936, 0)$ . The PHD  $(\pi_{e_3}, \mathbf{D}_{e_3})$  with  $\pi_{e_3} = (1)$  and  $\mathbf{D}_{e_3} = (-0.9999)$  models the sojourn time distribution of  $Q_3$  which is almost exponential because  $Q_3$  is an  $M/M/1$  system with a small additional load from  $Q_1$ .

To describe the correlation between the sojourn times along the edges  $e_1$  and  $e_2$  the values in matrix  $\mathbf{H}_{e_1e_2}$  have to be fitted. In this example we use the EM algorithm described above and the values from  $\mathcal{W}_{Q_1Q_2}$ . The final matrix  $\mathbf{Y}^{(k)}$  results from the EM step and the final approximation  $\mathbf{X}^{(k)}$  is computed by solving (12)

$$\mathbf{Y}^{(k)} = \begin{pmatrix} 0.499648 & 0 & 0.157417 & 0 \\ 0.335285 & 0.386282 & 0 & 0 \\ 0 & 3.429211 & 0 & 0 \\ 2.581962 & 2.135355 & 0 & 0 \end{pmatrix},$$

$$\mathbf{X}^{(k)} = \begin{pmatrix} 0.508965 & 0 & 0.148109 & 0 \\ 0.348646 & 0.372924 & 0 & 0 \\ 0 & 3.429212 & 0 & 0 \\ 2.418297 & 2.299020 & 0 & 0 \end{pmatrix}$$

The resulting matrix of the EM algorithm  $\mathbf{Y}^{(k)}$  exhibits the correlation  $\rho = 0.1296$  and the first joint moment 1.2099 but the condition  $\pi_{e_1} \mathbf{M}_{e_1} \mathbf{Y}^{(k)} = \pi_{e_2}$  is not satisfied. Thus, matrix  $\mathbf{X}^{(k)}$  is finally used as  $\mathbf{H}_{e_1e_2}$  and we obtain the value for the correlation  $\rho = 0.1294$  and for the first joint moment  $\mu_{e_1e_2}^{1,1} = 1.2097$ . Both are very good approximations of the values estimated from the traces of the simulation model.

Since sojourn times in  $Q_1$  and  $Q_3$  are uncorrelated, the fitting of the transfer matrix  $\mathbf{H}_{e_1e_3}$  is not required,  $\mathbf{H}_{e_1e_3} = \mathbf{d}_{e_1} \pi_{e_3}$ , where  $\mathbf{d}_{e_1} = (-\mathbf{D}_{e_1})^\parallel$  is the exit vector for the PHD that models the weight distribution of the edge  $e_1$ , is an appropriate choice.

### 5. Analysis of paths in PH graphs

For a given PHG paths can be analyzed via the analysis of absorbing Markov chains. Let  $(i_1, \dots, i_K) \in \mathcal{P}$ . The traveling time along a path is described by an absorbing continuous time Markov chain (CTMC) with  $n = \sum_{k=1}^K n_{i_k}$  states. Let  $\mathcal{S}$  be the state space of the CTMC. The initial vector equals  $\pi = (\pi_{i_1}, \mathbf{0})$  where the zero part is of length  $n - n_{i_1}$ . The subgenerator matrix of the absorbing CTMC equals

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D}_{i_1} & \mathbf{H}_{i_1i_2} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{i_2} & \mathbf{H}_{i_2i_3} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & & & \mathbf{D}_{i_{K-1}} & \mathbf{H}_{i_{K-1}i_K} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & \mathbf{D}_K \end{pmatrix}.$$

Matrix  $\mathbf{Q}$  is non-singular since all diagonal blocks are non-singular. Let  $\mathbf{M} = -\mathbf{Q}^{-1}$ , then

$$\mu_{i_1, \dots, i_K}^l = l! \pi \mathbf{M}^l \mathbb{1} \tag{13}$$

is the  $l$ th moment of the weight of the path and

$$F_{i_1, \dots, i_K}(w) = 1.0 - \pi e^{w\mathbf{Q}} \mathbb{1} \tag{14}$$

is the probability that the weight of the path less than or equal to  $w$ . If weights are traveling times, then  $\mu_{i_1, \dots, i_K}^l$  is the  $l$ th moment of the traveling time and  $F_{i_1, \dots, i_K}(t)$  is the distribution function of the traveling time.

Moments and distribution function can as well be computed from a discrete time Markov chain (DTMC) resulting from uniformization [21]. Let  $\alpha \geq \max_{x \in \mathcal{S}} (|\mathbf{Q}(x, x)|)$ , then  $\mathbf{P} = \mathbf{Q}/\alpha + \mathbf{I}$  is a stochastic matrix which is the transition probability matrix of an absorbing DTMC. We have  $\mathbf{N} = (\mathbf{I} - \mathbf{P})^{-1} = (\mathbf{I} - (\mathbf{Q}/\alpha + \mathbf{I}))^{-1} = \mathbf{M}\alpha$ . Furthermore  $\mathbf{N} = \sum_{k=0}^{\infty} \mathbf{P}^k$  [27]. Consequently, the moments can be computed from  $\mathbf{N}$  rather than  $\mathbf{M}$  and the probability  $F_{i_1, \dots, i_K}(w)$  can be computed using uniformization as follows:

$$F_{i_1, \dots, i_K}(w) = 1.0 - \pi e^{-\alpha w} \left( \sum_{h=0}^{\infty} \frac{(\alpha w)^h}{h!} \mathbf{P}^h \right) \mathbb{1} \tag{15}$$

The infinite sum can be truncated at a finite point with a predefined error bound [21]. The computation of (15) is more stable than the direct evaluation of the matrix exponential in (14) as shown several times in the literature (e.g., [21,41]).

If  $(i_1, \dots, i_K) \in \mathcal{P}$  and for some  $L < K$  ( $w_1, \dots, w_L$ ) ( $w_l > 0$ ) are the weights along the first  $L$  edges of the path, then the vector

$$\psi_{(i_1, w_1, \dots, i_L, w_L)} = \pi_{i_1} \left( \prod_{l=1}^{L-1} e^{\mathbf{D}_{i_l} w_l} \mathbf{H}_{i_l, i_{l+1}} \right) e^{\mathbf{D}_{i_L} w_L} \tag{16}$$

describes the distribution after passing the edges  $i_1, \dots, i_{L-1}$  with weights  $w_1, \dots, w_{L-1}$  and having accumulated weight  $w_L$  at edge  $i_L$ . A natural interpretation for the weights can again be traveling times. If  $w_L$  is the traveling time for  $i_L$ , then the vector describes the distribution among the states of the PHD for edge  $i_L$  immediately before leaving edge  $i_L$ . If  $w_L = 0$ , then the vector includes the distribution immediately after entering edge  $i_L$ . The vector can be computed using uniformization as follows:

$$\psi_{(i_1, w_1, \dots, i_L, w_L)} = \pi_{i_1} \prod_{l=1}^{L-1} \left( e^{-\alpha_l w_l} \left( \sum_{h=0}^{\infty} \frac{(\alpha_l w_l)^h}{h!} (\mathbf{P}_{i_l})^h \right) \left( \frac{\mathbf{H}_{i_l, i_{l+1}}}{\alpha_l} \right) \right) \times e^{-\alpha_L w_L} \left( \sum_{h=0}^{\infty} \frac{(\alpha_L w_L)^h}{h!} (\mathbf{P}_{i_L})^h \right) \tag{17}$$

where  $\alpha_l \geq \max_x (|\mathbf{D}_{i_l}(x, x)|)$  and  $\mathbf{P}_{i_l} = \mathbf{Q}_{i_l}/\alpha_l + \mathbf{I}$ . Let

$$\bar{\psi}_{(i_1, w_1, \dots, i_L, w_L)} = \frac{\psi_{(i_1, w_1, \dots, i_L, w_L)}}{\psi_{(i_1, w_1, \dots, i_L, w_L)} \mathbb{1}} \tag{18}$$

the vector normalized to 1. Then conditional weights of the remaining path through the edges  $i_{L+1}, \dots, i_K$  can be computed from an

absorbing CTMC with generator matrix

$$Q = \begin{pmatrix} D_{i_L} & H_{i_L, i_{L+1}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & D_{i_{L+1}} & H_{i_{L+1}, i_{L+2}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & \dots & \dots & D_{i_{K-1}} & H_{i_{K-1}, i_K} \\ \mathbf{0} & \dots & \dots & \mathbf{0} & D_K \end{pmatrix}$$

and initial vector  $(\bar{\psi}_{(i_1, w_1, \dots, i_L, w_L)}, \mathbf{0})$  as described above for the whole path.

