Stochastic Graph Models with Phase Type Distributed Edge Weights

Dissertation

zur Erlangung des Grades eines

Doktors der Ingenieurwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik
von

Iryna Dohndorf geb. Felko

Dortmund
2017

Iryna Dohndorf geb. Felko
Lehrstuhl IV - Modellierung und Simulation
Fakultät für Informatik
Technische Universität Dortmund
Otto-Hahn-Straße 16
44227 Dortmund

Tag der mündlichen Prüfung: 08.03.2017

Dekan	**Prof. Dr.-Ing. Gernot A. Fink**
Gutachter	**Prof. Dr. Peter Buchholz** (TU Dortmund, Fakultät für Informatik) **Prof. Dr. Ir. Boudewijn R. Haverkort** (University of Twente, Faculty for Electrical Engineering, Mathematics and Computer Science)

Abstract

Most stochastic shortest path problems include an assumption of independent weights at edges. For many practical problems, however, this assumption is often violated indicating an increased number of applications with stochastic graphs where edge weights follow a distribution that has a possible correlation with weights at adjacent edges. Real-world information in conjunction with existing dependencies in networks can significantly contribute to the computation of the optimal solution to stochastic shortest path problems. For example, the knowledge of a congestion arising on the current road results in a better traveler's choice of an alternative route. Markov dependability models describing the correlation in the length of availability and unavailability intervals of technical components could support optimal decisions to avoid high maintenance costs.

In this thesis, an innovative model class for stochastic graphs with correlated weights at the edges is introduced. In the developed model edge weights are modeled by PH distributions and correlations between them can be encoded using transfer matrices for PH distributions of adjacent edge weights. Stochastic graph models including correlations are important to describe many practical situations where the knowledge about system parameters like traveling times and costs is incomplete or changes over time.

Based on PH-Graphs efficient solution methods for Stochastic Shortest Path Problems with correlations can be developed. Competing paths from origin to destination in a PH-Graph can be interpreted as CTMDPs. Optimal solutions to different shortest path problems can be obtained from finding an optimal policy in a CTMDP. For example, the problem of finding the reliable shortest path to maximize the probability of arriving on time under realistic assumptions can be efficiently treated. Formulations of shortest path problems with correlations as well as solution methods from the CTMDP field are presented.

We address the parameterization of PH-Graphs based on measured data from simulated systems. Fitting methods for parameterization of transfer matrices are adopted from MAP fitting approaches. Also similarity transformations for order 2 acyclic PHDs in composition are considered. Our experiments and examples show that correlation has a significant impact on shortest paths in stochastic weighted networks and that our solution methods are effective and usable in online decision senarios.

Acknowledgment

First and foremost, I thank my supervisor *Prof. Dr. Peter Buchholz* for his support, ideas and valuable advice during the course of this work. I consider myself very lucky because I had an opportunity to work with him. He has been an excellent mentor who is very professional, extremely helpful, and understanding. His inspiration, immense knowledge and kindness guided me throughout my period of research.

Special thanks go to my co-advisor *Prof. Dr. Ir. Boudewijn R. Haverkort* for his hints and suggestions that have inspired me and helped me to improve this work. I would like to thank *Prof. Dr. Heinrich Müller* and *Prof. Dr. Günter Rudolph* for agreeing to serve as members of my defense committee. I am grateful to *Prof. Dr. Heiko Krumm* for sharing his broad knowledge, and for many fruitful discussions.

I want to express my gratitude to my colleagues in the group *Dr. Jan Kriege, Dr. Sebastian Vastag, Dimitri Scheftelowitsch, Dr. Falko Bause, Andrea Berger* for their help, for sharing their knowledge and experience, and friendship.

I would like to express my biggest love and gratitude to my *family*. I am grateful to my mother *Olga*, my husband *Oliver* and my son *Leonard* for their endless love, patience and support.

Contents

Chapter 1

Introduction

The optimization of shortest path problems in weighted stochastic graphs has been studied extensively and has been applied in various fields of computer science, communications, transportation systems, and engineering, to name just a few. Often weights in a graph have been assumed to be deterministic, or independent and identically distributed random variables (see, e.g., [84, 119]). Then efficient algorithms for computing shortest paths are well known [64]. However, these assumptions are violated in most practical problems. Typically, real-world shortest path problems are too complex such that there is a major challenge in building an accurate model to define weights in a stochastic graph model. For this reason it is worthwhile finding the appropriate description of uncertainty and dependencies in weighted graph models.

When probabilistic edge weights are considered one is dealing with different versions of *Stochastic Shortest Path Problems* (SSPPs) [159]. Various settings and problem formulations have been considered [7, 29, 33, 34, 35, 36, 78, 82, 135, 136, 143, 145, 160, 164, 167, 174, 175]. Classically, the problem of finding optimal paths is given when a path with the minimal expected time should be computed (see, e.g., [29, 71, 99]). Another variant is given when a path maximizing the probability of arriving at the destination node within a given time interval should be computed (see, e.g., [72, 77, 123]). One important issue which should be investigated is how established stochastic modeling methods can help to incorporate more realism in SSPPs and shed some light of how dependencies between weights can influence optimal paths.

The aspect of correlation is rarely considered when solving different SSPPs. Some of the earliest and most noteworthy results concern the form of the optimal policy under various assumptions about edge congestion states and risk aversion [71, 99, 172]. In transportation and traffic networks, correlations often occur due to network disruptions, risks of accidents, construction zones on highways and city roads, natural disasters, and congestions. For example, the regular morning congestion in rush hour traffic is typically between $6:30$ and $8:30$, and long traveling times on highways also imply long traveling times on periphery [99]. For these reasons different kind of dependencies among edge traveling times can occur and it is known that the negligence of these correlations can result in non-optimal solutions.

In this thesis, we propose a stochastic model class for the analysis of stochastic graphs with dependencies, called a *Phase-type Graph* (PH-Graph). The new model is

based on *Phase-type distributions* (PHDs) [69, 133] and *Markovian Arrival Processes* (MAPs) [132] and is represented by a Markov chain which describes correlated durations of consecutive time intervals. When interpreting Phase-type distributed durations as weights, a PH-Graph describes a stochastic graph with correlated weights on adjacent edges. Dependencies between adjacent edge weights can be introduced by defining a *transfer matrix* which is similar to the $\mathbf{D}_1$ matrix of a MAP. A PH-Graph model collects PHDs for weights of all edges and transfer matrices for adjacent edges.

The main feature of PH-Graph models is that edge weights can be interpreted depending on the context of the problem making a large variety of applications possible. Nevertheless, open questions regarding their applicability in real-world problems arise. In particular, how PH-Graphs based on PHDs and MAPs can be integrated in stochastic control models for optimization in order to determine the optimal decision under realistic conditions being considered. For example, how PH-Graphs capturing correlations can be used in conjunction with real-world information to determine the optimal policy for the route to traverse? Then, the knowledge about congestion on a highway, vehicle accident, weather-related hazards on roads, shipments or plane routes can be used by a decision maker to choose an optimal path.

PH-Graph is a Markovian model which can be transformed into a *Continuous-Time Markov Decision Process* (CTMDP) such that the optimal paths correspond to the optimal policies and can computed efficiently [148]. CTMDPs are sequential decision models which have been applied to a variety of problems in computer and communication systems, inventory and manufacturing control, nuclear plant and epidemic management, to name just a few. They are widely used in order to improve the associated real-world system or to determine the effective way to control it. Often modeled problem situations correspond to real-world problem situations and one is interested in the optimal policy due to various assumptions about system parameters.

These assumptions contribute to understanding how such models could help to provide insight about optimal system behavior. Research about parameterization of CTMDPs using sophisticated distributions like PHDs and transfer matrices is needed to accommodate empirical behavior of a modeled process.

As already mentioned, PH-Graphs are based on PHDs and MAPs, that are popular among researchers in the field of stochastic modeling. These are Markov processes with an intuitive stochastic interpretation integrating the concept of *phases* and *events* [115]. In particular, PHDs are described by *Continuous-Time Markov Chains* (CTMC) with an absorbing state providing several exponentially distributed time intervals, also known as stages. PHDs represent a versatile and computationally tractable class of probability distributions which lead to an easier numerical analysis since they make use of Markov property and efficient matrix analytic methods [115]. Their main conversing feature is that using PHDs any non-negative distribution can be approximated arbitrary close [138]. PHDs allow one to capture empirical and stochastic behavior from the measured data efficiently [108, 168]. However, if one is interested in description of correlated data MAPs should be used rather than PHDs. MAPs represent a Markovian modeling technique which is strongly connected to PHDs. They are based on CTMCs with marked transitions such that the process generates an *arrival event* when particular transitions occur. These stochastic behavior enables modeling of autocorrelated interevent times which are represented by a fixed PHD, but there are still many open questions when transfering these concepts to PH-Graphs. The first

question arising is: how can the correlation between interevent times which are distributed according to an arbitrary PHD be described? One of the ways to deal with this challenge is to adopt the concept of $\mathbf{D}_1$ matrix from MAPs to transfer matrices.

The mathematical representation of PHDs and MAPs is determined by an underlying CTMC. Due to the non-uniqueness of the matrix representation, representations maximizing the first joint moment that can be reached when PHD is expanded to a MAP have been investigated in the past [41]. Nevertheless, the next question is: which representations of different PHDs are most suitable when maximizing their correlation? This thesis proposes the treatment of these questions.

Although, PHDs and MAPs are not broadly used in practical system modeling yet their flexibility and practicability in matching empirical data to their parameters cannot be neglected today. Parameter fitting for PHDs and MAPs is a complex non-linear problem. Much theoretical and practical research has focused on the features of PHDs and MAPs, their applicability in stochastic models and fitting algorithms, such that several software solutions to generate PHDs and MAPs according to observed data are available [38, 44, 52, 96, 108, 139, 140, 165, 166, 168]. For example, efficient algorithms allow generation of PHDs with up to 20 phases in at most a few minutes. Parameter fitting for MAPs is a more complex optimization problem than parameter fitting for PHDs since also long range behavior should be considered to match parameters adequately. Nevertheless, the technology is mature enough and well accepted by researchers such that the Markovian models based on PHDs and MAPs can be widely used in applied probability.

1.1. Contribution

The main theoretical contribution of this thesis is a novel model class for weighted stochastic graphs with correlated weights at the edges. In the developed model, edge weights are modeled with Phase-type distributions, a versatile class of distributions which can be used to approximate any continuous distribution. Modeling of correlated edge weights is done by adding dependencies between the PHDs of adjacent edge weights. The concept of a transfer matrix adopted from $\mathbf{D}_1$ matrix from MAPs field is introduced.

In this way we first provide modeling of graphs with stochastic edge weights which can describe many real-world problems with uncertain and time varying parameters. Based on PH-Graphs efficient solution methods for Stochastic Shortest Path Problems with correlations is developed. This is done by generation of a suitable Continuous-Time MDP and the application of established solution methods. The new PH-Graph model, integration and adopting of methods for its parameterization is the core contribution of this thesis.

The proposed parameterization algorithms of the transfer matrix are adopted from MAP fitting approaches. Additionally, we present similarity transformations for Phase-type distributions that can be applied to increase the range of correlation that can be modeled by the newly developed PH-Graph model. These theoretical insights are used in an expectation-maximization fitting algorithm to identify the most suitable initial transfer matrix. In the two phase fitting approach, first the parameters of PH distributions have to be determined. Hence, PHDs with representations which are most

suitable for subsequent fitting of transfer matrix with maximal first joint moment can be generated.

Around our PH-Graph model we develop a corresponding CTMDP that captures possible correlations in order to compute the optimal policy under realistic conditions. When a human decision maker needs to decide on which policy to follow, information about currently realized edge weight is required to exploit the effect of correlation, as visualized in Fig. 1.1. The developed CTMDP method returns the optimal policy for each possible realization of the current edge weight, such that a human decision maker can select one to execute. Thus providing a framework for decision support. Finally, the thesis is accompanied by case studies to demonstrate the practical potential of optimization based on PH-Graphs.

Several parts of this thesis have already been published before. Basic concepts described in Sections 2.1.3, 2.4, 2.5 appeared in [47]. The basic PH-Graph model concepts in Sections 3.2, 5.1 and 5.2 have been published in [40]. Parameterization concepts and results from Chapter 4 appeared in [47] and [40]. Results appeared in Chapter 5 are published in [40]. The empirical work described in Section 7.1 appeared also in [40]. Parts of theoretical results presented in Chapter 6 are in principle based on [41].

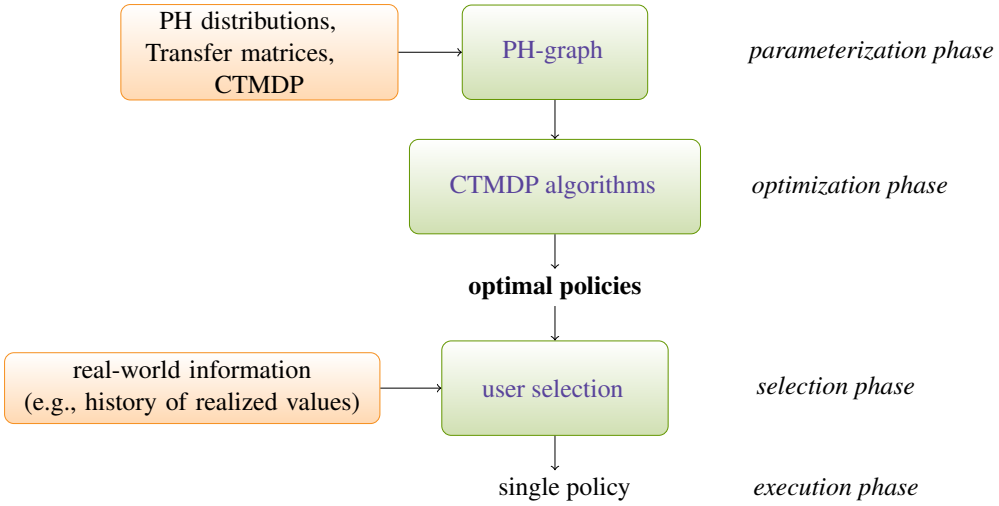


Figure 1.1.: Decision support based on PH-Graph and corresponding CTMDP

1.2. Outline

The thesis consists of six chapters which are organised as follows:

- In Chapter 2 basic concepts from the field of Markov Decision Processes and Stochastic Shortest Path Problems are summarized. We give a comprehensive introduction to Phase-type distributions and Markovian Arrival Processes which are of great importance throughout the thesis. Indispensable knowledge therefore like concepts of Markov processes and dynamic programming are also included.

- In Chapter 3 we develop our PH-Graph to model weighted stochastic graphs with correlated PHD weights at the edges. We first model edge weights by Phase-type distributions. Then we introduce a composition of two Phase-type distributions which describes correlated weights at adjacent edges using the transfer matrix. We add dependencies between the entry phase of the second PHD and the exit phase of the first PHD.

- Chapter 4 presents methods to parameterize the required PHDs for edges and transfer matrices based on measured data. We first describe several fundamental algorithms and results in the field of fitting PHDs and MAPs. We adopted fitting methods to parameterize transfer matrices. From several examples the fitting quality of our algorithms can be evaluated.

- Chapter 5 gives the formulation of the PH-Graph model as Continuous-Time Markov Decision Process. We discuss how solutions to many shortest path problems can be obtained as the computation of an optimal policy in a CTMDP. We formulate basic shortest path problems and present the corresponding solution algorithms from the field of CTMDPs.

- In Chapter 6 we describe similarity transformations for acyclic PHDs. We investigate which representations of both PHDs are most suitable in order to maximize the first joint moment of PHD composition. We present similarity transformation method for 2-order APHDs in order to generate a representation which allows for fitting a transfer matrix with maximal first joint moment. When optimal representations are known, parameterization of PHDs and transfer matrices in PH-Graphs can demonstrably be simplified.

- In Chapter 7 we demonstrate the usability and effectiveness of our solution methods by solving stochastic shortest path problems in real-world case studies. We compute shortest paths using a vehicular traffic model under realistic assumptions and analyze the computational effort using different types of graphs approximating road networks. The second case study considers finance application. In the third case study we analyze model from the maintenance field. When, for example, a human decision maker needs to decide which policy to follow, our methods provide useful results taking correlations in real-world networks into consideration.

Chapter 2

Preliminaries

2.1. Markov Processes

A Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ is a stochastic process where $X(t)$ is a random variable with parameter t describing the time. The values corresponding to random variables are called states such that the set of all possible states defines the state space $\mathscr{S}$. In the following we consider Markov chains building the basis for Markovian Decision Processes and Phase-type Distributions, which we consider in Sections 2.2, and 2.4.

The set T denotes the parameter space: If $T = \{0,1,2,\ldots\}$, then we have a discrete-time parameter space; If $T = \{t : 0 \leq t < \infty\}$, then we have a continuous time parameter space. Thus the Markov chain can evolve either at a discrete set of time points or continuously in time.

Markov chains have a conditional probability distribution function characterized by the *Markov property* [162]:

$$Prob(X(t_{k+1}) = x_{k+1} \,|\, X(t_k) = x_k, \ldots, X(t_1) = x_1) = Prob(X(t_{k+1}) = x_{k+1} \,|\, X(t_k) = x_k), \tag{2.1}$$

for $t_{k+1} > t_k > t_{k-1} > \cdots > t_0$. Eq. 2.1 states that given the current state x_k and the time t_k, the next state of the Markov process $X(t_{k+1})$ depends only on x_k and on t_k but not on the past of the process.

A Markov process is called *nonhomogeneous* when transitions out of state $X(t)$ depend on t. When transitions out of state $X(t)$ do not depend on t, the process is called *homogeneous* [162]. All Markov processes discussed in this thesis are homogeneous.

The time spent in a state is denoted as a *sojourn time*. In the continuous time case it is exponentially distributed. Whereas in the discrete-time case it exhibits the geometrical distribution. Exponential and geometrical distributions are the only distributions which satisfy the memoryless property, i.e., the sojourn time Y spent by a Markov process in state i is independent of how long the process has previously been in state i

$$Prob(Y > t+s \,|\, Y > t) = Prob(Y > s) \text{ for all } t, s \geq 0,$$

such that the behavior satisfies the Markov property.

2.1.1. Discrete-Time Markov Chains

A discrete-time Markov chain (DTMC) is a Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ with discrete infinite set of times which satisfies the Markov property from Eq. 2.1

Conditional probabilities $p_{ij}(k) = Prob(X(t_{k+1}) = j \,|\, X(t_k) = i)$ are the single-step transition probabilities. They specify the probability of making a transition from state $x_k = i$ to state $x_{k+1} = j$ when time increases from k to $k+1$. In homogeneous Markov chains these probabilities are independent of time parameter k such that the time index k can be skipped resulting in $p_{ij} = Prob(X(t_{k+1}) = j \,|\, X(t_k) = i)$ for all $k = 1, 2, \ldots$. Transition probabilities for all states $i, j \in \mathscr{S}$ are collected in the *transition probabilities matrix* $\hat{\mathbf{P}}$

$$\hat{\mathbf{P}}(i,j) = p_{ij}, \quad 0 \leq p_{ij} \leq 1, \quad \sum_j p_{ij} = 1.$$

A DTMC is fully characterized by the tuple $(\pi_0, \hat{\mathbf{P}})$ with initial probability vector π_0, where $\pi_0(i)$ gives the probability that the Markov process starts in state i, $i \in \mathscr{S}$.

Single-step transition probabilities may be generalized to n-step transition probabilities $p_{ij}^{(n)} = Prob(X(t_{k+n}) = j \,|\, X(t_k) = i)$, where $p_{ij}^{(1)} = p_{ij}$. $p_{ij}^{(n)}$ can be computed using the Chapman-Kolmogorov equations [162] resulting in a n-step transition matrix $\hat{\mathbf{P}}^n$. Then transient probabilities after n time steps can be obtained as

$$\pi^{(n)} = \pi_0 \hat{\mathbf{P}}^n, \tag{2.2}$$

such that $\pi^{(n)}(i)$ describes the probability with which the Markov process occupies state i after n transitions have occured. In some cases transient probabilities converge to a limiting distribution of the Markov chain. Particularly, if the limit $\lim_{n\to\infty} \hat{\mathbf{P}}^n$ exists, then the n-step transition probabilities $p_{ij}^{(n)}$ become independent of n

$$\pi = \lim_{n\to\infty} \pi^{(n)} = \pi_0 \lim_{n\to\infty} \hat{\mathbf{P}}^n$$

Then a distribution π is called a limiting distribution of the Markov chain. The conditions under which the limiting probabilities exist are, e.g., irreducibility and aperiodicity of a finite Markov chain [162, 115], and ergodicity for infinite Markov chains. A limiting distribution π is called a *steady-state distribution* if it converges to a vector with strictly positive components with $\pi\mathbb{1} = 1$, independently of the initial distribution π_0. Steady-state probabilities can be obtained from the system of linear equations

$$\pi = \pi\hat{\mathbf{P}}, \quad \pi\mathbb{1} = 1, \quad 0 \leq \pi(i) \leq 1. \tag{2.3}$$

2.1.2. Continuous-Time Markov Chains

A continuous-time Markov chain (CTMC) is a Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ with continuous time parameter set $T = \mathbb{R}_{\geq 0}$ which satisfies the Markov property from Eq. 2.1

Transition probabilities of a continuous-time Markov process are given by $p_{ij}(t) = Prob(X(t+s) = j \,|\, X(s) = i)$ in the homogeneous time case and depend on the difference t between s and $t+s$ and not on the actual values s and $t+s$. Values $p_{ij}(t)$ are collected in a matrix $\mathbf{P}_t$.

State probabilities at time t are given by $\pi^{(t)}(j) = Prob(X(t) = j)$ for $j \in \mathscr{S}$ such that $\sum_j \pi^{(t)}(j) = 1$. Then the vector $\pi_0 = (\pi^{(t)}(1), \pi^{(t)}(2), \ldots)$ is the initial probability

vector of a Markov process. As mentioned above, the sojourn time in state i of a CTMC is exponentially distributed with parameter $\lambda(i)$, $0 \leq \lambda(i) < \infty$. Thus the times between state transitions are exponentially distributed. Let Y_i describe the sojourn time in state i, its distribution function is

$$Prob(Y_i \leq t) = 1 - e^{\lambda(i)t},\ t \geq 0. \tag{2.4}$$

This distribution describes the time a Markov process spends in a state before making any transition.

Now the evolution of a CTMC can be described. At any given point in time, the process occupies one of the states, i.e., $X(t) = i$. State holding time is exponentially distributed with parameter $\lambda(i)$ such that after this time has elapsed the process jumps to the next state j with probability $p(i,j) = \lambda(i,j)/\lambda(i)$. Here $\lambda(i,j)$ gives the transition rate from i to j. Summation of all transition rates result in $\sum_j \lambda(i,j) = \lambda(i)$ such that $\lambda(i)$ is the total event rate of the state i. Therefore, the behavior of CTMC can be described by $n \times n$ *infinitesimal generator (transition rate) matrix* $\mathbf{Q}$

$$\mathbf{Q}(i,j) = \begin{cases} -\lambda(i) & \text{if } i = j, \\ \lambda(i,j) & \text{if } i \neq j. \end{cases} \tag{2.5}$$

If a transition from i to j can occur, then it holds that $\lambda(i,j) > 0$ and consequently all non-diagonal entries of $\mathbf{Q}$ are non-negative, i.e. $\mathbf{Q}(i,j) \geq 0$. In contrast, all diagonal elements of matrix $\mathbf{Q}$ are non-positive assuming that $\lambda(i) > 0$, i.e., $\mathbf{Q}(i,i) < 0$. It holds that

$$\sum_j \mathbf{Q}(i,j) = 0.$$

For a CTMC we can specify the embedded process $\{X_r\}_{r \in \mathbb{N}_0}$ if only the sequence of transitions that can occur is considered. Single-step transition probabilities are collected in transition probability matrix $\tilde{\mathbf{P}}$ with entries $\tilde{\mathbf{P}}(i,j) = Prob(X(r+1) = j \,|\, X(r) = i)$ which are equal to zero if $i = j$

$$\tilde{\mathbf{P}}(i,j) = \frac{\mathbf{Q}(i,j)}{-\mathbf{Q}(i,i)},\ \text{ for } j \neq i, \mathbf{Q}(i,i) \neq 0. \tag{2.6}$$

All elements of matrix $\tilde{\mathbf{P}}$ satisfy $0 \leq \tilde{\mathbf{P}}(i,j) \leq 1$ and $\sum_j \tilde{\mathbf{P}}(i,j) = 1$.

Consider a homogeneous CTMC. Let $\pi^{(t)}(i)$ be probability that the process is in state i at time t.

$$\pi^{(t)}(i) = Prob(X(t) = i).$$

These transient probabilities can be obtained by solving the system of differential equations [162]

$$\frac{d}{dt}\pi^{(t)} = \pi_0 \mathbf{Q}. \tag{2.7}$$

The solution $\pi^{(t)}$ is analytically given by

$$\pi^{(t)} = \pi_0 e^{\mathbf{Q}t}, \tag{2.8}$$

where $e^{\mathbf{Q}t}$ is the matrix exponential and is defined by the infinite series

$$e^{\mathbf{Q}t} = \mathbf{I} + \mathbf{Q}t + \frac{\mathbf{Q}^2 t^2}{2!} + \frac{\mathbf{Q}^3 t^3}{3!} + \cdots. \tag{2.9}$$

Unfortunately, evaluation of Eq. 2.9 can be computationally unstable and difficult to compute [55, 152, 169]. In particular, numerical methods for evaluation of the matrix exponential can be complex and often require the optimal parameter selection [5, 6, 63, 106, 128, 144, 157]. Methods for solving linear differential equations like the Runge-Kutta-Fehlberg method or an implicit method TR-BDF2 [80, 124, 151] can be used but possess no stochastical context. Uniformization (also known as Jensen's method or the method of randomization) provides an accurate numerical solution and yields the stochastical interpretation [101, 163]. In the following we give the description of this method along the lines of [163].

Uniformization The method is based on Taylor series expansion for the matrix exponential. Recall that $e^{\mathbf{Q}t} = \sum_{n=0}^{\infty} \frac{(\mathbf{Q}t)^n}{n!}$. Thus the numerical solution of Eq. 2.7 can be given by

$$\pi^{(t)} \approx \sum_{n=0}^{r(t,\varepsilon)} \pi_0 \frac{(\mathbf{Q}t)^n}{n!}, \tag{2.10}$$

where $r(t,\varepsilon)$ denotes the upper truncation point such that the required error tolerance ε is satisfied. However, since diagonal elements of $\mathbf{Q}$ are negative the computation of Eq. 2.10 can lead to rounding errors.

Next the *uniformization rate* $\alpha \geq \max_i(|\mathbf{Q}(i,i)|)$ is determined. Then the rate of the sojourn time distribution of all states is uniformized with α by setting $\mathbf{Q}(i,i) = -\alpha$ for all $i \in \mathscr{S}$. In fact, the original CTMC with non-identical transition rates is transformed into a stochastic process in which transition epochs are generated by a Poisson process at a rate α. Therefore the infinitesimal generator $\mathbf{Q}$ is transformed to obtain the matrix

$$\mathbf{P} = \mathbf{I} + \frac{1}{\alpha}\mathbf{Q}, \tag{2.11}$$

which is the transition probability matrix of embedded process of a CTMC with elements in the interval $[0,1]$. Then it holds that $\mathbf{Q} = \mathbf{P}\alpha - \mathbf{I}\alpha$ and the matrix exponential relation becomes

$$e^{\mathbf{Q}t} = e^{(\mathbf{P}\alpha - \mathbf{I}\alpha)t} = e^{\mathbf{P}\alpha t} e^{-\mathbf{I}\alpha t} = e^{\mathbf{P}\alpha t} \left(\sum_{n=0}^{\infty} \mathbf{I}^n \frac{(\alpha t)^n}{n!} \right)^{-1} = e^{\mathbf{P}\alpha t} \mathbf{I}^{-1} e^{-(\alpha t)} = e^{\mathbf{P}\alpha t} e^{-(\alpha t)},$$

since $\mathbf{I}^n = \mathbf{I}$ for all n and $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. Expanding the term $e^{\mathbf{P}\alpha t}$ in Taylor series Eq. 2.10 can be written as

$$\begin{aligned} \pi^{(t)} &= \sum_{n=0}^{\infty} \underbrace{\pi_0 \mathbf{P}^n}\, \underbrace{e^{-(\alpha t)} \frac{(\alpha t)^n}{n!}} \\ &= \sum_{n=0}^{\infty} \phi(n) \quad \beta(n, \alpha t). \end{aligned} \tag{2.12}$$

The expression

$$\beta(n, \alpha t) = e^{-(\alpha t)} \frac{(\alpha t)^n}{n!} \tag{2.13}$$

gives the density function of Poisson process $\{N_t \,|\, t \geq 0\}$ with rate α. The specified Poisson process is associated with an uniformized CTMC such that $\beta(n, \alpha t)$ describes

the probability that exactly n transitions have occured in time interval $[0,t)$. Each time the Poisson process with rate α generates a transition epoch the state transitions are governed by the embedded DTMC. Poisson probabilities can be computed with the method described in [76] where all computations are numerically stable and only positive values are used.

The term $\phi(n) = \pi_0 \mathbf{P}^n$ gives transient probabilities for the embedded DTMC, i.e., the i-th element of the vector $\phi(n)$ is the probability that the embedded process is in state i after n transitions.

The sum formula 2.12 needs to be truncated such that we obtain

$$\pi^{(t)} = \sum_{i=l(\alpha t,\varepsilon)}^{r(\alpha t,\varepsilon)} \pi_0 \mathbf{P}^i e^{-(\alpha t)} \frac{(\alpha t)^i}{i!} + \varepsilon, \tag{2.14}$$

where the lower truncation point $l(\alpha t,\varepsilon)$ and the upper truncation point $r(\alpha t,\varepsilon)$ can be pre-computed such that the required error tolerance ε is satisfied. Numerical methods on computation of lower and upper truncation points can be found in [162].

Stationary Distribution The steady-state distribution of a CTMC is given by a long-run probability vector π such that $\pi(i)$ is the probability of being in state i when statistical equilibrium has been reached. Under the condition that the stationary distribution exists, probability $\pi(i)$ no longer depends on time t for all $i \in \mathscr{S}$. The steady-state distribution exists when there is a point in time at which the rate of change of transient probability vector $\pi^{(t)}$ is zero, i.e., when $\frac{d}{dt}\pi^{(t)} = 0$ holds in Eq. 2.7. In a finite, irreducible, homogeneous CTMC the limit $\lim_{t\to\infty} \pi^{(t)}$ exists and the steady-state distribution may be determined by solving the system of linear equations [162]

$$\pi \mathbf{Q} = \mathbf{0}, \quad \sum_{i\in\mathscr{S}} \pi(i) = 1. \tag{2.15}$$

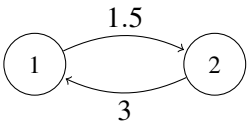
***Example* 2.1.** *Consider a 2-state CTMC with infinitesimal generator* $\mathbf{Q}$ *and a state transition diagram as shown in Figure 2.1. The initial distribution vector is* $\pi^{(0)} = (0,1)$. *The uniformization rate can be determined as* $\alpha = 3$ *and we obtain the DTMC shown in Figure 2.2. We computed transient probability vectors from Eq. 2.14 for some values of* t. *For accuracy* $\varepsilon = 10^{-3}$ *we obtain*

$$\begin{aligned} \pi^{(0.1)} &= (0.2415, 0.7585), & \pi^{(1.5)} &= (0.6659, 0.3341), \\ \pi^{(0.5)} &= (0.5963, 0.4037), & \pi^{(2)} &= (0.6666, 0.3334), \\ \pi^{(1)} &= (0.6593, 0.3407), & \pi^{(3)} &= (0.6667, 0.3333). \end{aligned}$$

Observe that, after a certain time t, the transient probabilities no longer change. Solving Eq. 2.15 we obtain the steady state distribution $\pi = (0.6667, 0.3333)$.

2.1.3. Absorbing Markov Chains

An important class of CTMCs are *absorbing Markov chains* since they provide the basis for analysis of the process behavior up to the moment that it enters an absorbing state. For Markov chains there exists a broad theory on the description of states (see [105, 162, 47] and references therein). In the following we describe the

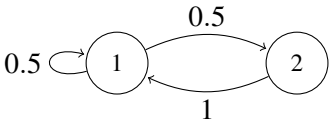


(a) A state transition diagram of the CTMC.

$$Q = \begin{bmatrix} -1.5 & 1.5 \\ 3 & -3 \end{bmatrix}$$

(b) The infinitesimal generator $\mathbf{Q}$.

Figure 2.1.: State transition diagram and generator matrix of a CTMC



(a) A state transition diagram of the uniformized CTMC with uniformization rate $\alpha = 3$.

$$\mathbf{P} = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \end{bmatrix}.$$

(b) Transition matrix $\mathbf{P}$.

Figure 2.2.: State transition diagram and generator matrix of the uniformized CTMC

concepts of reachability, communicating, transient and absorbing states from these sources.

Definition 2.1. *A state j is called to be* reachable *from a state i if it holds for the transition probability*

$$p_{ij}^{(t)} = Prob(X(t+u) = j | X(u) = i) > 0$$

for some t. In that case the process can move from state i to state j after some amount of time t.

Then states i and j can *communicate* with each other if i is reachable from j and vice versa [105, 47]. Now assume that some subset $\mathscr{C}$ of the state space contains only communicating states. Then this subset $\mathscr{C}$ is called a *communicating* set. Additionally, if $\tilde{\mathbf{P}}(i,j) = 0$ for all $i \in \mathscr{C}$ and all $j \in \mathscr{S} \setminus \mathscr{C}$, then there is no feasible transition from $\mathscr{C}$ to outside states. In that case $\mathscr{C}$ is called a *closed* set [105, 47].

Definition 2.2 (Def. 2.2 in [47])**.** *If $\mathscr{C}$ consists of a single state i, then i is said to be an absorbing state.*

By definition it holds for the absorbing state i that $\tilde{\mathbf{P}}(i,j) = 0$ for all $j \in \mathscr{S} \setminus \mathscr{C}$ and thus we have $\tilde{\mathbf{P}}(i,i) = 1$. After entering an absorbing state the Markov process can never leave it. From this point the process behavior will not change and its lifetime can be determined. In performance models entering an absorbing state corresponds with an occurrence of some event such that we are only interested in the process behavior until absorption.

The states of a Markov chain can be classified according to whether and when it is possible to return to a state after leaving it.

Definition 2.3. *For a Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ we define the following probability*

$$\begin{aligned} f_i &= Prob(\text{Eventually return to state } i \mid X(0) = i) \\ &= Prob(X(k) = i \text{ for some } k \geq 1 \mid X(0) = i). \end{aligned}$$

If $f_i = 1$, then the state i is said to be recurrent. *Otherwise, if $f_i < 1$, the state i is said to be* transient.

For a transient state $i \in \mathscr{S}$ there is a positive probability of leaving it forever, and a recurrent state i is one which will be visited infinitely often [47].

Let $\mathscr{C}$ denote the set of all transient states. In a Markov chain with transient sets there is the possibility of moving to some state j from which there is no return to this set, and it can never enter this set again once it leaves it. A transient state may be visited again, but with some positive probability it will not [105, 47].

A Markov chain where every state is either absorbing or transient is defined as *absorbing Markov chain* [47]. Without loss of generality we can assume that there is a single absorbing state $n+1$. Absorbing Markov chains have the important property that the probability to reach an absorbing state tends to 1 independently of the initial state, i.e., $\lim_{t\to\infty} Prob(X(t) < n+1) = 0$ [114, Theorem 2.4.3].

Theorem 2.1. *The probability that a finite absorbing Markov chain reaches an absorbing state in k steps tends to 1 as $k \to \infty$.*

We now describe the canonical matrix representation for absorbing CTMCs as given in [47]. Let $\mathscr{S}$ be the finite state space of a continuous time absorbing Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ where a set of transient states is denoted by $\mathscr{S}_T = \{1,\ldots,n\}$ and a single absorbing state is $n+1$. States of the CTMC are ordered as shown in the following infinitesimal generator matrix $\mathbf{Q}$ [47]

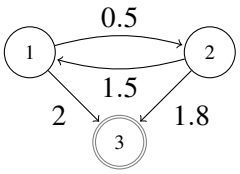
$$\mathbf{Q} = \begin{array}{c} \overbrace{\phantom{\mathbf{D}_0}}^{n} \quad \overbrace{\phantom{\mathbf{d}_1}}^{1} \\ \left[\begin{array}{c|c} \mathbf{D}_0 & \mathbf{d}_1 \\ \hline \mathbf{0} & 0 \end{array}\right] \end{array} \begin{array}{l} \left.\phantom{\begin{array}{c}a\\a\end{array}}\right\}n \\ \left.\right\}1 \end{array} \tag{2.16}$$

In $\mathbf{Q}$ the n transient states occur first following by an absorbing state $n+1$. The $n \times n$ submatrix $\mathbf{D_0}$ contains only transition rates between transient states. The $n \times 1$ vector $\mathbf{d}_1$ describes transitions from transient states to the single absorbing state. Since no transitions from the absorbing state to any transient state are possible, the row vector $\mathbf{0}$ consists entirely of 0's. The element 0 in the right lower corner defines the transition rate out of the absorbing state $n+1$.

Subgenerator matrix $\mathbf{D}_0$ plays an important role in the numerical analysis of absorbing Markov chains and Phase-type distributions. In particular, the matrix $\mathbf{D}_0$ is nonsingular [114, Theorem 2.4.3]. We give the definition of the *fundamental matrix* according to [105].

Definition 2.4. *The matrix $(-\mathbf{D}_0)^{-1}$ is the fundamental matrix of an absorbing continuous time Markov chain. The entry $(-\mathbf{D}_0)^{-1}(i,j)$ gives the expected total time spent in state j before absorption given that the initial state is i.*

***Example* 2.2.** *The absorbing Markov chain with transition matrix $\mathbf{Q}$ in Figure 2.3 has two transient states 1 and 2. The absorbing state 3 has no transition rates to any other state. Regions of the matrix $\mathbf{Q}$, namely $\mathbf{D_0}$, $\mathbf{d}_1$, and $\mathbf{0}$ are marked.*



(a) The state transition diagram for the absorbing CTMC.

$$\mathbf{Q} = \left[\begin{array}{cc|c} -2.5 & 0.5 & 2 \\ 1.5 & -3.3 & 1.8 \\ \hline 0 & 0 & 0 \end{array}\right]$$

(b) The matrix for the absorbing CTMC.

Figure 2.3.: An absorbing CTMC with two transient and one absorbing state. Hence $\mathbf{D}_0$ is a 2×2 matrix and the vector $\mathbf{d}_1$ is of dimension 2×1.

2.2. Markov Decision Processes

This section deals with *continuous-time Markov Decision Processes (CTMDP)* which are closely related to continuous-time Markov chains. CTMDPs represent a class of stochastic processes with a countable discrete state space [162] like CTMCs. The latter forms the basis for CTMDPs which are also known as *stochastic dynamic programming, or continuous-time controlled Markov chains* [148]. CTMDPs were first introduced by Howard in [97] and have found a wide application in performance evaluation, e.g., in queueing systems [149], manufacturing control processes [111, 65], e.g., inventory control [26], system biological processes, as e.g., stochastic models for infectious diseases control [22, 176], dynamic routing processes [130, 68], and finance [24], e.g., optimization problems in insurance [155, 156, 161].

The key idea is to use decision making by adding decisions and rewards to Markovian process in order to reach an optimization goal. The resulting Markovian structure can then be exploited in numerical analysis of the model and in generating optimizing decisions.

In this section, we first provide the basic definitions and notations for CTMDPs. We then proceed with an explanation of the basic processes associated with this model. Our attention can be restricted to CTMDPs in which decisions are made when a state has been entered and to DTMDPs in which decisions are made at transition times as explained in [148, 24].

Continuous-time MDPs A CTMDP is a probabilistic model concerning a nondeterministic choice with multi-periods that corresponds to dynamic decision making in stochastic environments. In MDPs decisions are made in sequential manner such that the results of current decisions and induced possibilities for future decisions are considered. The graphical representation of the decision making process and of the state evolution process is given in Fig 2.4.

CTMDPs can be defined by the tuple $(\mathscr{S}, \mathscr{D}, \mathscr{H}, \mathscr{R}, \mathscr{Q})$ [148, 24] with the following model components:

- A countable state space $\mathscr{S}$. Each state i is associated with a set of possible actions. Given the current state at time point t_n, an action from the set of available actions in that state has to be selected. This decision is carried out by the *decision maker, controller, or agent*.

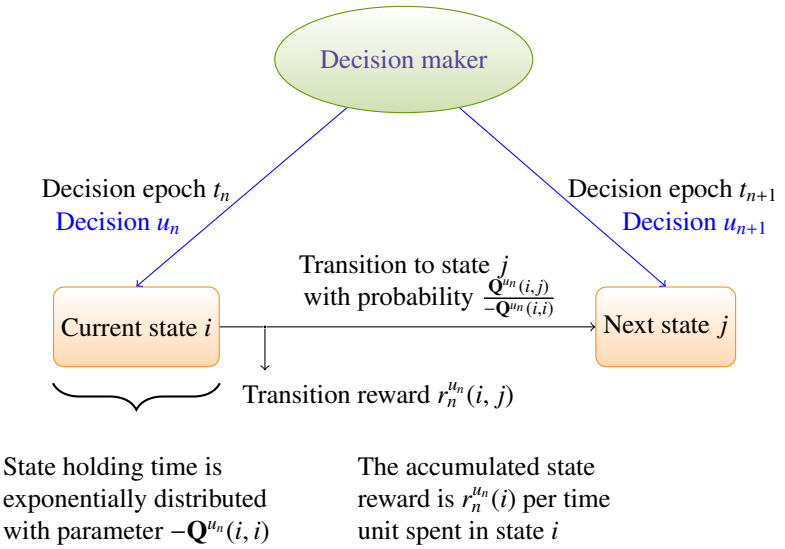


Figure 2.4.: Graphic representation of the decision making process, state evolution process, and reward process in a CTMDP.

- A countable decision space $\mathscr{D} := \bigcup_{i\in\mathscr{S}} \mathscr{D}(i)$. Let m_i be the number of different decisions in set $\mathscr{D}(i)$. Given current state i at time point t_n, the admissible set of decisions is a subset $\mathscr{D}_n(i) \in \mathscr{D}$.

- A set of decision epochs $\mathscr{H}$. At each decision epoch $t_k \in \mathscr{H}$, or point in time decisions are made through selecting an available action u. The set $\mathscr{H}$ contains non-negative real numbers. Furthermore, it is either a discrete set or a continuous set, and it is either finite or infite, i.e. $|\mathscr{H}| \leq \infty$. The set of decision epochs determines the sequence of selected actions $Y(k)$, $k = 0,\ldots,|\mathscr{H}|-1$, where random variable $Y(k)$ denotes a nondeterministically selected decision u in state i, $u \in \mathscr{D}_k(i)$, $i \in \mathscr{S}$.

- A set of rewards $\mathscr{R}$. Each time an action in state i is taken, the decision maker obtains a certain reward $r_n^u(i)$ at some point in time t_n. The received *state* reward depends on the state i, and on the selected action u, $u \in \mathscr{D}_n(i)$, $i \in \mathscr{S}$.

  The *transition* reward is denoted by $r_n^u(i,j)$, $r_n^u(i,j) = 0$ for $i = k$, $r_n^u(i,j) = 0$ for $\mathbf{Q}^u(i,j) = 0$, $r_n^u(i,j)$, $r_n^u(i) < \infty$. It is received when transition from state i to state j occurs. This reward is also referred as impulse reward.

  Since $r_n^u(i,j)$, $r_n^u(i)$ can take positive and negative values, it could be also interpreted as an incurred *costs*, or as an *income*.

- A set of transition rates $\mathscr{Q} = \{\mathbf{Q}^u(i,j) | i,j \in \mathscr{S}, u \in \mathscr{D}(i)\}$, $\mathbf{Q}^u(i,j) > 0$, for $i \neq j$,, $\sum_j \mathbf{Q}^u(i,j) = 0$. Furthermore, we denote $\lambda(i) = -\mathbf{Q}^u(i,i)$. The decision u chosen in some state i determines the transition probability distribution for

the next state. Then infinitesimal generator $\mathbf{Q}^u$ prescribes transition rates for successor states.

Process Evolution We denote the corresponding stochastic state evolution process as $\{X(t)\}_{t\geq 0}^{\infty}$, and the corresponding stochastic decision process as $\{Y(t)\}_{t\geq 0}^{\infty}$ [148]. The initial state of the CTMDP at the time t_0 can be determined according to the initial distribution vector π. Then the evolution of a system can be decribed as follows [127, 148, 81].

Assume that the system is observed to be in state i, $i \in \mathscr{S}$ at some point in time $t \geq 0$. Then the decision maker knows a set of available decisions, and selects an action $u \in \mathscr{D}_t(i)$. The process stays an exponentially distributed time in state i, i.e., the sojourn time in state i is exponentially distributed with parameter $-\mathbf{Q}^u(i,i) = \sum_j \mathbf{Q}^u(i,j)$ [81].

Afterwards the decision maker obtains a transition reward, and system state changes to a different state j. If we consider the system on the time interval $[t, t+\Delta t)$, $\Delta t > 0$, then the received reward per transition is $r_t^u(i,j)$ [127]. If the system is in state i at time t, the probability that the system is in state j at time $t+\Delta t$ given that decision u is always made in the interval $[t, t+\Delta t)$ when the system is in state i is given by $\mathbf{Q}^u(i,j) + o(\Delta t)$.

While the process stays an exponentially distributed time in a state, the decision maker accumulates state reward $r^u(i)$ per time unit in state i. In principle, the model allows for a reward received at random or predefined point in time prior to next decision epoch, or for a reward collected continuously in some interval, or a combination of both as described in [148].

Policies To optimize some performance criteria of a CTMDP decision rules and policies are needed. Assume that at some point in time t information about admissible decisions $\mathscr{D}_t(i)$ is available. A *decision rule* prescribes a decision that has to be chosen from this set. Formally, the decision rule is a mapping $\mathbf{u}_t : \mathscr{S} \to \mathscr{D}$ such that $\mathbf{u}_t(i) \in \mathscr{D}_t(i)$, i.e. $\mathbf{u}_t$ contains decision rules for all states at some point in time t [148].

The policy can be defined as a sequence of decison rules $\mathbf{d} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1})$ for $T \leq \infty$. The policy defines a decision rule for all states to be used at each decision epoch t [148]. Formally, let $\mathscr{P} = \bigtimes_{i=1}^{n} \mathscr{D}(i)$ and the vector $\mathbf{u} \in \mathscr{P}$ describes decision rules for all states. Then, a piecewise constant deterministic policy $\mathbf{d}$ is a function $\mathbf{d} : [0, \ldots, T] \to \mathscr{P}$ [127, 42].

It is assumed that the policy $\mathbf{d}$ is a measurable function where measurable should be understood as Lebesque measurable [127]. Let Π denote the set of piecewise constant deterministic policies.

For a *stationary* deterministic policy it holds for elements in the sequence $\mathbf{d} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1})$ that $\mathbf{u}_i = \mathbf{u}$ for $\forall i \in \{0, \ldots, T-1\}$. The same decision rule given by $\mathbf{u}$ is used at each decision epoch [42]. Let Π_{SD} be the set of stationary deterministic policies. It holds that $\Pi_{SD} \subset \Pi$ [148].

We call a measurable policy $\mathbf{d}$ *piecewise constant* if there exists some finite index m with $0 = t_0 < t_1 < \ldots < t_{m-1} < t_m = T$ such that $\mathbf{u}_t = \mathbf{u}_{t'}$ for $t, t' \in [t_i, t_{i+1})$ where $0 \leq i < m$ [127]. Note that in the case of a stationary policy the index $m = 1$.

For any $\mathbf{u} \in \mathscr{P}$ the infinitesimal generator matrix $\mathbf{Q}^{\mathbf{u}}$ is known whose (i,j)-th element is given by $\mathbf{Q}^{\mathbf{u}(i)}(i,j)$. Analogously, received rewards per transition are described

through $\mathbf{r}^{\mathbf{u}}$ whose i-th component is given by $r^{\mathbf{u}(i)}(i)$. Here transition rates and rewards depend on the selected decision and on the current state. Once decision u has been chosen the system operates as an induced CTMC [148].

For *Markov* policies [148] it holds that the selected decision depends on the system's past only through the current state. Similarly, since CTMDPs are generalizations of Markov chains, transition rates and rewards depend on the system's past only through the current state and the selected decision in that state [148]. Additionally, *deterministic* policies have the property that decisions are chosen with certainty.

Recall that at each decision epoch the decision maker has the required information about system's state and allowed decisions in that state. When the set of decision epochs is discrete, decisions are made at discrete points in time. When the set of decision epochs $\mathscr{H}$ is continuous, decisions could be made either continuously or randomly, e.g., at decision epochs correponding to occurrence of some events. Our description can be restricted to CTMDPs in which decisions are made at transition times [148, 24].

Time Horizon Infinite horizon problems are given when the set of decision epochs $\mathscr{H}$ is infinite. The model including an infinite horizon often occurs as an approximation of model with finite horizon, e.g., when the problem under study has finite but very large horizon, random horizon, or when horizon is fixed but random steps are used in the solution method [24].

Finite horizon CTMDP problems are given when the set of decision epochs $\mathscr{H} = \{0,\dots,T\}$ is finite [24]. Then decisions are not made at the final decision epoch denoted by T, instead the final decision is made at the previous decision epoch. In that case the reward at the last decision epoch T is only the function of state.

One can also deal with the case when the considered time horizon is an interval [148], it is then denoted as $\mathscr{H} = [0,T]$ with $T < \infty$. In the following we define the CTMDP model [24].

Definition 2.5. *A CTMDP is given by the tuple $(\mathscr{S},\mathscr{D},\mathscr{H},\mathscr{R},\mathscr{Q})$, where $\mathscr{S}$ is a state space, $\mathscr{D}$ is a decision space, $\mathscr{H}$ is a set of decision epochs, $\mathscr{R}$ is a set of rewards, and $\mathscr{Q}$ is a set of transition rates for all possible state-decision combinations. For a finite horizon $\mathscr{H}$ the terminal reward is a mapping $r_{fin} : \mathscr{S} \to \mathbb{R}$, $r_{fin}(i)$ giving the reward at final decision epoch T. If $\mathscr{H}$ is an infinite set, there is no terminal reward.*

Now suppose that we consider a system in a finite time interval $[0,T]$. As already defined in the literature [127, 42], let Ω be a set of all step functions on $[0,T]$ into $\mathscr{S}$, and let $\mathscr{F}$ be the σ-algebra of the sets in the space Ω induced by sets $\{\omega \in \Omega \,|\, \omega(t) = s_i\}$ for all $t \in [0,T]$, $i \in \mathscr{S}$. Applying a measurable policy $\mathbf{d}$ on the system results in a sample path ω describing states of the system at time t, $t \le T$. CTMC of the system is then generated by the probability space $(\Omega,\mathscr{F},\mathbf{P}^{\mathbf{d}}_{\pi})$ [42].

For almost all $r \in \{0,\dots,T\}$ with $0 \le r \le t \le T$ a matrix $\mathbf{V}^{\mathbf{d}}_{r,t}$ is defined [127, 42] such that the (i,j)-th element is defined as $\mathbf{P}^{\mathbf{d}}_{\pi}\{\omega(r) = s_i,\, \omega(t) = s_j\} \,/\, \mathbf{P}^{\mathbf{d}}_{\pi}\{\omega(r) = s_i\}$, for a sample path $\omega \in \Omega$ when policy $\mathbf{d}$ is used.

The element $\mathbf{V}^{\mathbf{d}}_{r,t}(i,j)$ contains the conditional probability that the induced CTMC is in state j at time point t given that CTMC is in state i at time point r and the policy $\mathbf{d}$ is used in interval $[r,t]$ [42]. Along the lines of Miller [127] the following condition

holds

$$\mathbf{V}_{r,t}^{\mathbf{d}} = \mathbf{I} + \mathbf{Q}^{\mathbf{d}(r)}(t-r) + o(t-r), \tag{2.17}$$

such that the matrix $\mathbf{V}_{r,t}^{\mathbf{d}}$ is shown to be defined by solutions of the differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{V}_{r,t}^{\mathbf{d}} = \mathbf{V}_{r,t}^{\mathbf{d}}\,\mathbf{Q}^{\mathbf{d}(t)}, \tag{2.18}$$

where the initial condition is $\mathbf{V}_{r,r}^{\mathbf{d}} = \mathbf{I}$. Transient probabilities $\mathbf{p}_t^{\mathbf{d}}$ of the CTMDP with initial probability distribution vector π at time 0 can be obtained by computing

$$\mathbf{p}_t^{\mathbf{d}} = \pi\mathbf{V}_{0,t}^{\mathbf{d}} \text{ and } \mathbf{p}_t^{\mathbf{d}} = \mathbf{p}_r^{\mathbf{d}}\mathbf{V}_{r,t}^{\mathbf{d}}. \tag{2.19}$$

Note that for a fixed policy $\mathbf{d}$ we obtain a stochastic process $\{X(t), Y(t)\}_{t\geq 0}^{T}$ which defines a *gain process* $\{G(t)\}_{t\geq 0}^{T}$ [42]. $G(t)$ describes the accumulated reward in the finite time interval $[0,t)$, $t \leq T$. Let vector $\mathbf{g}_{t,T}^{\mathbf{d}}$ contain the values of accumulated reward in the time interval $[t,T]$. Each time the process stays in state i and decision u is chosen the gain process changes with a rate $r^{u}(i)$.

Optimality Criterion Accordingly to a nondeterministic choice of decision and induced transition rates the sequence of rewards obtained in an CTMDP is stochastic. Since the aim of a CTMDP model is to control the system in such a way that some predefined optimization function is either maximized or minimized, the rewards are used to evaluate the selected decisions. In particular, different policies could be compared using the decision criteria, e.g., the *expected total reward* [98]. In the following basic decision criteria existing in MDP theory [98, 81] are described.

Let $\mathbf{r}^{\mathbf{u}_t}$ be the reward vector for decision vector $\mathbf{u}_t$ taken at time t, such that $\mathbf{r}^{\mathbf{u}_t}(i)$ is the expected reward gained by staying one time unit in state i . Furthermore, let $\mathbf{g}_T$ be the vector containing rewards gained at final decision epoch T. For the *total reward criterion* in finite horizons [98, 42], the expected reward accumulated in the time interval $[0,T]$ should be computed. The accumulated reward for some fixed policy $\mathbf{d}$ can be obtained as

$$\mathbf{g}_{t,T}^{\mathbf{d}} = \mathbf{V}_{t,T}^{\mathbf{d}}\mathbf{g}_T + \int_t^T \mathbf{V}_{\tau,T}^{\mathbf{d}}\mathbf{r}^{\mathbf{d}}\,\mathrm{d}\tau, \tag{2.20}$$

such that the second term describes the accumulated gain until time T.

In particular, matrix $\mathbf{V}_{\tau,T}^{\mathbf{d}}$ contains in position (i,j) the conditional probability that CTMDP is in state j at time T when it has been in state i at time τ and the policy $\mathbf{d}$ is used in the interval $[\tau,T]$. The vector $\mathbf{r}^{\mathbf{d}}$ is determined by $\mathbf{d}(\tau) = \mathbf{u}_\tau$ which is the corresponding decision vector at time $\tau \in [t,T]$. Then $\mathbf{u}_\tau(i)$ contains decisions if the system is in state i at time τ, and the vector $\mathbf{r}^{\mathbf{u}_\tau}(i)$ contains the corresponding expected rewards.

In particular, the vector $\mathbf{g}_{t,T}^{\mathbf{d}}$ contains at position i the expected reward accumulated in the time interval $[t,T]$ if the CTMDP is in state i at time t and the policy $\mathbf{d}$ has been used. Vector $\mathbf{g}^{\mathbf{d}}$ is denoted as the gain vector. The gain per state of policy $\mathbf{d}$ is defined as [42, 49]

$$\mathbf{g}_{0,T}^{\mathbf{d}} = \mathbf{V}_{0,T}^{\mathbf{d}}\mathbf{g}_T + \int_0^T \mathbf{V}_{\tau,T}^{\mathbf{d}}\mathbf{r}^{\mathbf{d}}\,\mathrm{d}\tau. \tag{2.21}$$

The expected reward which is also called the gain of policy $\mathbf{d}$ with initial probability distribution π is given by [42, 39]

$$G^{\mathbf{d}} = \pi\, \mathbf{g}^{\mathbf{d}}_{0,T}. \tag{2.22}$$

For the *time averaged reward criterion* [98, 42] one is commonly interested in an average reward accumulated per time unit in a long run. The gain vector is given by

$$\mathbf{g}^{\mathbf{d}}_{t,T} = \frac{1}{(T-t)} \left(\mathbf{V}^{\mathbf{d}}_{t,T} \mathbf{g}_T + \int_t^T \mathbf{V}^{\mathbf{d}}_{\tau,T} \mathbf{r}^{\mathbf{d}}\, \mathrm{d}\tau \right), \text{ for } T > t. \tag{2.23}$$

For the *accumulated reward to absorption criterion* [42] the absorbing states correspond to events of interest in the system, i.e. states satisfying certain properties. Then the gain vector describes the accumulated expected reward till absorption. Let $\mathscr{S}_A \in \mathscr{S}$ be a set of absorbing states, i.e. states satisfying some properties of interest. Let the vector $\mathbf{r}^{\mathbf{d}}_A$ contain zero rewards for absorbing states $i \in \mathscr{S}_A$ and $\mathbf{r}^{\mathbf{d}}_A(j)$ is determined by $\mathbf{r}^{\mathbf{u}_\tau}(j)$ otherwise. Then the gain vector is given by [42]

$$\mathbf{g}^{\mathbf{d}}_{t,T} = \mathbf{V}^{\mathbf{d}}_{t,T} \mathbf{g}_T + \int_t^T \mathbf{V}^{\mathbf{d}}_{0,\tau} \mathbf{r}^{\mathbf{d}}_A\, \mathrm{d}\tau, \tag{2.24}$$

where the second term determines the accumulated expected reward until time T on the state space $\mathscr{S} \setminus \mathscr{S}_A$. The vector $\mathbf{g}_T$ is the reward vector of the final decision epoch T. $\mathbf{g}_T$ is independent of the policy, and its initialization depends on the underlying computational problem as described in [42].

In the case of infinite horizons MDPs we consider the system on the interval $[0, \infty)$. Possible result measures for the system are the *accumuluted average reward*, the *accumulated reward to absorption*, or the *discounted reward*. For the *average reward criterion* [98] the gain vector at time 0 is given by

$$\mathbf{g}^{\mathbf{d}}_0 = \lim_{T \to \infty} \frac{1}{T} \left(\int_0^T \mathbf{V}^{\mathbf{d}}_{\tau,T} \mathbf{r}^{\mathbf{d}}\, \mathrm{d}\tau \right). \tag{2.25}$$

For the accumulated reward to absorption criterion [98, 42] the state space $\mathscr{S}$ is defined in a similar way to the above criterion for finite horizons.

Optimal values and policies Since the gain vector contains at position i the accumulated reward of the system under certain policy $\mathbf{d} \in \Pi$, the objective is to choose a measurable policy which minimizes (maximizes) the gain vector in all components and this policy is called the *optimal* policy [127]. We define [127, 42]

$$\mathbf{g}^{min}_{t,T} = \inf_{\mathbf{d} \in \Pi} \left(\mathbf{g}^{\mathbf{d}}_{t,T} \right), \qquad \mathbf{g}^{max}_{t,T} = \sup_{\mathbf{d} \in \Pi} \left(\mathbf{g}^{\mathbf{d}}_{t,T} \right) \tag{2.26}$$

as the extreme (optimal) values for the gain vector $\mathbf{g}^{\mathbf{d}}_{t,T}$, $0 \le t \le T$. A policy is called optimal if it results in minimal (maximal) extreme values [127, 42]

$$\mathbf{d}^{min} = \arg \inf_{\mathbf{d} \in \Pi} \left(\mathbf{g}^{\mathbf{d}}_{t,T} \right), \qquad \mathbf{d}^{max} = \arg \sup_{\mathbf{d} \in \Pi} \left(\mathbf{g}^{\mathbf{d}}_{t,T} \right). \tag{2.27}$$

The policies $\mathbf{d}^{min}$, $\mathbf{d}^{max}$ need not be unique. Often the ε-optimal policies are considered, where it holds for an ε-optimal policy $\mathbf{d}$ that $\left| \mathbf{g}^{min/max}_{t,T} - \mathbf{g}^{\mathbf{d}}_{t,T} \right| \le \varepsilon$. The problem of computing the minimum (maximum) accumulated average reward with infinite

horizon and the problem of computing the minimum (maximum) long-run average reward with finite horizon admit stationary optimal policies as described in [28] – optimal policies can be computed using dynamic programming [28, 148]. The minimum (maximum) total reward problem admits piecewise constant optimal policies, which is proved in [127].

DTMDPs After discussing continuous-time MDPs we will give a precise definition of *discrete-time* MDPs (DTMDP) where time is also divided into periods or *stages* [148, 24], such that time steps are assumed to be equal. If time horizon $\mathscr{H}$ is infinite, there is no terminal reward.

Definition 2.6. *[24] A DTMDP is given by the tuple $(\mathscr{S},\mathscr{D},\mathscr{H},\mathscr{R}',\mathscr{Q}')$, where $\mathscr{S}$ is a state space, $\mathscr{D}$ is a decision space, $\mathscr{H}$ is a set of decision epochs with $\mathscr{H}=\{0,\ldots,T\}$, $T\leq\infty$, $\mathscr{R}'$ is a set of rewards, and $\mathscr{Q}'$ is a set of transition probabilities for all possible state-decision combinations.*

The value given by $r_n^{'u}(i)$ is a *one-stage* reward at decision epoch n if the current state is i and decision u is chosen. At next decision epoch $n+1$ the system state is determined according to probability distribution given by $\mathscr{Q}'$. Set $\mathscr{Q}'$ contains stochastic transition kernels $\mathbf{P}^u$ such that entry $\mathbf{P}^u(i,j)$ gives the probability that a next state at time $n+1$ is j if the current state is i and decision u is taken at time n [24].

DTMDPs are of fundamental interest in the analysis of CTMDPs. Since CTMDPs have to be solved numerically, a possible solution can be also obtained by considering an embedded state process which can be determined by converting a CTMDP model using uniformization [148].

Similarly to the continuous-time case, the basic criteria are the total reward, and the average reward criterion for problems with infinite horizon [148, 28]. For the expected total reward criterion on infinite horizons the gain vector at time 0 for the given initial state i is given by

$$\mathbf{g}_0^{\mathbf{d}}(i)=\lim_{T\to\infty}\frac{1}{T}E_{\mathbf{d},i}\left(\sum_{t=0}^{T}\mathbf{r}'^{\mathbf{u}_t}(X_t)\right),\mathbf{d}\in\Pi,i\in\mathscr{S},\tag{2.28}$$

where the limit defining the total reward $\mathbf{g}_0^{\mathbf{d}}(i)$ exists [28].

If we consider the system on the finite interval $[0,T]$, the total expected reward is given by [28]

$$\mathbf{g}_0^{\mathbf{d}}(i)=E_{\mathbf{d},i}\left(\mathbf{g}_T+\sum_{t=0}^{T-1}\mathbf{r}'^{\mathbf{u}_t}(X_t)\right),\mathbf{d}\in\Pi,i\in\mathscr{S},\tag{2.29}$$

where vector $\mathbf{g}_T$ is the initial gain vector at time T. $\mathbf{g}_T$ is independent of policy $\mathbf{d}\in\Pi$ and is added to the accumulated reward of the first T stages. The author in [28] proposes the dynamic programming approach to determine optimal value and policy for the above criterion. In particular, the optimal gain vector for a given initial state i is the solution of the following equation

$$\mathbf{g}_0^{max}(i)=\max_{u\in\mathscr{D}(i)}\left(\mathbf{r}'^{u}(i)+\sum_{j\in\mathscr{S}}\mathbf{P}^u(i,j)\mathbf{g}_0^{max}(j)\right),\forall i\in\mathscr{S}.\tag{2.30}$$

Observe that infinite horizont CTMDPs can be numerically analyzed in several ways [148]. One possible solution is to transform CTMDP into DTMDP and Poisson process through a uniformization method. Afterwards, e.g., algorithms for computation of optimal policy and its value can be applied to the resulting DTMDP such that Poisson process need not to be considered. The uniformization method for CTMCs is described in Sec. 2.1.2.

***Example* 2.3.** *In this example we consider the stochastic job scheduling problem (sJSP) presented in [37]. The authors considered two problems, namely minimizing the expected makespan and minimizing the expected flow time. Therefore the scheduling problem is given by a finite set of tasks with exponentially distributed service time on more than one identical processors.*

Main results developed in [37] show, that policy $\mathbf{d}$ *with a longest expected time first strategy (LEPT) is makespan optimal, i.e. it minimizes the expected completion time of the sJSP. Policy with a shortest expected processing time first strategy (SEPT) is flow time optimal.*

We present the CTMDP model formalized in [131]. Let $J = \{1,\ldots,n\}$ *be a set of jobs. A state space* $\mathscr{S} = 2^J \times 2^J$ *is given by tuples* (R,W)*, such that R describes a set of jobs already assigned to processors, W describing a set of unfinished jobs waiting for a processor. For each job* $i \in J$ $\mu(i)$ *describes a rate of the exponentially distributed processing time. All processing times are independent and exponentially distributed.*

Decision epochs are determined by completion times of jobs. Each time a job i finishes a scheduling decision which job to schedule next is made. Formally, a decision $u \in \mathscr{D}((R,W))$ *defines the preemptive schedule and a set* $u(i)$ *determines the tasks to be assigned to a processor. When job i finishes and decision u has been chosen, a next state is* (R',W') *with* $R' = u(i)$ *and* $W' = R \cup W \setminus \{i\} \cup \{u(i)\}$.

For decision $u \in \mathscr{D}((R,W))$ *the transition rate matrix* $\mathbf{Q}^u$ *is defined as*

$$\mathbf{Q}^u((R,W),(R',W')) = \begin{cases} \mu(i) & \text{if job } i \text{ finishes} \\ 0 & \text{otherwise.} \end{cases}$$

Consider now the instance with $m = 2$ *identical processors and* 4 *jobs with* $\mu(1) = 3$, $\mu(2) = 2$, $\mu(3) = 5$, $\mu(4) = 7$. *Fig. 2.5 shows the CTMDP with initial state* $(R = \{2,4\}, W = \{1,3\})$. *In the case when job* 2 *finishes first, job* 4 *is preempted and jobs*

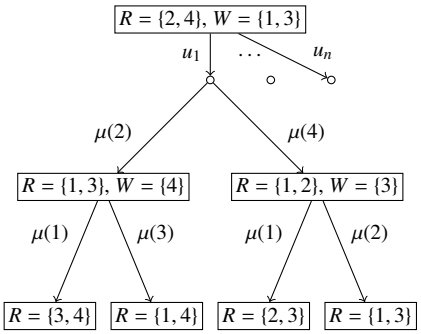


Figure 2.5.: CTMDP for the sJSP instance.

1, 3 *are assigned to processors. In the case when job* 4 *finishes first, jobs* 1,2 *are*

assigned next. Note that jobs with a smallest rate, i.e. with a longest mean time are assigned first. For the visualized scheduling decision u_1 waiting jobs are jobs with a highest rate which corresponds to the scheduling policy with the smallest expected makespan.

2.3. Stochastic Shortest Path Problems

In this section we describe the *stochastic shortest path problem* (SSPP) which is the most studied problem in random graphs. It is a stochastic version of the deterministic shortest path problem.

Assume that a graph with n nodes, edge weights, and a certain destination node d is given. The edge (i, j) is defined when for a node i the successor node j is selected. A path $q = ((i, j), (j,k), \ldots, (l,d))$ is defined as a sequence of edges that connect one node with another. Let $c_q(i,d)$ define the sum of edge weights for a path q starting in i and ending in d. Then starting at some node i we are interested in a successor node j for each node, such that the shortest path formed by a sequence of successor nodes satisfies [148, 28]

$$\arg\min_{\forall q} c_q(i,d).$$

In a SSPP the edge weights are given by random variables and are represented by rewards. Often nodes of the graph are represented by states. For some node i the possible successor nodes define edges such that the choice of the successor edge is associated with an admissible decision in the current state. This implies that transitions between nodes are random, such that some control over their probability distributions is allowed in a SSPP.

The problem is to find a policy which minimizes the expected cost of reaching a given target state, such that the stochastic shortest path has the minimal expected length. The SSPP are undiscounted MDPs with an absorbing, cost-free terminal state corresponding to the destination d. Based on this formulation the SSPP can be solved using standard MDP methods, as shown in [29, 28].

The deterministic version of the dynamic program is given when the selected decision determines the successor state with certainty, i.e. when the associated probability distribution assigns probability 1 to the successor state. The graphic representation of the deterministic dynamic program formulation for a SSPP instance is shown in Fig. 2.6. The aim is to compute the optimal policy which leads to the destination node with probability 1 and results in minimal expected total reward [148, 28].

2.3.1. Literature Overview

The huge application area of stochastic shortest path problems includes online stochastic route planning [136, 135], robot navigation [160], minimum and maximum reachability times [59, 42] to name a few. There are some prominent treatments in the literature, e.g., Andreatta and Romeo [7] introduced the version of *I-SSPP* (independent edge weights) where random edges can be either active or inactive in deterministic graphs with stochastic topology. The decision maker knows the system state containing information about active and inactive edges which enables him to make a decision to reroute each time the system state corresponding to a node in a graph is reached.

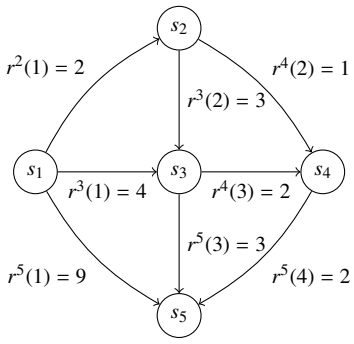


Figure 2.6.: Example of the deterministic SSPP. The destination node is s_5. All actions are deterministic since they lead to an unique successor state with probability 1.

The I-SSPP problem was subsequently studied in [145]. The authors also considered the joint probability distribution of random variables describing edge weights which presumes dependent random variables. The problem under this model is called *R-SSPP*. In this variant the edge weights are learned as the decision maker traverses the graph such that the realizations of random variables describing edge weights afterwards remain constant and known by it. However, the proposed R-SSPP algorithm based on a dynamic programming approach has exponential run time in the number of realizations of the network. The algorithm for I-SSPP is exponential in the number of edges in the graph. The authors also show that the recognition version of the R-SSPP is NP-complete, and that the I-SSPP is #P-hard, and can be computed in polynomial space [145, Theorem 3,4].

In [29] the authors obtained optimality results for the problem where at each node the probability distribution over all possible successor nodes is available. The problem with negative edge weights has been treated under some important assumptions. Then it has been shown that the optimal cost vector is the unique solution of Bellman's equation and that the policy iteration approach computes an optimal stationary policy starting from arbitrary policy. Further real-time dynamic programming approach based on Markov decision theory was proposed in [33]. For further complexity results and heuristic algorithms we refer to [143, 145, 136, 135, 175].

In [34] the authors treated the bus network problem in order to compute an optimal plan within a city minimizing the expected traveling time. The formulation of the bus network problem included a time-dependent Markov decision process where the decisions in each state are whether or not to take a bus when it arrives. The model has then been extended in [35] such that stochastic state transitions as well as stochastic, time-dependent action durations were added to the CTMDP. For further time-dependent versions of SSPP we refer to the literature [82, 78, 174].

In the field of robotics, SSPP often builds a basis for mobile robot navigation. In [160] the authors developed partially observable MDPs for autonomous office navigation. The system state contains information about environment topology, distance, sensor and actuator data. This enables one to estimate the position of a robot from the Markov model and the decision maker chooses the decision about navigation with respect to temporary uncertainty in position and sensor data. The 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, was proposed in [36]. We refer to the literature for further treatments using partially observable MDPs [160, 167].

2.3.2. SSPP Definition

In this section we first provide basic definitions and notations for the stochastic version of the shortest path problem. Then we proceed with a description of basic solution methods and relevant approaches on the field.

Definition 2.7. *[59] A SSPP can be described as an infinite horizon discrete-time MDP* $(\mathscr{S},\mathscr{D},\mathscr{H},\mathscr{R}',\mathscr{Q}')$*, where it holds for the state space* $\mathscr{S}=\mathscr{S}_T\cup\mathscr{S}_A$*. A set* $\mathscr{S}_T=\{1,\ldots,n\}$ *contains transient states which are associated with n graph nodes. A set* $\mathscr{S}_A$ *contains absorbing states. For a single destination problem, the set* $\mathscr{S}_A$ *contains a single absorbing state* $n+1$ *corresponding to the destination node in the graph.*

The considered graph is deterministic in the sense that only edge weights are described by random variables but the edges themselves are certain. In the SSPP the discount factor $\gamma=1$ and absorbing states are reward free.

In each state i the set of admissible decisions $\mathscr{D}(i)$ is associated with a set of possible successor nodes of the node i. When the decision u is selected at some time the transition probabilities $\mathbf{P}^u(i,j)$ for possible successor states j are specified.

In particular, $\mathbf{P}^u$ is a substochastic $|\mathscr{S}_T|\times|\mathscr{S}_T|$ matrix, where $\mathbf{P}^u(i,j)$ gives the probability $Prob(X(t+1)=j\,|\,X(t)=i,u_t=u)$, for $i,\ j\in\mathscr{S}_T$, and $u\in\mathscr{D}(i)$. If we are dealing with a single destination problem, it holds that $Prob(X(t+1)=n+1\,|\,X(t)=n+1)=1$ for all t. In the following we assume that $\mathscr{S}_A=\{n+1\}$.

Additionally, $\tilde{r}^u(i)$ are the costs of the system if the current state is i and decision u is chosen. The destination state has the zero costs $\tilde{r}(n+1)=0$ for all policies.

In the analysis of SSPP models one deals with averaging of rewards per stage over all possible successor states [28]. Thus the expected reward per stage for state i using decision $u\in\mathscr{D}(i)$ is defined as follows

$$\mathbf{r}^u(i)=\sum_{j\in\mathscr{S}}\mathbf{P}^u(i,j)\,\tilde{r}^u(j) \tag{2.31}$$

The SSPP is the computation of the minimum expected total reward of reaching the absorbing state $n+1$ in the defined DTMDP when the decision maker applies the policy that reaches $n+1$ with probability 1 [28, 29, 59]. However, the existence of such policies is not guaranteed and will be discussed in detail in Sec. 2.3.3.

The total expected reward for policy $\mathbf{d}=(\mathbf{u}_0,\mathbf{u}_1,\ldots)$, $\mathbf{d}\in\Pi$, starting in state i is given by [27]

$$\mathbf{g}^{\mathbf{d}}(i)=\lim_{N\to\infty}E\left[\sum_{t=0}^{N-1}\mathbf{r}^{\mathbf{u}_t(X(t))}(X(t))\,|\,X(0)=i\right]. \tag{2.32}$$

Note that the stationary policy is given by $\mathbf{d}=(\mathbf{u},\mathbf{u},\ldots)$, $\mathbf{d}\in\Pi$ (cf. Sec 2.2). In the following we simplify the notation for $\mathbf{d}$ by denoting $\mathbf{u}$ as a stationary policy and the corresponding gain function as $\mathbf{g}^{\mathbf{u}}$.

The minimal total expected reward starting in state i is defined by [28, 29]

$$\mathbf{g}^*(i) = \min_{\mathbf{u}\in\Pi} \mathbf{g}^{\mathbf{u}}(i), \tag{2.33}$$

such that it contains minimum of $\mathbf{g}^{\mathbf{u}}(i)$ for all admissible policies $\mathbf{u} \in \Pi$. Given stationary policy $\mathbf{u}$ the transition probability matrix $\mathbf{P}^{\mathbf{u}}$ of dimension $|\mathscr{S}_T| \times |\mathscr{S}_T|$ can be defined as

$$\mathbf{P}^{\mathbf{u}}(i,j) = \mathbf{P}^{\mathbf{u}(i)}(i,j), \tag{2.34}$$

which is substochastic since the absorbing state is not considered.

We first introduce mappings $T_{\mathbf{u}} : \mathbb{R}^n \to \mathbb{R}^n$ and $T : \mathbb{R}^n \to \mathbb{R}^n$ defined in [27, 28] by

$$T_{\mathbf{u}}\,\mathbf{g}(i) = \mathbf{r}^{\mathbf{u}(i)}(i) + \sum_{j\in\mathscr{S}_T} \mathbf{P}^{\mathbf{u}}(i,j)\,\mathbf{g}(j), \tag{2.35}$$

$$T\,\mathbf{g}(i) = \min_{u\in\mathscr{D}(i)} \left(\mathbf{r}^{u}(i) + \sum_{j\in\mathscr{S}_T} \mathbf{P}^{u}(i,j)\,\mathbf{g}(j) \right), \tag{2.36}$$

for any function $\mathbf{g} : \mathscr{S} \to \mathbb{R}$. The operator $T_{\mathbf{u}}$ maps $\mathbf{g}$ to the vector $T_{\mathbf{u}}\,\mathbf{g}$. The $T_{\mathbf{u}}\,\mathbf{g}$ is the reward function associated with the policy $\mathbf{u}$ for the one-period problem.

For any function $\mathbf{g} : \mathscr{S} \to \mathbb{R}$ the operators $T_{\mathbf{u}}$, T can be given in vector notation form as

$$\begin{aligned}
&\mathbf{g} = [\mathbf{g}(1), \ldots, \mathbf{g}(n)]^T, \\
&T_{\mathbf{u}}\,\mathbf{g} = [T_{\mathbf{u}}\,\mathbf{g}(1), \ldots, T_{\mathbf{u}}\,\mathbf{g}(n)]^T, \\
&T\,\mathbf{g} = [T\,\mathbf{g}(1), \ldots, T\,\mathbf{g}(n)]^T, \\
&\mathbf{r}^{\mathbf{u}} = \left[\mathbf{r}^{\mathbf{u}(1)}(1), \ldots, \mathbf{r}^{\mathbf{u}(n)}(n)\right]^T.
\end{aligned}$$

The operator $T_{\mathbf{u}}\,\mathbf{g}$ in Eq. 2.35 associated with a stationary policy $\mathbf{u}$ can be written in vector matrix notation

$$T_{\mathbf{u}}\mathbf{g} = \mathbf{r}^{\mathbf{u}} + \mathbf{P}^{\mathbf{u}}\mathbf{g}. \tag{2.37}$$

The reward function of an arbitrary policy $\mathbf{d} = (\mathbf{u}_0, \mathbf{u}_1, \ldots)$, $\mathbf{d} \in \Pi$, and N describing the time horizon length, can be written as [28]

$$\mathbf{g}^{\mathbf{d}} = \limsup_{N\to\infty} T_{\mathbf{u}_0} T_{\mathbf{u}_1} \cdots T_{\mathbf{u}_{N-1}} \mathbf{0} = \limsup_{N\to\infty} \left(\mathbf{r}^{\mathbf{d}(0)} + \sum_{t=1}^{N-1} \mathbf{P}^{\mathbf{u}_0} \cdots \mathbf{P}^{\mathbf{u}_{t-1}} \mathbf{r}^{\mathbf{d}(t)} \right), \tag{2.38}$$

which results in the following equation for a stationary policy $\mathbf{u}$

$$\mathbf{g}^{\mathbf{u}} = \limsup_{N\to\infty} (T_{\mathbf{u}})^N \mathbf{0} = \limsup_{N\to\infty} \sum_{t=0}^{N-1} (\mathbf{P}^{\mathbf{u}})^t \mathbf{r}^{\mathbf{u}}. \tag{2.39}$$

2.3.3. Proper and improper policies

According to the definition of the SSPP involving a policy that reaches the absorbing state $n+1$ with probability 1 we introduce the concept for such a policies along the lines of Bertsekas [28]. Intuitively, if an admissible policy $\mathbf{d} \in \Pi$ reaches the state $n+1$ with probability 1, then in the induced absorbing Markov chain there is a path connecting each state i with the absorbing state $n+1$ and containing non-zero transition probabilities along this path.

Definition 2.8. *[29, 28, Definition 2.1.1] A stationary policy* $\mathbf{u}$ *is* proper *if it reaches the absorbing state with positive probability after at most n stages. That is*

$$p^{\mathbf{u}} = \max_{i \in \mathscr{S}_T} Prob(X(n) \neq n+1 \,|\, X(0) = i, \mathbf{u}) < 1, \tag{2.40}$$

such that $\sum_{t=0}^{\infty} (\mathbf{P}^{\mathbf{u}})^t$ *is finite. Otherwise the stationary policy* $\mathbf{u}$ *is said to be* improper.

It has been shown in [28] that if the proper policy $\mathbf{u}$ is used the following holds

$$Prob(X(t) \neq n+1 \,|\, X(0) = i, \mathbf{u}) \leq (p^{\mathbf{u}})^{\lfloor \frac{t}{n} \rfloor} < 1, \tag{2.41}$$

for $i \in \mathscr{S}_T$, which is the maximal probability of not reaching the absorbing state $n+1$ after t stages. It follows from Eq. 2.41 that the absorbing state will be reached with probability 1 [28], i.e. $\lim_{t\to\infty} (\mathbf{P}^{\mathbf{u}})^t = 0$. In particular, if the proper policy $\mathbf{u}$ is used, the associated total gain is finite, i.e. the expected total reward in the t-th decision epoch is bounded by [28]

$$(p^{\mathbf{u}})^{\lfloor \frac{t}{n} \rfloor} \cdot \max_{i \in \mathscr{S}_T} | \, r^{\mathbf{u}(i)}(i) \, |, \tag{2.42}$$

such that the expected total reward starting in state i is finite

$$\mathbf{g}^{\mathbf{u}}(i) \leq \lim_{T\to\infty} \sum_{t=0}^{T-1} \left[(p^{\mathbf{u}})^{\lfloor \frac{t}{n} \rfloor} \cdot \max_{i \in \mathscr{S}_T} | \, r^{\mathbf{u}(i)}(i) \, | \right] < \infty. \tag{2.43}$$

In [28] the authors introduce the following two important assumptions for the dynamic programming theory according to the SSPP.

1. The existence of at least one proper policy is required.

2. Each improper policy $\mathbf{u}$ results in an infinite expected total reward for at least one initial state, such that some component of the sum $\sum_{t=0}^{N-1} (\mathbf{P}^{\mathbf{u}})^t \, \mathbf{r}^{\mathbf{u}}$ diverges to ∞ as $N \to \infty$.

These assumptions are satisfied in, e.g., the deterministic versions of SSPP shown in Fig. 2.7. Here at least one proper policy exists if there is a path connecting every initial node i with the destination node $(n+1)$ [29]. For example the policy $\mathbf{u}_1$ shown in Fig. 2.7 is proper because the resulting paths connect each node with the destination 4.

A policy is improper if there exists some initial state i, such that the path starting in i doesn't lead to the destination state. The same holds if the path starting in i contains infinitely cycles of positive length. In that case the costs incurred for an initial state i are infinite.

***Example* 2.4.** *The policy* $\mathbf{u}_2$ *shown in Fig. 2.7 is improper since the second assumption does not hold. The policy minimizing Eq. 2.39 will always choose the decision* $\mathbf{u}_2(3') = 2$ *since* $r^{\mathbf{u}_2}(3') = 0$*. The decision rule* $\mathbf{u}_2(3') = 2$ *results in expected total reward* $\mathbf{g}^{\mathbf{d}_2}(1') = 1$ *for the initial state* $1'$*. Though the proper policy with decision rule* $\mathbf{u}_2(3') = 4$ *would result in value* 2 *for the expected total reward starting in* $1'$*.*

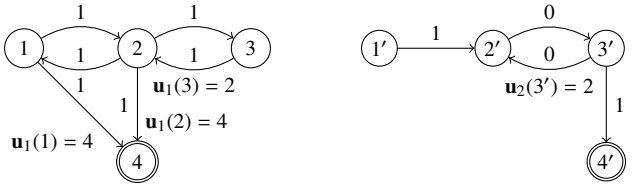


Figure 2.7.: Two instances of deterministic SSPP. The destination node is 4, and $4'$ respectively. All actions are deterministic since they lead with probability 1 to the unique successor state. The rewards are depicted on edges.

If all policies terminate inevitably in the destination node $n+1$ the two assumptions are also satisfied. Furthermore if it holds for all rewards $r^u(i) > 0$, then the second assumption is satisfied [28].

Observe that the first assumption about the existence of at least one proper policy in fact states that the destination node will be reached with probability 1 in a finite number of steps. The length of the time horizon depends on the policy and thus is random [27].

We can conclude from the above assumptions that a proper policy is given when paths starting in initial states reach a destination with probability 1, such that the corresponding total reward is equal to the path length. An important property of a proper policy $\mathbf{u}$ is that the attained rewards $\mathbf{g}^{\mathbf{u}}(i)$ starting in state i, for $i = 1,\ldots,n$, are the unique solution of Bellman's equation [27, 28, 148]

$$\mathbf{g}^{\mathbf{u}}(i) = \mathbf{r}^{\mathbf{u}(i)}(i) + \sum_{j \in \mathscr{S}_T} \mathbf{P}^{\mathbf{u}}(i,j)\,\mathbf{g}^{\mathbf{u}}(j),\ i = 1,\ldots,n, \tag{2.44}$$

such that the gain vector $\mathbf{g}^{\mathbf{u}}$ with components $\mathbf{g}^{\mathbf{u}}(i)$ is the unique fixed point of the mapping $T_{\mathbf{u}}$.

In turn, the optimal total expected rewards $\mathbf{g}^*(i)$ starting in state i, $i = 1,\ldots,n$, are the unique solution of Bellman's equation (see [27, Proposition 7.2.1(b)])

$$\mathbf{g}^*(i) = \min_{u \in \mathscr{D}(i)} \left(\mathbf{r}^u(i) + \sum_{j \in \mathscr{S}_T} \mathbf{P}^u(i,j)\,\mathbf{g}^*(j) \right),\ i = 1,\ldots,n, \tag{2.45}$$

where the optimal reward vector $\mathbf{g}^*$ containing components $\mathbf{g}^*(i)$ is the unique fixed point of the mapping T.

2.3.4. Dynamic programming algorithm

In this section we discuss the *dynamic programming method* and the *Bellman's optimality equation* which holds for the expected total reward minimization in SSPP under the above assumptions. In the following we propose the main results from the dynamic programming theory for SSPP developed in [29, 28].

First, let $\mathbf{g}_0$ be the $1 \times n$ vector of zeros such that $\mathbf{g}_0(i) = 0$ for all $i \in \mathscr{S}_T$. The mapping T^k denotes the function which results from applying the mapping T in Eq. 2.36 to the function T^{k-1}

$$(T^k \mathbf{g}_0)(i) = (T\,(T^{k-1}\mathbf{g}_0))(i), \tag{2.46}$$

such that if we start with $(T^0\mathbf{g}_0)(i) = \mathbf{g}_0(i)$, the optimal reward $T^k\mathbf{g}_0(i)$ for the k-stage problem can be obtained. In particular, $T^k\mathbf{g}_0(i)$ is the minimal expected total reward for reaching an absorbing state $n+1$ starting from the initial state i in the k-stage problem.

Similarly, the mapping $T_{\mathbf{u}}^k$ is given by

$$(T_{\mathbf{u}}^k \mathbf{g}_0)(i) = (T_{\mathbf{u}}\,(T_{\mathbf{u}}^{k-1}\mathbf{g}_0))(i), \tag{2.47}$$

describing the rewards corresponding to a stationary policy $\mathbf{u}$ for reaching an absorbing state $n+1$ starting from the initial state i in the k-stage problem.

Applying the mapping $T_{\mathbf{u}}$ to the function $(T_{\mathbf{u}}^{k-1}\mathbf{g})$ for a stationary policy $\mathbf{u}$ can be resolved inductively by

$$(T_{\mathbf{u}}^k \mathbf{g}) = (\mathbf{P}^{\mathbf{u}})^k \mathbf{g} + \sum_{n=0}^{k-1} (\mathbf{P}^{\mathbf{u}})^n \mathbf{r}^{\mathbf{u}}.$$

To illustrate the case where $k = 2$, first observe for $k = 1$ that

$$T_{\mathbf{u}}\mathbf{g} = \mathbf{r}^{\mathbf{u}} + \mathbf{P}^{\mathbf{u}}\mathbf{g},$$

and

$$(T_{\mathbf{u}}^2 \mathbf{g}) = T_{\mathbf{u}}(T_{\mathbf{u}}\mathbf{g}) = T_{\mathbf{u}}(\mathbf{r}^{\mathbf{u}} + \mathbf{P}^{\mathbf{u}}\mathbf{g}) = \mathbf{r}^{\mathbf{u}} + \mathbf{P}^{\mathbf{u}}(\mathbf{r}^{\mathbf{u}} + \mathbf{P}^{\mathbf{u}}\mathbf{g}) = \mathbf{r}^{\mathbf{u}} + \mathbf{P}^{\mathbf{u}}\mathbf{r}^{\mathbf{u}} + (\mathbf{P}^{\mathbf{u}})^2\mathbf{r}^{\mathbf{u}}.$$

***Example* 2.5.** *Let us consider a SSPP instance with $\mathscr{S}_T = \{1,2\}$, $\mathscr{S}_A = \{3\}$, and the set of admissible decisions $\mathscr{D}(i) = \{u_1, u_2\}$ for all $i \in \mathscr{S}_T$. The transition probability*

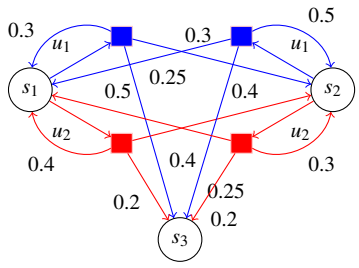


Figure 2.8.: State transition graph including admissible decisions u_1, u_2 and the induced transition probabilities.

matrices corresponding to the decisions u_1, u_2 and rewards are given below

$$\mathbf{P}^{u_1} = \begin{bmatrix} 0.3 & 0.3 & 0.4 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{P}^{u_2} = \begin{bmatrix} 0.4 & 0.4 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{r}^{u_1} = (0.6, 0.75, 0)^T, \mathbf{r}^{u_2} = (0.85, 0.8, 0)^T.$$

The function T can be now computed for the states $1, 2, 3$ as

$$T\mathbf{g}(i) = \min_{u_1, u_2} \left(\mathbf{r}^{u_1}(i) + \sum_{j=1}^{3} \mathbf{P}^{u_1}(i,j)\,\mathbf{g}(j), \mathbf{r}^{u_2}(i) + \sum_{j=1}^{3} \mathbf{P}^{u_2}(i,j)\,\mathbf{g}(j) \right).$$

When the initial expected reward vector is defined as $\mathbf{g}_0 = (0,0,0)$, we obtain

$$T\mathbf{g}_0(1) = 0.6, \;\; T\mathbf{g}_0(2) = 0.75, \;\; T\mathbf{g}_0(3) = 0.$$

Convergency results Let again $\mathbf{g}_0 = \mathbf{0}$. Observe that for every stationary policy $\mathbf{u}$ its infinite horizon reward is the limit of the k-period reward associated with $\mathbf{u}$ as $k \to \infty$ [28, Corollary 1.2.1.1]

$$\mathbf{g}^{\mathbf{u}} = \lim_{k\to\infty} \left(T_{\mathbf{u}}^k \mathbf{g}_0 \right). \tag{2.48}$$

Observe that Eq. 2.44 states that the gain vector $\mathbf{g}^{\mathbf{u}}$ corresponding to policy $\mathbf{u}$ is the unique solution of the Bellman's equation. Equivalently the gain vector $\mathbf{g}^{\mathbf{u}}$ is the solution of the equation

$$\mathbf{g}^{\mathbf{u}} = T_{\mathbf{u}} \mathbf{g}^{\mathbf{u}} = \mathbf{r}^{\mathbf{u}} + \mathbf{P}^{\mathbf{u}} \mathbf{g}^{\mathbf{u}}. \tag{2.49}$$

The dynamic programming iteration computes the values

$$\mathbf{g}_{k+1}(i) = \min_{u\in\mathscr{D}(i)} \left(\mathbf{r}^u(i) + \sum_{j\in\mathscr{S}_T} \mathbf{P}^u(i,j)\, \mathbf{g}_k(j) \right),\ i = 1,\ldots,n, \tag{2.50}$$

such that the sequence of generated values converges to $\mathbf{g}^*(i)$. Then the optimal infinite horizon total expected reward is equal to the limit of the optimal k-period expected reward [28, Corollary 1.2.1]

$$\mathbf{g}^* = \lim_{k\to\infty} \left(T^k \mathbf{g}_0 \right). \tag{2.51}$$

Thus if we start with $(T^0\mathbf{g}_0)(i) = \mathbf{g}_0(i)$ and iterate the dynamic programming algorithm infinitely often, we compute in the limit the optimal reward $\mathbf{g}^*$.

Eq. 2.50 can be transformed to dynamic programming for finite horizon by reversing the time index. The initial reward function $\mathbf{g}_0$ equals to the terminal reward function. If we consider the k-decision epoch problem, then the value $\mathbf{g}_k(i)$ represents the optimal reward starting from state i and obtaining terminal reward $\mathbf{g}_0$ at the end of k-th decision epoch.

Observe that the optimal gain vector $\mathbf{g}^*$ satisfies the *Bellman's equation* [28, Proposition 1.2.2]

$$\mathbf{g}^* = T\mathbf{g}^*, \tag{2.52}$$

which is equivalent to Eq. 2.45. In fact, the Bellman's equation can be expressed as the dynamic programming algorithm taken to its limit as $k \to \infty$.

The stationary policy $\mathbf{u}$ is optimal if $\mathbf{u}(i)$ results in the minimum in the righthand side of the Bellman's equation 2.52 for each $i \in \mathscr{S}$. Thus it holds that

$$T\mathbf{g}^* = T_{\mathbf{u}}\mathbf{g}^*, \tag{2.53}$$

where $T_{\mathbf{u}}\mathbf{g}$ is defined in Eq. 2.49.

For the stationary policy $\mathbf{u}$ the total expected reward function can be computed as

$$(\mathbf{I} - \mathbf{P}^{\mathbf{u}})\,\mathbf{g}^{\mathbf{u}} = \mathbf{r}^{\mathbf{u}}, \tag{2.54}$$

which follows from the fact that the equation $\mathbf{g}^{\mathbf{u}} = T_{\mathbf{u}}\mathbf{g}^{\mathbf{u}}$ results in a system of n linear equations with n unknowns $i = 1,\ldots,n$, each corresponding to the vector component $\mathbf{g}^{\mathbf{u}}(i)$

$$\mathbf{g}^{\mathbf{u}}(i) = \mathbf{r}^{\mathbf{u}}(i) + \sum_{j=1}^{n} \mathbf{P}^{\mathbf{u}}(i,j)\, \mathbf{g}^{\mathbf{u}}(j). \tag{2.55}$$

Furthermore the Eq. 2.54 can be equivalently written as

$$\mathbf{g}^{\mathbf{u}} = (\mathbf{I} - \mathbf{P}^{\mathbf{u}})^{-1}\mathbf{r}^{\mathbf{u}}. \tag{2.56}$$

Note, that the matrix $\mathbf{P}^{\mathbf{u}}$ is substochastic since it is restricted to transient states. We now have to show that the matrix $(\mathbf{I} - \mathbf{P}^{\mathbf{u}})$ is invertible. Consider the homogeneous equation $(\mathbf{I} - \mathbf{P}^{\mathbf{u}})\mathbf{x} = 0$, i.e. $\mathbf{x} = \mathbf{P}^{\mathbf{u}}\mathbf{x}$. Taking the power of $(\mathbf{P}^{\mathbf{u}})^n$ it admits that $\mathbf{x} = (\mathbf{P}^{\mathbf{u}})^n\mathbf{x}$. Observe that $\lim_{n\to\infty}(\mathbf{P}^{\mathbf{u}})^n = \mathbf{0}$, i.e. the absorption occurs with probability 1. Then $\lim_{n\to\infty}(\mathbf{P}^{\mathbf{u}})^n\mathbf{x} = \mathbf{0}$, so $\mathbf{x} = \mathbf{0}$. In this case the equation $(\mathbf{I} - \mathbf{P}^{\mathbf{u}})\mathbf{x} = 0$ has only the trivial solution $\mathbf{x} = \mathbf{0}$, which is the necessary and sufficient condition for the matrix $(\mathbf{I} - \mathbf{P}^{\mathbf{u}})$ to be invertible (cf. [105, Theorem 3.2.1]).

The optimal stationary policy $\mathbf{u}^*$ and the corresponding gain vector $\mathbf{g}^*$ can be computed using value iteration, policy iteration and linear programming which we briefly describe in Chap. 5.

2.4. Phase-Type Distributions

Markov chains introduced in Sec. 2.1 characterize probability distributions based on the exponential distribution. This phase-type distributions (PHDs) are more complex than the exponential distribution, and can be described by the time until absorption in a CTMC. The roots of PHDs open on to the method of stages first introduced by Erlang where time intervals should be modeled as a random number of exponentially distributed phases [115]. In the following we describe the concept of PHDs, basic definitions, and their classification based on work [47].

Continuous-time Phase type distributions The PHDs belong to matrix analytic probabilistic models where the distribution of a PHD random variable is defined using a matrix $\mathbf{D}_0$ and initial distribution vector $\boldsymbol{\pi}$. The pdf, cdf, moments and variance are also defined in terms of the matrix and initial vector. In this section we concentrate on continuous time PHD and give the basic definitions from the sources [132, 47].

Before we introduce the definition of a PHD, let $\mathscr{S}$ be the state space of the continuous time absorbing Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ with n transient states contained in the transient set $\mathscr{S}_T = \{1,\ldots,n\}$ and one absorbing state $n+1$ contained the set $\mathscr{S}_A = \{n+1\}$.

A phase-type distribution is defined as the distribution of the lifetime X, i.e., the time to enter an absorbing state from the set of transient states $\mathscr{S}_T$ of an absorbing continuous time Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ [47]. The background absorbing CTMC $\{X(t)\}_{t\geq 0}^{\infty}$ has an initial probability vector $\boldsymbol{\pi}$ and the infinitesimal generator $\mathbf{Q}$ given in Eq. 2.16. The intensity matrix $\mathbf{Q}$ contains the matrix $\mathbf{D}_0$ describing transition intensities between transient states, which are also called *phases*. Then a PHD with n transient states is said to have *order n* [47]. The rows of the intensity matrix $\mathbf{Q}$ sum to zero [47]

$$\mathbf{D}_0\mathbf{1} + \mathbf{d}_1 = \mathbf{0}. \tag{2.57}$$

In the following let $\mathbf{1}$ and $\mathbf{0}$ be the vectors of an appropriate dimension. In [47] the following inequalities are givens

$$\mathbf{D}_0(i,i) \leq 0,\ \mathbf{D}_0(i,j) \geq 0 \text{ for } i \neq j,\ \mathbf{d}_1(i) \geq 0 \text{ and } \sum_{j\in\mathscr{S}_T} \mathbf{D}_0(i,j) \leq 0. \tag{2.58}$$

Observe that the underlying absorbing Markov chain has the initial distribution vector $[\pi, \pi(n+1)]$ with $\pi\mathbf{1} + \pi(n+1) = 1$. π is a $1 \times n$ vector and gives the probabilities to start in any transient state. $\pi(n+1)$ is the initial probability for the absorbing state $n+1$ [47]. In the following we assume that $\pi(n+1) = 0$.

In the underlying absorbing Markov process, the sojourn time of each phase i, $1 \leq i \leq n$, is exponentially distributed with parameter $\lambda(i) = -\mathbf{D}_0(i,i)$ as explained in Eq. 2.4. Consequently, the parameters of the involved exponential distributions can be obtained from the diagonal elements of the subintensity $\mathbf{D}_0$ [47]

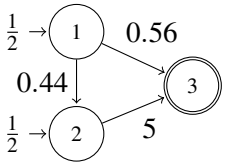
$$\lambda(i) = -\mathbf{D}_0(i,i) = \left(\sum_{j \neq i} \mathbf{D}_0(i,j) + \mathbf{d}_1(i) \right). \tag{2.59}$$

The exit vector $\mathbf{d}_1$ gives the exit rates. The column vector can be determined as [47]

$$\mathbf{d}_1(i) = \lambda(i) - \sum_{j \neq i} \mathbf{D}_0(i,j). \tag{2.60}$$

The random variable X describing the time before absorption is of phase-type with representation $(\pi, \mathbf{D}_0)$, which is a sufficient representation since the exit vector $\mathbf{d}_1$ and $\pi(n+1)$ can be obtained from it [47].