**Example 2.** As an example we present the small four-node PHG shown in Fig. 3. The weight distributions of the edges in the PHG are modeled by PH distributions  $(\pi_x, D_x)$ ,  $x \in \{A, B, C, D\}$ , in hyper-exponential and Hyper-Erlang representation, and are summarized below. We interpret the weights as traveling times

$$D_A = \begin{pmatrix} -0.657 & 0 & 0 & 0 \\ 0 & -0.721 & 0 & 0 \\ 0 & 0 & -3.429 & 0 \\ 0 & 0 & 0 & -4.717 \end{pmatrix},$$

$$D_B = \begin{pmatrix} -0.724 & 0 & 0 & 0 \\ 0 & -3.534 & 0 & 0 \\ 0 & 0 & -1.085 & 1.085 \\ 0 & 0 & 0 & -1.085 \end{pmatrix},$$

$$D_C = \begin{pmatrix} -0.4745 & 0 & 0 & 0 \\ 0 & -0.4900 & 0 & 0 \\ 0 & 0 & -0.5230 & 0 \\ 0 & 0 & 0 & -0.5940 \end{pmatrix},$$

$$D_D = \begin{pmatrix} -0.3000 & 0 & 0 & 0 \\ 0 & -4.8847 & 0 & 0 \\ 0 & 0 & -6.8209 & 0 \\ 0 & 0 & 0 & -8.8200 \end{pmatrix}.$$

Initial distributions are given by  $\pi_A = (0.4148, 0.1958, 0.1851, 0.2043)$ ,  $\pi_B = (0.5206, 0.3858, 0.0936, 0)$ ,  $\pi_C = (0.4227, 0.2707, 0.1814, 0.1252)$ ,  $\pi_D = (0.3977, 0.3945, 0.2078, 0)$ . Path  $(AB) \in \mathcal{P}$  can be described by an acyclic absorbing CTMC with eight non-absorbing states as shown in Fig. 4. The generator matrix  $Q^{AB}$  equals

$$Q^{AB} = \begin{pmatrix} -0.657 & 0 & 0 & 0 & 0.509 & 0 & 0.148 & 0 \\ 0 & -0.721 & 0 & 0 & 0.721 & 0 & 0 & 0 \\ 0 & 0 & -3.429 & 0 & 0.064 & 3.364 & 0 & 0 \\ 0 & 0 & 0 & -4.717 & 0 & 4.717 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.724 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -3.534 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.085 & 1.085 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.085 \end{pmatrix}.$$

The initial vector of the CTMC  $Q^{AB}$  is given by  $\pi_{AB} = (\pi_A, 0, 0, 0, 0)$ . We compute the first moment from  $M^{AB} = (-Q^{AB})^{-1}$  which results in  $\mu_{AB}^1 = \pi_{AB} M^{AB} \mathbf{1} = 1.9999$ . Using  $H_{CD} = c\pi_D$ , where  $c = (-D_C)\mathbf{1}$  is the exit vector for the PHD that models the weight distribution of the edge C, we obtain  $Q^{CD}$  and using (13) the expected travel time for the path CD is  $\mu_{CD}^1 = \pi_{CD} M^{CD} \mathbf{1} = 3.4377$ .

The vector  $\psi_{(A, w, B, 0)}$  is computed using (17) and (18). If we assume that the traveling time on the edge A was  $w=0.5$ , then the vector including the distribution immediately after entering the edge B is

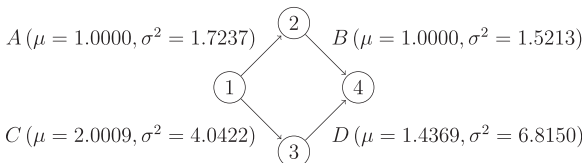


Fig. 3. A four-node PHG containing paths AB, CD from  $v_{ini} = 1$  to  $v_{fin} = 4$ .

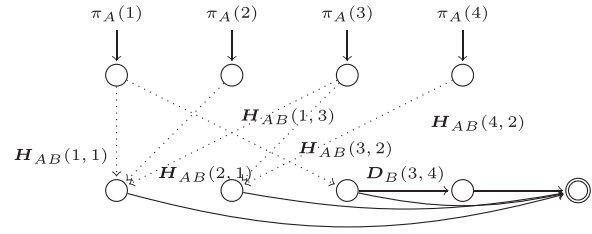


Fig. 4. The absorbing CTMC corresponding to the path AB of the PHG appearing in Fig. 3.

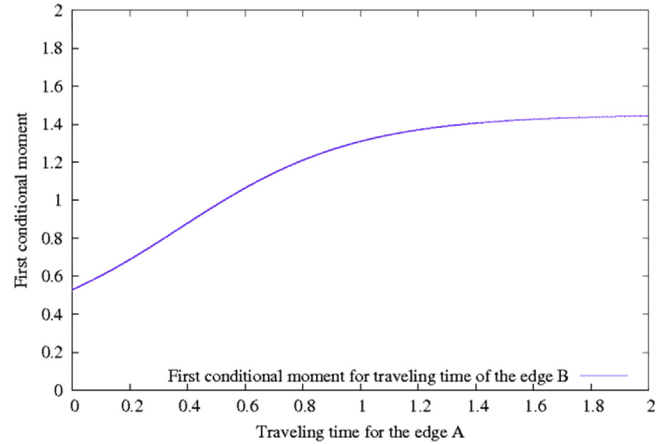


Fig. 5. The expected traveling time of the adjacent edge B depending on the weight of edge A.

$\psi_{(A, 0.5, B, 0)} = \pi_A e^{D_A 0.5} H_{AB} e^{D_B 0}$ . We obtain  $\bar{\psi}_{(A, w, B, 0)} = (0.5052, 0.4064, 0.0884, 0)$ . The conditional weights of the remaining path through the adjacent edge B can be computed from an absorbing CTMC with a generator matrix containing only the generator  $D_B$ . The first conditional moment is  $\bar{\psi}_{(A, 0.5, B, 0)} M^B \mathbf{1} = 0.9752$  in this example. For  $w \in [0, 2]$ , the values of the first conditional moment of the traveling time for the adjacent edge B are shown in Fig. 5.

### 6. Shortest path computation

For the computation of the shortest path between  $v_{ini}$  and  $v_{fin}$ , we use a continuous time Markov decision process (CTMDP) [6,40]. The state space of the CTMDP is given by tuples  $(i, x)$  where  $i \in \mathcal{E}$  and  $x \in \{1, \dots, n_i\}$  is the current phase of the PHD of edge  $i$ . For notational convenience let  $m$  be the number of edges and assume that edges are numbered consecutively. We add an additional state  $(0, 0)$  which will be used as the final absorbing state of the CTMDP. Then the state space  $\mathcal{S}$  of the CTMDP is defined as

$$\mathcal{S} = \{(i, x) \mid i \in \{0, \dots, m\}, x \in \{1, \dots, n_i\} \text{ if } i > 0 \text{ and } 0 \text{ otherwise}\} \tag{19}$$

$\mathcal{S}$  contains  $n = \sum_{i=1}^m n_i + 1$  states. Define  $\mathcal{S}_t = \mathcal{S} \setminus \{(0, 0)\}$  as the set of states without the absorbing state. Let  $U(i, x)$  be the set of possible decisions in state  $(i, x)$  and  $q^u((i, x), (j, y))$  is the transition rate from state  $(i, x)$  into state  $(j, y)$  under decision  $u \in U(i, x)$ . Furthermore, define

$$q^u((i, x), (i, x)) = - \sum_{(j, y) \in \mathcal{S}, (j, y) \neq (i, x)} q^u((i, x), (j, y)) \tag{20}$$

In state  $(0, 0)$  there is only a single decision  $u$  and the transition rates under decision  $u$  are  $q^u((0, 0), (j, y)) = 0$  for all  $(j, y) \in \mathcal{S}$  since state  $(0, 0)$  is absorbing. For state  $(i, x)$  the set  $U(i, x) = \{j \mid j \in i \bullet\}$  if  $i \bullet \neq \emptyset$ , i.e., the set of all edges which can be successors of edge  $i$ ,

and  $U(i, x) = \{0\}$ , if  $i \bullet = \emptyset$ . We can use the notation  $U(i)$  rather than  $U(i, x)$  because the set does not depend on  $x$ . The transition rates for  $j \in U(i)$  are defined as follows:

$$q^j((i, x), (h, y)) = \begin{cases} \mathbf{D}_i(x, y) & \text{if } h = i, \\ \mathbf{H}_{ij}(x, y) & \text{if } h = j > 0, \\ -\sum_{y=1}^{n_i} \mathbf{D}_i(x, y) & \text{if } h = j = 0 \ (\Rightarrow y = 0), \\ 0 & \text{otherwise.} \end{cases} \quad (21)$$

The PHG defines a CTMDP with an absorbing state. We consider the computation of an optimal path in two different scenarios, which have already been described in the previous section for the analysis of a path. First, a path with a minimal expected weight starting in  $v_{ini}$  and ending in  $v_{fin}$  should be computed. Second, we compute a path that maximizes the probability of reaching  $v_{fin}$  with a weight of at most  $w$ . If we interpret the weights as traveling times, this corresponds to a path that maximizes the probability of reaching the destination in the interval  $[0, w]$ .

We begin with the first problem that corresponds to the stochastic shortest path problem (see [6, Section 2]). First uniformization is applied to transform the CTMDP into a discrete time Markov decision process (DTMDP) [6, Section 5.1] or [44].

Let  $\alpha \geq \max_{(i,x) \in S} (\max_{u \in U(i)} (|q^u((i, x), (i, x))|))$  and define

$$p^u((i, x), (j, y)) = \begin{cases} q^u((i, x), (j, y))/\alpha & \text{if } (i, x) \neq (j, y) \\ 1 + q^u((i, x), (j, y))/\alpha & \text{if } (i, x) = (j, y) \end{cases} \quad (22)$$

The transformation implies that  $p^u((i, x), (j, y)) \geq 0$  and  $\sum_{(j,y) \in S} p^u((i, x), (j, y)) = 1$  for all  $(i, x) \in S$  and  $u \in U(i)$ . Consequently, the new values define a DTMDP. We choose reward values for the states such that the reward for state  $(i, x)$  is 1 for  $(i, x) \neq (0, 0)$  and 0 for  $(i, x) = (0, 0)$ .

A policy assigns to each state  $(i, x) \in S$  a decision  $u \in U(i)$ . If the policy depends only on the state, it is denoted as *stationary*.  $\mathcal{U}$  is the set of all stationary policies. Stationary policies can be defined by vector  $\mathbf{u}$  such that  $\mathbf{u}(i, x) \in U(i)$ . Let  $\mathbf{P}^{\mathbf{u}}$  be a  $|S_t| \times |S_t|$  matrix with  $\mathbf{P}^{\mathbf{u}}((i, x), (j, y)) = p^{\mathbf{u}(i, x)}((i, x), (j, y))$  for  $(i, x), (j, y) \in S_t$ . Depending on the choice of  $\mathbf{u}$ , matrix  $\mathbf{P}^{\mathbf{u}}$  can be stochastic or substochastic.

A policy is *proper* if it reaches the absorbing state with probability 1, i.e.,  $\sum_{k=0}^{\infty} (\mathbf{P}^{\mathbf{u}})^k$  is finite. In this case,  $\mathbf{N}^{\mathbf{u}} = (\mathbf{I} - \mathbf{P}^{\mathbf{u}})^{-1} = \sum_{k=0}^{\infty} (\mathbf{P}^{\mathbf{u}})^k$  exists and  $\mathbf{N}^{\mathbf{u}}((i, x), (j, y))$  is the mean number of visits of state  $(j, y)$  before the absorbing state is reached starting from state

$(i, x)$  (see [27]). Then

$$\xi^{\mathbf{u}}(i, x) = \sum_{(j,y) \in S} \mathbf{N}^{\mathbf{u}}((i, x), (j, y)) \quad (23)$$

is the mean number of steps before entering the absorbing state from  $(i, x)$  and  $\xi^{\mathbf{u}}(i, x)/\alpha$  is the expected weight of the corresponding CTMC resulting from policy  $\mathbf{u}$  applied to the CTMDP.

A proper policy exists because each matrix  $\mathbf{D}_i$  is non-singular and  $\lim_{t \rightarrow \infty} e^{\mathbf{D}_i t} = \mathbf{0}$  such that the PHD for edge  $i$  is eventually left and starting from an arbitrary edge, a path to the vertex  $v_{fin}$  that corresponds to the absorbing state exists by assumption.

If policy  $\mathbf{u}$  does not assure that from every state the absorbing state is eventually reached, then  $\sum_{k=0}^{\infty} (\mathbf{P}^{\mathbf{u}})^k$  does not converge which implies that some of the elements in the resulting matrix converge to infinity for  $k$  going to infinity. In this case policy  $\mathbf{u}$  is denoted as *improper*.

We can now use the results of [6] to compute the optimal stationary policy and the corresponding weight. The two conditions, that a proper policy exists and for each improper policy some values in the matrix tend to infinity are necessary and sufficient for the existence of an optimal stationary policy (see [6, Section 2]).

For a stationary policy  $\mathbf{u}$  let

$$\mathbf{w}^{\mathbf{u}} = \sum_{k=0}^{\infty} (\mathbf{P}^{\mathbf{u}})^k \mathbf{1} = \mathbf{N}^{\mathbf{u}} \mathbf{1} \quad (24)$$

be the weight vector. Element  $\mathbf{w}^{\mathbf{u}}(i, x)$  is the expected weight of the path from state  $(i, x)$  to  $(0, 0)$  under policy  $\mathbf{u}$ . If the policy is proper, then  $\mathbf{w}^{\mathbf{u}}$  can be computed as the solution of

$$(\mathbf{I} - \mathbf{P}^{\mathbf{u}}) \mathbf{w}^{\mathbf{u}} = \mathbf{1}. \quad (25)$$

Let  $\mathbf{u}^*$  be an optimal policy with weight vector  $\mathbf{w}^*$ , then

$$\mathbf{w}^*(i, x) = \min_{\mathbf{u} \in \mathcal{U}} (\mathbf{w}^{\mathbf{u}}(i, x)). \quad (26)$$

$\mathbf{u}^*$  and  $\mathbf{w}^*$  can be computed using linear programming, value iteration or policy iteration [6]. We briefly describe the computation via policy iteration which is often the most efficient approach.

Initially we define some proper policy which for example can be derived by running a shortest path algorithm on the problem where PH distributions are substituted by their expectations and correlations are neglected. Let  $\mathbf{u}^{(0)}$  be the corresponding policy. Then the following algorithm is used.

---

**Algorithm 1.** To minimize the mean weight.

Input: Set of policies  $\mathcal{U}$  and matrices  $\mathbf{P}^{\mathbf{u}}$ ;

Output: Weight vector  $\mathbf{w}^*$  and policy  $\mathbf{u}^*$ ;

$k=0$ ;

find a proper initial policy  $\mathbf{u}^{(k)}$ ;

repeat

    solve  $(\mathbf{I} - \mathbf{P}^{\mathbf{u}^{(k)}}) \mathbf{w}^{(k)} = \mathbf{1}$ ;

$k=k+1$ ;

    compute an improved policy  $\mathbf{u}^{(k)}$  for all  $(i, x) \in S$  as

$$\mathbf{u}^{(k)}(i, x) = \arg \min_{u \in U(i, x)} \left( \sum_{(j,y) \in S} p^u((i, x), (j, y)) \mathbf{w}^{(k-1)}(j, y) \right);$$

until  $\mathbf{u}^{(k)} = \mathbf{u}^{(k-1)}$ ;

$\mathbf{w}^* = \mathbf{w}^{(k)}$  and  $\mathbf{u}^* = \mathbf{u}^{(k)}$ ;

---

The algorithm computes a sequence of policies with decreasing weight vectors and converges towards an optimal policy  $\mathbf{u}^*$ . Since for improper policies some values in the weight vector become infinite and we start with a proper policy, all computed policies are proper. The effort of the algorithm results from the solution of systems of linear equations for new policies. These systems can be solved with a direct or an iterative solver. With a direct solver the effort is cubic in the number of states, whereas an iterative solver requires a variable number of iterations which is often fairly small if the iteration is started with the solution vector resulting from the previous policy. We will consider an example in Section 7.

The final step is the computation of the edge where the shortest path starts. Define for  $i \in \mathcal{E}_{ini}$   $\mathbf{a}^i = (\mathbf{0}_{<i}, \boldsymbol{\pi}_i, \mathbf{0}_{>i})$  where  $\mathbf{0}_{<i}$  is a zero row vector of length  $\sum_{j \in \mathcal{E}_{j < i}} n_j$  and  $\mathbf{0}_{>i}$  is a zero row vector of length  $\sum_{j \in \mathcal{E}_{j > i}} n_j$ . Then the initial edge is chosen as

$$\tilde{i}^* = \arg \min_{i \in \mathcal{E}_{ini}} (\mathbf{a}^i \mathbf{w}^*) \quad (27)$$

and the weight of the shortest path equals

$$\xi^* = \mathbf{a}^{\tilde{i}^*} \mathbf{w}^* \quad (28)$$

which equals  $\xi^*/\alpha$  in the original CTMDP.

The decisions resulting from the algorithm depend on the states of the PHD which are not part of the real system. In the real system decisions have to depend on accumulated weights using the vectors  $\bar{w}_{(i_1, w_1, \dots, i_L, w_L)}$  computed with (17) and (18). The subsequent edge at vertex  $fin(i_L)$  after passing the edges  $i_1, \dots, i_L$  with weights  $w_1, \dots, w_L$  is then computed as

$$\begin{aligned} \tilde{i}^* = \arg \min_{j \in U(i_L)} & \left( \sum_{x=1}^{n_{i_L}} \bar{w}_{(i_1, w_1, \dots, i_L, w_L)}(i_L, x) \right. \\ & \left. \times \left( \sum_{y=1}^{n_j} p^j((i_L, x), (i_L, y)) \mathbf{w}^*(i_L, y) + \sum_{y=1}^{n_j} p^j((i_L, x), (j, y)) \mathbf{w}^*(j, y) \right) \right) \end{aligned} \quad (29)$$

The initial edge  $\tilde{i}^*$  depends on the weights of edges that have been passed and it is independent of the states of PHDs.

The second problem, namely finding the maximal probability to reach  $v_{fin}$  with a weight less or equal to  $w$ , is harder to solve. One can approximate the optimal policy using discretization based on the results of [30]. Define a discretization factor  $\Delta = w/K$  for some  $K > 0$ . Then define for  $(i, x) \in \mathcal{S}$  and  $u \in U(i)$

$$r^u((i, x), (j, y)) = \begin{cases} \Delta q^u((i, x), (j, y)) & \text{if } (i, x) \neq (j, y) \\ 1 + \Delta q^u((i, x), (i, x)) & \text{if } (i, x) = (j, y). \end{cases} \quad (30)$$

For policy  $\mathbf{u} \in \mathcal{U}$  define matrix  $\mathbf{R}^{\mathbf{u}}((i, x), (j, y)) = r^{\mathbf{u}(i, x)}((i, x), (j, y))$ . Observe that matrix  $\mathbf{R}^{\mathbf{u}}$  is a matrix over  $\mathcal{S}$  rather than  $\mathcal{S}_c$ , i.e., the absorbing state is included. If  $\Delta$  is small enough,  $\mathbf{R}^{\mathbf{u}}$  is a stochastic matrix and

$$e^{\Delta \mathbf{Q}^{\mathbf{u}}} = \sum_{k=0}^{\infty} \frac{\Delta \mathbf{Q}^{\mathbf{u}}}{k!} = \mathbf{R}^{\mathbf{u}} + o(\Delta^2). \quad (31)$$

Thus, matrices  $\mathbf{R}^{\mathbf{u}}$  can be used to approximate the optimal policy and probability. Let  $\mathbf{z}^{(k)}$  be a vector with  $\mathbf{z}^{(k)}(i, x) = 0$  for  $(i, x) \neq (0, 0)$  and  $\mathbf{z}^{(k)}(0, 0) = 1$ . Then the following algorithm can be applied.