***Example* 2.6.** *Consider a PHD with the subgenerator $\mathbf{D}_0$ given in Figure 2.9(b). For this PHD it holds that all states are entry states, i.e., $\pi(i) \neq 0$ for $i \in \mathscr{S}_T$. Furthermore, it is possible to escape from every transient state i.*



(a) A state transition diagram of a PHD.

$$\mathbf{D}_0 = \begin{bmatrix} -1 & 0.44 \\ 0 & -5 \end{bmatrix}, \mathbf{d}_1 = \begin{bmatrix} 0.56 \\ 5 \end{bmatrix}$$
$$\pi = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

(b) The infinitesimal subgenerator matrix $\mathbf{D}_0$, the exit-rate vector $\mathbf{d}_1$ and the initial probabilities π of the PHD.

Figure 2.9.: Symbolic representation of the PHD of Example 2.6.

Distribution and Moments In this paragraph we give the basic analytic properties of PHDs from [47]. The random variable X is PH distributed with representation $(\pi, \mathbf{D}_0)$. Then the underlying Markov process $\{X(t)\}_{t \geq 0}^{\infty}$ has the intensity matrix $\mathbf{Q}$ given in Eq. 2.16. The distribution function of the random variable X is defined as

$$F(x) = 1 - \pi e^{\mathbf{D}_0 x}\mathbf{1} \text{ for } x \geq 0 \tag{2.61}$$

and its density function is given by

$$f(x) = \pi e^{\mathbf{D}_0 x}\mathbf{d}_1 \text{ for } x \geq 0, \tag{2.62}$$

where the vectors π and $\mathbf{d}_1$ are strictly positive such that the value of the density $\pi e^{\mathbf{D}_0 x}\mathbf{d}_1$ equals or is greater than 0 [115]. Observe that the matrix exponential $e^{\mathbf{D}_0 x}$ in

Eq. 2.61 is defined by the standard series expansion [114]

$$e^{\mathbf{D}_0 x} = \sum_{k\geq 0} \frac{1}{k!}(\mathbf{D}_0 x)^k, \tag{2.63}$$

where it analogously holds that $e^{\mathbf{Q}t} = \sum_{k\geq 0} \frac{1}{k!}(\mathbf{Q}t)^k$.

Let us consider the behavior of the underlying Markov process $\{X(t)\}_{t\geq 0}^{\infty}$. Analyzing Markov chains one is often interested in transient probabilities. Consider the transition probability matrix $\mathbf{P}_t$ defined in [47]. The entry $\mathbf{P}_t(i,j) = Prob(X(t) = j | X(0) = i)$ gives the probability of being in state j at time t, given that the initial phase is i [47]. These probabilities can be obtained using the matrix exponential [47]

$$e^{\mathbf{Q}t} = \begin{bmatrix} e^{\mathbf{D}_0 t} & \mathbf{1} - e^{\mathbf{D}_0 t}\mathbf{1} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.64}$$

as

$$\pi^{(t)} = \pi \mathbf{P}_t = \pi e^{\mathbf{Q}t}, \tag{2.65}$$

where $\pi^{(t)}(i)$ gives the probability that the Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ is in phase i at time t.

In Def. 2.4 we give the definition of fundamental matrix $(-\mathbf{D}_0)^{-1}$. Since the entry $-\mathbf{D}_0^{-1}(i,j)$ gives the expected total time spent in phase j before absorption, given that the initial phase is i, the moments of the PHD can be expressed in terms of the moment matrix $\mathbf{M} = -\mathbf{D}_0^{-1}$. Particularly, the i-th moment of a PHD is defined as [47]

$$\mu_i = E[X^i] = i!\pi\mathbf{M}^i\mathbf{1}. \tag{2.66}$$

A further important property of PHDs is that continuous PHDs are dense in the class of distributions on $\mathbb{R}_{\geq 0}$ (see [47] and references therein). In turn, discrete time PHDs are dense in the class of distributions on $\mathbb{N}$. Furthermore, Erlang distributions with n phases can approximate deterministic distributions as $n \to \infty$. All this makes the PHDs a flexible and versatile stochastic tool, as e.g., any distribution with a strictly positive density in $(0,\infty)$ can be approximated arbitrarily close by a PHD [138].

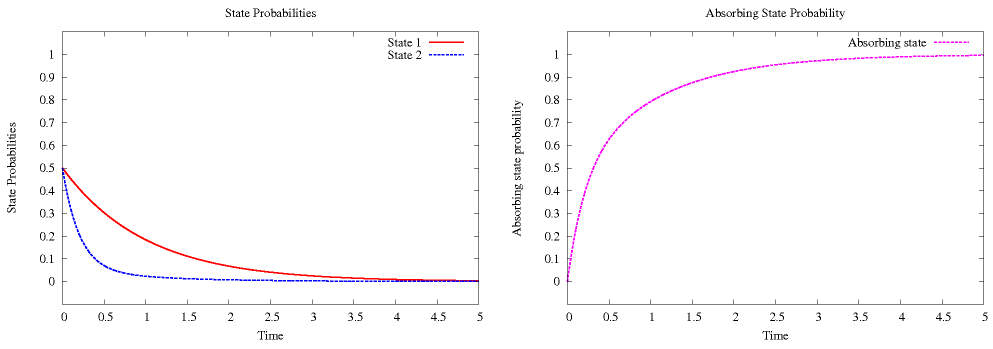


Figure 2.10.: Transient probabilities for states 1, 2, and absorbing state 3.

***Example* 2.7.** *Consider the 2-order PHD given in example 2.6. The transient probabilities are plotted in Fig. 2.10. The pdf and cdf are visualized in Fig. 2.11.*

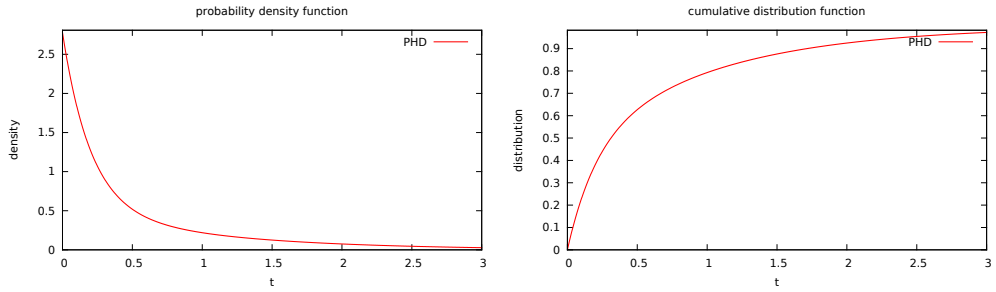


Figure 2.11.: The pdf and cfd of PHD defined in Example 2.6.

2.4.1. Acyclic Phase-Type Distributions

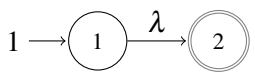
Acyclic PHDs represent the largest subclass of PHDs for which canonical representations exist. We represent the PHDs belonging to this subclass along the lines of [47].

The PHD $(\pi, \mathbf{D}_0)$ can be represented as an acyclic phase-type distribution (APHD), if the transition rate matrix $\mathbf{D}_0$ can be transformed into an upper (or lower) triangular matrix by symmetric permutations of rows and columns. As described in [47], the matrix representation $(\pi, \mathbf{D}_0)$ has $(n^2+n)/2$ parameters for the matrix and $n-1$ parameters for the initial vector.

Note that if the matrix $\mathbf{D}_0$ is of an upper triangular form, phase i can only be connected with phase j if $j > i$. Then no cycles along the paths from initial to the absorbing state can occur. We now give an overview of some relevant PHD subclasses.

Exponential and Erlang Distributions Originally, PHDs were introduced in the method of stages, where random time intervals are modeled as an aggregate of exponentially distributed time intervals. Thus we consider the class of exponential and Erlang distributions first.

The exponential distribution is characterized by its rate parameter λ. The corresponding PHD has only one single state and initial distribution $\pi = [1]$ as shown in Figures 2.12(a), 2.12(b) from [47].



(a) An exponential distribution with parameter λ, and 2 being an absorbing state.

$$\mathbf{Q} = \begin{bmatrix} -\lambda & \lambda \\ 0 & 0 \end{bmatrix} \tag{2.67}$$

(b) The infinitesimal subgenerator $\mathbf{D}_0$.

Figure 2.12.: Markovian representation of the exponential distribution as given in [47].

A. K. Erlang introduced in [69] the representation of distributions as a sum of n exponential phases with the same intensity λ. Let X_i be n mutually independent, exponentially distributed random variables with parameter $\lambda > 0$, $1 \leq i \leq n$. The random variable Y for their sum can be defined as $Y = \sum_{1 \leq i \leq n} X_i$. Then it has an Erlang distri-

bution denoted by $E(n,\lambda)$, and its density is given by [47]

$$f(x) = \frac{\lambda^n}{(n-1)!} x^{n-1} e^{-\lambda x} \text{ for } x \geq 0. \tag{2.68}$$

The distribution function is defined by [47]

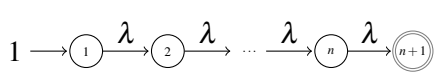
$$F(x) = 1 - \sum_{i=0}^{n-1} \frac{(\lambda x)^i}{i!} e^{-\lambda x} \text{ for } x \geq 0. \tag{2.69}$$

The i-th moment of the Erlang distributed random variable Y is given by [47]

$$E[Y^i] = \frac{(n+i-1)!}{(n-1)!} \frac{1}{\lambda^i}. \tag{2.70}$$

Thus, the mean of Y is $E[Y] = \frac{n}{\lambda}$ and the variance equals $VAR[Y] = \frac{n}{\lambda^2}$ [47].

For the initial distribution vector it holds that $\pi = [1,0,\ldots,0]$. The underlying Markov process is visualized in Figure 2.13 [47].



(a) A graphic representation of the Erlang(n,λ) PHD.

$$\mathbf{D}_0 = \begin{bmatrix} -\lambda & \lambda & \ldots & 0 & 0 \\ 0 & -\lambda & \ldots & 0 & 0 \\ \ldots & \ldots & \ddots & \ldots & \ldots \\ 0 & 0 & \ldots & -\lambda & \lambda \\ 0 & 0 & \ldots & 0 & -\lambda \end{bmatrix} \tag{2.71}$$

(b) The infinitesimal subgenerator $\mathbf{D}_0$.

Figure 2.13.: Erlang representation of a PHD from [47].

The Markov process starts in phase 1 and traverses through the successive states until it reaches the absorbing state $n+1$. Thus, the time to absorption described by Y is the summation of all holding times which are identically exponentially distributed with parameter λ. The Erlang distribution $E(n,\lambda)$ has a squared coefficient of variation of $C^2 = n^{-1}$ which is less than one for $n > 1$. Distributions with a coefficient of variation greater than one can be modeled as finite mixtures of exponential distributions.

The Erlang distribution can be used as an approximation for a deterministic distribution. Particularly, $n = \infty$ phases are required to represent a deterministic distribution. In practice, smaller number of phases can also be used, e.g. a PHD of order 10 is often manageable and can be good enough. For example, deterministic distribution with mean time λ can be approximated by the Erlang(n, $\frac{n}{\lambda}$) PHD. In this case the coefficient of variation is close to zero.

Hypo-exponential and hyperexponential distributions The hypo-exponential distribution is a generalized Erlang distribution. Let $F_i(\cdot)$ be the exponential distribution as described in [47] with

$$F_i(x) = 1 - e^{-\lambda(i)x} \text{ for } x \geq 0,\ 1 \leq i \leq n.$$

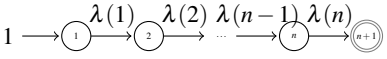
The density function for the rate $\lambda(i)$ is given by [47]

$$f_i(x) = \lambda(i)e^{-\lambda(i)x} \text{ for } x \geq 0.$$

The hypo-exponential distribution is defined by the number of stages n and the set of parameters $\lambda(i)$. Its density function is defined as [47]

$$f(x) = \sum_{i=1}^{n} \left(\prod_{j=1, j\neq i}^{n} \frac{\lambda(j)}{\lambda(j) - \lambda(i)} \right) f_i(x) \text{ for } x \geq 0, \lambda(i) \neq \lambda(j) \text{ for } i \neq j. \tag{2.72}$$

For the initial distribution vector it holds that $\pi = [1, 0, \ldots, 0]$. The graphical representation is given in Fig. 2.14(a) and 2.14(b) from [47].



(a) Markovian representation of the hypo-exponential distribution. In particular $\lambda(1), \ldots, \lambda(n)$ are not necessarily identical.

$$\mathbf{D}_0 = \begin{bmatrix} -\lambda(1) & \lambda(1) & \ldots & 0 & 0 \\ 0 & -\lambda(2) & \ldots & 0 & 0 \\ \ldots & \ldots & \ddots & \ldots & \ldots \\ 0 & 0 & \ldots & -\lambda(n-1) & \lambda(n-1) \\ 0 & 0 & \ldots & 0 & -\lambda(n) \end{bmatrix} \tag{2.73}$$

(b) The infinitesimal subgenerator $\mathbf{D}_0$.

Figure 2.14.: The hypo-exponential distribution as given in [47].

The hyperexponential distribution is defined as a convex mixture of n exponential distributions [47]. Its graphical representation is given in Figure 2.15 from [47]. The density function is defined as [47]

$$f(x) = \sum_{i=1}^{n} \pi(i)\lambda(i)e^{-\lambda(i)x} \text{ for } x \geq 0. \tag{2.74}$$

The distribution function is defined as [47]

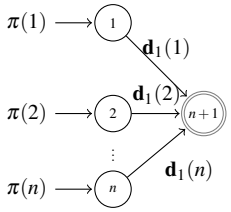
$$F(x) = \sum_{i=1}^{n} \pi(i)(1 - e^{-\lambda(i)x}) \text{ for } x \geq 0. \tag{2.75}$$

The first moment is defined as $E[X] = \sum_{i=1}^{n} \frac{\pi(i)}{\lambda(i)}$ and its variance is given by [47]

$$VAR[X] = 2\sum_{i=1}^{n} \frac{\pi(i)}{\lambda(i)^2} - \left(\sum_{i=1}^{n} \frac{\pi(i)}{\lambda(i)} \right)^2. \tag{2.76}$$

For the squared coefficient of variation it holds [47]

$$C^2 = \frac{E[Y^2]}{(E[Y])^2} - 1 = 2\frac{\sum_{i=1}^{n} \frac{\pi(i)}{\lambda(i)^2}}{(\sum_{i=1}^{n} \frac{\pi(i)}{\lambda(i)})^2} - 1 \tag{2.78}$$

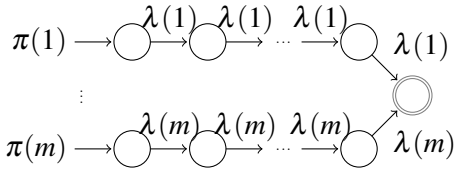


(a) A graphic representation of the hyperexponential distribution.

$$\mathbf{D}_0 = \begin{bmatrix} -\lambda(1) & 0 & \dots & 0 & 0 \\ 0 & -\lambda(2) & \dots & 0 & 0 \\ \dots & \dots & \ddots & \dots & \dots \\ 0 & 0 & \dots & -\lambda(n-1) & 0 \\ 0 & 0 & \dots & 0 & -\lambda(n) \end{bmatrix} \tag{2.77}$$

(b) The infinitesimal subgenerator $\mathbf{D}_0$.

Figure 2.15.: The hyperexponential distribution as given in [47].



(a) A graphic representation of the HErD.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & 0 & \dots & 0 \\ 0 & \mathbf{Q}_2 & \dots & 0 \\ & & \ddots & \\ \dots & \dots & & \dots \\ 0 & 0 & \dots & \mathbf{Q}_m \end{bmatrix} \tag{2.79}$$

(b) The infinitesimal generator $\mathbf{Q}$.

Figure 2.16.: Symbolic representation of the HErD from [47].

Hyper-Erlang distribution A hyper-Erlang distribution denoted as HErD [73], is a mixture of m mutually independent Erlang distributions weighted with the initial probabilities $\pi(1),\dots,\pi(m)$, where $\sum_{i=1}^{m}\pi(i)=1$ [47]. Its graphical representation is given in Figure 2.16 from [47].

2.4.2. Series Canonical Representation

Cumani developed in [56] canonical representations for PHDs which provide the important advantage of having only $2n-1$ free parameters in contrast to APHDs with $(n^2+n)/2$ for the matrix $\mathbf{D}_0$ and $n-1$ parameters for the initial distribution vector (cf. Sect. 2.4.1). To achieve a minimal representation, the APHD should be considered as a stochastic mixture of all possible paths from initial states to the absorbing state. In the following we describe the concepts as given in [56, 47]. We first introduce the concept of *elementary series*.

Definition 2.9. *[47] Let n be the order of the considered APHD. An elementary series of order $m \leq n$ is defined as the following series*

$$ES = < \lambda(i_1)\lambda(i_2)\dots\lambda(i_m) >,$$

where $i_1, i_2, \dots, i_{m-1}, i_m$ is a sequence of states along the acyclic path from an initial state i_1 to the absorbing state $i_m = n+1$.

As described in [47] it holds that $\mathbf{D}_0(i_k, i_{k+1}) \neq 0$ for $k = 1,2,\dots,m$, $\lambda(i_m) = 0$ and the rate between two states i_k, i_{k+1} is given by $\lambda(i_k)$. The maximal number of possible elementary series in a n-order APHD is given by $2^n - 1$ [47].

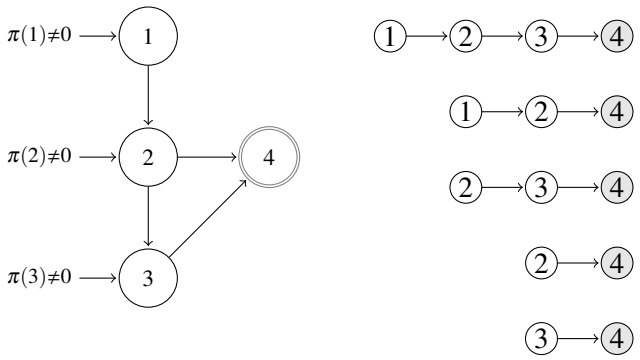


Figure 2.17.: An acyclic 3-phase PHD and its elementary series.

***Example* 2.8.** *Figure 2.17 represents a* 3*-order APHD with its elementary series.*

Let λ and μ be two rates of exponential distribution with $\lambda \leq \mu$. Then the following equality holds [56, 47]

$$\frac{\lambda}{s+\lambda} = \tau\frac{\mu}{s+\mu} + (1-\tau)\frac{\lambda\mu}{(s+\lambda)(s+\mu)}. \tag{2.80}$$

In Eq. 2.80 $\frac{\lambda}{\mu} \in (0,1]$ is the probability for the path until absorption containing only a phase with the transition rate μ, as described in [47]. Then, the probability $(1-\frac{\lambda}{\mu})$ is used to describe the path until absorption containing two phases with intensities λ and μ [47]. Since it holds that $\lambda \leq \mu$, the two successive phases are given in ascending order of the transition rates.

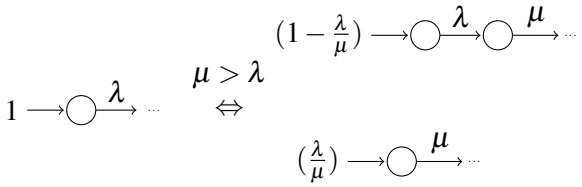


Figure 2.18.: Substitution step for the exponential distribution with rate λ using rate $\mu > \lambda$ as visualized in [47].

Using Eq. 2.80 an elementary series for some phase with transition rate λ can be substituted by a mixture of two elementary series, one containing a phase with transition rate $\mu > \lambda$, and the other containing both phases with the rates λ and μ. This substitution is illustrated in Figure 2.18 from [47]. It is known that each ES has a hypo-exponential representation (2.73). The cdf of an elementary series has the Laplace transform [47]

$$F(s) = \frac{\lambda(i_1)\lambda(i_2)\ldots\lambda(i_{m-1})}{s(s+\lambda(i_1))(s+\lambda(i_2))\ldots(s+\lambda(i_{m-1}))} = \frac{1}{s}\prod_{k=1}^{m-1}\frac{\lambda(i_k)}{(s+\lambda(i_k))}. \tag{2.81}$$

As described in [47] elementary series build the basis of the minimal APHD representation since its cdf can be represented by the mixture of the cdfs of its elementary

series. Then each ES is weighted proportionally to its probability, which is given by the product of transition intensities along the ES path and the initial probability of the first state of the ES [47]. For m states $i_1, i_2, \ldots, i_m$ from the j-th elementary series ES_j of an APHD, the probability of the elementary series is defined by [47]

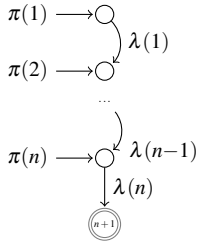
$$\tau_j = \pi(i_1) \frac{\mathbf{D}_0(i_1,i_2)}{-\mathbf{D}_0(i_1,i_1)} \frac{\mathbf{D}_0(i_2,i_3)}{-\mathbf{D}_0(i_2,i_2)} \cdots \frac{\mathbf{D}_0(i_{m-1},i_m)}{-\mathbf{D}_0(i_{m-1},i_{m-1})} \frac{\mathbf{d}_1(i_m)}{-\mathbf{D}_0(i_m,i_m)}. \tag{2.82}$$

Observe from Eq. 2.81 that each of the exponential distributions in the convolution can in principle be reordered [147]. Thus one is interesting in a basic representation, also called a *basic series*, which is in hypo-exponential representation with increasing rates. By repeated use of the substitution in Eq. 2.80 the elementary series can be transformed to a mixture of basic series [47].

Definition 2.10. *[47] Let $0 < \lambda(1) \leq \lambda(2) \leq \ldots \leq \lambda(n)$ be n positive real numbers in ascending order. Then the basic series (BS) is defined as the following series*

$$BS_i = < \lambda(i) \ldots \lambda(n-1)\lambda(n) >,$$

where each tuple of $i, \ldots, n$ transient states determines the acyclic path till absorption.



(a) The series canonical form, in particular $0 < \lambda(1) \leq \lambda(2) \leq \ldots \leq \lambda(n-1) \leq \lambda(n)$.

$$\mathbf{D}_0 = \begin{bmatrix} -\lambda(1) & \lambda(1) & 0 & \ldots & 0 \\ 0 & -\lambda(2) & \lambda(2) & & \\ & & \ddots & \ddots & \vdots \\ \vdots & & & -\lambda(n-1) & \lambda(n-1) \\ 0 & & & & -\lambda(n) \end{bmatrix}$$

(b) The infinitesimal subgenerator $\mathbf{D}_0$.

Figure 2.19.: PHD in series canonical form as defined in [47].

Definition 2.11. *[47] The subgenerator in Figure 2.19 from [47] defines a series canonical form. A mixture of basic series of an APHD with transition intensities in ascending order, i.e. $\lambda(i) \leq \lambda(i+1) \leq \ldots \leq \lambda(n)$, defines an APHD.*

The series canonical form is one of the minimal representations. It has $2n-1$ degrees of freedom: n transition rates and $n-1$ initial probabilities.

In the series canonical form the rates are ordered ascendingly such that transitions are only possible from phase i to the successor phase $i+1$. Furthermore the process can escape only from the last phase n. Thus we need only n transition parameters for the matrix $\mathbf{D}_0$. All transient states may be entry states satisfying $\pi(i) \geq 0$, for all $i = 1, \ldots, n$. Since the absorbing state cannot be entered initially, we need only $n-1$ initial probabilities.

***Example* 2.9.** *Consider the following APHD*

$$\mathbf{D}_0 = \begin{bmatrix} -1 & 0.8 \\ 0 & -1.5 \end{bmatrix}, \mathbf{d}_1 = \begin{bmatrix} 0.2 \\ 1.5 \end{bmatrix}, \pi = \begin{bmatrix} 0.6 & 0.4 \end{bmatrix}.$$

To obtain the series canonical form the transition from phase 1 *to the absorbing state should be eliminated. In particular, the* $ES_1 =< 0.8\ 1.5 >$ *occurs with probability* $0.6\frac{0.8}{1} = 0.48$*, the* $ES_2 =< 0.2 >$ *occurs with probability* $0.6\frac{0.2}{1} = 0.12$*, and the* $ES_3 =< 1.5 >$ *occurs with probability* $0.4\frac{1.5.}{1.5} = 0.4$*. If we apply the substituting step to the* ES_2 *with* $\lambda = 1$ *and* $\mu = 1.5$*, we obtain two series, namely* $< 1.5 >$ *with probability* $\frac{\lambda}{\mu} = \frac{1}{1.5} = 0.66$*, and the series* $< 1\ 1.5 >$ *with complementary probability* $1 - \frac{\lambda}{\mu} = 0.33$*. Since the original series* ES_2 *occurs with probability* 0.12*, the probability of* $< 1.5 >$ *is given by* $0.12 \cdot 0.66 = 0.08$*, and the probability of* $< 1\ 1.5 >$ *is given by* $0.12 \cdot 0.33 = 0.04$*.*

Now observe that all basic series are determined and the resulting distribution in series canonical form is given below.

$$\mathbf{D}_0^{can} = \begin{bmatrix} -1 & 1 \\ 0 & -1.5 \end{bmatrix}, \mathbf{d}_1^{can} = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \pi^{can} = \begin{bmatrix} 0.52 & 0.48 \end{bmatrix}.$$

The probability of the $ES_1 =< 1\ 1.5 >$ *is completed to* $0.48 + 0.04 = 0.52$*, and the probability of* $ES_2 =< 1.5 >$ *is given by* $0.4 + 0.08 = 0.48$*.*

2.4.3. Bilateral Phase-type Distributions

The extension of PHD to *bilateral* phase type distribution (BPHD) on the entire line $(-\infty, \infty)$ was first introduced in the work of Ahn and Ramaswami [4]. In the earlier work of Shanthikumar [158] a class of bilateral PHDs was defined where positive and negative parts of a BPHD random variable can be represented as a mixture of sums of iid exponentially distributed random variables. These mixtures of sums could also be infinite since infinite state space Markov chains were incorporated. It has been shown that the class of BPHDs is closed under convolution and mixtures involving consideration of the infinite Markov chains. However, in [4] only finite state space Markov chains were considered. In the following we describe the concepts from [158, 4, 95].

Assume that a partitioning of n transient states is given by $\mathscr{S}_T = \mathscr{S}_1 \cup \mathscr{S}_2$. Now we can introduce the Markov modulated reward process $B = \{B(t) : t \geq 0\}$. During the sojourn time of the underlying Markov process in the state $i \in \mathscr{S}_1$ the accumulated reward increases with rate $c_i > 0$. Analogously, during the sojourn time of the Markov process in the state $j \in \mathscr{S}_2$ the accumulated reward decreases with rate $c_j > 0$. The reward function $B(t)$ describes the total accumulated reward of the Markov process $\{X(t)\}_{t \geq 0}^{\infty}$ in the time interval $(0,t)$, i.e. over the finite horizon.

Consider that $B(0) = 0$ and once the absorbing state is reached no further changes occur to the reward $B(X)$, where X is the absorbing time of the underlying Markov process. We first introduce the *diagonal reward matrices*

$$\mathbf{C}_1 = diag(c_i, i \in \mathscr{S}_1), \quad \mathbf{C}_2 = diag(c_j, j \in \mathscr{S}_2), \quad \mathbf{C}^* = diag(\mathbf{C}_1, -\mathbf{C}_2). \tag{2.83}$$

Note that $diag(c_i, i \in \mathscr{S})$ represents the matrix with elements $c_i, i \in \mathscr{S}$ on the diagonal. The $diag(A_1, A_2)$ represents the matrix with the matrix A_1 on the diagonal following by the matrix A_2 on the diagonal.

Definition 2.12. *The total accumulated reward till absorption $Y = B(X)$ is a bilateral phase-type distributed random variable with representation $(\pi, \mathbf{D}_0, \mathbf{C}^*)$.*

We also represent the result from [4].

Theorem 2.2 (Theorem 3 in [4])**.** *Let Y be a bilateral phase-type distributed random variable. Then we obtain two random variables $Y^+ = \max(0, Y)$ and $Y^- = -\min(0, Y)$ which are both phase-type distributed.*

Note that Y^+, Y^- have both realizations in $\mathbb{R}^+$. The general formula for the k-th moment of Y is given by

$$E[Y^k] = k!\pi(\mathbf{M}\mathbf{C}^*)^k\mathbf{1}, \tag{2.84}$$

where $\mathbf{M} = -\mathbf{D}_0^{-1}$ is given in Def. 2.4.

Properties BPHDs inherit several important properties of PHDs first mentioned in [4]. BPHDs are closed under convolutions and mixtures, the proof of these properties can be adapted from proofs presented in [115]. Furthermore, the BPH class is closed under the minimum and the maximum operation, as mentioned in [95]. If we consider the case where the partition $\mathscr{S}_2$ is empty and all reward rates $c_i = 1$ for the transient states from $\mathscr{S}_1$, one can verify that PHDs on $[0, \infty)$ represent a subset of BPHDs.

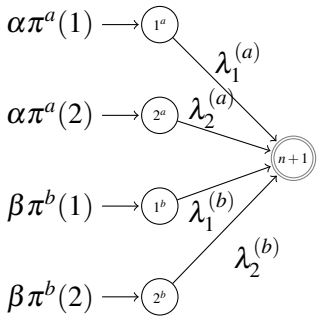
Similarly to PHDs, the matrix representation is not unique. Let $\mathbf{V}$ be a non-singular diagonal matrix with positive elements, then the tuple $(\pi, \mathbf{V}\mathbf{D}_0, \mathbf{V}\mathbf{C}^*)$ defines the same BPHD [95]. This similarity property allows the definition of all reward rates as $c_i = 1$ for all transient phases $i \in \mathscr{S}_T$ such that $\mathbf{C}^* = \mathbf{I}^*$, where $\mathbf{I}^*$ is a diagonal matrix containing $\mathbf{I}^*(i,i) = 1$ for state $i \in \mathscr{S}_1$ and $\mathbf{I}^*(i,i) = -1$ for state $i \in \mathscr{S}_2$ as given in [4].

As mentioned above, one of the most interesting properties of BPHDs is that they can be represented as a mixture of a positive and negative PHDs. For this purpose consider two PHDs $(\pi^{(a)}, \mathbf{D}_0^{(a)})$, $(\pi^{(b)}, \mathbf{D}_0^{(b)})$ and two constants $0 \leq \alpha, \beta \leq 1$ satisfying $\alpha + \beta \leq 1$. Then the positive part of a BPHD can be modeled by the PHD given by $(\pi^{(a)}, \mathbf{D}_0^{(a)})$.

Correspondigly, the negative part is modeled by the PHD represented as $(\pi^{(b)}, \mathbf{D}_0^{(b)})$. The initial probabilities $\pi^{(a)}$, $\pi^{(b)}$ can be weighted with constants α and β such that $P(Y > 0) = \alpha$, $P(Y < 0) = \beta$ and an atom at 0 obtains the probability $P(Y = 0) = 1 - \alpha - \beta$. The given PHDs $(\pi^{(a)}, \mathbf{D}_0^{(a)})$, $(\pi^{(b)}, \mathbf{D}_0^{(b)})$ are then conditional distributions of $|Y|$, given that $Y > 0$, and that $Y < 0$ respectively.

The constructed BPHD Y has the representation

$$([\alpha\pi^{(a)}, \beta\pi^{(b)}],\ diag(\mathbf{D}_0^{(a)}, \mathbf{D}_0^{(b)}),\ diag(\mathbf{I}, -\mathbf{I})).$$



(a) A graphic representation of the BPH distribution.

$$\mathbf{D}_0 = \begin{bmatrix} -\lambda_1^{(a)} & 0 & 0 & 0 \\ 0 & -\lambda_2^{(a)} & 0 & 0 \\ 0 & 0 & -\lambda_1^{(b)} & 0 \\ 0 & 0 & 0 & -\lambda_2^{(b)} \end{bmatrix} \tag{2.85}$$

(b) The infinitesimal subgenerator D_0

Figure 2.20.: The bilateral phase-type distribution constructed as the mixture of two PHDs in hyperexponential representation based on [95].

Fig. 2.20 contains the graphic representation of the construction for PHDs $(\pi^{(a)}, \mathbf{D}_0^{(a)})$, $(\pi^{(b)}, \mathbf{D}_0^{(b)})$ in hyperexponential representation which are both of order 2. Then the k-th moment of the BPH distributed random variable Y can be also represented by

$$E[Y^k] = k!\,\alpha\,\pi^{(a)}(\mathbf{M}^{(a)})^k\mathbf{1} + (-1)^k k!\,\beta\,\pi^{(b)}(\mathbf{M}^{(b)})^k\mathbf{1}, \tag{2.86}$$

such that the BPHD is given by the convex mixture of two PHDs.

***Example* 2.10.** *We present an example from [4] to demonstrate capabilities of BPHDs that can be used modeling interesting characteristics. Consider the acyclic BPHD*

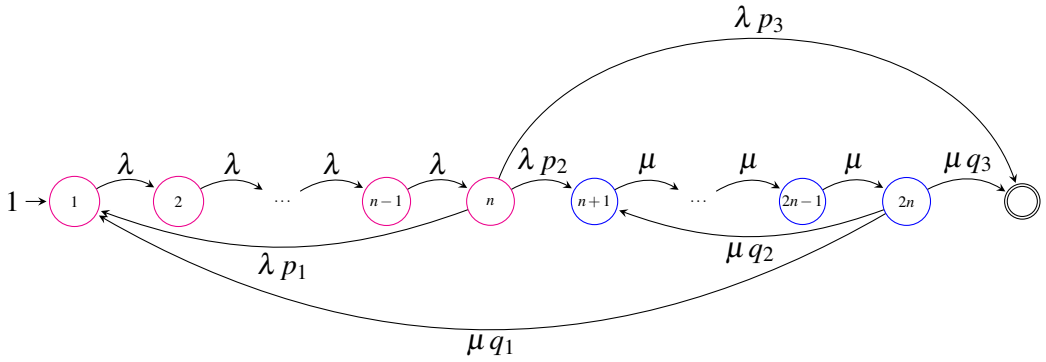


Figure 2.21.: The BPHD composed as the convolution of two Erlang distributions both of order n (see [4]).

of order $2n$ visualized in Figure 2.21. It holds that $\mathscr{S}_1 = \{1,\ldots,n\}$ and $\mathscr{S}_2 = \{n+1,\ldots,2n\}$. The first n transient phases have mean sojourn times $1/\lambda$ and reward rates 1, whereas the remaining n phases have mean sojourn times $1/\mu$ and reward rates -1.

The process starts in the phase $1 \in \mathscr{S}_1$ and goes first through the sequence of phases from the set $\mathscr{S}_1$. With probability p_2 the process traverses the path containing remaining phases from the set $\mathscr{S}_2$. At the end of this path the process gets absorbed with

probability q_3. In that case the BPH random variable is realized as the difference of two independent Erlang random variables X_1 and X_2, where $X_1 \sim Erlang(n,\lambda)$ and $X_2 \sim Erlang(n,\mu)$.

After a start in phase 1 and going through the sequence of states $1,\ldots,n$, the process may also go back to the initial phase 1 with probability p_1, which would lead to the repeated visit of all states from the set $\mathscr{S}_1$. The other possibility is to jump from the state n to the absorbing state with probability p_3.

Analogously, once the sequence $n+1,\ldots,2n$ is passed through, the process may jump to the initial state 1 with probability q_1 or goes back to the state $n+1$ with probability q_2, which leads to the repeated traversing of the phases from the set $\mathscr{S}_2$.

In the case where the probabilities $p_3 = 0$ and $q_1 = 0$ the sequence of phases $n+1,\ldots,2n$ from the set $\mathscr{S}_2$ is traversed at least once and the process cannot jump back to initial state 1, if the whole sequence of states has been traversed. In the case that $p_2 = 1$, $q_3 = 1$, and $\lambda = \mu$, the BPHD has mean zero. Observe that the return probabilities p_1, q_2 can be used to model the positive and negative parts as a geometric mixture of successive convolutions of the corresponding Erlang distribution.

Fitting methods for BPHDs A possible fitting approach for BPHDs considers positive and negative values from the trace separately and then determines parameters for both PHDs according to these parts [95]. In the last step established fitting methods can be applied to obtain a PH distribution. Then the construction in Fig. 2.20 can be used to obtain a BPHD for the whole sample [4]. However, this approach has a disadvantage that the fitting of BPHDs with continuity behavior at zero will in general result in small likelihood at zero. The approach developed in [95] presents moments bounds and the moment matching algorithm for ABPHDs dealing with pdf functions having equal left and right limit at zero.

2.5. Markovian Arrival Processes

In this section we consider *Markovian Arrival Processes* (MAPs) which belong to powerful stochastic models enabling models with correlated inter-event times. MAPs were first introduced by Neuts [132] and can be considered as a generalization of PHDs. They were originally used as input processes for queueing systems enabling analytical solutions [121]. In this thesis MAPs are used in some simulation models. Application areas of MAPs range from queueing, communications, performance analysis, reliability and finance to name a few. However, there is a huge amount of work about MAPs and their applications. We refer to [139, 53, 129, 14, 54] for further studying of MAPs and their applications.

The flexibility and versatility of MAPs are based on their main properties; first, the inter-arrival times are PH distributed, which allows an application of established fitting methods to the real world data. Secondly, the correlation between inter-arrival times can be modeled, which is present in many real data traces. In the following we describe the concept of MAPs and give an overview based on [47].

Basic definitions A MAP is a Markov process which can be described by an irreducible Markov chain where some transitions correspond to the occurrence of an *event*

also called as *arrival*.

Definition 2.13. *[47] Formally, a MAP with representation* $(\pi, \mathbf{D}_0, \mathbf{D}_1)$ *is a Markov chain with a finite state space* $\mathscr{S}$*, initial vector* π *and irreducible infinitesimal generator matrix* $\mathbf{Q}$ *with*

$$\mathbf{Q} = \mathbf{D}_0 + \mathbf{D}_1,\ \mathbf{D}_1 \geq \mathbf{0},\ \mathbf{D}_1 \neq \mathbf{0},\ \mathbf{D}_0(i,j) \geq 0 \text{ for } i \neq j.$$

Similarly to PHDs, the size of the state space $n = |\mathscr{S}|$ defines the order of the MAP. The constraint on the row sums is given by [47]

$$\mathbf{D}_0 \mathbb{1} = -\mathbf{D}_1 \mathbb{1}. \tag{2.87}$$

The stochastic behavior of a MAP is as follows: The MAP starts in state i with probability $\pi(i)$, stays in the state i an exponentially distributed time with rate $\lambda(i)$ which can be derived from Eq. 2.87 as $-\mathbf{D}_0(i,i) = \sum_{j \neq i} \mathbf{D}_0(i,j) + \sum_j \mathbf{D}_1(i,j)$. Afterwards either the transition from $\mathbf{D}_0$ or from $\mathbf{D}_1$ occurs. Transitions from the infinitesimal subgenerator $\mathbf{D}_0$ are not associated with an event. The MAP only goes to the phase j with probability $\mathbf{D}_0(i,j)/\lambda(i)$ [112, 47].

If the transition from the matrix $\mathbf{D}_1$ occurs, a MAP generates an event. The whole probability of an event is given by $\sum_j \mathbf{D}_1(i,j)/\lambda(i)$. The process then goes to the state j with probability $\mathbf{D}_1(i,j)/\lambda(i)$ for $i \neq j$. With probability $\mathbf{D}_1(i,i)/\lambda(i)$ the successor state is again i [112, 47].

Observe that the matrix $\mathbf{D}_0 + \mathbf{D}_1$ is an irreducible infinitesimal generator matrix of the underlying Markov process. The stationary distribution vector π_c is the solution of $\pi_c \mathbf{Q} = \mathbf{0}$ with $\pi_c \mathbb{1} = 1$.

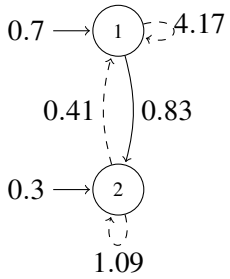
The event occurrence form an embedded DTMC with transition probability matrix $\mathbf{P}_s = (-\mathbf{D}_0)^{-1}\mathbf{D}_1$. The matrix $\mathbf{P}_s$ has the unique left eigenvector $\pi_s \mathbf{P}_s = \pi_s$ with $\pi_s \mathbb{1} = 1$ which describes the distribution instantly after an event occurrence. Thus π_s can be implicitly used as initial vector of the MAP such that its representation can be reduced to the tuple $(\mathbf{D}_0, \mathbf{D}_1)$ and the following relation holds [96, 38]

$$\pi_c = \frac{\pi_s (-\mathbf{D}_0)^{-1}}{\pi_s (-\mathbf{D}_0)^{-1} \mathbb{1}}, \qquad \pi_s = \frac{\pi_c \mathbf{D}_1}{\pi_c \mathbf{D}_1 \mathbb{1}}. \tag{2.88}$$

The vector π_s has a further important interpretation, namely the stationary inter-arrival times of a MAP are PH distributed with initial distribution vector π_s and infinitesimal subgenerator matrix $\mathbf{D}_0$. Note that $(\pi, \mathbf{D}_0)$ is also a PHD, which describes the inter-arrival time in the first event epoch. In that case the distribution of the initial phase of a MAP differs from the stationary inter-event distribution.

Remark that each PHD with representation $(\pi, \mathbf{D}_0)$ can be described by a MAP. The MAP $(\pi, \mathbf{D}_0, \mathbf{d}_1 \pi)$ has the same behavior as the mentioned PHD. In fact, $\mathbf{d}_1 = -\mathbf{D}_0 \mathbb{1}$ describes uncorrelated transition rates, it contains no relation between exit and entry phases of the MAP. Thus the inter-event times of the MAP are independently and identically distributed PHDs $(\pi, \mathbf{D}_0)$.

***Example* 2.11.** *Figure 2.22 shows a* 2*-state MAP. The stochastic behavior of the MAP is as follows. If it starts in state* 1*, it resides there for an exponentially distributed time with mean duration of* 0.2*. When the exponential distribution elapses, the transition to*



(a) The state transition diagram for the MAP.

$$\mathbf{D}_0 = \begin{bmatrix} -5 & 0.83 \\ 0 & -1.5 \end{bmatrix} \quad \mathbf{D}_1 = \begin{bmatrix} 4.17 & 0 \\ 0.41 & 1.09 \end{bmatrix}$$

(b) The matrices for the MAP.

Figure 2.22.: A 2-state MAP. The dashed transition arrows correspond to state transitions generating an event. The solid line transition arrows correspond to transitions according to $\mathbf{D}_0$.

state 2 *without generating an event occurs with probability* $\frac{0.83}{5} = 0.166$. *An event is generated with rate* 4.17*, i.e., the process stays in state* 1 *with probability* $\frac{4.17}{5} = 0.834$. *In state* 2 *the MAP generates events with an exponentially distributed inter-event time with mean duration of* 0.66. *After generating an event the MAP stays in state* 2 *with probability* $\frac{0.41}{1.5} = 0.273$ *and performs transition to state* 1 *with probability* $\frac{1.09}{1.5} = 0.726$.

Distribution and Moments Since the inter-event times of a MAP are dependent we introduce measures that describe dependencies. Let $X_1, \ldots, X_k$ be a sequence of arbitrary k consecutive inter-event times and X an arbitrary inter-event time. In the following we assume that MAP has representation $(\mathbf{D}_0, \mathbf{D}_1)$ and $\pi = \pi_s$. The probability density function and the distribution function of X are given by [112, 47]

$$f(x) = \pi e^{\mathbf{D}_0 x} \mathbf{D}_1 \mathbb{1}, \qquad F(x) = 1 - \pi e^{\mathbf{D}_0 x} \mathbb{1}. \tag{2.89}$$

The i-th moment μ_i of the inter-event time X can be computed as given in Eq. 2.66.

The joint density of a MAP generating k consecutive events with inter-event times x_i is given by [112, 47]

$$f(x_1, x_2, \ldots, x_k) = \pi e^{\mathbf{D}_0 x_1} \mathbf{D}_1 e^{\mathbf{D}_0 x_2} \mathbf{D}_1 \ldots e^{\mathbf{D}_0 x_k} \mathbf{D}_1 \mathbb{1}, \tag{2.90}$$

which can be evaluated using the uniformization method [162]. For this, let $\alpha \geq \max_i(|\mathbf{D}_0(i,i)|)$, $\mathbf{P}_0 = \mathbf{D}_0/\alpha + \mathbf{I}$ and $\mathbf{P}_1 = \mathbf{D}_1/\alpha$. Then Eq. 2.90 can be written as [47]

$$f(x_1, x_2, \ldots, x_k) = \pi \left(\prod_{i=1}^{k} \left(\sum_{l=0}^{\infty} \beta(\alpha x_i, l) \mathbf{P}_0^l \right) \mathbf{P}_1 \right) \mathbb{1} \tag{2.91}$$

where $\beta(q,l)$ is the probability of l events of a Poisson process with parameter q.

The joint moments of k consecutive inter-event times are [47]

$$E[X_1^{i_1}, X_2^{i_2}, \ldots, X_k^{i_k}] = i_1! i_2! \ldots i_k! \pi (-\mathbf{D}_0)^{-i_1} \mathbf{P}_s (-\mathbf{D}_0)^{-i_2} \ldots \mathbf{P}_s (-\mathbf{D}_0)^{-i_k} \mathbb{1}, \tag{2.92}$$

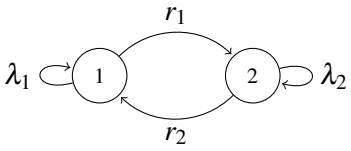
with orders i_l, $1 \leq l \leq k$.

Dependency between inter-event times that are lag-k apart is often expressed in terms of the coefficient of autocorrelation. It can be determined if dependency between the first and the k-th inter-event time should be computed [47]

$$\rho_k = \frac{E[X_1, X_{1+k}] - (E[X])^2}{E[X^2] - (E[X])^2} = \frac{\pi(-\mathbf{D}_0)^{-1}\mathbf{P}_s^k(-\mathbf{D}_0)^{-1}\mathbb{1} - \left(\pi(-\mathbf{D}_0)^{-1}\mathbb{1}\right)^2}{2\pi(-\mathbf{D}_0)^{-2}\mathbb{1} - \left(\pi(-\mathbf{D}_0)^{-1}\mathbb{1}\right)^2}. \quad (2.93)$$

Subclasses Since MAPs represent a powerful class of stochastic processes, they contain several well-known stochastic processes as subclasses. An important subclass allowing for modeling correlations between inter-event times while still remaining analytically tractable is a *Markov Modulated Poisson Process* (MMPP) [75]. The matrix $\mathbf{D}_1$ is constructed from rates of n Poisson processes with event rates λ_i. Thus $\mathbf{D}_1$ is a diagonal event rate matrix with λ_i values on the diagonal. Additionally, an auxiliary Markov process of order n selects one of the Poisson processes. In particular, if the Markov process is in state i, the events occur according to a Poisson process with rate λ_i. The MMPP can be described by a MAP.

***Example* 2.12.** *Fig. 2.23 shows the 2-state MMPP.*



(a) A state transition diagram of a MMPP.

$$\mathbf{D}_0 = \begin{bmatrix} -(r_1+\lambda_1) & r_1 \\ r_2 & -(r_2+\lambda_2) \end{bmatrix},$$
$$\mathbf{D}_1 = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

(b) The infinitesimal subgenerator matrix $\mathbf{D}_0$ and the event rate matrix $\mathbf{D}_1$ of the MMPP.

Figure 2.23.: Symbolic representation of the 2-state MMPP.

One can easily obtain the probabilities $Prob(\lambda = \lambda_1) = \frac{r_2}{r_1+r_2}$ *and* $Prob(\lambda = \lambda_2) = \frac{r_1}{r_1+r_2}$. *Then the mean arrival rate of the 2-state MMPP is defined as*

$$q = \sum_{k=1}^{2} \lambda_k Prob(\lambda = \lambda_k) = \frac{\lambda_1 r_2 + \lambda_2 r_1}{r_1 + r_2}. \quad (2.94)$$

An *Interrupted Poisson Process* (IPP) is a special case of MMPP [113]. The IPP is a Poisson process with the rate λ which is regulated by an auxiliary on-off Markov process. When the Markov process is in the *on*-state, events occur according to the basic Poisson process. Thus the diagonal elements of $\mathbf{D}_1$ are either given by λ or equals 0. On and off time intervals of the Poisson process are PH distributed.

Multiclass Markovian Arrival Processes (MMAPs) describe marked events and were originally introduced in [88]. It has the representation $(\pi, \mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_k)$ with $(\pi, \mathbf{D}_0, \sum_{k=1}^{K} \mathbf{D}_k)$ describing a MAP where matrices $\mathbf{D}_k$ define different event types. For further studying we refer to [9, 43, 86]. *Batch Markovian Arrival Processes* (BMAPs) differ in the interpretation of the event types [120, 87, 154]. Here matrices $\mathbf{D}_k$ contain transition rates accompanied by events occuring in the batch of size k. On the

other hand further MAP generalizations exist. *Rational Arrival Processes* (RAPs) also known as *Matrix Exponential Processes* (MEPs) [8, 51, 32, 50] describe stochastic processes with matrix exponential inter-event time. An additional freedom is added to this class since the involved matrices and vectors are not restricted to have an interpretation of stochastic behavior for the underlying process.

Chapter 3

PH-Graph Model

In this chapter we introduce a stochastic graph model with Phase-type distributed edge weights (PH-Graph). The kernel of the model is a composition of two PH distributed random variables. Additionally the information about the existing correlation between them is decoded. This property permits modeling of dependent edge weights on adjacent edges in a stochastic graph.

In the following we consider the composition of PH distributions which contains a pair of PHDs and a matrix encoding the correlation between them. Then the PH-Graph model containing a PHD composition for each adjacent edge pair is introduced. The PH-Graph model encodes the existing correlations for the whole graph where edge weights are described by PHDs.

The developed model leads to efficient solutions for issues in the context of *Stochastic Shortest Path Problems with correlations*. In particular, it enables the computation of the minimum/maximum expected total time for reaching an absorbing state, and the computation of the maximum/minimum probability of reaching an absorbing state within a given deadline, both in stochastic graphs including correlations. More computational issues can be handled by the model, e.g. the modeling of PH distributed interclaim times and claim sizes in insurance risk models.

3.1. The Composition of PH Distributions

Let PHD $PH_i = (\boldsymbol{\pi}_i, \mathbf{D}_i)$ be of order n_i, and PHD $PH_j = (\boldsymbol{\pi}_j, \mathbf{D}_j)$ be of order n_j where matrices $\mathbf{D}_i$, $\mathbf{D}_j$ are subgenerator matrices $\mathbf{D}_0$ of corresponding PHDs. PH_i describes the distribution of a random variable X_i, and PH_j is the distribution of a random variable X_j. Furthermore the random variables X_i, X_j are independent. Note that the sum $X = X_i + X_j$ is known to be a PHD [47]. When the dependency between two PHDs PH_i, PH_j is introduced the sum $X = X_i + X_j$ is also a PHD and is called a *composition*.

Definition 3.1. *The tuple* $((\boldsymbol{\pi}_i, \mathbf{D}_i), (\boldsymbol{\pi}_j, \mathbf{D}_j), \mathbf{H}_{ij})$ *defines a composition of two PHDs* PH_i *and* PH_j*. The infinitesimal subgenerator given in Eq. 3.1 describes the underlying Markov process* $\{X(t)\}_{t\geq 0}^{\infty}$ *for the composition of two dependent PHDs. The initial*

probability vector is given by $\pi = [\pi_i, \mathbf{0}]$ *where* $\mathbf{0}$ *is the row* n_j*-vector of* 0*'s.*

$$\mathbf{D}_0 = \begin{bmatrix} \mathbf{D}_i & \mathbf{H}_{ij} \\ 0 & \mathbf{D}_j \end{bmatrix}. \tag{3.1}$$

The $n_i \times n_j$ *matrix* $\mathbf{H}_{ij}$ *is called a* transfer matrix *with*

$$\mathbf{H}_{ij} \geq \mathbf{0},\ \mathbf{H}_{ij} \neq \mathbf{0},\ \mathbf{H}_{ij}\mathbb{1} = -\mathbf{D}_i\mathbb{1}. \tag{3.2}$$

Furthermore the following equation has to hold to keep the initial distribution of PH_j *invariant*

$$\pi_i(-\mathbf{D}_i)^{-1}\mathbf{H}_{ij} = \pi_j. \tag{3.3}$$

The PHD composition $((\pi_i, \mathbf{D}_i),\ (\pi_j, \mathbf{D}_j),\ \mathbf{H}_{ij})$ *is visualized in Fig. 3.1.*

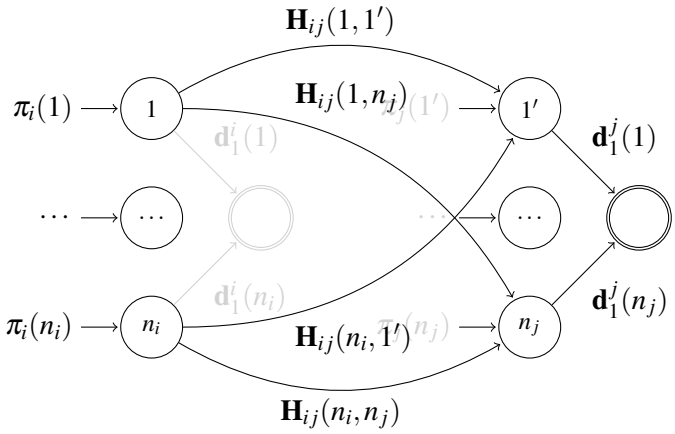


Figure 3.1.: Symbolic representation of the Markov chain corresponding to a PHD composition. Parameters implicitly given by the matrix $\mathbf{H}_{ij}$ are highlighted in gray.

The resulting number of transient phases in the underlying Markov process of the composition is $n_i + n_j$. The stochastic behavior of a composition process is as follows: In the underlying Markov chain $\{X(t)\}_{t\geq 0}^{\infty}$ the paths of Markov chains associated with PH_i and PH_j are concatenated. After the paths of the Markov chain $\{X_i(t)\}_{t\geq 0}^{\infty}$ have been traversed the process moves along the paths of the subsequent Markov chain $\{X_j(t)\}_{t\geq 0}^{\infty}$.

The composition process starts in state k with probability $\pi_i(k)$. Then the process moves along paths of the first Markov chain $\{X_i(t)\}_{t\geq 0}^{\infty}$ until some initial state of the Markov chain $\{X_j(t)\}_{t\geq 0}^{\infty}$ is reached.

When some initial state k of the first Markov chain has been entered, the process stays an exponentially distributed time in that state. Afterwards either the transition from $\mathbf{D}_i$ or from $\mathbf{H}_{ij}$ occurs. If the transition from transfer matrix $\mathbf{H}_{ij}$ occurs, the process goes to some entry state of the Markov chain $\{X_j(t)\}_{t\geq 0}^{\infty}$. In fact, transfer matrix $\mathbf{H}_{ij}$ works analogously as $\mathbf{D}_1$ matrix from the MAP theory. It describes transitions corresponding to the occurrence of an event in the first Markov process.

Similarly to a MAP (cf. 2.5) the absorption of the first Markov chain $\{X_i(t)\}_{t\geq 0}^{\infty}$ induces an embedded Markov process with transition probability matrix defined by

$$\mathbf{P}_{ij} = \mathbf{M}_i \mathbf{H}_{ij} \tag{3.4}$$

where $\mathbf{M}_i = (-\mathbf{D}_i)^{-1}$. It holds that

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

The matrix entry $\mathbf{P}_{ij}(k,l)$ gives the probability of starting in state l of the subsequent Markov chain, if the composition process begins at state k of the first Markov chain.

Since the transfer matrix $\mathbf{H}_{ij}$ satisfies the Eq. 3.3, it holds that $\pi_i \mathbf{P}_{ij} = \pi_j$ such that π_j describes the distribution instantaneously after the absorption of the Markov chain $\{X_i(t)\}_{t\geq 0}^{\infty}$ has occurred. If the composition process starts with probability distribution π_i, then it continues in the average with probability distribution π_j in entry state of the $\{X_j(t)\}_{t\geq 0}^{\infty}$. Thus the distribution function, i.e. initial vector π_j of the subsequent Markov process $\{X_j(t)\}_{t\geq 0}^{\infty}$ remains invariant.

In fact, the transfer matrix $\mathbf{H}_{ij}$ of a composition describes correlated transition rates. In particular, the entry $\mathbf{H}_{ij}(k,l)$ contains a transition rate if k is the exit phase of the Markov process $\{X_i(t)\}_{t\geq 0}^{\infty}$ and l is the entry phase of the Markov process $\{X_j(t)\}_{t\geq 0}^{\infty}$. Thus the transfer matrix $\mathbf{H}_{ij}$ contains the relations between exit and entry phases of the two PHDs in the composition. Observe that if $\mathbf{H}_{ij} = \mathbf{d}_1^i \, \pi_j$ the PHDs in the composition are uncorrelated.

Since two PHDs PH_i, PH_j in the composition are no longer independent, the dependency measures can be derived, i.e. adapted from the MAP theory. Let (X_i, X_j) be two consecutive absorbing times. The joint density of a composition generating two consecutive absorbing times x_i, x_j is given by

$$Prob(X_i = x_i, X_j = x_j) = f(x_i, x_j) = \pi_i \, e^{x_i \mathbf{D}_i} \, \mathbf{H}_{ij} \, e^{x_j \mathbf{D}_j} \, (-\mathbf{D}_j)\mathbb{1}. \tag{3.5}$$

The joint moments of two consecutive absorption times X_i, X_j which are correlated according to the transfer matrix $\mathbf{H}_{ij}$ are defined as

$$E[X_i^k, X_j^l] = k!\, l!\, \pi_i \mathbf{M}_i^k \mathbf{P}_{ij} \mathbf{M}_j^l \mathbb{1} \tag{3.6}$$

with orders k, l, $1 \leq k \leq l$, where $\mathbf{M}_i = (-\mathbf{D}_i)^{-1}$ and $\mathbf{M}_j = (-\mathbf{D}_j)^{-1}$.

The covariance of two consecutive absorption times X_i, X_j is given by

$$Cov[X_i, X_j] = E[X_i^1, X_j^1] - E[X_i^1] \cdot E[X_j^1] = \pi_i \mathbf{M}_i \mathbf{P}_{ij} \mathbf{M}_j \mathbb{1} - (\pi_i \mathbf{M}_i \mathbb{1}) \cdot (\pi_j \mathbf{M}_j \mathbb{1}). \tag{3.7}$$

Then the correlation coefficient of two consecutive absorption times in the composition can be determined as

$$\rho_{X_i X_j} = \frac{Cov[X_i, X_j]}{\sqrt{Var[X_i] \cdot Var[X_j]}} \tag{3.8}$$

$$= \frac{\pi_i \mathbf{M}_i \mathbf{P}_{ij} \mathbf{M}_j \mathbb{1} - (\pi_i \mathbf{M}_i \mathbb{1}) \cdot (\pi_j \mathbf{M}_j \mathbb{1})}{\sqrt{(2\pi_i (-\mathbf{D}_i)^{-2}\mathbb{1} - (\pi_i (-\mathbf{D}_i)^{-1}\mathbb{1})^2) \cdot (2\pi_j (-\mathbf{D}_j)^{-2}\mathbb{1} - (\pi_j (-\mathbf{D}_j)^{-1}\mathbb{1})^2)}} \tag{3.9}$$

Using the transfer matrix $\mathbf{H}_{ij}$ the correlation of two consecutive absorption times in the composition can be modeled. The absorption times of PHDs in the composition can be used in modeling and computation in different application areas. In particular, the absorption time is often interpreted as a time which is required until some action is performed. The consecutive absorption times in the composition can be interpreted as correlated edge weights for adjacent edges in a stochastic graph.

Note that in stochastic graphs edge weights are defined by random variables. Since the class of PHDs is dense in the sense of weak convergence in the class of all distributions with positive support [15], edge weights can be modeled w. l. o. g. by PHDs in stochastic graphs. Then PH distributed edge weights can be interpreted as traveling times or costs for adjacent edges [40], failure rates of components in series [46], interclaim times and claim sizes in insurance risk models [15] or strength of connection in functional brain networks [60].

***Example* 3.1.** *We consider the following two PHDs in the composition. The first PHD PH_i is given by*

$$\pi_i = (0.4, 0.6)\,,\ \mathbf{D}_i = \begin{pmatrix} -1.1529 & 0 \\ 0 & -12.941 \end{pmatrix},$$

and the subsequent PHD has the representation

$$\pi_j = (0.45, 0.55)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0.44444 \\ 0 & -5.0 \end{pmatrix}.$$

The transfer matrix resulting in $E[X_i^1, X_j^1] = 0.39050$ and $\rho_{X_i X_j} = 0.27798$ is

$$\mathbf{H}_{ij} = \begin{pmatrix} 1.1529 & 0 \\ 1.07842 & 11.86258 \end{pmatrix}.$$

The underlying Markov chain $\{X(t)\}_{t\geq 0}^{\infty}$ for the composition of two PHDs is visualized in Fig. 3.2. In particular, the equation $\pi_i\,(-\mathbf{D}_i)^{-1}\,\mathbf{H}_{ij}$ results in a vector $(0.45,\ 0.55)$ which is equal to π_j.

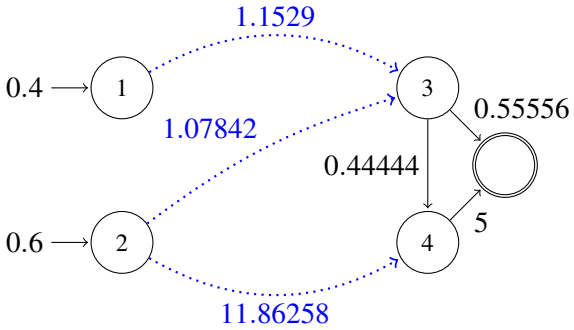


Figure 3.2.: The Markov process of the PHDs composition. The transition rates according to the matrix $\mathbf{H}_{ij}$ are highlighted in blue.

Dependencies between several PHDs Generally, a PHD composition can be extended to more than two PHDs. Let PHD PH_j depend on two PHDs PH_g and PH_i. Then additional phases should be introduced to code the exit phases of previous PHDs with representation $PH_g = (\pi_g, \mathbf{D}_g)$ and $PH_i = (\pi_i, \mathbf{D}_i)$. First we consider the following infinitesimal subgenerator matrix

$$\mathbf{D}_{min(g,i)} = \mathbf{D}_g \otimes \mathbf{I}_i + \mathbf{I}_g \otimes \mathbf{D}_i = \mathbf{D}_g \oplus \mathbf{D}_i, \tag{3.10}$$

where $\mathbf{I}_g$, $\mathbf{I}_i$ are identity matrices of order n_g and n_i, respectively. $\otimes$ and $\oplus$ denote the Kronecker product and Kronecker sum [58, 118] which are defined for two square matrices $\mathbf{A}$ and $\mathbf{B}$ of order a and b as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}(1,1)\mathbf{B} & \cdots & \mathbf{A}(1,a)\mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{A}(a,1)\mathbf{B} & \cdots & \mathbf{A}(a,a)\mathbf{B} \end{bmatrix} \text{ and } \mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_a + \mathbf{I}_b \otimes \mathbf{B}. \tag{3.11}$$

In fact, the underlying Markov process models the concurrent behavior of previous PHD processes $\{X_g(t)\}_{t\geq 0}^{\infty}$ and $\{X_i(t)\}_{t\geq 0}^{\infty}$. If it gets absorbed, the minimum of two PHDs PH_g, PH_i is determined [47]. Now the additional states $(n_g+1, \cdot)$ and $(\cdot, n_i+1)$ correspond to absorption of one of the PHDs involved. Reaching one of that states the Markov chain evolves according to the second PHD which has not been absorbed yet. The process described contains all combinations of transitions until absorption of two PHDs PH_g and PH_i. The following infinitesimal subgenerator matrix describes the behavior mentioned.

$$\mathbf{D}_{gi} = \begin{bmatrix} \mathbf{D}_{min(g,i)} & \mathbf{I}_g \otimes \mathbf{d}_1^i & \mathbf{d}_1^g \otimes \mathbf{I}_i \\ \mathbf{0} & \mathbf{D}_g & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_i \end{bmatrix}. \tag{3.12}$$

The corresponding Markov chain of order $n_g n_i + n_g + n_i$ contains pairs of phases $\{(k,l) : k \in \mathscr{S}_T^g, l \in \mathscr{S}_T^i\} \cup \{(n_g+1, l) \,|n_g+1 \in \mathscr{S}_A^g, l \in \mathscr{S}_T^i\} \cup \{(k, n_i+1) \mid , k \in \mathscr{S}_T^g, n_i+1 \in \mathscr{S}_A^i\}$, and the absorbing state. The initial probability vector is given by

$$\pi_{gi} = (\pi_g \otimes \pi_i, \pi_g \pi_i (n_i+1), \pi_g (n_g+1)\, \pi_i). \tag{3.13}$$

In fact, the Markov process with representation $(\pi_{gi}, \mathbf{D}_{gi})$ describes the PHD of the maximum of two PH_g and PH_i. In the case when PHD PH_j depend on more than one predecessor PHD the transfer matrix $\mathbf{H}_{(g,i)\,j}$ is of dimension $(n_g n_i + n_g + n_i) \times n_j$. Furthermore it holds

$$\mathbf{H}_{(g,i)\,j} \geq \mathbf{0},\ \mathbf{H}_{(g,i)\,j} \neq \mathbf{0},\ \mathbf{H}_{(g,i)\,j}\mathbb{1} = -\mathbf{D}_{gi}\mathbb{1},\ \pi_{gi}(-\mathbf{D}_{gi})\mathbf{H}_{(g,i)\,j} = \pi_j. \tag{3.14}$$

Formally, the tuple $((\pi_{gi}, \mathbf{D}_{gi}),\ (\pi_j, \mathbf{D}_j),\ \mathbf{H}_{(g,i)\,j})$ defines a composition of three PHDs PH_g, PH_i and PH_j. The infinitesimal subgenerator given in Eq. 3.15 and the initial distribution vector $\pi = (\pi_{gi},\ \mathbf{0})$, where $\mathbf{0}$ is of dimension $n_j \times 1$, describe the underlying Markov process for the composition containing PHD depending on two PHDs.

$$\mathbf{D}_0 = \begin{bmatrix} \mathbf{D}_{gi} & \mathbf{H}_{(g,i)\,j} \\ 0 & \mathbf{D}_j \end{bmatrix}. \tag{3.15}$$

***Example* 3.2.** *We consider the following PHDs in the composition. The first predecessor PHD PH_g is given by*

$$\pi_g = (0.5, 0.5)\,,\ \mathbf{D}_g = \begin{pmatrix} -7 & 0 \\ 0 & -0.5 \end{pmatrix},$$

and the second predecessor PHD PH_i is given by

$$\pi_i = (0.6, 0.4)\,,\ \mathbf{D}_i = \begin{pmatrix} -5 & 0 \\ 0 & -1 \end{pmatrix}.$$

We assume that the subsequent PHD PH_j is equal to PH_g. Let 1, 2 *denote the states of PH_g where state* 3 *is the absorbing state. PH_i contains transient states* 4, 5*, and the absorbing state* 6*. The subsequent PH_j contains transient states* 7 *and* 8*.*

Following transfer matrices are defined

$$\mathbf{H}_{gj} = \begin{pmatrix} 7 & 0 \\ 0 & 0.5 \end{pmatrix},\ \mathbf{H}_{ij} = \begin{pmatrix} 4.16667 & 0.83333 \\ 0 & 1 \end{pmatrix},$$

such that $\rho_{X_g X_j} = 0.3$ and $\rho_{X_i X_j} = 0.23$. The subgenerator $\mathbf{D}_{gi}$ is given by

$$\mathbf{D}_{gi} = \begin{bmatrix} -12 & 0 & 0 & 0 & 5 & 0 & 7 & 0 \\ 0 & -8 & 0 & 0 & 1 & 0 & 0 & 7 \\ 0 & 0 & -5.5 & 0 & 0 & 5 & 0.5 & 0 \\ 0 & 0 & 0 & -1.5 & 0 & 1 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & -7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

We computed the following transfer matrix $\mathbf{H}_{(g,i)\,j}$ with $\rho_{(X_g,X_i)X_j} = 0.162$

$$\mathbf{H}_{(g,i)\,j} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 7 & 0 \\ 0.167 & 0.333 \\ 5 & 0 \\ 0.0497 & 0.9503 \end{pmatrix}.$$

The underlying Markov process is visualized in Fig. 3.3.

3.2. Graphs with PH Distributed Edge Weights

We define a stochastic graph $G = (V, E, P)$ by the triple where V is a set of nodes, E is a set of edges, and a set P containing PHDs, and PHD compositions describing the statistics of edge weights. The graph G is then called a *PH-Graph*. In particular, the weight of an edge $i \in E$ is assumed to be a non-negative random variable X_i which is

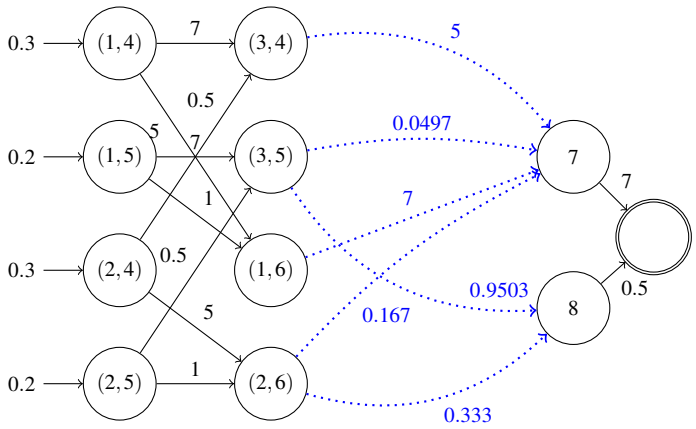


Figure 3.3.: The Markov process of the composition of three PHDs. The transition rates according to the matrix $\mathbf{H}_{(g,i)\,j}$ are highlighted in blue.

a PHD PH_i with representation $(\pi_i, \mathbf{D}_i)$ of order n_i. In addition, there is an origin node $v_{ini} \in V$ and a destination node $v_{fin} \in V$, $v_{ini} \neq v_{fin}$ [40].

The edges in graph G are directed. The weight of edge $i \in E$ is a random variable which can be interpreted as, for example a travel time from the starting node of i to the destination node of i. In particular, for the edge i $ini(i) \in V$ denotes the starting node, and $fin(i) \in V$ denotes the destination node. Any two edges i, j are adjacent if $fin(i) = ini(j)$. Furthermore we described the following sets of predecessor and successor edges for each edge $i \in E$ as described [40]

$$\begin{aligned} \bullet i &= \begin{cases} \emptyset & \text{if } ini(i) = v_{ini} \\ \{j | fin(j) = ini(i)\} & \text{otherwise} \end{cases} \\ i \bullet &= \begin{cases} \emptyset & \text{if } fin(i) = v_{fin} \\ \{j | ini(j) = fin(i)\} & \text{otherwise} \end{cases} \end{aligned} \tag{3.16}$$

Any adjacent edges i, j can have dependent weights and are modeled using the PHD composition. For i, $j \in E$, $i \in \bullet j$ the PHD PH_i in the composition corresponds to the weight of the edge i [40]. The subsequent PHD PH_j corresponds to the weight of the adjacent edge j which follows edge i in the graph. Then the dependency between weights of adjacent edges is encoded in the transfer matrix $\mathbf{H}_{ij}$. Thus the entry phase of PH_j of the successor edge is dependent on the exit phase of PH_i of the predecessor edge [40].

We assume that for every $v \in V$ at least one path to the destination v_{fin} exists and let $Paths$ be a set of all finite paths [40]. In addition, $E_{ini} = \{i \,|\, ini(i) = v_{ini}\}$ denotes a set of edges emanating from the origin v_{ini} and $E_{fin} = \{i \,|\, fin(i) = v_{fin}\}$ denotes a set of edges ending in the destination v_{fin}.

We define a path from the origin v_{ini} to the destination v_{fin} as a sequence of edges $(i_1, \ldots, i_K) \in Paths$ where $i_1 \in E_{ini}$, $i_K \in E_{fin}$ and $i_{k-1} \in \bullet i_k$ for $k = 2, \ldots, K$ [40]. The weight of a path is the sum of PHD weights of edges [40]. A possible interpretation of a path weight can be the length of a route traversed by a vehicle until the destination v_{fin} is reached. Furthermore, we assume that the stochastic graph can be cyclic where the weight of each cycle is non-negative.

For a given path $(i_1,\ldots,i_K) \in Paths$ between v_{ini} and v_{fin} for each $i_{k-1} \in \bullet i_k$ the PH-Graph contains the PHD composition $((\pi_{k-1},\mathbf{D}_{k-1}),\ (\pi_k,\mathbf{D}_k),\ \mathbf{H}_{(k-1)\,k}) \in P$. Accordingly, PHD $(\pi_k,\mathbf{D}_k)$ with $1 < k < K$ and $K > 2$, appears in two PHD compositions, once as second PHD in the composition, and once as first PHD in the composition (see Ex. 3.3). Let (i_{k-1},i_k,i_{k+1}) be the corresponding subpath. Then two PHD compositions $((\pi_{k-1},\mathbf{D}_{k-1}),\ (\pi_k,\mathbf{D}_k),\ H_{(k-1)\,k})$, $((\pi_k,\mathbf{D}_k),\ (\pi_{k+1},\mathbf{D}_{k+1}),\ H_{k\,(k+1)})$ describe the sum of edge weights along the subpath. Observe that even though correlation is defined for subsequent edges, i.e. is defined within a PHD composition, the effect of correlation cummulates for edges along the path of length greater than two.

In fact, a path $(i_1,\ldots,i_K) \in Paths$ in a PH-Graph corresponds to an absorbing Markov chain with $\sum_{k=1}^{K} n_{i_k}$ states (see Sec. 2.1.3). Then the state space of the absorbing CTMC $\mathscr{S}$ contains the states of PHDs corresponding to edges along the path.

Definition 3.2. *[40] The time until absorption along a path $(i_1,\ldots,i_K) \in Paths$ is defined by an absorbing CTMC with subgenerator matrix given in Eq. 3.17. The initial vector of the CTMC equals $\pi = (n_{i_1},\mathbf{0})$ where $\mathbf{0}$ is the row $(n-n_{i_1})$-vector of 0's.*

$$\mathbf{Q}_{(i_1,\ldots,i_K)} = \begin{pmatrix} \mathbf{D}_{i_1} & \mathbf{H}_{i_1 i_2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{i_2} & \mathbf{H}_{i_2 i_3} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & & \ddots & \mathbf{D}_{i_{K-1}} & \mathbf{H}_{i_{K-1} i_K} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{D}_K \end{pmatrix}. \tag{3.17}$$

***Example* 3.3.** *Consider the stochastic graph in Fig. 3.4. We have two paths for a given destination v_5, namely (i_1,i_2,i_4) and (i_1,i_3,i_5). The visualized PHDs of order 2 are assigned to edges such that corresponding transfer matrices are highlighted in blue. If edge weights are interpreted as traveling times, the traveling times along the*

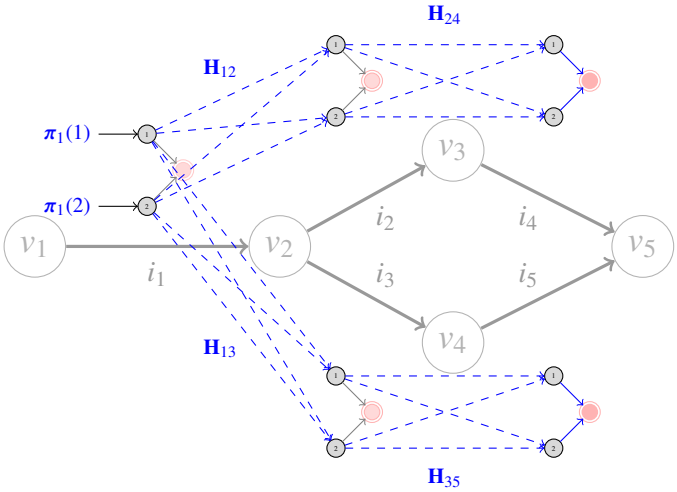


Figure 3.4.: Example of the PH-Graph with $v_{ini} = v_1$ and $v_{fin} = v_5$.

paths (i_1,i_2,i_4) and (i_1,i_3,i_5) are described by absorbing CTMCs with the following

subgenerator matrices and initial distribution vector $\pi = (\pi_1(1), \pi_1(2), \mathbf{0})$.

$$\mathbf{Q}_{(i_1,i_2,i_4)} = \begin{pmatrix} \mathbf{D}_{i_1} & \mathbf{H}_{i_1 i_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{i_2} & \mathbf{H}_{i_2 i_4} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_{i_4} \end{pmatrix}, \mathbf{Q}_{(i_1,i_3,i_5)} = \begin{pmatrix} \mathbf{D}_{i_1} & \mathbf{H}_{i_1 i_3} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{i_3} & \mathbf{H}_{i_3 i_5} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_{i_5} \end{pmatrix}.$$

3.3. Summary and Overview

In this chapter we first described the composition of two PH distributions and then presented the developed extension called a PH-Graph model. In a PH-Graph model edge weights are described by PHDs and correlations between adjacent edge pairs are modeled using transfer matrices - the concept adopted from MAPs. Thus PH-Graphs allow for modeling dependent edge weights in stochastic graphs.

Stochastic graph models including correlations find applications in numerous fields, e.g., in vehicle routing. If we consider vehicle routing as an application for PH-Graphs then given a stochastic graph G a vehicle traverses the graph edges from the origin v_{ini} to the destination v_{fin}. The traveling time which is spent by a vehicle traversing the graph is not known a priori. The vehicle only knows the PHD of edges which it travels and the existing dependencies between edges. As the vehicle moves through the graph the edge traveling time realized becomes known to it.

It can be assumed that every time the vehicle visits a node $v_{fin}(i)$ the realization of the random variable X_i is known and can be collected in a set of realizations of traveling times for edge i. Furthermore, in dependence of the known realized traveling times and dependencies between edge weights of adjacent edges the vehicle can decide which edge to traverse next. Thus one is interested in efficient parameterization methods for PH-Graphs which will be considered in Chapter 4. Existing fitting approaches [47] can be used to model PHDs at graph edges. More attention is devoted to fitting methods for transfer matrices $\mathbf{H}$ which add correlation in PH-Graph models.

Various fitting algorithms work on specific PHD representations. Furthermore, parameters of two PHDs in the composition are incorporated in fitting of the corresponding transfer matrix $\mathbf{H}$. This implies that matrix representations of PHDs PH_i, PH_j in the composition have significant influence on parameters of the transfer matrix $\mathbf{H}_{ij}$. In Chapter 6 we will show which matrix representations of two PHDs are most suitable in order to maximize the correlation of their composition.

Based on the PH-Graph model efficient solution methods for Stochastic Shortest Path Problems with Correlations can be developed. Competing paths from v_{ini} to v_{fin} in a PH-Graph can be interpreted over continuous-time Markov decision processes. SSPP with Correlations over CTMDP requires then computation of optimal control strategies to find the path with minimum expected total time of reaching a destination. The extension of the PH-Graph model over CTMDPs with rewards will be introduced in Chapter 5. The solution technique to compute the minimum expected total time of reaching the destination and the numerical approach for time bounded reachability results will also be presented in Chapter 5.

Chapter 4

Fitting Algorithms

In this chapter the parameterization problem of PH-Graphs is considered. Though various fitting approaches exist to compute PHDs for the weights of edges in a PHG, fitting the parameters for transfer matrices is much more complex. The major reason is that a long trace should be considered for adequate correlation fitting when trace-based fitting methods are used. Furthermore transfer matrices should preserve important conditions within a composition which requires an additional solution of optimization problems.

First an introduction of *trace-based* fitting methods for PHDs is given in order to explain the parameterization of the PHD composition. Then we present approaches for fitting the transfer matrix which have been derived from existing approaches for MAPs. In particular, an EM algorithm for MAPs has been extended to transfer matrices. Although its computational effort is higher than the effort of the EM algorithm for PHDs, it can be applied in practice. *Two phase* fitting approaches have also been derived from existing approaches for MAPs. The methods can be used to compute the parameters according to first joint moments or correlation coefficent from a trace and are also described.

4.1. Trace-Based Fitting Methods

Assume that there is a process of interest *Proc*. Often the behavior of the process *Proc* should be approximated using an adequate model. When models based on PHDs and MAPs are to be used, the objective of a fitting procedure is the computation of parameters $(\boldsymbol{\pi}, \mathbf{D}_0)$ (for PHDs), $(\mathbf{D}_0, \mathbf{D}_1)$ (for MAPs), and of $((\boldsymbol{\pi}_i, \mathbf{D}_i), (\boldsymbol{\pi}_j, \mathbf{D}_j), \mathbf{H}_{ij})$ (for PHD compositions). The models based on PHDs and MAPs should then exhibit statistical properties equal or similar to properties of the process *Proc*.

As described in [141], the behavior of process *Proc* is usually substituted by a finite observed sequence of data, e.g. inter-arrival or service times, which is denoted as a trace $\mathscr{T} = (t_1, \ldots, t_m)$. The reason is that the behavior of the process *Proc* cannot be infinitely observed from the real system or from an adequate simulation model. The trace $\mathscr{T}$ is defined as a realization of the process *Proc*. Its statistical properties should resemble the characteristics of the underlying process *Proc* [141].

Let K be a set of statistical measures characterizing process *Proc*. A statistical

measure $\kappa_i \in K$ can be either directly given by κ_{Proc} or can be estimated from the trace $\mathscr{T}$ as $\kappa_{\mathscr{T}}$ [141]. The fitting procedure approximates κ_{Proc} or $\kappa_{\mathscr{T}}$ by the statistical measure of the model based on PHDs or MAPs, e.g. by $\kappa_{(\pi,\mathbf{D}_0)}$ computed from a PHD with a representation $(\pi,\mathbf{D}_0)$.

4.1.1. Trace Definition and Properties

Often traces contain the measured data points which correspond to preliminary observations of the process. It can be described by a sequence of m chronologically ordered points in time $t_i > 0$, $i = 1,\ldots,m$. For our application example, the trace can result from measurements of traveling times on the road which requires bookkeeping of the time steps of vehicles. In this case an element t_i describes the traveling time of the ith vehicle passing through the road segment. Commonly, an element t_i describes the inter-event time of the i-th event, and many other interpretations are possible. Furthermore, the sequence $t_1,\ldots,t_m$ is assumed to be in a strict-sense stationary, which implies a common distribution of t_i independent on i.

Fitting methods can be divided in two classes. Trace-based fitting approaches use the complete trace for the parameter computation of PHDs and MAPs. The second class of fitting algorithms uses only some statistical measures estimated from the trace. Consequently, the parameter of models using PHDs and MAPs are computed in order to approximate the derived measures as close as possible. The advantages and disadvantages of both methods are described in [47].

In the following we describe some statistical measures which can be estimated from the trace in order to determine parameters of models based on PHDs and MAPs. The estimator for the i-th moment of the trace and the variance are given by [112, 47]

$$\hat{\mu}_i = \frac{1}{m}\sum_{j=1}^{m}(t_j)^i \quad \text{and} \quad \hat{\sigma}^2 = \frac{1}{m-1}\sum_{j=1}^{m}(t_j-\hat{\mu}_1)^2. \tag{4.1}$$

Often there are dependencies between consecutive data points t_i, t_{i+j}, $j = 1,\ldots$. Then the autocorrelation or the joint moments are of interest. The coefficient of autocorrelation of data points that are lag k apart is estimated by [112, 47]

$$\hat{\rho}_k = \frac{1}{(m-k-1)\,\hat{\sigma}^2}\sum_{j=1}^{m-k}(t_j-\hat{\mu}_1)(t_{j+k}-\hat{\mu}_1). \tag{4.2}$$

The estimator of the joint moments $\mu_{ij} = E[X_k^i,\, X_{k+1}^j]$ of two consecutive data points is given by [47]

$$\hat{\mu}_{ij} = \frac{1}{m-1}\sum_{j=1}^{m-1}(t_k)^i(t_{k+1})^j. \tag{4.3}$$

The empirical distribution function of a trace is given by a step function with m steps [112, 47]

$$F_{\mathscr{T}}(x) = \frac{\sum_{j=1}^{m}\delta(t_j \le x)}{m}, \qquad \delta(b) = \begin{cases} 1 & \text{if } b = true, \\ 0 & \text{if } b = false. \end{cases} \tag{4.4}$$

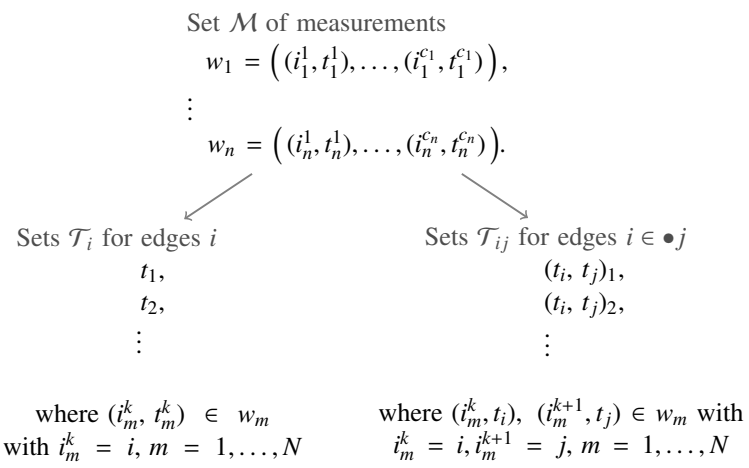


Figure 4.1.: Trace extraction from the set of measurements $\mathscr{M}$.

Edge Weights Trace To derive the parameters of the PH-Graph we need to consider the realizations of edge weights. In order to obtain coherent parameters of PHDs for edge weights, the measurements of entities that passed through the graph should be collected. Then several traces could be used to compute parameters of PHDs and transfer matrices.

A single measurement of related edge weights is a sequence $w_n = ((i_n^1, t_n^1), \ldots, (i_n^{c_n}, t_n^{c_n}))$ where edges $i_n^k \in \mathscr{E}$ build a path, i.e., $i_n^k \in \bullet i_n^{k+1}$, and $0 < t_n^k < \infty$, for the number of edges along the path $1 \leq k \leq c_n$. Here, the sequence need not start in v_{ini} or end in v_{fin}. Let $\mathscr{M}$ be a set of all measurements w_n and N the number of measured sequences.

Using sequences w_n the trace $\mathscr{T}_i$ containing all measured weights for edge $i \in \mathscr{E}$, and the trace $\mathscr{T}_{ij}$ containing all measured value pairs of adjacent edges i and j can be obtained obtained.

In particular, the trace $\mathscr{T}_i$ contains a sequence $t_1, t_2 \ldots$ of all measured edge weights t_n from $w_n \in \mathscr{M}$ where w_n contains a pair (i_n^k, t_n^k) with $i_n^k = i$. Observe that if the value t_n appears in several measurement sequences or several times in one measurement sequence, then it appears several times in the corresponding chronological order in the trace $\mathscr{T}_i$.

Similarly, the trace $\mathscr{T}_{ij}$ contains value pairs (t_i, t_j) which appear in some $w_n \in \mathscr{M}$ such that a pair of tuples (i_n^k, t_i), $(i_n^{k+1}, t_j) \in w_n$, and $i_n^k = i, i_n^{k+1} = j$. Again the trace $\mathscr{T}_{ij}$ contains several entries (t_i, t_j), if they appear several times in some w_n or in several measurement sequences. We denote N_i and N_{ij} as the number of elements in the trace $\mathscr{T}_i$ and $\mathscr{T}_{ij}$, respectively. Fig. 4.1 shows the extraction of traces $\mathscr{T}_i$, $\mathscr{T}_{ij}$ for $i, j \in \mathscr{E}$ graphically.

In particular, using a trace $\mathscr{T}_i$ and Eq. 4.1 the estimate for $\mu_k^i = E[X_i^k]$ is

$$\hat{\mu}_k^i = \frac{1}{N_i} \sum_{t_j \in \mathscr{T}_i} (t_j)^k \quad \text{and} \quad \hat{\sigma}_i^2 = \frac{1}{N_i - 1} \sum_{t_j \in \mathscr{T}_i} (t_j - \hat{\mu}_1^i)^2. \tag{4.5}$$

Considering the traces $\mathscr{T}_{ij}$ the estimator of the joint moments $\mu_{k,l}^{ij} = E[X_i^k\, X_j^l]$ for

weights of adjacent edges i and j is given by

$$\hat{\mu}_{k,l}^{ij} = \frac{1}{N_{ij}-2} \sum_{(t_i,t_j)\in\mathscr{T}_{ij}} (t_i)^k (t_j)^l. \tag{4.6}$$

In PH-Graphs there are dependencies between weights of adjacent edges t_i, t_j with $i \in \bullet j$. The coefficient of correlation of two consecutive edge weights is estimated by

$$\hat{\rho}_{ij} = \frac{1}{(N_{ij}-2)\,\hat{\sigma}_i\hat{\sigma}_j} (\hat{\mu}_{1,1}^{ij} - \hat{\mu}_1^i \hat{\mu}_1^j). \tag{4.7}$$

The coefficent of correlation of data points that are lag k apart is estimated by

$$\hat{\rho}_{ij}^k = \frac{1}{(N_{ij}-k-1)\,\hat{\sigma}_i\hat{\sigma}_j} \sum_{\substack{l=1 \\ (t_i,\cdot),(\cdot,t_j)\in\mathscr{T}_{ij}}}^{N_{ij}-k} ((t_i)_l - \hat{\mu}_i^1)((t_j)_{l+k} - \hat{\mu}_j^1). \tag{4.8}$$

4.1.2. Expectation Maximization Algorithm for PHDs

The Expectation Maximization (EM) approach is an iterative algorithm which computes the maximum-likelihood estimate of parameters of an underlying distribution from the complete trace. There are several EM algorithms for PHDs. Asmussen *et al.* developed in [10] the EM algorithm for general PHDs. The EM fitting approach for APHDs in canonical form was proposed in [30]. Uniformization based methods allow for more efficient implementations of the EM algorithm [38, 48, 109]. In this section we give an overview of the EM algorithm for general PHDs including the improvement using uniformization described in [10, 38, 47].

If the complete measured data collected in a trace $\mathscr{T} = (t_1, \ldots, t_m)$ should be used, then usually the likelihood

$$\mathscr{L}((\boldsymbol{\pi}, \mathbf{D}_0)|\mathscr{T}) = \prod_{i=1}^{m} \boldsymbol{\pi} e^{\mathbf{D}_0 t_i} \mathbf{d}_1, \tag{4.9}$$

is maximized. In (4.9) the value $\mathscr{L}((\boldsymbol{\pi}, \mathbf{D}_0)|\mathscr{T})$ gives the likelihood that the PHD $(\boldsymbol{\pi}, \mathbf{D}_0)$ generates the trace $\mathscr{T}$. Then the optimal parameter estimation satisfies

$$(\boldsymbol{\pi}, \mathbf{D}_0)^* = \arg\max_{(\boldsymbol{\pi},\mathbf{D}_0)} \prod_{i=1}^{m} \boldsymbol{\pi} e^{\mathbf{D}_0 t_i} \mathbf{d}_1. \tag{4.10}$$

The expectation (E) step determines the distribution of the unobserved data, given known values of observations and the current estimate of the distribution parameters $(\hat{\boldsymbol{\pi}}, \hat{\mathbf{D}_0})$. The observed data corresponds to absorption times $(t_1, \ldots, t_m)$. The unobserved data corresponds to the states of PHD visited before absorption, state holding times etc. In the maximization (M) step the parameters are reestimated to be those with the maximum-likelihood, given that the distribution was computed correctly in the E step.

To obtain the complete data in the E-step one can use the embedded Markov process $\{X_r\}_{0\le r\le t_i}$ (see Sec. 2.1.2). Let k be the number of steps of the Markov process before absorption occurs, and let n be the defined order of the PHD. Furthermore, the sequence $X_0, \ldots, X_{k-1}$ denotes the sequence of visited states of the PHD

until absorption. The sequence $S_0,\ldots,S_{k-1}$ denotes the corresponding state holding times [10, 47]. The behavior of $\{X_r\}_{0\le r\le t_i}$ on the interval $[0,t_i]$ is given by a tuple $z=(x_0,\ldots,x_{k-1},s_0,\ldots,s_{k-1})$. Then the density function of the complete observation can be written as [47]

$$\begin{aligned} f(z|(\pi,\mathbf{D}_0)) &= \pi(x_0)\lambda(x_0)e^{-\lambda(x_0)s_0}\tilde{\mathbf{P}}(x_0,x_1)\cdots\lambda(x_{k-1})e^{-\lambda(x_{k-1})s_{k-1}}\tilde{\mathbf{P}}(x_{k-1},n+1)\\ &= \pi(x_0)e^{-\lambda(x_0)s_0}\mathbf{D}_0(x_0,x_1)\cdots e^{-\lambda(x_{k-1})s_{k-1}}\mathbf{d}_1(x_{k-1}). \end{aligned}$$

Considering the whole trace data $(t_1,\ldots,t_m)$ the observation contains m outcomes, such that the tuple $z=(x_0^1,\ldots,x_0^m,\ldots,s_{k-1}^1,\ldots,s_{k-1}^m)$. The likelihood function is then given by [47]

$$\mathscr{L}((\pi,\mathbf{D}_0)|\mathscr{T}) = f(z|(\pi,\mathbf{D}_0)) = \prod_{i=1}^{n}\pi(i)^{B_i}\prod_{i=1}^{n}e^{Z_i\mathbf{D}_0(i,i)}\prod_{i=1}^{n}\prod_{j=1,j\neq i}^{n+1}\mathbf{D}_0(i,j)^{N_{ij}},$$

where the variable B_i denotes the number of times the Markov process started in state i, Z_i denotes the total time spent in state i, and N_{ij} is the total number of jumps from state i to state j, for $i\neq j$, $i\in\mathscr{S}_T$, and $j\in\mathscr{S}$.

The forward vector $\mathbf{f}$, the backward vector $\mathbf{b}$, and the flow matrix $\mathbf{F}$ are defined as

$$\begin{aligned} \mathbf{f}_{(\pi,\mathbf{D}_0),t} &= \pi e^{\mathbf{D}_0 t},\\ \mathbf{b}_{(\pi,\mathbf{D}_0),t} &= e^{\mathbf{D}_0 t}\mathbf{d}_1,\\ \mathbf{F}_{(\pi,\mathbf{D}_0),t} &= \int_0^t \left(\mathbf{f}_{(\pi,\mathbf{D}_0),t-u}\right)^T\left(\mathbf{b}_{(\pi,\mathbf{D}_0),u}\right)^T du, \end{aligned} \tag{4.11}$$

and can be computed in the E-step with standard methods like Runge-Kutta method [11], or using uniformization [162, 47]. Using these vectors the likelihood can be computed as [47]

$$\mathscr{L}((\pi,\mathbf{D}_0)|\mathscr{T}) = \prod_{k=1}^{m}\pi\mathbf{b}_{(\pi,\mathbf{D}_0),t_k} \text{ and} \log\left(\mathscr{L}((\pi,\mathbf{D}_0)|\mathscr{T})\right) = \sum_{k=1}^{m}\log\left(\pi\mathbf{b}_{(\pi,\mathbf{D}_0),t_k}\right) \tag{4.12}$$

Given the current estimate $(\pi,\mathbf{D}_0)$ of the PHD the conditional expectations of B_i,Z_i,N_{ij} can be obtained as [10, 47]

$$\begin{aligned} E_{(\pi,\mathbf{D}_0),\mathscr{T}}[B_i] &= \tfrac{1}{m}\sum_{k=1}^{m}\frac{\pi(i)\mathbf{b}_{(\pi,\mathbf{D}_0),t_k}(i)}{\pi\mathbf{b}_{(\pi,\mathbf{D}_0),t_k}},\\ E_{(\pi,\mathbf{D}_0),\mathscr{T}}[Z_i] &= \tfrac{1}{m}\sum_{k=1}^{m}\frac{\mathbf{F}_{(\pi,\mathbf{D}_0),t_k}(i,i)}{\pi\mathbf{b}_{(\pi,\mathbf{D}_0),t_k}},\\ E_{(\pi,\mathbf{D}_0),\mathscr{T}}[N_{ij}] &= \tfrac{1}{m}\sum_{k=1}^{m}\frac{\mathbf{D}_0(i,j)\mathbf{F}_{(\pi,\mathbf{D}_0),t_k}(i,j)}{\pi\mathbf{b}_{(\pi,\mathbf{D}_0),t_k}},\\ E_{(\pi,\mathbf{D}_0),\mathscr{T}}[N_{in+1}] &= \tfrac{1}{m}\sum_{k=1}^{m}\frac{\mathbf{d}_1(i)\mathbf{f}_{(\pi,\mathbf{D}_0),t_k}(i)}{\pi\mathbf{b}_{(\pi,\mathbf{D}_0),t_k}}. \end{aligned} \tag{4.13}$$

In the M-step the new parameters of the PHD are estimated in order to maximize the likelihood

$$\begin{aligned} \hat{\pi}(i) &= E_{(\pi,\mathbf{D}_0),\mathscr{T}}[B_i], \quad \hat{\mathbf{D}}_0(i,j) = \tfrac{E_{(\pi,\mathbf{D}_0),\mathscr{T}}[N_{ij}]}{E_{(\pi,\mathbf{D}_0),\mathscr{T}}[Z_i]},\\ \hat{\mathbf{d}}_1(i) &= \tfrac{E_{(\pi,\mathbf{D}_0),\mathscr{T}}[N_{in+1}]}{E_{(\pi,\mathbf{D}_0),\mathscr{T}}[Z_i]}, \quad \hat{\mathbf{D}}_0(i,i) = -(\hat{\mathbf{d}}_1(i)+\sum_{i\neq j}^{n}\hat{\mathbf{D}}_0(i,j)). \end{aligned} \tag{4.14}$$

Algorithm 4.1: EM algorithm for computing PHD $(\boldsymbol{\pi}, \mathbf{D}_0)$ using $\mathscr{T} = (t_1, \ldots, t_m)$

1: Choose PHD $(\boldsymbol{\pi}^{(0)}, \mathbf{D}_0^{(0)})$ and set $r = 0$;
2: **repeat**
3: Compute $\mathbf{f}_{(\boldsymbol{\pi},\mathbf{D}_0),t}$, $\mathbf{b}_{(\boldsymbol{\pi},\mathbf{D}_0),t}$ and $\mathbf{F}_{(\boldsymbol{\pi},\mathbf{D}_0),t}$ using $(\boldsymbol{\pi}^{(r)}, \mathbf{D}_0^{(r)})$ in Eq. 4.11 or using uniformization [47, Eq. 3.12] for $t = t_1, \ldots, t_m$;
4: **E-step**: Compute the conditional expectations using Eq. 4.14;
5: **M-step**: Compute $(\boldsymbol{\pi}^{(r+1)}, \mathbf{D}_0^{(r+1)})$ using Eq. 4.14 and set $r = r + 1$;
6: **until** $\|\boldsymbol{\pi}^{(r)} - \boldsymbol{\pi}^{(r-1)}\| + \|\mathbf{D}_0^{(r)} - \mathbf{D}_0^{(r-1)}\| < \varepsilon$;
7: Return $(\boldsymbol{\pi}^{(r)}, \mathbf{D}_0^{(r)})$;

Algorithm 4.1 from [47] describes the complete EM algorithm for general PHDs.

Observe that EM algorithm is an iterative local maximization method, such that the sequence of estimates with a non-decreasing likelihood values is generated. The sequence of estimates may either result in a local maximum or a saddle point which often depends on the initial parameter choice [62, 177].

Note that it is also possible to fit a PHD according to some statistical quantities from the measurements collected in a trace $\mathscr{T}$. Commonly, k order moments are estimated from the trace (see Eq. 4.5). Fitting algorithms according to the moments are described in [31, 44, 94]. Then the obtained PHD representation $(\boldsymbol{\pi}, \mathbf{D}_0)$ could be used as the initial parameter estimation in the EM algorithm to reduce the runtime. Methods which use the complete trace or moments from the trace are implemented in several tools [25, 93].

EM Algorithm for Hyper-Erlang PHDs EM algorithms for PHDs have been extended to restricted classes [108, 168] which enables more efficient fitting algorithms without reducing the resulting PHD's quality significantly. In this section we describe the approach developed in [168] which is implemented in the freely avalaible tool *gfit*. We used this software for parameterization of PH-Graphs in our experiments. The presented EM algorithm computes Hyper-Erlang PHDs (HErD) (see Sec. 2.4.1). The Hyper-Erlang class constitutes no restriction since any pmf of a non-negative random variable can be approximated arbitrary close by mixtures of Erlang distributions of unlimited order (cf. Theorem 1 from [168]).

For HErD fitting a mixture of K mutually independent Erlang branches should be determined. The i-th Erlang distribution is weighted with initial probability $\pi(i)$, where $i = 1, \ldots, K$ and $\sum_{i=1}^{K} \pi(i) = 1$. Furthermore the i-th Erlang branch contains s_i phases with rate parameter $\lambda(i)$ [47]. The density of the i-th Erlang distribution is given by (see Eq. 4.15)

$$f(x) = \sum_{i=1}^{m} \pi(i) \frac{(\lambda(i)x)^{s_i-1}}{(s_i-1)!} \lambda(i) e^{-\lambda(i)x} \text{ for } x \geq 0, \tag{4.15}$$

Let $\Theta = \{\pi(1), \ldots, \pi(K), \lambda(1), \ldots, \lambda(K)\}$ be the vector of parameters. The trace $\mathscr{T} = (t_1, \ldots, t_m)$ corresponds to the observed data, such that t_i are independent and

identically distributed with the density

$$f(t_i|\Theta) = \sum_{k=1}^{K} \pi(k) f_k(t_i|\lambda(k)). \tag{4.16}$$

The unobserved data is denoted by $z_i \in \{1,\ldots,K\}$, $i = 1,\ldots,m$, which corresponds to the Erlang branch that generated the trace element t_i. Considering the whole observation $\mathscr{T} = (t_1,\ldots,t_m)$ and $z = (z_1,\ldots,z_m)$ the log-likelihood can be given as

$$\log \mathscr{L}(\Theta \,|\, \mathscr{T}, z) = \sum_{i=1}^{m} \log(\pi(z_i)\, f_{z_i}(t_i \,|\, \lambda(z_i))). \tag{4.17}$$

Assume that $\hat{\Theta} = \{\hat{\pi}(1),\ldots,\hat{\pi}(K),\hat{\lambda}(1),\ldots,\hat{\lambda}(K)\}$ is the current estimation of the parameters, and Z is the random variable generating z_i. Then the probability mass function of the unobserved data z given the observed data $\mathscr{T}$ and the estimates $\hat{\Theta}$ can be computed by

$$q(z_i \,|\, t_i, \hat{\Theta}) = \frac{q(z_i \,|\, \hat{\Theta}) \cdot f(t_i \,|\, z_i, \hat{\Theta})}{f(t_i \,|\, \hat{\Theta})} = \frac{\hat{\pi}(i) \cdot f_{z_i}(t_i \,|\, \hat{\lambda}(z_i))}{\sum_{k=1}^{K} \hat{\pi}(k)\, f_k(t_i \,|\, \hat{\lambda}(k))}, \tag{4.18}$$

$$q(z \,|\, \mathscr{T}, \hat{\Theta}) = \prod_{i=1}^{m} q(z_i \,|\, t_i, \hat{\lambda}(z_i)), \tag{4.19}$$

and the conditional expectation of the complete data log-likelihood is given by

$$\begin{aligned} E(\log \mathscr{L}(\Theta \,|\, Z, \mathscr{T}) \,|\, \mathscr{T}, \hat{\Theta}) &= \sum_{z \in \{1,\ldots,K\}^m} \log \mathscr{L}(\Theta \,|\, z, \mathscr{T}) \cdot q(z \,|\, \mathscr{T}, \hat{\Theta}) \\ = \sum_{k=1}^{K} \sum_{i=1}^{m} \log(\pi(k)) \cdot q(k|t_i, \hat{\lambda}(k)) &+ \sum_{k=1}^{K} \sum_{i=1}^{m} \log(f_k(t_i \,|\, \lambda(k))) \cdot q(k|t_i, \hat{\lambda}(k)). \end{aligned} \tag{4.20}$$

The maximization of the conditional expectation according to the parameters $\hat{\Theta}$ is performed in the M-step. In (4.20) the first term containing $\pi(k)$ and the second term containing $\lambda(k)$ can be maximized independently using the following closed-form formulas [168, 47]

$$\pi(k) = \frac{1}{m} \sum_{i=1}^{m} q(k \,|\, t_i, \hat{\lambda}(k)), \qquad \lambda(k) = \frac{s_k \sum_{i=1}^{m} q(k \,|\, t_i, \hat{\lambda}(k))}{\sum_{i=1}^{m} q(k \,|\, t_i, \hat{\lambda}(k)) t_i}. \tag{4.21}$$

In each iteration of successive E and M-steps the sequence of non-decreasing log-likelihood values is generated until the algorithm converges to a local maximum. In [168, 47] the two following convergence criteria were proposed. First the maximal difference of the values of the parameter vectors of successive iterations can be considered. Secondly the relative difference of the log-likelihood values of successive iterations is important. The EM algorithm for HErD fitting based on $\Theta = (\pi(1), \ldots, \pi(K), \lambda(1), \ldots, \lambda(K))$ is presented in [168, 47] and is given below.

Note that if an aggregated trace [47, Sec. 3.1.1] of length $m^* < m$ is considered, an efficient speed up of the EM algorithm can be achieved, such that the runtime complexity is reduced from $\mathscr{O}(mK)$ to $\mathscr{O}(m^*K)$ which is independent of the PHD order in both cases.

Algorithm 4.2: EM algorithm for computing HErD $(\boldsymbol{\pi}, \mathbf{D}_0)$ using $\mathscr{T} = (t_1, \ldots, t_m)$

1: Choose initial parameter estimates $\hat{\Theta} = (\hat{\pi}(1), \ldots, \hat{\pi}(K), \hat{\lambda}(1), \ldots, \hat{\lambda}(K))$;
2: **repeat**
3: Compute the logarithmic form of the density function given by Eq. 4.15
4: $f_k(t_i \,|\, \hat{\lambda}(k)) = \hat{\lambda}(k)\, e^{(s_k-1)\ln(\hat{\lambda}(k)t_i) - \ln(s_k-1)! - \hat{\lambda}(k)t_i}$ for all $i = 1, \ldots, m$, $k = 1, \ldots, K$;
5: **E-step**: Compute the pmf of the unobserved data for $i = 1, \ldots, m$, $k = 1, \ldots, K$
6: $q(k \,|\, t_i, \hat{\lambda}(k)) = \frac{\hat{\pi}(k) \cdot f_k(t_i \,|\, \hat{\lambda}(k))}{\sum_{j=1}^{K} \hat{\pi}(j) f_j(t_i \,|\, \hat{\lambda}(j))}$;
7: **M-step**: Compute $\pi(k)$ and $\lambda(k)$ that maximize the conditional expectation Eq. 4.20 for $i = 1, \ldots, m$ according to Eq. 4.21;
8: Set $\hat{\Theta} := \Theta$;
9: **until** described convergence criterion reached;

4.1.3. EM Algorithm for Transfer Matrices

In this section we consider an EM algorithm for computing the parameters of transfer matrix $\mathbf{H}_{ij}$ using a given trace $\mathscr{T}_{ij}$. If we assume that PHDs $(\boldsymbol{\pi}_i, \mathbf{D}_i)$, $(\boldsymbol{\pi}_j, \mathbf{D}_j)$ of a composition are given from a preceding PHD fitting step the EM algorithm computes only the elements of the transfer matrix $\mathbf{H}_{ij}$. The proposed approach is the extention of EM algorithms for the parameter fitting of PHDs and MAPs (cf. Sec. 4.1.2, [47, 38, Sec. 5.2]). Considering the whole trace $\mathscr{T}_{ij}$ with consecutive traveling times of adjacent edges i, j the likelihood function is defined as

$$\mathscr{L}(\mathbf{H}_{ij} \,|\, \mathscr{T}_{ij}) = \boldsymbol{\pi}_i \prod_{(t_i^{(k)}, t_j^{(k)}) \in \mathscr{T}_{ij}} e^{t_i^{(k)} \mathbf{D}_i} \mathbf{H}_{ij}\, e^{t_j^{(k)} \mathbf{D}_j} (-\mathbf{D}_j) \mathbb{1}. \tag{4.22}$$

The optimal model parameters for the transfer matrix $\mathbf{H}_{ij}$ satisfy

$$\mathbf{H}_{ij}^{*} = \arg\max_{\mathbf{H}_{ij}} \boldsymbol{\pi}_i \prod_{(t_i^{(k)}, t_j^{(k)}) \in \mathscr{T}_{ij}} e^{t_i^{(k)} \mathbf{D}_i} \mathbf{H}_{ij}\, e^{t_j^{(k)} \mathbf{D}_j} \mathbf{d}_1^j, \tag{4.23}$$

such that conditions $\mathbf{H}_{ij} \geq \mathbf{0}$, $\mathbf{H}_{ij}\mathbb{1} = -\mathbf{D}_i\mathbb{1}$, $\boldsymbol{\pi}_i \mathbf{M}_i \mathbf{H}_{ij} = \boldsymbol{\pi}_j$ are satisfied, and $\mathbf{d}_1^j = (-\mathbf{D}_j)\mathbb{1}$.

First the forward and backward vectors are defined. For the trace $\mathscr{T}_{ij} = ((t_i^{(1)}, t_j^{(1)}), (t_i^{(2)}, t_j^{(2)}), \ldots, (t_i^{(K)}, t_j^{(K)}))$, where the number of value pairs is $K = N_{ij}$, the kth observation corresponds to the value pair $(t_i^{(k)}, t_j^{(k)})$. Consider the forward variable $\mathbf{f}_{t_i}^{(k)}(x)$ defined as

$$\mathbf{f}_{t_i}^{(k)}(x) = \boldsymbol{\pi}_i\, e^{t_i^{(k)} \mathbf{D}_i}(x), \tag{4.24}$$

which is computed using the partial absorption time $t_i^{(k)} \in (t_i^{(k)}, t_j^{(k)})$ and phase x at time $t_i^{(k)}$. The forward value 4.24 is obtained as the joint density of being in phase x of PH_i at time $t_i^{(k)}$ and initial distribution of the PHD PH_i.

In a similar manner, the backward variable $\mathbf{b}_{t_j}^{(k)}(y)$ is defined as

$$\mathbf{b}_{t_j}^{(k)}(y) = e^{t_j^{(k)} \mathbf{D}_j} \mathbf{d}_1^j(y), \tag{4.25}$$

which is computed using the remaining absorption time $t_j^{(k)} \in (t_i^{(k)}, t_j^{(k)})$ of the PHD composition, given that the initial phase of PH_j is y. The backward value (4.25) is computed as the joint density of being in phase y of PH_j directly after time $t_i^{(k)}$ and the probability of getting absorbed at time $t_j^{(k)}$. The corresponding vectors of the kth observation are denoted as the forward vector $\mathbf{f}_{t_i}^{(k)}$ and the backward vector $\mathbf{b}_{t_i}^{(k)}$. The expressions (4.24), (4.25) can be computed using the uniformization method described in Sec. 2.1.2.

Let $\beta(n, \alpha t)$ and matrices $\mathbf{P}_i$, $\mathbf{P}_j$ be parameters computed using the uniformization method (2.11), (2.13). Now we can use the forward-backward procedure to compute the transition likelihoods between states of PH_i and PH_j in composition. The normalized likelihoods are then used as estimates for the transition rates in the transfer matrix $\mathbf{H}_{ij}$.

According to the Poisson process we define the following vectors

$$\begin{aligned} \mathbf{v}^{(0)} &= \boldsymbol{\pi}_i \text{ and} \quad & \mathbf{v}^{(n+1)} &= \mathbf{v}^{(n)}\mathbf{P}_i \\ \mathbf{w}^{(0)} &= \mathbf{d}_1^j \text{ and} \quad & \mathbf{w}^{(n+1)} &= \mathbf{P}_j\mathbf{w}^{(n)} \end{aligned} \tag{4.26}$$

for $n = 0, 1, \ldots$. Then forward and backward vectors can be computed as

$$\begin{aligned} \mathbf{f}_{t_i}^{(k)} &= \sum_{n=0}^{\infty} \beta(n, \alpha t_i^{(k)})\, \mathbf{v}^{(n)}, \\ \mathbf{b}_{t_j}^{(k)} &= \sum_{m=0}^{\infty} \beta(m, \alpha t_j^{(k)})\, \mathbf{w}^{(m)}, \end{aligned} \tag{4.27}$$

where the lower truncation points $l_i, l_j \geq 0$ and the upper truncation points $r_i, r_j < \infty$ of the infinite sum can be pre-computed such that the required error tolerance ε is satisfied [162]. Now observe that the value

$$\mathbf{X}^{(k)}(x,y) = Pr(X_i(t_i^{(k)}) = x, X_j(t_j^{(k)}) = y \,|\, (t_i^{(k)}, t_j^{(k)}) \in \mathscr{T}_{ij},\ (\boldsymbol{\pi}_i, \mathbf{D}_i),\ (\boldsymbol{\pi}_j, \mathbf{D}_j),\ \mathbf{H}_{ij}))$$

gives the likelihood of being in phase x of PH_i at time $t_i^{(k)}$ and phase y of PH_j at time $t_j^{(k)}$, given the PHD composition model and the observed value pair $(t_i^{(k)}, t_j^{(k)})$. The above transition likelihoods can be estimated elementwise using forward and backward vectors as

$$\mathbf{X}^{(k)}(x,y) = \mathbf{f}_{t_i}^{(k)}(x)\mathbf{H}_{ij}(x,y)\mathbf{b}_{t_j}^{(k)}(y), \tag{4.28}$$

for $1 \leq x \leq n_i$, $1 \leq y \leq n_j$, and $(t_i^{(k)}, t_j^{(k)}) \in \mathscr{T}_{ij}$. Summing over all pairs in the trace $\mathscr{T}_{ij}$ we obtain the expected value for transition likelihoods, such that

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{X}^{(k)}. \tag{4.29}$$

The matrix $\mathbf{Y}$ should be normalized to have the same row sum as the matrix $\mathbf{H}_{ij}$

$$\hat{\mathbf{Y}}(x,y) = \frac{\sum_{z=1}^{n_j} \mathbf{H}_{ij}(x,z)}{\sum_{z=1}^{n_j} \mathbf{Y}(x,z)} \mathbf{Y}(x,y), \tag{4.30}$$

which can be used as a new estimate for $\mathbf{H}_{ij}(x,y)$. The initialization of the EM algorithm for the transfer matrix parameter fitting should be performed with a valid matrix $\mathbf{H}_{ij}$ satisfying conditions $\mathbf{H}_{ij} \geq \mathbf{0}$, $\mathbf{H}_{ij}\mathbb{1} = (-\mathbf{D}_i)\mathbb{1}$, $\pi_i \mathbf{M}_i \mathbf{H}_{ij} = \pi_j$. E.g., the initial transfer matrix $\mathbf{H}_{ij} = (-\mathbf{D}_i)\mathbb{1}\pi_j$ can be used, which describes uncorrelated absorption times of PH_i and PH_j in composition.

Additionally, the initial transfer matrix $\mathbf{H}_{ij}$ can be computed using parameter fitting methods according to empirical moments and correlation coefficient of the trace (see Sec. 4.2). Finally, one can use methods described in Chap. 6 to compute the initial transfer matrix maximizing the first joint moment of the PHD composition.

According to [38] the elements $\mathbf{H}_{ij}(x,y)$ which are initialized with 0.0 or become 0.0 during the EM step will remain 0.0 because the elements $\mathbf{X}^{(k)}(x,y)$ computed in (4.28) will result in 0.0. This property enables us to perform computations with sparse matrices $\mathbf{H}_{ij}$ or with special structures within $\mathbf{H}_{ij}$. Then the fitted matrix $\mathbf{H}_{ij}$ resulting from this initialization preserves some predefined structure.

After steps (4.28)-(4.29) are computed, it holds that $\hat{\mathbf{Y}}\mathbb{1} = (-\mathbf{D}_i)\mathbb{1}$ and $\hat{\mathbf{Y}} \geq 0$. The condition $\pi_i \mathbf{M}_i \hat{\mathbf{Y}} = \pi_j$ is usually not satisfied after one iteration of the EM algorithm. Thus, an additional optimization step is required to guarantee that the resulting transfer matrix assures the invariance of the initial distribution π_j within a composition.

Iterative EM Approach In the following we present a combination of the EM algorithm and a non-negative least squares approach to obtain the transfer matrix $\hat{\mathbf{Y}}'$ from $\hat{\mathbf{Y}}$, such that the condition $\pi_i \mathbf{M}_i \hat{\mathbf{Y}}' = \pi_j$ holds.

In particular, to preserve the initial distribution π_j the transfer matrix $\hat{\mathbf{Y}}$ has to be repaired after each iteration or after a few iterations of the EM algorithm. This can be achieved by solving the following non-negative least squares problem

$$\min_{\hat{\mathbf{Y}}' : \hat{\mathbf{Y}}' \geq \mathbf{0},\ \hat{\mathbf{Y}}'\mathbf{1} = \hat{\mathbf{Y}}\mathbf{1},\ \pi_i \mathbf{M}_i \hat{\mathbf{Y}}' = \pi_j} \left(\left\| \hat{\mathbf{Y}}' - \hat{\mathbf{Y}} \right\|^2 \right), \tag{4.31}$$

where $\hat{\mathbf{Y}}$ is the matrix from the EM iteration and $\hat{\mathbf{Y}}'$ is the repaired transfer matrix satisfying the initial distribution π_j. The defined optimization problem with equality constraints minimizes the entry-wise Frobenius norm $\left\| \hat{\mathbf{Y}}' - \hat{\mathbf{Y}} \right\|^2$. After solving the repair step (4.31) a new iteration of the EM algorithm using the repaired matrix $\hat{\mathbf{Y}}'$ is performed.

Since the EM algorithm is a local maximization algorithm generating a sequence of estimates $\hat{\mathbf{Y}}$ with a non-decreasing likelihood, the sequence may result in a local maximum which often depends on the initial transfer matrix. Generally, the final estimate $\hat{\mathbf{Y}}$ has a large likelihood, such that the repaired matrix $\hat{\mathbf{Y}}'$ will often result in a decreased likelihood value. In principle, starting with the repaired matrix $\hat{\mathbf{Y}}'$ could result in a different local maximum. The iterative EM approach using uniformization method is summarized in Algorithm 4.3. The proposed EM algorithm uses the set $\mathscr{T}_{ij} = ((t_i^{(1)}, t_j^{(1)}), \ldots, (t_i^{(K)}, t_j^{(K)}))$ and two PHDs with representation $(\pi_i, \mathbf{D}_i)$, $(\pi_j, \mathbf{D}_j)$ as an input.

Examples and Experiments In the following we perform series of experiments to validate the results from the EM approach. As mentioned above the EM algorithm converges to a local optimal solution and is not guaranteed to find a global optimum.

Algorithm 4.3: EM algorithm for computing transfer matrix $\mathbf{H}_{ij}$

1: Choose $\mathbf{H}_{ij}^{(0)}$ and set $r = 0$;
2: Compute $\mathbf{f}_{t_i}^{(k)}$, $\mathbf{b}_{t_j}^{(k)}$ using $(\pi_i, \mathbf{D}_i)$, $(\pi_j, \mathbf{D}_j)$ in Eq. 4.27 for $k = 1, \ldots, K$;
3: **repeat**
4: Set $\mathbf{H}_0^{(r)} = \mathbf{H}_{ij}^{(r)} / \alpha$ and compute $\mathbf{X}^{(k)}$ using $\mathbf{H}_0^{(r)}$ in Eq. 4.28 for $k = 1, \ldots, K$;
5: **E step**: Compute the conditional expectation $\mathbf{Y}^{(r)}$ using Eq. 4.29;
6: **M step**: Compute $\hat{\mathbf{Y}}^{(r)}$ using Eq. 4.30;
7: **Repair step**: Set $\hat{\mathbf{Y}}^{(r)} = \alpha \hat{\mathbf{Y}}^{(r)}$. Compute $\hat{\mathbf{Y}}'$ using $\hat{\mathbf{Y}}^{(r)}$ in Eq. 4.31 and set $\mathbf{H}_{ij}^{(r+1)} = \hat{\mathbf{Y}}'$, $r = r + 1$;
8: **until** $\|\mathbf{H}_{ij}^{(r)} - \mathbf{H}_{ij}^{(r-1)}\| < \varepsilon$;
9: Return $\mathbf{H}_{ij}^{(r)}$;

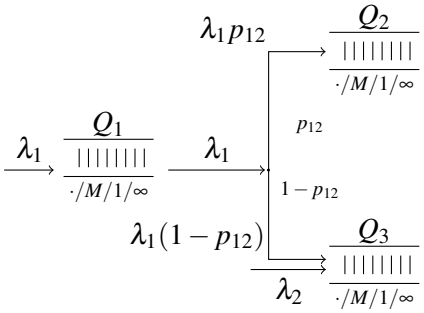


Figure 4.2.: The network of $\cdot/M/1/\infty$ queues.

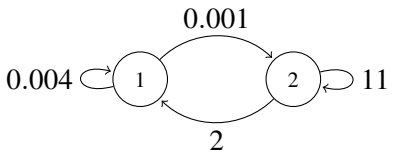
Often the quality of resulting transfer matrix depends on the initial transfer matrix $\mathbf{H}_{ij}^{(0)}$. The same holds for the number of iterations which also depends on a number of repair steps and is usually sensitive to the choice of randomization parameter α and the stopping criterion [38].

We consider a simple queueing network simulation model to obtain correlated data. From the data PH distributions are generated using the software tools *gfit* [168] and *Momfit* [44]. Traces resulting from measurements and corresponding PH distributions are used as input for the EM algorithm. For the trace containing about 10^4 samples the runtime of the iterative EM approach for $\varepsilon = 10^{-5}$ is in range from 5 to 25 seconds on a standard PC.

To demonstrate the EM fitting procedure for transfer matrices we use a queueing model [40] visualized in Fig. 4.1.3. The queueing network model is parameterized as follows. It consists of three queues Q_1, Q_2, and Q_3. Arrivals to Q_1 are generated by a MMPP with two states (see Sec. 2.5). The MMPP used in the queueing network model is shown in Fig. 4.3.

Note that the matrix $\mathbf{D}_1$ is a diagonal event rate matrix with λ_i values on the diagonal. If the Markov process is in state i, events occur according to a Poisson process with rate λ_i.

The service time at Q_1 is exponentially distributed. After leaving Q_1 an entity enters with probability p_{12} the queue Q_2, and with probability $p_{13} = 1 - p_{12}$ the queue Q_3.



(a) A state transition diagram of a 2-state MMPP.

$$\mathbf{D}_0 = \begin{pmatrix} -0.005 & 0.001 \\ 2 & -13 \end{pmatrix},$$

$$\mathbf{D}_1 = \begin{pmatrix} 0.004 & 0 \\ 0 & 11 \end{pmatrix}$$

(b) The infinitesimal subgenerator matrix $\mathbf{D}_0$ and the event rate matrix $\mathbf{D}_1$ of the MMPP.

Figure 4.3.: The 2-state MMPP for the queueing network.

Service times at Q_2 and Q_3 are also exponentially distributed. The probability p_{12} is set to 0.99. The queue Q_3 receives an additional Poisson arrival stream with rate λ_2.

The visualized MMPP generating the arrivals of Q_1 has a high arrival rate in state 2 and a low arrival rate in state 1. Furthermore 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 significantly. Thus 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.

The PH-Graph corresponding to the queueing network model is shown in Fig. 4.1.3. The sojourn time of Q_1 corresponds to the weight of the edge i_1 in Fig. 4.1.3. Analo-

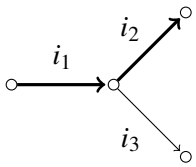


Figure 4.4.: The weights of the edges are modeled by queue residence times of Q_1, Q_2, and Q_3 in the open queue network in Fig. 4.1.3. The edges i_1, i_2 with correlated weights are highlighted.

gously, the sojourn times of Q_2 and Q_3 correspond to the weights of the adjacent edges i_2 and i_3, respectively.

In particular, the mean arrival rate of the MMPP is given by Eq. 2.94 and results in $\lambda_1 = 0.0095$. The service rate of Q_1 is $\mu_1 = 0.1429$, i.e., the service time is exponential with mean 7. The service rate of Q_2 is $\mu_2 = 0.0526$, i.e., the service time is exponential with mean 19. The service rate of Q_3 is $\mu_3 = 5$, i.e., the service time is exponential with mean 0.2, and $\lambda_2 = 0.1429$.

The proposed queueing network is implemented in the OMNeT++ simulator [92]. Then the measurements from the simulation model were recorded to the trace. From this trace we consider first $14,300$ samples for the fitting of PH distributions and transfer matrix.

First three sets $\mathscr{T}_{Q_1}$, $\mathscr{T}_{Q_2}$, and $\mathscr{T}_{Q_3}$ containing measured sojourn times in queues are extracted from the trace. Then, two sets $\mathscr{T}_{Q_1Q_2}$, $\mathscr{T}_{Q_1Q_3}$ containing all measured value

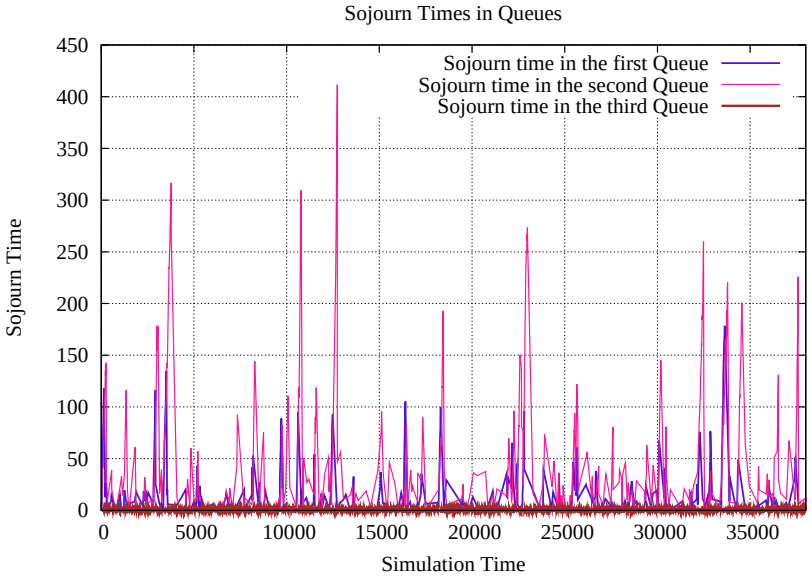


Figure 4.5.: Sojourn times of entities traveling through queues Q_1, Q_2, and Q_3.

pairs of consecutive queues are extracted.

Example* 4.1.** ***[Fitting of PHD composition] *Three traces $\mathscr{T}_{Q_1}$, $\mathscr{T}_{Q_2}$, and $\mathscr{T}_{Q_3}$ are used for fitting PH distributions of order* 4 *using the software tool gfit [168]. The sojourn time of entities traveling through Q_1 and Q_2 is correlated with a correlation coefficient $\hat{\rho} = 0.1345$ and the first joint moment $\hat{\mu} = 1.2142$. The sojourn times for each queue are summarized graphically in Fig. 4.5.*

One can see that the sojourn times in Q_1 and Q_2 have large peaks and the correlation between both sojourn times becomes visible. The sojourn time in Q_3 is less variable, such that no correlation with the sojourn time in Q_1 or Q_2 is visible. The hyperexponential PHD PH_{i_1} defined in (4.32) *describes the sojourn time distribution of queue Q_1. Whereas the hyper-Erlang PHD PH_{i_2} defined in* (4.33) *gives the sojourn time distribution of queue Q_2.*

$$\pi_{i_1} = (0.414, 0.195, 0.185, 0.204), \mathbf{D}_{i_1} = \begin{pmatrix} -0.657 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.721 & 0.000 & 0.000 \\ 0.000 & 0.000 & -3.429 & 0.000 \\ 0.000 & 0.000 & 0.000 & -4.717 \end{pmatrix}, \tag{4.32}$$

$$\pi_{i_2} = (0.5206, 0.3858, 0.0936, 0), \mathbf{D}_{i_2} = \begin{pmatrix} -0.724 & 0.000 & 0.000 & 0.000 \\ 0.000 & -3.534 & 0.000 & 0.000 \\ 0.000 & 0.000 & -1.085 & 1.085 \\ 0.000 & 0.000 & 0.000 & -1.085 \end{pmatrix}. \tag{4.33}$$

Here the maximal achievable first joint moment is $\mu^*_{1,1} = 1.34$*, maximal correlation coefficent is* $\rho^* = 0.21$ *and the corresponding transfer matrix*

$$\mathbf{H}^{\rho^*=0.21}_{i_1\,i_2} = \begin{pmatrix} 0.5088 & 0.0 & 0.1482 & 0.0 \\ 0.7215 & 0.0 & 0.0 & 0.0 \\ 0.0656 & 3.3635 & 0.0 & 0.0 \\ 0.0 & 4.7173 & 0.0 & 0.0 \end{pmatrix}.$$

The PHD PH_{i_3} *with* $\pi_{i_3} = (1)$ *and* $\mathbf{D}_{i_3} = (-0.9999)$ *models the sojourn time distribution of* Q_3 *which is exponential because* Q_3 *is a M/M/1 queue with a small additional load from* Q_1*.*

The PHD composition containing PH_{i_1} *and* PH_{i_2} *requires the transfer matrix* $\mathbf{H}_{i_1\,i_2}$ *to describe the correlation between the sojourn times along the edges* i_1 *and* i_2*. We used the trace* $\mathscr{T}_{Q_1\,Q_2}$ *as an input for the iterative EM algorithm 4.3. The matrix* $\hat{\mathbf{Y}}^{(r)}$ *resulting from the EM step and the approximation* $\hat{\mathbf{Y}}'$ *resulting from the repair step are given below*

$$\hat{\mathbf{Y}}^{(r)} = \begin{pmatrix} 0.499648 & 0 & 0.157417 & 0 \\ 0.335285 & 0.386282 & 0 & 0 \\ 0 & 3.429211 & 0 & 0 \\ 2.581962 & 2.135355 & 0 & 0 \end{pmatrix}, \hat{\mathbf{Y}}' = \begin{pmatrix} 0.508965 & 0 & 0.148109 & 0 \\ 0.348646 & 0.372924 & 0 & 0 \\ 0 & 3.429212 & 0 & 0 \\ 2.418297 & 2.299020 & 0 & 0 \end{pmatrix}$$

The matrix $\hat{\mathbf{Y}}^{(r)}$ *from the EM step results in the correlation coefficent* $\rho = 0.1296$ *and the first joint moment* 1.2099*. As expected, the condition* $\pi_{i_1}\mathbf{M}_{i_1}\hat{\mathbf{Y}}^{(r)} = \pi_{i_2}$ *is not satisfied.*

The approximation $\hat{\mathbf{Y}}'$ *is then used as* $\mathbf{H}_{i_1\,i_2}$*, such that* $\rho = 0.1294$ *and the first joint moment* 1.2097 *which are very good approximations of the values estimated from the trace* $\mathscr{T}_{Q_1Q_2}$ *of the simulation model.*

Note that the sojourn times in Q_1 *and* Q_3 *are uncorrelated. Thus the fitting of the transfer matrix* $\mathbf{H}_{i_1\,i_3}$ *is not required to obtain the PHD composition containing* PH_{i_1} *and* PH_{i_3}*. The uncorrelated transfer matrix can be obtained as* $\mathbf{H}_{i_1\,i_3} = \mathbf{d}_1^{i_1}\,\pi_{i_3}$*.*

***Example* 4.2 (Fitting of Transfer Matrix).** *We use the hyperexponential PHD* PH_{i_1} *defined in* (4.32) *as input. The PHD* PH_{i_2} *in canonical representation is obtained using software tool Momfit [44] and is defined as*

$$\pi'_{i_2} = (0.0483, 0.2650, 0.2358, 0.4507),\ \mathbf{D}'_{i_2} = \begin{pmatrix} -0.8126 & 0.8126 & 0.000 & 0.000 \\ 0.000 & -0.8629 & 0.8629 & 0.000 \\ 0.000 & 0.000 & -1.7067 & 1.7067 \\ 0.000 & 0.000 & 0.000 & -3.8052 \end{pmatrix}. \tag{4.34}$$

The maximal achievable first joint moment is $\mu^*_{1,1} = 1.385$*, maximal correlation coefficent is* $\rho^* = 0.2339$ *and the corresponding transfer matrix*

$$\mathbf{H}^{\rho^*=0.2339}_{i_1\,i_2} = \begin{pmatrix} 0.0765 & 0.4198 & 0.1606 & 0.0 \\ 0.0 & 0.0 & 0.4953 & 0.2261 \\ 0.0 & 0.0 & 0.0 & 3.4292 \\ 0.0 & 0.0 & 0.0 & 4.7173 \end{pmatrix}.$$

We computed the estimate of the transfer matrix using three different initial matrices $\mathbf{H}^{(0)}_{i_1\,i_2}$*, namely the zero correlation matrix computed as* $\mathbf{H}^{\rho=0}_{i_1\,i_2} = \mathbf{d}_1^{i_1}\,\pi'_{i_2}$*, the maximal correlation transfer matrix* $\mathbf{H}^{\rho^*=0.2339}_{i_1\,i_2}$*, and the matrix* $\mathbf{H}^{\rho=0.1129}_{i_1\,i_2}$ *which is computed using a fitting method according to empirical moments of the trace (see Sec. 4.2).*

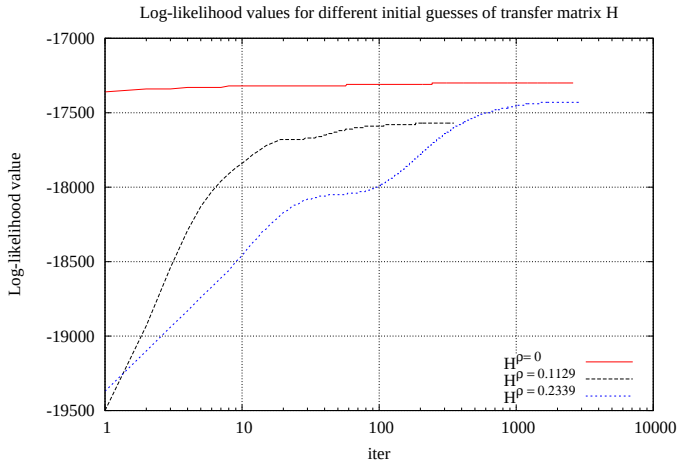


Figure 4.6.: Log-likelihood values for different initial transfer matrices.

	Queueing network trace			
	Trace	$\mathbf{H}_{i_1 i_2}^{\rho=0}$	$\mathbf{H}_{i_1 i_2}^{\rho=0.1129}$	$\mathbf{H}_{i_1 i_2}^{\rho^*=0.2339}$
$\rho_{i_1 i_2}$	0.1345	0.0806	0.0622	0.1070
$\mu_{1,1}^{i_1 i_2}$	1.2142	1.1375	1.1079	1.1802
$\mu_{2,2}^{i_1 i_2}$	11.5112	8.927	8.4685	9.0546
$\mu_{3,3}^{i_1 i_2}$	233.7016	167.69	154.5642	160.5274
log-likelihood		$-1.730e+04$	$-1.757e+04$	$-1.743e+04$

Table 4.1.: Moments and log-likelihood values for the queueing network simulation trace for different initial transfer matrices.

Figure 4.6 shows log-likelihood value curves for different initial transfer matrices. One can see the impact of initial transfer matrix on initial log-likelihood value and on the convergence speed of the EM algorithm.

Table 4.1 contains the correlation coefficent, the first three joint moments of the trace and log-likelihood values of the fitted transfer matrices when the EM algorithm is initialized with three different transfer matrices. The repair step has been performed ones for the final estimate.

Starting the EM algorithm with the initial matrices $\mathbf{H}_{i_1 i_2}^{(0)} = \mathbf{H}_{i_1 i_2}^{\rho=0}$ *and* $\mathbf{H}_{i_1 i_2}^{(0)} = \mathbf{H}_{i_1 i_2}^{\rho=0.1129}$ *no significant correlation can be reached as shown in Tab. 4.1. However, the initial transfer matrix* $\mathbf{H}_{i_1 i_2}^{(0)} = \mathbf{H}_{i_1 i_2}^{\rho^*=0.2339}$ *results in good approximations for the quantities of the trace.*

The results differ if different frequencies of repair steps are used within the iterative EM approach. First, the EM algorithm is initialized with $\mathbf{H}_{i_1 i_2}^{\rho=0}$ *and a different number of repair steps is performed. Figure 4.7 shows the log-likelihood values resulting*

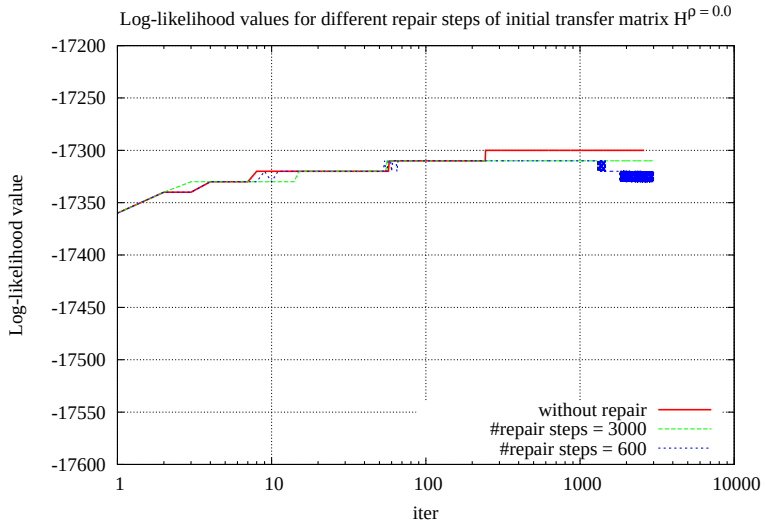


Figure 4.7.: Log-likelihood values for different number of repair steps within iterative EM approach for $\mathbf{H}^{(0)}_{i_1\,i_2} = \mathbf{H}^{\rho=0}_{i_1\,i_2}$.

	Queueing network trace			
	Trace	no repair steps	3000 repair steps	600 repair steps
$\rho_{i_1\,i_2}$	0.1345	0.0806	0.0798	0.1119
$\mu^{i_1\,i_2}_{1,1}$	1.2142	1.1375	1.1363	1.1882
$\mu^{i_1\,i_2}_{2,2}$	11.5112	8.927	8.7957	8.9916
$\mu^{i_1\,i_2}_{3,3}$	233.7016	167.69	163.3031	162.6987
log-likelihood		$-1.730e+04$	$-1.731e+04$	$-1.732e+04$

Table 4.2.: Moments and log-likelihood values for the queueing network simulation trace for different number of repair steps using initial transfer matrix $\mathbf{H}^{\rho=0}_{i_1\,i_2}$.

for different frequencies of repair steps. One can see that different number of repair steps result in different local maxima. Furthermore, the repaired transfer matrix often results in a decreased likelihood value. Nevertheless, the estimate with a smaller log-likelihood value can result in a better approximation of the correlation coefficent and the moments from the trace. The results according to the quantities of the trace are summarized in Table 4.2.

We also performed experiments using the initial transfer matrix with high correlation coefficent

$$\mathbf{H}^{\rho=0.2245}_{i_1\,i_2} = \begin{pmatrix} 0.0765 & 0.4081 & 0.1722 & 0.0 \\ 0.0 & 0.0 & 0.4682 & 0.2532 \\ 0.0 & 0.1362 & 0.0 & 3.2929 \\ 0.0 & 0.0 & 0.0 & 4.7173 \end{pmatrix},$$

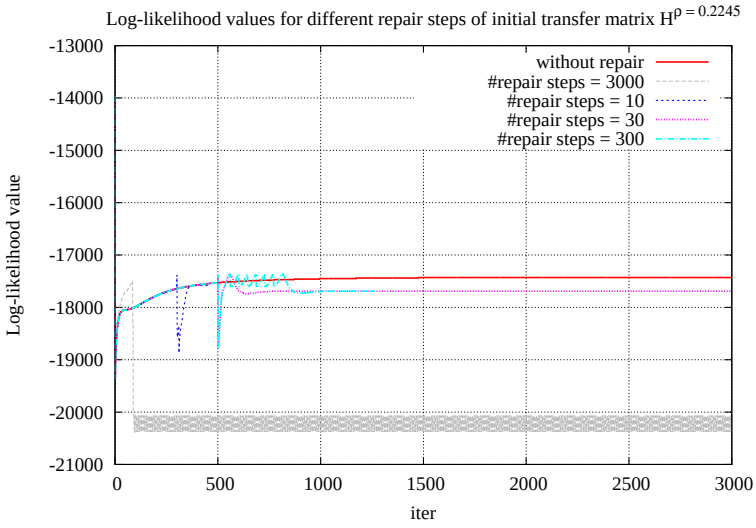


Figure 4.8.: Log-likelihood values for different number of repair steps within iterative EM approach for $\mathbf{H}^{(0)}_{i_1\,i_2} = \mathbf{H}^{\rho=0.2245}_{i_1\,i_2}$.

	Queueing network trace number of repair steps				
	Trace	0	10	300	3000
$\rho_{i_1\,i_2}$	0.1345	0.1070	−0.0481	0.1741	0.0389
$\mu^{i_1\,i_2}_{1,1}$	1.2142	1.1802	0.9295	1.2886	1.0702
$\mu^{i_1\,i_2}_{2,2}$	11.5112	9.0546	6.2133	9.9988	7.8254
$\mu^{i_1\,i_2}_{3,3}$	233.7016	160.5274	114.5717	176.8562	137.0110
log-likelihood		$-1.743e+04$	$-1.758e+04$	$-1.769e+04$	$-2.038e+04$

Table 4.3.: Moments and log-likelihood values for the queueing network simulation trace for different number of repair steps using initial transfer matrix $\mathbf{H}^{\rho=0.2245}_{i_1\,i_2}$.

which has been obtained using two phase fitting approach and results in $\rho_{i_1\,i_2} = 0.2245$. *Figure 4.8 shows the impact of different frequencies of repair steps for* $\mathbf{H}^{(0)}_{i_1\,i_2} = \mathbf{H}^{\rho=0.2245}_{i_1\,i_2}$. *The results according to the quantities of the trace are summarized in Table 4.3. One can see that larger number of repair steps not necessarily results in a better log-likelihood value.*

For example, the transfer matrix which has been repaired in each iteration results in the smallest log-likelihood value. Although the number of iterations is high, only unsatisfactory approximation of correlation from the trace can be obtained. In contrast, already 30 *repair steps could be an adequate choice to obtain a good approximation of the correlation coefficent and higher order moments.*

Since the proposed iterative EM approach is a heuristic the number of repair steps resulting in the optimal solution cannot be determined exactly. However, a good strategy is to start with repair steps after few EM iterations. Then, repairing a transfer matrix can result in a different local optimum.

EM Algorithm for Extended Composition The PHD composition with representation $((\pi_i, \mathbf{D}_i),\ (\pi_j, \mathbf{D}_j),\ \mathbf{H}_{ij})$ can be extended when additional correlation between PH_j, PH_i should be added. The composition process starts in some initial state x with probability $\pi_i(x)$. Then the process moves along the paths of the first Markov chain $\{X_i(t)\}_{t\geq 0}^{\infty}$ until some initial state of the second Markov chain $\{X_j(t)\}_{t\geq 0}^{\infty}$ is reached. Afterwards the composition restarts in some initial state of the first Markov chain $\{X_i(t)\}_{t\geq 0}^{\infty}$ and the process iterates. Thus the transfer matrix $\mathbf{H}_{ji}$ describes transitions corresponding to the occurence of an event in the second Markov process. The extended composition can then be defined as

$$((\pi_i, \mathbf{D}_i),\ (\pi_j, \mathbf{D}_j),\ \mathbf{H}_{ij},\ \mathbf{H}_{ji}). \tag{4.35}$$

In that case different traces could be used to compute parameters of the PHD composition. The trace $\mathscr{T}$ is described by a sequence of m chronologically ordered points in time $(t_i^1,\ t_j^2,\ t_i^3,\ t_j^4, \ldots, t_i^{m-1},\ t_j^m)$, such that the pairs $(t_i^k,\ t_j^{k+1})$ and $(t_j^k,\ t_i^{k+1})$ are correlated for $k = 1, \ldots, m-1$. From the above trace the measurements $w_n^{ij} = (\ (i,t_i)_n, (j,t_j)_n\ )$ and $w_n^{ji} = (\ (j,t_j)_n, (i,t_i)_n\ )$ can be constructed for $n = 1, \ldots, \lfloor m/2 \rfloor$ (cf. Sec. 4.1.1). Then the trace $\mathscr{T}_{ij}$ contains value pairs (t_i, t_j) which appear in w_n^{ij}, and the trace $\mathscr{T}_{ji}$ contains value pairs (t_j, t_i) from w_n^{ji}.

For the computation of the parameters for the transfer matrix $\mathbf{H}_{ij}$ the Algorithm 4.3 with the trace $\mathscr{T}_{ij}$ can be used as described in Sec. 4.1.3. The parameters of the second transfer matrix $\mathbf{H}_{ji}$ can be also determined using the EM-algorithm 4.3 with the trace $\mathscr{T}_{ji}$ and by reversing the order of PHDs, such that $((\pi_j, \mathbf{D}_j),\ (\pi_i, \mathbf{D}_i),\ \mathbf{H}_{ji})$ is the resulting PHD composition. Note that in this case the likelihood function is computed for every trace entry separately as defined in (4.22).

It is in principle possible to interpret the extended composition as a MAP with representation

$$\mathbf{D}_0 = \begin{bmatrix} \mathbf{D}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_j \end{bmatrix},\ \mathbf{D}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{H}_{ij} \\ \mathbf{H}_{ji} & \mathbf{0} \end{bmatrix}.$$

In that case the two phase iterative EM algorithm for MAPs (see [47, Sec. 5.3.3]) can be used to compute parameters of the above matrix $\mathbf{D}_1$. The parameters $\mathbf{D}_i$, $\mathbf{D}_j$ are assumed to be precomputed from the previous PH fitting step, such that only elements of $\mathbf{D}_1$, i.e. of $\mathbf{H}_{ij}$ and $\mathbf{H}_{ji}$ are determined.

Observe that the likelihood function of the extended composition can be defined considering the whole trace

$$\mathscr{L}(\mathbf{H}_{ij}, \mathbf{H}_{ji} \,|\, \mathscr{T}) = \pi_i \prod_{k=1}^{m-1} e^{t_i^k \mathbf{D}_i} \mathbf{H}_{ij}\, e^{t_j^{k+1} \mathbf{D}_j} \mathbf{H}_{ji}. \tag{4.36}$$

For evaluating of (4.36) the forward-backward procedure defined in EM algorithm for MAPs (see [47, Sec. 5.2]) can be in principle used. However, in that case both conditions $\pi_i \mathbf{M}_i \mathbf{H}_{ij} = \pi_j$ and $\pi_j \mathbf{M}_j \mathbf{H}_{ji} = \pi_i$ will usually not hold and should be repaired

after each EM iteration. The further drawback is the high computational effort of the resulting fitting procedure. Like in the EM algorithm for MAPs forward vectors all have to be precomputed and stored. For very large traces this means a huge number of vectors have to be computed and a high amount of memory is needed to store the vectors.

4.2. Two Phase Approaches

Fitting methods directly using the complete trace often have a high complexity, since traces can be very large containing up to a million or more entries. In turn, fitting algorithms based on derived measures are usually much more efficient. Indeed they are limited to an information according to a concrete statistical measure, but avoid an increased computational complexity and instability. In the case when, e.g. parameters of a MAP should be computed, it can be fitted to a process given by its empirical density function of the inter-arrival time distribution and by the lag correlation of the trace.

In this section we consider two phase fitting approaches for determining the parameters of a transfer matrix. The methods are based on two phase fitting approaches for MAPs where first the parameters of a PHD with representation $(\pi, \mathbf{D}_0)$ are computed. Similarly to MAPs, in the first step the parameters of two PHDs in composition with representation $(\pi_i, \mathbf{D}_i)$, $(\pi_j, \mathbf{D}_j)$ are determined. Here various fitting methods for PHDs can be applied (see e.g. [10, 30, 93, 108, 168, 47]). Generally, PHD fitting algorithms can handle different input data. The approaches described in [10, 168, 108] fit PHDs to a trace. The algorithms from [30, 93] consider pdf, cdf, and the trace. The moments based fitting algorithm from [90] determines parameters of a PHD, such that up to three moments can be matched.

In the second phase of the MAP fitting approach, the parameters of the matrix $\mathbf{D}_1$ are determined, such that the inter-arrival time distribution of MAP remains unchanged. This requires that π is the stationary distribution of the resulting MAP with representation $(\mathbf{D}_0, \mathbf{D}_1)$. Furthermore the lag correlation function of the resulted MAP approximates the correlation of the inter-arrival times collected in the trace.

Similarly to the MAP fitting approach, in the second phase the matrix $\mathbf{H}_{ij}$ of PHD composition is constructed. Therefore the matrix $\mathbf{H}_{ij}$ should be parameterized in order to satisfy conditions given in Eq. 3.2, 3.3 Furthermore the correlation function of the resulting composition with representation $((\pi_i, \mathbf{D}_i), (\pi_j, \mathbf{D}_j), \mathbf{H}_{ij})$ approximates the correlation of the edge weights collected in the trace $\mathscr{T}_{ij}$.

In the following we first present two phase fitting algorithms for MAPs according to the joint moments and the autocorrelation which are briefly described in [45, 96, 47]. Then we describe the adapted two phase approach for fitting of a transfer matrix $\mathbf{H}$ according to the joint moments and the correlation coefficient.

4.2.1. Joint Moment Fitting of MAPs

For joint moment fitting of MAP $(\mathbf{D}_0, \mathbf{D}_1)$ matrices $\mathbf{D}_0$ and $\mathbf{D}_1$ are constructed in separate steps. First, the matrix $\mathbf{D}_0$ and the vector π are determined using algorithms for fitting PHDs.

In the second step, the matrix $\mathbf{D}_1$ is determined such that lag correlation function of the resulting MAP approximates the one of the trace. Therefore the following conditions have to be satisfied to ensure that the inter-arrival time distribution determined in the first step remains invariant, namely $\mathbf{D}_1(i,j) \geq 0$, $\mathbf{D}_1\mathbb{1} = -\mathbf{D}_0\mathbb{1}$, and $\pi(-\mathbf{D}_0)^{-1}\mathbf{D}_1 = \pi$ as described [45, 96].

We can formulate the linear system of equations considering these constraints [45, 96, 47]. Let $\mathbf{x}$ be the column vector of size n^2, which is composed by the columns of the matrix $\mathbf{D}_1$ as shown below:

$$\mathbf{D}_1 = \begin{bmatrix} \{\mathbf{D}_1\}_1 & \{\mathbf{D}_1\}_2 & \dots & \{\mathbf{D}_1\}_n \end{bmatrix} \text{ with } \mathbf{x} = \begin{bmatrix} \{\mathbf{D}_1\}_1 \\ \{\mathbf{D}_1\}_2 \\ \vdots \\ \{\mathbf{D}_1\}_n \end{bmatrix} \tag{4.37}$$

where $\{\mathbf{D}_1\}_i$ denotes the ith column of $\mathbf{D}_1$. The coefficient matrix $\mathscr{A}$ and the column vector $\mathbf{b}$ encode the necessary conditions as follows:

$$\underbrace{\begin{bmatrix} \boxed{I_{n\times n}} & \boxed{I_{n\times n}} & \cdots & \boxed{I_{n\times n}} \\ \boxed{\psi} & & & \\ & \boxed{\psi} & & \\ & & \ddots & \\ & & & \boxed{\psi} \end{bmatrix}}_{\mathscr{A}_{2n\times n^2}} \cdot \underbrace{\begin{bmatrix} \\ \mathbf{x} \\ \\ \end{bmatrix}}_{\mathbf{x}_{n^2}} = \underbrace{\begin{bmatrix} \\ \mathbf{d}_1 \\ \\ \pi \end{bmatrix}}_{\mathbf{b}_{2n}}, \tag{4.38}$$

where $\psi = \pi\mathbf{M}$, such that the first n rows of $\mathscr{A}$ correspond to the condition given in Eq. 2.87, and the other rows correspond to the second condition $\pi(-\mathbf{D}_0)^{-1}\mathbf{D}_1 = \pi$.

A proper matrix $\mathbf{D}_1$, i.e. the corresponding vector $\mathbf{x}$, satisfying necessary constraints is the solution of the following system of linear equations and inequalities

$$\mathscr{A}\mathbf{x} = \mathbf{b}, \qquad \mathbf{x} \geq 0. \tag{4.39}$$

Observe that the $\mathscr{A}\mathbf{x} = \mathbf{b}$ is determined for $n = 2$ and under-determined for $n \geq 3$ [45]. In the last case, e.g. the simplex algorithm can be used to solve the problem (4.39).

Non-Negative Least Squares Problem Assume that some joint moments $\hat{\mu}_{ij}$ from the set of measured joint moments $\mathscr{J}$ should be matched by the matrix $\mathbf{D}_1$. In this case the fitting problem can be written as the minimization problem [96, 47]:

$$\min_{\mathbf{D}_1(i,j)\geq 0, \mathbf{D}_1\mathbf{1}=-\mathbf{D}_0\mathbf{1}, \pi\mathbf{M}\mathbf{D}_1=\pi} \left(\sum_{\hat{\mu}_{ij}\in\mathscr{J}} \left(\beta_{ij}\frac{\mu_{ij}}{\hat{\mu}_{ij}} - \beta_{ij} \right)^2 \right). \tag{4.40}$$

In Eq. 4.40 $(\pi, \mathbf{D}_0)$ is a representation of a PHD that is expanded to a MAP representation $(\mathbf{D}_0, \mathbf{D}_1)$ with i,j-th order joint moments μ_{ij}. The set $\mathscr{J}$ contains joint moments which have to be approximated. The non-negative coefficients β_{ij} can be used to privilege some of the moments.

The exact solution of the Eq. 4.40 is found if the minimum becomes zero. Otherwise an approximation in terms of the Euclidean norm is determined.

The i,j-th order joint moment for two consecutive events of a MAP is given by

$$\mu_{ij} = E[X_1^i, X_2^j] = i!\,j!\,\pi\mathbf{M}^{i+1}\mathbf{D}_1\mathbf{M}^j\mathbb{1}. \tag{4.41}$$

Letting $\mathbf{x}^i = \pi\mathbf{M}^{i+1}$ and $\mathbf{y}^j = \mathbf{M}^j\mathbb{1}$ the i,j-th order joint moment can be expressed as a linear constraint

$$\mu_{ij} = \sum_{k=1}^{n}\sum_{m=1}^{n} \mathbf{x}^i(k)\mathbf{D}_1(k,m)\mathbf{y}^j(m). \tag{4.42}$$

Assume we are interested in the first order joint moment

$$\mu_{11} = E[X_t, X_{t+1}] = \pi(-\mathbf{D}_0)^{-1}\mathbf{P}_s(-\mathbf{D}_0)^{-1}\mathbb{1}.$$

The splitted vectors result in $\mathbf{x}^1 = \pi(-\mathbf{D}_0)^{-1}(-\mathbf{D}_0)^{-1}$, $\mathbf{y}^1 = (-\mathbf{D}_0)^{-1}\mathbb{1}$. The resulting linear condition (4.42) can be concatenated to the matrix $\mathscr{A}$ and vector $\mathbf{b}$ as shown below [96, 47].

$$\underbrace{\begin{bmatrix} \boxed{\begin{bmatrix} I_{n\times n} \\ \psi \end{bmatrix}} & \boxed{\begin{bmatrix} I_{n\times n} \\ \psi \end{bmatrix}} & \cdots & \boxed{\begin{bmatrix} I_{n\times n} \\ \psi \end{bmatrix}} \\ & & \ddots & \\ \boxed{\mathbf{x}^1(1)\cdot\mathbf{y}^1(1)} & \boxed{\mathbf{x}^1(2)\cdot\mathbf{y}^1(1)} & \ldots & \boxed{\mathbf{x}^1(n)\cdot\mathbf{y}^1(n)} \end{bmatrix}}_{\mathscr{A}_{(2n+1)\times n^2}} \cdot \underbrace{\begin{bmatrix} \\ \mathbf{x} \\ \\ \end{bmatrix}}_{\mathbf{x}_{n^2}} = \underbrace{\begin{bmatrix} \\ \mathbf{d} \\ \\ \pi \\ \\ \hat{\mu}_{11} \end{bmatrix}}_{\mathbf{b}_{2n+1}}. \tag{4.43}$$

Here the coefficient β_{11} is set to one. Observe that (4.43) is a non-negative least squares problem with n^2 variables and $2n$ linear constraints which can be solved with standard algorithms for non-negative least squares problems [47, 116].

4.2.2. Joint Moment Fitting of Transfer Matrices

We adapted the described fitting method for MAPs from Sec. 4.2.1 to compute the parameters of the transfer matrix $\mathbf{H}$. In the second fitting step, we start with two PHDs $(\pi_i,\mathbf{D}_i)$, $(\pi_j,\mathbf{D}_j)$ that are the part of composition $((\pi_i,\mathbf{D}_i),\ (\pi_j,\mathbf{D}_j),\ \mathbf{H}_{ij})$. The matrix $\mathbf{H}_{ij}$ has to satisfy the following two conditions to maintain the PHDs determined in the first step, namely $\mathbf{H}_{ij}\mathbb{1} = -\mathbf{D}_i\mathbb{1}$, and $\pi_i(-\mathbf{D}_i)^{-1}\mathbf{H}_{ij} = \pi_j$ as given in (3.2), (3.3).

We can formulate the constraints as a linear system of equations. Since the transfer matrix is of dimension $n_i \times n_j$, the vector $\mathbf{x}$ is a column vector of size $n_i n_j$, which is composed similarly to (4.37)

$$\mathbf{H}_{ij} = \begin{bmatrix} \{\mathbf{H}_{ij}\}_1 & \{\mathbf{H}_{ij}\}_2 & \ldots & \{\mathbf{H}_{ij}\}_n \end{bmatrix} \text{ with } \mathbf{x} = \begin{bmatrix} \{\mathbf{H}_{ij}\}_1 \\ \{\mathbf{H}_{ij}\}_2 \\ \vdots \\ \{\mathbf{H}_{ij}\}_n \end{bmatrix}$$

where $\{\mathbf{H}_{ij}\}_i$ denotes the ith column of $\mathbf{H}$. The coefficient matrix $\mathscr{A}$ and the vector $\mathbf{b}$ can be rewritten as

$$\underbrace{\begin{bmatrix} \begin{bmatrix} I_{n_i\times n_i} \\ \boxed{\psi_i} \end{bmatrix} & \begin{bmatrix} I_{n_i\times n_i} \\ \boxed{\psi_i} \end{bmatrix} & \cdots & \begin{bmatrix} I_{n_i\times n_i} \\ \boxed{\psi_i} \end{bmatrix} \\ & & \ddots & \\ & & & \end{bmatrix}}_{\mathscr{A}_{(n_i+n_j)\times(n_in_j)}} \cdot \underbrace{\begin{bmatrix} \\ \mathbf{x} \\ \\ \end{bmatrix}}_{\mathbf{x}_{(n_in_j)}} = \underbrace{\begin{bmatrix} \\ \mathbf{d}_1^i \\ \\ \pi_j \end{bmatrix}}_{\mathbf{b}_{(n_i+n_j)}}, \tag{4.44}$$

where $\psi_i = \pi_i\mathbf{M}_i$ and $\mathbf{d}_1^i = (-\mathbf{D}_i)\mathbb{1}$. The first n_i rows of $\mathscr{A}$ correspond to the first constraint $\mathbf{H}_{ij}\mathbb{1} = -\mathbf{D}_i\mathbb{1}$, and the second n_j rows are added according to the condition $\pi_i\mathbf{M}_i\mathbf{H}_{ij} = \pi_j$. A proper $\mathbf{H}$ matrix which can be obtained from vector $\mathbf{x}$ satisfying (4.39).

Assume now that joint moments $\hat{\mu}_{ij}^{k,l}$ from the set of measured joint moments $\mathscr{J}_{ij}$ should be approximated by PHD composition containing $\mathbf{H}$. The fitting problem can be formulated as the optimization problem [96, 47]:

$$\min_{\mathbf{H}_{ij}(i,j)\geq 0,\, \mathbf{H1}=-\mathbf{H1},\, \pi_i\mathbf{M}_i\mathbf{H}_{ij}=\pi_j} \left(\sum_{\hat{\mu}_{ij}^{k,l}\in\mathscr{J}_{ij}} \left(\beta_{k,l}\frac{\mu_{ij}^{k,l}}{\hat{\mu}_{ij}^{k,l}} - \beta_{k,l} \right)^2 \right), \tag{4.45}$$

where $\mu_{ij}^{k,l}$ is the k,l order joint moment of PHD composition $((\pi_i,\mathbf{D}_i),\ (\pi_j,\mathbf{D}_j),\ \mathbf{H}_{ij})$ (see (3.6)). The joint moments of two consecutive absorption times in PHD composition can be derived using moment matrices $\mathbf{M}_i = (-\mathbf{D}_i)^{-1}$ and $\mathbf{M}_j = (-\mathbf{D}_j)^{-1}$ of the corresponding PHDs

$$\mu_{ij}^{k,l} = E[X_i^k, X_j^l] = k!l!\pi_i\mathbf{M}_i^{k+1}\mathbf{H}_{ij}\mathbf{M}_j^l\mathbb{1},$$

which can be expressed as a linear constraint

$$\mu_{ij}^{k,l} = \sum_{r=1}^{n_i}\sum_{m=1}^{n_j}\mathbf{x}^k(r)\mathbf{H}_{ij}(r,m)\mathbf{y}^l(m), \tag{4.46}$$

where $\mathbf{x}^k = \pi_i\mathbf{M}_i^{k+1}$ and $\mathbf{y}^l = \mathbf{M}_j^l\mathbb{1}$. The above expression for $\mu_{ij}^{k,l}$ (4.46) can be plugged in for $\mu_{ij}^{k,l}$ in the minimization problem (4.45).

We now consider how the first joint moment can be expressed as a linear constraint in detail. The first order joint moment of two consecutive absorption times in PHD composition $((\pi_i,\mathbf{D}_i),\ (\pi_j,\mathbf{D}_j),\ \mathbf{H}_{ij})$ is given by $\mu_{ij}^{1,1} = \pi_i\mathbf{M}_i^2\mathbf{H}_{ij}\mathbf{M}_j\mathbb{1}$. The splitted vectors are $\mathbf{x}^1 = \pi_i(-\mathbf{D}_i^{-1})(-\mathbf{D}_i^{-1})$ and $\mathbf{y}^1 = (-\mathbf{D}_j^{-1})\mathbb{1}$. Then the linear constraint (4.46) can be concatenated to the matrix $\mathscr{A}$ and the vector $\mathbf{b}$ from Eq. 4.44 as

follows

$$\underbrace{\begin{bmatrix} \boxed{\begin{bmatrix} I_{n_i \times n_i} \end{bmatrix}} & \boxed{\begin{bmatrix} I_{n_i \times n_i} \end{bmatrix}} & \cdots & \boxed{\begin{bmatrix} I_{n_i \times n_i} \end{bmatrix}} \\ \boxed{\psi_i} & & & \\ & \boxed{\psi_i} & & \\ & & \ddots & \\ & & & \boxed{\psi_i} \\ \boxed{\mathbf{x}^1(1) \cdot \mathbf{y}^1(1)} & \boxed{\mathbf{x}^1(2) \cdot \mathbf{y}^1(1)} & \ldots & \boxed{\mathbf{x}^1(n_i) \cdot \mathbf{y}^1(n_j)} \end{bmatrix}}_{\mathscr{A}_{(n_i+n_j+1)\times(n_i n_j)}} \cdot \underbrace{\begin{bmatrix} \mathbf{x} \end{bmatrix}}_{\mathbf{x}_{(n_i n_j)}} = \underbrace{\begin{bmatrix} \mathbf{d}_1^i \\ \pi_j \\ \hat{\mu}_{ij}^{1,1} \end{bmatrix}}_{\mathbf{b}_{(n_i+n_j+1)}}, \tag{4.47}$$

where the weight $\beta_{1,1}$ is set to one. The resulting problem (4.47) is a non-negative least squares problem with $n_i n_j$ unknowns and $n_i + n_j + 1$ constraints. The problem can be solved with standard algorithms for non-negative least squares problems [116].

***Example* 4.3.** *We consider again the trace $\mathscr{T}_{Q_1 Q_2}$ from the small queueing network model in example 4.1. The set $\mathscr{T}_{Q_1 Q_2}$ contains measured sojourn time pairs of consecutive queues. The estimated first joint moment of the trace is $\hat{\mu} = 1.2142$, and the sojourn time of entities traveling through Q_1 and Q_2 is correlated with correlation coefficent $\hat{\rho} = 0.1345$.*

Furthermore the sets $\mathscr{T}_{Q_1}$, $\mathscr{T}_{Q_2}$ contain the measured sojourn times in queues. Following PHDs have been fitted to the traces $\mathscr{T}_{Q_1}$, $\mathscr{T}_{Q_2}$

$$\pi_{i_1} = (0.414, 0.195, 0.185, 0.204), \mathbf{D}_{i_1} = \begin{pmatrix} -0.657 & 0.000 & 0.000 & 0.000 \\ 0.000 & -0.721 & 0.000 & 0.000 \\ 0.000 & 0.000 & -3.429 & 0.000 \\ 0.000 & 0.000 & 0.000 & -4.717 \end{pmatrix},$$

$$\pi_{i_2} = (0.5206, 0.3858, 0.0936, 0), \mathbf{D}_{i_2} = \begin{pmatrix} -0.724 & 0.000 & 0.000 & 0.000 \\ 0.000 & -3.534 & 0.000 & 0.000 \\ 0.000 & 0.000 & -1.085 & 1.085 \\ 0.000 & 0.000 & 0.000 & -1.085 \end{pmatrix}.$$

We formulate the following linear system of equations. The coefficient matrix $\mathscr{A}$ is of dimension 9×16 and results in

$$\begin{bmatrix} I_{4\times4} & I_{4\times4} & I_{4\times4} & I_{4\times4} \\ \psi_i & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \psi_i & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \psi_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \psi_i \\ \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 & \mathbf{z}_4 \end{bmatrix},$$

where the vector $\psi_{i_1} = \pi_{i_1} \mathbf{M}_{i_1} = [0.6313, 0.2714, 0.0540, 0.0433]$ and $\mathbf{0}$ is the 1×4 vector containing zeros.

The linear constraint according to the first joint moment corresponds to the last row of the matrix $\mathscr{A}$. In particular, the row vector $\mathbf{x}^1 = [0.9608, 0.3761, 0.0157, 0.0092]$

and the row vector $\mathbf{y}^1 = [1.3802, 0.2830, 1.8421, 0.9211]$*, such that*

$$\begin{aligned}\mathbf{z}_1 &= [1.326,\ 0.519,\ 0.021,\ 0.012],\\ \mathbf{z}_2 &= [0.271,\ 1.77,\ 0.692,\ 0.029],\\ \mathbf{z}_3 &= [1.77,\ 0.692,\ 0.029,\ 0.016],\\ \mathbf{z}_4 &= [0.885,\ 0.346,\ 0.014,\ 0.008].\end{aligned}$$

Furthermore, the vector

$$\mathbf{b} = [0.6571, 0.7216, 3.4292, 4.7173, 0.5206, 0.3859, 0.0935, 0, 1.2142]',$$

where the first 4 *values correspond to the exit rates of the PHD* $(\pi_{i_1}, \mathbf{D}_{i_1})$*, the next* 4 *values correspond to the* π_{i_2}*, and the last value is* $\hat{\mu} = 1.2142$*. The problem has been implemented in matlab where the function lsqnonneg has been used to obtain the transfer matrix* $\mathbf{H}_{i_1 i_2}$

$$\begin{pmatrix} 0.5090 & 0 & 0.1481 & 0\\ 0.3617 & 0.3599 & 0 & 0\\ 0 & 3.4292 & 0 & 0\\ 2.3363 & 2.3811 & 0 & 0 \end{pmatrix}.$$

The matrix $\mathbf{H}_{i_1 i_2}$ *results in the correlation coefficient* $\rho = 0.1322779$ *and the first joint moment* $\mu_{1,1} = 1.2142$ *which is a very good approximation of the* $\hat{\rho} = 0.1345$ *estimated from the trace* $\mathscr{T}_{Q_1 Q_2}$*.*

4.2.3. Autocorrelation Fitting of MAPs

In a two phase approach fitting of a given number of lag k correlations can be incorporated. In this section we present how this fitting method can be expressed as a linear constrained non-linear optimization problem from [96, 47]. In the second phase the given PHD with representation $(\pi, \mathbf{D}_0)$ should be expanded to a MAP $(\mathbf{D}_1, \mathbf{D}_0)$, such that the lag k autocorrelation coefficients of the MAP approximates lag k autocorrelation coefficients $\tilde{\rho}_k$, $k = 1, \ldots, K$ of some observed process.

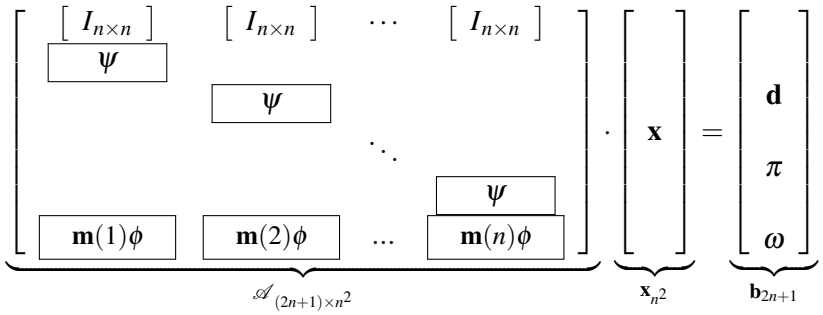
In the following we summarize an exact lag 1 correlation fitting problem described in [96]. First observe that the formula for lag 1 autocorrelation in Eq. 2.93 can be expressed in terms of arrivals intensity $\lambda = \frac{1}{\pi(-\mathbf{D}_0)^{-1}\mathbb{1}}$ as

$$\rho_1 = \frac{\lambda^2 \pi (-\mathbf{D}_0)^{-2} \mathbf{D}_1 (-\mathbf{D}_0)^{-1} \mathbb{1} - 1}{\lambda^2 2\pi (-\mathbf{D}_0)^{-2} \mathbb{1} - 1}.$$

Using $\mathbf{m} = (-\mathbf{D}_0)^{-1}\mathbb{1}$ the above term can be reformulated as

$$\lambda^2 \pi (-\mathbf{D}_0)^{-2} \mathbf{D}_1 \mathbf{m} = \rho_1 \left[\lambda^2 2\pi (-\mathbf{D}_0)^{-2} \mathbb{1} - 1\right] + 1,$$

which can be concatenated to the matrix $\mathscr{A}$ and vector $\mathbf{b}$ as



(4.48)

where $\phi = \lambda^2 \pi (-\mathbf{D}_0)^{-2}$, and $\mathbf{m}(i)$ is the i-th element of the vector $\mathbf{m}$. The right hand side is denoted as $\omega = \tilde{\rho}_1 \left[\lambda^2 2\pi (-\mathbf{D}_0)^{-2} \mathbb{1} - 1\right] + 1$. Observe that $\tilde{\rho}_1$ is the lag 1 autocorrelation coefficient which has to be approximated by the expanded MAP $(\mathbf{D}_0, \mathbf{D}_1)$, and can be often estimated from the trace.

In the case where more lag k autocorrelation values should be matched, the following optimization problem with linear constraints given in Eq. 4.39 can be defined

$$\min_{\mathbf{D}_1(i,j)\geq 0, \mathbf{D}_1\mathbf{1}=-\mathbf{D}_0\mathbf{1}, \pi(-\mathbf{D}_0)^{-1}\mathbf{D}_1=\pi} \left(\sum_{k=2}^{K} \beta_k (\rho_k - \hat{\rho}_k)^2 \right). \tag{4.49}$$

The problem (4.49) is the squared difference between lag k autocorrelation coefficients of the observed process and the fitted MAP. The lag K is the largest autocorrelation coefficient that should be matched, and weight β_k is used to privilege lower lag autocorrelations.

As mentioned in [96] higher higher lag autocorrelations result in non-linear constraints. For example the lag 2 first order joint moment

$$E[X_t, X_{t+2}] = \pi(-\mathbf{D}_0)^{-1}\mathbf{P}_s^2(-\mathbf{D}_0)^{-1}\mathbb{1}$$

would lead to a term containing squared elements of the matrix $\mathbf{D}_1$ [47].

4.2.4. Correlation Fitting of Transfer Matrices

Similarly to the autocorrelation fitting of MAPs described in Sec. 4.2.3, the exact correlation fitting of transfer matrix $\mathbf{H}_{ij}$ represents a linear constrained non-linear optimization problem.

We start with two PHDs with representation $(\pi_i, \mathbf{D}_i)$, $(\pi_j, \mathbf{D}_j)$ which are computed separately. In the second phase the matrix $\mathbf{H}_{ij}$ has to be constructed, such that the correlation function of a composition $((\pi_i, \mathbf{D}_i), (\pi_j, \mathbf{D}_j), \mathbf{H}_{ij})$ approximates the one of the trace. In particular, the correlation of a composition can be defined as a linear constraint. According to formula (3.8) we obtain

$$\rho_{ij} = \frac{\pi_i \mathbf{M}_i^2 \mathbf{H}_{ij} \mathbf{M}_j \mathbb{1} - [(\pi_i \mathbf{M}_i \mathbb{1}) \cdot (\pi_j \mathbf{M}_j \mathbb{1})]}{\sqrt{(2\pi_i (-\mathbf{D}_i)^{-2}\mathbb{1} - (\pi_i (-\mathbf{D}_i)^{-1}\mathbb{1})^2)} \cdot \sqrt{(2\pi_j (-\mathbf{D}_j)^{-2}\mathbb{1} - (\pi_j (-\mathbf{D}_j)^{-1}\mathbb{1})^2)}}$$

Let $\sigma_i^2 = (2\pi_i (\mathbf{M}_i)^2 \mathbb{1} - (\pi_i \mathbf{M}_i \mathbb{1})^2)$ and $\sigma_j^2 = (2\pi_j \mathbf{M}_j^2 \mathbb{1} - (\pi_j \mathbf{M}_j \mathbb{1})^2)$. Then the expression

$$\rho_{ij} \sqrt{\sigma_i^2 \, \sigma_j^2} = \pi_i \mathbf{M}_i^2 \mathbf{H}_{ij} \mathbf{M}_j \mathbb{1} - [(\pi_i \mathbf{M}_i \mathbb{1}) \cdot (\pi_j \mathbf{M}_j \mathbb{1})]$$

can be rewritten as a linear constraint

$$\phi \mathbf{H}_{ij} f = \rho_{ij} \sqrt{\sigma_i^2 \, \sigma_j^2} + [(\pi_i \mathbf{M}_i \mathbb{1}) \cdot (\pi_j \mathbf{M}_j \mathbb{1})], \tag{4.50}$$

where $f = (-\mathbf{D}_j)\mathbb{1}$, and $\phi = \pi_i(-\mathbf{D}_i)^{-1}(-\mathbf{D}_i)^{-1}\mathbb{1}$. Let ω denote the right hand side of Eq. 4.50 which can be concatenated to the matrix $\mathscr{A}$ and vector $\mathbf{b}$ as

$$\underbrace{\begin{bmatrix} I_{n_i \times n_i} & I_{n_i \times n_i} & \cdots & I_{n_i \times n_i} \\ \psi_i & & & \\ & \psi_i & & \\ & & \ddots & \\ & & & \psi_i \\ f(1)\phi & f(2)\phi & \ldots & f(n_j)\phi \end{bmatrix}}_{\mathscr{A}_{(n_i+n_j+1)\times(n_i n_j)}} \cdot \underbrace{\begin{bmatrix} \\ \mathbf{x} \\ \\ \end{bmatrix}}_{\mathbf{x}_{(n_i n_j)}} = \underbrace{\begin{bmatrix} \mathbf{d}_1^i \\ \\ \pi_j \\ \\ \omega \end{bmatrix}}_{\mathbf{b}_{(n_i+n_j+1)}}, \tag{4.51}$$

where $f(i)$ is the ith element of vector f. The ω contains the correlation coefficient $\hat{\rho}_{ij}$ from the original process which can be often estimated from the trace $\mathscr{T}_{ij}$. The problem (4.51) is a non-negative least squares problem with $n_i n_j$ variables and $n_i + n_j + 1$ linear constraints, and can be solved using standard algorithms for non-negative least squares problems [116].

If more lag k correlation values should be matched, the fitting problem is a linearly constrained non-linear optimization problem with linear constraints given in Eq. 4.44. Assume that lag k correlation coefficents $\hat{\rho}_{ij}^k$ should be matched, such that K is the largest lag correlation coefficient. The fitting problem can be formulated as the following minimization problem

$$\min_{\mathbf{H}_{ij}(i,j)\geq 0, \mathbf{H}_{ij}\mathbf{1} = -\mathbf{D}_i\mathbf{1}, \pi_i \mathbf{M}_i \mathbf{H}_{ij} = \pi_j} \left(\sum_{k=2}^{K} \beta_k \left(\rho_{ij}^k - \hat{\rho}_{ij}^k \right)^2 \right). \tag{4.52}$$

In the problem (4.52) the weights β_k can be used to priviledge either the lower lag k correlation or the higher lag k correlation coefficient.

***Example* 4.4.** *Analogously to example 4.3 we used the traces $\mathscr{T}_{Q_1}$, $\mathscr{T}_{Q_2}$, $\mathscr{T}_{Q_1Q_2}$, and PHDs $(\pi_{i_1}, \mathbf{D}_{i_1})$, $(\pi_{i_2}, \mathbf{D}_{i_2})$ to fit the transfer matrix $\mathbf{H}_{i_1 i_2}$. We solve the problem* (4.51) *using matlab function lsqnonneg. The resulting transfer matrix is*

$$\mathbf{H}_{i_1 i_2} = \begin{pmatrix} 0.5145 & 0.1426 & 0 & 0 \\ 0.7216 & 0 & 0 & 0 \\ 0 & 1.6964 & 1.7324 & 0.0003 \\ 0 & 4.7173 & 0 & 0 \end{pmatrix}.$$

The matrix $\mathbf{H}_{i_1 i_2}$ results in the correlation coefficient $\rho = 0.102288$ and the first joint moment $\mu_{1,1} = 1.165641$ which are both very good approximations of the values $\hat{\rho} = 0.1345$, and $\hat{\mu}_{1,1} = 1.2142$ estimated from the trace $\mathscr{T}_{Q_1Q_2}$.

4.3. Summary and Remarks

Prior work has shown the effectiveness of methods to fit the parameters of PHDs according to some measured data. For example, EM algorithms can be applied to fit PHDs efficiently and guarantee good fitting quality, even for complex empirical density functions. We refer to [47] for PHD fitting methods according to pdf, cdf and a given number of moments of trace. These and other existing fitting techniques form a solid basis for obtaining parameters of PHDs in composition. However, fitting the parameters of transfer matrices according to some paired trace data $\mathscr{T}_{ij}$ has not been focused in the context of PHG parameter fitting.

In this chapter we adapted the EM algorithm from the area of PHD and MAP parameter fitting. The approach is an extension of iterative approaches considering the whole trace data which means that the effort is proportional to the uniformization method over value pairs of the trace. Consequently, its effort can be very high when large traces are used, e.g., with up to 10^6 value pairs.

Furthermore, the EM algorithm requires an additional optimization since the quality of the resulting transfer matrix is usually not satisfactory with respect to PHD composition. Specifically, it is not guaranteed that the initial distribution of the successive PHD in a composition is valid when the resulting transfer matrix is used.

However, after repairing the transfer matrix the EM algorithm often stucks in local solution areas which is actually typical for expectation-maximization approaches. In particular, it is not yet clear which initial transfer matrix and which PHD representations are best suited for some given data set. The adapted EM algorithm is applied to an example trace resulting from the queueing network simulation model. Our results provide transfer matrices which capture the trace correlation very well.

In addition, two phase fitting approaches for MAPs are adapted. Two phase methods start with fitting of both PHDs in a composition and then compute parameters of the transfer matrix according to joint moments or correlation coefficients from the trace. Although the efficiency of these methods has been shown several times in literature (see [47] and references therein), the representation of two PHDs has large influence when fitting a transfer matrix. Note that PHD representation is non-unique. Entries in PHDs $(\pi_i, \mathbf{D}_i)$ and $(\pi_j, \mathbf{D}_j)$ put constraints on the representation of transfer matrix $\mathbf{H}_{ij}$, such that the range of the first joint moment that can be fitted is usually limited. We investigate this aspect in Chap. 6.

It should also be mentioned that estimation of correlation from a trace is a non trivial task. First, real world traces are often not available, such that using traces from simulation models seems to be indispensable. Furthermore, it is not yet clear which data set is correlated if only poor information about correlations is available.

Chapter 5

Solutions to SSPP with correlated edge weights

In this chapter we develop a CTMDP based on the PHG model that can capture correlations between adjacent edge weights in stochastic graphs. The PHG model aims at decoding information about existing correlations for the whole stochastic graph according to a real or simulated system under consideration, e.g., simulated routing graph. In this chapter we formulate problems arising in the SSPP context in terms of decision problems in CTMDPs. Different algorithms and their variants to compute optimal policies for CTMDPs are described.

First we consider the computation of a path with minimum expected weight starting in the origin and ending in the destination node. Then we add a solution technique considering a history of realized edge weights along the covered path.

The other challenging problem for SSPP with correlations is to compute the path that has the maximum/minimum probability of reaching a destination within a given time horizon. This results in the problem of computing the maximum/minimum probability to reach a set of goal states within a given time bound in CTMDPs which is a well studied problem [127, 49, 131, 42]. We present a numerical method based on discretization to compute and to approximate the maximum/minimum gain vector per state in a CTMDP for a finite interval $[0,T]$ which is an adaptation of the algorithm [127] to compute the accumulated reward in a CTMDP over a finite interval.

In the following the model concept including the CTMDP extension of the PHG model is introduced and it is shown that efficient solution techniques for the SSPP with correlations based on the developed model can be applied.

5.1. SSPP formulation of the PH-Graph model

In this section describe a SSPP formulation associated with a PH-Graph (V,E,P) in order to solve the *stochastic shortest path problem with correlations* based on [40]. Then the stochastic shortest path problem in stochastic graphs with dependent edge weights can be efficiently solved using algorithms for CTMDPs. In particular, we build an undiscounted CTMDP with a single absorbing state such that the optimal policy minimizes the expected total reward of reaching the destination v_{fin} from the origin v_{ini}.

The states of the CTMDP are described by tuples (i,x) where $i \in E$ and $x \in \{1,\dots,n_i\}$ is a phase of the PHD assigned to the edge i. The set of transient states can be formalized as

$$\mathscr{S}_T = \{(i,x) \mid \forall i \in E,\ x \in \{1,\dots,n_i\}\}. \tag{5.1}$$

The set of absorbing states contains a single absorbing state denoted as $(0,0)$ which corresponds to the destination v_{fin}, such that $\mathscr{S}_A = \{(0,0)\}$. Then the state space is defined as $\mathscr{S} = \mathscr{S}_T \cup \mathscr{S}_A$ and contains $\sum_{i=1}^{|E|} n_i + 1$ states.

In particular, for some state (i,x) the choice of the successor edge $j \in i\bullet$ is associated with an admissible decision u. Let $\mathscr{D}(i)$ be the set of possible decisions for a current state (i,x) which only depends on the edge i and not on the current phase of the PHD PH_i. Then

$$\mathscr{D}(i) = \begin{cases} \{j \mid j \in i\bullet\} & \text{if } i\bullet \neq \emptyset, \\ \{0\} & \text{if } i\bullet = \emptyset. \end{cases} \tag{5.2}$$

Let $\mathbf{Q}^u((i,x),(j,y))$ be the transition rate from the state (i,x) to the state (j,y) if the decision $u \in \mathscr{D}(i)$ is made. It holds that

$$\mathbf{Q}^u((i,x),(i,x)) = -\sum_{(j,y)\in\mathscr{S},(j,y)\neq(i,x)} \mathbf{Q}^u((i,x),(j,y)) \tag{5.3}$$

Then the infinitesimal subgenerator $\mathbf{Q}^u$ prescribes the transition rates for successor states as follows.

$$\mathbf{Q}^u((i,x),(j,y)) = \begin{cases} \mathbf{D}_i(x,y) & \text{if } j = i,\ i > 0 \\ \mathbf{H}_{iu}(x,y) & \text{if } j = u,\ u \in i\bullet,\ i > 0,\ u > 0 \\ \mathbf{d}_1^i(x) & \text{if } j = 0 \text{ and } y = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

The absorbing state $(0,0)$ has a single decision $u \in \mathscr{D}(0)$ such that the transition rates are $\mathbf{Q}^u((0,0),(j,y)) = 0$ for all $(j,y) \in \mathscr{S}$. Note that the absorbing state is reached when some edge $i \in E_{fin}$ has been traversed and thus can be interpreted as the destination v_{fin}.

Given the set of transition rates $\mathscr{Q} = \{\mathbf{Q}^u((i,x),(j,y)) \mid (i,x),(j,y) \in \mathscr{S}, u \in \mathscr{D}(i)\}$ the CTMDP described can be transformed into a DTMDP using the method of uniformization (see Sec. 2.1.2). The uniformization rate α is selected such that $\alpha \geq \max_{(i,x)\in\mathscr{S}} \left(\max_{u\in\mathscr{D}(i)} \left(|\mathbf{Q}^u((i,x),(i,x))|\right)\right)$ holds. The entries of the transition matrices of the embedded Markov processes are then defined as

$$\mathbf{P}^u((i,x),(j,y)) = \begin{cases} \mathbf{Q}^u((i,x),(j,y))/\alpha & \text{if } (i,x) \neq (j,y) \\ 1 + \mathbf{Q}^u((i,x),(j,y))/\alpha & \text{if } (i,x) = (j,y) \end{cases} \tag{5.5}$$

such that for each $u \in \mathscr{D}(i)$ the matrix $\mathbf{P}^u$ is stochastic.

The expected rewards for all states have the same unit value (cf. Eq. 5.22)

$$\mathbf{r}^u(i,x) = \begin{cases} 1 & \text{for all } (i,x) \in \mathscr{S}_T,\ u \in \mathscr{D}(i), \\ 0 & \text{for } (i,x) = (0,0). \end{cases} \tag{5.6}$$

Now consider a stationary policy $\mathbf{u}$ which assignes a decision rule $\mathbf{u}(i,x) \in \mathscr{D}(i)$ to each state $(i,x) \in \mathscr{S}$. Then the transition probability matrix $\mathbf{P^u}$ of dimension $|\mathscr{S}| \times |\mathscr{S}|$ is defined by (cf. Eq. 2.34)

$$\mathbf{P^u}((i,x),(j,y)) = \mathbf{P}^{\mathbf{u}(i,x)}((i,x),(j,y)). \tag{5.7}$$

Observe that in the case where the policy $\mathbf{u}$ is improper, the absorbing state $(0,0)$ can never be reached from some states, i.e. the total expected reward of that policy from some states will be infinite. In that case some component of the sum $\sum_{t=0}^{\infty}(\mathbf{P}^u)^t\mathbf{r}^{\mathbf{u}}$ diverges to ∞ as $t \to \infty$. On the other hand, for every proper policy $\mathbf{u}$ these rewards are expected to be finite for every state. In this case a policy $\mathbf{u}$ reaches an absorbing state with probability 1, i.e. a policy is guaranteed to eventually reach an absorbing state. Then the sum $\sum_{t=0}^{\infty}(\mathbf{P}^u)^t\mathbf{r}^{\mathbf{u}}$ is finite such that the following matrix exists

$$\mathbf{N}^{\mathbf{u}} = (\mathbf{I} - \mathbf{P}^{\mathbf{u}})^{-1} = \sum_{t=0}^{\infty}(\mathbf{P}^{\mathbf{u}})^t, \tag{5.8}$$

where the value $\mathbf{N}^{\mathbf{u}}((i,x),(j,y))$ is the mean number of visits of state (j,y) before the absorbing state is reached starting from state (i,x) [105, 40]. We define for the proper policy $\mathbf{u}$

$$\xi^{\mathbf{u}}(i,x) = \sum_{(j,y)\in\mathscr{S}} \mathbf{N}^{\mathbf{u}}((i,x),(j,y)) \tag{5.9}$$

as the mean number of steps until absorption starting from (i,x). Note, that matrices $\mathbf{P}^u$ have been obtained using uniformization with rate α as given in Eq. 5.5. Consider now the CTMC induced by the proper policy $\mathbf{u}$ applied to the CTMDP. Then the scaled value $\frac{\xi^{\mathbf{u}}(i,x)}{\alpha}$ is the expected total time spent in transient states before absorption if the process starts in state (i,x), i.e. it is the expected path weight corresponding to the policy $\mathbf{u}$.

Note that the existence of at least one proper policy is required in SSPP MDP (see Sec. 2.3.3). Since the matrix $\mathbf{D}_i$ of a PHD PH_i is nonsingular, the absorption occurs with probability 1 (cf. Eq. 2.1). Consequently, the PHD PH_i for each edge i is eventually left. Thus starting from arbitrary edge $i \in E_{ini}$ on the path leading to v_{fin} the path to the absorbing state $(0,0)$ exists which satisfies the requirement.

***Example* 5.1.** *Consider the stochastic graph highlighted in grey in Fig. 5.1. The edge weights are described by hyperexponential PHDs of order* 2*. The absorbing state corresponds to the destination* v_3*. In states* $(i_1,1)$*,* $(i_1,2)$ *two successor edges exist such that* $\mathscr{D}(i_1) = \{i_2,\ i_3\}$*. Furthermore, it holds that* $\mathscr{D}(i_2) = \mathscr{D}(i_3) = \{0\}$ *since* $i_2,\ i_3 \in E_{fin}$*. The transition probabilities for successor states of the states* $(i_1,\cdot)$ *are given as*

$$\mathbf{P}^{i_2} = \begin{pmatrix} 0.9231 & 0 & 0.0769 & 0 \\ 0 & 0.1373 & 0 & 0.8627 \end{pmatrix},\ \mathbf{P}^{i_3} = \begin{pmatrix} 0.9231 & 0 & 0.0461 & 0.0307 \\ 0 & 0.1373 & 0.5176 & 0.3461 \end{pmatrix}.$$

The transition probabilities for the states associated with edges i_2 *and* i_3 *are given as*

$$\begin{gathered}\mathbf{P}^0((i_2,1),(0,0)) = \mathbf{P}^0((i_3,1),(0,0)) = 0.0667,\\ \mathbf{P}^0((i_2,2),(0,0)) = \mathbf{P}^0((i_3,2),(0,0)) = 0.3333,\\ \mathbf{P}^0((i_2,1),(i_2,1)) = \mathbf{P}^0((i_3,1),(i_3,1)) = 0.9333,\\ \mathbf{P}^0((i_2,2),(i_2,2)) = \mathbf{P}^0((i_3,2),(i_3,2)) = 0.6667.\end{gathered}$$

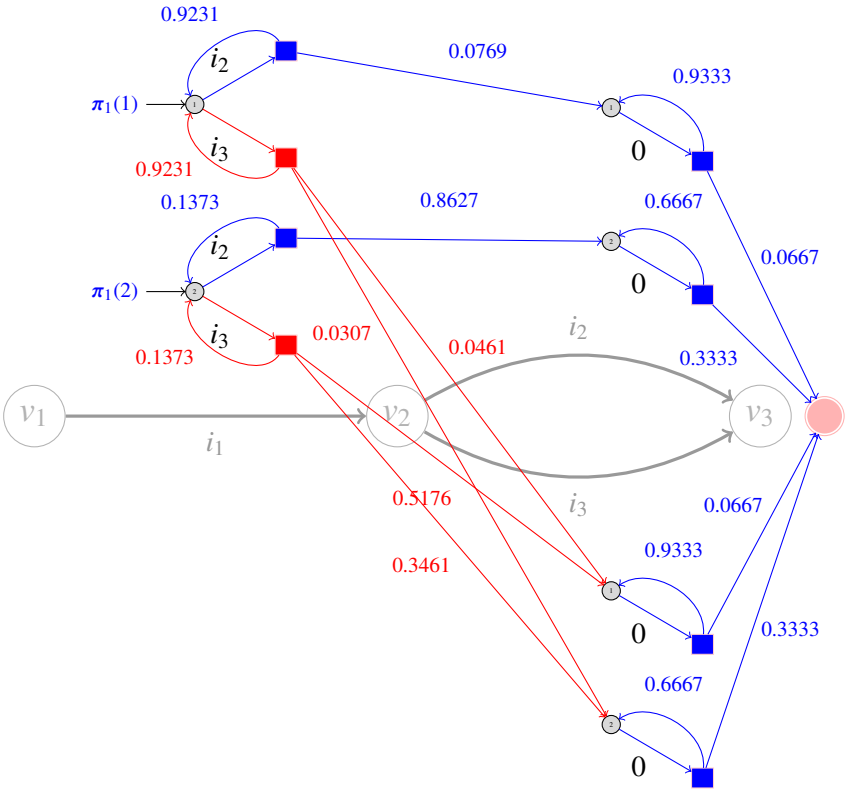


Figure 5.1.: Example for the SSPP of the PH-Graph with $v_{ini} = v_1$ and $v_{fin} = v_3$.

5.2. Analysis of Paths in PH-Graphs

In section 3.2 we defined the absorbing CTMC corresponding to a path $(i_1,\ldots,i_K)$ in a PH-Graph as given in [40]. The subgenerator matrix $\mathbf{Q}_{(i_1,\ldots,i_K)}$ defined in Eq. 3.17 contains nonsingular matrices $\mathbf{D}_i$ on its diagonal and thus is a nonsingular matrix. Then the matrix $(-\mathbf{Q}_{(i_1,\ldots,i_K)})^{-1}$ is the fundamental matrix (cf. Def. 2.4) of the absorbing CTMC corresponding to the PHG path. The entry $(-\mathbf{Q}_{(i_1,\ldots,i_K)})^{-1}((i,x),(j,y))$ gives the expected total time spent in state (j,y) before absorption given that the initial state is (i,x) [40]. Then the ith moment of the weight of the path is given by [40]

$$\mu^i_{(i_1,\ldots,i_K)} = i!\pi\mathbf{M}^i_{(i_1,\ldots,i_K)}\mathbb{1}, \tag{5.10}$$

where $\mathbf{M}_{(i_1,\ldots,i_K)} = (-\mathbf{Q}_{(i_1,\ldots,i_K)})^{-1}$ is the moment matrix and the initial distribution vector π is defined in Def. 3.2.

The probability that the path weight is less or equal to w is given by [40]

$$F_{(i_1,\ldots,i_K)}(w) = 1 - \pi e^{w\mathbf{Q}_{(i_1,\ldots,i_K)}}\mathbb{1}. \tag{5.11}$$

Depending on the application context, the first moment of the path weight $\mu_{(i_1,\ldots,i_K)}$ can be interpreted as the mean traveling time along the path or as the mean duration of the ruin process or as the mean strength of the connection between two nodes in

a stochastic graph. Correspondingly, $F_{(i_1,\ldots,i_K)}(w)$ is the distribution function of the traveling time along the path to a destination.

Note that the evaluation of the matrix exponential (see Eq. 2.9) in Eq. 5.11 can be computationally unstable such that the method of uniformization is used here to obtain an accurate numerical solution for the probability density function in Eq. 5.11 (see Sec. 2.1.2).

Given the subgenerator $\mathbf{Q}_{(i_1,\ldots,i_K)}$ the corresponding CTMC can be transformed into a DTMC using the uniformization method described in Sec. 2.1.2. The uniformization rate α is selected such that $\alpha \geq \max_{(i.x)\in\mathscr{S}_T}(|\mathbf{Q}_{(i_1,\ldots,i_K)}((i,x),(i,x))|)$. Then the matrix $\mathbf{P}_{(i_1,\ldots,i_K)} = \mathbf{Q}_{(i_1,\ldots,i_K)}/\alpha + \mathbf{I}$ is the transition probability matrix of the embedded Markov process as described in [40]. The fundamental matrix of the absorbing DTMC is then defined as [40]

$$\mathbf{N}_{(i_1,\ldots,i_K)} = (\mathbf{I}-\mathbf{P}_{(i_1,\ldots,i_K)})^{-1} = (\mathbf{I}-(\frac{\mathbf{Q}_{(i_1,\ldots,i_K)}}{\alpha}+\mathbf{I}))^{-1} = (-\mathbf{Q}_{(i_1,\ldots,i_K)}\frac{1}{\alpha})^{-1} = \mathbf{M}_{(i_1,\ldots,i_K)}\alpha. \tag{5.12}$$

The probability distribution function is then given by [40]

$$F_{(i_1,\ldots,i_K)}(w) = 1 - \left(\pi e^{-\alpha w}\sum_{n=0}^{\infty}\frac{(\alpha w)^n}{n!}\mathbf{P}^n_{(i_1,\ldots,i_K)}\right)\mathbb{1} \tag{5.13}$$

such that the sum in brackets need to be truncated. Finite truncation points can be pre-computed (cf. Eq. 2.14) to achieve the required error tolerance.

Now assume that edge weights are interpreted as traveling times. Then we are given a vehicle traversing a path $(i_1,\ldots,i_K)$ from the origin v_{ini} to the destination node v_{fin}. The vehicle only knows the PHD of the edges along the path and the existing dependencies between the edge traveling times. Every time the vehicle reaches a destination node $fin(i_k)$ of the edge k, $k \in \{1,\ldots,K\}$, the realization of the PHD random variable X_{i_k} is known. As the vehicle has passed through the subpath $(i_1,\ldots,i_l)$ for $l < K$, the realized traveling times $(w_1,\ldots,w_l)$ become known to it. Then the *history vector* as defined in [40]

$$\psi_{(i_1,w_1,\ldots,i_l,w_l)} = \pi_{i_1}\left(\prod_{k=1}^{l-1}e^{\mathbf{D}_{i_k}w_k}\mathbf{H}_{i_k\,i_{k+1}}\right)e^{\mathbf{D}_{i_l}w_l} \tag{5.14}$$

gives the conditional distribution after passing the edges $(i_1,\ldots,i_{l-1})$ with weights $(w_1,\ldots,w_{l-1})$ and having accumulated weight w_l at edge i_l, i.e. it gives the distribution among the states of PHD PH_l immediately before leaving the edge i_l. In the case that $w_l = 0$, the history vector gives the conditional distribution immediately after entering the edge i_l.

Again, the evaluation of the matrix exponential is required in equation 5.14. Using the uniformization method first the transition probability matrices $\mathbf{P}_{i_k}$ can be evaluated as described in [40]

$$\mathbf{P}_{i_k} = \mathbf{Q}_{i_k}/\alpha_{i_k} + \mathbf{I} \text{ with } \alpha_{i_k} \geq \max_x(|\mathbf{D}_{i_k}(x,x)|).$$

Then the history vector can be computed as given in [40]

$$\begin{aligned}\psi_{(i_1,w_1,\ldots,i_l,w_l)} = \quad & \pi_{i_1}\prod_{k=1}^{l-1}\left(e^{-\alpha_{i_k}w_k}\left(\sum_{n=0}^{\infty}\frac{(\alpha_{i_k}w_k)^n}{n!}(\mathbf{P}_{i_k})^n\right)\left(\frac{\mathbf{H}_{i_k\,i_{k+1}}}{\alpha_{i_k}}\right)\right)\cdot \\ & e^{-\alpha_{i_l}w_l}\left(\sum_{n=0}^{\infty}\frac{(\alpha_{i_l}w_l)^n}{n!}(\mathbf{P}_{i_l})^n\right),\end{aligned} \tag{5.15}$$

and should be normalized to 1 as defined in [40]

$$\bar{\psi}_{(i_1,w_1,\ldots,i_l,w_l)} = \frac{\psi_{(i_1,w_1,\ldots,i_l,w_l)}}{\psi_{(i_1,w_1,\ldots,i_l,w_l)}\mathbb{1}} \tag{5.16}$$

Depending on the known realized traveling times and correlations between edge weights the vehicle can decide which edge to traverse next using the history vector $\psi_{(i_1,w_1,\ldots,i_l,w_l)}$. In particular, the conditional moments of the remaining path $(i_{l+1},\ldots,i_K)$ can be computed analogously to Eq. 5.10 using the subgenerator as given in [40]

$$\mathbf{Q}_{(i_{l+1},\ldots,i_K)} = \begin{pmatrix} \mathbf{D}_{i_l} & \mathbf{H}_{i_l,i_{l+1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{i_{l+1}} & \mathbf{H}_{i_{l+1}i_{l+2}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & & \ddots & \mathbf{D}_{i_{K-1}} & \mathbf{H}_{i_{K-1}i_K} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{D}_K \end{pmatrix}$$

and initial vector $\pi' = (\bar{\psi}_{(i_1,w_1,\ldots,i_l,w_l)}, \mathbf{0})$ as defined in [40]

$$\mu'^{i}_{(i_{l+1},\ldots,i_K)} = i!\pi'\mathbf{M}^{i}_{(i_{l+1},\ldots,i_K)}\mathbb{1}, \tag{5.17}$$

***Example* 5.2.** *We present a stochastic network example from [40]. A four-node PHG is shown in Figure 5.2, it contains three paths from the origin v_1 to the destination v_4, namely (i_1,i_2), (i_3,i_4), and (i_5). The edge weights in the PHG are modeled by order*

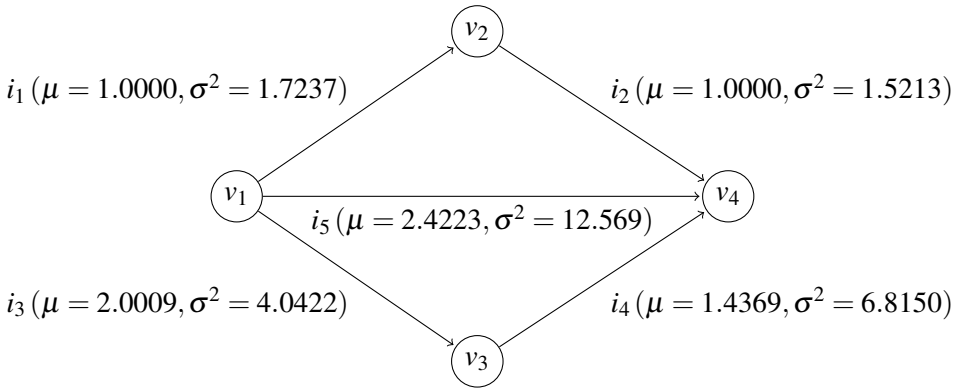


Figure 5.2.: The weights distribution for each edge is modeled by a PHD with expectation and variance given in a tuple at the corresponding edge.