**Algorithm 2.** To maximize the weight in a given time interval.

Input: Set of policies  $\mathcal{U}$ , matrices  $\mathbf{R}^{\mathbf{u}}$  and vector  $\mathbf{z}^{(k)}$ ;  
 Output: Weight vector  $\mathbf{z}^{(1)}$  and policy  $\mathbf{u}^{(k)}$  for  $k = 1, \dots, K-1$ ;  
 for  $k=K-1$  downto 1  
   compute policy  $\mathbf{u}^{(k)}$  for all  $(i, x) \in \mathcal{S}$  as  
    $\mathbf{u}^{(k)}(i, x) = \arg \max_{u \in \mathcal{U}(i, x)} \left( \sum_{(j, y) \in \mathcal{S}} r^u((i, x), (j, y)) \mathbf{z}^{(k+1)}(j, y) \right)$ ;  
    $\mathbf{z}^{(k)} = \mathbf{R}^{\mathbf{u}^{(k)}} \mathbf{z}^{(k+1)}$ ;

The initial edge is chosen as

$$\tilde{i}^* = \arg \max_{i \in \mathcal{E}_{ini}} (\mathbf{a}^i \mathbf{z}^{(1)}) \quad (32)$$

and the maximal probability is given by

$$\chi^* = \mathbf{a}^{\tilde{i}^*} \mathbf{z}^{(1)}. \quad (33)$$

Using the results of [30] it can be shown that the computed value and policy converges towards the optimal values of the continuous problem for  $\Delta \rightarrow 0$  which means  $N \rightarrow \infty$ . The resulting policy depends on the state and the remaining weight until the final state has to be reached.

A more efficient approach to compute the optimal policy is based on uniformization and allows one to compute lower and upper bounds for  $\chi^*$  and a policy that reaches at least the lower bound [15]. However, since this algorithm is more complex, we do not present it and refer for details to the literature.

Again, the decision in the real system cannot be made based on the detailed state of a PHD and instead has to be based on vector  $\bar{w}_{(i_1, w_1, \dots, i_L, w_L)}$ . Since we apply a discretization approach, the values  $w_l$  have to be multiples of step  $\Delta$ . In practice this means that measured values have to be rounded which is not critical if  $\Delta$  is small enough. We assume that  $w_l = k_l \Delta$  for  $k_l \in \mathbb{N}$ . Let  $k = \sum_{l=1}^L k_l$ , then the optimal decision at  $k\Delta$  can be approximated using (29) with vector  $\mathbf{z}^{(k)}$  rather than  $\mathbf{w}^*$ .

## 7. Complexity issues

To use the approach for real applications, the computational effort has to be moderate. This implies that models can be parameterized and analyzed within a few minutes at most. The whole approach consists of different steps for which often alternative algorithms exist. We briefly analyze the effort and start with the algorithms for building the model which means that the parameters for the PHDs and the transfer matrices have to be computed from measured values. Afterwards algorithms for shortest path computation are analyzed.

For parameterization usually data is available from measurements or simulations. Depending on the application area such traces contain between a few hundred and several millions of entries. As shown in Section 4, algorithms for the computation of PHD parameters can either use moments and joint moments or the whole trace in an EM algorithm.

The effort for computing empirical moments and joint moments from traces of length  $d$  is in  $O(d)$ . After moments and joint moments are available, the remaining computations are independent of the trace length. If PHDs parameters are fitted according to the first 2 or 3 moments, then closed form expressions are available [8] such that the computation time is negligible. If more than three moments should be matched, then the exact but not always feasible approach from [25] or the approximation approach from [14] have to be used. The former requires some numerical computations with an effort growing exponentially in the number of phases. The latter uses an iterative approach that includes a non-linear optimization problem with  $n$  variables for a PHD of order  $n$ . Since the approach is iterative, an exact analysis of the effort is impossible. However, for moderate  $n$  (values of 10 or less), the computation requires in both cases usually only a few seconds.

Computation of the elements of the transfer matrix to match the joint moments requires the solution of a linear non-negative least squares (NNLS) problem (Eq. (8)). The standard algorithm for the solution of NNLS problems is iterative and has been published in



[29]. Although the algorithm is guaranteed to find the exact solution, up to numerical inaccuracies, the effort has not been analyzed yet. However, the algorithm is usually extremely fast if the number of variables is not too large. As an example we consider the computation of a transfer matrix of order 10 according to the joint moments  $E(X_i^k, X_j^l)$  ( $k, l = 1, \dots, 9$ ) which requires 0.2 s on a standard PC. Moment based fitting uses usually only a small number of moments or joint moments and is then very efficient. Therefore it can be applied for small PHDs with less than 10 phases even in online situations where the weights are computed immediately after new measurements become available.

The effort for EM algorithms is significantly higher than the effort for moment based fitting methods. However, EM methods approximate the measured density such that they often yield better results for multimodal densities or if the correlation structure contains multiple peaks which can only roughly be approximated by fitting methods based on lower order moments or joint moments. The effort for EM algorithms depends linearly on the number of elements in the trace. If the elements in the transfer matrix are generated by an EM algorithm as in (10), then the effort is also linear in the number of non-zero elements in matrix  $H_{ij}$ . Furthermore, (10) contains two matrix exponentials  $\pi_i e^{w_i D_i}$  and  $e^{w_j D_j} (-D_j) \mathbb{1}$  which have to be computed for each trace element. Computation of  $\pi e^{wD}$  requires  $O(w \max_x |D(x, x)|)$  iterations with an effort of  $O(nz(D))$ , where  $nz(D)$  is the number of non-zero entries in  $D$ , using uniformization [41]. Thus, the overall effort for computing  $H_{ij}$  is in  $O(it \cdot d_{ij} \cdot (nz(H_{ij}) + w_i \max_x |D_i(x, x)| nz(D_i) + \max_y |D_j(y, y)| nz(D_j)))$ , where  $it$  is the number of EM iterations that depends on the required stopping criterion and the structure of the problem. EM algorithms can also be applied to determine the matrices  $D_i$ . In this case the effort is in  $O(it \cdot d_i \cdot nz(D_i) \cdot w_i \max_x |D_i(x, x)|)$  and can be reduced for specific PH distributions like Hyper-Erlang [50] or by first aggregating the elements of the trace [37].

The proposed fitting methods all compute parameters for a PHD of a given order. To match the first 2 or 3 moments, two phases are sufficient if the coefficient of variation is not too small. If an EM algorithm is used to maximize the likelihood, then a larger number of phases usually result in a larger likelihood value but a larger number of phases also result in longer runtimes of the EM algorithm and more states in the DTMDP. Thus, a compromise between fitting quality and effort has to be found. A heuristic approach starts the EM algorithm with a small number of phases and increases the number of phases until the differences in the likelihood values become small or a maximal number of phases has been reached. We apply this approach in the following example and use the software *gfit* [50] for parameter fitting.

As an example we consider the traveling times of vehicles on one of the main streets of Cologne which have been derived from the dataset given in [1] (see also the example presented in Section 8).

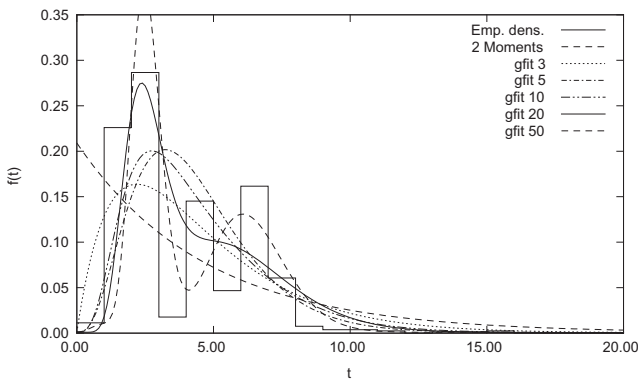


Fig. 6. Empirical density and densities of some PHDs.

Table 1

Moments, likelihood values and fitting times for different PHDs that are fitted according to the traveling time.

$n$	$E(T)$	CV	Skew	Log likel	Time	Log likel	Time
Moment fitting							
2	-	-	-	-	-	-2066.0	0.00
				EM without moment fitting		EM with mom. fit.	
3	5.29	1.69	11.52	-1881.4	0.14	-1887.2	0.16
4	5.29	1.62	11.26	-1820.4	0.21	-1829.6	0.23
5	5.29	1.57	10.97	-1806.9	0.27	-1818.1	0.30
6	5.29	1.45	9.06	-1806.6	0.32	-1809.8	0.35
7	5.29	1.41	8.22	-1806.1	0.35	-1807.8	0.38
8	5.29	1.52	10.09	-1805.8	0.63	-1811.7	0.66
9	5.29	1.59	11.13	-1805.1	1.01	-1816.1	1.05
10	5.29	1.62	11.27	-1796.8	1.57	-1807.3	1.61
12	5.29	1.64	11.36	-1780.0	2.60	-1790.0	2.64
15	5.29	1.63	11.28	-1756.3	3.87	-1767.0	3.92
20	5.29	1.60	10.99	-1725.9	10.68	-1737.8	10.74
25	5.29	1.57	10.69	-1703.1	25.19	-1717.1	25.92
30	5.29	1.53	10.32	-1687.2	46.37	-1702.2	46.45
40	5.29	1.53	10.25	-1669.8	128.48	-1687.4	128.59
50	5.29	1.52	10.10	-1661.1	328.24	-1669.9	328.36

The trace contains traveling times of 792 cars. The empirical density of the traveling time is shown in Fig. 6. The expectation, coefficient of variation and skewness of the traveling time are 5.29, 1.44 and 7.68, respectively.

The first three moments of the trace can be matched exactly by a PHD with two phases. Additionally, we use the EM-algorithm of the tool *gfit* to fit the parameters of Hyper-Erlang distributions with a growing number of phases. After the Hyper-Erlang distribution is available, it is possible to adjust the parameters to fit the first three moments of the trace. This step usually reduces the likelihood value slightly and is described in [50]. The first moment is always exactly matched by the PHD resulting from an EM algorithm. Table 1 contains the results. The first column includes the number of phases. In the columns 2–6 the results for the EM algorithms without subsequent moment fitting are shown. In the last two columns the results for the EM algorithm with subsequent moment fitting ( $n=3, \dots, 50$ ) and a simple moment fitting (case  $n=2$ ) are presented. With moment fitting in all cases the first three moments and thus also the coefficient of variation and the skewness are matched exactly by the PHDs. Therefore these values are not shown in the table. The column time contains the runtimes of the algorithm in seconds on a PC with a 2.5 GHz Quad-Core processor and 4 GB main memory. It can be seen that within 10 s the EM algorithm can compute PHDs with 1–12 phases and within 1 min PHDs with 1–20 phases are generated. If the number of phases grows, then also the effort grows. The runtime of the algorithm can probably be further reduced by restricting the search space if PHDs are generated consecutively from the same data set.