4 PHDs $(\pi_{i_k}, \mathbf{D}_{i_k})$, $k \in \{1,\ldots,5\}$, in hyperexponential and hyper-Erlang representation, and are summarized below. Furthermore, weights can be interpreted as traveling times.

$$\pi_{i_1} = (0.4148, 0.1958, 0.1851, 0.2043), \mathbf{D}_{i_1} = \begin{pmatrix} -0.657 & 0 & 0 & 0 \\ 0 & -0.721 & 0 & 0 \\ 0 & 0 & -3.429 & 0 \\ 0 & 0 & 0 & -4.717 \end{pmatrix},$$

$$\pi_{i_2} = (0.5206, 0.3858, 0.0936, 0), \mathbf{D}_{i_2} = \begin{pmatrix} -0.724 & 0 & 0 & 0 \\ 0 & -3.534 & 0 & 0 \\ 0 & 0 & -1.085 & 1.085 \\ 0 & 0 & 0 & -1.085 \end{pmatrix},$$

$$\pi_{i_3} = (0.4227, 0.2707, 0.1814, 0.1252), \mathbf{D}_{i_3} = \begin{pmatrix} -0.4745 & 0 & 0 & 0 \\ 0 & -0.4900 & 0 & 0 \\ 0 & 0 & -0.5230 & 0 \\ 0 & 0 & 0 & -0.5940 \end{pmatrix}.$$

$$\pi_{i_4} = (0.3977, 0.3945, 0.2078, 0), \mathbf{D}_{i_4} = \begin{pmatrix} -0.3000 & 0 & 0 & 0 \\ 0 & -4.8847 & 0 & 0 \\ 0 & 0 & -6.8209 & 0 \\ 0 & 0 & 0 & -8.8200 \end{pmatrix},$$

$$\pi_{i_5} = (0.145, 0.478, 0.113, 0.264), \mathbf{D}_{i_5} = \begin{pmatrix} -0.385 & 0 & 0 & 0 \\ 0 & -0.241 & 0 & 0 \\ 0 & 0 & -5.5 & 0 \\ 0 & 0 & 0 & -6.325 \end{pmatrix}$$

The weights of the path $(i_1, i_2) \in Paths$ can be described by an acyclic absorbing CTMC with 8 transient states with the following subgenerator

$$\mathbf{Q}_{(i_1,i_2)} = \begin{pmatrix} -0.657 & 0 & 0 & 0 & 0.509 & 0 & 0.148 & 0 \\ 0 & -0.721 & 0 & 0 & 0.721 & 0 & 0 & 0 \\ 0 & 0 & -3.429 & 0 & 0.064 & 3.364 & 0 & 0 \\ 0 & 0 & 0 & -4.717 & 0 & 4.717 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.724 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -3.534 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.085 & 1.085 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.085 \end{pmatrix}$$

and the initial distribution vector $\pi_{(i_1,i_2)} = (\pi_{i_1}, 0, 0, 0, 0)$.

The first moment for the weight of the path (i_1, i_2) is computed using the moment matrix $\mathbf{M}_{(i_1,i_2)} = (-\mathbf{Q}_{(i_1,i_2)})^{-1}$ which results in $\mu^1_{(i_1,i_2)} = \pi_{(i_1,i_2)}\mathbf{M}_{(i_1,i_2)}\mathbb{1} = 1.9999$.

Furthermore, we obtain the following subgenerator for the CTMC corresponding to the weight of the path (i_3, i_4)

$$\mathbf{Q}_{(i_3,i_4)} = \begin{pmatrix} -0.4745 & 0 & 0 & 0 & 0.18871 & 0.18719 & 0.09806 & 0 \\ 0 & -0.49 & 0 & 0 & 0.19487 & 0.19331 & 0.10182 & 0 \\ 0 & 0 & -0.523 & 0 & 0.208 & 0.20632 & 0.10868 & 0 \\ 0 & 0 & 0 & -0.594 & 0.23623 & 0.23433 & 0.12343 & 0 \\ 0 & 0 & 0 & 0 & -0.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4.8847 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -6.8209 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -8.82 \end{pmatrix}$$

with initial distribution $\pi_{(i_3,i_4)} = (\pi_{i_3}, 0, 0, 0, 0)$. The expected travel time for the path (i_3, i_4) results in $\mu^1_{(i_3,i_4)} = \pi_{(i_3,i_4)}\mathbf{M}_{(i_3,i_4)}\mathbb{1} = 3.4377$. Finally, the weight of the path (i_5) is described by $\mathbf{Q}_{i_5} = \mathbf{D}_{i_5}$ such that the expected travel time equals 2.4223.

We can now compute the history vector $\psi_{(i_1,w_1,i_2,0)}$ using (5.15), and (5.16). Assume that the traveling time on the edge i_1 was $w_1 = 0.5$. Then the vector including the distribution immediately after entering the edge i_2 results in $\psi_{(i_1,0.5,i_2,0)} = \pi_{i_1} e^{\mathbf{D}_{i_1} 0.5} \mathbf{H}_{i_1\, i_2} e^{\mathbf{D}_{i_2} 0}$, and the normalized history vector is $\bar{\psi}_{(i_1,0.5,i_2,0)} = (0.5052, 0.4064, 0.0884, 0)$. The conditional weights of the remaining path through the adjacent edge i_2 can be computed from an absorbing CTMC with generator matrix containing only the generator $\mathbf{D}_{i_2}$. Then the first conditional moment results in $\bar{\psi}_{(i_1,0.5,i_2,0)}\mathbf{M}_{i_2}\mathbb{1} = 0.9752$. We plotted the values of the first conditional moment using weights in interval $[0, 2]$ which is visualized in Fig. 5.3.

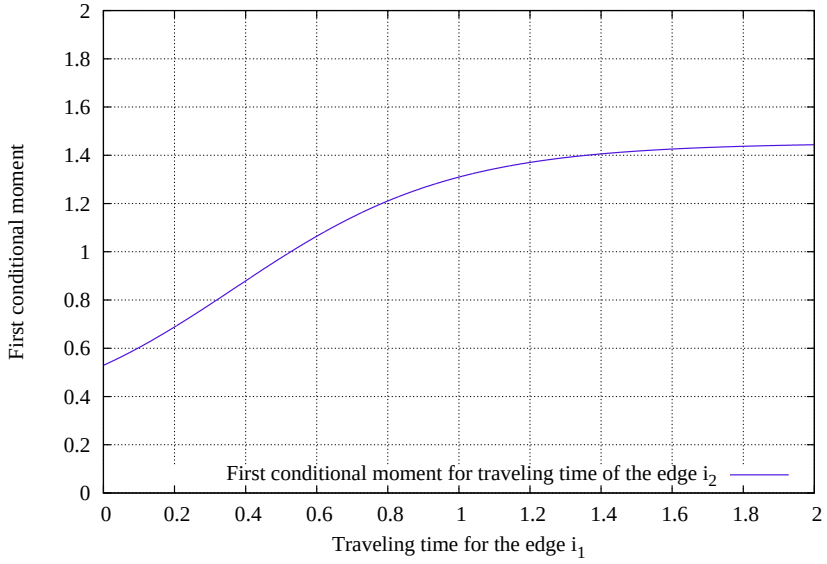


Figure 5.3.: The expected traveling time of the adjacent edge $i_2 \in i_1\bullet$ depending on the weight of the edge i_1.

5.3. Solution methods for SSPP

As already mentioned in Sec. 2.3 the optimal stationary policy $\mathbf{u}^*$ and the corresponding gain vector $\mathbf{g}^*$ of a SSPP can be computed using value iteration, policy iteration and linear programming which we describe from the sources [29, 27, 28, 40] and references therein.

5.3.1. Value Iteration

Using *value iteration* (VI) the dynamic programming iteration given in Eq. 2.50 is computed starting with some initial gain vector $\mathbf{g}_0$ as

$$T\mathbf{g}(i) = \min_{u \in \mathscr{D}(i)} \left(\mathbf{r}^u(i) + \sum_{j=1}^{n} \mathbf{P}^u(i,j)\,\mathbf{g}(j) \right), \quad \text{for all } i \in \mathscr{S}_T,$$

such that the sequence $T\mathbf{g}_0, T^2\mathbf{g}_0, \ldots$ is generated successively. In particular, the sequence of value representations $\mathbf{g}_k = T\mathbf{g}_{k-1}$ is computed iteratively for $k = 1$, where the T operator is defined in Eq. 2.36, and the resulting optimal reward vector $\mathbf{g}_k$ after k-th iteration. In fact, $\mathbf{g}_k = T\mathbf{g}_{k-1}$ is computed by applying the mapping T k times to $\mathbf{g}_0$, i.e. $\mathbf{g}_k = T^k\mathbf{g}_0$.

The convergence of the value iteration to the optimal total reward vector $\mathbf{g}^*$ has been shown in [28, Proposition 2.2.2] (see Eq. 2.51). During the iteration of the dynamic

programming algorithm the sequence of errors $|\mathbf{g}_k(i) - \mathbf{g}^*(i)|$ is given in the k-th iteration, such that the errors are bounded as follows

$$|\,\mathbf{g}_{nK}(i) - \mathbf{g}^*(i)\,| \leq p^K M, \tag{5.18}$$

where $p = \max_{\mathbf{u}\in\Pi} p^{\mathbf{u}}$, and the reachability probability $p^{\mathbf{u}}$ is defined in Eq. 2.40. Observe that n denotes in Eq. 2.40 an integer such that there is a positive probability that the absorbing state will be reached after no more that n steps, or rather n decision epochs. K is a positive integer, and $M = n \cdot \max\limits_{i\in\mathscr{S}_T,\, u\in\mathscr{D}(i)} |\,r^u(i)\,|$. Consider now the following error bounds

$$\varepsilon_k^{min} = \min_{s_i\in\mathscr{S}_T} \left(\mathbf{g}_{k+1}(i) - \mathbf{g}_k(i)\right), \quad \varepsilon_k^{max} = \max_{s_i\in\mathscr{S}_T} \left(\mathbf{g}_{k+1}(i) - \mathbf{g}_k(i)\right). \tag{5.19}$$

Then it holds for every state $i \in \mathscr{S}_T$, iteration k and arbitrary vector $\mathbf{g}$ [27, Eq. 7.17]

$$\mathbf{g}_{k+1}(i) + \varepsilon_k^{min}(N_*(i) - 1) \leq \mathbf{g}^*(i) \leq \mathbf{g}^{\mathbf{u}_k}(i) \leq \mathbf{g}_{k+1}(i) + \varepsilon_k^{max}(N_{\mathbf{u}_k}(i) - 1), \tag{5.20}$$

where the following terms occur

- $\mathbf{u}_k$ denotes the stationary policy whose $\mathbf{u}_k(i)$ element results in the minimum in the k-th iteration for all i.
- $N_*(i)$ denotes the expected number of steps before absorption starting from state i and following the optimal policy.
- $N_{\mathbf{u}_k}(i)$ denotes the expected number of steps before absorption starting from state i and using the stationary policy $\mathbf{u}_k$.

In principle, the value iteration method runs for an infinite number of iterations. Thus using the established error bounds one can decide when to stop the value iteration method. In each value iteration $\mathbf{g}_k$ approximates $\mathbf{g}^*$ with sufficient accuracy, even though the error bounds in Eq. 5.20 can be computed only if values N_*, $N_{\mathbf{u}_k}$ can be determined. Unfortunately, the values N_*, $N_{\mathbf{u}_k}$ can be not be efficiently obtained in general as explained in [27].

The following example demonstrates the convergence properties of the value iteration method.

***Example* 5.3.** *Consider the SSPP instance in example 2.5. Since we have two admissible decisions in states* $1, 2$*, namely* u_1 *and* u_2*, the k-th value iteration performs the computation of the function* T *as*

$$\mathbf{g}_k(i) = T\mathbf{g}_{k-1}(i) = \min_{u_1,u_2} \left(\mathbf{r}^{u_1}(i) + \sum_{j=1}^{3} \mathbf{P}^{u_1}(i,j)\,\mathbf{g}_{k-1}(j),\, \mathbf{r}^{u_2}(i) + \sum_{j=1}^{3} \mathbf{P}^{u_2}(i,j)\,\mathbf{g}_{k-1}(j)\right).$$

Initially, $k = 1$ *and the value iteration method is initialized with the initial cost vector* $\mathbf{g}_0 = (0,0,0)$*. Then the computation of the above equation is performed such that the values vector* $\mathbf{g}_1 = T\mathbf{g}_0$ *is obtained*

$$\mathbf{g}_1(1) = 0.65,\ \ \mathbf{g}_1(2) = 0.75,\ \ \mathbf{g}_1(3) = 0.$$

In the next iteration $k=2$ and the values vector $\mathbf{g}_2 = T\mathbf{g}_1$ is obtained after computation of the above dynamic programming expression

$$\mathbf{g}_2(1) = 1.005,\ \mathbf{g}_2(2) = 1.275,\ \mathbf{g}_2(3) = 0.$$

To compute the error bounds we first evaluate

$$\varepsilon_k^{min} = \min\left(\mathbf{g}_k(1) - \mathbf{g}_{k-1}(1),\ \mathbf{g}_k(2) - \mathbf{g}_{k-1}(2)\right),$$
$$\varepsilon_k^{max} = \max\left(\mathbf{g}_k(1) - \mathbf{g}_{k-1}(1),\ \mathbf{g}_k(2) - \mathbf{g}_{k-1}(2)\right),$$

which results for the first two value iterations in

$$\varepsilon_1^{min} = 0.6,\ \varepsilon_1^{max} = 0.75,$$
$$\varepsilon_2^{min} = 0.405,\ \varepsilon_2^{max} = 0.525.$$

In our example all policies are proper and it is obvious that the optimal policy $\mathbf{d}^{min}$ minimizes the expected infinite-horizon total reward for $\mathbf{u}(1) = u_1$ and $\mathbf{u}(2) = u_1$. In that case the value $N_* = N_{u_1}$ is finite and can be computed as $(\mathbf{I} - \mathbf{P}^{u_1})^{-1}$. Analogously, N_{u_2} can be computed as $(\mathbf{I} - \mathbf{P}^{u_2})^{-1}$. Observe that the matrices $\mathbf{P}^{u_1}$, $\mathbf{P}^{u_2}$ are of dimension $\mathscr{S}_T \times \mathscr{S}_T$ and describe transition probabilities for transient states

$$\mathbf{P}^{u_1} = \begin{bmatrix} 0.3 & 0.3 \\ 0.25 & 0.5 \end{bmatrix}, \mathbf{P}^{u_2} = \begin{bmatrix} 0.4 & 0.4 \\ 0.3 & 0.5 \end{bmatrix}.$$

We compute $N_{u_1}\mathbb{1} = (2.9091, 3.4545)^T$, $N_{u_2}\mathbb{1} = (5,5)^T$ which results in the following error bounds

$$\mathbf{g}_1(1) + \varepsilon_1^{min}(N_{u_1}(1) - 1) = 0.6 + 0.6 * 1.909 = 1.7454,$$
$$\mathbf{g}_1(1) + \varepsilon_1^{max}(N_{u_1}(1) - 1) = 0.6 + 0.75 * 1.909 = 2.03175,$$
$$\mathbf{g}_1(2) + \varepsilon_1^{min}(N_{u_1}(2) - 1) = 0.75 + 0.6 * 2.4545 = 2.2227,$$
$$\mathbf{g}_1(2) + \varepsilon_1^{max}(N_{u_1}(2) - 1) = 0.75 + 0.75 * 2.4545 = 2.590875.$$

The results of the value iteration method with error bounds are shown in Fig 5.4.

Discounted Problems as a variant of SSPP We will now discuss the case where the error bounds can be easily obtained as described in [27]. It corresponds to the infinite horizon discounted problems. Particularly, any discounted infinite-horizon MDP with discount factor $\gamma \in [0,1)$ can be reduced to an equivalent SSPP with the following structure.

Assume that in the original discounted MDP the state space consists of n states, i.e. $\mathscr{S} = \{1,\ldots,n\}$. In the corresponding SSPP the absorbing state $n+1$ with $\mathbf{P}(n+1, n+1) = 1$ is added. Additionally, there is a probability $(1-\gamma)$ to get absorbed for every state and decision pair (s_i, u), $i \neq n+1$, $u \in \mathscr{D}(i)$. The transition to the state j occurs with the normalized probability $\gamma\mathbf{P}^u(i,j)$ for each transient state and decision pair. Then in the constructed SSPP instance the assumption about existence of at least one proper policy is satisfied.

***Example* 5.4.** *Observe the following infinite horizon discounted problem and the associated SSPP in Fig. 5.5. One can see that the SSPP Markov chain represents a discrete*

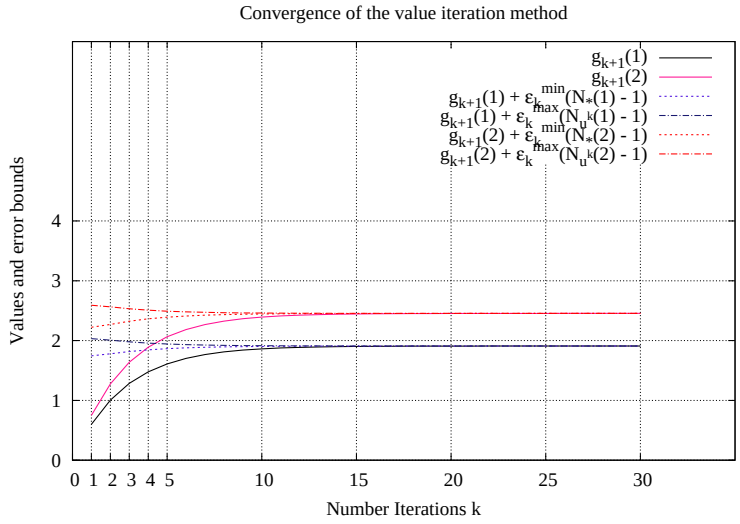


Figure 5.4.: The progress of the value iteration method with corresponding error bounds. Value iteration converges to the optimal values $\mathbf{g}^*(1) = 1.9091, \mathbf{g}^*(2) = 2.4545$.

time phase-type distribution where all transient states have the same absorbing probability $\mathbf{P}^u(i,n+1) = 1-\gamma$, *for* $i \in \mathscr{S}_T$ *and* $u \in \mathscr{D}(i)$. *Correspondingly, the probability to not getting absorbed is* γ.

Since the sojourn time X in a state is geometrically distributed with success probability $p = 1-\gamma$, *the expected number of steps before absorption is given by* $E(X) = \frac{1}{p} = \frac{1}{1-\gamma}$ *for any transient state. In fact, the expected value of the represented dicrete time phase-type distribution is given by* $\frac{1}{1-\gamma}$. *Thus the values* N_* *and* $N_{\mathbf{u}_k}$ *can be assumed to be* $\frac{1}{1-\gamma}$.

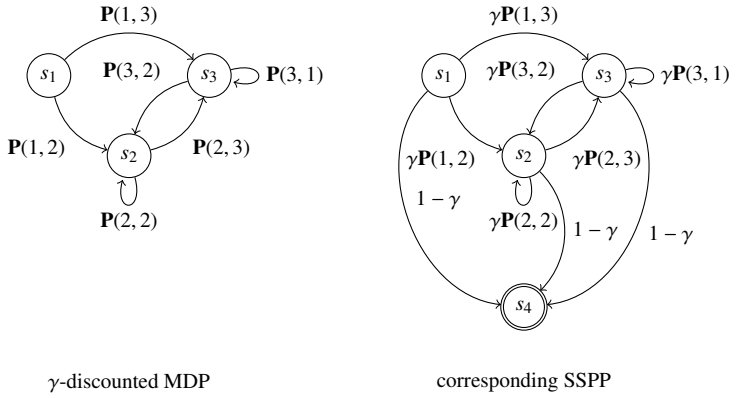
In the discounting case the obtained expected reward after n steps is $\gamma^n r^u(i)$ if decision u in state i is chosen. In the associated SSPP the expected reward after n steps in some state i also equals to $\gamma^n r^u(i)$. The reason for this is that the reward $r^u(i)$ is obtained with probability γ^n which is the probability of not get absorbed after n steps. With this insights the error bounds for the discounting case are given by

$$\mathbf{g}_{k+1}(i) + \varepsilon_k^{min}\left(\frac{\gamma}{1-\gamma}\right) \leq \mathbf{g}^*(i) \leq \mathbf{g}^{\mathbf{u}_k}(i) \leq \mathbf{g}_{k+1}(i) + \varepsilon_k^{max}\left(\frac{\gamma}{1-\gamma}\right). \tag{5.21}$$

Minimization of the expected time till absorption Observe the case introduced in [27] where expected rewards for all states have the same unit value

$$\mathbf{r}^u(i) = 1, \text{ for all } i \in \mathscr{S}_T,\ u \in \mathscr{D}(i). \tag{5.22}$$

In infinite horizon MDPs with the above structure and under the expected total reward criterion the aim is to minimize the expected lifetime of the induced Markov chain.



γ-discounted MDP corresponding SSPP

Figure 5.5.: Transition diagramms for an γ-discounted MDP and the equivalent SSPP.

This lifetime corresponds to the expected time till reaching the absorbing state. As the absorbing state $n+1$ is interpreted as the destination, the SSPP solution corresponds to the fastest path on the average. The value $\mathbf{g}^*(i)$ gives the minimum expected time until absorption starting from state i. Then the values $\mathbf{g}^*(i)$ for all $i \in \mathscr{S}_T$ are the unique solution of Bellman's equation [27]

$$\mathbf{g}^*(i) = \min_{u \in \mathscr{D}(i)} \left(1 + \sum_{j=1}^{n} \mathbf{P}^u(i,j)\, \mathbf{g}^*(j) \right). \tag{5.23}$$

Note that in discrete time case $\mathbf{g}^*(i)$ is the minimum expected number of steps until absorption [27, 40].

Gauss-Seidel Value Iteration Method *Gauss-Seidel* version of dynamic programming differs from the value iteration method described earlier in that the values $\mathbf{g}(i)$ are no longer computed for all states i simultaneously. The method computes the value $\mathbf{g}(i)$ for one state at a time using the recent values of other states. Then the sequence of values $\mathbf{g}(i)$ is generated successively starting with the first state 1. For this the mapping $F : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$(F\mathbf{g})(1) = \min_{u \in \mathscr{D}(1)} \left(r^u(1) + \sum_{j=1}^{n} \mathbf{P}^u(1,j)\, \mathbf{g}(j) \right), \tag{5.24}$$

and proceeding for states i with $i = 2, \ldots, n$

$$(F\mathbf{g})(i) = \min_{u \in \mathscr{D}(i)} \left(r^u(i) + \sum_{j=1}^{i-1} \mathbf{P}^u(i,j)\, (F\mathbf{g})(j) + \sum_{j=i}^{n} \mathbf{P}^u(1,j)\, \mathbf{g}(j) \right). \tag{5.25}$$

Here states are numbered in some predefined or problem specific order. Generally, the order in which the values $(F\mathbf{g})(i)$ are computed has an influence on the course of the method, e.g., on its convergence. In [23] it has been mentioned that in SSPP it is

advantageous to start the computation with states which are directly connected to the absorbing state and then to proceed backwards along potential shortest paths. It is less efficient to start the computation with initial states in the forward direction.

Observe that $(F\mathbf{g})(i)$ works analogously to the T operator when computing $(T\mathbf{g})(i)$ as in Eq. 2.36. Additionally, the recently computed values $(F\mathbf{g})(1),\ldots,(F\mathbf{g})(i-1)$ are incorporated in the above formula, such that values $\mathbf{g}(1),\ldots,\mathbf{g}(i-1)$ are replaced in one iteration. After n iterations all $\mathbf{g}(i)$ values are replaced by $(F\mathbf{g})(i)$. Analogously to the value iteration method the sequence $F\mathbf{g}_0, F^2\mathbf{g}_0,\ldots$ is generated successively. In [27] it has been shown that the Gauss-Seidel dynamic programming converges to $\mathbf{g}^*$ under the same conditions that ensure the convergence of the value iteration method for SSPP, but the essential advantage is that the Gauss-Seidel converges faster.

Asynchronous fixed point iteration *Asynchronous dynamic programming* is a variant of Gauss-Seidel method where the expected rewards of states are computed in arbitrary order as described in [27]. The algorithm selects an arbitrary state i at a time and computes its new value $\mathbf{g}_k(i) = T\mathbf{g}_{k-1}(i)$ in the k-th decision epoch. Values for other states j, $j \neq i$, remain unchanged. Depending on the state selection rule it is possible that the expected reward for some state may be computed several times before the rewards for other states are computed once [23].

Using asynchronous dynamic programming the arbitrary initial reward vector $\mathbf{g}_0$ converges to the optimal gain vector $\mathbf{g}^*$ as stated by results of Bertsekas [27]. The convergence result holds under the condition that all states are selected infinitely often which means that a state selection rule should never exclude some state from selection in the future. Further conditions are the existence of at least one proper policy and that all improper policies incur infinite expected rewards for at least one initial state [27, 23]. Similarly to Gauss-Seidel method the order in which the values are computed has an influence on the rate of convergence.

Reduction of the set of decisions It is possible to eliminate non optimal decisions from computation in the progress of the value iteration method as described in [27, 28]. As mentioned above the optimal policy $\mathbf{u}$ minimizes the righthand side of the equation $\mathbf{g}^* = T\,\mathbf{g}^*$ such that the equation $T\,\mathbf{g}^* = T_{\mathbf{u}}\,\mathbf{g}^*$ holds (see Eq. 2.53). If some $u \in \mathscr{D}(i)$ exists such that

$$\mathbf{g}^*(i) < \mathbf{r}^u(i) + \sum_{j=1}^{n} \mathbf{P}^u(i,j)\,\mathbf{g}^*(j)$$

for some $i \in \mathscr{S}_T$ then the decision u is non optimal. Thus it can be eliminated from the considered set of decisions in Eq. 2.50. Unfortunately, the optimal gain vector $\mathbf{g}^*$ is not known in advance. However, it is stated in [28] that the following inequality holds for non optimal decision u

$$\overline{\mathbf{g}}(i) < \mathbf{r}^u(i) + \sum_{j=1}^{n} \mathbf{P}^u(i,j)\,\underline{\mathbf{g}}(j), \tag{5.26}$$

where $\underline{\mathbf{g}}(i)$, $\overline{\mathbf{g}}(i)$ are the lower and upper bounds with

$$\underline{\mathbf{g}}(i) \leq \mathbf{g}^*(i) \leq \overline{\mathbf{g}}(i), \text{ for all } i \in \mathscr{S}_T.$$

The lower and upper bounds converge to $\mathbf{g}^*$ and can be computed using error bounds introduced in Eq. 5.20. Then Eq. 5.26 can be evaluated for all $u \in \mathscr{D}(i)$ in the course of value iteration. The decisions satisfying Eq. 5.26 can be eliminated. After a finite number of value iterations possibly all non optimal decisions can be eliminated since the set of decisions is finite. Then the considered set $\mathscr{D}(i)$ can be reduced to optimal decisions for the state i thus accelerating the value iteration method.

Pathological SSPP instances We now describe the case from [27, 28] where the assumption that each improper policy results in an infinite expected total reward for at least one initial state does not hold. In this case the mapping T has multiple fixed points. Then the results $\mathbf{g}^* = T\,\mathbf{g}^*$, $lim_{t\to\infty}(T^k\mathbf{g})(i) = \mathbf{g}^*(i)$ for $i = 1,\ldots,n$ are fragile and their validity cannot be guaranteed any more.

Consider the deterministic shortest path example in Fig. 5.6 involving a cycle with zero rewards and first described by Bertsekas [28]. Note that given an initial reward

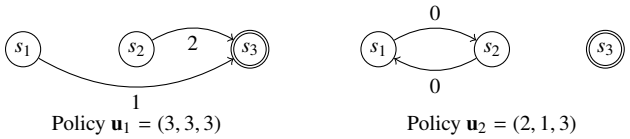


Figure 5.6.: Instance of deterministic SSPP with cycle involving zero rewards.

vector $\mathbf{g}_0$, value iteration generates the sequence $\mathbf{g}_1, \mathbf{g}_2, \ldots$. The method computes the k-th value representation as $\mathbf{g}_k = T\,\mathbf{g}_{k-1}$ and returns $\mathbf{g}^+ = \lim_{k\to\infty}(T^k\,\mathbf{g}_0)$. The optimal expected reward $\mathbf{g}^*$ is a fixed point of the Bellman operator T. If the required assumptions are not satisfied it is possible that the T operator admits more than one fixpoint. In that case the sequence $\mathbf{g}_1, \mathbf{g}_2, \ldots$ can converge to any fixed point depending on the initial gain vector $\mathbf{g}_0$ [59].

In particular, the vector $(0,0,0)$ satisfies the Bellman's equation $\mathbf{g} = T\,\mathbf{g}$. Then the improper policy $\mathbf{u}_2$ admits 0 expected total reward for any initial state. In contrast it is required that every improper policy yields an infinite expected total reward for at least one initial state which is not the case here. Similar SSPP instances can be solved by eliminating the zero rewards state-action pairs from the MDP (see, e.g. [59]).

Another prominent example is the *pure stopping problem* where all rewards are 0 and the reward obtained when the stopping decision is chosen equals -1 [27, 28]. Here the stopping decision can be associated with the decision directly to go to the absorbing state. Since in stopping problems eventual stopping in each decision epoch is required, the smallest reward -1 should make the decision to stop in each state favorable. It has been shown that an improper policy would never contain the decision to stop. Though it yields finite expected reward 0 for each initial state.

5.3.2. Policy Iteration

Generally, the value iteration method yields in the limit the optimal reward function and an optimal policy. In this section we describe an alternative approach based on

work [148, 27, 28] which terminates in a finite number of iterations assuming that the state space and desicion space are finite.

Concerning the fact that there are finitely many proper policies $\mathbf{u} \in \Pi$ the aim is to iterate over the finite set Π. This implies that termination occurs in a finite number of steps. The corresponding method is implemented in *policy iteration* (PI) which starts with an arbitrary proper policy $\mathbf{u}_0$, and generates the sequence of improving policies $\mathbf{u}_1$, $\mathbf{u}_2, \ldots$. Particularly, we start with policy $\mathbf{u}_k$ in the k-th iteration and compute the values $\mathbf{g}^{\mathbf{u}_k}$ in the *policy evaluation* step as the solution of the linear system of equations

$$\mathbf{g}(i) = \mathbf{r}^{\mathbf{u}_k}(i) + \sum_{j=1}^{n} \mathbf{P}^{\mathbf{u}_k}(i,j)\,\mathbf{g}(j), \text{ for all } i \in \mathscr{S}_T, \tag{5.27}$$

in the n unknowns corresponding to the values $\mathbf{g}(i)$. Then the values $\mathbf{g}(i)$ from the solution determine the vector $\mathbf{g}^{\mathbf{u}_k}$. Observe that Eq. 5.27 can be equivalently written as Eq. 2.56. After the value $\mathbf{g}^{\mathbf{u}_k}$ of a policy $\mathbf{u}_k$ is known it can be minimized in the *policy improvement step* by the computation of

$$\mathbf{u}_{k+1}(i) = \arg\min_{u \in \mathscr{D}(i)} \left(\mathbf{r}^{u}(i) + \sum_{j=1}^{n} \mathbf{P}^{u}(i,j)\,\mathbf{g}^{\mathbf{u}_k}(j) \right), \tag{5.28}$$

for all $i = 1, \ldots, n$. Observe that in Eq. 5.28 the value of the k-th policy is used in the second term. Here the policy $\mathbf{u}_{k+1}$ is improved by minimization in the dynamic programming equation considering $\mathbf{g}^{\mathbf{u}_k}$. A new stationary policy satisfies the equation $T_{\mathbf{u}_{k+1}}\,\mathbf{g}^{\mathbf{u}_k} = T\,\mathbf{g}^{\mathbf{u}_k}$ in the T-operator notation.

In the $(k+1)$-th iteration the policy evaluation step is performed using the policy $\mathbf{u}_{k+1}$ instead of $\mathbf{u}_k$. In the course of alternating policy evaluation and improvement steps an improving sequence of policies is produced. In particular, $\mathbf{g}^{\mathbf{u}_{k+1}}(i) \leq \mathbf{g}^{\mathbf{u}_k}(i)$ holds for all states $i \in \mathscr{S}_T$ and all iterations k [27, Prop. 7.2.2]. Furthermore the policy improvement step in Eq. 5.28 is based on the following proposition:

Proposition 5.1. *[28, Prop. 1.3.4] Let* $\mathbf{u}$, $\mathbf{u}'$ *be two stationary policies in the policy iteration step* $T_{\mathbf{u}'}\,\mathbf{g}^{\mathbf{u}} = T\mathbf{g}^{\mathbf{u}}$, *i.e.*

$$\mathbf{r}^{\mathbf{u}'}(i) + \sum_{j=1}^{n} \mathbf{P}^{\mathbf{u}'}(i,j)\,\mathbf{g}^{\mathbf{u}}(j) = \min_{u \in \mathscr{D}(i)} \left(\mathbf{r}^{u}(i) + \sum_{j=1}^{n} \mathbf{P}^{u}(i,j)\,\mathbf{g}^{\mathbf{u}}(j) \right).$$

Then it holds that

$$\mathbf{g}^{\mathbf{u}'}(i) \leq \mathbf{g}^{\mathbf{u}}(i), \text{ for all } i \in \mathscr{S}_T,$$

where the strict inequality arises for at least one state i if the policy $\mathbf{u}$ *is not optimal.*

Policy iteration terminates in the $(k+1)$-th iteration with the optimal policy $\mathbf{u}_k$ when $\mathbf{g}^{\mathbf{u}_{k+1}}(i) = \mathbf{g}^{\mathbf{u}_k}(i)$ holds for all $i \in \mathscr{S}_T$.

***Example* 5.5.** *Consider the SSPP instance given in Example 2.5 where we have already obtained the convergence result using value iteration in Example 5.3. Policy iteration finds the optimal policy* $\mathbf{u}^*$ *and the corresponding gain vector in two steps.*

Let $\mathbf{u}^0 = (u_2, u_2)$ *be the initial policy. In the policy evaluation step solution of the linear system of equations* $(\mathbf{I} - \mathbf{P}^{u_2})^{-1}\mathbf{r}^{u_2}$ *is required. This corresponds to*

$$\begin{aligned}\mathbf{g}^{\mathbf{u}_0}(1) &= \mathbf{r}^{u_2}(1) + \mathbf{P}^{u_2}(1,1)\,\mathbf{g}^{\mathbf{u}_0}(1) + \mathbf{P}^{u_2}(1,2)\,\mathbf{g}^{\mathbf{u}_0}(2)\\ \mathbf{g}^{\mathbf{u}_0}(2) &= \mathbf{r}^{u_2}(2) + \mathbf{P}^{u_2}(2,1)\,\mathbf{g}^{\mathbf{u}_0}(1) + \mathbf{P}^{u_2}(2,2)\,\mathbf{g}^{\mathbf{u}_0}(2),\end{aligned}$$

where by substituting the expected rewards and transition probabilities we obtain

$$\begin{aligned}\mathbf{g}^{\mathbf{u}_0}(1) &= 0.85 + 0.4\,\mathbf{g}^{\mathbf{u}_0}(1) + 0.4\,\mathbf{g}^{\mathbf{u}_0}(2)\\ \mathbf{g}^{\mathbf{u}_0}(2) &= 0.8 + 0.3\,\mathbf{g}^{\mathbf{u}_0}(1) + 0.5\,\mathbf{g}^{\mathbf{u}_0}(2).\end{aligned}$$

Solving the above system of equations for $\mathbf{g}^{\mathbf{u}_0}(1)$ *and* $\mathbf{g}^{\mathbf{u}_0}(2)$ *we obtain the vector*

$$\mathbf{g}^{\mathbf{u}_0} = (4.138,\ 4.083).$$

In the policy improvement step the values $\mathbf{u}_1(1)$, $\mathbf{u}_1(2)$ *satisfying* $T_{\mathbf{u}_1}\,\mathbf{g}^{\mathbf{u}_0} = T\,\mathbf{g}^{\mathbf{u}_0}$ *are computed as*

$$\begin{aligned}\mathbf{u}_1(1) &= \arg\min_{u\in\mathscr{D}(i)}\left\{\mathbf{r}^{u}(1) + \sum_{j=1}^{n}\mathbf{P}^{u}(1,j)\,\mathbf{g}^{\mathbf{u}_0}(j)\right\}\\ &= \arg\min_{u_1,u_2}\{\,\mathbf{r}^{u_1}(1) + \mathbf{P}^{u_1}(1,1)\,\mathbf{g}^{\mathbf{u}_0}(1) + \mathbf{P}^{u_1}(1,2)\,\mathbf{g}^{\mathbf{u}_0}(2),\\ &\qquad\qquad \mathbf{r}^{u_2}(1) + \mathbf{P}^{u_2}(1,1)\,\mathbf{g}^{\mathbf{u}_0}(1) + \mathbf{P}^{u_2}(1,2)\,\mathbf{g}^{\mathbf{u}_0}(2)\,\}\\ &= \arg\min_{u_1,u_2}\{\,0.6 + 0.3\cdot 4.138 + 0.3\cdot 4.083,\\ &\qquad\qquad 0.85 + 0.4\cdot 4.138 + 0.4\cdot 4.083\,\}\\ &= \arg\min_{u_1,\,u_2}(\,3.06, 4.138\,) = u_1,\end{aligned}$$

and

$$\begin{aligned}\mathbf{u}_1(2) &= \arg\min_{u\in\mathscr{D}(i)}\left\{\mathbf{r}^{u}(2) + \sum_{j=1}^{n}\mathbf{P}^{u}(2,j)\,\mathbf{g}^{\mathbf{u}_0}(j)\right\}\\ &= \arg\min_{u_1,u_2}\{\,\mathbf{r}^{u_1}(2) + \mathbf{P}^{u_1}(2,1)\,\mathbf{g}^{\mathbf{u}_0}(1) + \mathbf{P}^{u_1}(2,2)\,\mathbf{g}^{\mathbf{u}_0}(2),\\ &\qquad\qquad \mathbf{r}^{u_2}(2) + \mathbf{P}^{u_2}(2,1)\,\mathbf{g}^{\mathbf{u}_0}(1) + \mathbf{P}^{u_2}(2,2)\,\mathbf{g}^{\mathbf{u}_0}(2)\,\}\\ &= \arg\min_{u_1,u_2}\{\,0.75 + 0.25\cdot 4.138 + 0.5\cdot 4.083,\\ &\qquad\qquad 0.8 + 0.3\cdot 4.138 + 0.5\cdot 4.083\,\}\\ &= \arg\min_{u_1,\,u_2}(\,3.82638, 4.0829\,) = u_1,\end{aligned}$$

such that minimizing decisions are

$$\mathbf{u}_1 = (u_1, u_1).$$

In the next policy evaluation step the solution of $(\mathbf{I} - \mathbf{P}^{u_1})^{-1}\mathbf{r}^{u_1}$ *has to be found, or equivalently*

$$\begin{aligned}\mathbf{g}^{\mathbf{u}_1}(1) &= \mathbf{r}^{u_1}(1) + \mathbf{P}^{u_1}(1,1)\,\mathbf{g}^{\mathbf{u}_1}(1) + \mathbf{P}^{u_1}(1,2)\,\mathbf{g}^{\mathbf{u}_1}(2)\\ &= 0.6 + 0.3\,\mathbf{g}^{\mathbf{u}_1}(1) + 0.3\,\mathbf{g}^{\mathbf{u}_1}(2),\\ \mathbf{g}^{\mathbf{u}_1}(2) &= \mathbf{r}^{u_1}(2) + \mathbf{P}^{u_1}(2,1)\,\mathbf{g}^{\mathbf{u}_1}(1) + \mathbf{P}^{u_1}(2,2)\,\mathbf{g}^{\mathbf{u}_1}(2)\\ &= 0.75 + 0.25\,\mathbf{g}^{\mathbf{u}_1}(1) + 0.5\,\mathbf{g}^{\mathbf{u}_1}(2).\end{aligned}$$

The linear system of equations attains the gain vector

$$\mathbf{g}^{\mathbf{u}_1} = (1.909, 2.4545).$$

Performing the policy improvement step we obtain

$$\begin{aligned}
\mathbf{u}_2(1) &= \arg\min_{u\in\mathscr{D}(i)} \left\{ \mathbf{r}^u(1) + \sum_{j=1}^{n} \mathbf{P}^u(1,j)\,\mathbf{g}^{\mathbf{u}_1}(j) \right\} \\
&= \arg\min_{u_1,u_2} \{ \mathbf{r}^{u_1}(1) + \mathbf{P}^{u_1}(1,1)\,\mathbf{g}^{\mathbf{u}_1}(1) + \mathbf{P}^{u_1}(1,2)\,\mathbf{g}^{\mathbf{u}_1}(2), \\
&\qquad\qquad \mathbf{r}^{u_2}(1) + \mathbf{P}^{u_2}(1,1)\,\mathbf{g}^{\mathbf{u}_1}(1) + \mathbf{P}^{u_2}(1,2)\,\mathbf{g}^{\mathbf{u}_1}(2) \} \\
&= \arg\min_{u_1,u_2} \{ 0.6 + 0.3\cdot 1.909 + 0.3\cdot 2.4545, \\
&\qquad\qquad 0.85 + 0.4\cdot 1.909 + 0.4\cdot 2.4545 \} \\
&= \arg\min_{u_1,u_2} (\, 1.909, 2.5954 \,) = u_1,
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{u}_2(2) &= \arg\min_{u\in\mathscr{D}(i)} \left\{ \mathbf{r}^u(2) + \sum_{j=1}^{n} \mathbf{P}^u(2,j)\,\mathbf{g}^{\mathbf{u}_1}(j) \right\} \\
&= \arg\min_{u_1,u_2} \{ \mathbf{r}^{u_1}(2) + \mathbf{P}^{u_1}(2,1)\,\mathbf{g}^{\mathbf{u}_1}(1) + \mathbf{P}^{u_1}(2,2)\,\mathbf{g}^{\mathbf{u}_1}(2), \\
&\qquad\qquad \mathbf{r}^{u_2}(2) + \mathbf{P}^{u_2}(2,1)\,\mathbf{g}^{\mathbf{u}_1}(1) + \mathbf{P}^{u_2}(2,2)\,\mathbf{g}^{\mathbf{u}_1}(2) \} \\
&= \arg\min_{u_1,u_2} \{ 0.75 + 0.25\cdot 1.909 + 0.5\cdot 2.4545, \\
&\qquad\qquad 0.8 + 0.3\cdot 1.909 + 0.5\cdot 2.4545 \} \\
&= \arg\min_{u_1,u_2} (\, 2.4545, 2.6 \,) = u_1.
\end{aligned}$$

The policy iteration method terminates since $\mathbf{u}_2 = \mathbf{u}_1 = (u_1, u_1)$ *which implies that* $\mathbf{u}_1$ *is optimal. The optimal total expected reward vector is* $\mathbf{g}^{u_1} = \mathbf{g}^* = (1.909, 2.4545)$.

Modified policy iteration Observe that in the policy evaluation step (5.27) the system of linear equations with a dimension equal to the number of states has to be solved. If the state space is very large this step can be computationally complex, e.g., when standard methods such as Gaussian elemination are applied [27] whose complexity is $\mathscr{O}(n^3)$ for an $n \times n$ input matrix. In the following we describe the method as proposed in [27].

The system of equations (5.27), or equivalently

$$(\mathbf{I} - \mathbf{P}^{\mathbf{u}_k})\,\mathbf{g}^{\mathbf{u}_k} = \mathbf{r}^{\mathbf{u}_k}$$

can be approximated iteratively using a certain number of value iterations. In particular, after determining a $\mathbf{g}^{\mathbf{u}_k}$-improving decision rule in the policy improvement step, the value iteration is performed with the same policy several times.

The algorithm is initialized with an arbitrary vector $\mathbf{g}_0$. Then the policy $\mathbf{u}_0$ satisfying $T_{\mathbf{u}_0}\mathbf{g}_0 = T\mathbf{g}_0$ is determined, i.e., the policy is obtained as $\mathbf{u}_0 = \arg\min_{u\in\mathscr{D}(i)} (\mathbf{r}^u + \mathbf{P}^u\,\mathbf{g}_0)$.

After that policy improvement step the expected rewards $\mathbf{g}^{\mathbf{u}_0}$ are determined using value iteration with the same policy $\mathbf{u}_0$ for several times. The evaluated value $\mathbf{g}^{\mathbf{u}_0}$ of policy $\mathbf{u}_0$ provides the vector $\mathbf{g}_1$ which is used in the next policy improvement step to determine the policy $\mathbf{u}_1$ as described above.

Let $m_0, m_1, \ldots$ be a sequence of positive integers. Then the sequence of values $\mathbf{g}_{k+1}$ is generated by computation

$$\mathbf{g}_{k+1} = (T_{\mathbf{u}_k})^{m_k} \mathbf{g}_k, \tag{5.29}$$

such that the $\mathbf{g}^{\mathbf{u}_k}$-improving policy $\mathbf{u}_{k+1}$ is evaluated using m_k iteration steps according to the current gain vector $\mathbf{g}_k$.

The sequence m_k, for $k = 0, 1, \ldots$, may be chosen according to some heuristic pattern, or uniformly for all policy iterations, i.e. $m_k = m$. Note that if $m = 1$ we obtain the value iteration method [28], and if $m = \infty$ we obtain the policy iteration algorithm with a policy evaluation step which is solved iteratively using the classical value iteration method. It has also been mentioned in the literature [28] that the value iteration step involving only a single policy is less complex than evaluating $T\mathbf{g}^{\mathbf{u}_k}$ which considers all policies. One can also use Gauss-Seidel iterations or any other iterative numerical solution technique to solve Eq. 5.27 in place of evaluating Eq. 5.29.

The sequence $\mathbf{g}_{k+1}$ generated in the *modified policy iteration* method converges monotonically to the optimal gain vector $\mathbf{g}^*$ under the assumption that $T\mathbf{g}_0 \leq \mathbf{g}_0$ unless all policies are proper (see e.g. [148, Theorem 7.2.17], [28]).

Asynchronous policy iteration *Asynchronous policy iteration* is a generalization of the policy iteration scheme where value updates and policy updates are performed for predefined sets of states and can be combined in various ways as described in [28]. The assumption $T_{\mathbf{u}_0}\mathbf{g}_0 \leq \mathbf{g}_0$ is also required here to guarantee that the algorithm yields in the limit the optimal total expected reward and an optimal stationary policy. This can be achieved, e.g. by selecting an arbitrary initial policy $\mathbf{u}_0$ and then obtaining $\mathbf{g}_0 = \mathbf{g}^{\mathbf{u}_0}$.

Let $(\mathbf{g}_k, \mathbf{u}_k)$ be the generated sequence for $k = 0, 1, \ldots$. Then the set of states $\mathscr{S}_k$ is selected such that the new pair $(\mathbf{g}_{k+1}, \mathbf{u}_{k+1})$ can be determined as a value update

$$\mathbf{g}_{k+1}(i) = T_{\mathbf{u}_k}\mathbf{g}_k(i), \text{ for } i \in \mathscr{S}_k, \tag{5.30}$$

and $\mathbf{g}_{k+1}$ is left unchanged for the remaining states

$$\mathbf{g}_{k+1}(i) = \mathbf{g}_k(i), \text{ for } i \notin \mathscr{S}_k. \tag{5.31}$$

In that case the policy remains unchanged, i.e. $\mathbf{u}_{k+1} = \mathbf{u}_k$. Another way to compute the new pair $(\mathbf{g}_{k+1}, \mathbf{u}_{k+1})$ is to determine the next policy

$$\mathbf{u}_{k+1}(i) = \underset{u \in \mathscr{D}(i)}{\arg\min} \left(\mathbf{r}^u(i) + \sum_{j=1}^{n} \mathbf{P}^u(i,j)\, \mathbf{g}_k \right), \text{ for } i \in \mathscr{S}_k, \tag{5.32}$$

and $\mathbf{u}_{k+1}$ is again left unchanged for the remaining states

$$\mathbf{u}_{k+1}(i) = \mathbf{u}_k(i), \text{ for } i \notin \mathscr{S}_k, \tag{5.33}$$

which is called a policy update. In that case the values of the gain vector $\mathbf{g}_{k+1} = \mathbf{g}_k$ remain unchanged.

Now suppose that the value update follows the policy update, and the same set of states $\mathscr{S}_k$ is used in both steps. Since in the policy update step in Eq. 5.32 the policy is computed by minimization in the dynamic programming equation considering all possible actions, the subsequent value iteration computes in fact $\mathbf{g}_{k+1} = T\,\mathbf{g}_k$.

For the case where $\mathscr{S}_k = \mathscr{S}_T$ and m_k value updates are done, Eq. 5.30 becomes Eq. 5.29 and the method implements the modified policy iteration. If $m_k = 1$ value updates are done we obtain value iteration as described above. If $m_k = \infty$ value updates are done before updating the policy, then we have policy iteration method where the policy evaluation step is realized through value iteration.

As next method we consider the variant of asynchronous policy iteration where only subsets of $\mathscr{S}_T$ are used in the calculation. Then policy iteration is performed with one of the subsets $\mathscr{S}_k$ at a time. Suppose that in the k-th iteration the policy improvement in Eq. 5.32 has been performed for the states in $\mathscr{S}_k$ and the resulting policy $\mathbf{u}_{k+1}$ is known. Then the policy evaluation is executed only for the states in $\mathscr{S}_k$. This can be done either using value iteration or using restricted linear programm with unknowns corresponding to the expected total rewards for states in $\mathscr{S}_k$.

It has been shown in [28], that, under the above assumption on the initial pair $(\mathbf{g}_0, \mathbf{u}_0)$ and additionally assuming that value update in Eq. 5.30 and policy update step in Eq. 5.32 are executed infinitely often for all states, the algorithm converges to $\mathbf{g}^*$.

5.3.3. Linear Programming

We now describe how the optimal stationary policy $\mathbf{u}^*$ and the corresponding optimal gain vector $\mathbf{g}^*$ can be computed using *linear programming* as introduced in [148, 27, 28]. As explained above value iteration computes the vector $\mathbf{g}_k(i) = T^k\,\mathbf{g}_0(i)$ in the k-th iteration such that $\lim_{n\to\infty} T^n\mathbf{g} = \mathbf{g}^*$ for all vectors $\mathbf{g}$ (cf. Eq. 2.51).

In fact, dynamic programming algorithm is a system of equations with one equation per state i

$$\mathbf{g}(i) \leq \min_{u\in\mathscr{D}(i)} \left(\mathbf{r}^u(i) + \sum_{j=1}^{n} \mathbf{P}^u(i,j)\,\mathbf{g}(j) \right), \tag{5.34}$$

the solution of the system of equations for $i = 1,\ldots,n$ is the minimal total expected reward for all states in the one-stage problem. According to the monotonicity property of the dynamic programming [27, p. 376] we obtain $\mathbf{g}_k(i) \leq \mathbf{g}_{k+1}(i) = T\mathbf{g}_k(i)$ for all k and i. Incorporating the convergence criterion given in (2.51) it holds that

$$\mathbf{g}_k \leq \mathbf{g}^* = T\mathbf{g}^*,$$

where the inequality is satisfied elementwise for an arbitrary k. It implies that the elements of the optimal reward vector belong to the "largest" vector $\mathbf{g}$ satisfying the inequality

$$\mathbf{g}(i) \leq \mathbf{r}^u(i) + \sum_{j=1}^{n} \mathbf{P}^u(i,j)\,\mathbf{g}(j), \tag{5.35}$$

for all $i \in \mathscr{S}_T$ and $u \in \mathscr{D}(i)$. Then the system of resulting inequalities depicts a polyhedron in $\mathbb{R}^n$ the northeast corner of which represents the optimal solution corresponding to $\mathbf{g}^*$.

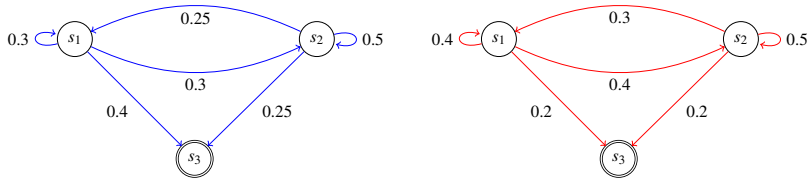
Since $\mathbf{g}^*$ is the "largest" vector $\mathbf{g}$ the objective function of the linear programm is to maximize the sum of its elements

$$\sum_{i \in \mathscr{S}_T} \mathbf{g}(i),$$

subject to the constraints given in Eq. 5.35. The vector elements $\mathbf{g}^*(1), \ldots, \mathbf{g}^*(n)$ represent the solution of the proposed linear program.

As described in [148, 27, 28] the defined linear program contains n variables and $n * u_{max}$ constraints, where u_{max} is the maximal cardinality of the sets $\mathscr{D}(i)$, the dimension of this programm can be very large for large n and u_{max}. Then special large-scale linear programming methods (e.g., interior-point algorithms) are required to keep practicability of this approach.

***Example* 5.6.** *Consider the SSPP instance in example 2.5 which has already been solved via value iteration and policy iteration methods. The policies* $\mathbf{u}_1$, $\mathbf{u}_2$ *with* $\mathbf{u}_j(i) = u_j$ *for* $j = 1,2$ *and all* $i \in \mathscr{S}_T$, *induce the Markov chains visualized in Fig. 5.7.*



Policy $\mathbf{u}_1$ with $\mathbf{r}^{u_1} = (0.6, 0.75, 0)^T$ Policy $\mathbf{u}_2$ with $\mathbf{r}^{u_2} = (0.85, 0.8, 0)^T$

Figure 5.7.: Transition diagrams for policies $\mathbf{u}_1$, $\mathbf{u}_2$ in the SSPP instance.

The corresponding linear program is given by

$$\textit{minimize} \; -\mathbf{g}(1) - \mathbf{g}(2)$$

subject to

$$\begin{aligned}
\mathbf{g}(1) &\leq 0.6 + (0.3\mathbf{g}(1) + 0.3\mathbf{g}(2)) \\
\mathbf{g}(2) &\leq 0.75 + (0.25\mathbf{g}(1) + 0.5\mathbf{g}(2)) \\
\mathbf{g}(1) &\leq 0.85 + (0.4\mathbf{g}(1) + 0.4\mathbf{g}(2)) \\
\mathbf{g}(2) &\leq 0.8 + (0.3\mathbf{g}(1) + 0.5\mathbf{g}(2)),
\end{aligned}$$

where the constraints are constructed according to the reward vectors $\mathbf{r}^{u_1}$, $\mathbf{r}^{u_2}$ *and the visualized transition diagrams in Fig. 5.7. The optimal solution vector is*

$$\mathbf{g}^* = \mathbf{g} = (1.9091, 2.4545).$$

5.4. Complexity of Solving MDPs

In the following we summarize the main computational complexity results for infinite horizon MDPs according to a review given in [91] and references therein. Observe that any MDP can be transformed to a linear program and solved in a weakly polynomial time. More precisely, using linear programming MDPs can be solved in a number of arithmetic operations polynomial in s, d, and b [117], where $s = |\mathscr{S}|$, $d = |\mathscr{D}|$, and b is the maximum number of bits required to represent the input data. Then general LP solution techniques, as e.g., Ellipsoid method or interior-point algorithm can be applied [103, 107, 178]. However, the mentioned algorithms are often impractical for solving MDPs [117].

Since MDPs and thus also SSPPs can be reformulated as a linear program, the simplex method [57] can be used to solve it. However, even though the simplex results in exponential running times in the worst case, the method performs very well in practice [117]. It is still an open question if there exist a pivoting rule that results in polynomial time simplex for solving general linear programs. Note that several pivoting rules have been shown to result in exponential number of iterations, which may not necessarily hold for linear programs for solving SSPPs [117]. This is due to the fact that linear programming techniques do not exploit the special structure of SSPPs.

A significant result on the MDP complexity field was obtained in [178], where it has been shown that the policy iteration including the simplex method with most-negative-reduced-cost pivoting rule is a strongly polynomial time algorithm for solving MDPs with a fixed discount rate $0 \leq \gamma < 1$. The author showed that the simplex method terminates after at most $\mathscr{O}(\frac{sd}{1-\gamma}\log\left(\frac{s}{1-\gamma}\right))$ iterations. Interestingly, strongly polynomial time algorithms for deterministic MDPs also exist [142, 83], where a quadaratic lower bound of deterministic problems has been shown. To the best of our knowledge, the question if SSPPs which belong to the class of undiscounted MDPs can be solved in strongly polynomial time still remains open.

In the following, we give a short overview of complexity results for standard MDP methods according to [91] and references therein. The policy iteration method is the most used iterative approach for solving MDPs [97]. The improvement step can be performed in $\mathscr{O}(d\,s^2)$. The evaluation of the current policy, i.e. computing of its value requires inverting a $s \times s$ matrix which takes $\mathscr{O}(s^{2.376})$ operations [117].

Since PI is guaranteed to find the optimal or ε-optimal solution in a finite number of iterations, its complexity depends on their bounds. In principle, there are d^s distinct policies in a MDP, such that greedy PI requires at most an exponential number of iterations until convergence [148, 117]. In each iteration PI computes a set of actions for which an improvement can be achieved, such that a subset of these switchable actions can be determined. Greedy PI updates every state with switchable actions [74].

As stated in [117], the number of iterations depend on the sequential improvement rule of PI. When, e.g., a policy is updated accordingly to the smallest index rule, PI can take an exponential number of iterations to converge (see [117] and references therein).

In [122] an upper bound of $\mathscr{O}(\frac{d^s}{s})$ for the greedy PI is proved. Later results provide an exponential lower bound for the greedy PI with the average reward and the total reward criterion [74].

For discounted MDPs with a fixed $0 \leq \gamma < 1$ a strongly polynomial time algorithm exists [178]. It has been shown that for a fixed discount rate $0 \leq \gamma < 1$ the number of iterations in the policy iteration with the most-negative-reduced-cost pivoting rule is bounded by $\left(\frac{s^2(d-1)}{1-\gamma} \log\left(\frac{s^2}{1-\gamma}\right)\right)$. For discounted infinite horizon MDPs the results in [148, 117] show that the policy iteration needs at least as many iterations as the value iteration to compute the optimal policy. However, [117] gives an example for which the number of iterations of the value iteration is bounded by $\left(\frac{1}{1-\gamma}\log(\frac{1}{1-\gamma})\right)$ in worst case. Concluding, policy iteration and value iteration can converge in polynomial time for MDPs with fixed discount rate. In contrast, the complexity results for undiscounted MDPs and thus for SSPPs state that policy iteration runs in a weakly polynomial time under particular assumptions needed to compute the optimal policy [170, 148].

5.5. Solving of SSPP with Correlations

In this section we briefly describe solution methods adapted to SSPPs with correlations. First we give a solution for finding a minimal expected shortest path between v_{ini} and v_{fin}. Let $\mathbf{u}$ be a stationary policy. Then its total expected reward is given by

$$\mathbf{g}^{\mathbf{u}} = \sum_{t=0}^{\infty} (\mathbf{P}^{\mathbf{u}})^t \, \mathbb{1} = \mathbf{N}^{\mathbf{u}} \mathbb{1}, \tag{5.36}$$

where the matrix $\mathbf{N}^{\mathbf{u}}$ is defined in Eq. 5.8. The value $\mathbf{g}^{\mathbf{u}}(i,x)$ is the expected weight of the path from state (i,x) to the absorbing state $(0,0)$ when decision vector $\mathbf{u}$ is used. For the proper policy $\mathbf{u}$ the total expected reward function can be computed as (cf. Eq. 2.54)

$$(\mathbf{I} - \mathbf{P}^{\mathbf{u}})\,\mathbf{g}^{\mathbf{u}} = \mathbb{1}, \tag{5.37}$$

or accordingly to Eq. 2.56 as

$$\mathbf{g}^{\mathbf{u}} = (\mathbf{I} - \mathbf{P}^{\mathbf{u}})^{-1} \mathbb{1}. \tag{5.38}$$

As already defined in Eq. 2.33 the minimal total expected reward starting in state (i,x) is given by

$$\mathbf{g}^*(i,x) = \min_{\mathbf{u} \in \Pi} \mathbf{g}^{\mathbf{u}}(i,x), \tag{5.39}$$

and the corresponding optimal stationary policy satisfying

$$\mathbf{u}^*(i,x) = \arg\min_{u \in \mathscr{D}(i)} \left(\mathbf{g}^u(i,x)\right). \tag{5.40}$$

The optimal stationary policy $\mathbf{u}^*$ and the corresponding gain vector $\mathbf{g}^*$ can be computed using methods described in Sec. 5.3.1 - Sec. 5.3.3. In the following we first describe the policy iteration method and then give an approach for computing an optimal policy in dependence of the realized edge traveling times.

Policy Iteration Approach Policy iteration starts with an arbitrary proper policy $\mathbf{u}_0$ which can be obtained by running a shortest path algorithm on the instance where PHDs are substituted by their expectations and correlations are completely neglected.

Then the sequence of improved policies $\mathbf{u}_1, \mathbf{u}_2, \ldots$ is generated such that $\mathbf{g}^{\mathbf{u}_1}(i,x) \geq \mathbf{g}^{\mathbf{u}_2}(i,x) \geq \ldots$ for all $(i,x) \in \mathscr{S}$ (see Prop. 5.1).

In the k-th policy evaluation step the reward function $T_{\mathbf{u}_k}\mathbf{g}^{\mathbf{u}_k}$ associated with a proper policy $\mathbf{u}_k$ is computed as given in Eq. 2.49

$$\mathbf{g}^{\mathbf{u}_k} = T_{\mathbf{u}_k}\mathbf{g}^{\mathbf{u}_k} = \mathbb{1} + \mathbf{P}^{\mathbf{u}_k}\mathbf{g}^{\mathbf{u}_k}, \tag{5.41}$$

such that the gain vector $\mathbf{g}^{\mathbf{u}_k}$ obtained is the solution of the linear system of equations given $\mathbf{u}_k$ (5.27)

$$\mathbf{g}^{\mathbf{u}_k}(i,x) = 1 + \sum_{j \in \mathscr{S}} \mathbf{P}^{\mathbf{u}_k}((i,x),(j,y))\,\mathbf{g}^{\mathbf{u}_k}(j,y), \text{ for all } (i,x) \in \mathscr{S}. \tag{5.42}$$

Observe that Eq. 5.42 can also be written as (see Eq. 5.37)

$$(\mathbf{I} - \mathbf{P}^{\mathbf{u}_k})\,\mathbf{g}^{\mathbf{u}_k} = \mathbb{1}. \tag{5.43}$$

After the value $\mathbf{g}^{\mathbf{u}_k}$ of the policy $\mathbf{u}_k$ is known the policy improvement is applied to obtain the improved policy $\mathbf{u}_{k+1}$ satisfying the equation $T_{\mathbf{u}_{k+1}}\mathbf{g}^{\mathbf{u}_k} = T\,\mathbf{g}^{\mathbf{u}_k}$

$$\mathbf{u}_{k+1}(i,x) = \arg\min_{u \in \mathscr{D}(i)} \left(1 + \sum_{(j,y) \in \mathscr{S}} \mathbf{P}^{u}((i,x),(j,y))\,\mathbf{g}^{\mathbf{u}_k}(j,y) \right). \tag{5.44}$$

The Algorithm 5.1 iterates between policy evaluation and policy improvement steps until $\mathbf{g}^k(i,x) = \mathbf{g}^{k-1}(i,x)$ for all $(i,x) \in \mathscr{S}$ and some k.

Algorithm 5.1: Computing the optimal stationary policy $\mathbf{u}^*$ and the gain vector $\mathbf{g}^*$

1: Initialize matrices $\mathbf{P}^{\mathbf{u}}$ for all $\mathbf{u} \in \mathscr{P}$;
2: Set $k = 0$ and compute $\mathbf{u}_k$ using shortest path algorithm;
3: **repeat**
4: Compute gain vector $\mathbf{g}^{\mathbf{u}_k}$ using Eq. 5.43; ▷ *policy evaluation*
5: Compute policy $\mathbf{u}_{k+1}$ for all $(i,x) \in \mathscr{S}$ using Eq. 5.44; ▷ *policy improvement*
6: $k = k + 1$
7: **until** $\mathbf{g}^k(i,x) = \mathbf{g}^{k-1}(i,x)$ for all $(i,x) \in \mathscr{S}$
8: Terminate with $\mathbf{u}^* = \mathbf{u}_k$ and $\mathbf{g}^* = \mathbf{g}^{\mathbf{u}_k}$.

Observe that the value $\mathbf{g}^*(i,x)$ is the minimal expected number of steps until absorption starting from state (i,x). Since the absorbing state is interpreted as the destination v_{fin} the optimal solution $\mathbf{u}^*$ corresponds to the fastest path on the average. Knowing the values $\mathbf{u}^*$ and $\mathbf{g}^*$ the optimal decisions depend on the state of the CTMDP.

Let $\mathbf{a}_i = (\mathbf{0}_{<i}, \pi_i, \mathbf{0}_{>i})$ be the initial distribution vector for the initial edge $i \in E_{ini}$ where zeros vector $\mathbf{0}_{<i}$ is of length $\sum_{j \in E,\, j<i} n_j$, and zeros vector $\mathbf{0}_{>i}$ is of length $\sum_{j \in E,\, j>i} n_j$. Then the minimal expected weight of the path starting from $ini(i) = v_{ini}$ to v_{fin} is computed as

$$\xi_i = \mathbf{a}_i\,\mathbf{g}^*. \tag{5.45}$$

The optimal initial edge can be chosen as

$$i^* = \arg\min_{i \in E_{ini}} (\xi_i), \tag{5.46}$$

and the corresponding optimal minimal expected path weight is

$$\xi^* = \mathbf{a}_{i^*}\,\mathbf{g}^*, \tag{5.47}$$

such that the value ξ^*/α corresponds to the *minimum expected time* until absorption, i.e. to the minimum expected traveling time until the destination v_{fin} is reached.

Now assume that the history of realized values $(w_1,\ldots,w_l)$ become known to the vehicle as it traverses the path to the destination. Assume that it arrives at node $fin(i_l)$. Then the decisions of the vehicle should be based on history vectors $\bar{\psi}_{(i_1,w_1,\ldots,i_l,w_l)}$ defined in Eq. 5.16. In particular, at node $fin(i_l)$ the decision on the next edge to traverse is based on

$$i^* = \arg\min_{j\in\mathscr{D}(i_l)} \left(\sum_{x=1}^{n_{i_l}} \bar{\psi}_{(i_1,w_1,\ldots,i_l,w_l)}(i_l,x) \cdot \left(\sum_{y=1}^{n_{i_l}} \mathbf{P}^j((i_l,x),(i_l,y))\mathbf{g}^*(i_l,y) + \sum_{y=1}^{n_j} \mathbf{P}^j((i_l,x),(j,y))\mathbf{g}^*(j,y) \right) \right). \tag{5.48}$$

***Example* 5.7.** *We consider a simple graph visualized in Fig. 5.8 where two nodes allow choices between two outgoing edges. The weights of all edges are described by the following* 2*-order hyperexponential PHD*

$$\pi = (0.5, 0.5),\ \mathbf{D}_1 = \begin{pmatrix} -7 & 0 \\ 0 & -0.5 \end{pmatrix},$$

which has the mean 1.0714 *and the squared coefficient of variation* 2.5*.*

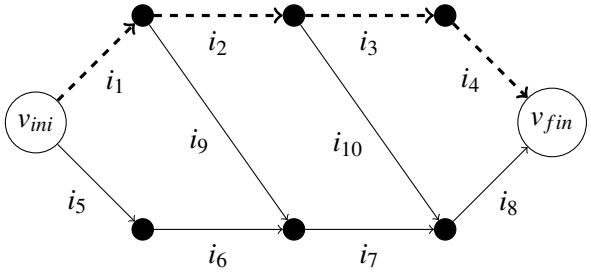


Figure 5.8.: Acyclic graph for shortest path computation. Edges with correlated weights are highlighted in dashed style.

The mean weight of every path between v_{ini} and v_{fin} is 4.2857*. The weights of the edges along the upper path are positively correlated with correlation coefficient $\rho = 0.3$. The weights of the edges i_1 and i_9, i_2 and i_{10} are negatively correlated with correlation coefficient $\rho = -0.3$. Furthermore, the weights of the edges i_5, i_6, i_7, i_8 are uncorrelated. The following transfer matrices* **H** *are chosen*

$$\mathbf{H}^{\rho=0.3}_{i_k\,i_l} = \begin{pmatrix} 7 & 0 \\ 0 & 0.5 \end{pmatrix},\ \mathbf{H}^{\rho=-0.3}_{i_m\,i_n} = \begin{pmatrix} 0 & 7 \\ 0.5 & 0 \end{pmatrix},\ \mathbf{H}^{\rho=0}_{i_s\,i_t} = \begin{pmatrix} 3.5 & 3.5 \\ 0.25 & 0.25 \end{pmatrix},$$

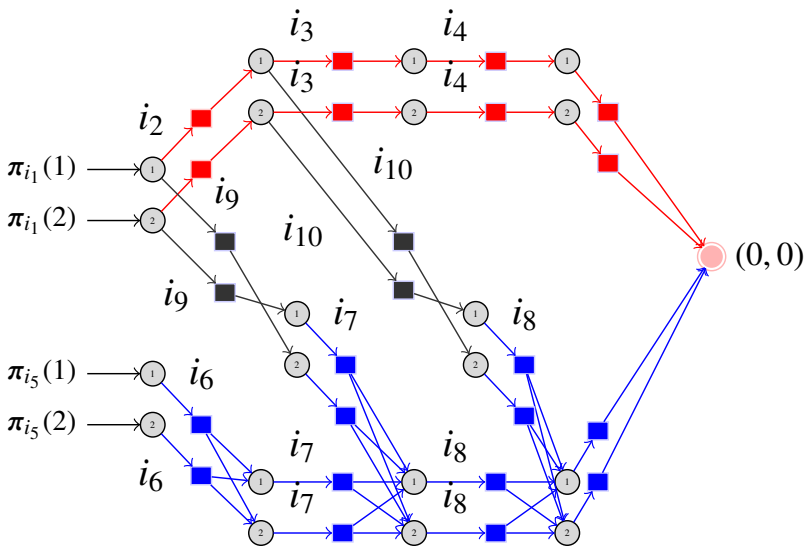


Figure 5.9.: The PHG corresponding to the graph in Fig. 5.8.

for $k = 1,2,3$, $l = 2,3,4$, $m = 1,2$, $n = 9,10$, $s = 5,6,7$, *and* $t = 6,7,8$. *The path matrices are given by*

$$\mathbf{Q}_{(i_1,i_2,i_3,i_4)} = \begin{pmatrix} -7 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{pmatrix},$$

$$\mathbf{Q}_{(i_1,i_9,i_7,i_8)} = \begin{pmatrix} -7 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 0 & 3.5 & 3.5 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0.25 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 3.5 & 3.5 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{pmatrix},$$

$$\mathbf{Q}_{(i_1,i_2,i_{10},i_8)} = \begin{pmatrix} -7 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 3.5 & 3.5 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{pmatrix},$$

$$\mathbf{Q}_{(i_5,i_6,i_7,i_8)} = \begin{pmatrix} -7 & 0 & 3.5 & 3.5 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0.25 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 0 & 3.5 & 3.5 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0.25 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 3.5 & 3.5 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{pmatrix}.$$

Optimal policy *The behavior of the optimal policy to minimize the mean travel time from v_{ini} to v_{fin} is as follows. For a short time required on the edge, the optimal choice is the upper edge to exploit the positive correlation. In turn, if the time on an edge becomes longer, it is better to choose the down edge to exploit the effect of negative correlation. The initial policy is defined such that in the upper path always the lower edge is chosen. The policy for the lower path chooses always the lower edge.*

Policy Iteration *Consider the following initial policy*

$$\begin{aligned} &\mathbf{u}_0(i_1,1) = i_9, \quad \mathbf{u}_0(i_2,1) = i_{10}, \quad \mathbf{u}_0(i_3,1) = i_4, \\ &\mathbf{u}_0(i_1,2) = i_9, \quad \mathbf{u}_0(i_2,2) = i_{10}, \quad \mathbf{u}_0(i_3,2) = i_4, \\ &\mathbf{u}_0(i_5,1) = i_6, \quad \mathbf{u}_0(i_6,1) = i_7, \quad \mathbf{u}_0(i_7,1) = i_8, \\ &\mathbf{u}_0(i_5,2) = i_6, \quad \mathbf{u}_0(i_6,2) = i_7, \quad \mathbf{u}_0(i_7,2) = i_8, \\ &\mathbf{u}_0(i_9,1) = i_7, \quad \mathbf{u}_0(i_{10},1) = i_8, \\ &\mathbf{u}_0(i_9,2) = i_7, \quad \mathbf{u}_0(i_{10},2) = i_8 \end{aligned}$$

The weight of the policy for the uncorrelated lower path $(i_5,\ i_6,\ i_7,\ i_8)$ is given by

	$(i_5,1)$	$(i_5,2)$	$(i_6,1)$	$(i_6,2)$	$(i_7,1)$	$(i_7,2)$	$(i_8,1)$	$(i_8,2)$
Vector $\mathbf{g}^{u_0}$	3.357	5.2143	2.2857	4.1429	1.2143	3.071	0.1429	2

The policy $\mathbf{u}_0$ for the path $(i_1,\ i_9,\ i_7,\ i_8)$ results in the following gain

	$(i_1,1)$	$(i_1,2)$	$(i_9,1)$	$(i_9,2)$	$(i_7,1)$	$(i_7,2)$	$(i_8,1)$	$(i_8,2)$
Vector $\mathbf{g}^{u_0}$	4.2857	4.2857	2.2857	4.1429	1.2143	3.0714	0.1429	2

Finally, the policy weights of the remaining states are given by

	$(i_2,1)$	$(i_2,2)$	$(i_{10},1)$	$(i_{10},2)$	$(i_3,1)$	$(i_3,2)$	$(i_4,1)$	$(i_4,2)$
Vector $\mathbf{g}^{u_0}$	3.2143	3.2143	1.2143	3.0714	0.2857	4	0.1429	2

In the policy improvement step the values $\mathbf{u}_1(i_2,1)$, $\mathbf{u}_1(i_2,2)$ satisfying

$T_{u_1}\,\mathbf{g}^{\mathbf{u}_0} = T\,\mathbf{g}^{\mathbf{u}_0}$ *are computed as*

$$\begin{aligned}
\mathbf{u}_1(i_2,1) &= \underset{u\in\{i_{10},\,i_3\}}{\arg\min}\left\{\,1+\sum_{j=1}^{n_{i_2}}\mathbf{P}^{u}((i_2,1),j)\,\mathbf{g}^{\mathbf{u}_0}(j)\,\right\}\\
&= \underset{i_{10},i_3}{\arg\min}\,\{\,1+\mathbf{P}_{i_1,i_2,i_{10},i_8}((i_2,1),(i_{10},2))\,\mathbf{g}^{\mathbf{u}_0}(i_{10},2),\\
&\qquad\qquad 1+\mathbf{P}_{i_1,i_2,i_3,i_4}((i_2,1),(i_3,1))\,\mathbf{g}^{\mathbf{u}_0}(i_3,1)\,\}\\
&= \underset{i_{10},i_3}{\arg\min}\,\{\,1+1\cdot 3.0714,1+1\cdot 0.2857\,\}\\
&= \underset{i_{10},i_3}{\arg\min}\,(\,4.0714,1.2857\,) = i_3,
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{u}_1(i_2,2) &= \underset{u\in\{i_{10},\,i_3\}}{\arg\min}\left\{\,1+\sum_{j=1}^{n_{i_2}}\mathbf{P}^{u}((i_2,2),j)\,\mathbf{g}^{\mathbf{u}_0}(j)\,\right\}\\
&= \underset{i_{10},i_3}{\arg\min}\,\{\,1+\mathbf{P}_{i_1,i_2,i_{10},i_8}((i_2,2),(i_{10},1))\,\mathbf{g}^{\mathbf{u}_0}(i_{10},1)+\\
&\qquad\qquad \mathbf{P}_{i_1,i_2,i_{10},i_8}((i_2,2),(i_2,2))\,\mathbf{g}^{\mathbf{u}_0}(i_2,2),\\
&\qquad\qquad 1+\mathbf{P}_{i_1,i_2,i_3,i_4}((i_2,2),(i_3,2))\,\mathbf{g}^{\mathbf{u}_0}(i_3,2)+\\
&\qquad\qquad \mathbf{P}_{i_1,i_2,i_3,i_4}((i_2,2),(i_2,2))\,\mathbf{g}^{\mathbf{u}_0}(i_2,2)\,\}\\
&= \underset{i_{10},i_3}{\arg\min}\,\{\,1+0.0714\cdot 1.2143+0.9286\cdot 3.0714,\\
&\qquad\qquad 1+0.0714\cdot 4+0.9286\cdot 3.0714\,\}\\
&= \underset{i_{10},i_3}{\arg\min}\,(\,3.0715,3.2041\,) = i_{10}.
\end{aligned}$$

Analogously, the values $\mathbf{u}_1(i_1,1)$, $\mathbf{u}_1(i_1,2)$ *are computed as*

$$\begin{aligned}
\mathbf{u}_1(i_1,1) &= \underset{u\in\{i_9,\,i_2\}}{\arg\min}\left\{\,1+\sum_{j=1}^{n_{i_1}}\mathbf{P}^{u}((i_1,1),j)\,\mathbf{g}^{\mathbf{u}}(j)\,\right\}\\
&= \underset{i_9,i_2}{\arg\min}\,\{\,1+\mathbf{P}_{i_1,i_9,i_7,i_8}((i_1,1),(i_9,2))\,\mathbf{g}^{\mathbf{u}_0}(i_9,2),\\
&\qquad\qquad 1+\mathbf{P}_{i_1,i_2,i_3,i_4}((i_1,1),(i_2,1))\,\mathbf{g}^{\mathbf{u}_1}(i_2,1)\,\}\\
&= \underset{i_9,i_2}{\arg\min}\,\{\,1+1\cdot 4.1429,1+1\cdot 1.2857\,\}\\
&= \underset{i_9,i_2}{\arg\min}\,(\,5.1429,2.2857\,) = i_2,
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{u}_1(i_1,2) &= \arg\min_{u\in\{i_9,\,i_2\}}\left\{\,1+\sum_{j=1}^{n_{i_1}}\mathbf{P}^{u}((i_1,2),j)\,\mathbf{g}^{\mathbf{u}}(j)\,\right\}\\
&= \arg\min_{i_9,i_2}\{\,1+\mathbf{P}_{i_1,i_9,i_7,i_8}((i_1,2),(i_9,1))\,\mathbf{g}^{\mathbf{u}_0}(i_9,1)+\\
&\qquad\qquad \mathbf{P}_{i_1,i_9,i_7,i_8}((i_1,2),(i_1,2))\,\mathbf{g}^{\mathbf{u}_0}(i_1,2),\\
&\qquad 1+\mathbf{P}_{i_1,i_2,i_3,i_4}((i_1,2),(i_2,2))\,\mathbf{g}^{\mathbf{u}_1}(i_2,2)+\\
&\qquad\qquad \mathbf{P}_{i_1,i_2,i_3,i_4}((i_1,2),(i_1,2))\,\mathbf{g}^{\mathbf{u}_0}(i_1,2)\,\}\\
&= \arg\min_{i_9,i_2}\{\,1+0.0714\cdot 2.2857+0.9286\cdot 4.2857,\\
&\qquad\qquad 1+0.0714\cdot 3.0715+0.9286\cdot 4.2857\,\}\\
&= \arg\min_{i_9,i_2}(\,5.1429,5.1990\,) = i_9,
\end{aligned}$$

where uniformized path matrices are given in Sec. D.1. Observe that PI algorithm requires only 2 *iterations to find the optimal policy of the instance in this example.*

State-based optimal decisions *One can see that the optimal decisions depend on either the process is in the long phase* 2 *or in the short phase* 1 *of the PHD for an edge. For a long time required on the edge* i_2 *, the best decision is to choose the adjacent edge* i_{10}*, since the weights of* i_2 *and* i_{10} *are negatively correlated. If the exit phase of the PHD for the edge* i_2 *was the long phase* 2*, the process switches to the short phase* 1 *of the edge* i_{10} *with the full rate* 0.5*.*

If the time required for the edge i_2 *becomes smaller, the optimal decision is to choose the adjacent edge* i_3*. The reason is that the weights of the edges* i_2 *and* i_3 *are positively correlated. If the on the average shorter phase* 1 *was the exit phase of the PHD for the edge* i_2*, the process switches to the likewise on the average shorter phase* 1 *of the adjacent edge* i_3 *with the full exit rate* 7*.*

The expected travel times to the destination node v_{fin} *and the optimal decisions of the successor edge in dependence of the phase of the PHD are given in Table 7.2. From Table 7.2 one can see that the optimal policy can change based on the congestion level of the traversed edge.*

History-based optimal decisions *In the real system decisions should depend on realized values and not on the phase of the PHD. We computed the history vectors* $\bar{\Psi}_{(i_2,w_2,i_l,0)}$ *with* $w_2\in[0,2]$*,* $l\in\{3,10\}$*, and* $\bar{\Psi}_{(i_1,w_1,i_m,0)}$ *with* $w_1\in[0,2]$*,* $m\in\{2,9\}$ *using formula* (5.15)-(5.16)*. Using the history vectors the conditional weights of the remaining paths* (i_{10},i_8)*,* (i_3,i_4)*,* (i_9,i_7,i_8)*,* (i_2,i_3,i_4) *can be computed using* (5.17)*, such that decisions depend on the previous weights and not on the state of the PHD.*

Suppose that the node $fin(i_2)$ *is reached, such that the realization of the weight* w *of* i_2 *is known now. We consider two cases, namely* $w=0.3$*, and* $w=1.5$*, which results in the history vectors shown in Tab. 5.2.*

Using (5.48) *the optimal decision based on the realized time value* $w=0.3$ *can be*

Table 5.1.: Mean travel times to the destination node v_{fin} depending on the exit phase of the PHD for the edge i_2 and i_1, and optimal decisions of successor edges.

	Mean Travel Time to v_{fin}		
Exit Phase of PHD for i_2	i_3	i_{10}	Optimal Successor Edge
Phase 1	0.2857	1.2143	i_3
Phase 2	4.0	3.0714	i_{10}
Exit Phase of PHD for i_1	i_2	i_9	Optimal Successor Edge
Phase 1	0.4286	2.2857	i_2
Phase 2	6.0	4.1429	i_9

Table 5.2.: Realizations of weight w of the edge i_2 and the corresponding history vectors. Here $\rho_{i_2\,i_3} = 0.3$ and $\rho_{i_2\,i_{10}} = -0.3$.

Successor edge	$w = 0.3$	$w = 1.5$
i_3	$\bar{\Psi}_{(i_2,0.3,i_3,0)} = (0.6658, 0.3342)$	$\bar{\Psi}_{(i_2,1.5,i_3,0)} = (0.0008, 0.9992)$
i_{10}	$\bar{\Psi}_{(i_2,0.3,i_{10},0)} = (0.3342, 0.6658)$	$\bar{\Psi}_{(i_2,1.5,i_{10},0)} = (0.9992, 0.0008)$

determined as follows

$$\begin{aligned} i^* = \underset{u\in\{i_{10},\,i_3\}}{\arg\min}\{ &(\bar{\Psi}_{(i_2,0.3,i_{10},0)}(i_{10},1)\cdot \mathbf{P}_{i_1,i_2,i_{10},i_8}((i_{10},1),:)\cdot \mathbf{g}^* + \\ &\bar{\Psi}_{(i_2,0.3,i_{10},0)}(i_{10},2)\cdot \mathbf{P}_{i_1,i_2,i_{10},i_8}((i_{10},2),:)\cdot \mathbf{g}^*), \\ &(\bar{\Psi}_{(i_2,0.3,i_3,0)}(i_3,1)\cdot \mathbf{P}_{i_1,i_2,i_3,i_4}((i_3,1),:)\cdot \mathbf{g}^* + \\ &\bar{\Psi}_{(i_2,0.3,i_3,0)}(i_3,2)\cdot \mathbf{P}_{i_1,i_2,i_3,i_4}((i_3,2),:)\cdot \mathbf{g}^*)\} \\ &= \{2.3080, 1.0960\} = i_3, \end{aligned}$$

and for $w = 1.5$

$$\begin{aligned} i^* = \underset{u\in\{i_{10},\,i_3\}}{\arg\min}\{ &(\bar{\Psi}_{(i_2,1.5,i_{10},0)}(i_{10},1)\cdot \mathbf{P}_{i_1,i_2,i_{10},i_8}((i_{10},1),:)\cdot \mathbf{g}^* + \\ &\bar{\Psi}_{(i_2,1.5,i_{10},0)}(i_{10},2)\cdot \mathbf{P}_{i_1,i_2,i_{10},i_8}((i_{10},2),:)\cdot \mathbf{g}^*), \\ &(\bar{\Psi}_{(i_2,1.5,i_3,0)}(i_3,1)\cdot \mathbf{P}_{i_1,i_2,i_3,i_4}((i_3,1),:)\cdot \mathbf{g}^* + \\ &\bar{\Psi}_{(i_2,1.5,i_3,0)}(i_3,2)\cdot \mathbf{P}_{i_1,i_2,i_3,i_4}((i_3,2),:)\cdot \mathbf{g}^*)\} \\ &= \{1.0729, 2.9926\} = i_{10}. \end{aligned}$$

Under the optimal policy, the mean traveling time from v_{ini} *to* v_{fin} *is* 0.5714 *rather than* 3.3571 *which is the mean traveling time assuming independent edge weights. The mean duration* 0.5714 *is the result of starting from the state* $(i_1, 1)$ *and following the path* (i_1, i_2, i_3, i_4)*. If we start from the state* $(i_1, 2)$ *the mean traveling time to the*

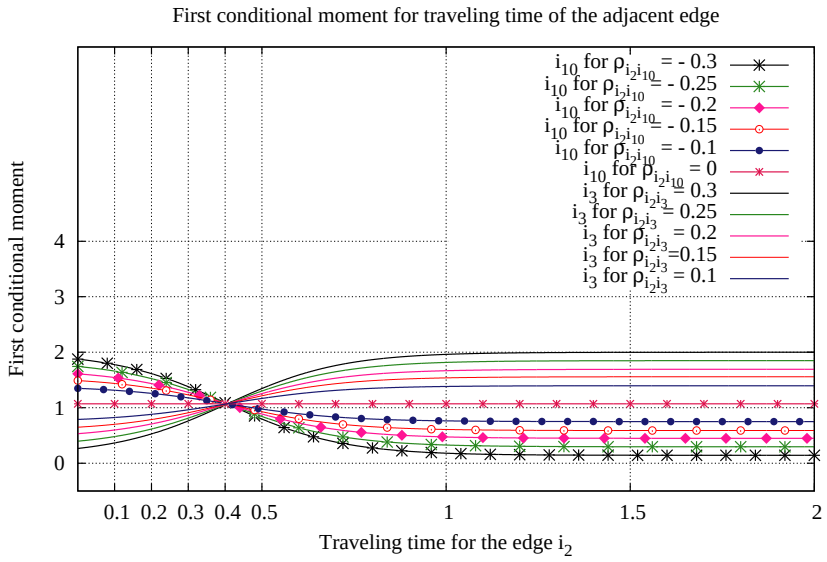


Figure 5.10.: The expected traveling time for the adjacent edges i_{10}, and i_3 depending on the weight of the edge i_2 and the correlation coefficients $\rho_{i_2 i_{10}}$, and $\rho_{i_2 i_3}$.

destination v_{fin} is 8 *rather than* 5.2143 *which is the mean duration assuming independent weights. In that case the route containing independent edge weights is the better choice.*

Effect of correlation *The values of the first conditional moment of the traveling time for the adjacent edges are shown in Fig. 5.10. Here the effect for different values of correlation coefficients $\rho_{i_2 i_{10}}$, $\rho_{i_2 i_3}$ is summarized graphically.*

One can see that the highest and the lowest correlation assures the maximal deviation from the first conditional moment of the uncorrelated successor edge. Furthermore the effect of the positive or negative correlation can be exploited for already small correlation coefficients.

The values of the first conditional moments of the traveling time for the remaining paths are visualized in Fig. 5.11, 5.12.

5.6. Computation of the Probability of Arriving on Time

In this section we turn to the question how the path with maximal/minimal probability of reaching a destination within a given deadline value can be computed. The problem results in the analysis of a CTMDP in the finite interval $[0,T]$ where T corresponds to the given deadline. We first introduce the basic algorithm to compute the maximal/minimal accumulated reward in a CTMDP over the finite interval as proposed

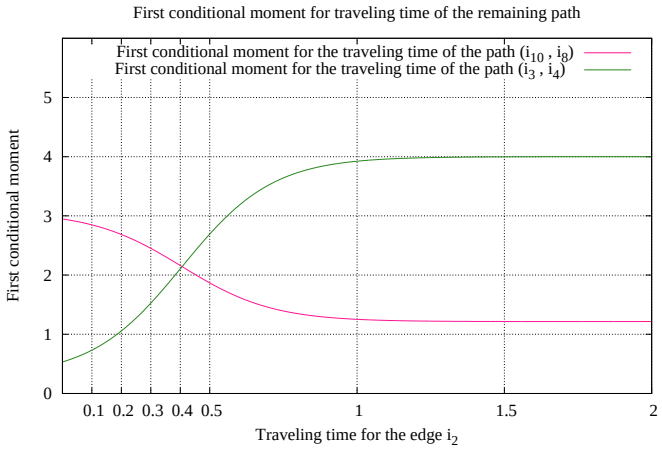


Figure 5.11.: The expected traveling time for the paths (i_{10}, i_8), and (i_3, i_4) depending on the weight of the edge i_2.

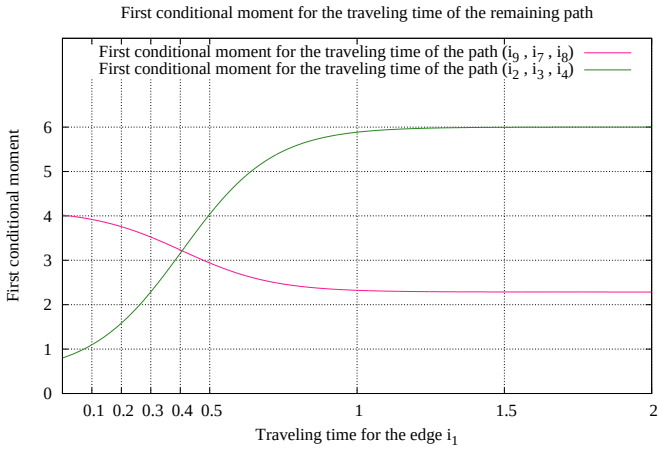


Figure 5.12.: The expected traveling time for the paths (i_9, i_7, i_8), and (i_2, i_3, i_4) depending on the weight of the edge i_1.

in the literature [127, 42, 49]. Then we discuss numerical approximation algorithms which generate the optimal policy maximizing/minimizing the expected gain accumulated in the finite interval. In the following we consider a maximization problem.

Now suppose that we consider a CTMDP for a PHG defined in Sec. 5.1 in the interval $[0,T]$. Recall that for the given measurable policy $\mathbf{d}$ the matrix $\mathbf{V}^{\mathbf{d}}_{r,t}$ with $0 \leq r \leq t \leq T$ is the Markov transition matrix (see Eq. 2.18). The value at position $\mathbf{V}^{\mathbf{d}}_{r,t}(i,j)$ gives the probability that the CTMDP is at time t in state j under the condition that it

was at time r in state i and the policy $\mathbf{d}$ is used in the interval $[r,t]$. Then the matrix $\mathbf{V}^{\mathbf{d}}_{r,t}$ can be used to obtain the distribution at time t under policy $\mathbf{d}$ as $\mathbf{p}^{\mathbf{d}}_t = \pi \mathbf{V}^{\mathbf{d}}_{0,t}$ as given in Eq. 2.19, where π is the initial distribution vector of the CTMDP [42, 49].

Note that the expected reward under policy $\mathbf{d}$ with initial distribution π in the time interval $[0,T]$ is given by $\pi \mathbf{g}^{\mathbf{d}}_{0,T}$ (cf. Eq. 2.22). As given in [42, 49], $\mathbf{g}^{\mathbf{d}}_{0,T}$ is the gain vector containing at position i the expected reward accumulated in the interval $[0,T]$ if the CTMDP is in state i at time 0 and the policy $\mathbf{d}$ is used. It is computed as

$$\mathbf{g}^{\mathbf{d}}_{0,T} = \mathbf{V}^{\mathbf{d}}_{0,T}\mathbf{g}_T + \int_0^T \mathbf{V}^{\mathbf{d}}_{\tau,T}\mathbf{r}^{\mathbf{d}}\,\mathrm{d}\tau$$

as given in Eq. 2.21. The vector $\mathbf{g}_T$ is the policy-independent initial gain vector at time T. The second term describes the accumulated gain until time T.

Let a CTMDP for a PHG have rewards $\mathbf{r} = \mathbf{0}$. As discussed in [42] for a CTMDP with zero rewards and non-trivial initial gain vector $\mathbf{g}_T$ the computation of the gain vector reduces to

$$\mathbf{g}^{\mathbf{d}}_{0,T} = \mathbf{V}^{\mathbf{d}}_{0,T}\mathbf{g}_T, \tag{5.49}$$

since the second term becomes zero. Let $\mathbf{g}_T$ be the terminal gain vector with $\mathbf{g}_T(i) = 1$, if $i \in \mathscr{S}_A$, and $\mathbf{g}_T(i) = 0$ for all $i \in \mathscr{S}_T$. Recall that $\mathscr{S}_A = \{(0,0)\}$, i.e. the single absorbing state $(0,0)$ corresponds to the destination node. Then the optimal gain vector $\mathbf{g}^*_{0,T}$ can be used to compute the *maximal reachability probability*. According to [42, Lemma 1] the measurable policy $\mathbf{d}^*$ maximizing the gain vector in the interval $[0,T]$ in all elements is optimal when

$$\mathbf{d}^* = \arg\max_{\mathbf{d}\in\Pi} \left(\mathbf{V}^{\mathbf{d}}_{0,T}\mathbf{g}_T \text{ in all elements}\right). \tag{5.50}$$

Then $\mathbf{g}^*_{0,T} = \mathbf{V}^{\mathbf{d}^*}_{0,T}\mathbf{g}_T$, which is maximal in all elements. Observe that the entry $\mathbf{g}^*_{0,T}(i)$ is the maximal transient probability for reaching the absorbing state $(0,0)$ within time T starting from state $i \in \mathscr{S}_T$. Thus for the CTMDP with described settings the maximal transient probability is equivalent to the *maximal time bounded reachability*.

Corollary 1. *Assume that the CTMDP for a PHG is defined using* $\mathbf{r} = \mathbf{0}$, $\mathbf{g}_T(i) = 1$ *for* $i \in \mathscr{S}_A$ *and* 0 *otherwise. Let* $\mathbf{d}^*$ *be the optimal policy maximizing the Eq. 5.49 and* $\mathbf{g}^*_{0,T}$ *the corresponding gain vector. Then the maximal probability of reaching a destination state* $(0,0)$ *from the edge* $i \in E_{ini}$ *in the finite interval* $[0,T]$ *is given by [40]*

$$G_i = \mathbf{a}_i\,\mathbf{g}^*_{0,T}, \tag{5.51}$$

where $\mathbf{a}_i = (\mathbf{0}_{<i}, \pi_i, \mathbf{0}_{>i})$, *with zeros vector* $\mathbf{0}_{<i}$ *of length* $\sum_{j\in E,\, j<i} n_j$, *and zeros vector* $\mathbf{0}_{>i}$ *of length* $\sum_{j\in E,\, j>i} n_j$.

In the following we describe the basic algorithm developed by Miller [127] for the computation of the optimal policy $\mathbf{d}^*$ maximizing the expected reward over the finite planning horizon $[0,T]$ and the optimal gain vector $\mathbf{g}^*_{0,T}$.

Miller's Algorithm The algorithm developed in [127] computes $\mathbf{d}^*$ maximizing the gain vector per state for a finite interval $[0,T]$ with the initial terminal reward $\mathbf{g}_T = \mathbf{0}$. Let $\mathbf{r}^{\mathbf{u}_t}$ be the reward vector for the CTMDP as defined in Sec. 2.2 (see Page 17), and $\mathbf{u}_i$ be the decision vector in $[t_{i-1}, t_i)$.

Theorem 5.1 (Theorem 1 and Theorem 6 of [127]). *A policy* $\mathbf{d} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_T)$ *is optimal if it maximizes for almost all* $t \in [0,T]$

$$\mathbf{Q}^{\mathbf{u}}\,\mathbf{g}_{t,T} \;+\; \mathbf{r}^{\mathbf{u}}, \tag{5.52}$$

where

$$-\frac{d}{dt}\mathbf{g}_{t,T} = \mathbf{Q}^{\mathbf{u}_t}\,\mathbf{g}_{t,T} \;+\; \mathbf{r}^{\mathbf{u}_t}, \quad \textit{and } \mathbf{g}_T \geq \mathbf{0}.$$

The maximization of the Eq. 5.52 is equivalent to

$$-\frac{d}{dt}\mathbf{g}_{t,T} = \max_{\mathbf{u}\in\mathscr{P}}\left(\mathbf{Q}^{\mathbf{u}}\,\mathbf{g}_{t,T} \;+\; \mathbf{r}^{\mathbf{u}}\right),\ \mathbf{g}_T \geq \mathbf{0}.$$

There is a piecewise constant policy $\mathbf{d} \in \Pi$ *which results in* $\mathbf{g}^*_{0,T}$ *and maximizes the Eq. 5.52.*

Miller presents a method for choosing an optimal piecewise constant policy. The following sets are defined for some policy $\mathbf{d} \in \Pi$ and the corresponding gain vector $\mathbf{g}^{\mathbf{d}}_{t,T}$ at time t

$$\begin{aligned}
\mathscr{F}_1(\mathbf{g}^{\mathbf{d}}_{t,T}) &= \left\{\mathbf{u} \in \mathscr{P} \mid \mathbf{u} \text{ maximizes } \mathbf{v}^{(1)}(\mathbf{u})\right\},\\
\mathscr{F}_2(\mathbf{g}^{\mathbf{d}}_{t,T}) &= \left\{\mathbf{u} \in \mathscr{F}_1(\mathbf{g}_{t,T}) \mid \mathbf{u} \text{ maximizes } -\mathbf{v}^{(2)}(\mathbf{u})\right\},\\
&\vdots\\
\mathscr{F}_{n+1}(\mathbf{g}^{\mathbf{d}}_{t,T}) &= \left\{\mathbf{u} \in \mathscr{F}_n(\mathbf{g}_{t,T}) \mid \mathbf{u} \text{ maximizes } (-1)^n\mathbf{v}^{(n+1)}(\mathbf{u})\right\},
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{v}^{(1)}(\mathbf{u}) &= \mathbf{Q}^{\mathbf{u}}\,\mathbf{g}^{\mathbf{d}}_{t,T} \;+\; \mathbf{r}^{\mathbf{u}},\\
\mathbf{v}^{(j)}(\mathbf{u}) &= \mathbf{Q}^{\mathbf{u}}\,\mathbf{v}^{(j-1)} \text{ for } 2 \leq j \leq n+1 \text{ with}\\
\mathbf{v}^{(j-1)} &= \mathbf{v}^{(j-1)}(\mathbf{u}) \text{ for any } \mathbf{u} \in \mathscr{F}_{j-1}.
\end{aligned}$$

Here the vector $\mathbf{v}^{(i)}(\mathbf{u})$ contains the values of the i-th derivative of $\mathbf{g}_{t,T}$ multiplied with $(-1)^{(i-1)}$. For the given $\mathbf{g}^{\mathbf{d}}_{t,T}$ and the derivatives at t, the vector $\mathbf{g}^{\mathbf{d}}_{t',T}$ for $t' < t$ can be computed as [49]

$$\mathbf{g}^{\mathbf{d}}_{t',T} = \mathbf{g}^{\mathbf{d}}_{t,T} + \sum_{i=1}^{\infty}\frac{(t'-t)^i}{i!}\mathbf{v}^{(i)}(\mathbf{u}).$$

Assume that the vectors $\mathbf{u}$ are lexicographically ordered

$$\mathbf{u}(1),\ \mathbf{u}(2), \ldots,\ \mathbf{u}(n).$$

For each point in time $t \in [0,T]$ we consider the set of actions which maximizes the first derivative of the vector function $\mathbf{g}_{t,T}$ and thus is optimal. It is sufficient to consider only up to the first $n+1$ derivatives [127]. Then the selection rule obtains the optimal policy $\mathbf{d}^*(t)$ at time t as the lexicographically smallest vector $\mathbf{u} \in \mathscr{F}_{n+1}(\mathbf{g}^{\mathbf{d}}_{t,T})$. Furthermore, the selected $\mathbf{d}^*(t)$ is optimal in $(t-\varepsilon, t)$ for some $\varepsilon > 0$ which is summarized in the following Theorem.