For the selection of an appropriate PHD, the values of the likelihood function should be analyzed. The likelihood value for the pure moment fitting is much worse than the likelihood values of the PHDs resulting from the EM algorithm. The likelihood values of the PHDs resulting from the EM algorithm increase with the number of phases but the gradient varies significantly. Thus, the steps from 3 to 4 and then to five phases result in a better fitting. Afterwards from 6 to 9 phases the likelihood remains more or less the same. The next step in the likelihood can be observed for 10 phases. The flexibility of PHDs allows one to choose from a set of distributions. In our example the PHDs with 3, 5 or 10 phases are good choices, much better than the PHD with two phases resulting from moment fitting. The empirical density of the trace and the densities of some PHDs are shown in Fig. 6.

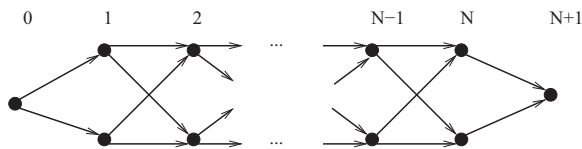


Fig. 7. Acyclic example graph for shortest path computation.

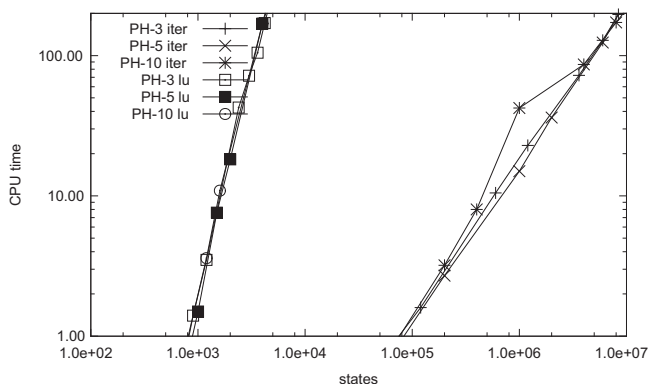


Fig. 8. Computational effort to compute the optimal policy minimizing the mean traveling time.

To evaluate the approximation quality of PHDs one can compare them with other distributions used in statistical modeling. We used the *arena input analyzer* [42] to find distributions that model the data set. The two best distributions are  $0.999 + Weibull(4.02, 1.1)$  and  $(0.999 +) Gamma(3.79, 1.13)$  where 0.999 is a constant offset. The likelihood values for these distributions are  $-1954.0$  and  $-2481.4$ . All PHDs resulting from the EM algorithm provide a larger likelihood which shows that even PHDs with a small number of phases are a valuable model for the data. This finding confirms results for failure data in [16].

The example shows that for traces with a few hundred entries, like most vehicular traffic traces, the fitting can be performed efficiently. The situation is different for traces from computer networks which may contain more than a million entries. If EM algorithms are applied without any preprocessing the effort can be huge. However, by first using trace aggregation, also in these cases PHDs of moderate order can be generated within a few seconds.

For a PHG with  $m$  edges where an edge has in the average  $s$  successor edges,  $m$  PHDs and  $ms$  transfer matrices have to be determined. Parameter fitting of the matrices  $D_i$  and afterwards of the matrices  $H_{ij}$  are independent such that the matrices can be computed in parallel. Thus, in most relevant situations PHGs can be generated from available data in an acceptable time which means that at most a few minutes are necessary to compute all required matrices.

The effort of the shortest path computation depends on the size and the structure of the CTMDP. Computation times are often moderate even for fairly large CTMDPs, in particular, if the graph is (almost) acyclic. This will be shown by a simple example.

The example graph shown in Fig. 7 contains  $N$  levels with two nodes in each level plus an initial and a final node. Nodes in the levels 0 through  $N-1$  allow choices between two outgoing edges. The weights of the edges are modeled by PHDs that have been generated with respect to the traveling times described above. Transfer matrices are generated from a convex linear combination of the transfer matrix for the minimal and maximal correlation between adjacent edges. Weights of the convex combination are randomly selected for every pair of adjacent edges. The example model with  $N$  level contains  $2N+2$  nodes and  $4N$  edges. If weights

are modeled with PHDs with  $n_i$  phases, the resulting CTMDP has  $4N \cdot n_i$  states.

First, the computational effort of Algorithm 1 in Section 6 to minimize the mean traveling time from the initial to the final node is analyzed for the graph with a growing number of levels and weights modeled by PHDs with 3, 5 and 10 phases. Two versions of the algorithm are considered. In the first version (*mean direct*) the linear equations are solved using an LU-decomposition with an effort cubic in the number of states. The required time to compute the optimal policy is shown in Fig. 8. It is also possible to solve the linear equations in Algorithm 1 by an iterative technique. In our implementation we applied GMRES with an ILU0 preconditioner [43]. This method is extremely fast for the equations occurring in the example. Policy iteration requires less than 20 policy iterations to find the optimal policy and GMRES requires very few, usually less than 10, iterations to compute the solution for the set of equations that results from a new policy, if the method is started with the solution vector of the previous policy. The effort of the direct method depends only on the number of states of the CTMDP and not on the number of phases of the PHDs. The effort for the iterative technique is also mainly determined by the number of states but it can be seen that for the PHD with 10 phases, GMRES requires slightly more iterations which slows down the solution in some cases. With the iterative technique within 200 s optimal policies for CTMDPs with almost 10 million states can be computed. This corresponds to a graph with 1 million edges if edge weights are modeled by PHDs with 10 phases. However, it should be mentioned that the example is acyclic and optimal policies can be computed very efficiently. In graphs with many cycles the effort is higher but normally still large graphs can be handled.

The second result which is computed is the maximal probability to reach the final node from the initial node within the expected traveling time under a random selection of the successor edge. With a growing number of levels also the time horizon grows in this case. Algorithm 2 is used to compute the probability. In the algorithm, discrete steps are used and the effort depends linearly on the inverse of the length of the discretization step. The curves in Fig. 9 show the effort for different choices of the discretization interval length and for PHDs with 3, 5 and 10 phases describing the weights of the edges. The computational effort depends on the number of states of the CTMDP, the size of discretization step and the number of phases of the PHD. For a larger number of phases, the method becomes faster if we fix the other two parameters. The reason is that for a larger number of phases and a fixed number of states the number of levels and also the expected time to reach the destination is smaller. In general, the computation of an optimal policy for the finite horizon case

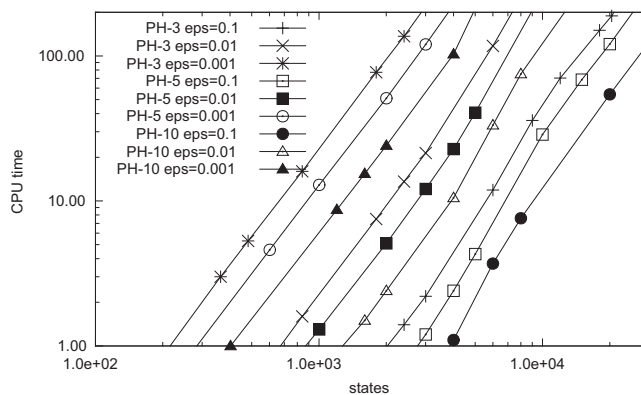


Fig. 9. Computational effort to compute the optimal policy maximizing the probability of reaching the final location in less than the expected traveling time under a random policy.

$$\pi_A = (0.01614, 0.49193, 0, 0.24597, 0, 0.24597),$$

$$D_A = \begin{pmatrix} -0.01968 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.21991 & 0.21991 & 0 & 0 & 0 \\ 0 & 0 & -0.43981 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.43981 & 0.43981 & 0 \\ 0 & 0 & 0 & 0 & -0.87962 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.87962 \end{pmatrix}$$

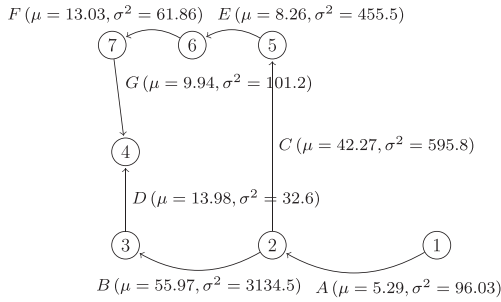


Fig. 10. The PHG with  $\mathcal{E} = \{A, B, C, D, E, F, G\}$ .

(with Algorithm 2) requires more effort than the computation of the expected weight (with Algorithm 1).

Of course, the results of experiments for one example should not be generalized. However, results for this and similar examples show that the available methods allow one to compute optimal policies for fairly large state spaces in a moderate time of a few minutes at most. This means that the models can be applied in many realistic situations and even in some online situations.

### 8. A realistic example

We present the shortest path computation in a real traffic network containing some of the main streets of the city Cologne, namely the Niehler Strasse, Neusser Strasse, and the Innere Kanalstrasse where congestion often occurs. The weights of the edges describe traveling times of vehicles and have been derived from the dataset given in [1]. The PHG has two paths from initial node 1 to the destination node 4 and is shown in Fig. 10.

The destination can be reached via the initial edge A and two adjacent edges B or C plus some successor edges to reach the final destination. The weight of the edge A is correlated with the weight of the adjacent edge B. Thus, edge B is usually congested if the predecessor edge A is congested. The edge C is assumed to be a detour and its weight is independent of the weight of the predecessor edge A.

The weights from the dataset were used for fitting PHDs of order  $n_i$ , ( $n_i = 1, \dots, 20$ ), using the software *gfit* (see, e.g., [50]). The values of the log-likelihood function according to the traces are shown in Fig. 11.

The PH-approximation becomes better with an increasing number of phases which is represented by the rise in the curves shown in Fig. 11. However, the tendency is that the largest slope can be observed up to six phases, i.e. adding additional phases ( $n > 6$ ) increases the log-likelihood values only slightly. The log-likelihood of the PHD for the initial edge A converges against  $-2063.92251$  as shown in Fig. 11. We also tried other fitting tools like *momfit* or *phfit* [14,24], but only *gfit* generates the hyperexponential representation which has the largest flexibility for the subsequent fitting according to the correlation [13]. From Fig. 11 we conclude that the traces are fitted adequately by PHDs with six phases.

Table 2 shows the maximum value of the coefficient of correlation which can be modeled if weights of edge A and B are described by

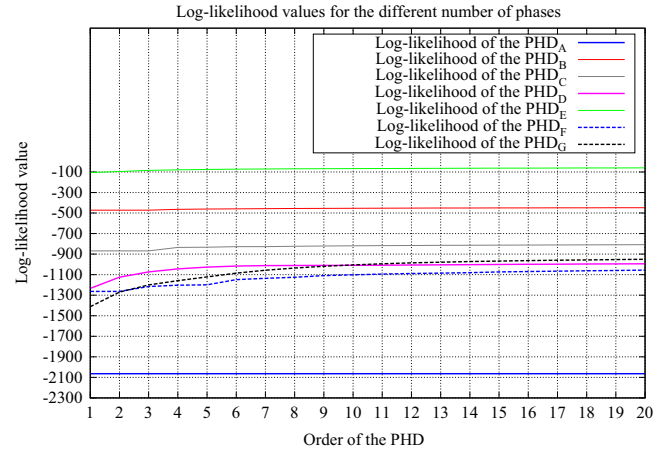


Fig. 11. Log-likelihood values for PHDs of order  $i$ ,  $i = 1, \dots, 20$ .

Table 2

Impact of the PHD order on the correlation for the PHD  $(\pi_A, D_A)$  and the PHD  $(\pi_B, D_B)$ .

PHD order	$\rho_{AB}$
2	1.3643e-05
3	0.027237
4	0.18756
5	0.18758
6	0.19576
9	0.19577
11	0.19726
13	0.19751
15	0.19754

PHDs of an increasing order. Again it can be seen that with an increased number of phases the coefficient of correlation that can be represented grows. Again for a number of phases  $n$  between 6 and 20 no major improvement in the reachable coefficient of correlation can be achieved which is again an argument to choose PHDs of order 6 for the example.

The following acyclic PHDs of order 6 are computed according to the traffic traces of the edges A through G:

$$\pi_A = (0.01614, 0.49193, 0, 0.24597, 0, 0.24597),$$

$$D_A = \begin{pmatrix} -0.01968 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.21991 & 0.21991 & 0 & 0 & 0 \\ 0 & 0 & -0.43981 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.43981 & 0.43981 & 0 \\ 0 & 0 & 0 & 0 & -0.87962 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.87962 \end{pmatrix}$$

$$\pi_B = (0.47669, 0.14445, 0.18822, 0.09532, 0, 0.09532),$$

$$D_B = \begin{pmatrix} -0.01786 & 0.01786 & 0 & 0 & 0 & 0 \\ 0 & -0.07680 & 0.07680 & 0 & 0 & 0 \\ 0 & 0 & -0.07680 & 0.03840 & 0 & 0.03840 \\ 0 & 0 & 0 & -0.09377 & 0.09377 & 0 \\ 0 & 0 & 0 & 0 & -0.18755 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.18755 \end{pmatrix}$$

$$\pi_C = (0.99999, 0, 0, 3.98773e-06, 0, 0),$$

$$D_C = \begin{pmatrix} -0.07095 & 0.07095 & 0 & 0 & 0 & 0 \\ 0 & -0.07095 & 0.07095 & 0 & 0 & 0 \\ 0 & 0 & -0.07095 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.18958 & 0.18958 & 0 \\ 0 & 0 & 0 & 0 & -0.18958 & 0.18958 \\ 0 & 0 & 0 & 0 & 0 & -0.18958 \end{pmatrix}$$

$$\pi_D = (1, 0, 0, 0, 0, 0),$$

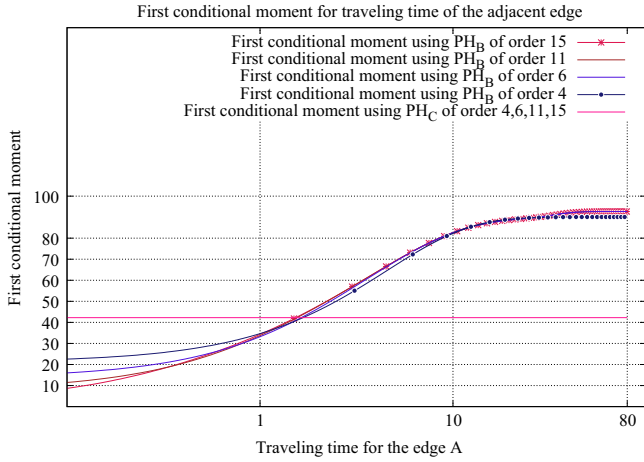


Fig. 12. The expected traveling time for the adjacent edges B, and C depending on the weight of edge A.

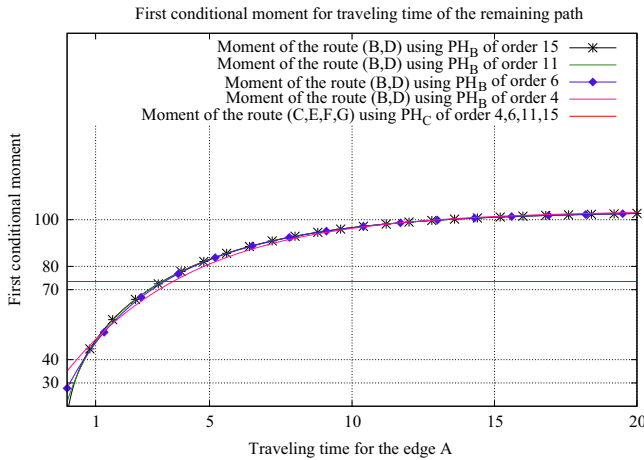


Fig. 13. The expected traveling time for the paths B, D, and C, E, F, G in dependence of the weight of edge A.

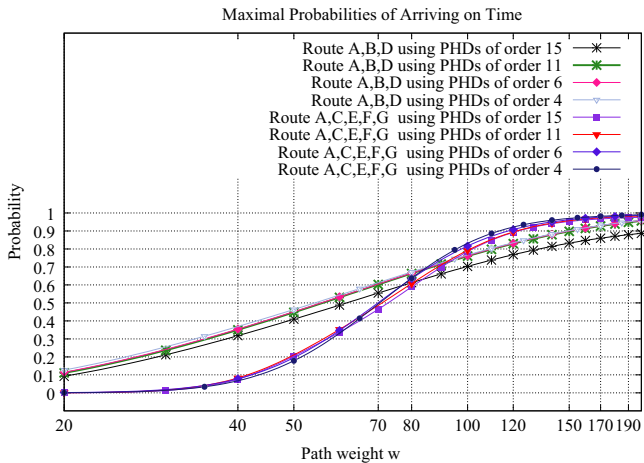


Fig. 14. The probabilities of arriving at the destination node 4 with a path weight less than or equal to w.

$$D_D = \begin{pmatrix} -0.42893 & 0.42893 & 0 & 0 & 0 & 0 \\ 0 & -0.42893 & 0.42893 & 0 & 0 & 0 \\ 0 & 0 & -0.42893 & 0.42893 & 0 & 0 \\ 0 & 0 & 0 & -0.42893 & 0.42893 & 0 \\ 0 & 0 & 0 & 0 & -0.42893 & 0.42893 \\ 0 & 0 & 0 & 0 & 0 & -0.42893 \end{pmatrix}$$

$$\pi_E = (0.09701, 0.90298, 0, 0, 0, 0),$$

$$D_E = \begin{pmatrix} -0.01951 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.37078 & 1.37078 & 0 & 0 & 0 \\ 0 & 0 & -1.37078 & 1.37078 & 0 & 0 \\ 0 & 0 & 0 & -1.37078 & 1.37078 & 0 \\ 0 & 0 & 0 & 0 & -1.37078 & 1.37078 \\ 0 & 0 & 0 & 0 & 0 & -1.37078 \end{pmatrix}$$

$$\pi_F = (0.16103, 0.83897, 0, 0, 0, 0),$$

$$D_F = \begin{pmatrix} -0.46405 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.33057 & 0.33057 & 0 & 0 & 0 \\ 0 & 0 & -0.33057 & 0.33057 & 0 & 0 \\ 0 & 0 & 0 & -0.33057 & 0.33057 & 0 \\ 0 & 0 & 0 & 0 & -0.33057 & 0.33057 \\ 0 & 0 & 0 & 0 & 0 & -0.33057 \end{pmatrix}$$

$$\pi_G = (0.00263, 0.99737, 0, 0, 0, 0),$$

$$D_G = \begin{pmatrix} -0.00768 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.51913 & 0.51913 & 0 & 0 & 0 \\ 0 & 0 & -0.51913 & 0.51913 & 0 & 0 \\ 0 & 0 & 0 & -0.51913 & 0.51913 & 0 \\ 0 & 0 & 0 & 0 & -0.51913 & 0.51913 \\ 0 & 0 & 0 & 0 & 0 & -0.51913 \end{pmatrix}$$

The coefficient of correlation between edge A and B equals  $\rho_{AB} = 0.264$ . Matrix  $H_{AB}$  is computed according to the joint moments (Eq. (8)). The resulting transfer matrix equals

$$H_{AB} = \begin{pmatrix} 0.01968 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.41176 & 0.02805 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.40437 & 0.47526 & 0 & 0 & 0 \\ 0 & 0 & 0.19786 & 0.34088 & 0 & 0.34088 \end{pmatrix}, \quad \rho_{AB} = 0.19576.$$

Now suppose that a traveler has traversed edge A and the current position is at node 2. There are two competing options for the next possible edge to visit from node 2, namely the edges B and C. In any case, the traveler's objective is to select the next edge to traverse such that the expected time until arriving at the destination node 4 is minimized. The initial proper policy equals  $u_0 = \{A, C, E, F, G\}$  and results from a simple shortest path computation neglecting distributions or correlations. The weight of policy  $u_0$  equals 73.1208, and the weight of the alternative policy  $u_1 = \{A, B, D\}$  equals 75.2639. If we take correlated travel times on adjacent edges into consideration, the optimal policy has to be determined depending on the time required to pass edge A.