Theorem 5.2 (Lemma 4 of [127], Theorem 2 [42, 49])**.** *Consider the policy* $\mathbf{d} \in \Pi$ *defined on the interval* $(t', T]$*. Let* $\mathbf{u} \in \mathscr{F}_{n+1}(\mathbf{g}^{\mathbf{d}}_{t,T})$ *for* $t' < t < T$*. Then exists some* ε *where* $0 < \varepsilon < t - t'$ *such that* $\mathbf{u} \in \mathscr{F}_{n+1}(\mathbf{g}^{\mathbf{d}}_{t'',T})$ *for all* $t'' \in [t-\varepsilon, t]$*.*

The algorithm developed by Miller [127] for computation of the optimal piecewise constant policy on $[0, T]$ maximizing the Eq. 5.52 is given in Algorithm 5.2.

Algorithm 5.2: Computing the optimal policy $\mathbf{d}^*$ and the gain vector $\mathbf{g}^*_{0,T}$

1: Initialize $\mathbf{g}_T$, $t' = T$;
2: Determine $\mathbf{u}_{t'}$ from $\mathscr{F}_{n+1}(\mathbf{g}_{t',T})$ and $\mathbf{g}_{t',T}$ using the selection procedure described;
3: Obtain $\mathbf{g}_{t,T}$, $0 \le t \le t'$, by solving

$$-\frac{d}{dt}\mathbf{g}_{t,T} = \mathbf{Q}^{\mathbf{u}_{t'}}\,\mathbf{g}_{t,T} \;+\; \mathbf{r}^{\mathbf{u}_{t'}}$$

with the previous value $\mathbf{g}_{t',T}$ as terminal condition;
4: Set $t'' = \inf\{t :\ \mathbf{u}_t$ satisfies the selection procedure in $[t, t')$ based on $\mathbf{g}_{t,T}\}$;
5: If $t'' \le 0$ terminate and return the gain vector $\mathbf{g}^*_{0,T}$ at $t = 0$ and the corresponding optimal piecewise constant policy $\mathbf{d}^*$. Else go to 2 with $t' = t''$.

The algorithm works in a backward manner. It starts with the initial gain vector $\mathbf{g}_T$ at time $t' = T$, computes the optimal policy $\mathbf{u}$ at t' and assumes that the policy is constant in the small time interval $(t, t']$. Then the gain vector $\mathbf{g}^{\mathbf{u}}_{t',T}$ can be computed for the whole interval and the optimal decision vector at t can be determined. The computations are repeated until the whole interval $[0, T]$ is covered.

In [127] it is shown that a piecewise constant policy has a finite number of switch points in $[0, T]$. Thus steps 2-5 of the Algorithm 5.2 are executed a finite number of times. However, step 4 is not implementable, which is used for obtaining the length of time the decision vector is valid. Thus we present approximation algorithms to obtain $\mathbf{d}^*$ and the corresponding $\mathbf{g}^*_{0,T}$ in the following.

Discretization Approach The discretization algorithm uses a fixed interval for the length of time the optimal policy $\mathbf{d}(t')$ obtained is valid instead of computing step 4 of the Algorithm 5.2. For a fixed length of the discretization step the corresponding DTMDP is obtained which is then used to compute the optimal policy and the optimal gain vector. It is shown in the literature [125, 127] that for the step length going to zero the policy converges towards the optimal policy. The algorithm computes ε-optimal policies, where a policy $\mathbf{d}$ is called to be ε-optimal if $\|\mathbf{g}^*_{t,T} - \mathbf{g}^{\mathbf{d}}_{t,T}\|_\infty \le \varepsilon$ for all $t \in [0, T]$. In the following we describe the algorithm from [127, 49, 40].

Let $h > 0$ be a fixed interval length. For h sufficiently small it holds that $e^{h\,\mathbf{Q}^{\mathbf{u}}} = \mathbf{P}^{\mathbf{u}}_h + o(h^2)$ such that the stochastic matrix $\mathbf{P}^{\mathbf{u}}_h$ is defined as

$$\mathbf{P}^{\mathbf{u}}_h = \mathbf{I} + h\,\mathbf{Q}^{\mathbf{u}}, \tag{5.53}$$

which is the transition matrix of the DTMC induced by the decision vector $\mathbf{u}$. Given h and for every $\mathbf{u} \in \mathscr{P}$ the matrices $\mathbf{P}^{\mathbf{u}}_h$ define a DTMDP.

Knowing the gain vector $\mathbf{g}_{t,T}$ at time t and the decision vector $\mathbf{u}$ holding in a fixed interval $(t-h, t]$ the approximation for the gain vector $\mathbf{g}_{t-h,T}$ at time $(t-h)$ can be

obtained as

$$\mathbf{g}_{t-h,T} = \mathbf{P}_h^{\mathbf{u}}\,\mathbf{g}_{t,T} \;+\; h\,\mathbf{r}^{\mathbf{u}} + o(h). \tag{5.54}$$

For sufficiently large N the discretization step can be set to $h = T/N$. Then dynamic programming algorithms for DTMDPs can be applied to obtain the optimal policy and its value for the problem with time horizon N.

Let nz be the average number of non-zero elements in $\mathbf{Q}^{\mathbf{u}}$. Then the overall effort of the described algorithm results in $\mathscr{O}(h^{-1}\,nz(\sum_{i=1}^{n} m_i))$ [42], where m_i is the cardinality of $\mathscr{D}(i)$ for $i \in \mathscr{S}$.

Since the value of ε cannot be determined in the discretization approach, the discretization step length h for a predefined ε cannot be chosen. Furthermore, the value of h is rather constant and not adaptable. In the following the uniformization based algorithm for computing an ε-optimal policy for a predefined ε is presented.

Uniformization Approach The numerical algorithm developed in [42] computes the ε-optimal policy in a backward manner starting from T with some predefined $\mathbf{g}_T$. In the following we describe the efficient algorithm proposed in [49]. For a given gain vector $\mathbf{g}_{t,T}$ and constant decision vector $\mathbf{u}$ in the interval $(t-\delta, t]$ the vector $\mathbf{g}_{t-\delta,T}$ can be computed as

$$\begin{aligned}\mathbf{g}_{t-\delta,T} &= e^{\delta\,\mathbf{Q}^{\mathbf{u}}}\mathbf{g}_{t,T} + \int_{\tau=0}^{\delta} e^{\tau\mathbf{Q}^{\mathbf{u}}}\mathbf{r}\,d\tau \\ &= \left(\sum_{k=0}^{\infty}\frac{(\delta\,\mathbf{Q}^{\mathbf{u}})^k}{k!}\right)\mathbf{g}_{t,T} + \int_{\tau=0}^{\delta}\left(\sum_{k=0}^{\infty}\frac{(\tau\mathbf{Q}^{\mathbf{u}})^k}{k!}\right)\mathbf{r}\,d\tau,\end{aligned} \tag{5.55}$$

where the first term describes the accumulated reward in the interval $(t, T]$ and the second term describes the accumulated reward in the interval $(t-\delta, t]$. The Eq. 5.55 can be solved using uniformization method as follows. Let $\alpha = \max_{\mathbf{u}\in\mathscr{P}}\left(\max_{i\in\mathscr{S}_T(\mathbf{Q}^{\mathbf{u}}(i,i))}\right)$, then the stochatic matrix can be defined as in Eq. 2.11

$$\mathbf{P}^{\mathbf{u}} = \mathbf{I} + \frac{1}{\alpha}\mathbf{Q}^{\mathbf{u}},$$

and $\beta(k, \alpha t)$ is the probability that a Poisson process with parameter αt performs k jumps (cf. Eq. 2.12). Let

$$\zeta(\alpha, \delta, K) = (1 - \sum_{l=0}^{K}\beta(l, \alpha\delta)).$$

Then the Eq. 5.55 can be computed as

$$\mathbf{g}_{t-\delta,T} = \sum_{k=0}^{\infty}(\mathbf{P}^{\mathbf{u}})^k\left(\beta(k,\alpha\delta)\mathbf{g}_{t,T} + \frac{\zeta(\alpha,\delta,K)}{\alpha}\mathbf{r}\right). \tag{5.56}$$

Assume now that the following bounds $\underline{\mathbf{g}}_{t,T} \le \mathbf{g}^*_{t,T} \le \overline{\mathbf{g}}_{t,T}$ are known. E.g. for $t = T$ the value of $\mathbf{g}_T$ is initialized with $\mathbf{0}$ in Miller's algorithm, then the bounds can be easy obtained. The following vectors are defined

$$\begin{aligned}&\underline{\mathbf{v}}^{(k)} = \mathbf{P}^{\mathbf{u}_t}\underline{\mathbf{v}}^{(k-1)},\ \underline{\mathbf{w}}^{(k)} = \mathbf{P}^{\mathbf{u}_t}\underline{\mathbf{w}}^{(k-1)} \\ &\overline{\mathbf{v}}^{(k)} = \max_{\mathbf{u}\in\mathscr{P}}\left(\mathbf{P}^{\mathbf{u}_t}\overline{\mathbf{v}}^{(k-1)}\right),\ \overline{\mathbf{w}}^{(k)} = \max_{\mathbf{u}\in\mathscr{P}}\left(\mathbf{P}^{\mathbf{u}_t}\overline{\mathbf{w}}^{(k-1)}\right) \\ &\text{with } \underline{\mathbf{v}}^{(0)} = \underline{\mathbf{g}}_{t,T},\ \overline{\mathbf{v}}^{(0)} = \overline{\mathbf{g}}_{t,T},\ \underline{\mathbf{w}}^{(0)} = \overline{\mathbf{w}}^{(0)} = \mathbf{r}.\end{aligned} \tag{5.57}$$

Here the decision vector $\mathbf{u}_t$ is the lexicographically smallest vector from $\mathscr{F}_{n+1}(\underline{\mathbf{g}}_{t,T})$. Vectors $\underline{\mathbf{v}}^{(k)}$, $\underline{\mathbf{w}}^{(k)}$ correspond to the policy using decision vector $\mathbf{u}_t$ in time interval $(t-\delta,t]$. The vector sequences $\overline{\mathbf{v}}^{(k)}$, $\overline{\mathbf{w}}^{(k)}$ can be obtained using methods for computation of optimal policy in a finite horizon DTMDP [27]. Given these vectors, decision vector $\mathbf{u}$, and bounds for $\mathbf{g}_{t,T}$ the error bounds for $\mathbf{g}_{t-\delta,T}$ can be computed as [49, Theorem 3]

$$\begin{aligned}\underline{\mathbf{g}}^K_{t-\delta,T} &= \sum_{k=0}^{K}\left(\beta(k,\alpha t)\underline{\mathbf{v}}^{(k)}+\zeta(\alpha,\delta,k)\underline{\mathbf{w}}^{(k)}\right)+\left(\underline{\gamma}^K_{t,\delta}+\underline{\nu}^K_{\delta}\right)\mathbb{1}\leq \mathbf{g}^*_{t-\delta}\\ &\leq \sum_{k=0}^{K}\left(\beta(k,\alpha t)\overline{\mathbf{v}}^{(k)}+\zeta(\alpha,\delta,k)\overline{\mathbf{w}}^{(k)}\right)+\left(\overline{\gamma}^K_{t,\delta}+\overline{\nu}^K_{\delta}\right)\mathbb{1}\leq \overline{\mathbf{g}}^K_{t-\delta,T}\end{aligned} \tag{5.58}$$

with

$$\begin{aligned}&\underline{\gamma}^K_{t,\delta}=\varepsilon_1(K,\alpha\delta)\max_{i\in\mathscr{S}}\left(\underline{\mathbf{v}}^{(K)(i)}\right),\ \overline{\gamma}^K_{t,\delta}=\varepsilon_1(K,\alpha\delta)\max_{i\in\mathscr{S}}\left(\overline{\mathbf{v}}^{(K)(i)}\right)\\ &\underline{\nu}^K_{\delta}=\varepsilon_1(K,\alpha\delta)\max_{i\in\mathscr{S}}\left(\underline{\mathbf{w}}^{(K)(i)}\right),\ \overline{\nu}^K_{\delta}=\varepsilon_1(K,\alpha\delta)\max_{i\in\mathscr{S}}\left(\overline{\mathbf{w}}^{(K)(i)}\right)\\ &\varepsilon_1(K,\alpha\delta)=\zeta(\alpha,\delta,K),\ \varepsilon_2(K,\alpha\delta)=\left(\delta\zeta(\alpha,\delta,K)-\tfrac{K+1}{\alpha}\zeta(\alpha,\delta,K+1)\right).\end{aligned}$$

The sums of Poisson probabilities can be truncated resulting in the following bounds for every $\delta>0$

$$\begin{aligned}\sum_{k=0}^{K}\beta(k,\alpha\delta)\overline{\mathbf{v}}^{(k)}+\overline{\gamma}^K_{t,\delta}&\geq\sum_{k=0}^{\infty}\beta(k,\alpha\delta)\overline{\mathbf{v}}^{(k)},\\ \sum_{k=0}^{K}\zeta(k,\alpha\delta)\overline{\mathbf{w}}^{(k)}+\overline{\nu}^K_{\delta}&\geq\sum_{k=0}^{\infty}\zeta(k,\alpha\delta)\overline{\mathbf{w}}^{(k)},\end{aligned}$$

which can be computed using the precomputed vectors $\underline{\mathbf{v}}^{(1)},\ldots,\underline{\mathbf{v}}^{(K)}$, $\overline{\mathbf{v}}^{(1)},\ldots,\overline{\mathbf{v}}^{(K)}$, $\underline{\mathbf{w}}^{(1)},\ldots,\underline{\mathbf{w}}^{(K)}$, $\overline{\mathbf{w}}^{(1)},\ldots,\overline{\mathbf{w}}^{(K)}$. Using $\underline{\mathbf{g}}_{t,T}$ and $\overline{\mathbf{g}}_{t,T}$ for the computation of $\underline{\mathbf{g}}^K_{t-\delta,T}$, $\overline{\mathbf{g}}^K_{t-\delta,T}$ it holds that $\|\overline{\mathbf{g}}_{t,T}-\underline{\mathbf{g}}_{t,T}\|\leq\|\overline{\mathbf{g}}^K_{t-\delta,T}-\underline{\mathbf{g}}^K_{t-\delta,T}\|$.

To ensure that $\|\overline{\mathbf{g}}_{0,T}-\underline{\mathbf{g}}_{0,T}\|_\infty\leq\varepsilon$ for a given ε the error at time t should be at range of $\varepsilon(T-t)/T$ which can be reached by an appropriate choice of δ [49]. Let ε_{trunc} be the error due to the truncation point of the Poisson probabilities and ε_{succ} be the error due to the computation of the vector at time t which determines the reward gained in $(t,T]$ such that

$$\varepsilon_{trunc}(t,\delta,K)=\overline{\gamma}^K_{t,\delta}-\underline{\gamma}^K_{t,\delta}+\overline{\nu}^K_{\delta}-\underline{\nu}^K_{\delta}\in\mathscr{O}(\delta^{K+1}). \tag{5.59}$$

Now let $\mathbf{u}^*_l$ be the decision vector chosen by the selection procedure at time of the l-th transition in the interval $[t-\delta,t]$. Then the difference between vectors $\overline{\mathbf{v}}^{(k)}$ and $(\prod_{l=1}^{k}\mathbf{P}^{\mathbf{u}^*_l})\,\underline{\mathbf{g}}_t$ determines the following error

$$\varepsilon_{succ}(t,\delta,K)=\left\|\sum_{k=0}^{K}\left(\overline{\mathbf{v}}^{(k)}-\left(\prod_{l=1}^{k}\mathbf{P}^{\mathbf{u}^*_l}\right)\right)e^{-\alpha\delta}\frac{(\alpha\delta)^k}{k!}\right\|_\infty. \tag{5.60}$$

Consider that $\overline{\mathbf{g}}_T=\underline{\mathbf{g}}_T=\mathbf{g}_T=\mathbf{0}$ in the initial step such that the local error $\varepsilon_T=0$. For a step length δ the local error is in $(O)(\delta^2)$ and theoretically the global error goes to 0 as $\delta\to 0$. Then it holds that [42]

$$\varepsilon(t,\delta,K)=\varepsilon_{trunc}(t,\delta,K)+\varepsilon_{succ}(t,\delta,K)\leq\varepsilon(t,\delta,K+1)=\varepsilon_{trunc}(t,\delta,K+1)+\varepsilon_{succ}(t,\delta,K+1)$$

With these considerations we can define the algorithm 5.3 for computing $\underline{\mathbf{g}}_{0,T}$, $\overline{\mathbf{g}}_{0,T}$ and the ε-optimal policy $\mathbf{d}$ with the corresponding gain vector of at least $\underline{\mathbf{g}}_{0,T}$ such that $\underline{\mathbf{g}}_{0,T} \leq \mathbf{g}^*_{0,T} \leq \overline{\mathbf{g}}_{0,T}$ and $\|\overline{\mathbf{g}}_{0,T} - \underline{\mathbf{g}}_{0,T}\|_\infty \leq \varepsilon$ for the given precision ε [49, 42]. The algorithm requires $\underline{\mathbf{g}}_T$, $\overline{\mathbf{g}}_T$, w, K_{max}, ε as an input.

Algorithm 5.3: Computing the optimal policy $\mathbf{d}^*$ and the bounding vectors for $\mathbf{g}^*_{0,T}$

1: Initialize $i = 0$, $t = T$;
2: set $stop = false$, $K = 1$, $\underline{\mathbf{v}}^{(0)} = \underline{\mathbf{g}}_T$, $\overline{\mathbf{v}}^{(0)} = \overline{\mathbf{g}}_T$, $\underline{\mathbf{w}}^{(0)} = \overline{\mathbf{w}}^{(0)} = \mathbf{r}$
3: select $\mathbf{u}_t$ from $\mathscr{F}_{n+1}(\mathbf{g}_{t,T})$, if $i = 0$ set $\mathbf{c}_0 = \mathbf{u}_t$
4: repeat
5: compute $\underline{\mathbf{v}}^{(K)}$, $\overline{\mathbf{v}}^{(K)}$, $\underline{\mathbf{w}}^{(K)}$, $\overline{\mathbf{w}}^{(K)}$ using Eq. 5.57
6: find $\delta = \max\left(\arg\max_{\delta' \in [0,t]} \left(\varepsilon_{trunc}(t,\delta',K) \leq \frac{w\delta'}{T-t_0}\varepsilon\right), \min\left(\delta_{min}, t-t_0\right)\right)$
7: compute $\varepsilon_{trunc}(t,\delta',K)$ and $\varepsilon_{succ}(t,\delta,K)$ using Eq. 5.59, Eq. 5.60
8: if $\varepsilon_{trunc}(t,\delta',K) + \varepsilon_{succ}(t,\delta,K) > \frac{T-t+\delta}{T-t_0}\varepsilon$ then
9: reduce δ until
10: $\varepsilon_{trunc}(t,\delta',K) + \varepsilon_{succ}(t,\delta,K) \leq \frac{T-t+\delta}{T-t_0}\varepsilon$
11: or $\delta = \min(\delta_{min}, t-t_0)$ and set $stop = true$
12: $K = K+1$
13: until $stop$ or $K = K_{max}+1$
14: compute $\underline{\mathbf{g}}_{t-\delta,T}$ from $\underline{\mathbf{v}}^{(k)}$, $\overline{\mathbf{w}}^{(k)}$, and $\overline{\mathbf{g}}_{t-\delta,T}$ from $\overline{\mathbf{v}}^{(k)}$, $\overline{\mathbf{w}}^{(k)}$, $k = 0,\ldots,K$ using Eq. 5.58
15: if $\mathbf{u}_t = \mathbf{c}_i$ then $\mathbf{c}_{i+1} = \mathbf{u}_t$, $t_t = t - \delta$ and $i = i+1$
16: if $t - t_0 = \delta$ then terminate else go to 2 with $t = t - \delta$

The uniformization algorithm is much more efficient than the discretization technique. The overall effort per time step is $\mathscr{O}(K nz(\sum_{i=1}^{n} m_i))$. There are practical instances where the difference between the number of required iterations is a factor of 1000 [42].

Computing the Probability of Arriving on Time Using approximation techniques for computing the optimal policy $\mathbf{d}$ and the optimal gain vector $\mathbf{g}_{0,T}$ we can formulate the algorithm for finding a path in a PHG that maximizes the probability of reaching a destination v_{fin} in the interval $[0,T]$ as proposed in [40].

The backward induction algorithm 5.4 is based on the discretization approach described earlier. The algorithm operates on the SSPP CTMDP with $\mathbf{r} = \mathbf{0}$, $\mathbf{g}_T(i,x) = 1$ for $(i,x) \in \mathscr{S}_A$ and $\mathbf{g}_T(i,x) = 0$ otherwise. Given the discretization step length h the matrices $\mathbf{P}^{\mathbf{u}}_h$ can be computed using Eq. 5.53 for all $\mathbf{u} \in \mathscr{P}$ and are used as an input.

For sufficiently large N the discretization step h is obtained as given in step (1). The obtained stochastic matrices $\mathbf{P}^{\mathbf{u}}_h$ for all $\mathbf{u} \in \mathscr{P}$ determine the DTMDP. Then the modified policy iteration algorithm can be applied for solving finite-horizon discrete-time MDP (cf. Sec. 5.3.2).

Since the discretization step h is fixed we obtain DTMDP on $[0,N]$, $N < \infty$. Then an optimal policy $\mathbf{u}_i$ in the interval $(t_{i-1}, t_i]$ is assumed to be constant and is computed at decision epoch $i \in \{0,\ldots,N\}$ of the DTMDP. Given the initial gain vector $\mathbf{g}_T$ and the optimal decision vector $\mathbf{u}$ holding in the interval $(t-h, t]$ the gain vector is then

Algorithm 5.4: Computing the ε-optimal policy $\mathbf{d}$ and the gain vector $\mathbf{g}^{\mathbf{d}}_{0,T}$

1: Initialize $h = \frac{T}{N}$, $t = N$, $\mathbf{g}^{\mathbf{u}_t} = \mathbf{g}_T$
2: for $t = t-1$ downto 1 do
3: determine new policy $\mathbf{u}_t$ satisfying $T_{\mathbf{u}_t}\,\mathbf{g}^{\mathbf{u}_{t+1}} = T\,\mathbf{g}^{\mathbf{u}_{t+1}}$ using Eq. 5.28 as

$$\mathbf{u}_t(i,x) = \arg\max_{u\in\mathscr{D}(i)} \left(\sum_{(j,y)\in\mathscr{S}} \mathbf{P}^u_h((i,x),(j,y))\,\mathbf{g}^{\mathbf{u}_{t+1}}(j,y) \right)$$

4: compute the gain vector using Eq. 5.54 as

$$\mathbf{g}^{\mathbf{u}_t} = \mathbf{P}^{\mathbf{u}_t}_h\,\mathbf{g}^{\mathbf{u}_{t+1}}$$

5: terminate with $\mathbf{d} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_N)$ and $\mathbf{g}^{\mathbf{d}}_{0,T} = \mathbf{g}^{\mathbf{u}_0}$

computed in step (4). The procedure is repeated until gain vector $\mathbf{g}^{\mathbf{u}_i}$ for each decision epoch $i \in \{0,\ldots,N\}$ is computed. Note that the value of ε and the value of h for some predefined precision ε cannot be determined in Algorithm 5.4.

Knowing the ε-optimal policy $\mathbf{d}$ maximizing the expected reward over the finite interval $[0,T]$ and the gain vector $\mathbf{g}^{\mathbf{d}}_{0,T}$ the maximal probability to reach the destination $(0,0)$ from the edge i can be computed for each $i \in E_{ini}$ using (5.51). Then the path from v_{ini} maximizing the probability to arrive at v_{fin} with a weight $\leq T$ can be obtained from $\mathbf{d}$. Its initial edge is selected according to

$$i^* = \arg\max_{i\in E_{ini}} (G_i)\,. \tag{5.61}$$

The maximal probability to reach v_{fin} in the interval $[0,T]$ can be obtained as

$$G^* = a_{i^*}\,\mathbf{g}^{\mathbf{d}}_{0,T}. \tag{5.62}$$

Using the discretization approach implemented in Algorithm 5.4 the decisions in the real system can be made based on history vectors $\psi_{(i_1,w_1,\ldots,i_l,w_l)}$ defined in Eq. 5.14. Then the weights w_k have to be multiples of the discretization step length h. Assume that the realized weight of the edge k is given by $w_k = t_k h$ for $t_k \in \mathbb{N}$ and $t = \sum_{k=1}^{l} t_k$. Considering the whole history of realizations $(w_1,\ldots,w_l)$ the optimal decision at time $t\cdot h$ can be approximated using vector $\mathbf{g}^{\mathbf{u}_t}$. Note that the value $\mathbf{g}^{\mathbf{u}_t}(i,x)$ is the maximal probability of reaching the absorbing state $(0,0)$ in the interval $[t,N]$ starting from state (i,x).

Assume that the vehicle arrives at node $fin(i_l)$. The decision on the next edge to traverse is based on

$$i^* = \arg\max_{j\in\mathscr{D}(i_l)} \left(\sum_{x=1}^{n_{i_l}} \bar{\psi}_{(i_1,w_1,\ldots,i_l,w_l)}(i_l,x)\cdot \left(\sum_{y=1}^{n_{i_l}} \mathbf{P}^j((i_l,x),(i_l,y))\mathbf{g}^{\mathbf{u}_t}(i_l,y) + \sum_{y=1}^{n_j} \mathbf{P}^j((i_l,x),(j,y))\mathbf{g}^{\mathbf{u}_t}(j,y) \right) \right). \tag{5.63}$$

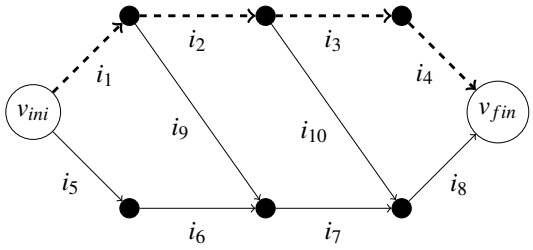


Figure 5.13.: A small acyclic graph. Edges with correlated weights are highlighted in dashed style.

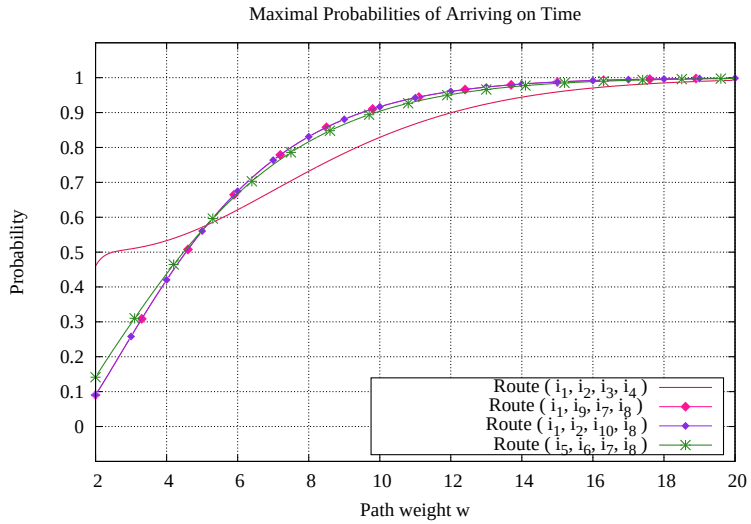


Figure 5.14.: The probabilities of arriving at the destination node v_{fin} with a path weight of less or equal to w.

***Example* 5.8.** *We consider the simple graph from example 5.7 visualized in Figure 5.13.*

The probability to reach the destination v_{fin} in a given time depends on the path. Note that the weights of all edges are described by order 2 hyperexponential PHDs with mean 1 and a squared coefficient of variation 2.5.

The upper path (i_1, i_2, i_3, i_4) contains edges with positively correlated edge weights where the correlation coefficient is $\rho = 0.3$. The lower path (i_5, i_6, i_7, i_8) contains edges with uncorrelated weights. The weights of the edges i_1, i_9, and i_2, i_{10} are negatively correlated.

We computed the maximal probability to reach the node v_{fin} with at most T time units, for $T = 2, \ldots, 20$, using the discretization based algorithm 5.4. We use $h = T/N$ for relatively large $N = 3000$ and solve the resulting DTMDP for a finite time horizon of N steps. The computed approximation results are summarized graphically in Fig. 5.14.

It can be seen that the more reliable route, e.g. (i_5, i_6, i_7, i_8) *results in a lower probability of meeting a deadline for small time horizons. Hence, if one starts just before a deadline, the most correlated route offers a higher probability of avoiding lateness. In this case, the path containing edges with correlated weights should be preferred to the path containing edges with uncorrelated or negatively correlated weights.*

On the other hand, the more reliable route (i_5, i_6, i_7, i_8) *results in a higher probability of meeting a deadline for relatively long time horizons, e.g.* $T > 5$.

The optimal policy behaves as follows: For a long time horizon, the best decision is the more reliable route (i_5, i_6, i_7, i_8). *If the time horizon becomes smaller, then the best decision is the route with positively correlated weights* (i_1, i_2, i_3, i_4).

Empirical Comparison of Uniformization and Discretization Methods *We compute probabilities of arriving on time for subroutes* (i_2, i_3, i_4), (i_2, i_{10}, i_8) *in different time intervals using the uniformization 5.3 and the discretization approach 5.4. The corresponding PHG is visualized in Figure 5.15.*

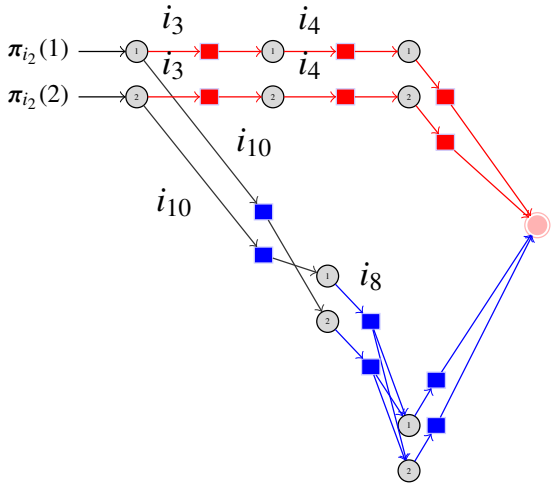


Figure 5.15.: The PHG corresponding to subroutes (i_2, i_3, i_4), (i_2, i_{10}, i_8) in Fig. 5.13.

Results obtained for interval length $T = 1$ *Table 5.3 includes bounds for the probability of arriving on time. We summarized results gained by policies computed with uniformization method. For that we present approximation results computed with discretization approach.*

We assume that the CTMDP has initial probability distribution $(0.5, 0.5)$. *The edge weights along the route* (i_2, i_3, i_4) *are correlated, i.e.,* $\rho_{i_2 i_3} = 0.3$ *and* $\rho_{i_3 i_4} = 0.3$. *On the contrary, route* (i_2, i_{10}, i_8) *contains uncorrelated edge weights, i.e.,* $\rho_{i_2 i_{10}} = 0$ *and* $\rho_{i_{10} i_8} = 0$. *Observe that the transient probability is higher for the correlated policy* (i_2, i_3, i_4) *than for the uncorrelated policy* (i_2, i_{10}, i_8).

The number of iterations in the uniformization algorithm is set adaptively according to the value ε. *To compare the results the discretization step* h *is set in such a way that the discretization algorithm computes the same number of iterations as in the*

Table 5.3.: Probability bounds and approximate probabilities for the routes (i_2, i_3, i_4) and (i_2, i_{10}, i_8) in the small graph example presented in Fig. 5.15. The interval lenght is 1. The initial probability is $(0.5, 0.5)$.

Subpath (i_2, i_3, i_4) with $\rho = 0.3$				
Uniformization				Discretization
ε	# Iter.	Bounds for G^*		Approximation for G^*
$1.0e+0$	10	$4.4061\,85e-01$	$5.3913\,93e-01$	$4.8051\,64e-01$
$1.0e-1$	18	$4.8829\,01e-01$	$4.9589\,44e-01$	$4.8636\,47e-01$
$1.0e-3$	32	$4.9232\,07e-01$	$4.9242\,07e-01$	$4.8915\,34e-01$
$1.0e-5$	56	$4.9237\,52e-01$	$4.9237\,61e-01$	$4.9058\,14e-01$
$1.0e-10$	180	$4.9237\,57e-01$	$4.9237\,57e-01$	$4.9183\,03e-01$
$1.0e-14$	99867	$4.9238\,53e-01$	$4.9238\,55e-01$	$4.9237\,47e-01$
Subpath $(i_2,\ i_{10}, i_8)$ with $\rho = 0$				
$1.0e+0$	10	$2.2189\,88e-01$	$3.2041\,96e-01$	$2.3968\,47e-01$
$1.0e-1$	18	$2.5766\,47e-01$	$2.6520\,73e-01$	$2.4930\,98e-01$
$1.0e-3$	30	$2.6092\,17e-01$	$2.6101\,97e-01$	$2.5401\,91e-01$
$1.0e-5$	56	$2.6096\,75e-01$	$2.6096\,84e-01$	$2.5726\,18e-01$
$1.0e-10$	180	$2.6096\,79e-01$	$2.6096\,79e-01$	$2.5981\,89e-01$
$1.0e-14$	99867	$2.6098\,85e-01$	$2.6098\,88e-01$	$2.6096\,59e-01$

uniformization algorithm. In all cases, the discretization step h allows for obtaining stochastic matrices $\mathbf{P}_h^{\mathbf{u}}$.

The results of the uniformization are more accurate than those of the discretization approach. From [127] we know that discretization approach converges towards the optimal gain vector for $h \to 0$. *However, discretization method is much slower than uniformization, its runtime difference is a factor of* 1000. *Let* λ_{max} *be the maximal exit rate in matrix* $\mathbf{Q}^{\mathbf{u}}$ *for some decision vector* $\mathbf{u}$. *As estimated in [42], the discretization method requires* $n \approx \lambda_{max} T/\varepsilon$ *iterations to reach a global accuracy of* ε. *In the small graph example, for* $\lambda_{max} = 7$, $T = 1$ *and* $\varepsilon = 0.00001$, *uniformization needs only* 56 *iterations. By contrast, discretization needs about* 180000 *iterations to obtain a comparable result* $4.92375214e-01$.

Results obtained for interval length $T = 3$ *Table 5.4 shows the results for the probability gained by both policies computed with uniformization and discretization in the interval* $[0,3]$. *The initial probability distribution is* $(0.5, 0.5)$.

For $T = 3$ *and* $\varepsilon = 0.001$, *even* 3000 *iterations of discretization method are required to reach a comparable value* $5.95513742e-01$ *which is computed with only* 100 *iterations of the uniformization. As already mentioned in [42], uniformization is much more efficient.*

Table 5.4.: Probability bounds and approximate probabilities for the routes (i_2, i_3, i_4) and (i_2, i_{10}, i_8) in the small graph example shown in Fig. 5.15. The interval length is 3. The initial probability distribution is $(0.5, 0.5)$.

Subpath (i_2, i_3, i_4) with $\rho = 0.3$				
Uniformization				Discretization
ε	# Iter.	Bounds for G^*		Approximation for G^*
$1.0e+0$	39	$5.3896\,15e-01$	$6.2728\,62e-01$	$5.9078\,09e-01$
$1.0e-1$	58	$5.8952\,88e-01$	$5.9877\,11e-01$	$5.9234\,47e-01$
$1.0e-3$	106	$5.9551\,08e-01$	$5.9560\,96e-01$	$5.9380\,46e-01$
$1.0e-5$	179	$5.9557\,58e-01$	$5.9557\,68e-01$	$5.9452\,61e-01$
$1.0e-10$	600	$5.9557\,64e-01$	$5.9557\,64e-01$	$5.9526\,28e-01$
$1.0e-14$	299844	$5.9557\,63e-01$	$5.9557\,65e-01$	$5.9557\,58e-01$
Subpath (i_2, i_{10}, i_8) with $\rho = 0$				
$1.0e+0$	38	$5.2582\,22e-01$	$6.1790\,76e-01$	$5.7327\,97e-01$
$1.0e-1$	57	$5.7742\,46e-01$	$5.8680\,89e-01$	$5.7671\,09e-01$
$1.0e-3$	104	$5.8344\,25e-01$	$5.8354\,20e-01$	$5.7979\,35e-01$
$1.0e-5$	177	$5.8350\,75e-01$	$5.8350\,85e-01$	$5.8132\,87e-01$
$1.0e-10$	590	$5.8350\,82e-01$	$5.8350\,82e-01$	$5.8285\,53e-01$
$1.0e-14$	299844	$5.8350\,81e-01$	$5.8350\,83e-01$	$5.8350\,69e-01$

Results obtained for interval length $T = 6$ *Table 5.5 shows the results for the probability gained by both policies computed with uniformization and discretization in the larger interval* $[0,6]$. *It holds that* $\pi = (0.5, 0.5)$.

For the larger interval $T = 6$ *and* $\varepsilon = 0.001$*, uniformization needs* 219 *iterations, whereas discretization needs about* 4300 *iterations to obtain a comparable result* $7.8832\,67e-01$. *We summarize the results obtained with an accuracy of* $\varepsilon = 0.001$ *in Fig. 5.16. The discretization step h is set in such a way that discretization runs the same number of iterations as uniformization.*

Table 5.5.: Probability bounds and approximate probabilities for the routes (i_2, i_3, i_4) and (i_2, i_{10}, i_8) in the small graph example shown in Fig. 5.15. The interval lenght is 6 and $\pi = (0.5, 0.5)$.

Subpath (i_2, i_3, i_4) with $\rho = 0.3$				
Uniformization				Discretization
ε	# Iter.	Bounds for G^*		Approximation for G^*
$1.0e+0$	85	$7.1468\,65e-01$	$8.0800\,19e-01$	$7.8442\,81e-01$
$1.0e-1$	120	$7.8072\,64e-01$	$7.9031\,66e-01$	$7.8559\,28e-01$
$1.0e-3$	219	$7.8832\,52e-01$	$7.8842\,42e-01$	$7.8686\,69e-01$
$1.0e-5$	374	$7.8840\,41e-01$	$7.8840\,51e-01$	$7.8750\,51e-01$
$1.0e-10$	1260	$7.8840\,49e-01$	$7.8840\,49e-01$	$7.8813\,81e-01$
$1.0e-14$	599852	$7.8841\,05e-01$	$7.8841\,06e-01$	$7.8840\,43e-01$
Subpath (i_2, i_{10}, i_8) with $\rho = 0$				
$1.0e+0$	80	$7.7619\,63e-01$	$8.6297\,47e-01$	$8.4245\,42e-01$
$1.0e-1$	117	$8.3880\,91e-01$	$8.4839\,74e-01$	$8.4376\,63e-01$
$1.0e-3$	210	$8.4648\,99e-01$	$8.4658\,76e-01$	$8.4501\,35e-01$
$1.0e-5$	359	$8.4656\,88e-01$	$8.4656\,98e-01$	$8.4566\,11e-01$
$1.0e-10$	1219	$8.4656\,96e-01$	$8.4656\,96e-01$	$8.4630\,25e-01$
$1.0e-14$	599852	$8.4657\,50e-01$	$8.4657\,51e-01$	$8.4656\,91e-01$

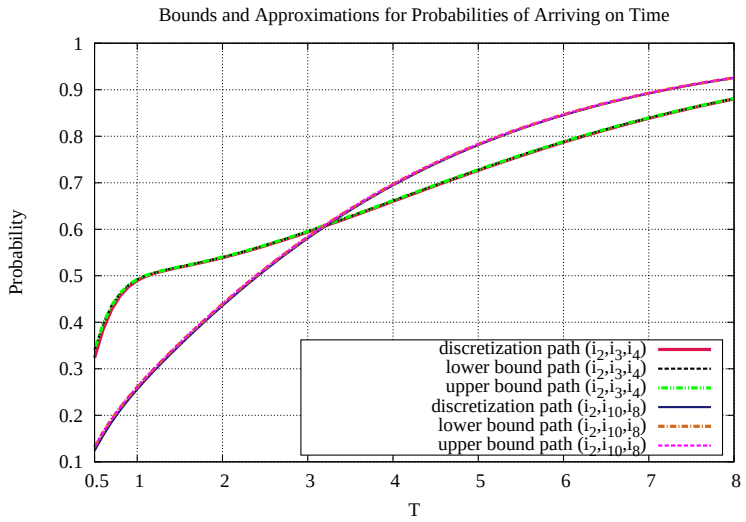


Figure 5.16.: Bounds and approximations for transient probabilities computed with the uniformization and discretizaton approach.

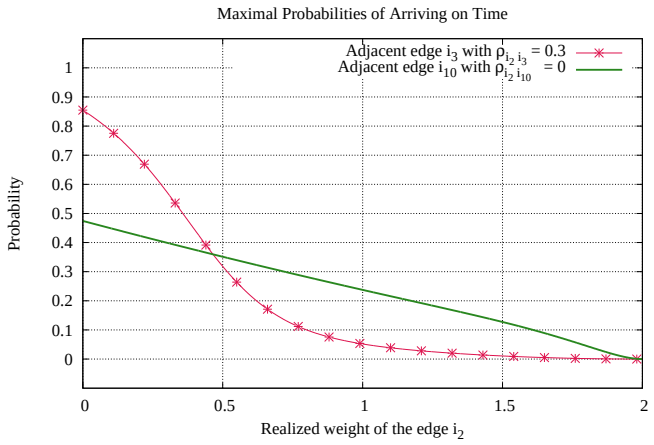


Figure 5.17.: Transient probabilities of arriving at the node $v_{fin}(i_3)$ and $v_{fin}(i_{10})$ with a path weight of less or equal to 3.

Conditional probability of arriving on time *In the real system, decisions have to depend on history vector $\bar{\psi}_{(i_1,w_1,\ldots,i_l,w_l)}$ and the remaining time to the deadline. We computed history vectors $\bar{\psi}_{(i_2,w_2,i_l,0)}$ for subroutes (i_2,i_3), (i_2,i_{10}), where $w_2 \in [0,2]$, $l = 3, 10$. Then the probabilities of passing the adjacent edges i_3, i_{10} in the time interval $[0,3]$ are computed with discretization approach. The results are shown in Figure 5.17. Observe that $\bar{\psi}_{(i_2,w_2,i_3,0)} \approx (0,1)$ for $w_2 \geq 1$, since $\bar{\psi}_{(i_2,w_2,i_{10},0)} = (0.5,0.5)$ for all $w_2 \in [0,2]$. We analyzed the subroutes (i_2,i_3), (i_2,i_{10}) for different correlation coefficients and different deadlines using history vectors $\bar{\psi}_{(i_2,w_2,i_l,0)}$, $w_2 \in [0,2]$. The resulting probabilities of arriving on time are summarized graphically below.*

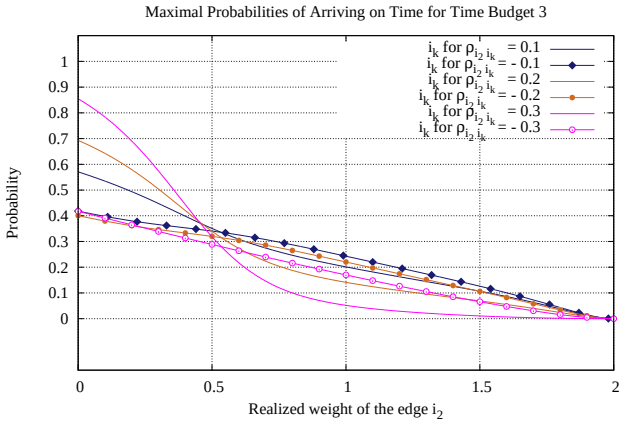


Figure 5.18.: Transient probabilities for $T = 3$ depending on the realized weight of edge i_2 and correlation coefficient of the adjacent edge i_k.

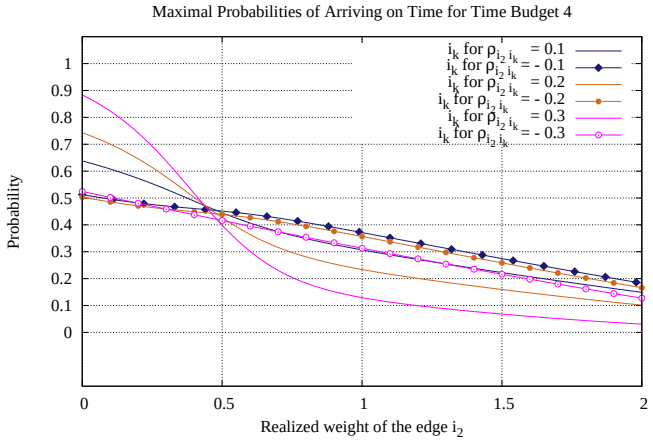


Figure 5.19.: Transient probabilities for $T = 4$ depending on the realized weight of edge i_2 and correlation coefficient of the adjacent edge i_k.

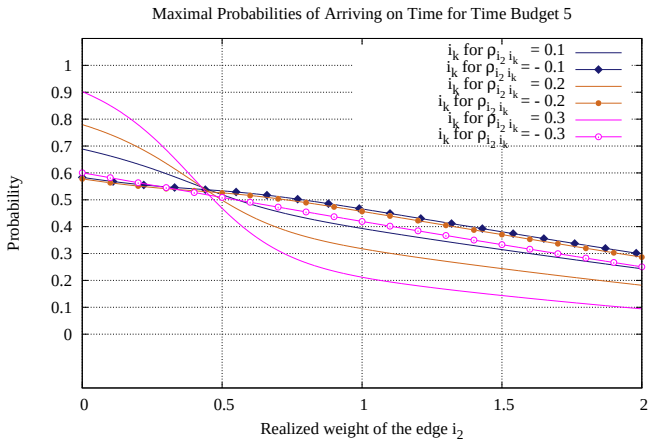


Figure 5.20.: Transient probabilities for $T = 5$ depending on the realized weight of edge i_2 and correlation coefficient of the adjacent edge i_k.

5.7. Model Checking Algorithms

Model checking for CTMDPs is a well established technique to verify quantitative properties of systems where some external control is available. Various performance properties such as performability, availability in a given time interval, dependability, interval and reward bounded reachability can be proved or disproved [18, 16, 42].

For example, consider the scheduling problem with a fixed number of jobs with exponentially distributed completion times and identical machines [18]. Computation

of the optimal policy maximizing the probability of meeting a time deadline can be reduced to the interval bounded reachability probability in a CTMDP [18]. Consider a fault-tolerant multiprocessor system with exponentially distributed times to failure and repair times. Availability of the system in the time interval $[0,T]$ using single repair unit with preemptive priorities can be analyzed. Computation of state-dependent repair priorities corresponds to the computation of the optimal policy maximizing availability which is the problem of maximizing the accumulated reward of a CTMDP in a finite interval $[0,T]$ [42].

We first introduce the Continuous Stochastic Logic (CSL) which is a formal method for system verification providing specification of performance measures of interest for CTMCs. Model checking problems of CSL can be interpreted in terms of decision problems in CTMDPs. The authors in [20] proposed an extention of CSL over CTMDPs and introduced an approach exploiting computational methods from CTMDP field to prove or disprove CSL formulae. According to the results from [42], model checking of CSL formulae can be linked to the computation of an optimal gain vector in CTMDPs. Since our PH-Graph model can be easily converted to a CTMDP with constant rewards, paths in a PH-Graph describe CTMCs. Thus the PH-Graph model is exactly the one for which the model checking of CSL formulae can be applied. We first introduce the logic CSL and then describe the model checking algorithm for CTMDPs based on work in [16, 17, 42].

5.7.1. Continuous Stochastic Logic

The logic CSL is a stochastic extension of Computational Tree Logic (CTL) [21]. It is widely used to specify and evaluate quantitative properties of systems as logical formulae over CTMCs [12, 17, 16, 19]. Basically, the syntax of CSL contains *state*- and *path*-formulae. Let $\mathbf{r}$ be a reward vector for a CTMDP and AP be a set of atomic propositions. J and T are intervals on $\mathbb{R}_{\geq 0}$, where interval T specifies some time-bound and interval J specifies some reward-bound. Let p be a probability and $\bowtie$ a comparison operator from the set $\{\leq,\geq,<,>\}$. The syntax of CSL can be defined as [16, 42]

State-formulae

$$\Phi ::= a \mid \neg\Phi \mid \Phi \vee \Phi \mid S_J(\Phi) \mid P_J(\phi) \mid C_J^T(\Phi) \mid I_J^t(\Phi),$$

Path-formulae

$$\phi ::= \Phi U^T \Psi \mid X^T \Phi,$$

where Φ and Ψ are state formulae. State-formulae are interpreted over the states of a CTMDP. The model checking problem for a state-formula Φ is to determine the set of states $Sat(\Phi)$ satisfying Φ. Formally, a state $i \in \mathscr{S}$ and CSL formula Φ belongs to the relation $\models$ which is denoted by $i \models \Phi$ if Φ is true in state i. The relation $\models$ is the so-called satisfaction relation (see, e.g., [16] for details). Usually, a CTMDP satisfies a CSL state-formula Φ if the initial state fulfills it. We denote by $\mathbf{r}|_{Sat(\Phi)}$ the vector elements belonging to states satisfying some formula Φ as defined in [42].

Furthermore, the following holds for CSL formulae:

- We have that $\phi \models \Phi U^T \Psi$ iff $\exists t' \in [0,T]$ with $\omega(t') \models \Psi \wedge \forall 0 \geq t \leq t'$ holds $\omega(t) \models \Phi$, where $\omega \in \Omega$ is a sample path when policy $\mathbf{d} \in \Pi$ is used (see Page 16 in Sec. 2.2).

- The path-formula $\phi \models X^T \Phi$ is fulfilled iff $\exists t' \in [0,T]$ with $\omega(t') \models \Phi \wedge \forall t < t'$ holds $\omega(t) = \omega(0)$. $\phi \models X^T \Phi$ means that the next state is reached and satisfies the state-formula Φ [16].

- $P_J(\phi)$ defines the probabilistic operator. CTL state-formula $P_{\bowtie p}(\phi)$ states that the probability of the paths satisfying the path-formula ϕ obeys the bound given by $\bowtie p$ [16]. E.g., $P_{\leq 0.5}(\Phi U^{[t,T]} \Psi)$ states that the probability of reaching a Ψ-state along Φ-states within $t' \in [t,T]$ time units meets the bound ≤ 0.5. The formula $P_{\leq 0.9}(true\, U^{[t,T]} \Psi)$ states that the probabilty of reaching a Ψ-state within $t' \in [t,T]$ time units meets a bound ≤ 0.9. In this case $P_J(\phi)$ is called a *reachability probability* (see [20, 42] and references therein). Formally, it holds that

$$i \models P_J(\phi) \qquad \text{iff } Prob(\omega \in \Omega, \omega(0) = i \,|\, \omega \models \phi) \in J$$

  for all policies $\mathbf{d} \in \Pi$, where ω is a sample path when policy $\mathbf{d}$ is used. The above probability is also denoted as $Prob_i^{\mathbf{d}}$ [42].

  If optimal values and optimal policies in a CTMDP are considered, the following extreme reachability probabilities for Ψ-states can be defined [42]

$$Prob_i^{min}(\Phi U^T \Psi) = \inf_{\mathbf{d} \in \Pi} Prob_i^{\mathbf{d}}(\Phi U^T \Psi)$$

  and

$$Prob_i^{max}(\Phi U^T \Psi) = \sup_{\mathbf{d} \in \Pi} Prob_i^{\mathbf{d}}(\Phi U^T \Psi).$$

  These probabilities correspond to the optimal values, i.e., values of the optimal policies $\mathbf{d} \in \Pi$ according to the probabilistic operator $P_J(\Phi U^T \Psi)$. The policy $\mathbf{d} \in \Pi$ is optimal if it leads to optimal values $Prob_i^{\mathbf{d}}(\Phi U^T \Psi) = Prob_i^{min}(\Phi U^T \Psi)$ or $Prob_i^{\mathbf{d}}(\Phi U^T \Psi) = Prob_i^{max}(\Phi U^T \Psi)$. Furthermore, the formula $P_J(\Phi U^T \Psi)$ is valid in state i iff $Prob_i^{min}(\Phi U^T \Psi) \geq \inf J$ (or $Prob_i^{max}(\Phi U^T \Psi) \geq \sup J$) as given in [42].

- $S_J(\Phi)$ refers to the steady-state behavior of an underlying Markov process. For a reward structure of constant reward rate 1, $\mathscr{S}$ is called a steady-state probability operator with the same interpretation as in CSL for CTMCs [16, 42]. In CSL for CTMCs the state-formula $S_{\bowtie p}(\Phi)$ is fulfilled in the set of states if the steady-state probability for these states obeys the bound given by $\bowtie p$ [16].

  For CTMDPs the formula $S_J(\Phi)$ specifies that the long-run average reward in $Sat(\Phi)$-states is in interval J. Formally, we have

$$i \models S_J(\Phi) \qquad \text{iff } \lim_{T \to \infty} \frac{1}{T}\, \mathbf{g}_{0,T}^{\mathbf{d}}(i) \in J$$

  for all policies $\mathbf{d} \in \Pi$. Considering the optimal reward values for the long-run average reward the minimal value can be defined as

$$Rew_i^{min}(S\Phi) = \inf_{\mathbf{d} \in \Pi} (\lim_{T \to \infty} \frac{1}{T}\, \mathbf{g}_{0,T}^{\mathbf{d}}|_{Sat(\Phi)}(i)),$$

  and the maximum value as

$$Rew_i^{max}(S\Phi) = \sup_{\mathbf{d} \in \Pi} (\lim_{T \to \infty} \frac{1}{T}\, \mathbf{g}_{0,T}^{\mathbf{d}}|_{Sat(\Phi)}(i)).$$

- $C_J^T(\Phi)$ specifies that the reward gained during the interval $[0,T]$ meets the bound defined by J. Formally, it holds that

$$i \models C_J^{[t,T]}(\Phi) \qquad \text{iff } \mathbf{V}^{\mathbf{d}}_{0,t}(i,:) \cdot \mathbf{g}^{\mathbf{d}}_{t,T}|_{Sat(\Phi)} \in J,$$

  Here the gain vector $\mathbf{g}^{\mathbf{d}}_{t,T}$ is defined in (2.21) with the terminal gain vector $\mathbf{g}_T = \mathbf{0}$.

  Recall that for the given measurable policy $\mathbf{d}$ the matrix $\mathbf{V}^{\mathbf{d}}_{r,t}$ with $0 \le r \le t \le T$ is the Markov transition matrix defined in (2.17). The $\mathbf{V}^{\mathbf{d}}_{0,t}(i,:)$ denotes the row corresponding to the state i. It gives the distribution at time t starting with state i and following the policy $\mathbf{d}$ in the interval $[0,t]$. The value at position $\mathbf{V}^{\mathbf{d}}_{r,t}(i,j)$ gives the probability that the CTMDP is at time t in state j under the condition that it was at time r in state i and the policy $\mathbf{d}$ is used in the interval $[r,t]$. The matrix $\mathbf{V}^{\mathbf{d}}_{r,t}$ can be used to obtain the distribution at time t under policy $\mathbf{d}$ as $\mathbf{p}^{\mathbf{d}}_t = \pi \mathbf{V}^{\mathbf{d}}_{0,t}$ Eq. 2.19.

  Optimal values are defined as

$$Rew_i^{min}(C^{[t,T]}\Phi) = \inf_{\mathbf{d}\in\Pi} (\mathbf{V}^{\mathbf{d}}_{0,t}(i,:) \cdot \mathbf{g}^{\mathbf{d}}_{t,T}|_{Sat(\Phi)}),$$

  or

$$Rew_i^{max}(C^{[t,T]}\Phi) = \sup_{\mathbf{d}\in\Pi} (\mathbf{V}^{\mathbf{d}}_{0,t}(i,:) \cdot \mathbf{g}^{\mathbf{d}}_{t,T}|_{Sat(\Phi)}).$$

- $I_J^t(\Phi)$ refers to the instantaneous reward gained at time t. If Φ is valid in state i then the instantaneous reward at time t in $Sat(\Phi)$-states is in the interval J. Formally, it holds that

$$i \models I_J^t(\Phi) \qquad \text{iff } \mathbf{V}^{\mathbf{d}}_{0,t}(i,:) \cdot \mathbf{r}|_{Sat(\Phi)} \in J.$$

  Optimal values are defined as

$$Rew_i^{min}(I^t\Phi) = \inf_{\mathbf{d}\in\Pi} (\mathbf{V}^{\mathbf{d}}_{0,t}(i,:) \cdot \mathbf{r}|_{Sat(\Phi)}),$$

  or

$$Rew_i^{max}(I^t\Phi) = \sup_{\mathbf{d}\in\Pi} (\mathbf{V}^{\mathbf{d}}_{0,t}(i,:) \cdot \mathbf{r}|_{Sat(\Phi)}).$$

5.7.2. Algorithms

Model checking of probabilistic and reward properties requires the computation of optimal values and optimal policies for a given CTMDP. For instance, the optimal value $Prob_i^{min/max}(\Phi U^T \Psi)$ should be determined to prove whether state i satisfies state-formula $P_J(\Phi U^T \Psi)$. In this section we describe model checking algorithms and specific reward structures to decide whether $i \in \mathscr{S}$ fulfills some CSL formula Φ.

In order to model check a given CTMDP one should determine the optimal gain vector $\mathbf{g}^*_{0,T}$ defined in (2.21). Recall that the expected reward under policy $\mathbf{d}$ with initial distribution π in the time interval $[0,T]$ is given by $\pi \mathbf{g}^{\mathbf{d}}_{0,T}$ (cf. Eq. 2.22). $\mathbf{g}^{\mathbf{d}}_{0,T}$ is the gain vector containing at position i the expected reward accumulated in the interval $[0,T]$ if the CTMDP is in state i at time 0 and policy $\mathbf{d} \in \Pi$ is used. It is computed

$$\mathbf{g}^{\mathbf{d}}_{0,T} = \mathbf{V}^{\mathbf{d}}_{0,T}\mathbf{g}_T + \int_0^T \mathbf{V}^{\mathbf{d}}_{\tau,T}\mathbf{r}^{\mathbf{d}}\,\mathrm{d}\tau$$

as given in Eq. 2.21. The vector $\mathbf{g}_T$ is the policy-independent initial gain vector at time T. The second term describes the accumulated gain until time T.

The terminal gain vector $\mathbf{g}_T$ is initialized differently to verify various model checking properties for a given CTMDP. In the following we consider maximization problems. Then the optimal gain vector $\mathbf{g}^*_{0,T}$ corresponds to the maximal gain that can be accumulated in the interval $[0,T]$ when the optimal policy is used (cf. Eq. 2.26 and Eq. 2.27). According to [42, Eq. 3] the measurable policy $\mathbf{d}^*$ maximizing the gain vector in the interval $[0,T]$ in all elements is optimal when (cf. Eq. 5.50)

$$\mathbf{d}^* = \arg\max_{\mathbf{d}\in\Pi}\left(\mathbf{V}^{\mathbf{d}}_{0,T}\mathbf{g}_T + \int_0^T \mathbf{V}^{\mathbf{d}}_{0,t}\mathbf{r}^{\mathbf{d}}\,\mathrm{d}t \text{ in all elements}\right), \tag{5.64}$$

A solution to the above problem has been proposed by B.L. Miller in [127] where an algorithm for the computation of the optimal policy $\mathbf{d}^*$ maximizing the expected reward over the finite planning horizon $[0,T]$ and the corresponding optimal gain vector $\mathbf{g}^*_{0,T}$ has been developed (cf. Alg.5.2 in Sec. 5.6). Numerical solution techniques described in Sec. 5.6 can be applied to approximate the optimal gain vector or to compute lower and upper bounds for it.

Maximal probability for timed reachability We consider the computation of the $\sup_{\mathbf{d}\in\Pi} Prob^{\mathbf{d}}_i(\Phi U^T \Psi)$. Generally, the computation of $Prob^{\mathbf{d}}_i(\Phi U^T \Psi)$ can be reduced to the computation of a transient probability. According to the method described in [16], a CTMDP is transformed to a CTMDP where states satisfying $\neg\Phi \vee \Psi$ are made absorbing. This is due to the fact that once some state satisfying formula Ψ along a Φ-path has been reached, the behavior afterwards is not relevant [42]. In that case the CSL path-formula $\Phi U \Psi$ is satisfied. Now the optimal gain vector $\mathbf{g}^*_{0,T}$ can be used to compute the maximal reachability probability. The reward vector is set to zero vector, i.e., $\mathbf{r} = \mathbf{0}$, and the terminal gain vector is initialized as follows

$$\mathbf{g}_T(i) = \begin{cases} 1 & \text{if } i \in Sat(\Psi) \\ 0 & \text{otherwise} \end{cases} \tag{5.65}$$

It holds that $\mathbf{g}_T(i) = 1$ for all absorbing states $i \in \mathscr{S}_A$ which are also called *goal* states and $\mathbf{g}_T(i) = 0$ for all $i \in \mathscr{S}_T$.

According to [42, Lemma 1] the optimal policy $\mathbf{d}^*$ maximizing the gain vector in the interval $[0,T]$ in all elements can be obtained using algorithms 5.3, 5.4 with described settings. E.g., we can compute the optimal policy $\mathbf{d}^*$ using the discretization based algorithm 7.1 with $\mathbf{r} = \mathbf{0}$, $\mathbf{g}_T$ as in (5.65) and time horizon $[0,T]$ [42]. The algorithm computes the approximation of the optimal gain vector $\mathbf{g}^*_{0,T}$.

Since states satisfying $\neg\Phi \vee \Psi$ are made absorbing, the maximal transient probability is equal to the maximal time bounded reachability [16, 42]. Then it holds for the maximal reachability probability $Prob^{max}_i(\Phi U^T \Psi) = \mathbf{g}^*_{0,T}(i)$.

In fact, the PHG problem of computing the maximal probability of arriving on time corresponds to the problem of computing the maximal probability for timed reachability. Then the discretization based algorithm 7.1 can be applied to determine the path maximizing probability of reaching a destination node in the interval $[0,T]$ (see Sec. 5.6).

Interval maximal probability for timed reachability Here we describe how the interval bounded reachability probability can be solved by computation of the optimal gain vector $\mathbf{g}^*_{t,T}$ for the given interval. We consider now the computation of the $\sup_{\mathbf{d}\in\Pi} Prob_i^{\mathbf{d}}(\Phi U^{[t,T]}\Psi)$ along the lines of [42]. Let $t > 0$ and $T \geq 0$. First, the optimal gain vector $\mathbf{g}^*_{t,T}$ accumulated in the interval $[t,T]$ can be computed with the method described above. In particular, the maximal probability $Prob_i^{max}(\Phi U^{[t,T]}\Psi)$ is determined for the CTMDP where states satisfying $\neg\Phi \vee \Psi$ are made absorbing. For example, the optimal policy at time t $\mathbf{d}^*_{t,T}$ can be obtained using the discretization based algorithm 7.1. Then this policy should be extended in the interval $[0,t)$ which results in the following maximization problem [42, Eq. 6]

$$\mathbf{g}^*_{0,T} = \max_{\mathbf{d}\in\Pi} \left(\mathbf{V}^{\mathbf{d}}_{0,t}\mathbf{g}^*_{t,T}\right).$$

Secondly, the CTMDP is transformed to an instance where states satisfying $\neg\Phi$ are made absorbing. Since the computed gain vector $\mathbf{g}^*_{t,T}$ is now used as terminal gain vector, the vector elements corresponding to states satisfying $\neg\Phi$ should be set to zero. It holds that

$$\mathbf{g}'_T(i) = \begin{cases} \mathbf{g}^*_{t,T}(i) & \text{if } i \in Sat(\Phi) \\ 0 & \text{otherwise.} \end{cases} \tag{5.66}$$

Then, the uniformization-based algorithm 5.3 or discretization-based algorithm 7.1 with $\mathbf{r} = \mathbf{0}$, $\mathbf{g}'_T$ as in (5.66) and time horizon $[0,t]$ can be used to obtain the optimal policy $\mathbf{d}^*_{0,t}$ extending the policy $\mathbf{d}^*_{t,T}$. The resulting gain vector $\mathbf{g}^*_{0,t}$ can be used to verify the desired property as $Prob_i^{max}(\Phi U^{[t,T]}\Psi) = \mathbf{g}^*_{0,t}(i)$ [42].

Cumulative reward We consider the computation of the $Rew_i^{max}(C^{[0,T]}\Phi)$ for a given CTMDP. Using uniformization-based algorithm 5.3 one can compute bounds for the optimal value of reward gained during the interval $[0,T]$. Using the discretization-based approach one can approximate the optimal gain vector $\mathbf{g}^*_{0,T}$ as given in Alg. 7.1. The terminal gain vector should be initially set to zero, i.e., $\mathbf{g}_T = \mathbf{0}$.

The optimal policy $\mathbf{d}^*$ results from, e.g., the discretization based approach 7.1 with $\mathbf{r}|_{Sat(\Phi)}$, $\mathbf{g}_T = \mathbf{0}$ and time horizon $[0,T]$ [42]. The algorithm computes an approximation of the optimal gain vector $\mathbf{g}^*_{0,T}|_{Sat(\Phi)}$. Then the optimal cumulative reward for a given state i can be obtained as $Rew_i^{max}(C^{[0,T]}\Phi) = \mathbf{g}^*_{0,T}(i)$.

Interval cumulative reward Now the computation of the optimal interval cumulative reward $Rew_i^{max}(C^{[t,T]}\Phi)$ can be described based on [42]. Let $t > 0$ and $T \geq 0$. For the maximal interval cumulative reward $Rew_i^{max}(C^{[t,T]}\Phi)$ computation of cumulative rewards and reachability probability should be combined. First, the optimal cumulative reward value is computed for the interval $[t,T]$. For example, one can use discretization algorithm 7.1 with $\mathbf{r}|_{Sat(\Phi)}$, $\mathbf{g}_T = \mathbf{0}$ and time horizon $[t,T]$. The optimal gain vector $\mathbf{g}^*_{t,T}|_{Sat(\Phi)}$ is returned by the algorithm which is the maximal gain accumulated in the interval $[t,T]$.

Now the vector $\mathbf{g}^*_{t,T}|_{Sat(\Phi)}$ can be used to compute the maximal reachability probability. An extension of the policy in the interval $[0,t)$ can be defined as the computation of policy $\mathbf{d}^*$ (cf. Eq. 5.50) corresponding to

$$\mathbf{g}^*_{0,t} = \max_{\mathbf{d}\in\Pi} \left(\mathbf{V}^{\mathbf{d}}_{0,t}\mathbf{g}^*_{t,T}|_{Sat(\Phi)}\right).$$

The terminal gain vector is set to $\mathbf{g}^*_{t,T}|_{Sat(\Phi)}$ and the maximal reachability can be computed using the discretization algorithm with $\mathbf{r} = \mathbf{0}$, $\mathbf{g}_T = \mathbf{g}^*_{t,T}|_{Sat(\Phi)}$ and time horizon $[0,T]$.

Instantaneous reward We consider the computation of the $Rew_i^{max}(I^t\Phi)$ for a given CTMDP which is the maximal instantaneous reward gained at time t as given in [42]. The terminal gain vector is set to the reward vector $\mathbf{r}|_{Sat(\Phi)}$. Then the discretization algorithm 7.1 is called with $\mathbf{r} = \mathbf{0}$, $\mathbf{g}_T = \mathbf{r}|_{Sat(\Phi)}$ and time horizon $[0,t]$. The optimal instantaneous reward for a given state i is then $Rew_i^{max}(I^t\Phi) = \mathbf{g}^*_{0,t}(i)$ [42].

***Example* 5.9.** *We consider a graph visualized in Fig. 5.21 to compute the maximal reachability probability for the node v_{fin}. The corresponding PH-Graph is shown in Fig. 5.22 where decisions are highlighted in red, black, and blue. The parameterization of the PHG is briefly described in Example 5.7.*

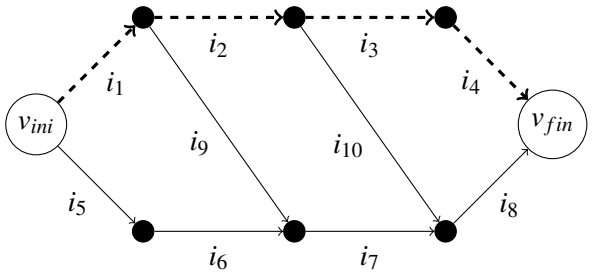


Figure 5.21.: Acyclic graph for shortest path computation. Edges with correlated weights are highlighted in dashed style.

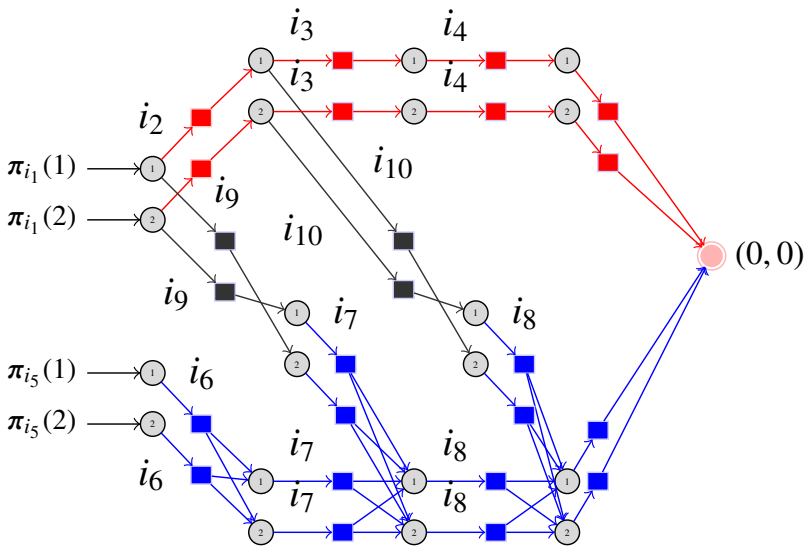


Figure 5.22.: The PHG corresponding to the acyclic graph in Fig. 5.21.

The mean weight of every path between v_{ini} and v_{fin} is 4.2857. *We consider a single atomic proposition* $(0,0)$ *which holds only in the goal state* $(0,0)$ *(the state* $(0,0)$ *is*

absorbing). The weights of all edges are described by the 2-order hyperexponential PHD.

To examine the maximal reachability probability for the state $(0,0)$ *we first analyze the formula* $\Phi = P_J(\Diamond^{[0,T]}(0,0))$ *for state* $(i_1,1)$. *Let* $true = a \vee \neg a$ *and* $\Diamond^{[0,T]}(\Phi) = true\,U^{[0,T]}\,\Phi$ *as given in [17, 42]. The CSL formula can be verified by computing* $\sup_{\mathbf{d}\in\Pi} Prob^{\mathbf{d}}_{(i_1,1)}(true\,U^{[0,T]}(0,0)) = Prob^{max}_{(i_1,1)}(\Diamond^{[0,T]}(0,0))$ *which corresponds to the optimal value* $\mathbf{g}^*_{0,T}(i_1,1)$.

The decision that maximizes the probability to reach $(0,0)$ *in the time interval* $[0,T]$ *depends on the phase of the PHD describing the edge weight. We verified the formula* $P_{\geq 0.5}(\Diamond^{[0,T]}(0,0))$ *for states* $(i_1,1)$, $(i_1,2)$ *and* $(i_5,1)$, $(i_5,2)$ *and different values of* T.

Tables 5.6-5.9 contain the results and efforts for computing the maximal reachability probabilities for $T = 1.5$, 3.5 *and* 5.5 *using the uniformization based algorithm 5.3.*

We analyze $\Phi_1 = P_{\geq 0.5}(\Diamond^{[0,1.5]}(0,0))$ *for state* $(i_1,1)$ *and* $\Phi_2 = P_{\geq 0.5}(\Diamond^{[0,3.5]}(0,0))$ *for state* $(i_1,2)$. *From Table 5.6 one can see that* $Prob^{max}_{(i_1,1)}(\Diamond^{[0,1.5]}(0,0)) = \mathbf{g}^*_{0,1.5}(i_1,1) = 9.9285e-01$ *for* $\varepsilon = 1.0e-5$, *which meets the specified probability bound. Thus we verify* $(i_1,1) \models \Phi_1$. *From Table 5.7 one can also see that* $Prob^{max}_{(i_1,2)}(\Diamond^{[0,3.5]}(0,0)) = \mathbf{g}^*_{0,3.5}(i_1,2) = 3.5703e-01$ *for* $\varepsilon = 1.0e-5$, *implying that* $(i_1,2) \not\models \Phi_2$.

5.8. Summary

In this chapter, we have brought together results on general solution methods for SSPP with recent results on solutions for SSPP including correlations. First, we show how the PHG model can be mapped to a CTMDP in order to find the optimal policy in stochastic graph with dependent weights on edges. We investigated two route planning problems under uncertainty where optimal policies depend on correlations.

The first problem is the computation of a path with a minimal expected weight. The second problem is finding a path maximizing the probability of arriving on time, i.e., having a weight below some threshold. The last problem corresponds to the computation of optimal policy and the corresponding reward value in the finite horizon CTMDP.

We show that standard methods from SSPPs and finite horizon CTMDPs can be applied to obtain optimal policies and corresponding reward values. However, existing approaches should be adapted such that optimal solutions depend on realized weights on edges and not on states of the CTMDP. The obtained results provide evidence that considering correlations results in much more realistic solutions on stochastic graphs with dependent weights on adjacent edges.

As shown by some examples, depending on the weight of the previous edge, the decision to choose a correlated adjacent edge could result in a shorter path weight than the choice of an uncorrelated adjacent edge. But if the previous edge has a large weight, choosing an uncorrelated adjacent edge could result in a much better path than the choice of a correlated edge. Thus approaches considering uncertain parameters and correlations gain relevance in most practical situations, e.g., in route planning problems with congestions when information about realized traveling times becomes available.

Table 5.6.: Bounds for the probability of reaching $(0,0)$ in $[0,T]$, i.e., $Prob^{max}_{(i_1,1)}(\Diamond^{[0,T]}(0,0))$.

	$(i_1,1)$. Time bounded reachability probability			
Interval $[0,T]$	ε	lower bound	upper bound	# Iter
$T=1.5$	$1.0e-4$	$9.9284e-01$	$9.9285e-01$	67
$T=1.5$	$1.0e-5$	$9.9285e-01$	$9.9285e-01$	86
$T=1.5$	$1.0e-9$	$9.9285e-01$	$9.9285e-01$	226
$T=3.5$	$1.0e-4$	$9.9998e-01$	$9.9999e-01$	165
$T=3.5$	$1.0e-5$	$9.9999e-01$	$9.9999e-01$	210
$T=3.5$	$1.0e-9$	$9.9999e-01$	$9.9999e-01$	567
$T=5.5$	$1.0e-4$	$9.9999e-01$	$1.0000e+00$	268
$T=5.5$	$1.0e-5$	$9.9999e-01$	$1.0e+00$	347
$T=5.5$	$1.0e-9$	$1.0e+00$	$1.0e+00$	920

Table 5.7.: Bounds for the probability of reaching $(0,0)$ in $[0,T]$, i.e., $Prob^{max}_{(i_1,2)}(\Diamond^{[0,T]}(0,0))$.

	$(i_1,2)$. Time bounded reachability probability			
Interval $[0,T]$	ε	lower bound	upper bound	# Iter
$T=1.5$	$1.0e-4$	$7.8866e-02$	$7.8876e-02$	66
$T=1.5$	$1.0e-5$	$7.8869e-02$	$7.8870e-02$	86
$T=1.5$	$1.0e-9$	$7.8869e-02$	$7.8869e-02$	225
$T=3.5$	$1.0e-4$	$3.5702e-01$	$3.5703e-01$	160
$T=3.5$	$1.0e-5$	$3.5703e-01$	$3.5703e-01$	209
$T=3.5$	$1.0e-9$	$3.5703e-01$	$3.5703e-01$	559
$T=5.5$	$1.0e-4$	$6.1716e-01$	$6.1717e-01$	259
$T=5.5$	$1.0e-5$	$6.1716e-01$	$6.1716e-01$	347
$T=5.5$	$1.0e-9$	$6.1716e-01$	$6.1716e-01$	899

Table 5.8.: Bounds for the probability of reaching $(0,0)$ in $[0,T]$, i.e., $Prob^{max}_{(i_5,1)}(\Diamond^{[0,T]}(0,0))$.

	$(i_5,1)$. Time bounded reachability probability			
Interval $[0,T]$	ε	lower bound	upper bound	# Iter
$T=1.5$	$1.0e-4$	$3.2903e-01$	$3.2904e-01$	66
$T=1.5$	$1.0e-5$	$3.2903e-01$	$3.2904e-01$	86
$T=1.5$	$1.0e-9$	$3.2904e-01$	$3.2904e-01$	224
$T=3.5$	$1.0e-4$	$6.2705e-01$	$6.2706e-01$	160
$T=3.5$	$1.0e-5$	$6.2706e-01$	$6.2706e-01$	209
$T=3.5$	$1.0e-9$	$6.2706e-01$	$6.2706e-01$	558
$T=5.5$	$1.0e-4$	$8.0768e-01$	$8.0769e-01$	259
$T=5.5$	$1.0e-5$	$8.0769e-01$	$8.0769e-01$	337
$T=5.5$	$1.0e-9$	$8.0769e-01$	$8.0769e-01$	898

Table 5.9.: Bounds for the probability of reaching $(0,0)$ in $[0,T]$, i.e., $Prob^{max}_{(i_5,2)}(\Diamond^{[0,T]}(0,0))$.

	$(i_5,2)$. Time bounded reachability probability			
Interval $[0,T]$	ε	lower bound	upper bound	# Iter
$T=1.5$	$1.0e-4$	$1.1133e-01$	$1.1134e-01$	66
$T=1.5$	$1.0e-5$	$1.1133e-01$	$1.1133e-01$	86
$T=1.5$	$1.0e-9$	$1.1133e-01$	$1.1133e-01$	224
$T=3.5$	$1.0e-4$	$3.78240e-01$	$3.78249e-01$	160
$T=3.5$	$1.0e-5$	$3.7824e-01$	$3.7824e-01$	209
$T=3.5$	$1.0e-9$	$3.7824e-01$	$3.7824e-01$	558
$T=5.5$	$1.0e-4$	$6.1500e-01$	$6.1501e-01$	259
$T=5.5$	$1.0e-5$	$6.1501e-01$	$6.1501e-01$	337
$T=5.5$	$1.0e-9$	$6.1501e-01$	$6.1501e-01$	898

Chapter 6

Transformation of 2-order APHDs for Correlation Fitting

In this chapter, we propose transformation results for PHDs in composition. As introduced in Chap. 3 a PHD composition is used for correlation modeling of edge weights in a stochastic graph. Our aim is to find the matrix representation of two PHDs in composition allowing for the maximal first order joint moment.

In stochastic modeling, PHDs and MAPs have emerged as versatile tools which can be used to approximate any continuous distribution and to model empirical behavior of real world processes. Fitting algorithms allow one to estimate parameters of PHD and MAP matrix representations from observations of a real or a simulated system.

Although a wider spectrum of fitting algorithms exists, parameter fitting for PHDs and MAPs remains a challenging task. The reasons for this are non-unique representations of PHDs, redundancy of the matrix representation, trade-off between accuracy and number of parameters, and complexity of resulting nonlinear optimization problems [165, 93, 96]. Thus fitting algorithms often work on specific representations such as canonical forms or Hyper-Erlang, and hyperexponential forms [93, 168].

In Chapter 4 we presented the fitting approach for transfer matrix $\mathbf{H}_{ij}$. First, two PHDs with representation $(\pi_i, \mathbf{D}_i)$ and $(\pi_j, \mathbf{D}_j)$ were fitted. Then the parameters of the transfer matrix $\mathbf{H}_{ij}$ are determined in order to model the correlation between two PHDs in composition. This means that both matrix representations of PHDs have a non-negligible influence when fitting transfer matrix $\mathbf{H}_{ij}$. Parameters of both PHDs determine constraints that should be met by $\mathbf{H}_{ij}$, thus fixing the range of possible values and possibly limiting the range of achievable correlation of PHD composition.

Similarity transformations compute an equivalent representation of $(\pi, \mathbf{D}_0)$ without altering the distribution. Numerous works on transformations in order to maximize the first joint moment when expanding PHD to a MAP exist [45, 44, 96, 126, 41]. In this chapter, we show which representations are best suitable in order to maximize the first joint moment of PHD composition.

6.1. Similarity Transformation

In this section we present the transformation approach for APHDs described in [45, 44] which generates an equivalent APHD without alternating the distribution function. The transformation can be used, e.g. to increase the number of exit states. The modified representation can have a positive impact on the range of achievable correlation when expanding a PHD to a MAP. Since a higher number of exit states allows for more variability of entries in the matrix $\mathbf{D}_1$, transformations can be helpful in finding of matrix $\mathbf{D}_1$ maximizing the first joint moment, higher order joint moments and lag-k autocorrelations. The main idea of the algorithm is to invert Cumani-steps developed in [56]. Cumani-steps transform PHD representation into series canonical form (cf. Fig. 2.18 in Sec. 2.4.2).

Although transformation steps modify a PHD representation, they still return the acyclic PHD. Each single transformation step is applied to two states i and j with $i < j$, transition rate $\mathbf{Q}(i,j) > 0$, and $\lambda(i) \leq \lambda(j)$. Note that each PHD can be always transformed in a PHD where the last condition is satisfied [56, 166]. The modified parameters are $\pi(i)$, $\pi(j)$, incoming and outgoing transitions of i, and outgoing transitions of j such that π^δ and $\mathbf{D}_0^\delta$ denote vector and matrix after transformation with parameter δ (cf. Eq. 6.3, Eq. 6.4).

First, the transformation parameter δ should be selected from the following interval

$$\left[\max\left(-\pi(i), \min_{l>j, \mathbf{D}_0(j,l)>0}\left(-\frac{\pi(i)\mathbf{D}_0(i,l)}{\mathbf{D}_0(j,l)}\right), -\frac{\pi(i)\mathbf{d}_1(i)}{\mathbf{d}_1(j)}\right), \tag{6.1}$$

$$\min\left(\pi(j), \min_{k<i, \mathbf{D}_0(k,i)>0}\left(\frac{\pi(i)\mathbf{D}_0(k,j)}{\mathbf{D}_0(k,i)}\right), \frac{\pi(i)\mathbf{D}_0(i,j)}{\lambda(j)-\lambda(i)}\right)\right], \tag{6.2}$$

which guarantees that the modified values remain non-negative. If $\lambda(i) = \lambda(j)$, then the third term in the minimum evaluates to ∞ and is not used. If $\mathbf{d}_1(i) = 0$, then the third term evaluates to $-\infty$ and is also not used.

For $\delta > 0$ new initial probabilities and transition rates are computed according to the following rules

$$\pi^\delta(k) = \begin{cases} \pi(i) + \delta & \text{for } k = i \\ \pi(j) - \delta & \text{for } k = j \\ \pi(k) & \text{otherwise} \end{cases} \tag{6.3}$$

$$\mathbf{D}_0(k,l)^\delta = \begin{cases} \mathbf{D}_0(i,j)\frac{\pi(i)}{\pi(i)+\delta} - \frac{(\lambda(j)-\lambda(i))\delta}{\pi(i)+\delta} & \text{for } k = i \text{ and } l = j \\ \mathbf{D}_0(i,l)\frac{\pi(i)}{\pi(i)+\delta} & \text{for } k = i \text{ and } l > i \wedge l < j \\ \mathbf{D}_0(i,l)\frac{\pi(i)}{\pi(i)+\delta} + \mathbf{D}_0(j,l)\frac{\delta}{\pi(i)+\delta} & \text{for } k = i \text{ and } l > j \\ \mathbf{D}_0(k,i)\frac{\pi(i)+\delta}{\pi(i)} & \text{for } k < i \text{ and } l = i \\ \mathbf{D}_0(k,j) - \mathbf{D}_0(k,i)\frac{\delta}{\pi(i)} & \text{for } k < i \text{ and } l = j \\ \mathbf{D}_0(k,l) & \text{otherwise} \end{cases} \tag{6.4}$$

Proofs of validity for transformation steps are given in [45]. Note that diagonal elements of $\mathbf{D}_0$ are not modified by the transformation. The exit vector is transformed according to

$$\mathbf{d}_1^\delta(i) = \frac{\pi(i)\mathbf{d}_1(i) + \delta\mathbf{d}_1(j)}{\pi(i)+\delta} = \mathbf{d}_1(i) + \delta\frac{\mathbf{d}_1(j) - \mathbf{d}_1(i)}{\pi(i)+\delta}, \tag{6.5}$$

such that other elements of $\mathbf{d}_1$ remain unchanged.

6.2. Transformation of 2-order APHDs

The PHG model incorporates composition of two PHDs for modelling edge weights and correlations between them. To optimize the PHD composition fitting procedure it is important to know what is the best representation of both PHDs for the subsequent transfer matrix fitting step. In particular, we consider which representations of two PHDs in composition allow for modeling of a transfer matrix with maximal first joint moment and present transformations to obtain the favorable representations.

Let $((\pi_i, \mathbf{D}_i), (\pi_j, \mathbf{D}_j), \mathbf{H}_{ij})$ be a PHD composition (see Def. 3.1). We consider order 2 APHDs since only little insights are available about which representation for PHDs of order $n > 2$ allows for the largest flexibily according to correlation fitting. In [41] an exact result for 2-order APHD is presented. In fact, the hyperexponential representation is optimal among all possible representations that can be reached using the similarity transformation described in Sec. 6.1. Our aim is to use the technique from [41] to obtain the best representations of 2-order PHDs in composition.

In the following we first examine what is the best representation of PHD PH_j in composition in dependence of the representation of PH_i. Then we study what is the best representation of predecessor PHD PH_i in dependence of the representation of PH_j. This implies that one representation is fixed, such that the transformation is applied to only one PHD. Therefore, the representation of the fixed PHD remains unchanged.

Our analysis results in four cases which are presented below. First we fix the representation of PHD PH_i to hyperexponential or canonical form and examine what is the best representation of the successor PHD PH_j. Then we fix the representation of PH_j in the composition to hyperexponential or canonical form and examine what is the best representation of the predecessor PHD PH_i.

6.2.1. The first case

The first case is given when the representation of PHD PH_i in composition is fixed to hyperexponential form. Then we are interested in the best suitable representation of the successor PHD PH_j in order to maximize the first joint moment of their composition. We assume that the hyperexponential APHD PH_i is given by

$$\mathbf{c} = (c, 1-c)\,,\ \mathbf{D}_i = \begin{pmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{pmatrix}, \tag{6.6}$$

where $-\mathbf{D}_i(1,1) = \mu_1$ is the transition rate of state 1, and $-\mathbf{D}_i(2,2) = \mu_2$ is the transition rate of state 2. Assume $\mu_2 \geq \mu_1$ (if it is not the case, we apply the transformation from [41]).

The second APHD PH_j has representation

$$\pi = (\pi, 1-\pi)\,,\ \mathbf{D}_j = \begin{pmatrix} -\lambda_1 & \lambda_{1,2} \\ 0 & -\lambda_2 \end{pmatrix} = \begin{pmatrix} -1 & \lambda_{1,2} \\ 0 & -\lambda_2 \end{pmatrix} \tag{6.7}$$

, where $-\mathbf{D}_j(1,1) = \lambda_1$ is the transition rate of state 1, $-\mathbf{D}_j(2,2) = \lambda_2$ is the transition rate of state 2, and $\mathbf{D}_j(1,2) = \lambda_{1,2}$ is the transition rate of state 1 to state 2.

Consider the following inequalities

$$0 \leq \pi \leq 1, \lambda_{1,2} \leq 1, \lambda_2 \geq 1, \tag{6.8}$$

such that $\lambda_2 \geq \lambda_1$, and λ_1 has been scaled to 1. In the following we denote states of PH_i as k', and states of PH_j as k for some $k \in \mathscr{S}$.

The transformation of the APHD $(\pi_j, \mathbf{D}_j)$ in the 2-state case (6.7) is according to

$$\pi^{\delta} = \left(\pi^{\delta}, 1-\pi^{\delta}\right) = (\pi+\delta, 1-\pi-\delta),\ \lambda_{1,2}^{\delta} = \frac{\pi\lambda_{1,2}}{\pi+\delta} - \frac{(\lambda_2-1)\delta}{\pi+\delta} \tag{6.9}$$

where δ-bounds are given by the interval

$$\delta \in \left[\max\left(-\pi, -\frac{\pi(1-\lambda_{1,2})}{\lambda_2}\right), \min\left(1-\pi, \frac{\pi\lambda_{1,2}}{\lambda_2-1}\right)\right]. \tag{6.10}$$

Note that only the element $\mathbf{D}_j(1,2)$ is transformed. The boundaries of the interval 6.10 are denoted as δ^- and δ^+. The first joint moment of PH_i and PH_j is given by

$$\mu_{1,1} = \mathbf{c}\mathbf{M}_i\mathbf{M}_i\mathbf{H}_{ij}\mathbf{M}_j\mathbb{1} \tag{6.11}$$

where $\mathbf{M}_{=}(-\mathbf{D}_i)^{-1}$, and $\mathbf{M}_j = (-\mathbf{D}_j)^{-1}$.

Theorem 6.1. *Let $((\pi_i, \mathbf{D}_i),\ (\pi_j, \mathbf{D}_j),\ \mathbf{H}_{ij})$ be a composition of PH_i and PH_j as defined in (3.1). If the PHD with representation $(\pi_j, \mathbf{D}_j)$ is transformed into canonical form (see Def. 2.11), then this representation results in the maximal first joint moment $\mu_{1,1}^*$, given that PHD $(\pi_i, \mathbf{D}_i)$ is in hyperexponential form.*

Theorem 6.1 states that series canonical form is the best suitable representation of PH_j for maximizing the first joint moment, given that PH_i is in hyperexponential form. Note that the transformation into a canonical representation can be performed using similarity transformations (6.3)-(6.4). Thus it is important to know which transformation parameter δ should be chosen. The result of Theorem 6.1 states that the parameter δ should be decreased to increase $\mu_{1,1}$. When δ is set to the minimum, i.e., to δ^-, then the parameter $\lambda_{1,2}$ is equal to $1 = \lambda_1$ which corresponds to series canonical form of $PH_j^{\delta^-}$.

Proof. Observe that first joint moment $\mu_{1,1}$ in Eq. 6.11 can be represented in terms of all possible paths in PH_i following all possible paths in PH_j. Let $a_{k,l}^{\delta}$ be the probability that APHD PH_j starts in the state k and l is the last state before absorption. It holds that

$$a_{1,1}^{\delta} = \pi^{\delta}(1-\lambda_{1,2}^{\delta}),\ a_{1,2}^{\delta} = \pi^{\delta}\lambda_{1,2}^{\delta},\ a_{2,2}^{\delta} = 1-\pi^{\delta} \text{ and } a_{2,1}^{\delta} = 0, \tag{6.12}$$

since PHD PH_j is acyclic, i.e. it is not possible to go to the state 1 if the process starts in state 2 [41]. We use $\nu_{k,l}$ originally defined in [41] which denotes the mean duration, if the process starts in state k and state l is the last state before absorption of PH_j. It holds that

$$\nu_{1,1} = \frac{1}{\lambda_1} = 1, \nu_{1,2} = 1 + \frac{1}{\lambda_2}, \text{ and } \nu_{2,2} = \frac{1}{\lambda_2}. \tag{6.13}$$

For transfer matrix $\mathbf{H}_{ij}$ it holds that $\mathbf{H}_{ij}\mathbb{1} = -\mathbf{D}_i\mathbb{1}$. Probability b_k originally defined in [41] gives the probability that PH_j starts in state 1 after k was the last state before absorption in PH_i. The transfer matrix $\mathbf{H}_{ij}$ can be written as

$$\mathbf{H}_{ij} = \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix} \begin{pmatrix} b_1 & 1-b_1 \\ b_2 & 1-b_2 \end{pmatrix} = \begin{pmatrix} b_1\mu_1 & (1-b_1)\mu_1 \\ b_2\mu_2 & (1-b_2)\mu_2 \end{pmatrix},$$

which satisfies the above condition $\mathbf{H}_{ij}\mathbb{1} = -\mathbf{D}_i\mathbb{1}$, since $0 \leq b_k \leq 1$. Furthermore, the condition $\mathbf{cMH}_{ij} = \pi$ should be satisfied to keep the initial distribution of the PH_j invariant. To ensure this and taking into account that $\mathbf{cM} = (\frac{c}{\mu_1} \ \frac{1-c}{\mu_2})$ the following has to hold

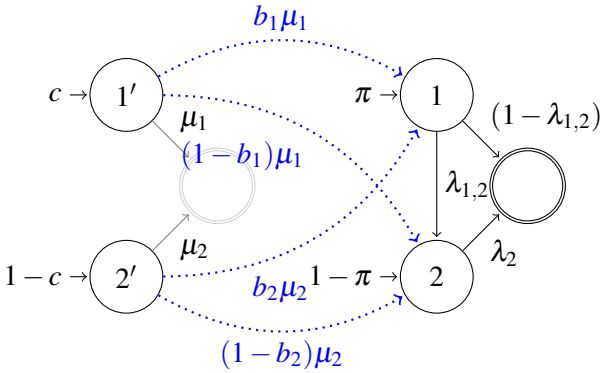
$$cb_1 + (1-c)b_2 = \pi^\delta, \text{ and } c(1-b_1) + (1-c)(1-b_2) = (1-\pi^\delta).$$

In the case that $b_1 = 0$, the probability $b_2 = \frac{\pi+\delta}{(1-c)}$. To ensure that $0 \leq b_2 \leq 1$, b_2 should be set to $\min(1, \frac{\pi+\delta}{(1-c)})$ in that case. E.g., if π is set to the maximum value, the term can become larger than 1 for $c \neq 1, c > 0$.

In the case that $b_1 = 1$, the probability $b_2 = \frac{\pi+\delta-c}{(1-c)}$ for $c \leq \pi$, $c \neq 1$, and thus $b_2 = \max(0, \frac{\pi+\delta-c}{(1-c)})$. The probability b_2 can be chosen from the interval

$$\left[\max(0, \frac{\pi+\delta-c}{(1-c)}), \min(1, \frac{\pi+\delta}{(1-c)})\right], \tag{6.14}$$

and the probabilty b_1 can be expressed in terms of b_2 as $b_1 = \frac{(\pi+\delta)-(1-c)b_2}{c}$.



(a) A graphical representation of the APHDs $(\mathbf{c}, \mathbf{D}_i)$, $(\pi, \mathbf{D}_j)$.

(b) Possible elementary series for both APHDs.

(1)	$< 1'\ 1 >$
(2)	$< 1'\ 1\ 2 >$
(3)	$< 1'\ 2 >$
(4)	$< 2'\ 1 >$
(5)	$< 2'\ 1\ 2 >$
(6)	$< 2'\ 2 >$

Figure 6.1.: Symbolic representation of two APHDs in composition for correlation fitting.

Now consider all possible paths in composition of two APHDs as visualized in Fig. 6.1, where $< k\,l \ldots >$ is a sequence of states along the acyclic path from an initial state of PH_i to the absorbing state of PH_j. The probability of each series is computed as the product of the transition rates along the corresponding path and the initial

probability of the entry state of the path. The mixture of mean durations of all series determines the first joint moment of the distribution of time till absorption for underlying process, where each series is weighted proportionally to its probability. We obtain the following results for the series listed in Fig. 6.1(b)

$$\begin{aligned}
b_1 c \tfrac{1}{\mu_1} a_{1,1}^{\delta} \nu_{1,1} \quad & \text{for the series } (1),\\
b_1 c \tfrac{1}{\mu_1} a_{1,2}^{\delta} \nu_{1,2} \quad & \text{for the series } (2),\\
(1-b_1) c \tfrac{1}{\mu_1} a_{2,2}^{\delta} \nu_{2,2} \quad & \text{for the series } (3),\\
b_2 (1-c) \tfrac{1}{\mu_2} a_{1,1}^{\delta} \nu_{1,1} \quad & \text{for the series } (4),\\
b_2 (1-c) \tfrac{1}{\mu_2} a_{1,2}^{\delta} \nu_{1,2} \quad & \text{for the series } (5),\\
(1-b_2)(1-c) \tfrac{1}{\mu_2} a_{2,2}^{\delta} \nu_{2,2} \quad & \text{for the series } (6).
\end{aligned}$$

Remark 1. *The first joint moment of PH_i and PH_j in composition can be maximized if the probability of starting in an on average longest phase of PH_j is maximized when the process PH_i escaped from the on average longest phase.*

Similarly, in the case that PH_i escaped from the shortest phase on average the probability of starting in an on average shortest phase of PH_j should be maximized to increase the first joint moment.

Using the notations for elementary series the first joint moment can be written as

$$\begin{aligned}
\mu_{1,1}(\delta) \quad &= b_1 c \mu_1^{-1} (a_{1,1}^{\delta} \nu_{1,1} + a_{1,2}^{\delta} \nu_{1,2}) / \pi^{\delta} + \\
&\quad b_2 (1-c) \mu_2^{-1} (a_{1,1}^{\delta} \nu_{1,1} + a_{1,2}^{\delta} \nu_{1,2}) / \pi^{\delta} + \\
&\quad (1-b_1) c \mu_1^{-1} \lambda_2^{-1} + (1-b_2)(1-c) \mu_2^{-1} \lambda_2^{-1}.
\end{aligned}$$

After substituting the values given in (6.12), (6.13), we obtain

$$\begin{aligned}
\mu_{1,1}(\delta) \quad &= b_1 c \mu_1^{-1} \left((\pi+\delta)(1-\lambda_{1,2}^{\delta}) + (\pi+\delta) \lambda_{1,2}^{\delta} (1+\lambda_2^{-1}) \right) / \pi^{\delta} + \\
&\quad b_2 (1-c) \mu_2^{-1} \left((\pi+\delta)(1-\lambda_{1,2}^{\delta}) + (\pi+\delta) \lambda_{1,2}^{\delta} (1+\lambda_2^{-1}) \right) / \pi^{\delta} + \\
&\quad (1-b_1) c \mu_1^{-1} \lambda_2^{-1} + (1-b_2)(1-c) \mu_2^{-1} \lambda_2^{-1}.
\end{aligned}$$

We denote the terms in the sum as x_1, x_2, x_3, and x_4. For the third term we obtain

$$\begin{aligned}
x_3 \quad &= \left(1 - \tfrac{(\pi+\delta)-(1-c)b_2}{c} \right) c \mu_1^{-1} \lambda_2^{-1} \\
&= \tfrac{c}{\mu_1 \lambda_2} - \tfrac{(\pi+\delta)-(1-c)b_2}{\mu_1 \lambda_2} \\
&= \tfrac{c-\pi-\delta}{\mu_1 \lambda_2} + \tfrac{b_2(1-c)}{\mu_1 \lambda_2}.
\end{aligned}$$

For the fourth term we obtain

$$\begin{aligned}
x_4 \quad &= (1-b_2)(1-c) \mu_2^{-1} \lambda_2^{-1} \\
&= \tfrac{(1-c)}{\mu_2 \lambda_2} - \tfrac{b_2(1-c)}{\mu_2 \lambda_2}.
\end{aligned}$$

The sum of the x_3 and x_4 results in

$$x_3 + x_4 = \frac{1}{\lambda_2} \left(\frac{\mu_1 - c(\mu_1 - \mu_2) - \mu_2(\pi+\delta)}{\mu_1 \mu_2} \right) + b_2 \left(\frac{(1-c)}{\lambda_2} \left[\frac{1}{\mu_1} - \frac{1}{\mu_2} \right] \right).$$

Now consider the term $(\pi+\delta)(1-\lambda_{1,2}^{\delta})+(\pi+\delta)\lambda_{1,2}^{\delta}(1+\lambda_2^{-1})$, which is denoted as x_5

$$\begin{aligned}
x_5 &= (\pi+\delta)\left(1-\lambda_{1,2}^{\delta}+\lambda_{1,2}^{\delta}(1+\lambda_2^{-1})\right)\\
&= (\pi+\delta)\left(1-\lambda_{1,2}^{\delta}+\lambda_{1,2}^{\delta}+\tfrac{\lambda_{1,2}^{\delta}}{\lambda_2}\right)\\
&= (\pi+\delta)\left(1+\tfrac{(\pi\lambda_{1,2}-\delta(\lambda_2-1))}{(\pi+\delta)\lambda_2}\right)\\
&= \pi+\delta+\tfrac{(\pi+\delta)(\pi\lambda_{1,2}-\delta(\lambda_2-1))}{(\pi+\delta)\lambda_2}\\
&= \pi+\delta+\tfrac{(\pi\lambda_{1,2}-\delta(\lambda_2-1))}{\lambda_2}\\
&= \tfrac{\pi\lambda_{1,2}-\delta\lambda_2+\delta\lambda_2+\delta+\pi\lambda_2}{\lambda_2}\\
&= \tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2}.
\end{aligned}$$

Using $b_1=\frac{(\pi+\delta)-(1-c)b_2}{c}$, the sum of x_1 and x_2 can be rewritten as

$$\begin{aligned}
x_1+x_2 &= \tfrac{b_1cx_5}{(\pi+\delta)\mu_1}+\tfrac{b_2(1-c)x_5}{(\pi+\delta)\mu_2}\\
&= \tfrac{((\pi+\delta)-(1-c)b_2)x_5}{(\pi+\delta)\mu_1}+\tfrac{b_2(1-c)x_5}{(\pi+\delta)\mu_2}\\
&= \tfrac{(\pi+\delta)x_5}{(\pi+\delta)\mu_1}-\tfrac{b_2(1-c)x_5}{(\pi+\delta)\mu_1}+\tfrac{b_2(1-c)x_5}{(\pi+\delta)\mu_2}\\
&= \tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2\mu_1}-b_2\left[\tfrac{(1-c)x_5}{(\pi+\delta)}\left(\tfrac{1}{\mu_1}-\tfrac{1}{\mu_2}\right)\right]\\
&= \tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2\mu_1}-b_2\left[\tfrac{(1-c)x_5(\mu_2-\mu_1)}{(\pi+\delta)\mu_1\mu_2}\right]\\
&= \tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2\mu_1}-b_2\left[\tfrac{1}{(\pi+\delta)}\tfrac{(\pi(\lambda_{1,2}+\lambda_2)+\delta)(1-c)(\mu_2-\mu_1)}{\lambda_2\mu_1\mu_2}\right]
\end{aligned}$$

We denote the term $(1-c)(\mu_2-\mu_1)$ as y_1 and resolve the subtrahend

$$\begin{aligned}
b_2\left[\tfrac{1}{(\pi+\delta)}\tfrac{(\pi(\lambda_{1,2}+\lambda_2)+\delta)(1-c)(\mu_2-\mu_1)}{\lambda_2\mu_1\mu_2}\right] &=\\
b_2\left[\tfrac{1}{(\pi+\delta)}\tfrac{(\pi(\lambda_{1,2}+\lambda_2)y_1+\delta y_1}{\lambda_2\mu_1\mu_2}\right] &=\\
b_2\left[\tfrac{1}{(\pi+\delta)}\tfrac{\pi(\lambda_{1,2}+\lambda_2)y_1}{\lambda_2\mu_1\mu_2}+\tfrac{\delta}{(\pi+\delta)}\tfrac{y_1}{\lambda_2\mu_1\mu_2}\right]
\end{aligned}$$

Now the expression for the first joint moment results in

$$\begin{aligned}
\mu_{1,1}(\delta) &= \tfrac{1}{\lambda_2}\left(\tfrac{\mu_1-c(\mu_1-\mu_2)-\mu_2(\pi+\delta)}{\mu_1\mu_2}\right)+b_2\left(\tfrac{(1-c)}{\lambda_2}\left[\tfrac{1}{\mu_1}-\tfrac{1}{\mu_2}\right]\right)+\\
&\quad \tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2\mu_1}-b_2\left[\tfrac{1}{(\pi+\delta)}\tfrac{\pi(\lambda_{1,2}+\lambda_2)y_1}{\lambda_2\mu_1\mu_2}+\tfrac{\delta}{(\pi+\delta)}\tfrac{y_1}{\lambda_2\mu_1\mu_2}\right]\\
&= \tfrac{1}{\lambda_2}\left(\tfrac{\mu_1-c(\mu_1-\mu_2)-\mu_2(\pi+\delta)}{\mu_1\mu_2}\right)+\tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2\mu_1}+\\
&\quad b_2\left[\tfrac{y_1}{\lambda_2\mu_1\mu_2}-\tfrac{1}{(\pi+\delta)}\tfrac{\pi(\lambda_{1,2}+\lambda_2)y_1}{\lambda_2\mu_1\mu_2}-\tfrac{\delta}{(\pi+\delta)}\tfrac{y_1}{\lambda_2\mu_1\mu_2}\right].
\end{aligned}$$

The first two terms can be simplified to

$$\begin{aligned}
\tfrac{1}{\lambda_2}\left(\tfrac{\mu_1-c(\mu_1-\mu_2)-\mu_2(\pi+\delta)}{\mu_1\mu_2}\right)+\tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2\mu_1} &= \tfrac{\mu_1-c(\mu_1-\mu_2)-\mu_2(\pi+\delta)+\pi(\lambda_{1,2}+\lambda_2)\mu_2+\delta\mu_2}{\lambda_2\mu_1\mu_2}\\
&= \tfrac{\mu_1-c(\mu_1-\mu_2)-\mu_2\pi-\mu_2\delta+\pi(\lambda_{1,2}+\lambda_2)\mu_2+\delta\mu_2}{\lambda_2\mu_1\mu_2}\\
&= \tfrac{\mu_1-c(\mu_1-\mu_2)-\mu_2\pi+\pi(\lambda_{1,2}+\lambda_2)\mu_2}{\lambda_2\mu_1\mu_2},
\end{aligned}$$

such that the numerator is denoted by y_2. Then the expression for the first joint moment

can be rewritten as

$$\begin{aligned}
\mu_{1,1}(\delta) &= \frac{y_2+b_2(y_1-\frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)y_1-\frac{\delta}{(\pi+\delta)}y_1)}{\lambda_2\mu_1\mu_2} \\
&= \frac{y_2+y_1b_2(1-\frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)-\frac{\delta}{(\pi+\delta)})}{\lambda_2\mu_1\mu_2}, \\
y_1 &= (1-c)(\mu_2-\mu_1), \\
y_2 &= \mu_1-c(\mu_1-\mu_2)-\mu_2\pi+\pi(\lambda_{1,2}+\lambda_2)\mu_2.
\end{aligned} \tag{6.15}$$

Note that the probability b_2 can be chosen from the interval $\left[\max(0,\frac{\pi+\delta-c}{(1-c)}),\min(1,\frac{\pi+\delta}{(1-c)})\right]$. In the case that $b_2=0$, the APHD PH_j has no flexibily in maximizing the first joint moment. The same holds if $c=1$ such that the value of $y_1=0$. Furthermore, if we consider the case with $\lambda_2=\lambda_1=1$ then the maximal achievable correlation is always equal to 0, i.e. the APHD PH_j represents an exponential distribution with rate 1. Thus, this case is not relevant such that $\lambda_2\neq\lambda_1$ can be assumed. It also holds that in cases where $c=1$, $(1-c)=1$, $\pi=1$ or $(1-\pi)=1$ the maximal achievable correlation is 0, because one of the given PHDs is an exponential distribution.

The terms y_1, y_2, and $\lambda_1\mu_1\mu_2$ are positive ($y_1>0$ for $c\neq 1$) and do not depend on varying parameters δ or b_2. The following holds for the term appearing in the expression for $\mu_{1,1}(\delta)$

$$\left(1-\frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)-\frac{\delta}{(\pi+\delta)}\right)=\frac{\pi+\delta-\pi(\lambda_{1,2}+\lambda_2)-\delta}{(\pi+\delta)}<0.$$

The denominator is always positive. For minimal $\delta=-\frac{\pi(1-\lambda_{1,2})}{\lambda_2}$ we obtain the term

$$\pi+\delta=\pi-\frac{\pi(1-\lambda_{1,2})}{\lambda_2}=\frac{\pi(\lambda_2-1+\lambda_{1,2})}{\lambda_2},$$

which is positive because $\lambda_2>0$, $\pi>0$, and $(\lambda_2-1+\lambda_{1,2})>0$, since we assume $\lambda_2>1$, and $\lambda_{1,2}\leq 1$. Furthermore, the numerator $\pi-\pi(\lambda_{1,2}+\lambda_2)<0$, since $(\lambda_{1,2}+\lambda_2)>1$ because $\lambda_2>1$.

Let $y_3(\delta)=y_1\left(1-\frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)-\frac{\delta}{(\pi+\delta)}\right)$, Since $y_3(\delta)<0$, the probability b_2 should be as small as possible to maximize the first joint moment $\mu_{1,1}(\delta)$.

We now consider the minimal b_2 for varying δ, i.e., $b_2(\delta)=\frac{\pi+\delta-c}{(1-c)}>0$, $c\neq 1$, $c<\pi$, and where $b_1=1$ since b_2 is fixed. The derivative with respect to δ equals

$$\frac{\partial db_2(\delta)}{\partial\delta}=\frac{1}{(1-c)}>0.$$

The term depending on δ in the numerator of $\mu_{1,1}(\delta)$ is $y_3(\delta)$. Its derivative equals

$$\begin{aligned}
\tfrac{\partial y_3(\delta)}{\partial\delta} &= \tfrac{\partial}{\partial\delta}\left(y_1-\tfrac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)y_1-\tfrac{\delta}{(\pi+\delta)}y_1\right) \\
&= \tfrac{\pi(\lambda_{1,2}+\lambda_2)y_1}{(\pi+\delta)^2}-\tfrac{y_1\pi}{(\pi+\delta)^2} \\
&= \tfrac{\pi y_1(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\geq 0 \\
\text{with}\quad & \tfrac{\partial y_1}{\partial\delta}=\tfrac{\partial}{\partial\delta}\left((1-c)(\mu_2-\mu_1)\right)=0, \\
& \tfrac{\partial\left(\frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)y_1\right)}{\partial\delta}=\tfrac{\partial}{\partial\delta}\left(\tfrac{\pi(\lambda_{1,2}+\lambda_2)y_1}{\pi+\delta}\right)=-\tfrac{\pi(\lambda_{1,2}+\lambda_2)y_1}{(\pi+\delta)^2}, \\
& \tfrac{\partial}{\partial\delta}\left(\tfrac{\delta y_1}{\pi+\delta}\right)=\tfrac{y_1(\pi+\delta)-\delta y_1}{(\pi+\delta)^2}=\tfrac{y_1\pi}{(\pi+\delta)^2}.
\end{aligned}$$

Then

$$
\begin{aligned}
\frac{\partial \mu_{1,1}(\delta)}{\partial \delta} &= \frac{\partial}{\partial \delta}\left(\frac{y_2+b_2(\delta)y_3(\delta)}{\lambda_1\mu_1\mu_2}\right) \\
&= \frac{(\mu_2-\mu_1)(-c(\pi\lambda_{1,2}+\pi\lambda_2-\pi))}{(\pi+\delta)^2(\lambda_2\mu_1\mu_2)^2} \\
&= \frac{(\mu_2-\mu_1)(-c\pi(\lambda_{1,2}+\lambda_2-1))}{(\pi+\delta)^2(\lambda_2\mu_1\mu_2)^2} \\
\text{with} \quad & \frac{\partial}{\partial \delta}(y_2+b_2(\delta)y_3(\delta)) = \frac{\partial}{\partial \delta}y_2 + \frac{\partial}{\partial \delta}(b_2(\delta)y_3(\delta)) \\
&= 0 + \frac{1}{(1-c)}y_3(\delta) + b_2(\delta)\left(\frac{\pi y_1(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= \frac{1}{(1-c)}\left(y_1 - \frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)y_1 - \frac{\delta}{(\pi+\delta)}y_1\right) + \left(\frac{\pi+\delta-c}{(1-c)}\right)\left(\frac{\pi y_1(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= \frac{1}{(1-c)}y_1\left(1 - \frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2) - \frac{\delta}{(\pi+\delta)}\right) + \left(\frac{\pi+\delta-c}{(1-c)}\right)\left(\frac{y_1\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= \frac{1}{(1-c)}(1-c)(\mu_2-\mu_1)\left(1 - \frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2) - \frac{\delta}{(\pi+\delta)}\right) + \\
&\left(\frac{\pi+\delta-c}{(1-c)}\right)\left(\frac{(1-c)(\mu_2-\mu_1)\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= (\mu_2-\mu_1)\left(1 - \frac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2) - \frac{\delta}{(\pi+\delta)}\right) + \\
&(\pi+\delta-c)\left(\frac{(\mu_2-\mu_1)\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= \frac{(\pi+\delta)(\mu_2-\mu_1)-(\mu_2-\mu_1)\pi(\lambda_{1,2}+\lambda_2)-(\mu_2-\mu_1)\delta}{(\pi+\delta)} + (\pi+\delta-c)\left(\frac{(\mu_2-\mu_1)\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= \frac{(\mu_2-\mu_1)(\pi+\delta-\pi(\lambda_{1,2}+\lambda_2)-\delta)}{(\pi+\delta)} + (\pi+\delta-c)\left(\frac{(\mu_2-\mu_1)\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= \frac{(\mu_2-\mu_1)(\pi-\pi(\lambda_{1,2}+\lambda_2))}{(\pi+\delta)} + \left(\frac{(\pi+\delta-c)(\mu_2-\mu_1)\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2}\right) \\
&= \frac{(\pi+\delta)(\mu_2-\mu_1)(\pi-\pi(\lambda_{1,2}+\lambda_2))+(\pi+\delta-c)(\mu_2-\mu_1)\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2} \\
&= \frac{(\mu_2-\mu_1)((\pi+\delta)(\pi-\pi\lambda_{1,2}-\pi\lambda_2)+(\pi+\delta)(\pi\lambda_{1,2}+\pi\lambda_2-\pi)-c(\pi\lambda_{1,2}+\pi\lambda_2-\pi))}{(\pi+\delta)^2} \\
&= \frac{(\mu_2-\mu_1)(-c(\pi\lambda_{1,2}+\pi\lambda_2-\pi))}{(\pi+\delta)^2}
\end{aligned}
$$

The value of the derivative $\frac{(\mu_2-\mu_1)(-c\pi(\lambda_{1,2}+\lambda_2-1))}{(\pi+\delta)^2(\lambda_2\mu_1\mu_2)^2}$ is negative, as can be shown. The denominator is always positive such that we only have to consider the expression in the numerator $(\mu_2-\mu_1)(-c\pi(\lambda_{1,2}+\lambda_2-1))$. Since $\lambda_2 > \lambda_1$, and $\lambda_1 = 1$ holds per assumption, the expression $\pi(\lambda_{1,2}+\lambda_2-1)$ is always positive. The case with $\mu_2 = \mu_1$ is similar to the case with $\lambda_2 = \lambda_1$, such that the first APHD represents the exponential distribution. In that case no correlation can be modeled and the case with $\mu_2 = \mu_1$ is not interesting. Thus we consider only the case with $\mu_2 > \mu_1$ such that the term $(\mu_2-\mu_1)$ is positive and the value of the derivative is negative such that δ should be decreased to increase $\mu_{1,1}(\delta)$. We set $\delta = -\frac{\pi(1-\lambda_{1,2})}{\lambda_2}$, then

$$
\begin{aligned}
\lambda_{1,2} &= \frac{\pi\lambda_{1,2}}{(\pi+\delta)} - \frac{(\lambda_2-1)\delta}{(\pi+\delta)} \\
&= \frac{\pi\lambda_{1,2}}{\pi-\frac{\pi(1-\lambda_{1,2})}{\lambda_2}} - \frac{(\lambda_2-1)(-\frac{\pi(1-\lambda_{1,2})}{\lambda_2})}{\pi-\frac{\pi(1-\lambda_{1,2})}{\lambda_2}} \\
&= \frac{\pi\lambda_{1,2}\lambda_2}{\pi(\lambda_2-1+\lambda_{1,2})} - \frac{(\lambda_2-1)(-\pi+\pi\lambda_{1,2})}{\pi(\lambda_2-1+\lambda_{1,2})} \\
&= \frac{\pi\lambda_{1,2}\lambda_2}{\pi(\lambda_2-1+\lambda_{1,2})} - \frac{(-\lambda_2\pi+\pi\lambda_{1,2}\lambda_2+\pi-\pi\lambda_{1,2})}{\pi(\lambda_2-1+\lambda_{1,2})} \\
&= \frac{\pi\lambda_{1,2}\lambda_2}{\pi(\lambda_2-1+\lambda_{1,2})} - \frac{\pi(-\lambda_2+\lambda_{1,2}\lambda_2+1-\lambda_{1,2})}{\pi(\lambda_2-1+\lambda_{1,2})} \\
&= \frac{\lambda_{1,2}\lambda_2}{(\lambda_2-1+\lambda_{1,2})} - \frac{(-\lambda_2+\lambda_{1,2}\lambda_2+1-\lambda_{1,2})}{(\lambda_2-1+\lambda_{1,2})} \\
&= \frac{\lambda_{1,2}\lambda_2+\lambda_2-\lambda_{1,2}\lambda_2-1+\lambda_{1,2}}{(\lambda_2-1+\lambda_{1,2})} = \frac{\lambda_2-1+\lambda_{1,2}}{(\lambda_2-1+\lambda_{1,2})} = 1 = \lambda_1,
\end{aligned}
$$

which implies the canonical representation of the successor PHD PH_j. Note that if δ is set to $-\pi$, then the value of $\lambda_{1,2}^{\delta}$ becomes infinite. □

***Example* 6.1.** *We consider following two APHDs in composition. PHD PH_i has the representation*

$$\mathbf{c} = (0.01, 0.99)\,,\ \mathbf{D}_i = \begin{pmatrix} -0.0196 & 0 \\ 0 & -0.22 \end{pmatrix},$$

and PHD PH_j has the representation

$$\pi = (0.99, 0.01)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0 \\ 0 & -5.0 \end{pmatrix}.$$

It results that

$$\mu_{1,1} = 4.9738,\ \rho = 4.5205e-04,\ \mathbf{H}_{ij} = \begin{pmatrix} 0.0196 & 0 \\ 0.21778 & 0.00222 \end{pmatrix}.$$

Furthermore we obtain $\delta^- = -0.198$, and $\delta^+ = 0$. Setting δ^* to δ^-

$$\pi^{\delta^*} = (0.792, 0.208)\,,\ (\mathbf{D}_j)^{\delta^*} = \begin{pmatrix} -1.0 & 1.0 \\ 0 & -5.0 \end{pmatrix},$$

$$\textit{with } \mu_{1,1}^{\delta^*} = 5.0668,\ \rho^{\delta^*} = 0.011753,\ \mathbf{H}_{ij}^{\delta^*} = \begin{pmatrix} 0.0196 & 0 \\ 0.17378 & 0.04622 \end{pmatrix}.$$

***Example* 6.2.** *Now consider two APHDs in composition with representations*

$$\mathbf{c} = (0.01, 0.99)\,,\ \mathbf{D}_i = \begin{pmatrix} -1.1529 & 0 \\ 0 & -12.941 \end{pmatrix},$$

and

$$\pi = (0.01, 0.99)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0 \\ 0 & -5.0 \end{pmatrix},$$

$$\textit{with } \mu_{1,1} = 0.023974,\ \rho = 0.18892,\ \mathbf{H}_{ij} = \begin{pmatrix} 1.15290 & 0 \\ 0 & 12.941 \end{pmatrix}.$$

We obtain $\delta^- = -0.002$, and $\delta^+ = 0$. Setting δ^* to δ^-

$$\pi^{\delta^*} = (0.008, 0.992)\,,\ (\mathbf{D}_j)^{\delta^*} = \begin{pmatrix} -1.0 & 1.0 \\ 0 & -5.0 \end{pmatrix},$$

$$\textit{with } \mu_{1,1}^{\delta^*} = 0.023974,\ \rho^{\delta^*} = 0.18892,\ \mathbf{H}_{ij}^{\delta^*} = \begin{pmatrix} 0.92232 & 0.23058 \\ 0 & 12.94100 \end{pmatrix}.$$

***Example* 6.3.** *Consider the following APHDs in composition*

$$\mathbf{c} = (0.4, 0.6)\,,\ \mathbf{D}_i = \begin{pmatrix} -1.1529 & 0 \\ 0 & -12.941 \end{pmatrix},$$

and

$$\pi = (0.45, 0.55)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0.44444 \\ 0 & -5.0 \end{pmatrix},$$

$$\text{with } \mu_{1,1} = 0.39050,\ \rho = 0.27798,\ \mathbf{H}_{ij} = \begin{pmatrix} 1.1529 & 0 \\ 1.07842 & 11.86258 \end{pmatrix}.$$

We obtain the following δ bounds $\delta^- = -0.05$, and $\delta^+ = 0.05$, such that setting δ^* to δ^- the transformation computes

$$\pi^{\delta^*} = (0.4, 0.6),\ (\mathbf{D}_j)^{\delta^*} = \begin{pmatrix} -1.0 & 1.0 \\ 0 & -5.0 \end{pmatrix},$$

$$\text{with } \mu_{1,1}^{\delta^*} = 0.42561,\ \rho^{\delta^*} = 0.34115,\ \mathbf{H}_{ij}^{\delta^*} = \begin{pmatrix} 1.1529 & 0 \\ 0 & 12.941 \end{pmatrix}.$$

6.2.2. The second case

In the second case the representation of PHD PH_i in the composition is fixed to series canonical form. Again we are interested in the best suitable representation of PH_j in composition in order to maximize the first joint moment $\mu_{1,1}$. Let PHD PH_i be acyclic, its canonical representation is given by

$$\mathbf{c} = (c, 1-c),\ \mathbf{D}_i = \begin{pmatrix} -1 & 1 \\ 0 & -\mu_2 \end{pmatrix},$$

where $-\mathbf{D}_i(1,1) = \mu_1$ is the transition rate of state 1, and $-\mathbf{D}_i(2,2) = \mu_2$ is the transition rate of state 2. The transition rate μ_1 has been scaled to 1, and we assume $\mu_2 \geq \mu_1$, i.e. $\mu_2 \geq 1$ (if it is not the case, we apply the transformation from [41]).

The second APHD PH_j has the representation given in Eq. 6.7 with assumptions stated in Eq. 6.8.

Since the representation of PH_i is in series canonical form, the corresponding Markov process has only one exit state. Thus no flexibility can be achieved applying the transformation to the successor PHD PH_j in composition. The result from theorem 6.2 is not surprising. It states that there is no suitable representation of PH_j for maximizing the first joint moment, if the representation of PH_i is in series canonical form.

Theorem 6.2. *Let $((\pi_i, \mathbf{D}_i), (\pi_j, \mathbf{D}_j), \mathbf{H}_{ij})$ be a composition of PH_i and PH_j as defined in (3.1). The PHD PH_i in series canonical representation enables no flexibily in maximizing the first joint moment. In that case the representation of PH_j has no influence on the range of achievable correlation, such that the transformation parameter δ can be chosen arbitrary.*

The formal proof of the above theorem can be found in C.1.

***Example* 6.4.** *Consider the two APHDs in composition*

$$\mathbf{c} = (0.36436, 0.63564),\ \mathbf{D}_i = \begin{pmatrix} -1 & 1 \\ 0.00000 & -11.22474 \end{pmatrix},$$

and

$$\pi = (0.45, 0.55),\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0.44 \\ 0 & -5.0 \end{pmatrix},$$

$$\text{with } \mu_{1,1} = 0.27189,\ \rho = 0,\ \mathbf{H}_{ij} = \begin{pmatrix} 0 & 0 \\ 5.05113 & 6.17361 \end{pmatrix}.$$

We obtain the following δ bounds $\delta^- = -0.0504$, and $\delta^+ = 0.0495$. Setting δ to δ^+ the transformation method generates an hyperexponential representation of PH_j

$$\pi^{\delta^+} = (0.4995, 0.5005)\,,\ (\mathbf{D}_j)^{\delta^+} = \begin{pmatrix} -1.0 & 0 \\ 0 & -5.0 \end{pmatrix},$$

$$\text{with } \mu_{1,1}^{\delta^+} = 0.27189,\ \rho^{\delta^+} = 0,\ \mathbf{H}_{ij}^{\delta^+} = \begin{pmatrix} 0 & 0 \\ 5.60676 & 5.61798 \end{pmatrix}.$$

Setting δ to δ^- the transformation generates canonical representation of PH_j

$$\pi^{\delta^-} = (0.3996, 0.6004)\,,\ (\mathbf{D}_j)^{\delta^-} = \begin{pmatrix} -1.0 & 1.0 \\ 0 & -5.0 \end{pmatrix},$$

$$\text{with } \mu_{1,1}^{\delta^-} = 0.27189,\ \rho^{\delta^*} = 0,\ \mathbf{H}_{ij}^{\delta^-} = \begin{pmatrix} 0 & 0 \\ 4.48541 & 6.73933 \end{pmatrix}.$$

***Example* 6.5.** *Now we have the two following APHDs in composition*

$$\mathbf{c} = (0.094118, 0.905882)\,,\ \mathbf{D}_i = \begin{pmatrix} -1 & 1 \\ 0.00000 & -8.5 \end{pmatrix},$$

and

$$\pi = (0.7, 0.3)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0.8 \\ 0 & -1.5 \end{pmatrix},$$

$$\text{with } \mu_{1,1} = 0.26965,\ \rho = 0,\ \mathbf{H}_{ij} = \begin{pmatrix} 0 & 0 \\ 5.95 & 2.55 \end{pmatrix}.$$

We obtain the following δ bounds $\delta^- = -0.093333$, and $\delta^+ = 0.3$. Setting δ to δ^+ the we obtain the following representation of PH_j

$$\pi^{\delta^+} = (1, 0)\,,\ (\mathbf{D}_j)^{\delta^+} = \begin{pmatrix} -1.0 & 0.41 \\ 0 & -1.5 \end{pmatrix},$$

$$\text{with } \mu_{1,1}^{\delta^+} = 0.26965,\ \rho^{\delta^+} = 0,\ \mathbf{H}_{ij}^{\delta^+} = \begin{pmatrix} 0 & 0 \\ 8.5 & 0 \end{pmatrix}.$$

Setting δ to δ^- the transformation method generates canonical representation of PH_j

$$\pi^{\delta^-} = (0.60667, 0.39333)\,,\ (\mathbf{D}_j)^{\delta^-} = \begin{pmatrix} -1.0 & 1 \\ 0 & -1.5 \end{pmatrix},$$

$$\text{with } \mu_{1,1}^{\delta^-} = 0.26965,\ \rho^{\delta^-} = 0,\ \mathbf{H}_{ij}^{\delta^-} = \begin{pmatrix} 0 & 0 \\ 5.15667 & 3.34333 \end{pmatrix}.$$

If we set $\delta = 0.1$ we obtain

$$\pi^{\delta} = (0.80000, 0.20000)\,,\ (\mathbf{D}_j)^{\delta} = \begin{pmatrix} -1.0 & 0.63750 \\ 0 & -1.5 \end{pmatrix},$$

$$\text{with } \mu_{1,1}^{\delta} = 0.26965,\ \rho^{\delta} = 0,\ \mathbf{H}_{ij}^{\delta} = \begin{pmatrix} 0 & 0 \\ 6.8 & 1.7 \end{pmatrix}.$$

6.2.3. The third case

Now we are interesting in the optimal representation of PHD PH_i in composition in order to maximize the first joint moment, when the representation of PHD PH_j is fixed. The third case is given when PH_j is in hyperexponential representation

$$\pi = (\pi, 1-\pi)\,,\ \mathbf{D}_j = \begin{pmatrix} -1 & 0 \\ 0 & -\lambda_2 \end{pmatrix},$$

where $-\mathbf{D}_j(1,1) = \lambda_1$ is the transition rate of state 1, $-\mathbf{D}_j(2,2) = \lambda_2$ is the transition rate of state 2, and $\mathbf{D}_j(1,2) = 0$. We assume $\lambda_2 > 1$, such that λ_1 has been scaled to 1, and $\lambda_2 \neq \lambda_1$.

The PHD PH_i is acyclic and has the representation

$$\mathbf{c} = (c, 1-c)\,,\ \mathbf{D}_i = \begin{pmatrix} -\mu_1 & \mu_{1,2} \\ 0 & -\mu_2 \end{pmatrix} = \begin{pmatrix} -1 & \mu_{1,2} \\ 0 & -\mu_2 \end{pmatrix}$$

where $-\mathbf{D}_i(1,1) = \mu_1$ is the transition rate of state 1, $-\mathbf{D}_i(2,2) = \mu_2$ is the transition rate of state 2, and $\mathbf{D}_i(1,2) = \mu_{1,2}$ is the transition rate from state 1 to state 2. There are the following inequalities

$$0 \leq c \leq 1, \mu_{1,2} \leq 1, \mu_2 \geq 1,$$

such that $\mu_2 \geq \mu_1$, and μ_1 has been also scaled to 1.

The transformation of APHD with representation $(\mathbf{c}, \mathbf{D}_i)$ in the 2-state case (6.7) is according to

$$\mathbf{c}^{\delta} = \left(c^{\delta}, 1-c^{\delta}\right) = (c+\delta, 1-c-\delta)\,,\ \mu_{1,2}^{\delta} = \frac{c\mu_{1,2}}{c+\delta} - \frac{(\mu_2-1)\delta}{c+\delta} \tag{6.16}$$

where δ-bounds are given by the interval

$$\delta \in \left[\max\left(-c, -\frac{c(1-\mu_{1,2})}{\mu_2}\right), \min\left(1-c, \frac{c\mu_{1,2}}{\mu_2 - 1}\right)\right]. \tag{6.17}$$

Theorem 6.3. *Let $((\pi_i, \mathbf{D}_i), (\pi_j, \mathbf{D}_j), \mathbf{H}_{ij})$ be a composition of PH_i and PH_j as defined in (3.1). If the representation of PH_i can be transformed into hyperexponential form, then this representation results in the maximal first joint moment $\mu_{1,1}^*$, given that the representation PHD PH_j is in hyperexponential form.*

Theorem 6.3 states that the best suitable representation of PH_i for maximizing the first joint moment is the hyperexponential form, when PH_j of composition is in hyperexponential form. The proof given in C.1 provides insights which transformation parameter δ should be chosen to achieve the optimal representation. The results show that the transformation parameter δ should be increased to increase $\mu_{1,1}$. When δ is set to the maximum, i.e., to δ^+, then the parameter $\mu_{1,2}$ is equal to zero which corresponds to the hyperexponential representation of the transformed $PH_i^{\delta^+}$.

***Example* 6.6.** *We consider the following APHDs in composition. PH_j has the representation*

$$\pi = (0.4, 0.6)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0 \\ 0 & -12.0 \end{pmatrix}.$$

and PH_i has the representation

$$\mathbf{c} = (0.4, 0.6)\,,\ \mathbf{D}_i = \begin{pmatrix} -1 & 0.4 \\ 0 & -3 \end{pmatrix},$$

The maximal first joint moment for these APHDs is

$$\mu_{1,1} = 0.35421,\ \rho = 0.096044,\ \mathbf{H}_{ij} = \begin{pmatrix} 0.6 & 0 \\ 0.63158 & 2.36842 \end{pmatrix}.$$

Furthermore we obtain $\delta^- = -0.08$, and $\delta^+ = 0.08$. Setting δ^* to δ^+

$$\mathbf{c}^{\delta^*} = (0.48, 0.52)\,,\ (\mathbf{D}_i)^{\delta^*} = \begin{pmatrix} -1.0 & 0 \\ 0 & -3.0 \end{pmatrix},$$

$$\textit{with } \mu_{1,1}^{\delta^*} = 0.42111,\ \rho^{\delta^*} = 0.20276,\ \mathbf{H}_{ij}^{\delta^*} = \begin{pmatrix} 0.83333 & 0.16667 \\ 0 & 3 \end{pmatrix}.$$

***Example* 6.7.** *Now let PH_j of composition*

$$\pi = (0.8, 0.2)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 0 \\ 0 & -5.0 \end{pmatrix}.$$

and PH_i has the representation

$$\mathbf{c} = (0.7, 0.3)\,,\ \mathbf{D}_i = \begin{pmatrix} -1 & 0.8 \\ 0 & -1.5 \end{pmatrix},$$

The maximal first joint moment for the APHDs

$$\mu_{1,1} = 1.0981,\ \rho = 0.026281,\ \mathbf{H}_{ij} = \begin{pmatrix} 0 & 0.2 \\ 1.39535 & 0.10465 \end{pmatrix}.$$

Furthermore we obtain $\delta^- = -0.093333$, and $\delta^+ = 0.3$. Here the PH_i cannot be transformed into hyperexponential representation. Setting δ^* to δ^+ the first joint moment can be maximized. Then the representation of $PH_i^{\delta^*}$ is

$$\mathbf{c}^{\delta^*} = (1, 0)\,,\ (\mathbf{D}_i)^{\delta^*} = \begin{pmatrix} -1.0 & 0.41 \\ 0 & -1.5 \end{pmatrix},$$

$$\textit{with } \mu_{1,1}^{\delta^*} = 1.1133,\ \rho^{\delta^*} = 0.040360,\ \mathbf{H}_{ij}^{\delta^*} = \begin{pmatrix} 0.39 & 0.2 \\ 1.5 & 0 \end{pmatrix}.$$

6.2.4. The fourth case

In the last case we fix the representation of PH_j in composition to series canonical form 2.19(b). We examine what is the optimal representation of PHD PH_i in order to maximize the first joint moment of the composition containing PH_i and PH_j.

Let PHD PH_j in composition be given by

$$\pi = (\pi, 1-\pi)\,,\ \mathbf{D}_j = \begin{pmatrix} -1 & 1 \\ 0 & -\lambda_2 \end{pmatrix},$$

where $-\mathbf{D}_j(1,1) = \lambda_1$ is the transition rate of state 1, $-\mathbf{D}_j(2,2) = \lambda_2$ is the transition rate of state 2, and $\mathbf{D}_j(1,2) = 1$. Here we assume $\lambda_2 > 1$, such that λ_1 has been scaled to 1, and $\lambda_2 \neq \lambda_1$, $\lambda_{1,2} = \lambda_1$.

The representation of PH_i in composition is

$$\mathbf{c} = (c, 1-c)\,,\ \mathbf{D}_i = \begin{pmatrix} -\mu_1 & \mu_{1,2} \\ 0 & -\mu_2 \end{pmatrix} = \begin{pmatrix} -1 & \mu_{1,2} \\ 0 & -\mu_2 \end{pmatrix},$$

where $-\mathbf{D}_i(1,1) = \mu_1$ is the transition rate of state 1, $-\mathbf{D}_i(2,2) = \mu_2$ is the transition rate of state 2, and $\mathbf{D}_i(1,2) = \mu_{1,2}$ is the transition rate from state 1 to state 2. There are the following inequalities

$$0 \leq c \leq 1, \mu_{1,2} \leq 1, \mu_2 \geq 1,$$

such that $\mu_2 \geq \mu_1$, and μ_1 has been scaled to 1. The transformation is performed according to steps (6.16)-(6.17).

Theorem 6.4. *Let $((\pi_i, \mathbf{D}_i),\ (\pi_j, \mathbf{D}_j),\ \mathbf{H}_{ij})$ be a composition of PHDs PH_i and PH_j as defined in (3.1). If the representation of PH_i can be transformed into hyperexponential form, then this representation results in the maximal first joint moment $\mu^*_{1,1}$, given that PHD PH_j is in series canonical form.*

Theorem 6.4 states that the best suitable representation of PH_i for maximizing the first joint moment is hyperexponential form such that we obtain the same result as stated in Theorem 6.3. When PHD PH_j is in series canonical form the transformation parameter δ should be increased to increase $\mu^*_{1,1}$. setting δ to its maximum value, parameter $\mu_{1,2}$ becomes 0 which corresponds to hyperexponential form of transformed $PH_i^{\delta^+}$. The proof is given in C.1.

***Example* 6.8.** *We consider following PHDs in composition. PHD PH_j in series canonical form*

$$\pi = (0.4, 0.6)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 1.0 \\ 0 & -8.5 \end{pmatrix}.$$

and PH_i has the representation

$$\mathbf{c} = (0.4, 0.6)\,,\ \mathbf{D}_i = \begin{pmatrix} -1 & 0.4 \\ 0 & -3 \end{pmatrix},$$

The maximal first joint moment for PHDs in composition is

$$\mu_{1,1} = 0.25015,\ \rho = 0.10086,\ \mathbf{H}_{ij} = \begin{pmatrix} 0.6 & 0 \\ 0.63158 & 2.36842 \end{pmatrix}.$$

Furthermore we obtain $\delta^- = -0.08$, and $\delta^+ = 0.08$. Setting δ^* to δ^+

$$\mathbf{c}^{\delta^*} = (0.48, 0.52)\,,\ (\mathbf{D}_i)^{\delta^*} = \begin{pmatrix} -1.0 & 0 \\ 0 & -3.0 \end{pmatrix},$$

$$\textit{with } \mu^{\delta^*}_{1,1} = 0.32314,\ \rho^{\delta^*} = 0.21292,\ \mathbf{H}^{\delta^*}_{ij} = \begin{pmatrix} 0.83333 & 0.16667 \\ 0 & 3 \end{pmatrix}.$$

***Example* 6.9.** *In the second example the representation of PH_j is in series canonical form*

$$\pi = (0.440.56)\,,\ \mathbf{D}_j = \begin{pmatrix} -1.0 & 1.0 \\ 0 & -12.347 \end{pmatrix}.$$

and PH_i is given by

$$\mathbf{c} = (0.7, 0.3)\,,\ \mathbf{D}_i = \begin{pmatrix} -1 & 0.8 \\ 0 & -1.5 \end{pmatrix},$$

The maximal first joint moment for PHDs in composition is

$$\mu_{1,1} = 0.68297,\ \rho = 0.020710,\ \mathbf{H}_{ij} = \begin{pmatrix} 0 & 0.2 \\ 0.76744 & 0.73256 \end{pmatrix}.$$

Furthermore we obtain $\delta^- = -0.09333$, and $\delta^+ = 0.3$. Here the PH_i cannot be transformed into hyperexponential representation. Setting δ^* to δ^+ the first joint moment can be maximized. Then the representation of the $PH_i^{\delta^*}$ is

$$\mathbf{c}^{\delta^*} = (1,0)\,,\ (\mathbf{D}_i)^{\delta^*} = \begin{pmatrix} -1.0 & 0.41 \\ 0 & -1.5 \end{pmatrix},$$

$$\text{with } \mu_{1,1}^{\delta^*} = 0.81646,\ \rho^{\delta^*} = 0.16191,\ \mathbf{H}_{ij}^{\delta^*} = \begin{pmatrix} 0.03 & 0.56 \\ 1.5 & 0 \end{pmatrix}.$$

6.2.5. Geometric Interpretation of the Transformation

In this section we present a geometrical interpretation of PHD with representation $(\pi, \mathbf{D}_0)$, of its hyperexponential and canonical forms according to [89, 61, 147]. Again, we consider the 2-order case. Furthermore, we interpret our transformation results (6.2) in context of PHD polytops and study the following problem: Given that both PHDs in composition $((\pi_i, \mathbf{D}_i),\ (\pi_j, \mathbf{D}_j),\ \mathbf{H}_{ij})$ are in hyperexponential representation, what is the optimal initial probability distribution π^* of PHD PH_j which maximizes the first joint moment of PHD composition; where is π^* positioned in the PHD polytop?

First, if matrix representations $\mathbf{D}_i$, $\mathbf{D}_j$ of both hyperexponential PHDs are fitted separately, it is important to know the initial distribution π^* of PH_j with the maximal or minimal first joint moment that can be reached by PHD composition. Secondly, using the knowledge about the position of the optimal π^* in the PHD polytop, we can show that the transformation of PHD PH_j using $\delta^* = \delta^-$ decreases the distance to the optimal π^* in the PHD polytop.

We consider both APHDs with representations given in (6.6) - (6.7), and (6.8), where PH_i is fixed to hyperexponential form. Observe that matrix representations $\mathbf{D}_i, \mathbf{D}_j$ in composition are in triangular form since both PHDs are acyclic.

We consider a polytope formed by PH_j which is the subject of the transformation into the series canonical form in Sec. 6.2.1. In the following π denotes an arbitrary initial distribution vector, and $\mathbf{D}_0$ an arbitrary PHD subgenerator.

Definition 6.1. *For PHD subgenerator matrix $\mathbf{D}_0$, the set $PHD(\mathbf{D}_0)$ collects all PHDs which can be represented by a tuple $(\pi, \mathbf{D}_0)$, such that π is any stochastic-vector*[1]

$$PHD(\mathbf{D}_0) = \{(\pi, \mathbf{D}_0) \mid 0 \le \pi(i) \le 1,\ i = 1, \ldots, n,\ \pi\mathbb{1} = 1\}.$$

[1] We assume that the point mass at zero, i.e. the probability of starting in absorbing state is 0. In this case we have $\pi(n+1) = 0$, and the time to absorption is strictly positive random variable.

Moreover, for matrix $\mathbf{D}_0$ let $\{\mathbf{D}_0(i,i),\ i=1,\ldots,n\}$ be the eigenvalues of the subgenerator $\mathbf{D}_0$, i.e., for a triangular subgenerator matrix its eigenvalues are given by its diagonal elements.

Let $\mathbf{u}_i$ be the i-th unit vector of $\mathbb{R}^n$, i.e. the vector of size n and the i-th element being one and all other elements being zero. Given a set of n vectors $\{\mathbf{x}_1,\ \mathbf{x}_2,\ldots,\ \mathbf{x}_n\}$, the convex set $conv(\{\mathbf{x}_1,\ \mathbf{x}_2,\ldots,\ \mathbf{x}_n\})$ is defined as $\{\mathbf{x}=a_1\mathbf{x}_1+a_2\mathbf{x}_2+\cdots+a_n\mathbf{x}_n,\ a_1+a_2+\cdots+a_n=1,\ 0\geq a_i\geq 1,\ i\in\{1,\ldots,n\}\}$ as mentioned in [89, 147].

For a finite n, the convex set $conv(\{\mathbf{x}_1,\ \mathbf{x}_2,\ldots,\ \mathbf{x}_n\})$ is denoted as a polytope. In particular, the polytope $conv(\{\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_n\})$ defined as $\{\mathbf{u}=a_1\mathbf{u}_1+a_2\mathbf{u}_2+\cdots+a_n\mathbf{u}_n,\ a_1+a_2+\cdots+a_n=1,\ 0\geq a_i\geq 1,\ i\in\{1,\ldots,n\}\}$ denotes the *probability vector polytope*, as described in [89, 147]. The affine set $aff(\{\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_n\})$ contains all vectors with a unit sum, and particularly all stochastic vectors [89].

Hyperexponential Representations In [147] it is pointed out that every eigenvector $\mathbf{v}_i$ of $n\times n$ matrix $\mathbf{D}_0$ corresponding to the real eigenvalue $\mathbf{D}_0(i,i)$ with a nonzero sum is associated with an exponential distribution with parameter $-\mathbf{D}_0(i,i)$. We have $\mathbf{v}_i\mathbf{D}_0=-\mathbf{D}_0(i,i)\,\mathbf{v}_i$ with $\mathbf{v}_i\mathbb{1}\neq 0$. If the vector $\mathbf{v}_i$ is normalized to have a unit sum, one can show that $\hat{\mathbf{v}}_i e^{\mathbf{D}_0 t}\mathbb{1}=e^{\mathbf{D}_0(i,i)\,t}$, for $t\geq 0$ and $\hat{\mathbf{v}}_i=\frac{\mathbf{v}_i}{\mathbf{v}_i\mathbb{1}}$. Thus the vector $\mathbf{v}_i$ represents an exponential PHD with the corresponding subgenerator matrix $\mathbf{D}_0^{(i)}=(\mathbf{D}_0(i,i))$ [147].

Since we have n eigenvectors $\mathbf{v}_i$, $i=1,\ \ldots,\ n$, each of them represents i-th exponential distribution with rate $\lambda(i)$ (see Eq. 2.59). The normalized eigenvectors $\hat{\mathbf{v}}_i$ are contained in the affine set $aff(\{\mathbf{u}_1,\mathbf{u}_2,\ldots,\mathbf{u}_n\})$ such that geometrical and stochastic interpretation can be related here as stated in [89].

In particular, the polytope $conv(\{\hat{\mathbf{v}}_1,\ \hat{\mathbf{v}}_2,\ \ldots,\ \hat{\mathbf{v}}_n\})$ corresponds to a diagonal matrix $\mathbf{D}_0^{diag}$, i.e. the matrix with elements $\mathbf{D}_0(i,i)$, $i=1,\ldots,n$, on the diagonal and all other elements are zero. In fact, one can use the similarity transformation with a matrix $\mathbf{B}$ with row vectors $\hat{\mathbf{v}}_i$, $i=1,\ldots,n$, such that $\mathbf{B}\,\mathbf{D}_0=\mathbf{D}_0^{diag}\,\mathbf{B}$ [47].

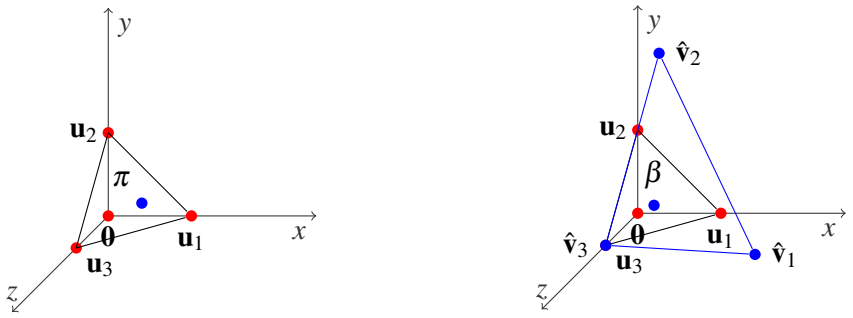
If a matrix $\mathbf{B}$ with the above property exists, then PHD $(\pi,\mathbf{D}_0)$ has hyperexponential representation $(\beta,\mathbf{D}_0^{diag})$. Particularly, $\pi\in conv(\{\hat{\mathbf{v}}_1,\ \hat{\mathbf{v}}_2,\ \ldots,\ \hat{\mathbf{v}}_n\})$ such that π is an affine combination of $\{\hat{\mathbf{v}}_1,\ \hat{\mathbf{v}}_2,\ \ldots,\ \hat{\mathbf{v}}_n\}$, i.e. $\pi=\beta(1)\hat{\mathbf{v}}_1+\beta(2)\hat{\mathbf{v}}_2+\ldots+\beta(n)\hat{\mathbf{v}}_n$ as given in [137].

***Example* 6.10.** *We now visualize the geometrical concepts from the literature [137, 89, 147]. Let $(\pi,\mathbf{D}_0)$ be a 3-order APHD with the following parameters*

$$\pi=(0.5,0.25,0.25)\,,\ \mathbf{D}_0=\begin{pmatrix}-4.5 & 0.5 & 0.2\\ 0 & -3 & 1\\ 0 & 0 & -0.5\end{pmatrix}. \tag{6.18}$$

In Figure 6.2(a), an example of the polytope of $\mathbf{D}_0$ in a 3-dimensional coordinate system is plotted. Points in the coordinate system represent possible initial probability vectors as discussed in [61, 147]. For example, the point at the origin $\mathbf{0}=(0,0,0)$ represents the PHD with representation $(\mathbf{0},\mathbf{D}_0)$ such that the probability of starting in the absorbing state is 1.

The points $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_3$ correspond to PHDs $(\mathbf{u}_1,\mathbf{D}_0)$, $(\mathbf{u}_2,\mathbf{D}_0)$, and $(\mathbf{u}_3,\mathbf{D}_0)$, respectively. E.g., the PHD with representation $(\mathbf{u}_3,\mathbf{D}_0)$ is the exponential distribution since $\mathbf{u}_3=(0,0,1)$, hence the PHD can be characterized by the rate parameter $\lambda(3)=$



(a) The polytope corresponding to APHD with subgenerator $\mathbf{D}_0$. Vector π is visualized on the face of the polytope.

(b) The extention of polytope $PHD(\mathbf{D}_0)$ to polytope corresponding to the subgenerator $\mathbf{D}_0^{diag}$.

Figure 6.2.: Graphical representation of the polytope corresponding to subgenerator matrix $\mathbf{D}_0$ and of the polytope corresponding to diagonal subgenerator matrix $\mathbf{D}_0^{diag}$ based on [61, 147].

0.5 *such that only the third phase is visited before absorption. In contrast, the less trivial PHD with representation* $(\mathbf{u}_2, \mathbf{D}_0)$ *decribes a PHD where the second transient phase is entered with probability* 1*, and where either the sequence of transient phases* $< 2,\ 3 >$ *is traversed, or the absorption occurs directly from the second phase with rate* 2*.*

Given a set of vectors $\{\mathbf{0}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ *from the three-dimensional real vector space, the visualized polytope is the convex hull of the zero vector* $\mathbf{0}$ *and the unit vectors in* $\mathbb{R}^3$*, i.e. it is the convex hull of the corresponding extreme points. Each element* $\pi \in conv(\{\mathbf{0}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\})$ *is denoted as a convex combination of the vectors* $\mathbf{0}$, $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$*.*

Then π *represents a valid probability distribution vector of PHD with subgenerator matrix* $\mathbf{D}_0$ *and the point mass at zero* ≥ 0*. In contrast, the face of the polytope which is given by the plane* $\mathbf{u}_1\mathbf{u}_2\mathbf{u}_3$ *contains initial probability vectors with point mass at zero* $\pi(n+1) = 0$ *[147]. Note, that line segments joining each of two PHDs also belong to the convex set.*

Suppose now that $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ *are eigenvectors of* $\mathbf{D}_0$ *given in Eq. 6.18 corresponding to three distinct real eigenvalues denoted as* $-\lambda(1), -\lambda(2), -\lambda(3)$*. We assume that* $\lambda(1) > \lambda(2) > \lambda(3)$*. Then we obtain*

$$\begin{aligned} \mathbf{v}_1 &= (0.9482, -0.3161, 0.0316) && \text{for the eigenvalue } -4.5, \\ \mathbf{v}_2 &= (0, 0.9285, -0.3714) && \text{for the eigenvalue } -3, \\ \mathbf{v}_3 &= (0, 0, 1) && \text{for the eigenvalue } -0.5. \end{aligned}$$

The normalized vectors result in

$$\begin{aligned} \hat{\mathbf{v}}_1 &= (1.4287, -0.4763, 0.0476), \\ \hat{\mathbf{v}}_2 &= (0, 1.6667, -0.6667), \\ \hat{\mathbf{v}}_3 &= (0, 0, 1). \end{aligned}$$

Each eigenvector $\mathbf{v}_i$*,* $i = 1, 2, 3$*, corresponds to an exponential distribution with the corresponding rate* $\lambda(1) = 4.5$*,* $\lambda(2) = 3$*, and* $\lambda(3) = 0.5$ *respectively. The polytope*

for the diagonal matrix $\mathbf{D}_0^{diag}$ is the convex hull of zero vector $\mathbf{0}$ and normalized eigenvectors $\hat{\mathbf{v}}_i$, $i = 1,\ldots,3$ [147]. We obtain the following representation for the diagonal matrix

$$\mathbf{D}_0^{diag} = \begin{pmatrix} -0.5 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & -4.5 \end{pmatrix}. \tag{6.19}$$

The corresponding polytope is visualized in Fig. 6.2(b).

The authors in [89, 137] stated that if the intersection of the probability vector polytop conv($\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$) and conv($\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3\}$) is not empty, then the PHD with representation $(\pi, \mathbf{D}_0)$ has a hyperexponential representation $(\pi', \mathbf{D}_0^{diag})$ for some initial probability vector π'. In general, the intersection set contains all PHDs which have representations using $\mathbf{D}_0$ and $\mathbf{D}_0^{diag}$ [147].

In our example, the PHD defined in Eq. 6.18 has hyperexponential representation $(\beta, \mathbf{D}_0^{diag})$. The following holds for the initial distribution vector

$$\pi = (0.5, 0.25, 0.25) = 0.35\,\hat{\mathbf{v}}_1 + 0.25\,\hat{\mathbf{v}}_3 + 0.4\,\hat{\mathbf{v}}_3.$$

Thus the vector $\beta = (0.4, 0.25, 0.35)$ together with subgenerator matrix $\mathbf{D}_0^{diag}$ defines the corresponding hyperexponential PHD. In Figure 6.2(b), the polytope conv($\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3\}$) together with point β is visualized. It is in two-dimensional affine space $aff(\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\})$ as has been presented in the example in [89]. Particularly, the intersection of both polytopes is given by $\mathbf{0}\mathbf{u}_2\mathbf{u}_3$ similarly to the example presented in [147].

Canonical Representations In [61] it was shown how the polytope $PHD(\mathbf{D}_0^{diag})$ can be expanded to a polytope containing linear combinations of eigenvectors $\hat{\mathbf{v}}_i$, $i = 1,\ldots,n$. The authors first denoted $\mathscr{V}$ as the n-dimensional real vector space generated by vectors $\hat{\mathbf{v}}_i$ for n-order PHD. Then $\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \ldots, \hat{\mathbf{v}}_n\}$ is the basis of $\mathscr{V}$. The subset of $\mathscr{V}$ containing vectors which represent probability distributions is denoted by $\mathscr{C}_n$. Furthermore, it holds for the eigenvalues that $\lambda(1) > \lambda(2) > \lambda(3)$.

The important result from [61] is that for $i,\ j \in \{1, \ldots, n\}$, $i < j$, the function $a\hat{\mathbf{v}}_i + (1-a)\hat{\mathbf{v}}_j$, for $\frac{\lambda(j)}{(\lambda(j)-\lambda(i))} \leq a \leq 1$, is also a probability distribution function and thus belongs to the set $\mathscr{C}_n$ (see Lemma 2 in [61]). Based on this result we obtain the polytope $conv(\{\mathbf{f}_{1\ldots i}, 1 \leq i \leq 3\})$ in the three-dimensional case as follows

$$\begin{aligned}
\mathbf{f}_1 &= \hat{\mathbf{v}}_1, \\
\mathbf{f}_{12} &= \tfrac{\lambda(2)}{\lambda(2)-\lambda(1)}\hat{\mathbf{v}}_1 + \tfrac{\lambda(1)}{\lambda(1)-\lambda(2)}\hat{\mathbf{v}}_2, \\
\mathbf{f}_{123} &= \tfrac{\lambda(2)\lambda(3)}{(\lambda(2)-\lambda(1))(\lambda(3)-\lambda(1))}\hat{\mathbf{v}}_1 + \tfrac{\lambda(1)\lambda(3)}{(\lambda(1)-\lambda(2))(\lambda(3)-\lambda(2))}\hat{\mathbf{v}}_2 + \tfrac{\lambda(1)\lambda(2)}{(\lambda(1)-\lambda(3))(\lambda(2)-\lambda(3))}\hat{\mathbf{v}}_3,
\end{aligned}$$

as additionally explained in [89]. In general, the following formula can be used to construct the generalized Erlang distribution functions $\mathbf{f}_{ij}$ considering two exponential PHDs

$$\mathbf{f}_{ij} = \frac{\lambda(j)}{\lambda(j)-\lambda(i)}\hat{\mathbf{v}}_i + \frac{\lambda(i)}{\lambda(i)-\lambda(j)}\hat{\mathbf{v}}_j, \tag{6.20}$$

for $i, j \in \{1,2,3\}$ with $i < j$ and thus $\lambda(i) > \lambda(j)$. Then each vector $\mathbf{f}_{ij}$ represents hypoexponential PHD, i.e., it is the convolution of two exponential PHDs with rates $\lambda(i)$ and $\lambda(j)$, respectively.

***Example* 6.11.** *For example 6.10, we obtain the following vectors forming the set $\mathscr{C}_3$:*

$$\begin{aligned}
\mathbf{f}_1 &= \mathbf{v}_1 = (1.4287, -0.4763, 0.0476),\\
\mathbf{f}_{12} &= \tfrac{3}{3-4.5}(1.4287, -0.4763, 0.0476) + \tfrac{4.5}{4.5-3}(0, 1.6667, -0.6667)\\
&= (-2.8574, 5.9527, -2.0953)\\
\mathbf{f}_{13} &= \tfrac{0.5}{0.5-4.5}(1.4287, -0.4763, 0.0476) + \tfrac{4.5}{4.5-0.5}(0,0,1)\\
&= (-0.1786, 0.0595, 1.1191),
\end{aligned}$$

$$\mathbf{f}_{23} = \tfrac{0.5}{0.5-3}(0, 1.6667, -0.6667) + \tfrac{3}{3-0.5}(0,0,1) = (0, -0.3333, 1.3333),$$

$$\begin{aligned}
\mathbf{f}_{123} &= \tfrac{3\cdot 0.5}{(3-4.5)(0.5-4.5)}(1.4287, -0.4763, 0.0476) + \tfrac{4.5\cdot 0.5}{(4.5-3)(0.5-3)}(0, 1.6667, -0.6667)\\
&+ \tfrac{4.5\cdot 3}{(4.5-0.5)(3-0.5)}(0,0,1) = (0.3572, -1.1191, 1.7619).
\end{aligned}$$

The computed vectors and polytopes $PHD(\mathbf{D}_0^{diag})$, $PHD(\mathbf{D}_0^{can})$ are visualized in Fig. 6.3 based on [61, 89].

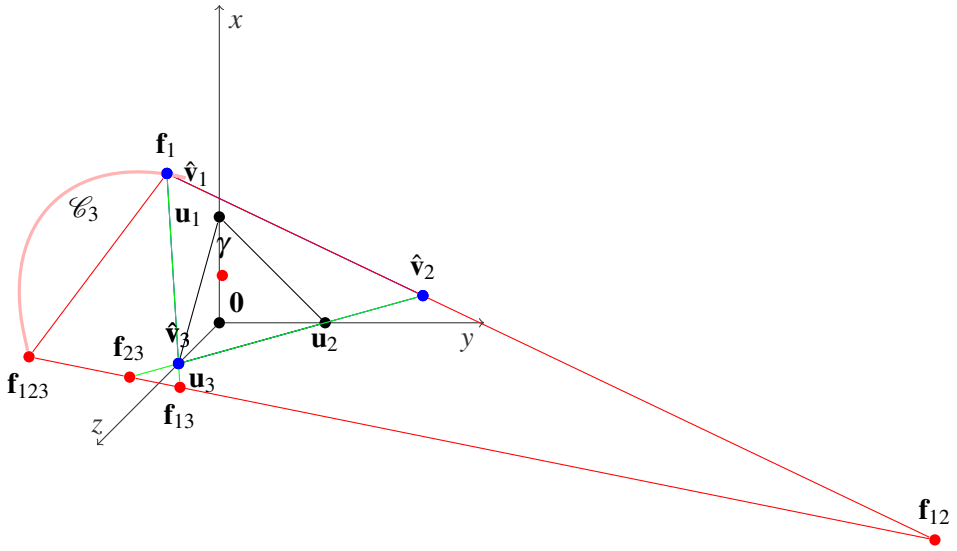


Figure 6.3.: Graphical representation of the convex set $\mathscr{C}_3$ from [61, 89]. The polytopes $PHD(\mathbf{D}_0^{diag})$, $PHD(\mathbf{D}_0^{can})$ are highlighted in green and red, respectively.

In fact, the Lemma from [61] states that the line segment joining exponential distributions given by $\hat{\mathbf{v}}_i$, $i = 1, \ldots, n$ is part of the convex set $\mathscr{C}_n$. In particular, all convex combinations of two eigenvectors $\hat{\mathbf{v}}_i$, $\hat{\mathbf{v}}_j$, $i, j \in \{1, \ldots, n\}$ are also probability distributions representing the convex mixture of corresponding exponential distributions and are positioned at line segments between vectors $\hat{\mathbf{v}}_i$, $\hat{\mathbf{v}}_j$. Furthermore, these line segments can be extended to obtain more probability vectors in $\mathscr{C}_n$ which, e.g. can be associated with canonical PHDs. An interesting fact is, that vectors $\mathbf{f}_1$, $\mathbf{f}_{12}$, and $\mathbf{f}_{123}$ are extremal points of the convex set $\mathscr{C}_3$. E.g., the line segment joining the exponential

PHD $\mathbf{f}_1$ and the hypoexponential PHD $\mathbf{f}_{12}$ contains PHDs which can be obtained as a random combination of these two PHDs.

Using vectors $\{\mathbf{f}_1, \mathbf{f}_{12}, \mathbf{f}_{123}\}$ the polytope $conv(\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3\})$ can be extended to the polytope $conv(\{\mathbf{f}_1, \mathbf{f}_{12}, \mathbf{f}_{123}\})$. Furthermore, the polytope $conv(\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2, \hat{\mathbf{v}}_3\})$ is the subset of the polytope $conv(\{\mathbf{f}_1, \mathbf{f}_{12}, \mathbf{f}_{123}\})$, which is the convex hull of the zero vector $\mathbf{0}$ and the vectors $\mathbf{f}_1, \mathbf{f}_{12}, \mathbf{f}_{123}$ [147]. It has been stated in [61] that $conv(\{\mathbf{f}_1, \mathbf{f}_{12}, \mathbf{f}_{123}\})$ contains PHDs which have bi-diagonal representation of order 3 or smaller. Thus, for all probability distribution vectors $\pi \in conv(\{\mathbf{f}_1, \mathbf{f}_{12}, \mathbf{f}_{123}\})$ PHD $(\pi, \mathbf{D}_0)$ has ordered Coxian representation which is also known as series canonical form [56], Eq. 2.19(b). Note that the ordered Coxian representation is given by APHD matrix in bi-diagonal form with rates $\lambda(1) \geq \lambda(2) \geq \ldots \lambda(n) > 0$.

***Example* 6.12.** *For examples 6.10 and 6.11 let the subgenerator of APHD in series canonical form be given by*

$$\mathbf{D}_0^{can} = \begin{pmatrix} -0.5 & 0.5 & 0 \\ 0 & -3 & 3 \\ 0 & 0 & -4.5 \end{pmatrix},$$

where three eigenvalues are given by corresponding diagonal elements, i.e., -0.5*,* -3 *and* -4.5*, respectively. The original APHD is given in Eq. 6.18. If* $\pi = \gamma(1)\mathbf{f}_{123} + \gamma(2)\mathbf{f}_{12} + \gamma(3)\mathbf{f}_1$ *with* $\gamma(1) + \gamma(2) + \gamma(3) = 1$*,* $\gamma(i) \geq 0$*, then* $(\gamma, \mathbf{D}_0^{can})$ *represents the same distribution as given by* (6.18)*, where* γ *is the initial distribution vector with the* i*-th element* $\gamma(i)$*. We obtain*

$$(0.5, 0.25, 0.25) = 0.56111\ \mathbf{f}_1 + 0.14259\ \mathbf{f}_{12} + 0.29630\ \mathbf{f}_{123},$$

which results in the initial distribution vector $\gamma = (0.29630, 0.14259, 0.56111)$ *for series canonical form of the considered APHD. The point given by* γ *is visualized in Fig. 6.3. We refer to [61] for the compelete construction of the set* $\mathscr{C}_3$ *and parameterization of the curve delimiting the region outside the polytope* $conv(\{\mathbf{f}_1, \mathbf{f}_{12}, \mathbf{f}_{123}\})$*.*

Geometric Results for 2-Phase Case We now proceed with polytope construction in two-dimensional case. Since the presented transformations in Sec. 6.2.1-6.2.4 operate on 2-order APHDs, we should consider the corresponding polytopes.

***Example* 6.13.** *We assume that PHDs in composition are both in hyperexponential representation. Let PHD* PH_i *be parameterized as following*

$$\mathbf{c} = (0.01, 0.99),\ \mathbf{D}_i = \begin{pmatrix} -1 & 0 \\ 0 & -11.22449 \end{pmatrix}, \tag{6.21}$$

with $\mu_1 = 1$*,* $\mu_2 = 11.22449$*. Let PHD* PH_j *in composition have the representation*

$$\pi = (0.99, 0.01),\ \mathbf{D}_j = \begin{pmatrix} -1 & 0 \\ 0 & -5 \end{pmatrix}, \tag{6.22}$$

where $\lambda(1) = 1$*,* $\lambda(2) = 5$*,* $c = 0.01$*, and* $\pi = 0.99$*. The polytope* $conv(\{\mathbf{u}_1, \mathbf{u}_2\})$ *corresponding to PHD subgenerator* $\mathbf{D}_j$ *is visualized in Fig. 6.4 where it is highlighted in blue.*

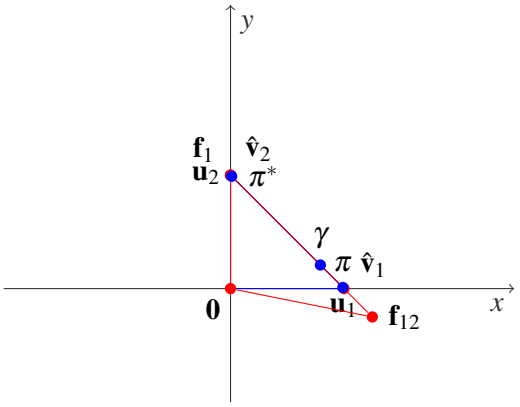


Figure 6.4.: Graphical representation of polytopes $PHD(\mathbf{D}_j)$ and $PHD(\mathbf{D}_j^{can})$ for 2-order PHDs. The extreme points and the position of initial distribution vector $\pi = (0.99, 0.01)$ are highlighted in red and blue.

To expand the polytope conv$(\{\mathbf{u}_1, \mathbf{u}_2\})$ *we first compute eigenvectors* $\hat{\mathbf{v}}_2 = (0,1)$ *corresponding to the eigenvalue* $-\lambda(2) = -5$*, and* $\hat{\mathbf{v}}_1 = (1,0)$ *corresponding to the eigenvalue* $-\lambda(1) = -1$.

Next we need to expand the polytope conv$(\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2\})$ *to cover more vectors which are probability functions and which are associated with APHDs in canonical form. To obtain the polytope conv*$(\{\mathbf{f}_1, \mathbf{f}_{12}\})$ *we compute*

$$\begin{aligned} \mathbf{f}_1 &= \hat{\mathbf{v}}_2 = (0,1), \\ \mathbf{f}_{12} &= \tfrac{1}{1-5}\hat{\mathbf{v}}_2 + \tfrac{5}{5-1}\hat{\mathbf{v}}_1 = -0.25 \cdot (0,1) + 1.25 \cdot (1,0) = (1.25, -0.25). \end{aligned}$$

It is easy to see that $\mathbf{f}_{1\ldots j}\mathbb{1} = 1$ *for* $j = 1,2$*. Then the polytope corresponding to the APHD subgenerator in series canonical form is visualized in Fig. 6.4 where it is highlighted in red.*

The polytope $PHD(\mathbf{D}_j^{can})$ *is formed by* $\mathbf{0}\mathbf{f}_1\mathbf{f}_{12}$*, the corresponding subgenerator is given below*

$$\gamma = (0.792, 0.208)\,,\ \mathbf{D}_j^{can} = \begin{pmatrix} -1 & 1 \\ 0 & -5 \end{pmatrix},$$

such that $\pi = (0.99, 0.01)$ *is affine combination of* $\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2\}$*, i.e.*
$\pi = \gamma(1)\mathbf{f}_{12} + \gamma(2)\mathbf{f}_1 = (0.99, -0.198) + (0, 0.208) = (0.99, 0.01)$ *for* $\gamma(1) + \gamma(2) = 1$.
PHD PH_j *in canonical form* $(\gamma, \mathbf{D}_j^{can})$ *is represented in Fig. 6.4 by the point* γ.

The interesting question is now: Given that both PHDs in composition are in hyperexponential representation, what is the optimal initial probability distribution π^* of PHD PH_j in composition maximizing the first joint moment, and where is it positioned in the polytope?

It should be clear, that finding π^* only gives insights about it's positioning in the polytope. In fact, the PHD with representation $(\pi^*, \mathbf{D}_j)$ is not equal to PH_j with representation $(\pi, \mathbf{D}_j)$. Furthermore, PHD $(\pi^*, \mathbf{D}_j)$ strongly depends on parameters of the first PHD $(\mathbf{c}, \mathbf{D}_i)$ in composition.

In other words, we are looking for π^* maximizing the achievable correlation of PHD composition containing $(\mathbf{c},\mathbf{D}_i)$ and $(\pi^*,\mathbf{D}_j)$ such that

$$\pi^* = \arg \max_{\pi \in PHD(\mathbf{D}_j)} (\rho\left((\mathbf{c},\mathbf{D}_i),(\pi,\mathbf{D}_j)\right)). \tag{6.23}$$

Corollary 2. *If PHD PH_j in composition is transformed into series canonical form 2.19(b) and $\pi^* = c$, then this representation results in the maximal correlation, given that first PHD PH_i in composition is in hyperexponential representation.*

The proof of Corollary 2 can be found in App.C.1

***Example* 6.14.** *In Figure 6.5 we plotted the correlation function for our numerical example given in (6.21)-(6.22).*

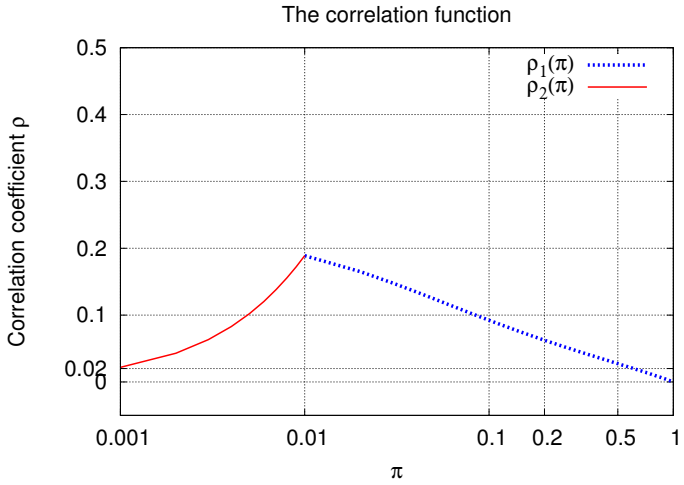


Figure 6.5.: The correlation functions $\rho_1(\pi)$, $\rho_2(\pi)$ defined in proof C.1.

The following corollary of Theorem 6.1 states that transformation of PH_j in composition using $\delta^* = \delta^-$ decreases the distance to the optimal point π^* in the corresponding polytope.

Corollary 3. *Transformation of PHD PH_j into series canonical form decreases the distance to the optimal point π^* in the polytope conv$(\{\hat{\mathbf{v}}_1, \hat{\mathbf{v}}_2\})$.*

Proof. First note that euclidian distance for two points $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{y}_i)^2}, \tag{6.24}$$

i.e, it is the length of line segment connecting these points.

Considering the euclidian distance between point π representing PH_j in hyperexponential form and point representing the optimal probability distribution π^* we obtain

$$\begin{aligned} d(\pi,\pi^*) &= \sqrt{(\pi - c)^2 + (1 - \pi - 1 + c)^2} = \sqrt{(\pi - c)^2 + (c - \pi)^2} \\ &= \sqrt{2(\pi^2 + c^2 - 2\pi c)}. \end{aligned}$$

The following holds for point γ which represents PH_j in series canonical form

$$\gamma = (\pi^\delta, 1-\pi^\delta),$$

such that $\pi^\delta = \pi + \delta = \pi - \frac{\pi(1-\lambda_{12})}{\lambda_2} = \frac{\pi(\lambda_2-1+\lambda_{12})}{\lambda_2}$, for $\delta = -\frac{\pi(1-\lambda_{12})}{\lambda_2}$ as shown in Sec. 6.2.1. In the following we denote the term $\frac{(\lambda_2-1+\lambda_{12})}{\lambda_2}$ as q such that $\pi^\delta = \pi q$.

The distance between the point γ and the optimal point π^*

$$\begin{aligned} d(\gamma,\pi^*) &= \sqrt{(\pi q - c)^2 + (1-\pi q - 1 + c)^2} = \sqrt{(\pi q - c)^2 + (c - \pi q)^2} \\ &= \sqrt{(\pi q)^2 - 2\pi q c + c^2 + c^2 - 2\pi q c + (\pi q)^2} \\ &= \sqrt{2((\pi q)^2 + c^2 - 2\pi q c)}. \end{aligned}$$

Now it can be shown that the following relation holds

$$d(\gamma,\pi^*) \leq d(\pi,\pi^*).$$

Observe that for PH_j in hyperexponential form it holds that $\lambda_{12} = 0$. Then $q = \frac{(\lambda_2-1)}{\lambda_2} < 1$ holds. Now consider the following relation

$$\begin{aligned} & \sqrt{2((\pi q)^2 - 2\pi q c + c^2)} && \leq && \sqrt{2(\pi^2 - 2\pi c + c^2)} \\ \Leftrightarrow\ & (\pi q)^2 - 2\pi q c + c^2 && \leq && \pi^2 - 2\pi c + c^2 \\ \Leftrightarrow\ & (\pi q)^2 - 2\pi q c && \leq && \pi^2 - 2\pi c \\ \Leftrightarrow\ & \pi q(\pi q - 2c) && \leq && \pi(\pi - 2c) \\ \Leftrightarrow\ & \pi q^2 - 2qc && \leq && \pi - 2c \\ \Leftrightarrow\ & q(\pi q - 2c) && \leq && \pi - 2c, \end{aligned}$$

where by multiplying π with some number $q < 1$ we obtain $\pi q < \pi$. Hence, $d(\gamma,\pi^*) \leq d(\pi,\pi^*)$ such that transformation of PH_j in composition in series canonical form increases the correlation of the given composition. □

6.3. Summary

In this chapter we presented a similarity transformation approach for 2-order PHDs in composition in order to generate a representation which is best suitable for fitting a transfer matrix with maximal first joint moment. When optimal representations are known, parameter estimation for both PHDs in composition and transfer matrices can be simplified. If only hyperexponential and canonical representations are considered the reduction of fitting complexity can be achieved by reducing the number of free parameters in the matrix representations. A further advantage is that for the case with two phases transfer matrices allowing for the maximal first joint moment can be generated directly.

When the representation of first PHD PH_i in composition is fixed, the bounds for the maximal achievable correlation can be specified. The maximal first joint moment $\mu^*_{1,1}$ of the composition can be achieved when successor PH_j is transformed into series canonical form, given that PH_i is in hyperexponential representation. This case corresponds to the upper bound of the maximal achievable correlation of a composition.

A lower bound for the achievable correlation is given when the representation of PHD PH_i is fixed to series canonical form. Unfortunately, the canonical form which is

often used in fitting algorithms is not suitable for PHD PH_i to model correlation of the composition. A PHD in canonical form has only one exit state which severely restricts possible entries and values in transfer matrix $\mathbf{H}_{ij}$, such that no flexibility for $\mathbf{H}_{ij}$ is available.

If the representation of successor PHD PH_j is fixed, then the following bounds for the maximal achievable correlation of composition are given. The upper bound for the maximal first joint moment $\mu^*_{1,1}$ of composition is given when PHD PH_i can be transformed into a hyperexponential representation, given that PH_j is in series canonical form.

If PH_j is in hyperexponential representation, then the optimal first joint moment $\mu^*_{1,1}$ is reached by PH_i in hyperexponential form. Obviously, the lower bound for the achievable correlation of composition is given when PHD PH_i is in canonical representation, given that the representation of PH_j is fixed.

In contrast, no results for PHDs in composition with an arbitrary number of phases are known. The optimal representation for n-order PHDs in composition in order to maximize the first joint moment remains subject to future research. In cases where the transformation does not result in an adequate first joint moment, e.g., if a PHD cannot be transformed into hyperexponential representation, the state space expansion technique described in [41] can be applied.

By adding additional states it is possible to find a representation allowing larger flexibility. We tried the expansion technique for hyperexponential PHDs where two additional states are added; first, an arbitrary phase is selected and represented by two states using Cumani's substitution step (see Fig. 2.18); then, the second state from Cumani's expansion is cloned. The third state then appears with probability $\frac{\lambda}{\mu}$, and the second cloned state appears with probability 0.

Our results indicate that following factors have impact on the maximal achievable correlation of expanded PHDs in composition: (1) the choice of the parameter μ in Cumani's substitution step, (2) the choice of the state which has to be expanded. Furthermore, it is unclear how many states are required to reach the maximal achievable correlation for both PHDs. All these questions remain for future investigations.

Chapter 7

Applications and Experiments

To demonstrate the potential of PH-Graph-based optimization in practice, we present several applications in this chapter. As already mentioned before, PHGs can be used to model various practical problems like finding shortest routes on streets for vehicles under congestions or natural disasters, rerouting of airlines and freight flows to avoid weather delays, routing in computer networks, reliability analysis of systems, and analysis of stochastic graphs. In many cases, correlation might be an important aspect, e.g., in large-scale applications correlated travel times on adjacent edges should not be neglected. In the first case study, we delve deeply in PHG-based modeling and optimization for finding shortest routes on streets, the application area where PHGs have a big potential. We use an available vehicle mobile trace for parameterization. The resulting PHG depicts vehicular traffic network with typical congestions often occuring in practice. At the same time, we increase the number of phases of involved PHDs in order to analyse its impact on achievable correlation and likelihood value. However, this relationship has not been studied closely for PH-Graphs and there might be traces for which increasing the PHD order can improve the quality of results.

Then, we extend PH-Graph model using bilateral PHDs to cover PHDs with support on the whole line $(-\infty, \infty)$. Then we show how analysis methods can be adapted for bilateral PHDs, and how PHGs with negative edge weights can be useful in computation of optimal investment policies for financial markets. Numerical investigation of solution methods for large PHG models are performed. The example from maintenance field is also introduced.

7.1. Shortest Path Computation

In this section we present a real world example from [40] for the shortest path computation based on simulation data. We used the synthetic vehicle mobility trace which collects a realistic simulation data of 24-hour car traffic in about 400 km^2 region of the city of Cologne, in Germany [2, 171]. The dataset captures microscopic and macroscopic dynamics of car movement patterns on realistic road data. The proposed vehicle trace has been generated using data available from TAPASCologne project [1] using different software tools, e.g. the microscopic road traffic simulation package SUMO and OpenStreetMap road information.

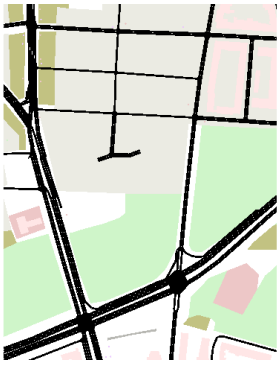


Figure 7.1.: Map of the modeled roads [2, 1].

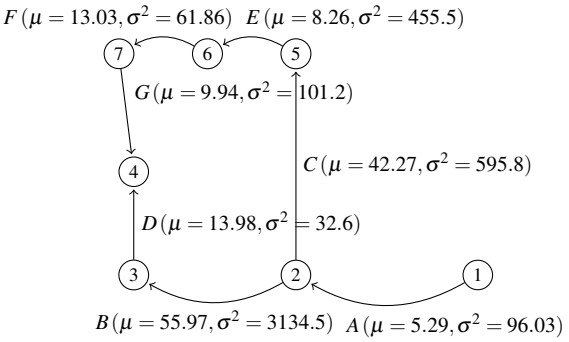


Figure 7.2.: The PHG corresponding to the modeled road area from [40].

High level of detail results from the real-world data collected by the German Federal Statistical Office, such that the information about vehicle driver behavior, time and location patterns could be exploited.

Each entry of the trace contains the simulation time, the identifying number of a vehicle, its xy-position on the plane and its speed in meters per second. Furthermore, we consider the car traffic trace for some main streets of the city Cologne, namely the Niehler Strasse, Neusser Strasse, and the Innere Kanalstrasse at the time 6 am to 8 am (see Fig. 7.1). During this time interval congestions on the roads often occur.

We filter the dataset [2] and extract the information about traveling times of vehicles passed through the road area mentioned, i.e. vehicles first passed through the Innere Kanalstrasse and then the Neusser Strasse. Similarly the traveling times for the vehicles first passed through the Innere Kanalstrasse and then the Niehler Strasse have been extracted. Then, in PHG the weights of edges correspond to the derived traveling times of vehicles. Figure 7.2 shows two paths from initial node 1 to the destination node 4.

PHG parameterization The PHG from [40] visualized in Fig. 7.2 has the set of edges $\mathscr{E} = \{A, B, C, D, E, F, G\}$. Let $PHD_J = (\boldsymbol{\pi}^{(J)}, \mathbf{D}_0^{(J)})$ be the PH distribution of the random variable $X^{(J)}$ describing the weight of the edge J for $J \in \mathscr{E}$.

There are two paths from the initial node 1 to the destination 4, namely $p_1 = (A, C, E, F, G)$ and $p_2 = (A, B, D)$. There is a positive correlation between traveling times along the Innere Kanalstrasse and the traveling times along the Neusser Strasse. Thus, if the Innere Kanalstrasse is congested, the Neusser Strasse is also congested. For this reason the weight of the edge A is correlated with the weight of the adjacent edge B.

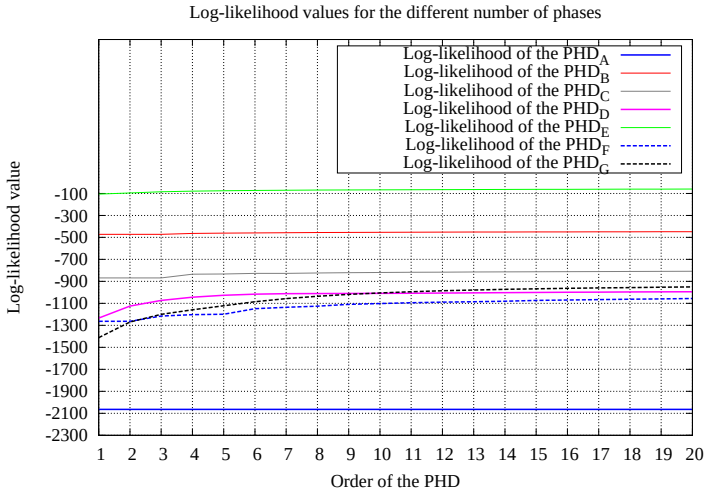


Figure 7.3.: Log-likelihood values for PHD_Js of order i, $i = 1,\ldots,20$, and $J \in \{A,B,C,D,E,F,G\}$.

We fitted the traveling time data from the trace to PHDs of order i, $i = 1,\ldots,20$, using the software $gfit$ [168]. Figure 7.3 shows the values of the log-likelihood function according to the traces. Additionally, the impact of the fitted PHD order on the achievable correlation coefficient ρ_{AB} between PHD_A and PHD_B is shown in Tab. 7.1. For the reasons explained in [40] the vehicle mobility traces can be fitted adequately by PHDs with already 6 phases. Here the fitting tool $gfit$ has been used to generate PHDs. The software produces PHDs in hyperexponential representation which is best suited for the subsequent correlation fitting [41] (see also Chap. 6).

One can see that increasing number of phases results in the rise in log-likelihood value curves as represented in Figure 7.3. For the given mobility traces already 6

PHD order	ρ_{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

Table 7.1.: Impact of the PHD order on the correlation for the PHD_A $(\pi^{(A)},\mathbf{D}_0^{(A)})$ and the PHD_B $(\pi^{(B)},\mathbf{D}_0^{(B)})$ as given in [40].

phases are sufficient, i.e. further increasing of PHD order would result in insignificant rise of the log-likelihood values.

The same holds for the achievable correlation coefficient of the PHD_A and PHD_B. Increasing number of phases results in a higher correlation coefficient. However, it is an open issue how the maximal achievable correlation coefficient and the PHD order coincide (the interested reader is referred to [41]). In this example, we choose PHDs of order 6 since the highest increase of the reachable coefficient of correlation takes place from the order $n_J = 6$ as shown in Table 7.1. The representation of the fitted PHDs and transfer matrices can be found in Sec D.3. Fitting of the transfer matrices has been performed according to the joint moments method (4.45).

Computation of the minimal expected traveling time We considered the graph instance shown in Fig. 7.2 where PHDs are substituted by their expectations and correlations between edge weights are completely neglected. Then, the initial proper policy results in $\mathbf{u}_0 = (A, B, D)$ with the weight of 75.2639. The weight of the alternative policy $\mathbf{u}_1 = (A, C, E, F, G)$ is 78.8236.

Assume that the vehicle arrives at the node $fin(A) = 2$. To arrive at the destination node 4 the vehicle has two competing options for the next possible edge to traverse. These are the edges $B \in A\bullet$ and $C \in A\bullet$. The decisions of the vehicle should be based on the time required to pass edge A if correlated traveling times on adjacent edges should be considered.

In the following we consider the policy iteration approach to minimize the expected time until arriving at the destination node 4. For a long time required on the edge A, the optimal choice is the upper adjacent edge C, since traveling times on edges A and C are not correlated whereas traveling times on edges A and B are positively correlated.

In turn, if the time on the edge A becomes shorter, it is better to choose the edge B to exploit the effect of positive correlation. More precisely, from a phase where remaining time before absorption is shorter than the average, the optimal policy chooses edge B. In the case where the process is in a phase where remaining time till absorption is longer than the average, the optimal decision is the successor edge C.

Table 7.2.: Expected travel times from initial node $v_{ini} = 1$ to the destination node $v_{fin} = 4$ depending on the exit phase of the PHD_A $(\boldsymbol{\pi}^{(A)}, \mathbf{D}_0^{(A)})$, and optimal decisions of successor edges as given in [40].

	Mean Travel Time to $v_{fin} = 4$		
Exit Phase of the PHD_A	Succ. B	Succ. C	Optimal Successor Edge
Phase 1	157.4980	118.6390	C
Phase 2	109.9351	74.6470	C
Phase 3	105.3878	70.0997	C
Phase 4	47.0699	71.2365	B
Phase 5	44.7962	68.9628	B
Phase 6	28.7180	68.9628	B

Phase-based optimal decisions We computed the optimal policy $\mathbf{u}^*$ and the corresponding weight vector $\mathbf{g}^*$ via policy iteration. The expected traveling times to the destination node 4 and the optimal choice of the successor edge depending on the phase of the PHD_A are given in Table 7.2.

The entries in Table 7.2 demonstrate that the optimal policy can change based on the congestion level of the traversed predecessor edge A. It can be seen that the optimal decision is edge C, if the exit phase of the PHD_A is one of the phases where the remaining time till absorption is on average longer, e.g., the phase 1. For a long time required on the edge A the probability to escape from a phase where the remaining time till absorption is on average longer is higher than the probability to escape from the phase where the remaining time till absorption is on average faster.

Consequently, the probability to enter a phase of the successor PHD_B where the remaining time before absorption is on average longer is higher than the probability to enter a phase with the on average faster remaining time before absorption. This behavior is modeled by the correlation matrix $\mathbf{H}_{AB}^{\rho=0.19576}$ (see Sec. D.3 for detail). Thus the case of the on average longer time required to pass the successor edge B is more likely to occur.

Assume now that the exit phase of the PHD_A has the remaining time before absorption which is on average faster, e.g. the phase 6. Since PHD_A and PHD_B are positively correlated, the probability to enter a phase of PHD_B where the remaining time before absorption is on average faster is higher than the probability to enter a phase where the remaining time before absorption is on average longer. Hence, the small time value required to traverse the edge B is more likely to occur.

History-based optimal decisions Of course, the decisions resulting from the PI algorithm and presented in Table 7.2 depend on phases of the PHD_A which are not part of the real system. Assume that the history of realized values $w_1, \ldots, w_l$ become known to the vehicle as it traverses the selected route. Then the decisions of the vehicle have to depend on the history vectors $\bar{\psi}_{(i_1,w_1,\ldots,i_l,w_l)}$ computed with (5.15)-(5.16).

We computed vectors $\bar{\psi}_{(A,w,i_J,0)}$ with $w \in [0.1, 80]$, $J \in \{B,C\}$. The values of the first conditional moment of the traveling time for the adjacent edges are summarized in Figure 7.4. Results are computed for PHDs with $4, 6, 11$, and 15 phases.

Figure 7.4 shows the expected traveling times at the adjacent edges B and C depending on the traveling time of the predecessor 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 the edge A. For larger values the curves are almost identical for all numbers of phases shown in the graph.

Using the history vectors $\bar{\psi}_{(i_1,w_1,\ldots,i_L,w_l)}$ the conditional weights of the remaining paths from the node $fin(A) = 2$ to the destination node $v_{fin} = 4$ can be computed. The best subsequent edge after passing the initial edge A can then be computed with (5.48), such that the optimal decision depends on the previous weight of the edge A and not on the state of the PHD.

The values of the first conditional moment for the remaining traveling time are

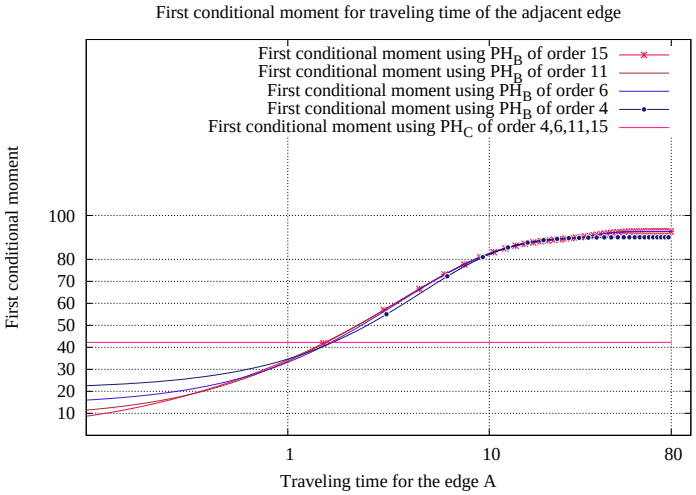


Figure 7.4.: The expected traveling time for the adjacent edges B and C depending on the weight of the predecessor edge A.

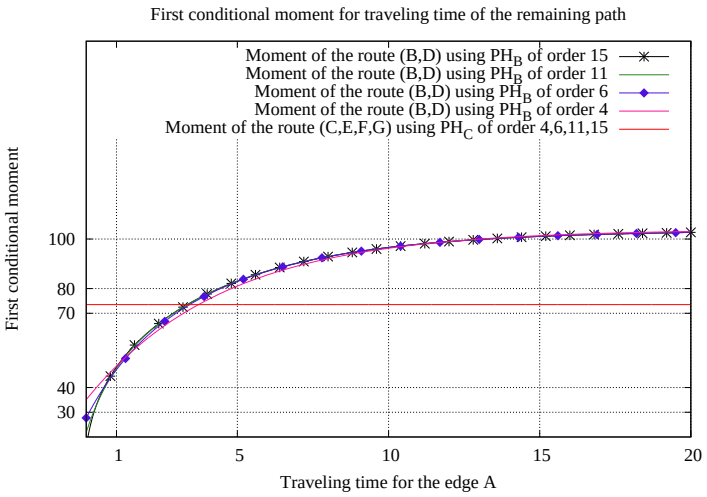


Figure 7.5.: The expected traveling time for the remaining paths (B,D) and (C,E,F,G) depending on the realized time at the edge A.

shown in Figure 7.5. Again it can be seen that curves for the PHDs of different orders differ slightly for small traveling times at the edge A and are very similar for

larger traveling times at A.

Figure 7.5 not only shows the effect of positive correlation for relatively small weights of the edge A but the effect of the PHD order and thus the maximal achieved correlation is visualized.

One can see that the minimum of the expected traveling time for the remaining route (B,D) is reached using the PHD of order 15 with the highest correlation $\rho_{AB} = 0.19754$.

In the case of large weights for the edge A the PHDs with a higher correlation again result in curves which have a large slope, such that the effect of correlation becomes more visible. As shown in Figure 7.5 the effect is most obvious for the largest PHDs with 15 states but is also significant for the smaller PHDs, e.g., with 4 phases.

Maximizing probability of arriving on time In this example we determine the path maximizing the probability of reaching the destination $v_{fin} = 4$ within a given deadline value w as explained in [40]. We analyzed the corresponding CTMDP in the time interval $[0,w]$ for deadline $w \in [20,200]$. In particular, we used the discretization approach described in 5.6 with relatively small discretization step $h = w/N$, for $N = 3000$. The computed results are shown in Figure 7.6 for models with PHDs of different orders.

In the examples weights at all edges are described by PHDs of a common order. Here the positive correlation has an effect of a higher probability for the path $p_2 = (A, B, D)$ to meet a short deadline but in a slightly smaller probability to meet a long deadline. E.g. for the model using PHDs of order 15 with the highest correlation, the probability of meeting a deadline decreases already for values $w > 70$.

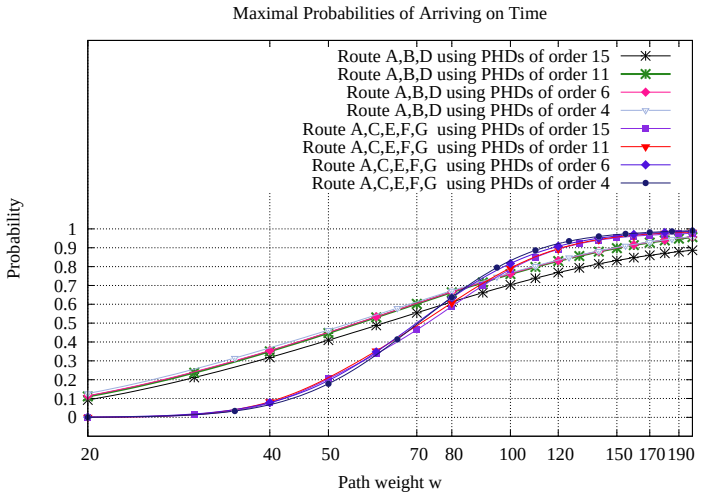


Figure 7.6.: The probabilities of arriving at the destination $v_{fin} = 4$ with a path weight less or equal to $w \in [20,200]$.

But if one starts just before a deadline, the most uncertain route offers a higher probability of avoiding lateness. In this case, the correlated path containing B should be preferred to the more reliable path containing edge C. From Figure 7.6 it can be seen that there is only a small difference between the results of the examples using PHDs of different orders.

The probabilities of meeting various deadlines via the path (B,D) and (C,E,F,G) are analyzed based on realized values at edge A and remaining time to deadline. History vectors $\bar{\psi}_{(A,w_A,i_J,0)}$ with $w_A \in [0,30]$, $J \in \{B,C\}$ have been evaluated. Results are summarized graphically in Figure 7.7 from [40].

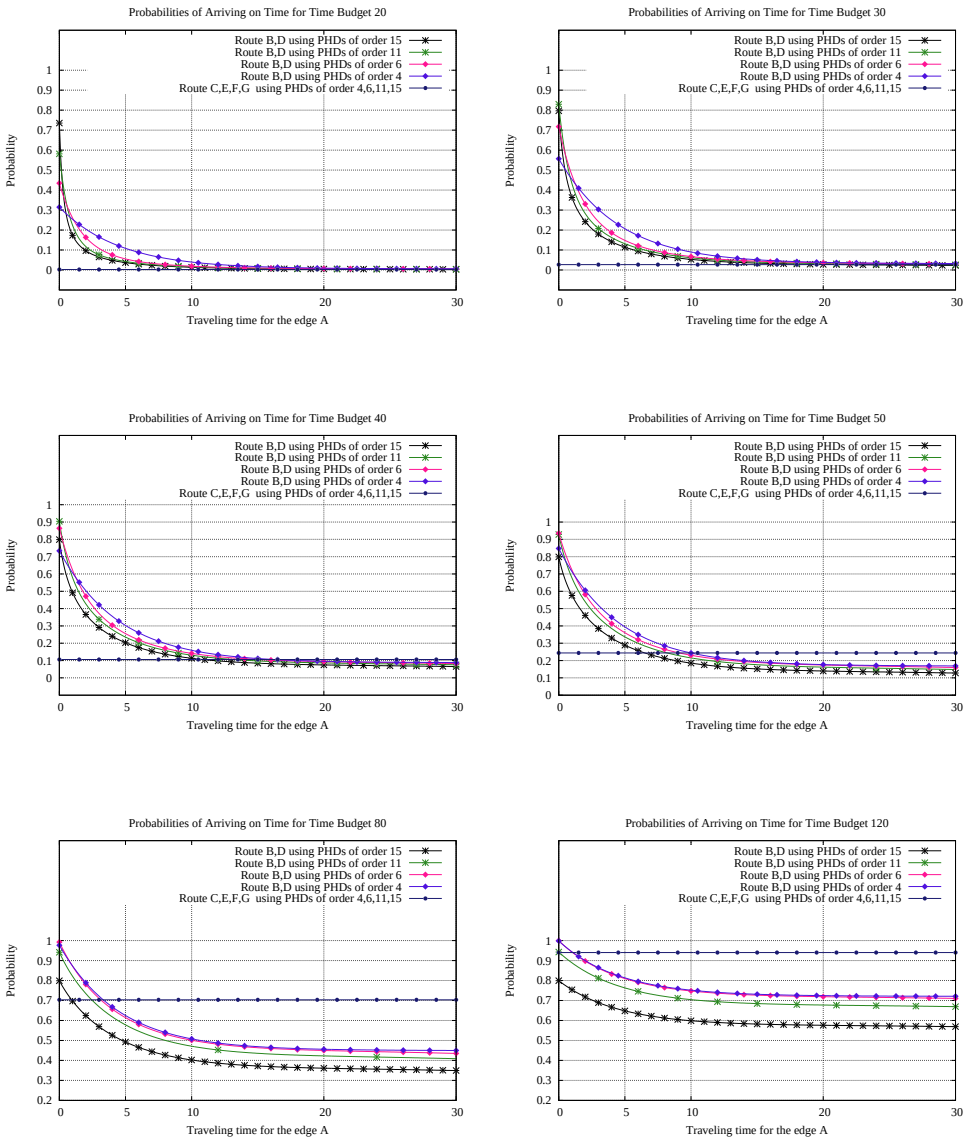


Figure 7.7.: Probabilities of arriving on time for different deadlines depending on the realized traveling time at the edge A from [40].

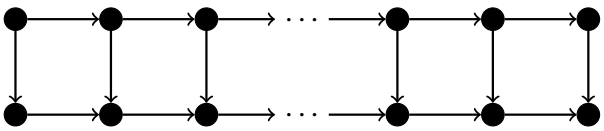


Figure 7.8.: Acyclic grid graph for shortest path computation.

It can be seen that for a short traveling time at A the correlated path (B, D) should be chosen by a vehicle. As the deadline increases, the traveling time at A for which it is preferable to choose the path (B, D) rather than (C, E, F, G) becomes smaller and smaller. The reason for this behavior is the relatively large variance of the traveling time of edge B.

Furthermore, one can see that for a long time at A the uncorrelated and more reliable path (C, E, F, G) offers a higher probability of arriving on time for deadline values $w = 50, 80, 120$. One can observe that behavior even for small weights at A, but for larger weights at A the effect intensifies. In that case one should prefer the path via C, since traveling times at A and B are positively correlated.

It can also be seen that the probabilities computed using PHDs of order 4 with the lowest correlation are rather conservative. Since the results computed using PHDs of higher order, e.g., order 15 which results in the highest correlation, are more sensitive in respect to traveling time at A and time horizon.

7.2. Numerical investigation

Computational times of the proposed algorithms have to be moderate, if the algorithms should be used in real-time applications. Then, the algorithms should be able to meet timing requirements of different real-time applications which is typically in the order of several seconds to several minutes. In order to investigate the effort of the shortest path computation we first generated three stochastic graph instances with different number of states. Then we ran the algorithms to analyze the computational effort needed to obtain the optimal policy and the optimal gain vector. All of the computations were carried out on a PC with a 3.6 GHz Octa-Core processor and 16 GB main memory. The shortest path algorithms are implemented in matlab and C.

The grid graph The first example graph shown in Fig. 7.8 is a two-dimensional grid graph containing paths on N vertices. An initial node is on the top left corner and the final node is on the bottom right corner. Upper nodes allow choices between two outgoing edges. Lower nodes have only one outgoing edge. The grid graph with N nodes has $\frac{N}{2} + 2(N-1)$ edges and paths with $\frac{N}{2}$ edges. The weights of the edges are modeled by the following PHD

$$\pi = (0.5, 0.5), \mathbf{D}_1 = \begin{pmatrix} -7 & 0 \\ 0 & -0.5 \end{pmatrix}.$$

Similarly to the experimental setup described in [40] transfer matrices are generated from a convex linear combination of the transfer matrix for the maximal and minimal

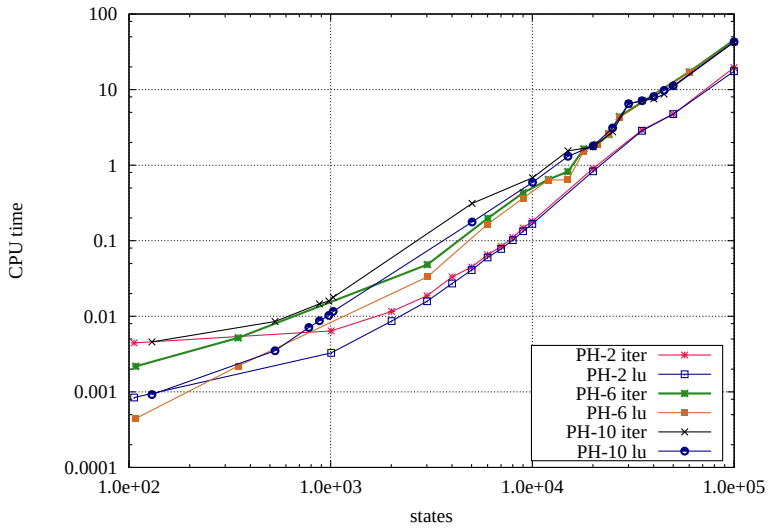


Figure 7.9.: Computational effort to compute the optimal policy in the grid graph.

correlation between adjacent edges. The following transfer matrices $\mathbf{H}$ result in the maximal correlation and minimal correlation ($\rho^{max} = 0.3$ and $\rho^{min} = 0$, respectively).

$$\mathbf{H}^{\rho^{max}=0.3} = \begin{pmatrix} 7 & 0 \\ 0 & 0.5 \end{pmatrix}, \mathbf{H}^{\rho^{min}=0} = \begin{pmatrix} 3.5 & 3.5 \\ 0.25 & 0.25 \end{pmatrix}.$$

For every pair of adjacent edges weights for the convex combination are selected randomly resulting in random transfer matrices. We used PHDs with 2, 6 and 10 phases to describe the distribution for the edge weights. Using PHDs of order n_i for the weights of the edges, the resulting CTMDP has $n_i \cdot (\frac{5N}{2} - 2)$ states.