The behavior of the optimal policy is as follows: for a long time required on the edge A, the best decision is to choose the adjacent edge C, since traveling on edges A and B is positively correlated. If the time required for the edge A becomes smaller, then the optimal choice is edge B. This means that from a phase where the remaining time before absorption (i.e., the remaining time to reach node 2) is longer than the average, C should be chosen as successor and otherwise B.

In the real system decisions cannot depend on the phase, they have to depend on the vectors  $\bar{\psi}_{(i_1, w_1, \dots, i_L, w_L)}$  computed with (17) and (18). We computed vectors  $\bar{\psi}_{(A, w, j, 0)}$  with  $w \in [0.1, 80]$ ,  $j \in \{B, C\}$ . The values of the first conditional moments of the traveling time for the adjacent edges are summarized in Fig. 12. Results are computed for PHDs with 4, 6, 11, and 15 phases. In Fig. 12 the notation  $PH_B$  is used for the PHD with representation  $(\pi_B, D_B)$ .

Fig. 12 shows the expected traveling times at the adjacent edges B and C depending on the traveling time at edge A. The time at C is not affected by the traveling time at A since both weights are independent. The positive correlation between the weights at the edges A and B results in a positive slope of the expected traveling time at B. It can be seen that the curves for the PHDs of different orders slightly differ. However, the difference occurs mainly for small traveling times at edge A, for larger values the curves are almost identical for all numbers of phases shown in the graph.



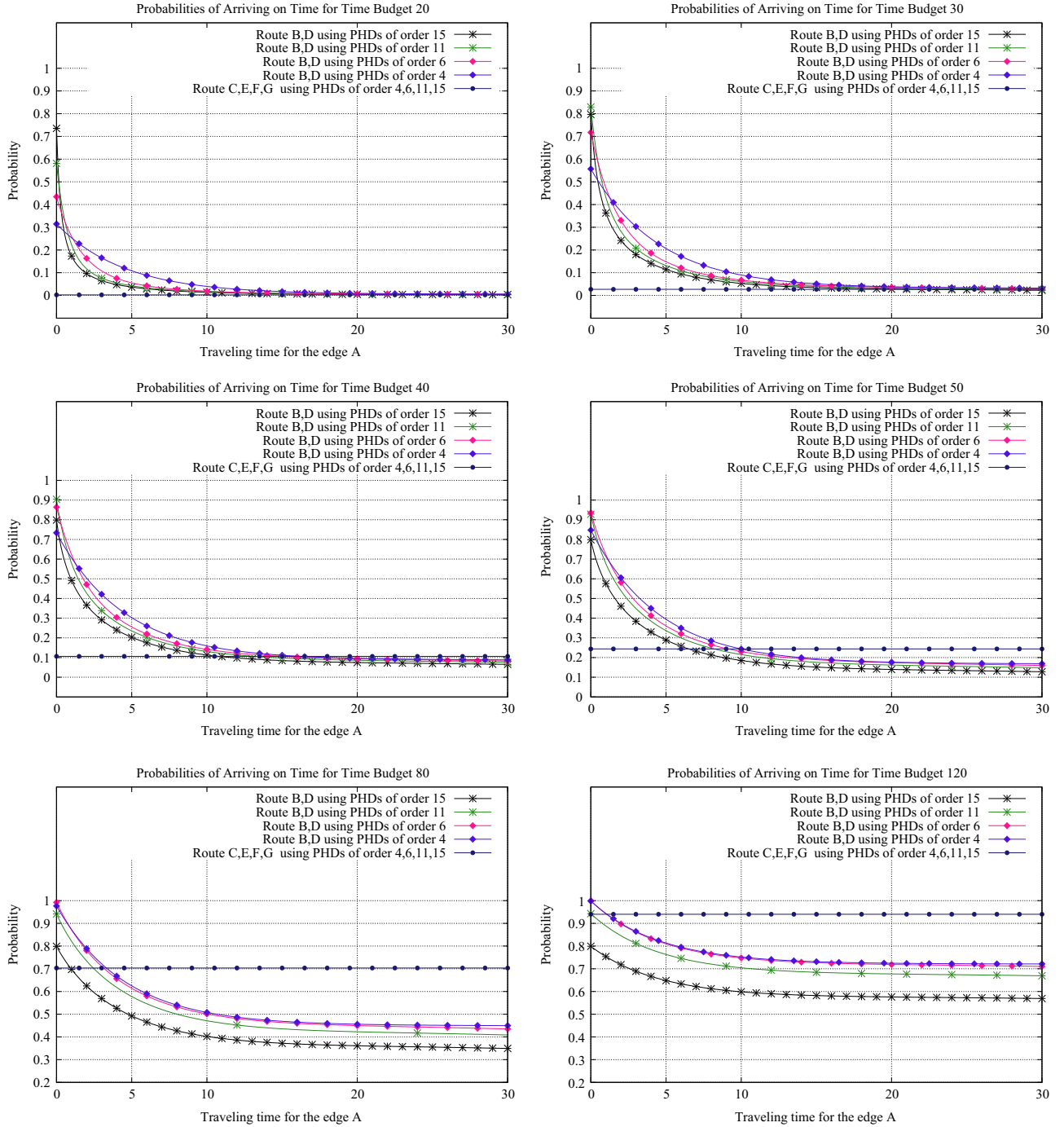


Fig. 15. Probabilities of arriving on time for different deadlines depending on the traveling time at the initial edge A.

Using the vectors  $\bar{w}(i_1, w_1, \dots, i_L, w_L)$  the conditional weights of the remaining path through the edges *B, D* and the remaining path through the edges *C, E, F, G* can be computed. The best subsequent edge at vertex 2 after passing the initial edge *A* with weight *w* can then be computed with (29) such that the decision depends on the previous weights of the edge *A* and not on the state of the PHD. The values of the conditional first moments of the traveling time for the remaining paths are shown in Fig. 13. Again it can be seen that curves for the PHDs of different orders differ slightly for small traveling times at edge *A* and are very similar for larger traveling times at *A*.

Additionally, we consider the computation of the path that maximizes the probability of reaching the destination node 4 with a weight of at least *w*. The presented model is analyzed for path

weights in the interval [20,200]. We use  $\Delta = w/N$  for a large  $N = 300,000$  and solve the resulting DTMDP for a horizon of *N* steps. The results computed with the discretization approach are shown in Fig. 14 for models with PHDs of different orders. In the examples weights at all edges are described by PHDs of a common order. Again it can be seen that there is only a small difference between the results of the examples using PHDs of different orders. The positive correlation results in a higher probability for the first path to meet a short deadline but in a slightly smaller probability to meet a long deadline.

Finally, we analyze the probabilities of meeting various deadlines via the path {*B, C*} and {*D, E, F*} depending on the time needed to pass *A*. Results are shown in Fig. 15.

Again it can be noticed that for a short traveling time via  $A$  it is better to choose the route via  $B$  and  $D$ . If the time horizon to reach the destination increases, then the traveling time via  $A$  for which is preferable to choose the route via  $B$  rather than via  $C$  becomes smaller and smaller. The reason for this behavior is the relatively large variance of the traveling time of edge  $B$ . It can also be seen that the number of phases of the PHDs has some effect on the computed probability if the traveling time at  $A$  and the time horizon are small.

The MDPs resulting from the example are very small with  $\sum_{i=1}^7 n_i + 1$  states, where  $n_i$  equals the number of phases of the PHDs for the traveling time at the  $i$ th street. Since the graph acyclic, computational times are all negligible.

## 9. Conclusions

In this paper we investigate two problems of route planning under uncertainty with dependent weights on adjacent edges of a weighted graph: finding a path with a minimal expected weight and selecting a path that maximizes the probabilities of paths with a weight below a threshold. The proposed framework of weighted graphs with correlated weights is based on phase type distributions (PHDs), Markovian arrival processes (MAPs) and continuous time Markov decision processes (CTMDPs). PHDs, a versatile class of distributions, allow the modeling of general distributions of the weights. The basic idea of MAPs, namely marked transitions to indicate events, is used to introduce correlations between the weights of adjacent edges. We denote the resulting graph model as PH graphs (PHGs) and show how the problem of finding of an optimal route in a PHG can be mapped on a CTMDP and handled with standard algorithms like policy iteration. Results of some examples indicate that the effect of correlation between edge weights should not be neglected when solving stochastic shortest path problems. Depending on the weight of the previous edge choosing a correlated adjacent edge could result in a much better path weight than the choice of an uncorrelated adjacent edge.

Our main interest in this paper was the definition of the class of PHGs and methods for the parameterization of models including dependencies that are based on measurements resulting from real systems or simulation models. For shortest path computation we applied standard approaches from stochastic dynamic programming. PHGs can be used to model various practical problems like route finding on streets, routing in computer networks or reliability analysis of systems built from components with correlated failure rates.