To obtain the optimal policy minimizing the expected path length we inserted up to 10000 edges. However, for the acyclic grid graph computation times are moderate even for large CTMDP with 100000 states. Computing times resulting from the Algorithm 5.1 are expressed in seconds and are shown in Fig. 7.9.

The policy iteration method implemented in the Algorithm 5.1 iterates between policy evaluation and policy improvement step. In each iteration during the policy evaluation step a system of linear equations has to be solved. The effort of computing a solution for larger systems of linear equations quickly increases since solving a linear system requires computational time cubic in the number of states, e.g., if direct methods like LU-decomposition are used [40]. Experiments indicate that the number of iterations which policy iteration requires to determine the optimal solution does not depend on the number of states and policies. In this example, policy iteration requires less than 10 iterations to find the optimal shortest path.

We considered policy iteration using LU-decomposition and GMRES as iterative method for solving the relevant system of linear equations. Particularly, we applied matlab implementation of GMRES with an ILU0 preconditioner [153, 40]. Both methods are very fast for the equations occurring from the acyclic PH-Graph. If the PH-

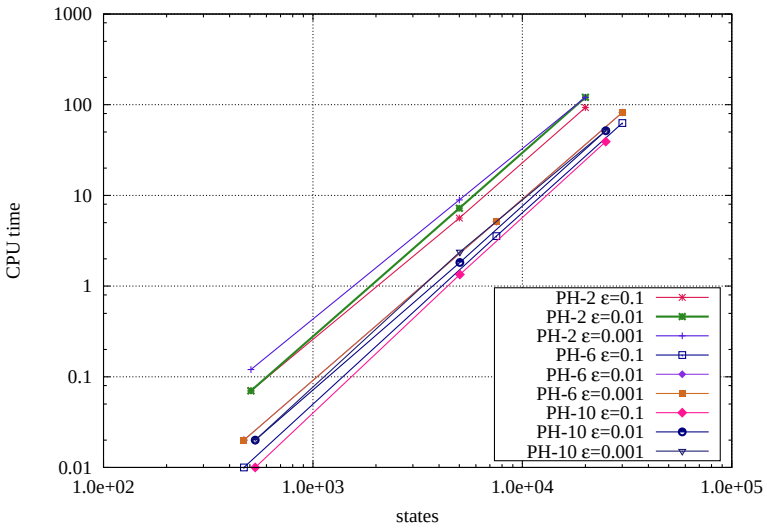


Figure 7.10.: Computational effort to compute the optimal policy maximizing the probability of reaching the destination state in less than the expected traveling time in the grid graph.

Graph is acyclic the matrix of linear equations has the properties of a sparse matrix (band diagonal matrix). In that case, the solution obtained using LU-decomposition methods (forward and back substitution) can be performed much faster than in the general case, and the whole solution vector can be determined very concisely [79, 104, 150]. In graphs with many cycles optimal policies can be determined very efficienly, although resulting in a higher computational effort as in the acyclic case.

The second problem under investigation is the computation of the maximal probability of arriving on time. The optimal policy and the maximal probability can be determined using the discretization algorithm 5.4 and the uniformization approach 5.3 which is much more efficient than the discretization technique. Furthermore the results computed by uniformization are much more accurate than those of the discretization approach. The main drawback of discretization is that the length of the discretization step in order to reach some predefined accuracy ε cannot be determined.

In the discretization algorithm, discrete time steps are used and the effort depends linearly on the inverse of the length of the discretization step [40]. Often discretization needs much more iterations than uniformization to reach the required accuracy. Since there are practical instances where the difference between the number of required iterations is a factor of 1000 [42], we computed the results using the uniformization 5.3. The computational effort of the uniformization depends on the number of states of the CTMDP, the required accuracy, time bound and the number of phases of the PHD.

The required time to compute the optimal policy maximizing the probability of arriving on time in the grid graph is shown in Figures 7.10, 7.11. For the larger number of states in the CTMDP more computational effort is required since the expected path

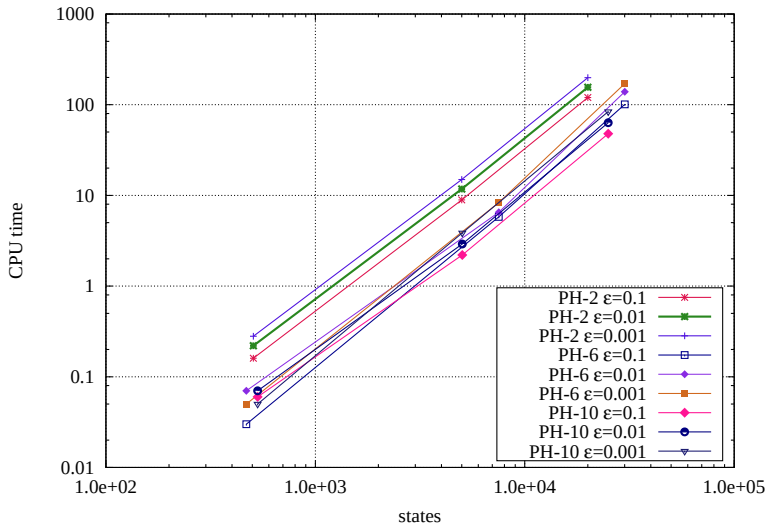


Figure 7.11.: Computational effort to compute the optimal policy maximizing the probability of reaching the destination state in less than the time bound larger than the expected traveling time in the grid graph.

weight increases. Generally, the transient analysis of finite horizon CTMDPs requires more computational time than the computation of the expected weight in infinite CTMDPs [40].

The level network As next example we considered the graph shown in Fig. 7.12 with N levels containing three nodes at each level. The initial node is the leftmost node in the graph, and the final node is the rightmost node. Each node except the final node has three outgoing edges. The weights of the edges and the transfer matrices are modeled as described above using PHDs of order 2, 6 and 10. We analyzed the computational effort of the Algorithm 5.1 for the graph with a growing number of levels. The required time to compute the optimal solution is shown in Fig. 7.13. Again, one can see that for the acyclic graph example the optimal policies can be computed efficiently, even if the policy iteration has to deal with an exponential number of policies.

We analyzed the computational effort required to compute the optimal policy maximizing the probability of arriving on time in the level graph with an increasing number

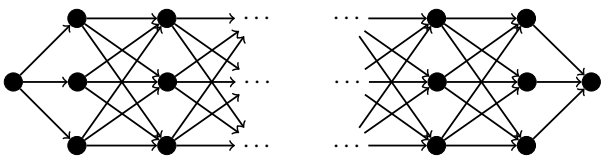


Figure 7.12.: Acyclic level graph for shortest path computation.

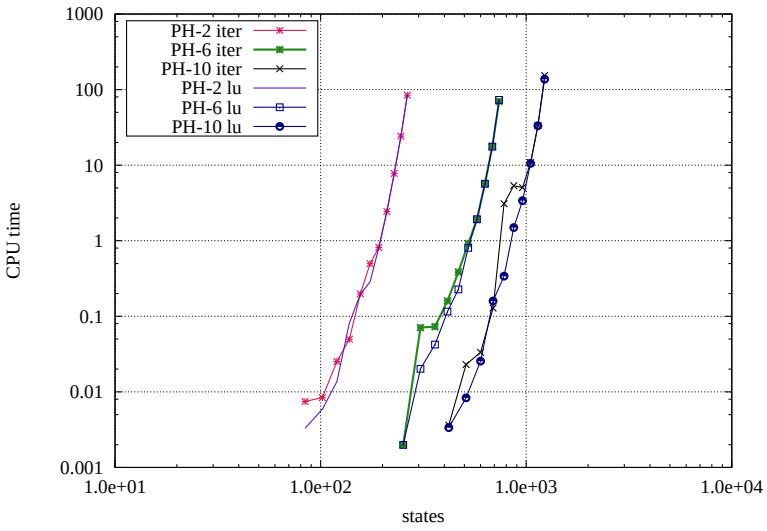


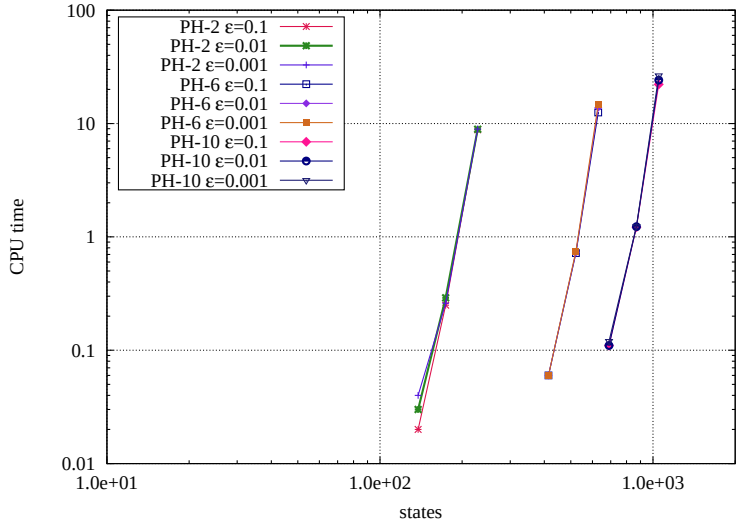
Figure 7.13.: Computational effort to compute shortest paths in the level graph shown in Fig. 7.12 with a growing number of levels.

of levels. The results in Fig. 7.14 show the effort for PHDs of order 2, 6 and 10 describing the weights of the edges and for different choices of ε. One can see that for the larger number of states in the CTMDP more effort is required. The reason is that for a larger number of states in the CTMDP the number of levels in the graph and also the expected time to reach the destination node increases.

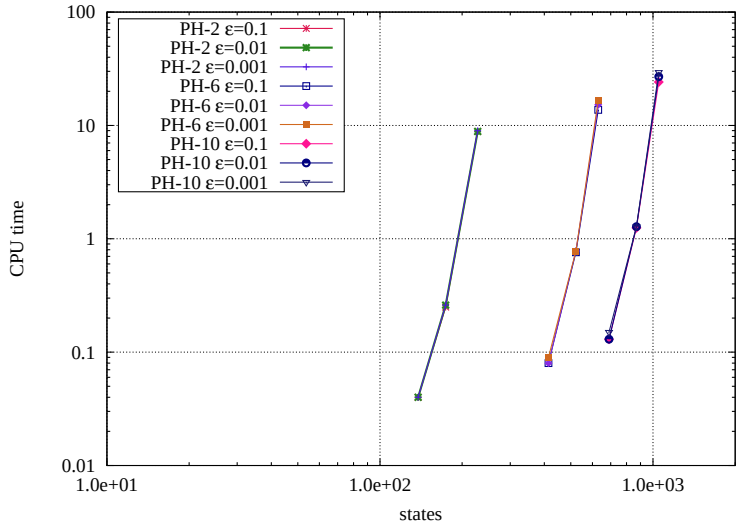
The quadtree model In [67] D. Eisenstat proposed a novel model of random road networks based on quadtrees. The specified generic model has the properties of planar and multiscale-disperse graphs still remaining realistic enough to model real road networks (see Fig. 7.15). Further features like simplicity, realistic variations in density of road graphs, self-similarity, avoidance of unrealistic abundance of intersections are also presented [67, 102].

The quadtree model is embedded in the plane. In order to resemble real road networks, initial square is divided into n squares of equal size. The procedure is repeated recursively by selecting some squares and subdividing them according to some chosen parameters [67, 102], e.g., distribution of road intersections and amount of sprawl. Each division of a square inserts additional roads and intersections as lines correspond to roads and line-crossings correspond to intersections [102].

Figure 7.16 (a) shows a road network resulting from subdiving two of four squares twice. A network resulting in further subdiving of six squares is shown in Figure 7.16 (b). The initial node is the leftmost top node, and the final node is the rightmost bottom node in the graph. Edges are directed from top to bottom and from left to right. We assume that each road is modeled by an edge and has unit capacity. In the quadtree shown in Figure 7.16 (a) each path from the initial node to the destination node con-



(a)



(b)

Figure 7.14.: Computational times required to compute the optimal policy in the level network maximizing the probability of reaching the destination within a time interval limited by the expected traveling time and within a larger time interval.

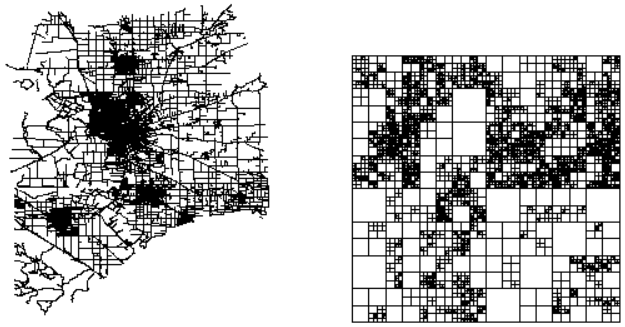
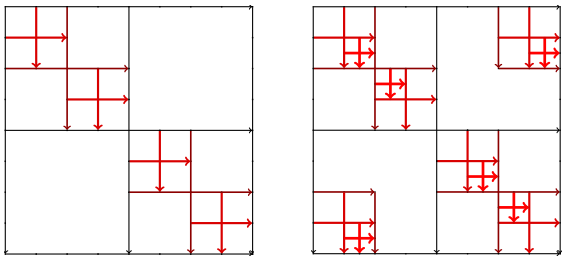


Figure 7.15.: The roads of San Joaquin County, California and the quadtree model represented in [67].



(a) (b)

Figure 7.16.: Versions of our quadtree models for road placement. Line color and thickness correspond to higher speed limits and traffic capacities.

tains 16 edges. In the quadtree shown in Figure 7.16 (b) each path from the initial node to the final node has 32 edges.

It is also possible to specify speeds or capacities for roads based on the depth of each edge in the quadtree model. In [102] it is assumed that roads are classified according to their capacities and speed limits. These are highways, high roads, minor and peripheral roads. In the context of urban shortest path computation the hierarchy of roads plays an important role. Often the driver starts his route in some peripheral street, e.g., corresponding to his home location. Then the driver moves to minor roads which are usually faster and larger. Reaching the fastest road the driver covers it until he again has to change to some secondary and peripheral roads. The quadtree model allow for hierarchichal structures of urban shortest paths.

In order to represent the described structure of real journeys we assign lower speed limits to the roads with a lower depth and higher speed limits to the roads with a higher depth on the quadtree. For our purposes, the weights of the edges on the quadtree are classified according to the first moment of the PHD representing the mean traveling time on the edge.

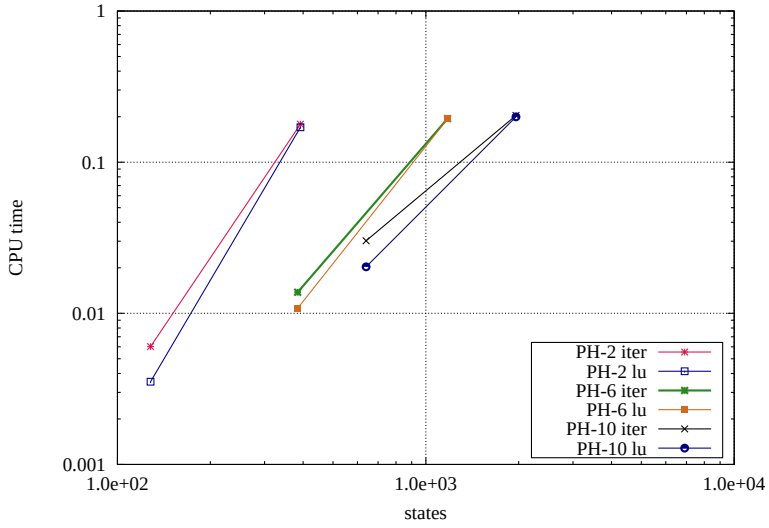
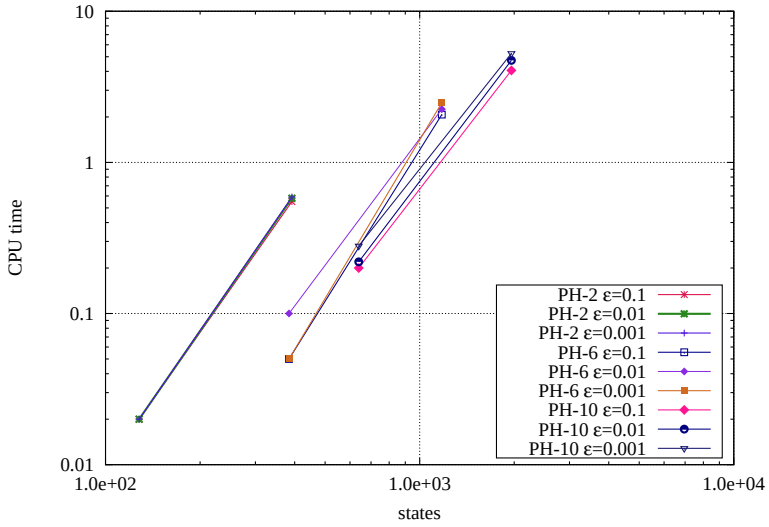


Figure 7.17.: Computational effort to compute the optimal policies in the quadtrees shown in Fig 7.16.

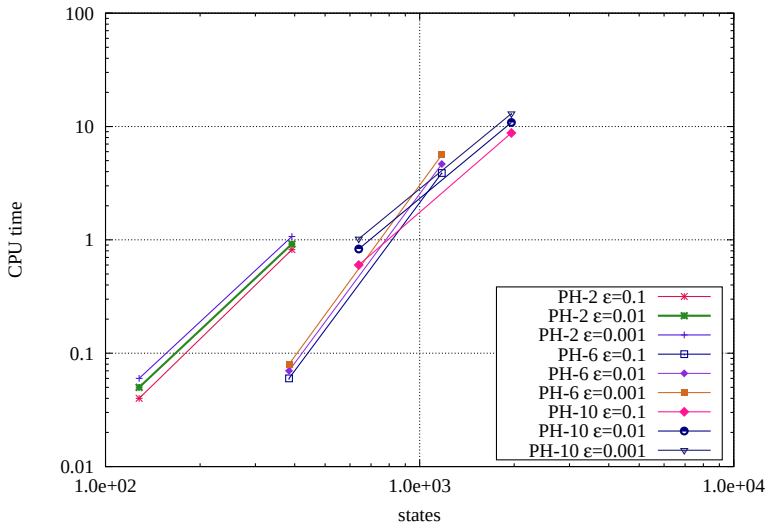
Weights of edges with the lower depth of the quadtree are modeled by PHDs with the first moment $\mu_1 = 5.35$ and $\mu_1 = 7$. Weights of edges with a higher depth on the quadtree are modeled by PHDs with the first moment $\mu_1 = 1$ and $\mu_1 = 3.75$. The transfer matrices are generated randomly as described in the above example. The computational effort of the Algorithm 5.1 to minimize the expected traveling time from the initial node to the destination node is analyzed for the two slightly different quadtrees shown in Fig. 7.16. The required computational times to determine the optimal solution is shown in Fig 7.17.

Furthermore, we analyzed the computational effort for the problem of maximizing the probability of reaching the final destination node within the time bound $[0,T]$. We computed results for the time bound limited by the expected traveling time and for the time bound at least twice as large as the expected traveling time. Fig. 7.18 show the computational time of uniformization for PHDs of order 2, 6 and 10 describing the weights of the edges in the quadtrees. For this small example the computational times of uniformization are very fast. The results show that the optimal policies in quadtrees with paths containing up to 32 edges can be analyzed within 10 s.

The quadtree model can be used to generate road networks with desired topological features. Both quadtrees shown in Fig. 7.16 specify a typical road network with different speeds and capacities. In this small example, the optimal policies could be determined very quickly. The obtained results of our experiments show that the developed methods allow one to analyze shortest paths in fairly large graph instances. Large state spaces and large policy spaces can be handled in a moderate time, often requiring a time of a few minutes to compute the optimal policies for different shortest path problems. In particular, small road networks can be analyzed very quickly indicating



(a)



(b)

Figure 7.18.: Computational times required to compute the optimal policy maximizing the probability of reaching the destination within a time interval limited by the expected traveling time and within a larger time interval.

that the PHG methods can be well applied in online navigation scenarios in cities.

7.3. PH-Graphs including negative edge weights

Bilateral PHDs have support on $(-\infty, \infty)$ and can be efficiently applied to PH-Graphs, such that negative edge weights can be incorporated into the model.

Currently, there are several open research questions on the field of BPH distributions (see [4], [95] and references therein). For instance, trace-based fitting methods for BPHDs with continuity behavior at zero. However, the existing moment based fitting method determines the parameters of acyclic BPHD using the first three moments under the assumption that there is no probability mass at zero [95]. The authors presented closed form solutions for ABPH(n^-, n^+), bounds for the first three moments, and a fitting procedure using the first three moments of the original distribution.

Here ABPH(n^-, n^+) denotes an acyclic BPH distribution, where n^- is the number of phases with negative rate and n^+ is the number of phases with positive rate. For example, using the approach in [95] we can determine the parameters α, λ^-, and λ^+ of ABPH(1, 1) given μ_1, μ_2, μ_3, where α is the mixing probability and λ^-, $\lambda^+ \geq 0$ (see Sec. 2.4.3).

Interestingly, a single negative phase already permits the whole flexibility for the first three moments in case of ABPH(n^-, n^+) with $n^- \geq 1$ and a positive first moment as shown in [95].

Note that n transient states are partitioned as $\mathscr{S}_T = \mathscr{S}_1 \cup \mathscr{S}_2$, where $\mathscr{S}_1$ is a set of positive phases and $\mathscr{S}_2$ is a set of negative phases. The Markov modulated reward process $B = \{B(t) : t \geq 0\}$ gives the total accumulated reward of the Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ in the time interval $[0,t)$ [95]. During the sojourn time of the underlying Markov process in positive phase i the accumulated reward increases with rate $c_i > 0$. Analogously, during the sojourn time of the Markov process in negative phase the accumulated reward decreases with rate $c_j > 0$.

In the following we adapt analysis methods for PH-Graphs with ABPH distributed edge weights. We consider ABPH distributions with support on the entire line $(-\infty, \infty)$ with no probability mass at zero. Furthermore the BPH distributions considered are constructed as the mixture of a positive and a negative PHD (see Fig. 2.20).

As mentioned in Sec. 2.4.3 a BPH has the representation $(\pi, \mathbf{D}_0, \mathbf{C}^*)$ where $\mathbf{C}^*$ is a nonsingular diagonal matrix with reward rates. Let the subgenerator path matrix $\mathbf{Q}_{(i_1,\ldots,i_K)}$ containing BPH subgenerator matrices $\mathbf{D}_0$ on its diagonal as defined in Eq. 3.17. Then the ith moment of the weight of the path is given by

$$\mu^i_{(i_1,\ldots,i_K)} = i!\pi \left(\mathbf{M}_{(i_1,\ldots,i_K)} \mathbf{C}^*_{(i_1,\ldots,i_K)} \right)^i \mathbb{1}, \tag{7.1}$$

with $\mathbf{C}^*_{(i_1,\ldots,i_K)} = \mathbf{I}^*$, where $\mathbf{I}^*$ is a diagonal matrix containing $\mathbf{I}^*(i,i) = 1$ for a positive phase $i \in \mathscr{S}_1$ and $\mathbf{I}^*(j,j) = -1$ for a negative phase $j \in \mathscr{S}_2$. Matrix $\mathbf{C}^*_{(i_1,\ldots,i_K)}$ is of dimension $\sum_{k=1}^K n_{i_k} \times \sum_{k=1}^K n_{i_k}$, i.e. it contains rates for all phases of BPHDs corresponding to the edges along the path. Moment matrix $\mathbf{M}_{(i_1,\ldots,i_K)} = (-\mathbf{Q}_{(i_1,\ldots,i_K)})^{-1}$ and the initial distribution vector π are defined in Def. 3.2.

Furthermore let $\mathbf{P}_{(i_1,\ldots,i_K)} = \mathbf{Q}_{(i_1,\ldots,i_K)}/\alpha + \mathbf{I}$ be the transition probability matrix of the corresponding embedded Markov process. According to Eq. 5.12 the fundamental

matrix of the absorbing DTMC is computed as $\mathbf{N}_{(i_1,\ldots,i_K)} = (\mathbf{I} - \mathbf{P}_{(i_1,\ldots,i_K)})^{-1}$. The fundamental matrix of the absorbing DTMC incorporating BPHDs is then computed as (cf. Eq. 5.12)

$$\mathbf{N}_{(i_1,\ldots,i_K)} = \mathbf{M}_{(i_1,\ldots,i_K)} \mathbf{C}^*_{(i_1,\ldots,i_K)} \alpha. \tag{7.2}$$

To compute the history vector ψ for accumulated weights along the subpath $(i_1,\ldots,i_l)$ the absolute values of realizations $(|w_1|,\ldots,|w_l|)$ should be considered. We obtain the history vector as (cf. Eq. 5.14)

$$\psi_{(i_1,|w_1|,\ldots,i_l,|w_l|)} = \pi_{i_1} \left(\prod_{k=1}^{l-1} e^{\mathbf{D}_{i_k}|w_k|} \mathbf{H}_{i_k\, i_{k+1}} \right) e^{\mathbf{D}_{i_l}|w_l|}. \tag{7.3}$$

In particular, the conditional moments of the remaining path $(i_{l+1},\ldots,i_K)$ can be computed as

$$\mu^i_{(i_{l+1},\ldots,i_K)} = i! \pi' \left(\mathbf{M}_{(i_{l+1},\ldots,i_K)} \mathbf{C}^*_{(i_{l+1},\ldots,i_K)} \right)^i \mathbb{1}, \tag{7.4}$$

where $\pi' = (\bar{\psi}_{(i_1,|w_1|,\ldots,i_l,|w_l|)}, \mathbf{0})$, $\bar{\psi}_{(i_1,|w_1|,\ldots,i_l,|w_l|)}$ is the normalized history vector defined in (5.16). Moment matrix $\mathbf{M}_{(i_{l+1},\ldots,i_K)}$ of the remaining path $(i_{l+1},\ldots,i_K)$ is computed using the subgenerator

$$\mathbf{Q}_{(i_{l+1},\ldots,i_K)} = \begin{pmatrix} \mathbf{D}_{i_l} & \mathbf{H}_{i_l,i_{l+1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{i_{l+1}} & \mathbf{H}_{i_{l+1}i_{l+2}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} \\ \vdots & & \ddots & \mathbf{D}_{i_{K-1}} & \mathbf{H}_{i_{K-1}i_K} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{D}_K \end{pmatrix}$$

as $\mathbf{M}_{(i_{l+1},\ldots,i_K)} = \left(-\mathbf{Q}_{(i_{l+1},\ldots,i_K)}\right)^{-1}$.

The total expected reward function for the proper policy $\mathbf{u}$ is obtained as (cf. Eq. 5.36, Eq. 5.38)

$$\mathbf{g}^{\mathbf{u}} = (\mathbf{I} - \mathbf{P}^{\mathbf{u}})^{-1} \mathbf{r}^{\mathbf{u}} = (\mathbf{I} - \mathbf{P}^{\mathbf{u}})^{-1} \mathbf{C}^* \mathbb{1}, \tag{7.5}$$

such that Eq. 7.5 is used in policy evaluation step 4 of the PI algorithm 5.1 rather than Eq. 5.43.

In the kth policy improvement step the improved policy $\mathbf{u}_{k+1}$ satisfying the equation $T_{\mathbf{u}_{k+1}} \mathbf{g}^{\mathbf{u}_k} = T\, \mathbf{g}^{\mathbf{u}_k}$ is obtained as (see Eq. 2.36 and Eq. 5.44)

$$\mathbf{u}_{k+1}(i,x) = \underset{u \in \mathscr{D}(i)}{\arg\min} \left(\mathbf{r}(i,x) + \sum_{(j,y) \in \mathscr{S}} \mathbf{P}^u((i,x),(j,y))\, \mathbf{g}^{\mathbf{u}_k}(j,y) \right), \tag{7.6}$$

where $\mathbf{r} = \mathbf{C}^* \mathbb{1}$. With these ingredients standard methods for solving MDPs can be applied to solve SSPP resulting from PH-Graph model with BPHDs.

***Example* 7.1.** *We consider the graph instance visualized in Fig. 7.19. The weights of all edges are described by the following ABPH(*1, 1*)*

$$\pi = (0.5, 0.5), \mathbf{D}_1 = \begin{pmatrix} -7 & 0 \\ 0 & -0.5 \end{pmatrix}, \mathbf{C}^* = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix},$$

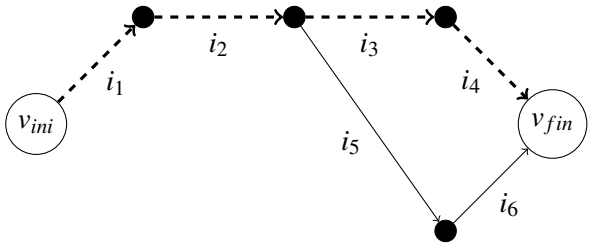


Figure 7.19.: Instance of the acyclic graph for shortest path computation. Edges with correlated weights are highlighted in dashed style.

which is the convex mixture of two PHDs. The parameter of the negative part is $\lambda^- = 7$*, of the positive part is* $\lambda^+ = 0.5$ *and* π *is the mixing probability vector. The first moment of ABPH is given by*

$$\mu^1 = (-1)\pi(1)\frac{1}{\lambda^-} + \pi(2)\frac{1}{\lambda^+} = -0.0714 + 1 = 0.9286.$$

The mean weight of every path $p \in Paths$ *from* v_{ini} *to* v_{fin} *is* 3.7143*. The weights of the edges along the upper path are positively correlated with correlation coefficient* $\rho = 0.3$*. The weights of the edges along the lower path* (i_1, i_2, i_5)*,* i_6 *are uncorrelated. The following transfer matrices* $\mathbf{H}$ *result in mentioned correlation coefficients* $\rho = 0.3$ *and* $\rho = 0$

$$\mathbf{H}^{\rho=0.3}_{i_k i_l} = \begin{pmatrix} 7 & 0 \\ 0 & 0.5 \end{pmatrix}, \mathbf{H}^{\rho=0}_{i_s i_t} = \begin{pmatrix} 3.5 & 3.5 \\ 0.25 & 0.25 \end{pmatrix},$$

for $k = 1,2,3$*,* $l = 2,3,4$*,* $s = 2,5$*, and* $t = 5,6$*. Furthermore, we have two path matrices*

$$\mathbf{Q}_{(i_1,i_2,i_3,i_4)} = \begin{pmatrix} -7 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 0 & 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{pmatrix},$$

$$\mathbf{Q}_{(i_1,i_2,i_5,i_6)} = \begin{pmatrix} -7 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & -7 & 0 & 3.5 & 3.5 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0.25 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 3.5 & 3.5 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & -7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 \end{pmatrix},$$

and reward matrix

$$\mathbf{C}^*_{(i_1,i_2,i_k,i_l)} = \begin{pmatrix} \mathbf{C}^* & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}^* & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C}^* \end{pmatrix},$$

for $k = 3, 5$, $l = 4, 6$ where $\mathbf{0}$ is a 2×2 matrix with zeros.

Conditional moment computation *Suppose now that the realization of weight w of the edge i_2 is known to the process. We consider the cases with $w = -0.2$ and $w = 1$. The computed history vectors are presented in Tab. 7.3. Fig. 7.20 shows values*

Table 7.3.: Realizations of weight w of the edge i_2 and the corresponding history vectors. Here $\rho_{i_2\, i_3} = 0.3$ and $\rho_{i_2\, i_5} = 0$.

Successor edge	$w = -0.2$	$w = 1$
i_3	$\bar{\Psi}_{(i_2,\, 0.2,\, i_3,\, 0)} = (0.7924,\, 0.2076)$	$\bar{\Psi}_{(i_2,\, 1,\, i_3,\, 0)} = (0.0206,\, 0.9794)$
i_5	$\bar{\Psi}_{(i_2,\, 0.2,\, i_5,\, 0)} = (0.5,\, 0.5)$	$\bar{\Psi}_{(i_2,\, 1,\, i_5,\, 0)} = (0.5,\, 0.5)$

of the first conditional moment for the remaining paths (i_3, i_4), (i_5, i_6).

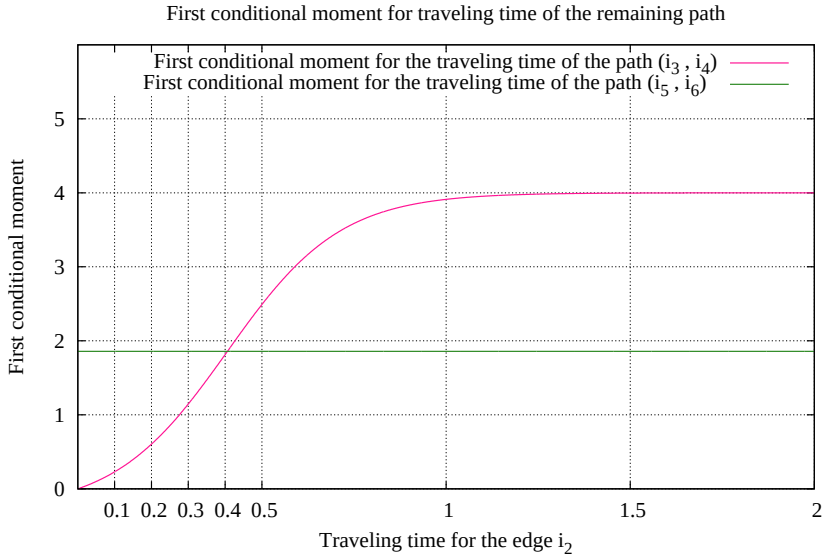


Figure 7.20.: The expected weight of the paths (i_3, i_4), and (i_5, i_6) depending on the absolute value of the weight of i_2.

Policy iteration *Given the set of transition rates from subgenerators $\mathbf{Q}_{(i_1,i_2,i_3,i_4)}$ and $\mathbf{Q}_{(i_1,i_2,i_5,i_6)}$ we transform the CTMPD into DTMDP using uniformization rate $\alpha = 7$. The expected rewards have the following values*

$$\mathbf{r}^u(i,x) = \begin{cases} -1 & \text{for } (i,1) \in \mathscr{S}_T,\ u \in \mathscr{D}(i), \\ 1 & \text{for } (i,2) \in \mathscr{S}_T,\ u \in \mathscr{D}(i). \end{cases}$$

The policy for the path with correlated edge weights $(i_1,\, i_2,\, i_3,\, i_4)$ *results in the following gain vector*

	$(i_1,1)$	$(i_1,2)$	$(i_2,1)$	$(i_2,2)$	$(i_3,1)$	$(i_3,2)$	$(i_4,1)$	$(i_4,2)$
$\mathbf{g}^{u_0}(i_k,x)$	-0.5714	8	-0.4286	6	-0.2857	4	-0.1429	2

The weight of the policy for the uncorrelated lower path $(i_1,\, i_2,\, i_5,\, i_6)$ *is given by the following gain vector*

	$(i_1,1)$	$(i_1,2)$	$(i_2,1)$	$(i_2,2)$	$(i_5,1)$	$(i_5,2)$	$(i_6,1)$	$(i_6,2)$
$\mathbf{g}^{u_0}(i_k,x)$	1.5714	5.8571	1.7143	3.8571	0.7857	2.9286	-0.1429	2

In the policy improvement step the values $\mathbf{u}_1(i_2,1)$, $\mathbf{u}_1(i_2,2)$ *satisfying* $T_{\mathbf{u}_1}\mathbf{g}^{\mathbf{u}_0} = T\,\mathbf{g}^{\mathbf{u}_0}$ *are computed as*

$$
\begin{aligned}
\mathbf{u}_1(i_2,1) &= \underset{u\in\{i_3,\,i_5\}}{\arg\min}\left\{-1+\sum_{j=1}^{n_{i_2}+n_{i_3}}\mathbf{P}^{u}((i_2,1),j)\,\mathbf{g}^{\mathbf{u}_0}(j)\right\}\\
&= \underset{i_3,i_5}{\arg\min}\,\{-1+\mathbf{P}_{i_1,i_2,i_3,i_4}((i_2,1),(i_3,1))\,\mathbf{g}^{\mathbf{u}_0}(i_3,1),\\
&\qquad -1+\mathbf{P}_{i_1,i_2,i_5,i_6}((i_2,1),(i_5,1))\,\mathbf{g}^{\mathbf{u}_0}(i_5,1)+\\
&\qquad \mathbf{P}_{i_1,i_2,i_5,i_6}((i_2,1),(i_5,2))\,\mathbf{g}^{\mathbf{u}_0}(i_5,2)\,\}\\
&= \underset{i_3,i_5}{\arg\min}\,\{-1+1\cdot(-0.2857),-1+0.5\cdot 0.7857+0.5\cdot 2.9286\,\}\\
&= \underset{i_3,i_5}{\arg\min}\,(-1.2857,\ 0.85715\,) = i_3,
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbf{u}_1(i_2,2) &= \underset{u\in\{i_3,\,i_5\}}{\arg\min}\left\{1+\sum_{j=1}^{n_{i_2}+n_{i_3}}\mathbf{P}^{u}((i_2,2),j)\,\mathbf{g}^{\mathbf{u}_0}(j)\right\}\\
&= \underset{i_3,i_5}{\arg\min}\,\{1+\mathbf{P}_{i_1,i_2,i_3,i_4}((i_2,2),(i_3,2))\,\mathbf{g}^{\mathbf{u}_0}(i_3,2)+\\
&\qquad \mathbf{P}_{i_1,i_2,i_3,i_4}((i_2,2),(i_2,2))\,\mathbf{g}^{\mathbf{u}_0}(i_2,2),\\
&\qquad 1+\mathbf{P}_{i_1,i_2,i_5,i_6}((i_2,2),(i_5,1))\,\mathbf{g}^{\mathbf{u}_0}(i_5,1)+\\
&\qquad \mathbf{P}_{i_1,i_2,i_5,i_6}((i_2,2),(i_5,2))\,\mathbf{g}^{\mathbf{u}_0}(i_5,2)+\\
&\qquad \mathbf{P}_{i_1,i_2,i_5,i_6}((i_2,2),(i_2,2))\,\mathbf{g}^{\mathbf{u}_0}(i_2,2)\,\}\\
&= \underset{i_3,i_5}{\arg\min}\,\{1+0.0714\cdot 4+0.9286\cdot 6,\\
&\qquad 1+0.0357\cdot 0.7857+0.0357\cdot 2.9286+0.9286\cdot 3.8571\,\}\\
&= \underset{i_3,i_5}{\arg\min}\,(6.8572,\ 3.7143\,) = i_5,
\end{aligned}
$$

where uniformized path matrices are given in Sec. D.2.

7.3.1. Financial Optimization under Uncertainty

This section is devoted to investment decision-making under uncertainty. In most practical situations, the knowledge about financial market parameters like investment

costs, taxes, stock prices, technical progress, political situations, technical or natural catastrophes, is incomplete or changes over time. Thus it is necessary to study how to plan investments in the uncertain financial market environment in order to maximize the gain.

An agent has to invest all his money into financial markets, e.g., an initial capital is invested in the risky assets and the riskless bonds. The problem is to allocate the money over available investments to maximize the total amount of money at the end of the time interval T. Suppose that his capital at time t consists only of the gains and losses which are accumulated over the time interval $[0,T]$ by trading into the assets. The aim is to maximize the expected value of his wealth at the end of time horizon T.

Let T_k and X_k for $k = 1,2,\ldots$ be random variables, where T_k describes the size of the kth loss (e.g., investment costs), and X_k giving the size of the kth income. Furthermore, let the gain process $\{G(t)\}_{t\geq 0}^{T}$ describing the wealth process (see also processes described in [13, 14, 15]). In most realistic situations, the $\{T_k\}_{k=1}^{T}$ and $\{X_k\}_{k=1}^{T}$ are not necessarily independent random variables, i.e., the random variables T_k and X_k could be possibly correlated for any $k = 1,2,\ldots$. Thus, PHGs can be well used for modeling the wealth process and optimization of the investment strategy.

Let $J(t)$ be the environmental Markov process with the partitioned state space $\mathscr{S} = \mathscr{S}_1 \cup \mathscr{S}_2$ as also described in [13, 15]. Let $c(i)$ be the reward rate assigned to the phase $i \in \mathscr{S}$. During the sojourn of $J(t)$ in phase $i \in \mathscr{S}_1$ the reward increases at rate $c(i)$. During the sojourn of $J(t)$ in phase $i \in \mathscr{S}_2$ the reward process decreases at rate $c(i)$, i.e., the rate $c(i)$ is negative for $i \in \mathscr{S}_2$ such that costs can be treated as negative income.

In the following we assume that $c(i) = c$ for all phases $i \in \mathscr{S}$. Let the kth loss T_k occur for $k = 1,2,\ldots$. Then the corresponding costs are paid out over a time interval of length T_k/c at rate c. In particular, let the uniform reward rate $c = 1$ be assigned to all $i \in \mathscr{S}$ and let τ_i be the sojourn time in phase i. Then the accumulated reward during the sojourn in phase i is $c\,\tau_i$ such that the loss of size T_k can be modeled by a BPHD. A sample path of the Markov process is shown in Fig. 7.21.

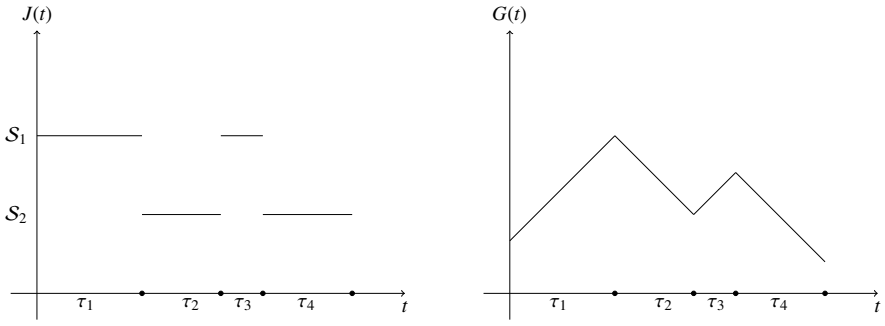


Figure 7.21.: Evolution of the environmental Markov process $J(t)$ and the associated wealth (reward) process $G(t)$ over time as visualized in [13, 15].

PHG for the investment model As mentioned above correlations between the random variables $\{T_k\}_{k=1}^{T}$ and $\{X_k\}_{k=1}^{T}$ cannot be neglected to decide how the agent

should invest in order to maximize his expected reward at the end of time horizon T?

Let

$$\left(\left(\boldsymbol{\pi}_{k-1}, \mathbf{D}_{k-1}\right), \left(\boldsymbol{\pi}_{k}, \mathbf{D}_{k}\right), \mathbf{H}_{(k-1)\,k} \right) \tag{7.7}$$

be the PHD composition in a PHG. Let PH_{k-1} be of order n_{k-1}, and PH_k be of order n_k. Here PH_{k-1} is the PHD of the random variable T_{k-1} giving the $(k-1)$th loss size, i.e., amount of money which is lost after investment in a risky asset. The PHD PH_k is the distribution of the subsequent gain size X_k, i.e., amount of money which is gained after investment in the risky asset (see Tab. 7.4). The transfer matrix $\mathbf{H}_{(k-1)\,k}$ describes possible correlation between the $k-1$th loss size and the subsequent kth gain size.

Table 7.4.: Interpretation of the PHD composition (7.7) in investment problem

Variable	Representation	Meaning
T_{k-1}	$PH_{k-1}\ (\boldsymbol{\pi}_{k-1}, \mathbf{D}_{k-1})$	costs/loss
X_k	$PH_k\ (\boldsymbol{\pi}_k, \mathbf{D}_k)$	income/profit

During the time period described by T_{k-1} the accumulated reward decreases at rate $c = 1$. During the time period described by X_k the accumulated reward increases at rate $c = 1$. The PHD composition describes the sum of PH distributed random variables $T_{k-1} + X_k$ which corresponds to the random surplus after the investment in a risky asset.

Accordingly, the PHD composition $((\boldsymbol{\pi}_k, \mathbf{D}_k),\ , (\boldsymbol{\pi}_{k+1}, \mathbf{D}_{k+1}), \mathbf{H}_{k\,(k+1)})$ describes the distribution of a pair $(X_k,\ T_{k+1})$. In that way the transfer matrix $\mathbf{H}_{k\,(k+1)}$ describes possible dependence between the kth gain size and the subsequent $(k+1)$th loss size.

Formally, in a given PHG edge weights are ABPH. The state space is partitioned as $\mathscr{S}_T = \mathscr{S}_1 \cup \mathscr{S}_2$, where $\mathscr{S}_1$ is a set of positive phases corresponding to PHDs describing gains X_k, and $\mathscr{S}_2$ is a set of negative phases corresponding to BPHDs describing costs or loss sizes T_ks.

Now an investment policy based on correlation between costs und incomes can be determined. The corresponding CTMDP contains a single absorbing state $(i_{fin}, 0)$ describing the end of investment activity. Then algorithms for solving CTMDPs can be applied to solve the *investment problem* resulting from the PHG described above.

In particular, a finite horizon CTMDP with maximum cumulative reward criteria is considered. The maximal cumulative reward is interpreted as maximal surplus after the investment in risky assets.

Maximization of cumulative reward requires fixing the initial gain vector to the zero vector, i.e., $\mathbf{g}_T = \mathbf{0}$. Then value iteration based on discretization approach can be applied to obtain the optimal policy which is piecewise constant [49, 42]. As described in Sec. 5.6, for sufficiently small discretization step $h > 0$ it holds that
$e^{h\,\mathbf{Q}^{\mathbf{u}}} = \mathbf{P}^{\mathbf{u}}_h + o(h^2)$, and the stochastic matrix $\mathbf{P}^{\mathbf{u}}_h$ is defined as

$$\mathbf{P}^{\mathbf{u}}_h = \mathbf{I} + h\,\mathbf{Q}^{\mathbf{u}},$$

which is the transition matrix of the DTMC induced by the decision vector $\mathbf{u}$. Given an appropriate h and for every policy $\mathbf{u}$ the matrices $\mathbf{P}^{\mathbf{u}}_h$ define a DTMDP. Knowing the gain vector $\mathbf{g}_{t,T}$ at time t and the decision vector $\mathbf{u}$ holding in a fixed interval $(t-h, t]$

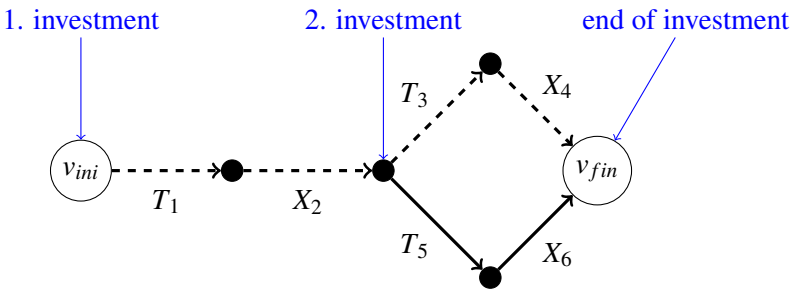


Figure 7.22.: The investor PHG for optimization of cumulative reward. Edges with correlated weights are highlighted in dashed style.

the approximation for the gain vector $\mathbf{g}_{t-h,T}$ at time $(t-h)$ can be obtained as

$$\mathbf{g}_{t-h,T} = \mathbf{P}_h^{\mathbf{u}}\,\mathbf{g}_{t,T} \;+\; h\,\mathbf{r}^{\mathbf{u}} + o(h).$$

For a finite horizon PHG the optimal policy maximizing the cumulative reward $\mathbf{g}_{0,T}$ can be determined using dynamic programming as shown in Algorithm 7.1.

Algorithm 7.1: Finite horizon dynamic programming for optimization of cumulative reward

1: Initialize $h = \frac{T}{N}$, $t = N$ and $\mathbf{g}^{\mathbf{u}_t} = (0,\ldots,0)$
2: for $t = N-1$ downto 0 do
3: for each $(i,x) \in \mathscr{S}$ determine new policy $\mathbf{u}_t$ as

$$\mathbf{u}_t(i,x) \in \arg\max_{u\in\mathscr{D}(i)} \left(h\mathbf{r}^u(i,x) + \sum_{(j,y)\in\mathscr{S}} \mathbf{P}_h^u((i,x),(j,y))\,\mathbf{g}^{\mathbf{u}_{t+1}}(j,y) \right)$$

4: compute the gain vector as

$$\mathbf{g}^{\mathbf{u}_t} = h\mathbf{r} + \mathbf{P}_h^{\mathbf{u}_t}\,\mathbf{g}^{\mathbf{u}_{t+1}}$$

5: terminate with $\mathbf{d} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_N)$ and $\mathbf{g}_{0,T}^{\mathbf{d}} = \mathbf{g}^{\mathbf{u}_0}$

The cumulative reward using policy $\mathbf{d}$ can be obtained as $\pi\mathbf{g}_{0,T}^{\mathbf{d}}$, where π is an initial distribution vector. The value $\mathbf{g}^{\mathbf{u}_t}(i,x)$ is the optimal expected cumulative reward starting from state (i,x) at time t, for $(i,x) \in \mathscr{S}$. The algorithm operates on CTMDP with $\mathbf{r} = \mathbf{C}^*\mathbb{1}$. The matrices $\mathbf{P}^{\mathbf{u}}$ can be computed as described above for all $\mathbf{u} \in \mathscr{P}$ and are used as an input.

***Example* 7.2.** *Suppose an agent has to invest his money in the risky assets over the time interval $[0,T]$. After his first investment in the risky asset he can choose between two available investments (denoted as two possible paths T_3X_4 and T_5X_6). This choice is visualized in a PHG given in Fig. 7.22.*

***Modeling**. The weights of all edges corresponding to costs and losses are described by a following 2-order hyperexponential PHD*

$$\pi_c = (0.5, 0.5), \mathbf{D}_c = \begin{pmatrix} -7 & 0 \\ 0 & -0.5 \end{pmatrix},$$

which has the mean 1.0714 *and the squared coefficient of variation* 2.5.

The weights of edges corresponding to an income/profit are given by a 2*-order hyperexponential PHD*

$$\pi_p = (0.5, 0.5), \mathbf{D}_p = \begin{pmatrix} -3.5 & 0 \\ 0 & -0.25 \end{pmatrix},$$

with the mean 2.1429 *and the squared coefficient of variation* 2.5.

The node corresponding to the second investment in Fig. 7.22 allows a choice between two outgoing edges. Thus, the investor has a choice between two risky assets.

Furthermore, the size of income X_2 and the size of loss T_3 are positively correlated whereas the size of income X_2 and the size of subsequent loss T_5 are uncorrelated. The transfer matrices are given as

$$\mathbf{H}_{i,j}^{\rho=0.3} = \begin{pmatrix} 7 & 0 \\ 0 & 0.5 \end{pmatrix}, \mathbf{H}_{5,6}^{\rho=0} = \begin{pmatrix} 3.5 & 3.5 \\ 0.25 & 0.25 \end{pmatrix},$$

for $(i, j) \in \{(1,2), (3,4)\}$*, and*

$$\mathbf{H}_{2,3}^{\rho=0.3} = \begin{pmatrix} 3.5 & 0 \\ 0 & 0.25 \end{pmatrix}, \mathbf{H}_{2,5}^{\rho=0} = \begin{pmatrix} 1.725 & 1.725 \\ 0.125 & 0.125 \end{pmatrix}.$$

The corresponding path matrices are modeled as

$$\mathbf{Q}_{T_1, X_2, T_3, X_4} = \begin{pmatrix} -7 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.5 & 0 & 3.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.25 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3.5 & 0 & 3.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$\mathbf{Q}_{T_1, X_2, T_5, X_6} = \begin{pmatrix} -7 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.5 & 0 & 1.725 & 1.725 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.25 & 0.125 & 0.125 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -7 & 0 & 3.5 & 3.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0.25 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -3.5 & 0 & 3.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where PHDs are highlighted in red and transfer matrices are highlighted in gray. The reward vector equals

$$\mathbf{r} = (-1, -1, \mathbf{1}, \mathbf{1}, -1, -1, \mathbf{1}, \mathbf{1}, 0).$$

Optimization *The PHG described can be analyzed using dynamic programming approach for different time horizons. Fig. 7.23 shows different values of cumulative rewards computed with Alg. 7.1.*

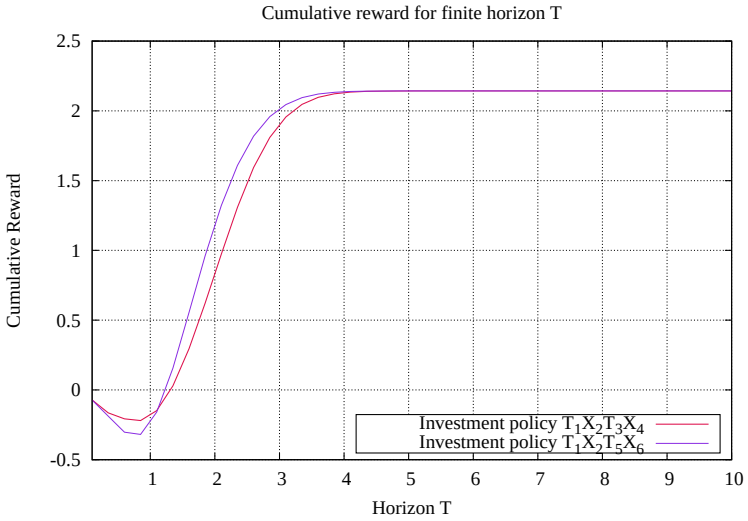


Figure 7.23.: Cumulative reward values of two investment policies $T_1 X_2 T_3 X_4$, $T_1 X_2 T_5 X_6$ and different time horizons.

After passing the edges T_1, X_2 the weights w_{T_1}, w_{X_2} become known to the investor such that he knows which income and costs are incurred. The path matrices describing remaining investment are

$$\mathbf{Q}_{T_3,\,X_4} = \begin{pmatrix} -7 & 0 & 7 & 0 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0 \\ 0 & 0 & -3.5 & 0 & 3.5 \\ 0 & 0 & 0 & -0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{Q}_{T_5,\,X_6} = \begin{pmatrix} -7 & 0 & 3.5 & 3.5 & 0 \\ 0 & -0.5 & 0.25 & 0.25 & 0 \\ 0 & 0 & -3.5 & 0 & 3.5 \\ 0 & 0 & 0 & -0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Arriving at node $fin(X_2)$ there are two competing options for the next possible investment, i.e., there are two possible edges to visit from $fin(X_2)$, namely T_3 and T_5. In any case, the investor's objective is to select the next edge to traverse such that the expected reward until arriving at the destination node v_{fin} is maximized. If he takes correlated costs on adjacent edges into consideration, the optimal policy has to be determined depending on the realized income w_{X_2} of the edge X_2.

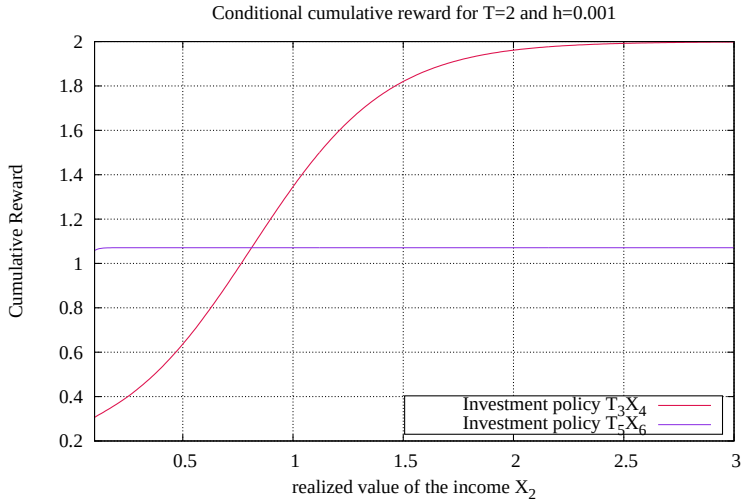


Figure 7.24.: The expected cumulative reward of subpaths T_3X_4 and T_5X_6 depending on the realized income at the edge X_2.

We computed vectors $\tilde{\psi}_{X_2,\,w,\,i_J,\,0}$ with $w \in [0.1, 3]$ and possible successor edges $J \in \{T_3, T_5\}$. Fig. 7.24 shows the values of the conditional expected cumulative reward for two different investment policies depending on the weight at the edge X_2.

Here, the weight of the remaining path $T_5\,X_6$ is not affected by the income at X_2 since weights of edges X_2 and T_5 are independent PHDs. The positive correlation between the weights at the edges X_2 and T_3 results in a positive slope of the expected reward for the remaining path T_3X_4. Thus, the investor should decide which strategy to follow in dependence of the realized income at X_2 in order to maximize his wealth.

7.4. PH-Graphs in Maintenance

We present here the analysis of maintenance models to show the practical applicability and efficiency of the proposed algorithms.

7.4.1. A Maintenance Model of the Power Module in Wind Turbines

This section describes a simple PH-Graph model to analyze maintenance costs of wind turbine components. In the final stage of an electronical component's life cycle the component may become difficult to repair and beyond economical repair. This is often due to the equipment design, e.g., failures can be caused by material corrosion and high ambient temperature. Often components can then no longer be manufactured or repaired leading to a higher exchange rate [146]. As an example reliability critical components in wind turbines can be mentioned, namely insulated-gate bipolar transistors (IGBTs). IGBTs technology is often used in high-power, high-frequency applica-

tions, e.g., in the power electronic module of wind turbines [85] and solar applications. Significant failure rates of power electronics in wind turbines are often mentioned in the literature [173, 66]. Failures caused by defective IGBTs can be corrected by the replacement of the component. But there is a point in time when the whole power module in the wind turbine should be replaced rather than repaired again.

In [146] the problem of controlling the inventory of electrical parts is considered. The authors stated that consumer electronic products prices often decrease significantly over time while their repair costs increase or remain at the same level. In this case an optimal policy is to replace the failed component with a new one at some point in time rather than to repair it.

In Markov modeling for dependability, PHDs have been applied to describe the duration of availability and unavailability intervals of components [134, 70, 46]. Then, e.g., distribution of the time to system failure can be easily computed. The Markov model presented in [46] incorporates PHDs from measured availability data using data from the failure trace archive (FTA) [110, 100]. The mentioned model also describes correlation in the length of availability or unavailability intervals which is a considerable aspect, since correlation of consecutive lengths of availability or unavailability intervals can often be observed [46].

The trace archive FTA contains large amounts of failure traces of distributed systems in a standard format. The failure data can be prepared in order to obtain time intervals in which a component is available. Then time intervals in which a component is not available (e.g., due to a failure, repair or maintenance) can also be computed [46]. To analyze dependability of wind turbine components we build a PH-Graph using four different traces from the FTA which have been already used in [46] for parameter fitting. The obtained PHDs for different traces are available online [3] such that they can be directly embedded into the PH-Graph model. See [46] and references therein for detailed information about FTA traces and computed PH distributions.

In particular, we use the following PH distributions to describe consecutive availability and unavailability intervals.

- PHD $PH^{(a_1)}$ describing the distribution of the length of the first availability interval

$$\boldsymbol{\pi}^{(a_1)} = (0.197,\ 0.1296,\ 0.1801,\ 0.4932)\,,$$

$$\mathbf{D}_0^{(a_1)} = \begin{pmatrix} -0.0013 & 0 & 0 & 0 \\ 0 & -0.0181 & 0 & 0 \\ 0 & 0 & -0.2034 & 0 \\ 0 & 0 & 0 & -1.5266 \end{pmatrix},$$

- PHD $PH^{(u_1)}$ describing the distribution of the length of the first unvailability interval

$$\boldsymbol{\pi}^{(u_1)} = (0.0329,\ 0.1471,\ 0.1879,\ 0.6320)\,,$$

$$\mathbf{D}_0^{(u_1)} = \begin{pmatrix} -0.0010 & 0 & 0 & 0 \\ 0 & -0.0085 & 0 & 0 \\ 0 & 0 & -0.1194 & 0 \\ 0 & 0 & 0 & -2.2045 \end{pmatrix},$$

- PHD $PH^{(a_2)}$ describing the distribution of the length of the second availability interval

$$\pi^{(a_2)} = (0.1212,\ 0.5197,\ 0.2305,\ 0.1285),$$
$$\mathbf{D}_0^{(a_2)} = \begin{pmatrix} -0.006 & 0 & 0 & 0 \\ 0 & -0.0439 & 0 & 0 \\ 0 & 0 & -0.6452 & 0 \\ 0 & 0 & 0 & -7.9133 \end{pmatrix},$$

- PHD $PH^{(u_2)}$ describing the distribution of the length of the second unavailability interval

$$\pi^{(u_2)} = (0.0352,\ 0.1085,\ 0.1211,\ 0.7350),$$
$$\mathbf{D}_0^{(u_2)} = \begin{pmatrix} -0.0059 & 0 & 0 & 0 \\ 0 & -0.0809 & 0 & 0 \\ 0 & 0 & -1.3065 & 0 \\ 0 & 0 & 0 & -17.6514 \end{pmatrix},$$

- PHD $PH^{(a_3)}$ describing the distribution of the length of the third availability interval

$$\pi^{(a_3)} = (0.0803,\ 0.3100,\ 0.3364,\ 0.2731),$$
$$\mathbf{D}_0^{(a_3)} = \begin{pmatrix} -0.0086 & 0 & 0 & 0 \\ 0 & -0.0773 & 0 & 0 \\ 0 & 0 & -0.8790 & 0 \\ 0 & 0 & 0 & -4.1413 \end{pmatrix},$$

- PHD $PH^{(u_3)}$ describing the distribution of the length of the third unavailability interval

$$\pi^{(u_3)} = (0.0151,\ 0.1323,\ 0.3724,\ 0.48),$$
$$\mathbf{D}_0^{(u_3)} = \begin{pmatrix} -0.0034 & 0 & 0 & 0 \\ 0 & -0.2657 & 0 & 0 \\ 0 & 0 & -2.2640 & 0 \\ 0 & 0 & 0 & -2.2648 \end{pmatrix},$$

- PHD $PH^{(a_4)}$ describing the distribution of the length of the fourth availability interval

$$\pi^{(a_4)} = (0.0840,\ 0.2220,\ 0.2112,\ 0.4826),$$
$$\mathbf{D}_0^{(a_4)} = \begin{pmatrix} -0.0022 & 0 & 0 & 0 \\ 0 & -0.0128 & 0 & 0 \\ 0 & 0 & -0.1004 & 0 \\ 0 & 0 & 0 & -0.5388 \end{pmatrix},$$

- PHD $PH^{(u_4)}$ describing the distribution of the length of the fourth unavailability interval

$$\boldsymbol{\pi}^{(u_4)} = (0.0024,\ 0.2448,\ 0.7527,\ 0),$$
$$\mathbf{D}_0^{(u_4)} = \begin{pmatrix} -0.0025 & 0 & 0 & 0 \\ 0 & -0.156 & 0 & 0 \\ 0 & 0 & -3.7614 & 3.7614 \\ 0 & 0 & 0 & -3.7614 \end{pmatrix}.$$

- and finally PHD $PH^{(ex)}$ describes the length of the interval required for an exchange of the component

$$\boldsymbol{\pi}^{(ex)} = (1.0,\ 0,\ 0,\ 0),$$
$$\mathbf{D}_0^{(ex)} = \begin{pmatrix} -2 & 2 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & -2 \end{pmatrix}.$$

Table 7.5.: First moment and coefficient of variation for the fitted PHDs from [46].

PHD	$E[PH]$	C^2
$PH^{(a_1)}\ (\boldsymbol{\pi}^{(a_1)}, \mathbf{D}_0^{(a_1)})$	159.4849	8.1455
$PH^{(u_1)}\ (\boldsymbol{\pi}^{(u_1)}, \mathbf{D}_0^{(u_1)})$	49.6136	23.6101
$PH^{(a_2)}\ (\boldsymbol{\pi}^{(a_2)}, \mathbf{D}_0^{(a_2)})$	32.406	5.9202
$PH^{(u_2)}\ (\boldsymbol{\pi}^{(u_2)}, \mathbf{D}_0^{(u_2)})$	7.4135	36.021
$PH^{(a_3)}\ (\boldsymbol{\pi}^{(a_3)}, \mathbf{D}_0^{(a_3)})$	13.7283	10.913
$PH^{(u_3)}\ (\boldsymbol{\pi}^{(u_3)}, \mathbf{D}_0^{(u_3)})$	5.3171	91.3286
$PH^{(a_4)}\ (\boldsymbol{\pi}^{(a_4)}, \mathbf{D}_0^{(a_4)})$	59.2509	10.036
$PH^{(u_4)}\ (\boldsymbol{\pi}^{(u_4)}, \mathbf{D}_0^{(u_4)})$	2.9446	92.12
$PH^{(ex)}\ (\boldsymbol{\pi}^{(ex)}, \mathbf{D}_0^{(ex)})$	2	0.25

Parameter estimation for the given PHDs describing the length of availability or unavailability intervals has been performed using the freely available software *gfit* [168]. Table 7.5 contains for all PH distributions first moment and squared coefficient of variation. Note that squared coefficient of variation is used to express the variance of the PHD random variable relative to its mean value [47].

Modeling Availability and Unavailability Paths with PH-Graphs The above PHDs can be used to model lengths of availability or unavailability of a component as described in [46]. If a component is initially available the following Markov chain can be used to consider availability analysis, e.g., to compute transient and stationary availability

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D}^{(a)} & \mathbf{d}_1^{(a)}\boldsymbol{\pi}^{(u)} \\ \mathbf{d}_1^{(u)}\boldsymbol{\pi}^{(a)} & \mathbf{D}^{(u)} \end{pmatrix}. \tag{7.8}$$

However when correlation of durations of availability and unavailability intervals is added, the Markov process (7.8) can be written as [46].

$$\mathbf{Q} = \begin{pmatrix} \mathbf{D}^{(a)} & \mathbf{H}_{(a,u)} \\ \mathbf{H}_{(u,a)} & \mathbf{D}^{(u)} \end{pmatrix}, \tag{7.9}$$

where transfer matrices $\mathbf{H}_{(a,u)}$, $\mathbf{H}_{(u,a)}$ describe the correlation between availability and unavailability intervals. Similarly to the model described in [46] we consider the time-dependent correlation between availability and unavailability durations of a component in our model. It is argued for instance that certain local sources for failures may result in sequences of consecutive failures. Additionally, long available intervals of components with low failure rates alternate with time interval where many components are failed and should be maintained/repaired [46].

For the modeling of availability and unavailability interval lengths during a component's lifecycle (e.g., according to a bathtub curve), several availability and unavailability intervals can be considered. Usually the duration of consecutive availability and unavailability intervals differs (see Tab. 7.5). We assume that during an unavailability interval the power module component is maintained and repair costs are incurred.

Then, let $\tau_{(i,x)}$ be the sojourn time in phase (i,x) of PHD describing the length of unavailability interval. During the sojourn of the process in phase $(i,x) \in \mathscr{S}_T$ the reward increases at rate $c(i,x) = 1$, i.e., $c\,\tau_{(i,x)}$ is the accumulated reward during the time $\tau_{(i,x)}$ representing the incurred costs.

In order to model *costs* required for an exchange of the component we use impulse (instanteneous) rewards. Let $\mathbf{S^u} \in \mathbb{R}^{n,n}$ be the reward matrix with element $\mathbf{S}^u((i,x),(j,y))$ giving reward of a transition from state (i,x) into state (j,y). We assume that $\mathbf{S}^u((i,x),(j,y)) < \infty$ and $\mathbf{S}^u((i,x),(j,y)) = 0$ for $i = j$ or $\mathbf{Q}^u((i,x),(j,y)) = 0$.
Let $\mathbf{s^u}$ be the impulse reward vector with elements

$$\mathbf{s^u}(i,x) = \sum\nolimits_{((j,y)\in\mathscr{S})} \mathbf{S}^{\mathbf{u}(i,x)}((i,x),(j,y)) \cdot \mathbf{Q}^{\mathbf{u}(i,x)}((i,x),(j,y)). \tag{7.10}$$

However, given reward rates $\mathbf{r^u}$ and impulse rewards $\mathbf{s^u}$, we can define cumulative reward rates $\mathbf{rc^u}$ as

$$\mathbf{rc^u}(i,x) = \mathbf{r^u}(i,x) + \mathbf{s^u}(i,x). \tag{7.11}$$

For the properties under consideration, this reward structure allows us to represent costs incurred when the component is exchanged and costs accumulated in the unavailability interval (i.e., during component's repair interval).

Policies In the following we define the PH-Graph for the maintenance example and give a description of its paths. The maintenance PH-Graph is shown in Figure 7.25. Paths include discrete availability and unavailability intervals. We consider paths with a fixed number of availability intervals which are required to cover some production cycle. In our example, the number of availability intervals is set to 6. A component can be either exchanged or repaired when failures caused by end of life wear-out occur. In our example, the end of the component's lifecycle includes 3 availability intervals.

Depending on incurred repair costs, an optimal policy is to replace the failed component with a new one after some availability interval rather than to repair it. The above path of the PH-Graph in Fig. 7.25 corresponds to the policy *R* where repair is

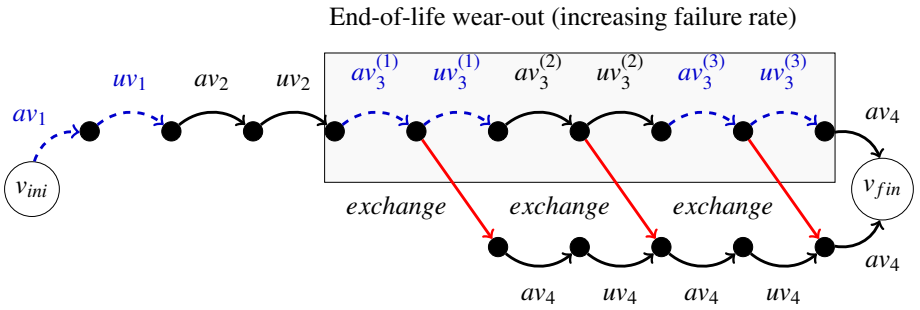


Figure 7.25.: PH-Graph corresponding to maintenance policies. av_i and uv_i describe the ith availability and unavailability interval, respectively. Edges with correlated length of availability and unavailability intervals are highlighted in the same style. The length of edges av_4 and uv_4 on the lower path are uncorrelated. The length of edges av_3 and *exchange* are also uncorrelated.

always performed, but the whole power component is never exchanged. Its generator matrix can be defined as

$$\mathbf{Q}^R = \begin{pmatrix} \mathbf{D}^{(a_1)} & \mathbf{H}_{(a_1,u_1)} & \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{(u_1)} & \mathbf{d}_1^{(u_1)}\boldsymbol{\pi}^{(a_2)} & \dots & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{D}^{(a_2)} & \mathbf{H}_{(a_2,u_2)} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{D}^{(u_2)} & \mathbf{d}_1^{(u_2)}\boldsymbol{\pi}^{(a_3)} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{D}^{(a_3)} & \mathbf{H}_{(a_3,u_3)} & \dots & \mathbf{0} \\ \vdots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{D}^{(a_4)} & \mathbf{d}_1^{(a_4)} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix}, \quad (7.12)$$

where the mean duration of availability intervals decreases after each repair. Here, composition of PHDs $((\boldsymbol{\pi}^{(a_3)}, \mathbf{D}_0^{(a_3)}), (\boldsymbol{\pi}^{(u_3)}, \mathbf{D}_0^{(u_3)}), \mathbf{H}_{(a_3,u_3)})$ describes the distribution of correlated durations of availability interval a_3 and unavailability interval u_3. The corresponding Markov chain has the following generator matrix

$$\mathbf{Q}^{(a_3,u_3)} = \begin{pmatrix} \mathbf{D}_0^{(a_3)} & \mathbf{H}_{(a_3,u_3)} \\ \mathbf{0} & \mathbf{D}_0^{(u_3)} \end{pmatrix}.$$

Transfer matrices for different availability and unavailability intervals are given in Sec. D.4. The PHD composition of consecutive unavailability and availability intervals

is given by the generator matrix

$$\mathbf{Q}^{(u_3,a_3)} = \begin{pmatrix} \mathbf{D}_0^{(u_3)} & \mathbf{d}_1^{(u_3)}\boldsymbol{\pi}^{(a_3)} \\ \mathbf{0} & \mathbf{D}^{(a_3)} \end{pmatrix},$$

such that lengths of consecutive unavailability and availability intervals are independently distributed. In fact, the Markov process presented in (7.8), (7.12) is enrolled using different PHDs for durations of availability and unavailability intervals. Then, the path matrix (7.12) can be rewritten as

$$\mathbf{Q}^R = \begin{pmatrix} \mathbf{Q}^{(a_1,u_1)} & \mathbf{D}_1^{(a_1,u_1)} & \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{(a_2,u_2)} & \mathbf{D}_1^{(a_2,u_2)} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{Q}^{(a_3,u_3)} & \mathbf{D}_1^{(a_3,u_3)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{Q}^{(a_3,u_3)} & \mathbf{D}_1^{(a_3,u_3)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{Q}^{(a_3,u_3)} & \mathbf{D}_1^{(a_3,u_3)} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{D}^{(a4)} & \mathbf{d}_1^{(a4)} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix}, \quad (7.13)$$

where matrices $\mathbf{D}_1^{(a_i,u_i)}$ are defined as follows

$$\mathbf{D}_1^{(a_i,u_i)} = \begin{pmatrix} \mathbf{0} \\ \mathbf{d}_1^{(u_i)}\boldsymbol{\pi}^{(a_{i+1})} \end{pmatrix},$$

where $\mathbf{d}_1^{(u_i)}$ is the exit vector of the PHD describing an unavailability interval u_i and $\boldsymbol{\pi}^{(a_{i+1})}$ is the initial distribution of the subsequent availability interval on the path.

The number of availability and unavailability intervals described by $PH^{(a_3)}$ and $PH^{(u_3)}$ can be parameterized in order to model higher failure rates at the end of the component's lifecycle. We use three consecutive availability and unavailability intervals given by $PH^{(a_3)}$, $PH^{(u_3)}$ which are highlighted in blue in (7.13). It holds that $\boldsymbol{\pi}^R = (\boldsymbol{\pi}^{(a_1)}, \mathbf{0})$.

However, the tendency is that towards the end of component's lifecycle, exchange of the whole module should be an option to be considered. In the case when an exchange of the component is performed after the first availability interval $av_3^{(1)}$ in Fig. 7.25, policy E_1 can be defined by the following path matrix

$$\mathbf{Q}^{E_1} = \begin{pmatrix} \mathbf{Q}^{(a_1,u_1)} & \mathbf{D}_1^{(a_1,u_1)} & \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{(a_2,u_2)} & \mathbf{D}_1^{(a_2,u_2)} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{Q}^{(a_3,ex)} & \mathbf{D}_1^{(a_1,ex)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{Q}^{(a_4,u_4)} & \mathbf{D}_1^{(a_4,u_4)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{Q}^{(a_4,u_4)} & \mathbf{D}_1^{(a_4,u_4)} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{D}^{(a_4)} & \mathbf{d}_1^{(a_4)} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix}. \quad (7.14)$$

In the case when the exchange of the component is performed after the second avail-

ability interval $av_3^{(2)}$ in Fig. 7.25, policy E_2 is given by the following path matrix

$$\mathbf{Q}^{E_2} = \begin{pmatrix} \mathbf{Q}^{(a_1,u_1)} & \mathbf{D}_1^{(a_1,u_1)} & \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{(a_2,u_2)} & \mathbf{D}_1^{(a_2,u_2)} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{Q}^{(a_3,u_3)} & \mathbf{D}_1^{(a_3,u_3)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{Q}^{(a_3,ex)} & \mathbf{D}_1^{(a_3,ex)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{Q}^{(a_4,u_4)} & \mathbf{D}_1^{(a_4,u_4)} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{D}^{(a_4)} & \mathbf{d}_1^{(a_4)} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix}. \quad (7.15)$$

Finally, when the exchange of the component is performed after the third availability interval $av_3^{(3)}$ in Fig. 7.25, policy E_3 is defined by the matrix

$$\mathbf{Q}^{E_3} = \begin{pmatrix} \mathbf{Q}^{(a_1,u_1)} & \mathbf{D}_1^{(a_1,u_1)} & \mathbf{0} & \dots & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^{(a_2,u_2)} & \mathbf{D}_1^{(a_2,u_2)} & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{Q}^{(a_3,u_3)} & \mathbf{D}_1^{(a_3,u_3)} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \mathbf{Q}^{(a_3,u_3)} & \mathbf{D}_1^{(a_3,u_3)} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \mathbf{Q}^{(a_3,ex)} & \mathbf{D}_1^{(a_3,ex)} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \mathbf{D}^{(a_4)} & \mathbf{d}_1^{(a_4)} \\ \mathbf{0} & \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix}. \quad (7.16)$$

Analyzing Costs for Maintenance of the Component Maintenance PH-Graph given by (7.13)- (7.16) defines a CTMDP. We analyze the induced final horizon CTMDP under minimal cumulative reward criteria. The minimal cumulative reward is interpreted as minimal incurred maintenance costs at the end of life of the power module. The maintenance costs are made up of repair costs and component exchange costs.

Minimization of cumulative reward can be computed using the t dynamic programming approach (see Alg. 7.1) where the initial gain vector is fixed to zero vector, i.e., $\mathbf{g}_T = \mathbf{0}$. The reward vector equals to $\mathbf{r} = (1,1,1,1)$ for PHDs describing the length of unavailability intervals, and $\mathbf{r} = (0,0,0,0)$ otherwise. Impulse rewards are defined as

$$\mathbf{S}^{exchange}((a_3,x),(ex,y)) = 55,$$

for all phases $x \in \mathscr{S}_T$ of $PH^{(a_3)}$ and all $y \in \mathscr{S}_T$ of $PH^{(ex)}$. Transfer matrix $\mathbf{H}_{(a_3,ex)}$ defines transition rates from $PH^{(a_3)}$ to $PH^{(ex)}$

$$\mathbf{H}_{(a_3,ex)} = \begin{pmatrix} 0.008663 & 0 & 0 & 0 \\ 0.077378 & 0 & 0 & 0 \\ 0.879091 & 0 & 0 & 0 \\ 4.141319 & 0 & 0 & 0 \end{pmatrix},$$

such that cumulative reward rates are obtained as $(55 \cdot \mathbf{H}_{(a_3,ex)}\mathbb{1})$, which results in vector $\mathbf{rc}^{exchange} = (0.4765, 4.2558, 48.35, 227.7726)$ giving cumulative rewards for phases $(a_3,x) \in \mathscr{S}_T$ of $PH^{(a_3)}$. Cumulative reward vectors $\mathbf{rc}^R$, $\mathbf{rc}^{E_1}$, $\mathbf{rc}^{E_2}$ and $\mathbf{rc}^{E_3}$ for policies defined by (7.13)- (7.16) are given in Sec. D.4.

We analyze the model in the interval $(0,1000]$ and obtain approximations of cumulative reward values with Alg. 7.1 using cumulative reward vectors $\mathbf{rc}^R$, $\mathbf{rc}^{E_1}$, $\mathbf{rc}^{E_2}$ and $\mathbf{rc}^{E_3}$ as an input.

First, we observe the policy R defined in (7.13). It corresponds to the path

$$av_1,\, uv_1, \ldots,\, av_3^{(1)},\, uv_3^{(1)}, \ldots,\, av_3^{(3)},\, uv_3^{(3)}, av_4,$$

where repair is always performed but the whole power component is never replaced (see the above path in Fig. 7.25). In contrast, the maintenance policy E_1 defined in (7.14) describes the exchange of the whole component after the first availability interval $av_3^{(1)}$. It corresponds to the following path containing *exchange* edge in Fig. (7.25)

$$av_1,\, uv_1, \ldots,\, av_3^{(1)},\, exchange,\, av_4,\, uv_4, \ldots, av_4.$$

The policies R and E_1 result in the following gain vectors:

	$(av_3^{(1)}, 1)$	$(av_3^{(1)}, 2)$	$(av_3^{(1)}, 3)$	$(av_3^{(1)}, 4)$
$\mathbf{g}^{R}_{0,T}(av_3^{(1)}, x_i)$	65.0437	11.5348	10.8388	10.8404
$\mathbf{g}^{E_1}_{0,T}(av_3^{(1)}, x_i)$	60.4728	60.5774	60.6075	60.6979

The maintenance policy E_2 defined by (7.14) where the exchange of the whole component is performed after the second availability interval $av_3^{(2)}$ corresponds to the following path in Fig. (7.25)

$$av_1,\, uv_1, \ldots,\, av_3^{(1)},\, uv_3^{(1)},\, av_3^{(2)},\, exchange,\, av_4,\, uv_4, \ldots, av_4.$$

Analysis of policies R and E_2 in the interval $(0, 1000]$ results in the following gain vectors:

	$(av_3^{(2)}, 1)$	$(av_3^{(2)}, 2)$	$(av_3^{(2)}, 3)$	$(av_3^{(2)}, 4)$
$\mathbf{g}^{R}_{0,T}(av_3^{(2)}, x_i)$	60.2203	6.3648	5.6563	5.6570
$\mathbf{g}^{E_2}_{0,T}(av_3^{(2)}, x_i)$	57.7639	57.8163	57.8418	57.9318

And finally, the policy E_3 with generator defined in (7.16) describes the behavior where the exchange of the whole component is performed as late as possible. It corresponds to the path

$$av_1,\, uv_1, \ldots,\, av_3^{(1)},\, uv_3^{(1)},\, av_3^{(2)},\, uv_3^{(2)},\, av_3^{(3)}\, exchange, av_4$$

in Fig. (7.25). One can compare gain vectors of the policies R and E_3.

	$(av_3^{(3)}, 1)$	$(av_3^{(3)}, 2)$	$(av_3^{(3)}, 3)$	$(av_3^{(3)}, 4)$
$\mathbf{g}^{R}_{0,T}(av_3^{(3)}, x_i)$	54.13	1.1613	0.4416	0.4415
$\mathbf{g}^{E_3}_{0,T}(av_3^{(3)}, x_i)$	54.9907	55.0021	55.0242	55.1140

One can see that in state $(av_3^{(3)}, 1)$ of the PHD for the third availability interval the optimal decision is to repair the component.

State dependent decisions When the effect of correlation should be explored, one can compute the optimal maintenance decision depending on the exit phase of the distribution $PH^{(av_3)}$. We summarize results on optimal policies in the interval $(0, 900]$ in Tables. 7.6, 7.7, 7.8.

Table 7.6.: Expected cumulative rewards for policies R and E_1.

	Accumulated reward in $(0,900]$		
Phase of the $PH^{(av_3)}$	Succ. $uv_3^{(1)}$	Succ. *exchange*	Optimal Successor Edge
Phase 1	63.4126	60.3607	*exchange*
Phase 2	11.3101	60.502	$uv_3^{(1)}$ / *repair*
Phase 3	10.6234	60.534	$uv_3^{(1)}$ / *repair*
Phase 4	10.6256	60.6245	$uv_3^{(1)}$ / *repair*

Table 7.7.: Expected cumulative rewards for policies R and E_2.

	Accumulated reward in $(0,900]$		
Phase of the $PH^{(av_3)}$	Succ. $uv_3^{(2)}$	Succ. *exchange*	Optimal Successor Edge
Phase 1	58.8	57.7071	*exchange*
Phase 2	6.2574	57.7834	$uv_3^{(2)}$ / *repair*
Phase 3	5.5535	57.8098	$uv_3^{(2)}$ / *repair*
Phase 4	5.5545	57.9	$uv_3^{(2)}$ / *repair*

Table 7.8.: Expected cumulative rewards for policies R and E_3.

	Accumulated reward in $(0,900]$		
Phase of the $PH^{(av_3)}$	Succ. $uv_3^{(3)}$	Succ. *exchange*	Optimal Successor Edge
Phase 1	54.13	54.9776	$uv_3^{(3)}$ / *repair*
Phase 2	1.1613	55.0021	$uv_3^{(3)}$ / *repair*
Phase 3	0.4416	55.0242	$uv_3^{(3)}$ / *repair*
Phase 4	0.4415	55.114	$uv_3^{(3)}$ / *repair*

One can see that for a short duration of the availability interval $av_3^{(1)}$ it is better to repair the component, i.e., to choose the successor edge $uv_3^{(1)}$. But for a long duration of the availability interval $av_3^{(1)}$, the best decision is to choose the adjacent edge *exchange*, i.e., to exchange the whole module. Since lengths of availability and unavailability intervals are positively correlated, duration of the adjacent unavailability interval $uv_3^{(1)}$ is affected by long duration of the predecessor $av_3^{(1)}$.

Assume now that the exit phase of $PH^{(av_3)}$ is a long phase, i.e., a phase where the remaining time till absorption is longer than the average of the PHD $PH^{(av_3)}$. We can see that from a long phase, higher repair costs at the successor edge $uv_3^{(1)}$ are incurred in $(0, 900]$. Thus, the positive correlation between interval lengths results in a positive slope of the expected cumulative reward at the adjacent edge $uv_3^{(1)}$. Whereas the expected cumulative reward at the adjacent edge *exchange* remains not affected by the duration at $av_3^{(1)}$, since both edge weights are independent.

Observe that from a long phase of interval $av_3^{(3)}$, the optimal choice is edge $uv_3^{(3)}$. Since results are computed backwards the effect of positive correlation cumulates for edges $av_3^{(1)}$, $av_3^{(2)}$. In particular, long duration of interval $av_3^{(1)}$ has an affect on durations of intervals $uv_3^{(1)}$, $av_3^{(2)}$, $uv_3^{(2)}$, $av_3^{(3)}$ and $uv_3^{(3)}$. Thus, costs accumulated in *end-of-life wear-out period* depend on duration of availability interval $av_3^{(1)}$.

Cumulative rewards for all maintenance policies are analyzed for time bound $T = 100, \ldots, 1500$. Results for long and short phases [2] of the PHD for availability interval $av_3^{(i)}$, $i = 1,\ 2,\ 3$ are summarized graphically in Figure 7.26. The positive correlation results in a smaller cumulative reward for the policy R (always repair) for a short time bound T but in a slightly higher cumulative reward for a longer time bound T.

In contrast, we analyze the maintenance PH-Graph where durations of availability interval av_3 and unavailability interval uv_3 are independent. The results are summarized in Figure 7.27. When the effect of correlation is neglected, the optimal policy is always to repair the component.

[2] A *long* phase is a phase where the remaining time till absorption is longer than the average of the PHD $PH^{(av_3)}$, i.e., it is the first phase with mean sojourn time $\frac{1}{\lambda(1)} = \frac{1}{0.008663} = 115.4$. A *short* phase is a phase where the remaining time till absorption is shorter than the average. The fourth phase is a short phase with mean sojourn time is $\frac{1}{\lambda(4)} = \frac{1}{4.1413} = 0.2414$.

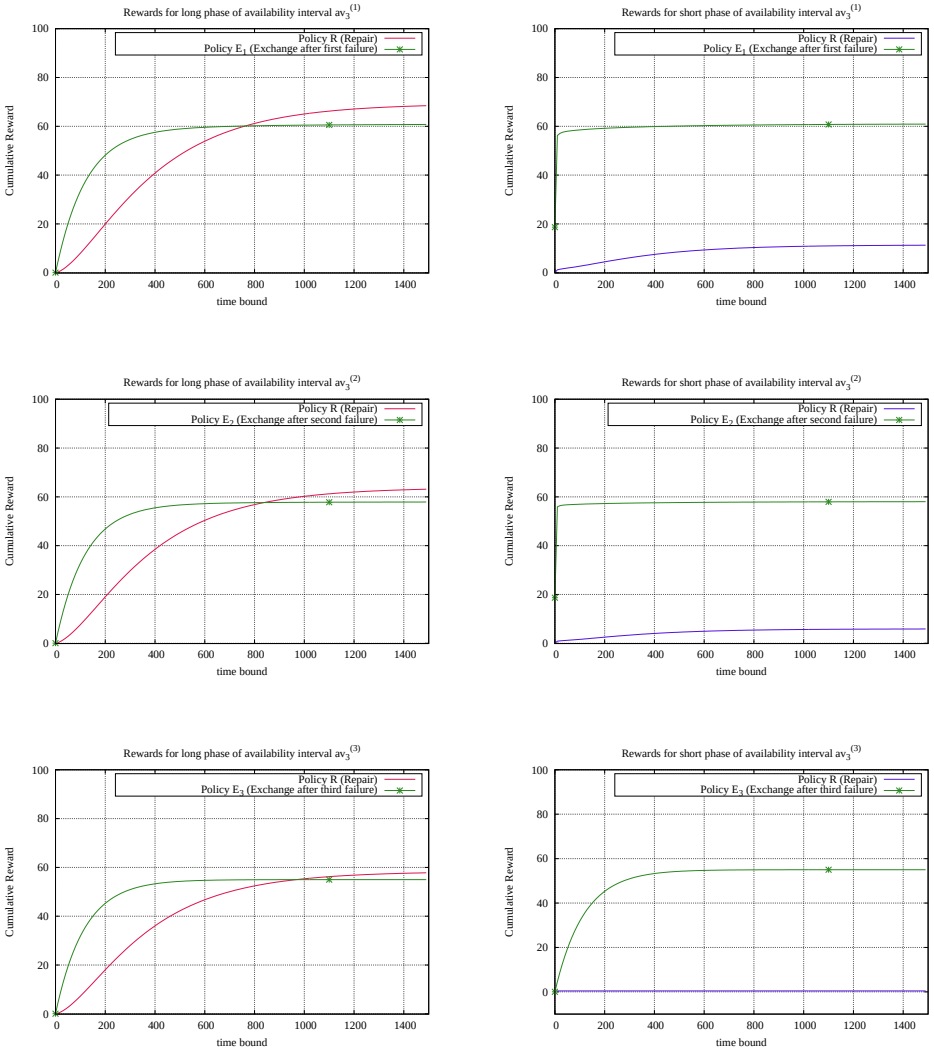


Figure 7.26.: Cumulative rewards depending on the phase of PHD for availability interval $av_3^{(i)}$, for $i = 1,\ 2,\ 3$ and $T = 0, \ldots, 1500$.

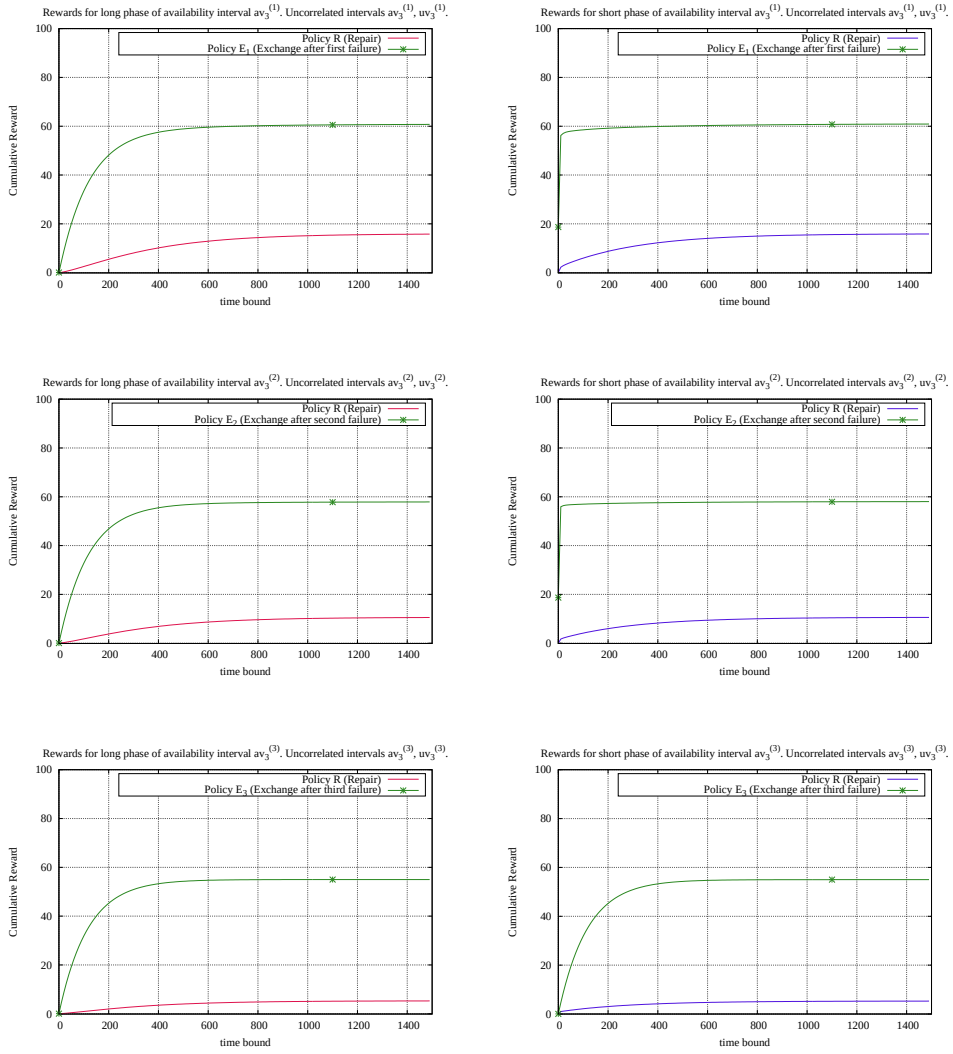


Figure 7.27.: Cumulative rewards depending on the phase of PHD for availability interval $av_3^{(i)}$, for $i = 1,\ 2,\ 3$ and $T = 0, \ldots, 1500$. Durations of availability and unavailability intervals are independent.

History-based optimal decisions In the real system maintenance decisions cannot depend on the phase, they have to depend on history vectors $\bar{\psi}_{(av_1,w_1,\ldots,av_l,w_l)}$ computed with (5.15)-(5.16). Using history vectors $\bar{\psi}_{(av_1,w_1,\ldots,av_l,w_l)}$ conditional cumulative rewards can be computed such that the optimal decision depends on the realized length of availability interval and not on the phase of the PHD $PH^{(av_3)}$.

We analyze the cumulative rewards via adjacent edges uv_3 and *exchange* depending on the realized length of availability interval av_3 for various time bounds. We compute vectors $\bar{\psi}_{(av_3,w,i_{next},0)}$ with $w \in [0.1, 300]$, $next \in \{uv_3, exchange\}$. The values of cumulative reward for policies *repair* and *exchange* are summarized in Figures 7.28, 7.29.

Here, the reward of decision *exchange* is not affected by the realized length of availability interval av_3. But the positive correlation between the lengths of availability and unavailability intervals av_3 and uv_3 results in a positive slope of the accumulated reward in unavailability interval uv_3. In Fig. 7.29 one can see that for larger time bounds and long durations of availability interval av_3 the optimal decision is to exchange the whole component because of high repair costs.

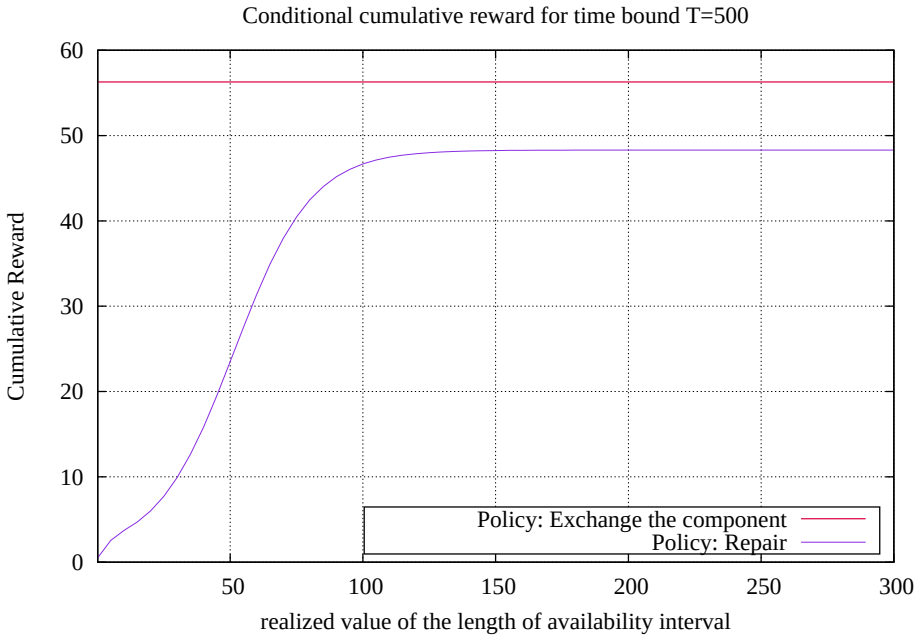


Figure 7.28.: Cumulative rewards depending on the realized length of availability interval av_3 for the time bound $T = 500$ and discretization parameter $h = 0.001$.

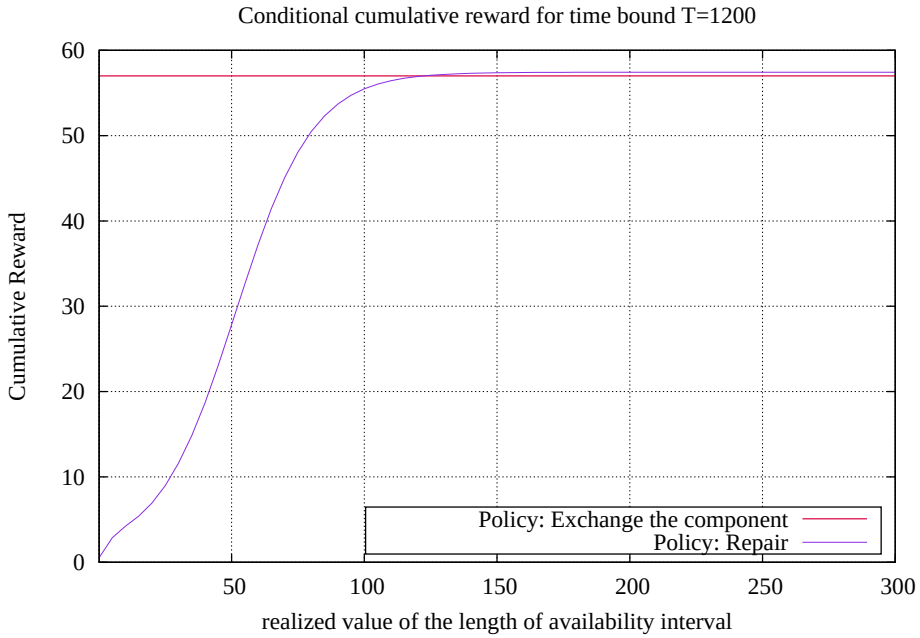


Figure 7.29.: Cumulative rewards depending on the realized length of availability interval av_3 for the time bound $T = 1200$ and discretization parameter $h = 0.001$.

7.5. Summary

In this Chapter, we first studied two practical problems of route planning under uncertainty with dependent weights on adjacent edges, namely computing the path with a minimal expected length and finding the path that maximizes the probability to reach the destination node within a specified time interval. We constructed a PH-Graph using measurements resulting from SUMO vehicular traffic simulation with the aim to obtain a realistic model for roads of the city of Cologne. We showed in the realistic example that depending on the currently realized traveling times, choosing a correlated adjacent edge results in a much better expected path weight of the route than the choice of an uncorrelated adjacent edge.

We investigated how increasing the PHD order could refine results obtained with an EM algorithm and enhance the achievable correlation. Slightly better results could be obtained for larger PH-Graphs that use PH distributions of orders $6,\ldots,20$. For a given vehicle mobility trace, already 6 phases are sufficient to obtain meaningful results. Further increasing of the number of phases would result in an insignificant rise of the log-likelihood values and correlation coefficient. Therefore, already small PHDs with less that 10 phases can be applied in real applications. We refer to [40] where extensive complexity results are presented. It can be summarized, that PH-Graph models can be parameterized and analyzed sufficiently efficient which indicates

their practical applicability, e.g., in a drivers decision support system.