The approach of this paper can be extended in various directions. More sophisticated measures like discounted weights or the optimization of the average weight under some variability constraints may be used as goal functions for an optimal route by adopting methods from MDPs. It is sometimes possible to improve the algorithms for policy computation by exploiting the specific structure of the problem which often results in acyclic paths such that optimal decisions can be computed by considering one edge at a time. The model class is extendable by allowing for example time-dependent weights which result in an inhomogeneous CTMDP. It is in principle also possible to define weights that depend on the weight of more than one predecessor edge. However, this modeling requires the introduction of additional states to code the exit states from previous PHDs and results in an exponential increase of the dimension of the transfer matrices such that from a practical point of view dependencies have to be restricted.

## References

[1] Vehicular mobility trace of the city of Cologne, Germany. (<http://kolntrace.project.citi-lab.fr/>).

- [2] [Andreatta G, Romeo L. Stochastic shortest paths with recourse. Networks 1988;18\(3\):193–204.](#)
- [3] [Asmussen S. Matrix-analytic models and their analysis. Scand J Stat 2000;27\(2\):193–226.](#)
- [4] [Asmussen S, Nerman O, Olsson M. Fitting phase-type distributions via the EM-algorithm. Scand J Stat 1996;23\(4\):419–41.](#)
- [5] [Bause F, Buchholz P, Kriege J. ProFiDo—the processes fitting toolkit Dortmund. In: QEST; 2010. p. 87–96. Available from: \(\[http://ls4-www.cs.tu-dortmund.de/cms/de/forschung/projekte/dfg-maps/software\\\_dfg/profido\]\(http://ls4-www.cs.tu-dortmund.de/cms/de/forschung/projekte/dfg-maps/software\_dfg/profido\)\).](#)
- [6] [Bertsekas DP. Dynamic programming and optimal control, vol. II. Belmont MA: Athena Scientific; 2007.](#)
- [7] [Bertsekas DP, Tsitsiklis JN. An analysis of stochastic shortest path problems. Math Oper Res 1991;16:580–95.](#)
- [8] [Bobbio A, Horváth A, Telek M. Matching three moments with minimal acyclic phase type distributions. Stoch Models 2005;21:303–26.](#)
- [9] [Bonet B, Geffner H. Solving stochastic shortest-path problems with RTDP. Technical report. Universidad Simon Bolivar; 2002. See \(<http://ldc.usb.ve/~bonet/reports/rtdp.pdf>\).](#)
- [10] [Boyan J, Mitzenmacher M. Improved results for route planning in stochastic transportation networks. In: Proceedings of the symposium of discrete algorithms; 2001.](#)
- [11] [Boyan JA, Littman ML. Exact solutions to time-dependent MDPs. In: Advances in neural information processing systems. Cambridge MA: MIT Press; 2000. p. 1026–32.](#)
- [12] [Briggs AJ, Detweiler C, Scharstein D, Vandenberg-Rodes A. Expected shortest paths for landmark-based robot navigation. Int J Robot Res 2004;23:717–28.](#)
- [13] [Buchholz P, Felko I, Kriege J. Transformation of acyclic phase type distributions for correlation fitting. In: ASMTA. Lecture notes in computer science, vol. 7984. Springer; 2013. p. 96–111.](#)
- [14] [Buchholz P, Kriege J. A heuristic approach for fitting MAPs to moments and joint moments. In: Proceedings of the sixth international conference on Quantitative Evaluation of SysTems \(QEST09\). Los Alamitos CA: IEEE; 2009.](#)
- [15] [Buchholz P, Schulz I. Numerical analysis of continuous time Markov decision processes over finite horizons. Comput OR 2011;38\(3\):651–9.](#)
- [16] [Peter Buchholz, Jan Kriege. Markov modeling of availability and unavailability data. In: EDCC. Los Alamitos CA: IEEE; 2014. p. 94–105.](#)
- [17] [Dijkstra EW. A note on two problems in connexion with graphs. Numer Math 1959;1:269–71.](#)
- [18] [Fan YY, Kalaba RE, Moore JE. Shortest paths in stochastic networks with correlated link costs. Comput Math Appl 2005;49\(9–10\):1549–64.](#)
- [19] [Fischer W, Meier-Hellstern K. The Markov-modulated Poisson process \(MMPP\) cookbook. Perform Eval 1992;18\(2\):149–71.](#)
- [20] [Song G, Ismail C. Optimal routing policy problems in stochastic time-dependent networks. Transp Res Part B: Methodol 2006;40\(2\):93–122.](#)
- [21] [Gross D, Miller D. The randomization technique as a modeling tool and solution procedure for transient Markov processes. Oper Res 1984;32\(2\):926–44.](#)
- [22] [Hall RW. The fastest path through a network with random time-dependent travel times. Transp Sci 1986;20\(3\):182–8.](#)
- [23] [Hornig R, Varga A. An Overview of the OMNeT++ simulation environment. In: Proceedings of the first international conference on simulation tools and techniques for communications, networks and systems \(SIMUtools\); 2008.](#)
- [24] [Horvath A, Telek M. Phfit: a general purpose phase type fitting tool. In: Performance tools 2002. Lecture notes in computer science, vol. 2324. Heidelberg, New York, Dordrecht, London: Springer; 2002. p. 82–91.](#)
- [25] [Horvath A, Telek M. Matching more than three moments with acyclic phase type distributions. Stoch Models 2007;23:167–94.](#)
- [26] [Horvath G, Telek M, Buchholz P. A MAP fitting approach with independent approximation of the inter-arrival time distribution and the lag-correlation. In: Proceedings of the second international conference on the quantitative analysis of systems. Los Alamitos CA: IEEE; 2005.](#)
- [27] [Kemeny JG, Snell JL. Finite Markov chains. New York, Berlin, Heidelberg, Tokyo: Springer; 1976.](#)
- [28] [El Abdouni Khayari R, Sadre R, Haverkort B. Fitting world-wide web request traces with the EM-algorithm. Perform Eval 2003;52:175–91.](#)
- [29] [Lawson CL, Hanson RJ. Solving least squares problems, classics in applied mathematics. Philadelphia: SIAM; 1995.](#)
- [30] [Miller BL. Finite state continuous time Markov decision processes with a finite planning horizon. SIAM J Control 1968;6\(2\):266–80.](#)
- [31] [Neuts MF. A versatile Markovian point process. J Appl Probab 1979;16:764–79.](#)
- [32] [Neuts MF. Matrix geometric solutions in stochastic models: an algorithmic approach. New York: Dover Publications Inc.; 1981.](#)
- [33] [Nikolova E, Brand M, Karger DR. Optimal route planning under uncertainty. In: ICAPS; 2006. p. 131–41.](#)
- [34] [Nikolova E, Karger DR. Route planning under uncertainty: the Canadian traveller problem. In: AAAI; 2008. p. 969–974.](#)
- [35] [O’Cinneide CA. Characterization of phase-type distributions. Stoch Models 1990;6\(1\):1–57.](#)
- [36] [O’Cinneide CA. Phase-type distributions: open problems and a few properties. Stoch Models 1999;15\(4\):731–57.](#)
- [37] [Panchenko A, Thümmel A. Efficient phase-type fitting with aggregated traffic traces. Perform Eval 2007;64\(7–8\):629–45.](#)
- [38] [Papadimitriou CH, Yannakakis M. Shortest paths without a map. Theor Comput Sci 1991;84\(1\):127–50.](#)
- [39] [Polychronopoulos GH, Tsitsiklis JN. Stochastic shortest path problems with recourse. Networks 1996;27\(2\):133–43.](#)

- [40] [Puterman ML. Markov decision processes. Hoboken NJ: Wiley; 2005.](#)
- [41] [Reibman AL, Trivedi KS. Numerical transient analysis of Markov models. Comput OR 1988;15\(1\):19–36.](#)
- [42] [Rossetti MD. Simulation modeling and arena. Hoboken NJ: Wiley; 2009.](#)
- [43] [Saad Y. Iterative methods for sparse linear systems. 2nd ed. Philadelphia: SIAM; 2003 \(\[http://www.google.com/url?q=http://www-users.cs.umn.edu/~saad/IterMethBook\\\_2ndEd.pdf\]\(http://www.google.com/url?q=http://www-users.cs.umn.edu/~saad/IterMethBook\_2ndEd.pdf\)\).](#)
- [44] [Serfozo RF. An equivalence between continuous and discrete time Markov decision processes. Oper Res 1979;27\(3\):616–20.](#)
- [45] [Sigal CE, Pritsker AAB, Solberg JJ. The stochastic shortest route problem. Oper Res 1980;1122–1129\(5\).](#)
- [46] [Simmons R, Koenig S. Probabilistic robot navigation in partially observable environments. In: Proceedings of the IJCAI-95. San Francisco CA:Morgan Kaufmann, 1995. p. 1080–7.](#)
- [47] [Sudholt D, Thyssen C. A simple ant colony optimizer for stochastic shortest path problems. Algorithmica 2012;64\(4\):643–72.](#)
- [48] [Telek M, Horváth G. A minimal representation of Markov arrival processes and a moments matching method. Perform Eval 2007;64\(9–12\):1153–68.](#)
- [49] [Thrun S, Beetz M, Bennewitz M, Burgard W, Cremers AB, Dellaert F, et al. Probabilistic algorithms and the interactive museum tour-guide robot minerva. J Robot Res 2000;19\(11\).](#)
- [50] [Thümmler A, Buchholz P, Telek M. A novel approach for phase-type fitting with the EM algorithm. IEEE Trans Dep Sec Comput 2006;3\(3\):245–58.](#)
- [51] [Wellman MP, Ford M, Larson K. Path planning under time-dependent uncertainty. In: Proceedings of the 11th conference on uncertainty in artificial intelligence; 1995. p. 532–9.](#)
- [52] [Wenner C. Hardness results for the shortest path problem under partial observability; 2009. See \(<http://cwenner.net/papers/sctp.pdf>\).](#)