We also show that for weighted graphs with negative weights, solution approaches can be used to compute optimal path weight and probability. We developed a simple investment model to show that PH-Graphs with negative weights can be well applied in financial optimization. In our third example, we showed that PH-Graphs can be applied in dependability analysis. We developed two PH-Graph models which describe correlation in the length of availability and unavailability intervals and can be used to analyze maintenance costs in dependence of duration of availability interval of a component.

Chapter 8

Conclusions

In this work we have proposed a new model class for stochastic weighted graphs with correlated weights based on PHDs, MAPs and CTMDPs. These famous stochastic approaches were combined together to model correlated weights on adjacent edges and to compute optimal solutions in applications where correlation is a prominent aspect.

PHDs and MAPs have a great potential in describing real processes. In particular, PHDs allow one to approximate even multimodal distributions closely. We use PHDs to model general distributions of the weights in a stochastic graph. When PHDs are extended to describe correlated inter-event times, the resulting model is called a MAP which is a very flexible and general class of stochastic processes. We used the concept of marked transitions indicating events in a MAP to introduce correlations between PHDs for weights of adjacent edges. The newly developed graph model is called a PH-Graph and can be used for optimization of systems including correlations, e.g., finding shortest paths on roads with congestion effects or investment analysis of uncertain financial markets containing components with correlated costs and incomes.

We proved that the parameters of the new PH-Graph model can be determined with well established and adopted methods. Still parameter fitting remains a complex optimization problem, we have shown that adopted fitting methods often give good results with acceptable computational effort. For acyclic PHDs of order 2 we investigated different transformations and representations in order to increase the possible range of correlation that PHDs for weights can capture. We proved which representations result in the maximal first joint moment for a subsequent transfer matrix fitting step.

We have also shown how PH-Graph can be mapped to a CTMDP to solve SSPP with correlations using standard methods from the CTMDP field. Our goal was to provide the combination of PHDs and transfer matrices with CTMDPs that helps to apply their great modeling potential in optimization and decision support. Our extensive numerical experiments indicate that the effect of correlation between weights should not be neglected. Optimal solutions of stochastic shortest path problems depend on correlated weights of adjacent edges. For example, when congestions occur choosing an uncorrelated adjacent edge can result in a much better route than the choice of a correlated adjacent edge.

8.1. Future work

In the following we present interesting future directions and several methodological questions. To describe many practical situations where knowledge about graph weights changes over time, time-dependent weights could be considered. This would result in inhomogeneous CTMDPs. Similar extensions should be possible for PHGs with discounted weights as well. We would also like to explore how mean-variance optimization methods can be adopted for SSPPs with correlations. In this case the decision maker is interested in both the mean and the variance of the path weight.

Another promising research field is to consider a multi-objective MDP problem. In Chapter 5, we developed solution methods for two problems of route planning in weighted graphs with dependent edge weights, namely computing a path with minimal expected weight and computing a path maximizing probability of arriving on time. One can be interested in optimizing these two objectives simultaneously which results in a multi-objective view of decision making. First, a weight vector $\mathbf{w}$ can be introduced to express the relative importance of each objective. Then, standard methods from MDP field can be extended to find the optimal policy of linearly weighted problem. However, often a weight vector $\mathbf{w}$ is not known apriori. Thus, methods for computing the set of optimal solutions for all possible weights for linearly weighted problem should be developed by adopting known methods from multi-objective MDPs.

It is known that PHDs even of high order can be generated within a few seconds such that PHGs can be parameterized in an acceptable time. Furthermore, the computation of an optimal policies for fairly large state spaces can be performed in a moderate time. However, it should be investigated whether large networks possibly containing cycles can be efficiently handled. Further reserach is needed to determine how the performance of the approach will scale with large network size. Successful efforts in this direction could contribute to the application of PH-Graphs in vehicle route guidance systems.

Another direction worth pursuing is investigation of similarity transformation methods for acyclic PHDs with an arbitrary number of states in order to increase the possible range of correlation that PHDs for weights of adjacent edges can capture. When a PHD cannot be transformed into the hyperexponential representation and in cases where the transformations do not result in a sufficiently large correlation, the optimal representation of PHDs should be determined.

Another interesting question is if a PHD composition of high order can be approximated by a PHD composition of a lower order. Generally, the problem of finding a smaller order representations is one of the most interesting and challenging theoretical questions on the field of PHDs, also because the PHD representation is not unique. Successful analytical and numerical work in this direction could treat the problem of state-space explosion when modeling PH-graphs for huge stochastic graphs with up to a billion edges and result in approximation methods.

Appendix A

Acronyms

A.1. Acronyms

APHD	Acyclic Phase-Type Distribution
BMAP	Batch Markovian Arrival Process
BS	Basic Series of an APH
cdf	cumulative distribution function
CTMC	Continuous Time Markov Chain
CTMDP	Continuous Time Markov Decision Process
DTMDP	Discrete Time Markov Decision Process
EM	Expectation Maximization
HErD	Hyper-Erlang Distribution
IPP	Interrupted Poisson Process
LP	Linear Program
MAP	Markovian Arrival Process
MDP	Markov Decision Process
ME	Matrix Exponential
ML	Maximum-Likelihood
MMPP	Markov Modulated Poisson Process
NNLS	Non-Negative Least Squares approach
pdf	probability distribution function
PH	Phase-Type

PHD	Phase-Type Distribution
PI	Policy Iteration
QN	Queueing Network
VI	Value Iteration

Appendix B

Notations

B.1. Markov Chains

α	A uniformization constant
$APHD(n)$	APHD of order n
$\beta(n, \alpha t)$	Density function of the Poisson process with rate α
B_i	number of times a PHD starts in phase i
$\mathbf{D}_0$	Matrix of transition rates between transient states of an absorbing CTMC
$\mathbf{D}_1$	Matrix of transition rates generating an event of a MAP
$\mathbf{d}_1$	Vector of transition rates from transient states to the single absorbing state $n+1$ of an absorbing CTMC, i.e., exit vector of a PHD
$E[X]$	Expectation of random variable X
$e^{\mathbf{Q}t}$	Matrix exponential
f_i	Probability eventually return to state i given that the process started in state i
$\mathbf{I}$	Identity matrix
$\mathbf{M}$	$= -\mathbf{D}_0^{-1}$ moment matrix of a PHD or MAP, i.e., fundamental matrix of an absorbing CTMC
n	Order of a PHD or MAP
$\{N_t \mid t \geq 0\}$	Poisson process with rate α
π_0	Initial probability distribution vector. $\pi_0(i)$ gives the probability that the Markov process starts in state i
π	Steady-state probability distribution vector

$\pi^{(n)}$	Transient probabilities after n time steps in a DTMC. $\pi^{(n)}(i)$ gives the probability with which the Markov process occupies state i after n transitions have occured
$\pi^{(t)}$	Transient probabilities of a homogeneous CTMC. $\pi^{(t)}(i)$ gives the probability that a CTMC is in state i at time t
p_{ij}	Single-step transition probability, i.e. $p_{ij} = Prob(X(t_{k+1}) = j \mid X(t_k) = i)$ for all $k = 1, 2, \ldots$
$p_{ij}^{(n)}$	n-step transition probability
$\hat{\mathbf{P}}$	Transition probability matrix of the discrete-time Markov chain
$\tilde{\mathbf{P}}$	Transition probability matrix of an embedded Markov chain of a continuous-time Markov process
$\mathbf{P}_t$	$= e^{\mathbf{Q}t}$, transition probability matrix with components $p_{ij}(t) = Prob(X(t) = j \mid X(0) = i)$
$\mathbf{P}$	$= \mathbf{I} + \frac{1}{\alpha}\mathbf{Q}$, matrix of the discrete-time Markov chain used for uniformization
$\mathbf{Q}$	Infinitesimal generator matrix of a Continuous-Time Markov chain
$\mathscr{S}$	State space of a Markov chain
$\mathscr{S}_T$	Set of transient states of an absorbing CTMC
$\mathscr{S}_A$	Set of absorbing states of an absorbing CTMC
$VAR[Y]$	Variance of random variable Y
$\{X(t)\}_{t\geq 0}^{\infty}$	Markov process
Z_i	Total time spent in phase i of a PHD before generating an event

B.2. Markov Decision Processes

$\mathscr{D}(i)$	Set of available decisions in state i
$\mathscr{D}$	$\mathscr{D} := \bigcup_{i \in \mathscr{S}} \mathscr{D}(i)$
$\mathbf{d}$	Policy $\mathbf{d} = (\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{T-1})$, $T \leq \infty$. Stationary policy $\mathbf{d}$ uses decision rule $\mathbf{u}$ every period
$\mathbf{d}^*$	Optimal policy
$G(t)$	Random variable representing accumulated reward in the time interval $[0, t)$
$G^{\mathbf{d}}$	$= \pi \mathbf{g}_{0,T}^{\mathbf{d}}$. The gain of the policy $\mathbf{d}$ for the expected total reward criterion

$\mathbf{g}^{\mathbf{d}}_{t,T}$	Gain vector containing values of accumulated expected rewards in the time interval $[t,T]$ in a finite horizon CTMDP
$\mathbf{g}^{\mathbf{d}}_{0,T}$	Gain vector containing values of accumulated expected rewards in the time interval $[0,T]$ in a finite horizon CTMDP
$\mathbf{g}^{\mathbf{d}}_{0}$	Gain vector at time 0 for infinite horizons average reward criterion, and accumulated reward to absorption criterion
$\mathbf{g}^{\mathbf{d}}_{t,T}(i)$	Expected total reward under policy $\mathbf{d}$ from time t onward, if the CTMDP is in state i at time t
$\mathbf{g}_T$	Vector containing values of rewards at the final decision epoch T in a finite horizon CTMDP
$\mathbf{g}^{min}_{t,T}$	Infimum of $\mathbf{g}^{\mathbf{d}}_{t,T}$ over all policies
$\mathbf{g}^{max}_{t,T}$	Supremum of $\mathbf{g}^{\mathbf{d}}_{t,T}$ over all policies
$\mathbf{g}^{*}_{t,T}$	Optimal gain vector
$\mathscr{H}$	Set of decision epochs
i	A state i
$\lambda(i)$	Continuous time rate of the state i. $\lambda(i) = -\mathbf{Q}^u(i,i)$
$\mathscr{P}$	Set of decision rules for all states. $\mathscr{P} = \bigtimes_{i=1}^{n} \mathscr{D}(i)$
$\mathbf{p}^{\mathbf{d}}_t$	Transient probability distribution vector under policy $\mathbf{d}$ at time t
π	Initial probability distribution vector at time 0
$\mathscr{Q}$	Set of transition rates
$\mathbf{Q}^u$	Transition rate matrix of the CTMDP when decision u is chosen
$\mathbf{Q}^{\mathbf{u}}$	Transition rate matrix of the CTMDP where (i,j)-th element is given by $\mathbf{Q}^{\mathbf{u}(i)}$
$\mathbf{Q}^u(i,j)$	Transition rate when action u is chosen in state i and the next state is j in the CTMDP
$\mathscr{R}$	Set of rewards
$r^u(i)$	Stationary version of $r^u_n(i)$
$r^u(i,j)$	Stationary version of $r^u_n(i,j)$
$r^u_n(i)$	A state reward. A value of accumulated reward per time unit if system is in state i at decision epoch t_n and action u is chosen.
$r^u_n(i,j)$	A transition reward. A value of one period reward if system is in state i at decision epoch t_n, action u is chosen, and the state at decision epoch t_{n+1} is j.

$r_{fin}(i)$	Terminal reward in state i in a finite horizon problem
$\mathbf{r}^{\mathbf{u}_t}$	Reward vector for the decision vector $\mathbf{u}_t$ taken at time t
$\mathbf{r}^{\mathbf{u}_t}(i)$	Expected value of reward gained by staying one time unit in state i for the decision vector $\mathbf{u}_t$ taken at time t
$\mathbf{r}^{\mathbf{d}}$	Vector determined by elements $\mathbf{d}(\tau) = \mathbf{u}_\tau$ which is the decision vector at time $\tau \in [t,T]$
$\mathbf{r}^{\mathbf{d}}_A$	Vector containing zero rewards for absorbing states $i \in \mathscr{S}_A$ and $\mathbf{r}^{\mathbf{u}_\tau}(i)$ otherwise
$\mathscr{S}$	Set of states
t	A decision epoch. t_n is the nth decision epoch
$\tau(i)$	A sojourn time in state i
u	An action
$\mathbf{u}$	Decision rule
$\mathbf{u}(i)$	Action chosen by decision rule $\mathbf{u}$ in state i
$\mathbf{u}_t(i)$	Action chosen by decision rule $\mathbf{u}_t$ if system is in state i at time t
$\mathbf{u}_t$	Decision rule at decision epoch t
$\mathbf{V}^{\mathbf{d}}_{r,t}(i,j)$	Conditional probability that CTMC is in state j at time point t given that the CTMC is in state i at time point r and policy $\mathbf{d}$ is used in the interval $[r,t]$
ω	Sample path of the state process
Ω	Set of all sample paths of CTMDP
$X(t)$	Random variable representing state of system at time t
$Y(t)$	Random variable representing action selected at time t
$\mathbf{0}$	Matrix or vector where every entry is 0
$\mathbb{1}$	Vector where every entry is 1
$\mathbf{z}^T$	Transpose of a vector $\mathbf{z}$

B.3. Stochastic Shortest Path Problems

γ	Discount factor
$\mathbf{g}^{\mathbf{u}}$	The total expected reward vector for policy $\mathbf{u}$
$\mathbf{g}^{\mathbf{u}_k}$	The total expected reward vector for policy $\mathbf{u}$ in the k-th policy iteration
$\mathbf{g}^*$	The minimal total expected reward vector
$\mathbf{g}_k$	The total expected reward after k-th iteration. E.g., $\mathbf{g}_k = T^k \mathbf{g}_0$, or $\mathbf{g}_k = T\,\mathbf{g}_{k-1}$
$\mathbf{P}^{\mathbf{u}}$	Transition probability matrix under stationary policy $\mathbf{u}$ where $\mathbf{P}^{\mathbf{u}}(i,j) = \mathbf{P}^{\mathbf{u}(i)}(i,j)$
$\mathbf{r}^u(i)$	Expected value of one period reward for state i when decision $u \in \mathscr{D}(i)$ is chosen
$T\,\mathbf{g}$	Optimal reward function for one period problem
$T_{\mathbf{u}}\,\mathbf{g}$	Reward function associated with the stationary policy $\mathbf{u}$ for one period problem
T^k	$T^k\,\mathbf{g} = T\,(T^{k-1}\,\mathbf{g})$. The composition of the mapping T with itself k times
$\mathbf{u}$	Stationary policy. $\mathbf{u}$ stays for a stationary policy $\mathbf{d} = (\mathbf{u}, \mathbf{u}, \ldots)$, $\mathbf{d} \in \Pi$.

Appendix C

Proofs

C.1. Transformation of 2-order APHDs

Proof of Theorem 6.2. We express the first joint moment $\mu_{1,1}$ in Eq. 6.11 in terms of all possible paths in the PH_i following all possible paths in the PH_j Let $a^{\delta}_{i,j}$ be the probability that the second APHD PH_j starts in the state i and j is the last state before absorption. Similarly to Eq. 6.12 it holds that

$$a^{\delta}_{1,1} = \pi^{\delta}(1-\lambda^{\delta}_{1,2}),\ a^{\delta}_{1,2} = \pi^{\delta}\lambda^{\delta}_{1,2},\ a^{\delta}_{2,2} = 1-\pi^{\delta} \text{ and } a^{\delta}_{2,1} = 0, \tag{C.1}$$

Remark that $\nu_{i,j}$ is the mean duration if the process PH_j starts in state i and state j is the last state before absorption of the PH_j. As given in Eq. 6.13 we have

$$\nu_{1,1} = \frac{1}{\lambda_1} = 1, \nu_{1,2} = 1 + \frac{1}{\lambda_2}, \text{ and } \nu_{2,2} = \frac{1}{\lambda_2}. \tag{C.2}$$

The transfer matrix $\mathbf{H}_{ij}$ can be written as

$$\mathbf{H}_{ij} = \begin{pmatrix} 0 & 0 \\ 0 & \mu_2 \end{pmatrix} \begin{pmatrix} b_1 & 1-b_1 \\ b_2 & 1-b_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ b_2\mu_2 & (1-b_2)\mu_2 \end{pmatrix},$$

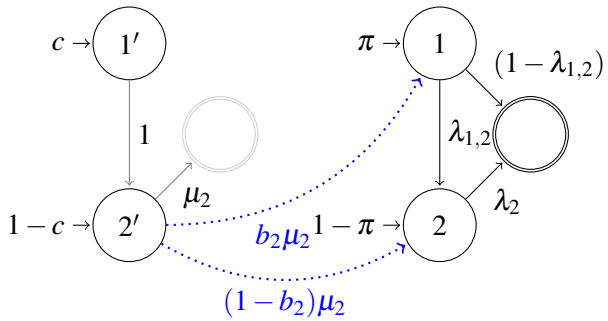
which satisfies the condition $\mathbf{H}_{ij}\mathbb{1} = -\mathbf{D}_i\mathbb{1}$, since no absorption from the state $1'$ can occur. We obtain $\mathbf{cM} = \left(c,\ \frac{1}{\mu_2}\right)$. The condition $\mathbf{cMH}_{ij} = \pi$ should be satisfied to keep the initial distribution of the PH_j invariant. We obtain

$$(c \quad \frac{1}{\mu_2}) \cdot \begin{pmatrix} 0 & 0 \\ b_2\mu_2 & (1-b_2)\mu_2 \end{pmatrix} = (\pi \quad (1-\pi)),$$

such that the following has to hold

$$b_2 = \pi^{\delta}, \text{ and } 1-b_2 = 1-\pi^{\delta},$$

where b_i gives the probability that the PH_j starts in state 1 after i' was the last state before absorption in the PH_i. Since PH_i is in series canonical form it cannot get absorbed directly from state $1'$, thus resulting in $b_1 = 0$. Thus PH_i enables no flexibily for reaching a larger first joint moment. In that case the representation of the PH_j has



(a) A graphical representation of the APHDs $(\mathbf{c}, \mathbf{D}_i)$, $(\pi, \mathbf{D}_j)$.

(b) Possible elementary series for both APHDs.

(1)	$< s_1' \; s_2' s_1 >$
(2)	$< s_1' \; s_2' \; s_2 >$
(3)	$< s_1' \; s_2' \; s_1 \; s_2 >$
(4)	$< s_2' \; s_1 >$
(5)	$< s_2' \; s_1 \; s_2 >$
(6)	$< s_2' \; s_2 >$

Figure C.1.: Symbolic representation of the two APHDs in composition for correlation fitting.

no influence on the range of the correlation, i.e., the transformation parameter δ can be arbitrary chosen having no influence on the first joint moment. It can also be shown that the value of derivative of $\mu_{1,1}(\delta)$ with respect to δ is zero.

Figure C.1 visualizes all possible paths in the composition of two given APHDs. Note that the probability of each elementary series can be determined as the product of the initial probability of the series and the transition rates along the elementary series. Then the first joint moment can be computed as the mixture of mean durations of all elementary series which are weighted proportionally to its probability. The weighted mean durations of the series listed in Fig. 6.1(b) are listed below

$$\begin{aligned}
b_2\, c\, (1+\tfrac{1}{\mu_2})\, a_{1,1}^{\delta}\, \nu_{1,1} &\quad \text{for the series (1)},\\
(1-b_2)\, c\, (1+\tfrac{1}{\mu_2})\, a_{2,2}^{\delta} \nu_{1,2} &\quad \text{for the series (2)},\\
b_2\, c\, (1+\tfrac{1}{\mu_2})\, a_{1,2}^{\delta} \nu_{1,2} &\quad \text{for the series (3)},\\
b_2\, (1-c)\, \tfrac{1}{\mu_2}\, a_{1,1}^{\delta}\, \nu_{1,1} &\quad \text{for the series (4)},\\
b_2\, (1-c)\, \tfrac{1}{\mu_2}\, a_{1,2}^{\delta}\, \nu_{1,2} &\quad \text{for the series (5)},\\
(1-b_2)\, (1-c)\, \tfrac{1}{\mu_2}\, a_{2,2}^{\delta} \nu_{2,2} &\quad \text{for the series (6)}.
\end{aligned}$$

Using the notations for elementary series the first joint moment can be written as

$$\begin{aligned}
\mu_{1,1}(\delta) \;&= b_2\, c \left(1+\tfrac{1}{\mu_2}\right) \left(a_{1,1}^{\delta}\nu_{1,1} + a_{1,2}^{\delta}\nu_{1,2}\right)/\pi^{\delta} + \\
&\quad\; b_2\, (1-c)\, \tfrac{1}{\mu_2} \left(a_{1,1}^{\delta}\nu_{1,1} + a_{1,2}^{\delta}\nu_{1,2}\right)/\pi^{\delta} + \\
&\quad\; (1-b_2)\, c \left(1+\tfrac{1}{\mu_2}\right) \nu_{2,2} + (1-b_2)\, (1-c)\, \tfrac{1}{\mu_2}\, \nu_{2,2}.
\end{aligned}$$

After substituting the values given in (C.1), (6.13), we obtain

$$\begin{aligned}
\mu_{1,1}(\delta) \;&= b_2\, c\, (1+\tfrac{1}{\mu_2})\, ((\pi+\delta)(1-\lambda_{1,2}^{\delta})\, \tfrac{1}{\lambda_1} + (\pi+\delta)\, \lambda_{1,2}^{\delta}\, (1+\tfrac{1}{\lambda_2}))/\pi^{\delta} + \\
&\quad\; b_2\, (1-c)\, \tfrac{1}{\mu_2}\, ((\pi+\delta)(1-\lambda_{1,2}^{\delta})\, \tfrac{1}{\lambda_1} + (\pi+\delta)\, \lambda_{1,2}^{\delta}\, (1+\tfrac{1}{\lambda_2}))/\pi^{\delta} + \\
&\quad\; (1-b_2)\, c\, (1+\tfrac{1}{\mu_2})\, \tfrac{1}{\lambda_2} + (1-b_2)\, (1-c)\, \tfrac{1}{\mu_2}\, \tfrac{1}{\lambda_2}.
\end{aligned}$$

The terms in the above sum are now denoted as x_1, x_2, x_3, and x_4. The sum of the x_3 and x_4 results in

$$x_3+x_4=(1-b_2)\frac{1}{\lambda_2}\left(c\frac{(\mu_2+1)}{\mu_2}+\frac{1}{\mu_2}\right)=(1-b_2)\frac{(c\mu_2+1)}{\lambda_2\mu_2}=\frac{(c\mu_2+1)}{\lambda_2\mu_2}-b_2\frac{(c\mu_2+1)}{\lambda_2\mu_2}.$$

The term in the brackets of x_1 and x_2 is denoted as x_5. It results in

$$\begin{aligned}
x_5 &= ((\pi+\delta)(1-\lambda_{1,2}^{\delta})\,\tfrac{1}{\lambda_1}+(\pi+\delta)\,\lambda_{1,2}^{\delta}\,(1+\tfrac{1}{\lambda_2})) \\
&= (\pi+\delta)(1-\lambda_{1,2}^{\delta})+(\pi+\delta)\,\lambda_{1,2}^{\delta}\,(1+\tfrac{1}{\lambda_2}) \\
&= \pi\lambda_{1,2}-(\lambda_2-1)\delta+\tfrac{\pi\lambda_{1,2}-(\lambda_2-1)\delta}{\lambda_2}+\pi+\delta-\pi\lambda_{1,2}+\delta(\lambda_2-1) \\
&= \tfrac{\pi\lambda_{1,2}-(\lambda_2-1)\delta+\pi\lambda_2+\delta\lambda_2}{\lambda_2} \\
&= \tfrac{\pi(\lambda_{1,2}+\lambda_2)+\delta}{\lambda_2}.
\end{aligned}$$

The sum of the x_1 and x_2 results in

$$\begin{aligned}
x_1+x_2 &= \tfrac{b_2(1-c)\frac{1}{\mu_2}x_5}{\pi+\delta}+\tfrac{b_2c\frac{(\mu_2+1)}{\mu_2}x_5}{\pi+\delta} \\
&= \tfrac{1}{(\pi+\delta)\mu_2}\left[b_2x_5(1-c+c(\mu_2+1))\right] \\
&= b_2\left[\tfrac{(1+c\mu_2)}{(\pi+\delta)\mu_2}x_5\right] \\
&= b_2\left[\tfrac{(1+c\mu_2)}{(\pi+\delta)\mu_2}\tfrac{(\pi(\lambda_{1,2}+\lambda_2)+\delta)}{\lambda_2}\right] \\
&= b_2\left[\tfrac{1}{(\pi+\delta)}\tfrac{(\pi(\lambda_2+\lambda_{1,2}))y_1}{\lambda_2\mu_2}+\tfrac{\delta}{(\pi+\delta)}\tfrac{y_1}{\lambda_2\mu_2}\right],
\end{aligned}$$

where $y_1=(1+c\mu_2)$. With this notations $\mu_{1,1}(\delta)$ can be written as

$$\begin{aligned}
\mu_{1,1}(\delta) &= y_2+b_2\left[\tfrac{1}{(\pi+\delta)}\tfrac{(\pi(\lambda_2+\lambda_{1,2}))y_1}{\lambda_2\mu_2}+\tfrac{\delta}{(\pi+\delta)}\tfrac{y_1}{\lambda_2\mu_2}-y_2\right] \\
&= y_2+b_2y_2\left[\tfrac{1}{(\pi+\delta)}(\pi(\lambda_2+\lambda_{1,2}))+\tfrac{\delta}{(\pi+\delta)}-1\right] \\
y_1 &= (1+c\mu_2) \\
y_2 &= \tfrac{c\mu_2+1}{\lambda_2\mu_2}=\tfrac{y_1}{\lambda_2\mu_2}.
\end{aligned}$$

We now denote the term in the brackets as $y_3(\delta)$ such that

$$\begin{aligned}
y_3(\delta) &= y_2\left[\tfrac{1}{(\pi+\delta)}(\pi(\lambda_2+\lambda_{1,2}))+\tfrac{\delta}{(\pi+\delta)}-1\right] \\
&= y_2\left[\tfrac{\pi(\lambda_2+\lambda_{1,2})+\delta-(\pi+\delta)}{(\pi+\delta)}\right] \\
&= y_2\left[\tfrac{\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)}\right],
\end{aligned}$$

which is ≥ 0, since $(\pi+\delta)>0$, $\pi>0$, $\lambda_2>1$ and $(\lambda_{1,2}+\lambda_2)>1$. Observe that if we consider the case with $\lambda_2=\lambda_1=1$, then the maximal reachable correlation is always equal to 0 because the PH_j represents the exponential distribution with rate 1. Thus, such a case is not interesting and we assume $\lambda_2\neq\lambda_1$, i.e. $\lambda_2>1$. Note that the probability b_2 equals to $\pi+\delta$, such that its derivative equals

$$\frac{\partial b_2(\delta)}{\partial\delta}=1.$$

Now consider the derivative of $y_3(\delta)$

$$\begin{aligned}
\tfrac{\partial y_3(\delta)}{\partial \delta} &= \tfrac{\partial}{\partial \delta}\left(\tfrac{1}{(\pi+\delta)}(\pi(\lambda_2+\lambda_{1,2}))y_2+\tfrac{\delta}{(\pi+\delta)}y_2-y_2\right)\\
&= -\tfrac{\pi(\lambda_{1,2}+\lambda_2)y_2}{(\pi+\delta)^2}+\tfrac{y_2\pi}{(\pi+\delta)^2}\\
&= -\tfrac{y_2\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2} < 0\\
\text{with}\quad &\tfrac{\partial y_2}{\partial \delta} = \tfrac{\partial}{\partial \delta}(\tfrac{c\mu_2+1}{\lambda_2\mu_2}) = 0,\\
&\tfrac{\partial}{\partial \delta}(\tfrac{1}{(\pi+\delta)}\pi(\lambda_{1,2}+\lambda_2)y_2) = -\tfrac{\pi(\lambda_{1,2}+\lambda_2)y_2}{(\pi+\delta)^2},\\
&\tfrac{\partial}{\partial \delta}\left(\tfrac{\delta y_2}{\pi+\delta}\right) = \tfrac{y_2(\pi+\delta)-\delta y_2}{(\pi+\delta)^2} = \tfrac{y_2\pi}{(\pi+\delta)^2}.
\end{aligned}$$

Then the derivative of $\mu_{1,1}(\delta)$ equals

$$\begin{aligned}
\tfrac{\partial \mu_{1,1}(\delta)}{\partial \delta} &= \tfrac{\partial}{\partial \delta}(y_2+b_2(\delta)y_3(\delta))\\
&= \tfrac{\partial}{\partial \delta}y_2+\tfrac{\partial}{\partial \delta}(b_2(\delta)y_3(\delta))\\
&= 0+y_3(\delta)+b_2(\delta)(-\tfrac{y_2\pi(\lambda_{1,2}+\lambda_2-1)}{(\pi+\delta)^2})\\
&= y_2\left(\tfrac{1}{(\pi+\delta)}(\pi(\lambda_2+\lambda_{1,2}))+\tfrac{\delta}{(\pi+\delta)}-1\right)-\tfrac{(y_2\pi(\lambda_{1,2}+\lambda_2-1))}{(\pi+\delta)}\\
&= \tfrac{y_2}{(\pi+\delta)}(\pi(\lambda_2+\lambda_{1,2})+\delta-(\pi+\delta)-\pi(\lambda_{1,2}+\lambda_2-1)) = 0.
\end{aligned}$$

□

Proof of Theorem 6.3. We write the first joint moment $\mu_{1,1}$ from Eq. 6.11 in terms of all possible paths in the PH_i following all possible paths in the PH_j. Let $a_{i,j}^{\delta}$ be the probability that the second APHD PH_j starts in the state i and j is the last state before absorption. Since PH_j is in hyperexponential form, it holds that

$$a_{1,1}=\pi,\ a_{1,2}=0,\ a_{2,2}=1-\pi \text{ and } a_{2,1}=0, \tag{C.3}$$

Let $d_{i,j}^{\delta}$ be the probability that the first APHD PH_i starts in the state i' and j' is the last state before absorption. It holds that

$$d_{1,1}^{\delta}=c^{\delta}(1-\mu_{1,2}^{\delta}),\ d_{1,2}^{\delta}=c^{\delta}\mu_{1,2}^{\delta},\ d_{2,2}^{\delta}=1-c^{\delta} \text{ and } d_{2,1}^{\delta}=0, \tag{C.4}$$

We use $\nu_{i,j}$ originally defined in [41] which denotes the mean duration if the process starts in state i and state j is the last state before absorption of the PH_j. It holds that

$$\nu_{1,1}=\frac{1}{\lambda_1}=1, \text{ and } \nu_{2,2}=\frac{1}{\lambda_2}. \tag{C.5}$$

Analogously we define $e_{i,j}$ as the mean duration if the process starts in state i' and state j' is the last state before absorption of the PH_i. Then we obtain

$$e_{1,1}=\frac{1}{\mu_1}=1, e_{1,2}=1+\frac{1}{\mu_2}, \text{ and } e_{2,2}=\frac{1}{\mu_2}. \tag{C.6}$$

For the transfer matrix $\mathbf{H}_{ij}$ it must hold that $\mathbf{H}_{ij}\mathbb{1}=-\mathbf{D}_i\mathbb{1}$. The probability b_i gives the probability that the PH_j starts in state 1 after i' was the last state before absorption in the PH_i. The transfer matrix $\mathbf{H}_{ij}$ can be written as

$$\mathbf{H}_{ij}=\begin{pmatrix} b_1(1-\mu_{1,2}) & (1-b_1)(1-\mu_{1,2}) \\ b_2\mu_2 & (1-b_2)\mu_2 \end{pmatrix}. \tag{C.7}$$

Furthermore the condition $\mathbf{cMH}_{ij} = \pi$ should be satisfied to keep the initial distribution of the PH_j invariant. For $\mathbf{c}(-\mathbf{D}_i)^{-1} = \mathbf{cM}$ we obtain

$$(c \quad 1-c) \cdot \begin{pmatrix} 1 & \frac{\mu_{1,2}}{\mu_2} \\ 0 & \frac{1}{\mu_2} \end{pmatrix} = (c \quad \frac{(c(\mu_{1,2}-1)+1)}{\mu_2}),$$

then the following has to hold

$$(c \quad \frac{(c(\mu_{1,2}-1)+1)}{\mu_2}) \cdot \begin{pmatrix} b_1(1-\mu_{1,2}) & (1-b_1)(1-\mu_{1,2}) \\ b_2\mu_2 & (1-b_2)\mu_2 \end{pmatrix} = (\pi \quad (1-\pi)).$$

In particular the following relation has to hold,

$$cb_1(1-\mu_{1,2}) + b_2(c(\mu_{1,2}-1)+1) = \pi,$$

such that

$$b_2 = \frac{\pi - cb_1(1-\mu_{1,2})}{c\mu_{1,2} - c + 1}.$$

In the case that $b_1 = 0$, the probability $b_2 = \frac{\pi}{c\mu_{1,2}-c+1}$. To ensure that $0 \leq b_2 \leq 1$, the probability b_2 should be set to $min(1, \frac{\pi}{c\mu_{1,2}-c+1})$.

In the case that $b_1 = 1$ the probability $b_2 = \frac{\pi - cb_1(1-\mu_{1,2})}{c\mu_{1,2}-c+1}$ and should be set to $max(0, \frac{\pi - cb_1(1-\mu_{1,2})}{c\mu_{1,2}-c+1})$. The probability b_2 can be chosen from the interval

$$\left[\max(0, \frac{\pi - c(1-\mu_{1,2})}{c\mu_{1,2}-c+1}), \min(1, \frac{\pi}{c\mu_{1,2}-c+1})\right], \tag{C.8}$$

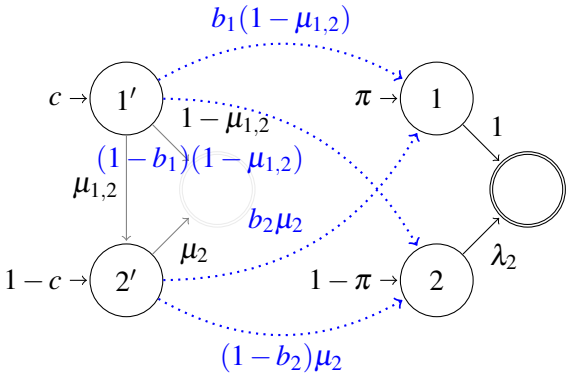
and the probabilty b_1 can be expressed in terms of b_2 as $b_1 = \frac{\pi - b_2(c\mu_{1,2}-c+1)}{c(1-\mu_{1,2})}$.

The composition of the APHDs where the PH_j is fixed to the hyperexponential representation is visualized in Fig. C.2. We compute the following mean durations for the elementary series listed in Fig. C.2(b)

$$\begin{aligned}
d^{\delta}_{1,1}\, e_{1,1}\, b_1\, \tfrac{1}{\lambda_1} &\quad \text{for the series } (1),\\
d^{\delta}_{1,2}\, e_{1,2}\, b_2\, \tfrac{1}{\lambda_1} &\quad \text{for the series } (2),\\
d^{\delta}_{1,2}\, e_{1,2}\, (1-b_2)\, \tfrac{1}{\lambda_2} &\quad \text{for the series } (3),\\
d^{\delta}_{1,1}\, e_{1,1}\, (1-b_1)\, \tfrac{1}{\lambda_2} &\quad \text{for the series } (4),\\
d^{\delta}_{2,2}\, e_{2,2}\, b_2\, \tfrac{1}{\lambda_1} &\quad \text{for the series } (5),\\
d^{\delta}_{2,2}\, e_{2,2}\, (1-b_2)\, \tfrac{1}{\lambda_2} &\quad \text{for the series } (6).
\end{aligned}$$

With these notations for the elementary series the first joint moment can be expressed as

$$\begin{aligned}
\mu_{1,1}(\delta) \quad &= b_2\, \tfrac{1}{\lambda_1}\, (d^{\delta}_{1,2}\, e_{1,2} + d^{\delta}_{2,2}\, e_{2,2}) +\\
&\quad (1-b_2)\, \tfrac{1}{\lambda_2}\, (d^{\delta}_{1,2}\, e_{1,2} + d^{\delta}_{2,2}\, e_{2,2}) +\\
&\quad (1-b_1)\, \tfrac{1}{\lambda_2}\, d^{\delta}_{1,1}\, e_{1,1} + b_1\, \tfrac{1}{\lambda_1}\, d^{\delta}_{1,1}\, e_{1,1}.
\end{aligned}$$



(a) A graphical representation of the APHDs $(\mathbf{c}, \mathbf{D}_i)$, $(\pi, \mathbf{D}_j)$.

(1) $< s_1' \; s_1 >$
(2) $< s_1' \; s_2' \; s_1 >$
(3) $< s_1' \; s_2' \; s_2 >$
(4) $< s_1' \; s_2 >$
(5) $< s_2' \; s_1 >$
(6) $< s_2' \; s_2 >$

(b) Possible elementary series for both APHDs.

Figure C.2.: Symbolic representation of the two given APHDs in composition for correlation fitting.

After substituting the values given in (C.3), (C.4), (C.5), (C.6), we obtain

$$\begin{aligned}\mu_{1,1}(\delta) \quad &= b_2 \left((c+\delta)(\tfrac{c\mu_{1,2}}{(c+\delta)} - \tfrac{(\mu_2-1)\delta}{(c+\delta)})(1+\tfrac{1}{\mu_2}) + (1-c-\delta)\tfrac{1}{\mu_2} \right) + \\ &(1-b_2)\ \tfrac{1}{\lambda_2} \left((c+\delta)(\tfrac{c\mu_{1,2}}{(c+\delta)} - \tfrac{(\mu_2-1)\delta}{(c+\delta)})(1+\tfrac{1}{\mu_2}) + (1-c-\delta)\tfrac{1}{\mu_2} \right) + \\ &(1-b_1)\ \tfrac{1}{\lambda_2}(c+\delta) \left(1 - (\tfrac{c\mu_{1,2}}{(c+\delta)} - \tfrac{(\mu_2-1)\delta}{(c+\delta)}) \right) + \\ &b_1\ (c+\delta) \left(1 - (\tfrac{c\mu_{1,2}}{(c+\delta)} - \tfrac{(\mu_2-1)\delta}{(c+\delta)}) \right).\end{aligned}$$

We denote the terms in the sum as x_1, x_2, x_3, and x_4. We first simplify the following term

$$\begin{aligned} x_5 \quad &= (c+\delta) \left(1 - (\tfrac{c\mu_{1,2}}{(c+\delta)} - \tfrac{(\mu_2-1)\delta}{(c+\delta)}) \right) \\ &= c(1-\mu_{1,2}) + \delta\mu_2. \end{aligned}$$

It results for the sum of terms x_3 and x_4

$$\begin{aligned} x_3 + x_4 \quad &= (1-b_1)\tfrac{1}{\lambda_2} x_5 + b_1 x_5 = \tfrac{1}{\lambda_2} x_5 - \tfrac{1}{\lambda_2} b_1 x_5 + b_1 x_5 \\ &= \tfrac{c(1-\mu_{1,2})+\delta\mu_2}{\lambda_2} + b_1(1-\tfrac{1}{\lambda_2})(c(1-\mu_{1,2})+\delta\mu_2). \end{aligned}$$

We now express the probability $b_1(\delta)$ in terms of b_2

$$\begin{aligned} b_1(\delta) \quad &= \tfrac{\pi - b_2(c^\delta \mu_{1,2}^\delta - c^\delta + 1)}{c^\delta(1-\mu_{1,2}^\delta)} \\ &= \tfrac{\pi - b_2(c(1-\mu_{1,2}) - \delta\mu_2 + 1)}{c(1-\mu_{1,2})+\delta\mu_2}. \end{aligned}$$

Using this notations the sum $x_3 + x_4$ can be further simplified

$$\begin{aligned} x_3 + x_4 \quad &= \tfrac{c(1-\mu_{1,2})+\delta\mu_2}{\lambda_2} + \left(\tfrac{\lambda_2-1}{\lambda_2}\right) b_1(c(1-\mu_{1,2})+\delta\mu_2) \\ &= \tfrac{c(1-\mu_{1,2})+\delta\mu_2}{\lambda_2} + \pi\left(\tfrac{\lambda_2-1}{\lambda_2}\right) - \left(\tfrac{\lambda_2-1}{\lambda_2}\right) b_2(c(\mu_{1,2}-1) - \delta\mu_2 + 1). \end{aligned}$$

Then

$$\begin{aligned} x_7 &= \tfrac{c(1-\mu_{1,2})+\delta\mu_2}{\lambda_2} + \pi\left(\tfrac{\lambda_2-1}{\lambda_2}\right) \\ &= \tfrac{c-c\mu_{1,2}+\delta\mu_2+\pi\lambda_2-\pi}{\lambda_2}. \end{aligned}$$

We compute now the term

$$\begin{aligned} x_6 &= \left((c+\delta)(\tfrac{c\mu_{1,2}}{(c+\delta)} - \tfrac{(\mu_2-1)\delta}{(c+\delta)})(1+\tfrac{1}{\mu_2}) + (1-c-\delta)\tfrac{1}{\mu_2}\right) \\ &= (c\mu_{1,2} - \delta\mu_2 + \delta)(1+\tfrac{1}{\mu_2}) + (1-c-\delta)\tfrac{1}{\mu_2} \\ &= c\mu_{1,2} + \tfrac{1}{\mu_2}\left[c\mu_{1,2} + 1 - c\right] - \delta\mu_2. \end{aligned}$$

It results for the sum of terms x_1 and x_2

$$\begin{aligned} x_1 + x_2 &= b_2 x_6 + (1-b_2)\tfrac{1}{\lambda_2} x_6 \\ &= \tfrac{1}{\lambda_2} x_6 + \left(\tfrac{\lambda_2-1}{\lambda_2}\right) b_2 x_6, \end{aligned}$$

where we denote the first term in the sum as x_8

$$x_8 = \frac{x_6}{\lambda_2} = \frac{c\mu_{1,2} + \frac{1}{\mu_2}\left[c\mu_{1,2} + 1 - c\right] - \delta\mu_2}{\lambda_2}.$$

Then the sum of x_7 and x_8 results in

$$\begin{aligned} x_7 + x_8 &= \tfrac{c-c\mu_{1,2}+\delta\mu_2+\pi\lambda_2-\pi}{\lambda_2} + \tfrac{c\mu_{1,2}+\frac{1}{\mu_2}[c\mu_{1,2}+1-c]-\delta\mu_2}{\lambda_2} \\ &= \tfrac{c+\pi\lambda_2-\pi+\frac{1}{\mu_2}c\mu_{1,2}+\frac{1}{\mu_2}-\frac{c}{\mu_2}}{\lambda_2} \\ &= \tfrac{c\mu_2+c\mu_{1,2}-c+1}{\lambda_2\mu_2} + \tfrac{\pi(\lambda_2-1)}{\lambda_2}. \end{aligned}$$

Now consider the remaining terms from the sums $x_1 + x_2$ and $x_3 + x_4$

$$\begin{aligned} &\left(\tfrac{\lambda_2-1}{\lambda_2}\right) b_2 x_6 - \left(\tfrac{\lambda_2-1}{\lambda_2}\right) b_2 (c(\mu_{1,2}-1) - \delta\mu_2 + 1) \\ &\left(\tfrac{\lambda_2-1}{\lambda_2}\right) b_2 (x_6 - (c(\mu_{1,2}-1) - \delta\mu_2 + 1)), \end{aligned}$$

such that

$$\begin{aligned} &x_6 - (c(\mu_{1,2}-1) - \delta\mu_2 + 1) &= \\ &c\mu_{1,2} + \tfrac{1}{\mu_2}\left[c\mu_{1,2} + 1 - c\right] - \delta\mu_2 - c(\mu_{1,2}-1) + \delta\mu_2 - 1 &= \\ &\tfrac{\mu_2(c-1)+c\mu_{1,2}+1-c}{\mu_2} \end{aligned}$$

Now the expression for the first joint moment can be rewritten as

$$\begin{aligned} \mu_{1,1}(\delta) &= \tfrac{y_2+(\lambda_2-1)b_2 y_3}{\lambda_2\mu_2}, \\ \\ y_2 &= c\mu_2 + c\mu_{1,2} - c + 1 + \pi\mu_2(\lambda_2-1), \\ y_3 &= \mu_2(c-1) + c\mu_{1,2} + 1 - c. \end{aligned}$$

The probability b_2 can take values from the interval given in Eq. 6.14. However, b_2 depends on δ such that the boundaries for $b_2(\delta)$ result in

$$\left[\max\left(0, \frac{\pi - c\,(1-\mu_{1,2}) - \delta\mu_2}{c(\mu_{1,2}-1) - \delta\mu_2 + 1}\right), \min\left(1, \frac{\pi}{c(\mu_{1,2}-1) - \delta\mu_2 + 1}\right)\right]. \tag{C.9}$$

Now consider that if $y_3 > 0$ b_2 should be as large as possible to maximize $\mu_{1,1}(\delta)$, and if $y_3 < 0$ b_2 should be as small as possible to maximize $\mu_{1,1}(\delta)$. On the other hand, if $y_3 = 0$ then the first joint moment $\mu_{1,1}(\delta)$ cannot be optimized, i.e. $\mu_{1,1}(\delta) = \frac{y_2}{\lambda_2\mu_2}$.

Note that $y_3 > 0$ for $c > \frac{\mu_2-1}{\mu_2-1+\mu_{1,2}}$, $y_3 < 0$ for $c < \frac{\mu_2-1}{\mu_2-1+\mu_{1,2}}$, and $y_3 = 0$ for $c = \frac{\mu_2-1}{\mu_2-1+\mu_{1,2}}$. Thus we should consider first two cases.

In **Case 1** we have $c < \frac{\mu_2-1}{\mu_2-1+\mu_{1,2}}$. Then $y_3 < 0$ and $b_2(\delta)$ should be set to minimum, i.e. $b_2(\delta) = \frac{\pi-c(1-\mu_{1,2})-\delta\mu_2}{c(\mu_{1,2}-1)-\delta\mu_2+1}$, and $b_1 = 1$. The derivative with respect to δ equals

$$\frac{\partial b_2(\delta)}{\partial \delta} = \frac{\mu_2(\pi-1)}{(-\delta\mu_2 + c(\mu_{1,2}-1)+1)^2} < 0.$$

The denominator is always positive, and the numerator is negative since we assume $\pi < 1$ in the two-phase case. The derivative of the $\mu_{1,1}(\delta)$ with respect to δ is

$$\begin{aligned}\frac{\partial\mu_{1,1}(\delta)}{\partial\delta} &= \frac{[y_2+y_3(\lambda_2-1)b_2(\delta)]'}{\lambda_2\mu_2} \\ &= \frac{y_3(\lambda_2-1)}{\lambda_2\mu_2}\frac{db_2(\delta)}{d\delta} \\ &= \frac{y_3(\lambda_2-1)}{\lambda_2\mu_2}\frac{\mu_2(\pi-1)}{(-\delta\mu_2+c(\mu_{1,2}-1)+1)^2} > 0.\end{aligned}$$

The second term is negative as shown above, and the y_3 is negative by assumption. In the first case, i.e. as long as $c < \frac{\mu_2-1}{\mu_2-1+\mu_{1,2}}$ the derivative with respect to δ is positive such that δ should be increased to increase $\mu_{1,1}(\delta)$. We set δ to the maximum value such that $\delta = \frac{c\mu_{1,2}}{\mu_2-1}$. The transformation then results in

$$\begin{aligned}\mu^{\delta}_{1,2} &= \frac{c\mu_{1,2}}{(c+\delta)} - \frac{(\mu_2-1)\delta}{(c+\delta)} \\ &= \frac{c\mu_{1,2}}{(c+\frac{c\mu_{1,2}}{\mu_2-1})} - \frac{(\mu_2-1)\frac{c\mu_{1,2}}{\mu_2-1}}{(c+\frac{c\mu_{1,2}}{\mu_2-1})} = 0,\end{aligned}$$

which corresponds to the hyperexponential representation of the PH_i where the maximal value for $\mu_{1,1}$ can be reached.

The **Case 2** is given when $c > \frac{\mu_2-1}{\mu_2-1+\mu_{1,2}}$. Then $y_3 > 0$, and the probability b_2 should be set to maximum to maximize the first joint moment. So we set
$b_2 = \frac{\pi}{c(\mu_{1,2}-1)-\delta\mu_2+1}$. The derivative with respect to δ equals

$$\frac{\partial b_2(\delta)}{\partial \delta} = \frac{\mu_2\pi}{(-\delta\mu_2 + c(\mu_{1,2}-1)+1)^2} > 0,$$

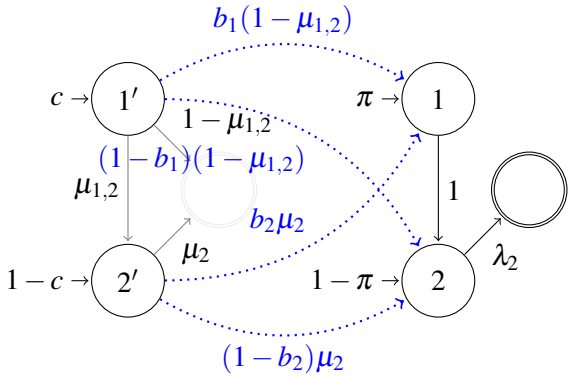
since the denominator is always positive. The derivative of the $\mu_{1,1}(\delta)$ with respect to δ is then

$$\begin{aligned}\frac{\partial\mu_{1,1}(\delta)}{\partial\delta} &= \frac{[y_2+y_3(\lambda_2-1)b_2(\delta)]'}{\lambda_2\mu_2} \\ &= \frac{y_3(\lambda_2-1)}{\lambda_2\mu_2}\frac{db_2(\delta)}{d\delta} \\ &= \frac{y_3(\lambda_2-1)}{\lambda_2\mu_2}\frac{\mu_2\pi}{(-\delta\mu_2+c(\mu_{1,2}-1)+1)^2} > 0,\end{aligned}$$

which is positive since $y_3 > 0$ by assumption. In that case δ should again be set to the maximum value to maximize $\mu_{1,1}$ which implies the hyperexponential representation of the PH_i.

Observe that in the second case the probability b_2 is selected from $\min\left(1, \frac{\pi}{c(\mu_{1,2}-1)-\delta\mu_2+1}\right)$. If $\frac{\pi}{c(\mu_{1,2}-1)-\delta\mu_2+1} > 1$, the probability $b_2 = 1$, which implies that the derivative of $\mu_{1,1}(\delta)$ is zero. □

Proof of Theorem 6.4. Again we write the first joint moment $\mu_{1,1}$ from Eq. 6.11 in terms of all possible paths in the PH_i following all possible paths in the PH_j. The probabilities $d^{\delta}_{i,j}$ are given by Eq. C.4. The mean durations are defined as in Eq. C.6,and Eq. C.5. The transfer matrix is given by Eq. C.7 such that the bounds for probability b_2 are given by Eq. C.8. The elementary series of the composition where the second APHD is fixed to canonical form are visualized in Fig. C.3.



(a) A graphical representation of the APHDs $(\mathbf{c}, \mathbf{D}_i)$, $(\pi, \mathbf{D}_j)$.

(1) $< s'_1\ s_1\ s_2 >$
(2) $< s'_2\ s_1\ s_2 >$
(3) $< s'_1\ s'_2\ s_1\ s_2 >$
(4) $< s'_1\ s'_2\ s_2 >$
(5) $< s'_1\ s_2 >$
(6) $< s'_2\ s_2 >$

(b) Possible elementary series for the composition of two given APHDs.

Figure C.3.: Symbolic representation of the two given APHDs in composition for correlation fitting.

We compute the following mean durations for the elementary series listed in Fig. C.3(b)

$$\begin{array}{ll}
d^{\delta}_{1,1}\, e_{1,1}\, b_1\, (1+\frac{1}{\lambda_2}) & \text{for the series } (1),\\
d^{\delta}_{2,2}\, e_{2,2}\, b_2\, (1+\frac{1}{\lambda_2}) & \text{for the series } (2),\\
d^{\delta}_{1,2}\, e_{1,2}\, b_2\, (1+\frac{1}{\lambda_2}) & \text{for the series } (3),\\
d^{\delta}_{1,2}\, e_{1,2}\, (1-b_2)\, \frac{1}{\lambda_2} & \text{for the series } (4),\\
d^{\delta}_{1,1}\, e_{1,1}\, (1-b_1)\, \frac{1}{\lambda_2} & \text{for the series } (5),\\
d^{\delta}_{2,2}\, e_{2,2}\, (1-b_2)\, \frac{1}{\lambda_2} & \text{for the series } (6).
\end{array}$$

With these notations for the elementary series the first joint moment can be expressed as

$$\begin{aligned}
\mu_{1,1}(\delta) \quad &= b_2\, (d^{\delta}_{2,2}\, e_{2,2}\, \nu_{1,2} + d^{\delta}_{1,2}\, e_{1,2}\, \nu_{1,2}) +\\
&\quad (1-b_2)\, (d^{\delta}_{1,2}\, e_{1,2}\, \nu_{2,2} + d^{\delta}_{2,2}\, e_{2,2}\, \nu_{2,2}) +\\
&\quad (1-b_1)\, d^{\delta}_{1,1}\, e_{1,1}\, \nu_{2,2} + b_1\, d^{\delta}_{1,1}\, e_{1,1}\, \nu_{1,2}.
\end{aligned}$$

We substitute values given in (C.3), (C.4), (C.5), (C.6), and obtain

$$\begin{aligned}\mu_{1,1}(\delta) \quad &= b_2(1+\tfrac{1}{\lambda_2})\left((c+\delta)(\tfrac{c\mu_{1,2}}{(c+\delta)}-\tfrac{(\mu_2-1)\delta}{(c+\delta)})(1+\tfrac{1}{\mu_2})+(1-c-\delta)\tfrac{1}{\mu_2}\right)+\\ &(1-b_2)\,\tfrac{1}{\lambda_2}\left((c+\delta)(\tfrac{c\mu_{1,2}}{(c+\delta)}-\tfrac{(\mu_2-1)\delta}{(c+\delta)})(1+\tfrac{1}{\mu_2})+(1-c-\delta)\tfrac{1}{\mu_2}\right)+\\ &(1-b_1)\,\tfrac{1}{\lambda_2}(c+\delta)\left(1-(\tfrac{c\mu_{1,2}}{(c+\delta)}-\tfrac{(\mu_2-1)\delta}{(c+\delta)})\right)+\\ &b_1\,(1+\tfrac{1}{\lambda_2})\,(c+\delta)\left(1-(\tfrac{c\mu_{1,2}}{(c+\delta)}-\tfrac{(\mu_2-1)\delta}{(c+\delta)})\right).\end{aligned}$$

First the term $x_5 = c(1-\mu_{1,2})+\delta\mu_2$ and can be adopted from the Sec. 6.2.4. Then we simplify the following term

$$\begin{aligned}x_6 \quad &= \left((c+\delta)(\tfrac{c\mu_{1,2}}{(c+\delta)}-\tfrac{(\mu_2-1)\delta}{(c+\delta)})(1+\tfrac{1}{\mu_2})+(1-c-\delta)\tfrac{1}{\mu_2}\right)\\ &= c\mu_{1,2}+\tfrac{1}{\mu_2}\left[c\mu_{1,2}+1-c\right]-\delta\mu_2.\end{aligned}$$

Then the sum of terms x_1 and x_2 results in

$$\begin{aligned}x_1+x_2 \quad &= b_2(1+\tfrac{1}{\lambda_2})x_6+(1-b_2)\tfrac{1}{\lambda_2}x_6\\ &= b_2x_6+b_2x_6\tfrac{1}{\lambda_2}+\tfrac{1}{\lambda_2}x_6-b_2x_6\tfrac{1}{\lambda_2}\\ &= b_2x_6+\tfrac{1}{\lambda_2}x_6.\end{aligned}$$

The sum of x_3 and x_4 results in

$$\begin{aligned}x_3+x_4 \quad &= \tfrac{x_5}{\lambda_2}+b_1x_5\\ &= \tfrac{(c(1-\mu_{1,2})+\delta\mu_2)}{\lambda_2}+\tfrac{\pi-b_2(c(1-\mu_{1,2})-\delta\mu_2+1)}{c(1-\mu_{1,2})+\delta\mu_2}(c(1-\mu_{1,2})+\delta\mu_2)\\ &= \tfrac{(c(1-\mu_{1,2})+\delta\mu_2)}{\lambda_2}+\pi-b_2(c(1-\mu_{1,2})-\delta\mu_2+1).\end{aligned}$$

Then we adopt $x_8 = \frac{x_6}{\lambda_2}$ from the Sec. 6.2.4 and resolve

$$\begin{aligned}y_1 \quad &= x_8+\tfrac{(c(1-\mu_{1,2})+\delta\mu_2)}{\lambda_2}+\pi\\ &= \tfrac{c\mu_2+c\mu_{1,2}-c+1}{\lambda_2\mu_2}+\pi.\end{aligned}$$

We also adopt the following substraction from the proof in Sec. 6.2.4

$$y_2 = x_6-(c(\mu_{1,2}-1)-\delta\mu_2+1) \quad = \tfrac{\mu_2(c-1)+c\mu_{1,2}+1-c}{\mu_2}.$$

Using these notation the first joint moment can be written as

$$\mu_{1,1}(\delta) \quad = y_1+b_2*y_2$$

The probability b_2 depends on the transformation parameter δ and can take values from the interval in Eq. C.9. Now consider that if $y_2<0$ then b_2 should be as small as possible to maximize $\mu_{1,1}(\delta)$, and if $y_2>0$ then b_2 should be as large as possible to maximize $\mu_{1,1}(\delta)$. The case $y_2=0$ allows no flexibility in maximizing $\mu_{1,1}(\delta)$ and thus is not interesting. We consider the first two cases in detail.

In **Case 1** we have $y_2 = \frac{\mu_2(c-1)+c\mu_{1,2}+1-c}{\mu_2} < 0$ and $b_2(\delta)$ is set to its minimum value $\frac{\pi-c(1-\mu_{1,2})-\delta\mu_2}{c(\mu_{1,2}-1)-\delta\mu_2+1}$. Its derivative with respect to δ equals

$$\frac{\partial b_2(\delta)}{\partial\delta} = \frac{\mu_2(\pi-1)}{(-\delta\mu_2+c(\mu_{1,2}-1)+1)^2} < 0.$$

The derivative of the $\mu_{1,1}(\delta)$ with respect to δ is

$$\frac{\partial \mu_{1,1}(\delta)}{\partial \delta} = \frac{\mu_2(\pi-1)}{(-\delta\mu_2+c(\mu_{1,2}-1)+1)^2}\frac{\mu_2(c-1)+c\mu_{1,2}+1-c}{\mu_2} > 0.$$

The first term is negative as shown above, and the y_2 is negative by assumption, such that the derivative with respect to δ is positive and δ should be increased to increase $\mu_{1,1}(\delta)$. We set δ to the maximum value such that $\delta = \frac{c\mu_{1,2}}{\mu_2-1}$ which implies $\mu_{1,2}^{\delta} = 0$ as has been shown in Sec. 6.2.4. This result corresponds to the hyperexponential representation of the PH_i.

In **Case 2** it holds that $y_2 = \frac{\mu_2(c-1)+c\mu_{1,2}+1-c}{\mu_2} > 0$ such that $b_2(\delta)$ is set to its maximum value $\frac{\pi}{c(\mu_{1,2}-1)-\delta\mu_2+1}$. The derivative with respect to δ equals

$$\frac{\partial b_2(\delta)}{\partial \delta} = \frac{\mu_2\pi}{(-\delta\mu_2+c(\mu_{1,2}-1)+1)^2} > 0,$$

The derivative of the $\mu_{1,1}(\delta)$ with respect to δ is

$$\frac{\partial \mu_{1,1}(\delta)}{\partial \delta} = \frac{\mu_2\pi}{(-\delta\mu_2+c(\mu_{1,2}-1)+1)^2}\frac{\mu_2(c-1)+c\mu_{1,2}+1-c}{\mu_2} > 0,$$

such that δ should be increased to increase $\mu_{1,1}(\delta)$. This result also corresponds to the hyperexponential representation of the PH_i. □

Proof of Corollary 2. The correlation function is given by

$$\rho(\pi) = \frac{\mu_{1,1} - E[X]E[Y]}{\sqrt{Var[X]Var[Y]}},$$

where X is the APHD $(\mathbf{c}, \mathbf{D}_0^A)$ random variable, and Y is the APHD $(\pi, \mathbf{D}_0^B)$ random variable. Furthermore $Var[X] = E[X^2] - (E[X])^2$, $Var[Y] = E[Y^2] - (E[Y])^2$. The first joint moment $\mu_{1,1}$ is given in Eq. 6.11. We first introduce the vector matrix representation of the correlation function. We distinguish two cases for the first joint moment.

If $(\pi - c) > 0$, i.e. $\pi > c$, the value of the probability b_2 from Eq. 6.14 is set to the $\frac{\pi-c}{1-c}$. In that case we obtain from Eq. 6.15

$$\mu_{1,1} = \frac{\mu_1 - c(\mu_1-\mu_2) - \mu_2\pi + \pi\lambda_2\mu_2 + (\mu_2-\mu_1)(1-\lambda_2)(\pi-c)}{\lambda_2\mu_1\mu_2} \tag{C.10}$$

In the case if $(\pi - c) \leq 0$, i.e. if $\pi \leq c$ the first joint moment from Eq. 6.15 results in

$$\mu_{1,1} = \frac{\mu_1 - c(\mu_1-\mu_2) - \mu_2\pi + \pi\lambda_2\mu_2}{\lambda_2\mu_1\mu_2} \tag{C.11}$$

Note that if $\pi = c$ in the first case the formula given in Eq. C.10 equals Eq. C.11. Using the formula in Eq. C.10 above we obtain the correlation function

$$\begin{aligned}\rho_1(\pi) &= \frac{\frac{\mu_1-c(\mu_1-\mu_2)-\mu_2\pi+\pi\lambda_2\mu_2+(\mu_2-\mu_1)(1-\lambda_2)(\pi-c)}{\lambda_2\mu_1\mu_2} - \left(\frac{c}{\mu_1}+\frac{1-c}{\mu_2}\right)\left(\pi+\frac{1-\pi}{\lambda_2}\right)}{\sqrt{\left(\frac{2c}{\mu_1^2}+\frac{2(1-c)}{\mu_2^2}-\left(\frac{c}{\mu_1}+\frac{1-c}{\mu_2}\right)^2\right)\cdot\left(2\pi+\frac{2(1-\pi)}{\lambda_2^2}-\left(\pi+\frac{1-\pi}{\lambda_2}\right)^2\right)}} \\ &= \frac{\frac{1-c(1-\mu_2)-\mu_2\pi+\pi\lambda_2\mu_2+(\mu_2-1)(1-\lambda_2)(\pi-c)}{\lambda_2\mu_2} - \left(c+\frac{1-c}{\mu_2}\right)\left(\pi+\frac{1-\pi}{\lambda_2}\right)}{\sqrt{\left(2c+\frac{2(1-c)}{\mu_2^2}-\left(c+\frac{1-c}{\mu_2}\right)^2\right)\cdot\left(2\pi+\frac{2(1-\pi)}{\lambda_2^2}-\left(\pi+\frac{1-\pi}{\lambda_2}\right)^2\right)}}\end{aligned} \tag{C.12}$$

for the first case. In the case $(\pi - c) \leq 0$, the formula for the correlation results in

$$\begin{aligned}\rho_2(\pi) &= \frac{\frac{\mu_1 - c(\mu_1-\mu_2) - \mu_2\pi + \pi\lambda_2\mu_2}{\lambda_2\mu_1\mu_2} - \left(\frac{c}{\mu_1} + \frac{1-c}{\mu_2}\right)\left(\pi + \frac{1-\pi}{\lambda_2}\right)}{\sqrt{\left(\frac{2c}{\mu_1^2} + \frac{2(1-c)}{\mu_2^2} - \left(\frac{c}{\mu_1} + \frac{1-c}{\mu_2}\right)^2\right)\cdot\left(2\pi + \frac{2(1-\pi)}{\lambda_2^2} - \left(\pi + \frac{1-\pi}{\lambda_2}\right)^2\right)}} \\ &= \frac{\frac{1 - c(1-\mu_2) - \mu_2\pi + \pi\lambda_2\mu_2}{\lambda_2\mu_2} - \left(c + \frac{1-c}{\mu_2}\right)\left(\pi + \frac{1-\pi}{\lambda_2}\right)}{\sqrt{\left(2c + \frac{2(1-c)}{\mu_2^2} - \left(c + \frac{1-c}{\mu_2}\right)^2\right)\cdot\left(2\pi + \frac{2(1-\pi)}{\lambda_2^2} - \left(\pi + \frac{1-\pi}{\lambda_2}\right)^2\right)}}.\end{aligned} \tag{C.13}$$

We first examine the behavior of the $\rho_1(\pi)$ given in Eq. C.12. Using *maple* we compute its derivative with respect to π and obtain the extreme point

$$\pi = \frac{\lambda_2 - \lambda_2^2 - 1}{\lambda_2 - 1} < 0, \tag{C.14}$$

which is negative, since the numerator is negative. The obtained extreme point is outside the defined range $[0,\ldots,1]$ for π. Furthermore, the value of the first derivative is negative, and the value of the second derivative is negative for the obtained π in Eq. C.14, which indicates the local maxima in that point. However, the value given by Eq. C.14 is negative and thus an invalid probability value. In the range $[0,\ldots,1]$ for π the function has no local extrema. Thus we consider points on the boundaries of the domain, i.e. in the intervall $[c,\ldots,1]$. In particular, for $\pi = 1$ we obtain

$$\begin{aligned}\rho_1(1) &= \frac{\frac{1 - c(1-\mu_2) - \mu_2 + \mu_2\lambda_2 + (\mu_2 - 1)(1-\lambda_2)(1-c)}{\lambda_2\mu_2} - c - \frac{1-c}{\mu_2}}{\sqrt{2c + \frac{2(1-c)}{\mu_2^2} - (c + \frac{1-c}{\mu_2})^2}} \\ &= \frac{\frac{1 - c + c\mu_2 - \mu_2 + \mu_2\lambda_2 - 1 - \mu_2\lambda_2 + c\lambda_2\mu_2 - c\mu_2 + \mu_2 - c\lambda_2 + c + \lambda_2}{\lambda_2\mu_2} - c - \frac{1-c}{\mu_2}}{\sqrt{2c + \frac{2(1-c)}{\mu_2^2} - (c + \frac{1-c}{\mu_2})^2}} \\ &= \frac{\frac{1 - c + c\mu_2 - \mu_2 + \mu_2\lambda_2 - 1 - \mu_2\lambda_2 + c\lambda_2\mu_2 - c\mu_2 + \mu_2 - c\lambda_2 + c + \lambda_2 - c\lambda_2\mu_2 - \lambda_2 + c\lambda_2}{\lambda_2\mu_2}}{\sqrt{2c + \frac{2(1-c)}{\mu_2^2} - (c + \frac{1-c}{\mu_2})^2}} \\ &= \frac{\frac{0}{\lambda_2\mu_2}}{\sqrt{2c + \frac{2(1-c)}{\mu_2^2} - (c + \frac{1-c}{\mu_2})^2}} = 0.\end{aligned}$$

Since the function is strictly decreasing, the leftmost point $\pi = c$ should be optimal. For the boundary $\pi = c$ the function results in

$$\begin{aligned}\rho_1(c) &= \frac{\frac{1 - c(1-\mu_2) - c\mu_2 + c\lambda_2\mu_2}{\lambda_2\mu_2} - \left(c + \frac{1-c}{\mu_2}\right)\left(c + \frac{1-c}{\lambda_2}\right)}{\sqrt{\left(2c + \frac{2(1-c)}{\mu_2^2} - (c + \frac{1-c}{\mu_2})^2\right)\left(2c + \frac{2(1-c)}{\lambda_2^2} - (c + \frac{1-c}{\lambda_2})^2\right)}} \\ &= \frac{-\frac{c(-1 - \lambda_2\mu_2 + c\lambda_2\mu_2 - c\mu_2 + \mu_2 - c\lambda_2 + c + \lambda_2)}{\lambda_2\mu_2}}{\sqrt{\left(2c + \frac{2(1-c)}{\mu_2^2} - (c + \frac{1-c}{\mu_2})^2\right)\left(2c + \frac{2(1-c)}{\lambda_2^2} - (c + \frac{1-c}{\lambda_2})^2\right)}},\end{aligned} \tag{C.15}$$

where the denominator is always positive. Thus we only have to consider the numerator

$$\begin{aligned}& -\frac{c(-1 - \lambda_2\mu_2 + c\lambda_2\mu_2 - c\mu_2 + \mu_2 - c\lambda_2 + c + \lambda_2)}{\lambda_2\mu_2} \\ = \quad & -\frac{c(-1 - \lambda_2\mu_2 + \lambda_2 + \mu_2 + c(1 + \lambda_2\mu_2 - \lambda_2 - \mu_2))}{\lambda_2\mu_2}.\end{aligned}$$

We assumed $\lambda_2, \mu_2 > 1$ which implies that the first part in the brackets of the numerator $(-1 - \lambda_2\mu_2 + \lambda_2 + \mu_2) < 0$ since $-1 - \lambda_2\mu_2 < -\lambda_2 - \mu_2$. As next we denote the part

in the brackets $(1+\lambda_2\mu_2-\lambda_2-\mu_2)$ as x. Here $x>0$ since $1+\lambda_2\mu_2>\lambda_2+\mu_2$ which holds by assumption $\lambda_2,\mu_2>1$. Now we obtain the term $(-x+cx)$ as the part in the brackets of the numerator, which is negative for $c\in[0,\ldots,1]$. The product in the numerator consists of two negative values such that the result is positive which implies that $\rho_1(c)>0$ and in particular $\rho_1(c)>\rho_1(1)$.

In the second case we consider the function $\rho_2(\pi)$ given in Eq. C.13 where $\pi\in[0,\ldots,c]$. The first derivative with respect to π is positive. Using maple we obtained the local extrema

$$\pi=-\frac{1}{\lambda_2(\lambda_2-1)},$$

which is outside the permissible range for $0\leq\pi\leq 1$. We again consider the function behavior on the boundaries of the interval $[0,\ldots,c]$. For $\pi=0$ we obtain

$$\begin{aligned}\rho_2(0) &= \frac{\frac{1-c(1-\mu_2)}{\lambda_2\mu_2}-\frac{c+\frac{1-c}{\mu_2}}{\lambda_2}}{\sqrt{\frac{2c+\frac{2(1-c)}{\mu_2^2}-(c+\frac{1-c}{\mu_2})^2}{\lambda_2^2}}}\\ &= \frac{\frac{1-c+c\mu_2-c\mu_2-1+c}{\lambda_2\mu_2}}{\sqrt{\frac{2c+\frac{2(1-c)}{\mu_2^2}-(c+\frac{1-c}{\mu_2})^2}{\lambda_2^2}}}\\ &= \frac{0}{\sqrt{\frac{2c+\frac{2(1-c)}{\mu_2^2}-(c+\frac{1-c}{\mu_2})^2}{\lambda_2^2}}}=0.\end{aligned}$$

For the rightmost point $\pi=c$ we obtain

$$\begin{aligned}\rho_2(c) &= \frac{\frac{1-c(1-\mu_2)-c\mu_2+c\lambda_2\mu_2}{\lambda_2\mu_2}-\left(c+\frac{1-c}{\mu_2}\right)\left(c+\frac{1-c}{\lambda_2}\right)}{\sqrt{\left(2c+\frac{2(1-c)}{\mu_2^2}-(c+\frac{1-c}{\mu_2})^2\right)\left(2c+\frac{2(1-c)}{\lambda_2^2}-(c+\frac{1-c}{\lambda_2})^2\right)}}\\ &= \frac{-\frac{c(-1-\lambda_2\mu_2+c\lambda_2\mu_2-c\mu_2+\mu_2-c\lambda_2+c+\lambda_2)}{\lambda_2\mu_2}}{\sqrt{\left(2c+\frac{2(1-c)}{\mu_2^2}-(c+\frac{1-c}{\mu_2})^2\right)\left(2c+\frac{2(1-c)}{\lambda_2^2}-(c+\frac{1-c}{\lambda_2})^2\right)}},\end{aligned}$$

which equals the formula given in Eq. C.15. As shown above $\rho_2(c)>0$ and in particular $\rho_2(c)>\rho_2(0)$. Thus the behavior is as follows: as long as $\pi\leq c$ the correlation function is increasing and reaches its optimum at point $\pi=c$, then the correlation function is decreasing for $\pi>c$, and $\pi\leq 1$. □

Appendix D

Examples

D.1. Example 5.7

Uniformized path matrices are

$$\mathbf{P}_{(i_1,i_2,i_3,i_4)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9286 & 0 & 0.0714 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9286 & 0 & 0.0714 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9286 & 0 & 0.0714 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9286 \end{pmatrix},$$

$$\mathbf{P}_{(i_1,i_9,i_7,i_8)} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0.9286 & 0.0714 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.9286 & 0.0357 & 0.0357 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.9286 & 0.0357 & 0.0357 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9286 \end{pmatrix},$$

$$\mathbf{P}_{(i_1,i_2,i_{10},i_8)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9286 & 0 & 0.0714 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.9286 & 0.0714 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.9286 & 0.0357 & 0.0357 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9286 \end{pmatrix},$$

$$\mathbf{P}_{(i_5,i_6,i_7,i_8)} = \begin{pmatrix} 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.9286 & 0.0357 & 0.0357 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.9286 & 0.0357 & 0.0357 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.9286 & 0.0357 & 0.0357 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9286 \end{pmatrix}.$$

D.2. Example 7.1

Uniformized path matrices are

$$\mathbf{P}_{(i_1,i_2,i_3,i_4)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9286 & 0 & 0.0714 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9286 & 0 & 0.0714 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9286 & 0 & 0.0714 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9286 \end{pmatrix},$$

$$\mathbf{P}_{(i_1,i_2,i_5,i_6)} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9286 & 0 & 0.0714 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.9286 & 0.0357 & 0.0357 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0.9286 & 0.0357 & 0.0357 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9286 \end{pmatrix},$$

D.3. Example 7.1

We computed the following PHDs of order 6 according to the vehicle mobility traces of the edges A through G [40]:

$$\pi^{(A)} = (0.01614, 0.49193, 0, 0.24597, 0, 0.24597),$$

$$\mathbf{D}_0^{(A)} = \begin{pmatrix} -0.01968 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.21991 & 0.21991 & 0 & 0 & 0 \\ 0 & 0 & -0.43981 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.43981 & 0.43981 & 0 \\ 0 & 0 & 0 & 0 & -0.87962 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.87962 \end{pmatrix}.$$

$$\pi^{(B)} = (0.47669, 0.14445, 0.18822, 0.09532, 0, 0.09532),$$

$$\mathbf{D}_0^{(B)} = \begin{pmatrix} -0.01786 & 0.01786 & 0 & 0 & 0 & 0 \\ 0 & -0.07680 & 0.07680 & 0 & 0 & 0 \\ 0 & 0 & -0.07680 & 0.03840 & 0 & 0.03840 \\ 0 & 0 & 0 & -0.09377 & 0.09377 & 0 \\ 0 & 0 & 0 & 0 & -0.18755 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.18755 \end{pmatrix}.$$

$$\pi^{(C)} = (0.99999, 0, 0, 3.98773e-06, 0, 0),$$

$$\mathbf{D}_0^{(C)} = \begin{pmatrix} -0.07095 & 0.07095 & 0 & 0 & 0 & 0 \\ 0 & -0.07095 & 0.07095 & 0 & 0 & 0 \\ 0 & 0 & -0.07095 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.18958 & 0.18958 & 0 \\ 0 & 0 & 0 & 0 & -0.18958 & 0.189581 \\ 0 & 0 & 0 & 0 & 0 & -0.18958 \end{pmatrix}.$$

$$\pi^{(D)} = (1, 0, 0, 0, 0, 0),$$

$$\mathbf{D}_0^{(D)} = \begin{pmatrix} -0.42893 & 0.42893 & 0 & 0 & 0 & 0 \\ 0 & -0.42893 & 0.42893 & 0 & 0 & 0 \\ 0 & 0 & -0.42893 & 0.42893 & 0 & 0 \\ 0 & 0 & 0 & -0.42893 & 0.42893 & 0 \\ 0 & 0 & 0 & 0 & -0.42893 & 0.42893 \\ 0 & 0 & 0 & 0 & 0 & -0.42893 \end{pmatrix},$$

$$\pi^{(E)} = (0.09701, 0.90298, 0, 0, 0, 0),$$

$$\mathbf{D}_0^{(E)} = \begin{pmatrix} -0.01951 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.37078 & 1.37078 & 0 & 0 & 0 \\ 0 & 0 & -1.37078 & 1.37078 & 0 & 0 \\ 0 & 0 & 0 & -1.37078 & 1.37078 & 0 \\ 0 & 0 & 0 & 0 & -1.37078 & 1.37078 \\ 0 & 0 & 0 & 0 & 0 & -1.37078 \end{pmatrix},$$

$$\pi^{(F)} = (0.16103, 0.83897, 0, 0, 0),$$

$$\mathbf{D}_0^{(F)} = \begin{pmatrix} -0.46405 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.33057 & 0.33057 & 0 & 0 & 0 \\ 0 & 0 & -0.33057 & 0.33057 & 0 & 0 \\ 0 & 0 & 0 & -0.33057 & 0.33057 & 0 \\ 0 & 0 & 0 & 0 & -0.33057 & 0.33057 \\ 0 & 0 & 0 & 0 & 0 & -0.33057 \end{pmatrix},$$

$$\pi^{(G)} = (0.00263, 0.99737, 0, 0, 0, 0),$$

$$\mathbf{D}_0^{(G)} = \begin{pmatrix} -0.00768 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.51913 & 0.51913 & 0 & 0 & 0 \\ 0 & 0 & -0.51913 & 0.51913 & 0 & 0 \\ 0 & 0 & 0 & -0.51913 & 0.51913 & 0 \\ 0 & 0 & 0 & 0 & -0.51913 & 0.51913 \\ 0 & 0 & 0 & 0 & 0 & -0.51913 \end{pmatrix}.$$

The coefficient of correlation between edge A and B equals $\hat{\rho}_{AB} = 0.264$. The fitted PHD transfer matrix equals

$$\mathbf{H}_{AB}^{\rho=0.19576} = \begin{pmatrix} 0.01968 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.41176 & 0.02805 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.40437 & 0.47526 & 0 & 0 & 0 \\ 0 & 0 & 0.19786 & 0.34088 & 0 & 0.34088 \end{pmatrix}.$$

D.4. Example 7.4

We used the following transfer matrices:

- $\mathbf{H}_{(a1,u1)}^{\rho=0.26}$ with correlation coefficient $\rho = 0.26$

$$\mathbf{H}_{(a1,u1)}^{\rho=0.26} = \begin{pmatrix} 0.000217810219561 & 0.000973611453869 & 0.000112273502867 & 0 \\ 0 & 0 & 0.018090925759206 & 0 \\ 0 & 0 & 0.046714481120500 & 0.156683568269293 \\ 0 & 0 & 0 & 1.526626942773592 \end{pmatrix},$$

- $\mathbf{H}^{\rho=0.26}_{(a2,u2)}$ with correlation coefficient $\rho = 0.26$

$$\mathbf{H}^{\rho=0.26}_{(a2,u2)} = \begin{pmatrix} 0.001745101910775 & 0.004259515065754 & 0 & 0 \\ 0 & 0.001902532607753 & 0.010233796461206 & 0.031774814060866 \\ 0 & 0 & 0 & 0.645219046820841 \\ 0 & 0 & 0 & 7.913323811792491 \end{pmatrix},$$

- $\mathbf{H}^{\rho=0.2}_{(a3,u3)}$ with correlation coefficient $\rho = 0.2$

$$\mathbf{H}^{\rho=0.2}_{(a3,u3)} = \begin{pmatrix} 0.001633334243295 & 0.007029882074461 & 0 & 0 \\ 0 & 0.016766724831434 & 0.060611402238600 & 0 \\ 0 & 0 & 0.338607657522560 & 0.540483989071130 \\ 0 & 0 & 0 & 4.141319680378356 \end{pmatrix}.$$

Cumulative reward vectors for all policies defined by (7.13)- (7.16) can be given as

$$\mathbf{rc}^{R} = (0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0),$$

$$\mathbf{rc}^{E_1} = (0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0.4764,4.2558,48.35,227.7726,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,0),$$

$$\mathbf{rc}^{E_2} = (0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0.4765,4.2558,48.35,227.7726,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0),$$

$$\mathbf{rc}^{E_3} = (0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0.4765,4.2558,48.35,227.7726,0,0,0,0,0,0,0,0,0).$$

Bibliography

[1] Tapascologne project. http://sourceforge.net/projects/sumo/.

[2] Vehicular mobility trace of the city of Cologne, Germany. http://kolntrace.project.citi-lab.fr/.

[3] PHDs for failure traces from the FTA. http://www-ls4.cs.tu-dortmund.de/failure, Feb. 2014.

[4] S. Ahn and V. Ramaswami. Bilateral Phase Type Distributions. *Stochastic Models*, 21(2-3):239–259, 2005.

[5] A. H. Al-Mohy and N. J. Higham. Computing the fréchet derivative of the matrix exponential, with an application to condition number estimation. *SIAM Journal on Matrix Analysis and Applications*, 30(4):1639–1657, 2009.

[6] A. H. Al-Mohy and N. J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and Applications*, 31(3):970–989, 2009.

[7] G. Andreatta and L. Romeo. Stochastic Shortest Paths with Recourse. *Networks*, 18(3):193–204, 1988.

[8] S. Asmussen and M. Bladt. Point processes with finite-dimensional conditional probabilities. *Stochastic Processes and their Applications*, 82:127–142, 1999.

[9] S. Asmussen and G. Koole. Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability*, 30:365–372, 1993.

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

[11] K. A. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons, 2nd edition, 1989.

[12] A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton. Model-checking continous-time Markov chains. *ACM Trans. Comput. Log.*, 1(1):162–170, 2000.

[13] A. L. Badescu, E. K. Cheung, and D. Landriault. Dependent Risk Models with Bivariate Phase-Type Distributions. *Journal of Applied Probability*, 46(1):113–131, 2009.

[14] A. L. Badescu, S. Drekic, and D. Landriault. Analysis of a Threshold Divided Strategy for a MAP Risk Model. *Scandinavian Acturial Journal*, 4:227–247, 2007.

[15] A. L. Badescu and D. Landriault. Applications of Fluid Flow Matrix Analytic Methods in Ruin Theory - A Review. *RACSAM - Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas*, 103(2):353–372, 2009.

[16] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Automated performance and dependability evaluation using model checking. In M. Calzarossa and S. Tucci, editors, *Performance*, volume 2459 of *Lecture Notes in Computer Science*, pages 261–289. Springer, 2002.

[17] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.

[18] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Reachability in continuous-time Markov reward decision processes. In J. Flum, E. Graedel, and T. Wilke, editors, *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*, pages 53–71, Amsterdam, February 2008. Amsterdam University Press.

[19] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen. Performance evaluation and model checking join forces. *Commun. ACM*, 53(9):76–85, 2010.

[20] C. Baier, H. Hermanns, J.-P. Katoen, and B. R. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theoretical Computer Science*, 345(1):2 – 26, 2005.

[21] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[22] N. Bailey. *The Mathematical Theory of Infectious Diseases and its Applications*. Griffin, 1975.

[23] A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to Act Using Real-time Dynamic Programming. *Artif. Intell.*, 72(1-2):81–138, Jan. 1995.

[24] N. Bäuerle and U. Rieder. *Markov Decision Processes with Applications to Finance: Markov Decision Processes with Applications to Finance*. Universitext. Springer, 2011.

[25] F. Bause, P. Buchholz, and J. Kriege. ProFiDo - The Processes Fitting Toolkit Dortmund. In *Proc. of the 7th International Conference on Quantitative Evaluation of SysTems (QEST 2010)*, pages 87–96. IEEE Computer Society, 2010.

[26] D. P. Berovic and R. B. Vinter. The Application of Dynamic Programming to Optimal Inventory Control. *IEEE Trans. Automat. Contr.*, 49(5):676–685, 2004.

[27] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 2005.

[28] D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume II. Athena Scientific, 2007.

[29] D. P. Bertsekas and J. N. Tsitsiklis. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16:580–595, 1991.

[30] A. Bobbio and A. Cumani. ML estimation of the parameters of a PH distribution in triangular canonical form. In G. Balbo and G. Serazzi, editors, *Computer Performance Evaluation*, pages 33–46. Elsevier, 1992.

[31] A. Bobbio, A. Horváth, and M. Telek. Matching Three Moments with Minimal Acyclic Phase Type Distributions. *Stochastic Models*, 21(2-3):303–326, 2005.

[32] L. Bodrog, A. Heindl, G. Horváth, M. Telek, and A. Horváth. A Markovian canonical form of second-order matrix-exponential processes. *European Journal of Operational Research*, 160(1):51–68, 2008.

[33] B. Bonet and H. Geffner. Solving Stochastic Shortest-Path Problems with RTDP. Technical report, Universidad Simon Bolivar, 2002. See http://ldc.usb.ve/ bonet/reports/rtdp.pdf.

[34] J. Boyan and M. Mitzenmacher. Improved Results for Route Planning in Stochastic Transportation Networks. In *In Proc. of Symposium of Discrete Algorithms*, 2001.

[35] J. A. Boyan and M. L. Littman. Exact Solutions to Time-Dependent MDPs. In *in Advances in Neural Information Processing Systems*, pages 1026–1032. MIT Press, 2000.

[36] A. J. Briggs, C. Detweiler, D. Scharstein, and A. Vandenberg-Rodes. Expected Shortest Paths for Landmark-based Robot Navigation. *International Journal of Robotics Research*, 23:717–728, 2004.

[37] J. L. Bruno, P. J. Downey, and G. N. Frederickson. Sequencing Tasks with Exponential Service Times to Minimize the Expected Flow Time or Makespan. *J. ACM*, 28(1):100–113, 1981.

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

[39] P. Buchholz. Finite Horizon Analysis of Infinite CTMDPs. *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 0:1–12, 2012.

[40] P. Buchholz and I. Felko. Ph-graphs for analyzing shortest path problems with correlated traveling times. *Computers & Operations Research*, 59:51–69, july 2015.

[41] P. Buchholz, I. Felko, and J. Kriege. Transformation of Acyclic Phase Type Distributions for Correlation Fitting. In *ASMTA*, pages 96–111, 2013.

[42] P. Buchholz, E. M. Hahn, H. Hermanns, and L. Zhang. Model checking algorithms for CTMDPs. In *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, pages 225–242, 2011.

[43] P. Buchholz, P. Kemper, and J. Kriege. Multi-class Markovian Arrival Arocesses and their Parameter Fitting. *Perform. Eval.*, 67(11):1092–1106, 2010.

[44] P. Buchholz and J. Kriege. A Heuristic Approach for Fitting MAPs to Moments and Joint Moments. In *QEST*, pages 53–62, 2009.

[45] P. Buchholz and J. Kriege. Equivalence Transformations for Acyclic Phase Type Distributions. Technical Report 827, Dep. of Computer Science, TU Dortmund, 2009. `http://www.cs.uni-dortmund.de/nps/de/Forschung/Publikationen/Graue_Reihe1/Ver__ffentlichungen_2009/827.pdf`.

[46] P. Buchholz and J. Kriege. Markov Modeling of Availability and Unavailability Data. In *Proc. of the Tenth European Dependable Computing Conference (EDCC)*, pages 94–105. IEEE Computer Society, 2014.

[47] P. Buchholz, J. Kriege, and I. Felko. *Input Modeling with Phase-Type Distributions and Markov Models - Theory and Applications*. SpringerBriefs in Mathematics. Springer, 2014.

[48] P. Buchholz and A. Panchenko. An EM Algorithm for Fitting of Real Traffic Traces to PH-Distribution. In *PARELEC*, pages 283–288. IEEE Computer Society, 2004.

[49] P. Buchholz and I. Schulz. Numerical Analysis of Continuous Time Markov Decision Processes over Finite Horizons. *Computers & Operations Research*, 38(3):651–659, 2011.

[50] P. Buchholz and M. Telek. Rational Arrival Processes Associated to Labelled Markov Processes. *Journal of Applied Probability*, 49(1):40–59, 2012.

[51] P. Buchholz and M. Telek. On Minimal Representations of Rational Arrival Processes. *Annals of Operations Research*, 202(1):35–58, 2013.

[52] G. Casale, E. Z. Zhang, and E. Smirni. KPC-Toolbox: Simple Yet Effective Trace Fitting Using Markovian Arrival Processes. In *QEST*, pages 83–92. IEEE Computer Society, 2008.

[53] G. Casale, E. Z. Zhang, and E. Smirni. Trace Data Characterization and Fitting for Markov Modeling. *Perform. Eval.*, 67(2):61–79, 2010.

[54] E. K. Cheung and D. Landriault. A generalized penalty function with the maximum surplus prior to ruin in a MAP risk model. *Insurance: Mathematics and Economics*, 46:127–2134, 2010.

[55] E. Cinlar. *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, N. J., 1975.

[56] A. Cumani. On the Canonical Representation of Homogeneous Markov Processes Modeling Failure-Time distributions. *Micorelectronics and Reliability*, 22(3):583–602, 1982.

[57] G. Dantzig. *Linear Programming and Extensions*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1963.

[58] T. Dayar. *Analyzing Markov Chains using Kronecker Products*. Briefs in Mathematics. Springer, 2012.

[59] L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In J. Baeten and S. Mauw, editors, *CONCUR99 Concurrency Theory*, volume 1664 of *Lecture Notes in Computer Science*, pages 66–81. Springer Berlin Heidelberg, 1999.

[60] F. De Vico Fallani, J. Richiardi, M. Chavez, and S. Achard. Graph Analysis of Functional Brain Networks: Practical Issues in Translational Neuroscience. *Philosophical Transactions B: Biological Sciences*, 369(1653), Oct 2014.

[61] M. Dehon and G.Latouche. A Geometric Interpretation of the Relations between Exponential and Generalized Erlang Distributions. *Advanced Applied Probability*, 14:885–897, 1982.

[62] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[63] L. Dieci and A. Papini. Padé approximation for the exponential of a block triangular matrix. *Linear Algebra and its Applications*, 308(1):183 – 202, 2000.

[64] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[65] R. T. Dunn and K. D. Glazebrook. Discounted Multiarmed Bandit Problems on a Collection of Machines with Varying Speeds. *Math. Oper. Res.*, 29(2):266–279, 2004.

[66] E. Echavarria, B. Hahn, G. van Bussel, and T. Tomiyama. Reliability of wind turbine technology through time. *Journal of Solar Energy Engineering*, 130(3):031005, 2008.

[67] D. Eisenstat. Random road networks: the quadtree model. *CoRR*, abs/1008.4916, 2010.

[68] A. Ephremides and S. Verdú. Control and Optimization Methods in Communication Network Problems. *IEEE Transactions on Automatic Control*, 34(9):930–942, 1989.

[69] A. K. Erlang. Solution of some Problems in the Theory of Probabilities of Significance in Automatic Telephone Exchanges. *Elektrotkeknikeren*, 13, 1917.

[70] M. Faddy. Phase-type distributions for failure times. *Mathematical and Computer Modelling*, 22(10):63 – 70, 1995.

[71] Y. Fan, R. Kalaba, and J. M. II. Shortest paths in stochastic networks with correlated link costs. *Computers & Mathematics with Applications*, 49(9):1549 – 1564, 2005.

[72] Y. Fan, R. Kalaba, and I. Moore, J.E. Arriving on time. *Journal of Optimization Theory and Applications*, 127(3):497–513, 2005.

[73] Y. Fang. Hyper-Erlang distribution model and its application in wireless mobile networks. *Wirel. Netw.*, 7(3):211–219, May 2001.

[74] J. Fearnley. Exponential Lower Bounds For Policy Iteration. *CoRR*, abs/1003.3418, 2010.

[75] W. Fischer and K. S. Meier-Hellstern. The Markov-Modulated Poisson Process (MMPP) Cookbook. *Perform. Eval.*, 18(2):149–171, 1993.

[76] B. L. Fox and P. W. Glynn. Computing Poisson Probabilities. *Commun. ACM*, 31(4):440–445, 1988.

[77] H. Frank. Shortest paths in probabilistic graphs. *Operations Research*, 17(4):583–599, 1969.

[78] S. G. and C. Ismail. Optimal Routing Policy Problems in Stochastic Time-dependent Networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.

[79] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[80] W. K. Grassmann. Transient Solutions in Markovian Queueing Systems. *Computers & OR*, 4(1):47–53, 1977.

[81] X. Guo and O. Hernandez-Lerma. *Continuous-Time Markov Decision Processes: Theory and Applications*. Stochastic Modelling and Applied Probability. Springer, 2009.

[82] R. W. Hall. The Fastest Path through a Network with Random Time-Dependent Travel Times. *Transportation Science*, 20(3):182–188, 1986.

[83] T. D. Hansen and U. Zwick. Lower Bounds for Howard's Algorithm for Finding Minimum Mean-Cost Cycles. In *Algorithms and Computation - 21st International Symposium, ISAAC 2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I*, pages 415–426, 2010.

[84] R. Hassin and E. Zemel. On shortest paths in graphs with random weights. *Mathematics of Operations Research*, 10(4):557–564, 1985.

[85] E. Hau. *Windkraftanlagen: Grundlagen, Technik, Einsatz, Wirtschaftlichkeit*. Springer Berlin Heidelberg, 2014.

[86] Q.-M. He. The Versatility of MMAP[K] and the MMAP[K]/G[K]/1 Queue. *Queueing Syst.*, 38(4):397–418, 2001.

[87] Q.-M. He, H. Li, and Y. Q. Zhao. Ergodicity of the BMAP/PH/s/s+K retrial queue with PH-retrial times. *Queueing Syst.*, 35(1-4):323–347, 2000.

[88] Q.-M. He and M. Neuts. Markov Arrival Processes with Marked Transitions. *Stochastic Processes and their Applications*, 74:37–52, 1998.

[89] Q.-M. He and H. Zhang. PH-invariant Polytopes and Coxian Representations of Phase Type Distributions. *Stochastic Models*, 22(3):383–409, 2006.

[90] A. Heindl. Inverse Characterization of hyperexponential MAP(2)s. In *Proc. ASMTA*, pages 183–189, 2004.

[91] R. Hollanders. On the policy iteration algorithm for pagerank optimization. Master's thesis, Université Catholique de Louvain, École Polytechnique de Louvane, Pôle d'Ingénierie Mathématique (INMA) and Massachusett's Institute of Technology, Laboratory for Information and Decision Systems, 2010.

[92] R. Hornig and A. Varga. An Overview of the OMNeT++ Simulation Environment. In *Proc. of 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools)*, 2008.

[93] A. Horváth and M. Telek. PhFit: A general purpose phase type fitting tool. In *Performance Tools 2002*, volume 2324 of *LNCS*, pages 82–91. Springer, 2002.

[94] A. Horváth and M. Telek. Matching more than Three Moments with Acyclic Phase Type Distributions. *Stochastic Models*, 23:167–194, 2007.

[95] A. Horváth and M. Telek. On the Properties of Acyclic Bilateral Phase Type Distributions. In *VALUETOOLS*, page 79, 2007.

[96] G. Horváth, M. Telek, and P. Buchholz. A MAP Fitting Approach with Independent Approximation of the Inter-Arrival Time Distribution and the Lag-Correlation. In *QEST*, pages 124–133. IEEE CS Press, 2005.

[97] R. Howard. *Dynamic Programming and Markov Processes*. Published jointly by the Technology Press of the Massachusetts Institute of Technology, 1960.

[98] Q. Hu and W. Yue. *Markov Decision Processes with their Applications*. Advances in Mechanics and Mathematics. Springer, 2007.

[99] H. Huang and S. Gao. Optimal paths in dynamic networks with dependent random link travel times. *Transportation Research Part B: Methodological*, 46(5):579–598, 2012.

[100] B. Javadi, D. Kondo, A. Iosup, and D. H. J. Epema. The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems. *J. Parallel Distrib. Comput.*, 73(8):1208–1223, 2013.

[101] A. Jensen. Markoff Chains as an Aid in the Study of Markoff Processes. *Skand. Aktuarietiedskr.*, 36:87–91, 1953.

[102] V. Kalapala, V. Sanwalani, A. Clauset, and C. Moore. Scale invariance in road networks. *Phys. Rev. E*, 73:026130, Feb 2006.

[103] N. Karmarkar. A New Polynomial-time Algorithm for Linear Programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, pages 302–311. ACM, 1984.

[104] H. Keller. *Numerical Methods for Two-point Boundary-value Problems*. Blaisdell book in numerical analysis and computer science. Blaisdell, 1968.

[105] J. G. Kemeny and J. L. Snell. *Finite Markov chains*. University series in undergraduate mathematics. VanNostrand, New York, repr edition, 1969.

[106] C. S. Kenney and A. J. Laub. A schur–fréchet algorithm for computing the logarithm and exponential of a matrix. *SIAM J. Matrix Anal. Appl.*, 19(3):640–663, jul 1998.

[107] L. G. Khachiyan. A Polynomial Algorithm in Linear Programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979.

[108] R. E. A. Khayari, R. Sadre, and B. Haverkort. Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation*, 52:175–191, 2003.

[109] A. Klemm, C. Lindemann, and M. Lohmann. Modeling IP Traffic Using the Batch Markovian Arrival Process. *Perform. Eval.*, 54(2):149–173, 2003.

[110] D. Kondo, B. Javadi, A. Iosup, and D. H. J. Epema. The failure trace archive: Enabling comparative analysis of failures in diverse distributed systems. In *CCGRID*, pages 398–407. IEEE, 2010.

[111] G. Koole. *Stochastic Scheduling and Dynamic Programming*. PhD thesis, CWI Tract 113, CWI Amsterdam, 1995.

[112] J. Kriege. *Fitting Simulation Input Models for Correlated Traffic Data*. PhD thesis, Technische Universität Dortmund, Fakultät für Informatik, 2012.

[113] A. Kuczura. The Interrupted Poisson Process as an Overflow Process. *The Bell System Technical Journal*, 52(3):437–448, 1973.

[114] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability. Society for Industrial and Applied Mathematics, 1987.

[115] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM, 1999.

[116] C. L. Lawson and B. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, 1974.

[117] M. L. Littman, T. L. Dean, and L. P. Kaelbling. On the Complexity of Solving Markov Decision Problems. In *In Proc. of the eleventh international conference on uncertainty in artificial intelligence*, pages 394–402, 1995.

[118] C. F. Loan. The Ubiquitous Kronecker Product. *Journal of Computational and Applied Mathematics*, 123(1-2):85–100, 2000.

[119] R. P. Loui. Optimal paths in graphs with stochastic or multidimensional weights. *Commun. ACM*, 26(9):670–676, sep 1983.

[120] D. M. Lucantoni. The BMAP/G/1 Queue: A Tutorial. In L. Donatiello and R. D. Nelson, editors, *Performance/SIGMETRICS Tutorials*, volume 729 of *Lecture Notes in Computer Science*, pages 330–358. Springer, 1993.

[121] D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts. A Single-Server Queue with Server Vacations and a Class of Non-Renewal Arrival Processes. *Advances in Applied Probability*, 22(3):676–705, 1990.

[122] Y. Mansour and S. P. Singh. On the Complexity of Policy Iteration. In *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Stockholm, Sweden, July 30 - August 1, 1999*, pages 401–408, 1999.

[123] Y. (Marco), Nie and X. Wu. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological*, 43(6):597–613, 2009.

[124] M. Maron. *Numerical Analysis: A Practical Approach*. Macmillan, 1982.

[125] A. Martin-Löfs. Optimal Control of a Continuous-time Markov Chain with Periodic Transition Probabilities. *Operations Research*, 15:872–881, 1967.

[126] A. Mészáros and M. Telek. A Two-Phase MAP Fitting Method with APH Interarrival Time Distribution. In *Winter Simulation Conference*. ACM, 2012.

[127] B. L. Miller. Finite State Continuous Time Markov Decision Processes with a Finite Planning Horizon. *SIAM Journal on Control*, 6(2):266–280, 1968.

[128] C. B. Moler and C. F. V. Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 45(1):3–49, 1978.

[129] D. Montoro-Cazorla and R. Pérez-Ocón. A Maintenance Model with Failures and Inspection Following Markovian Arrival Processes and Two Repair Modes. *European Journal of Operational Research*, 186(2):694–707, 2008.

[130] J. Mosely and P. Humblet. A Class of Efficient Contention Resolution Algorithms for Multiple Access Channels. *Communications, IEEE Transactions on*, 33(2):145–151, Feb 1985.

[131] M. R. Neuhäußer and L. Zhang. Time-Bounded Reachability Probabilities in Continuous-Time Markov Decision Processes. In *QEST 2010, Seventh International Conference on the Quantitative Evaluation of Systems, Williamsburg, Virginia, USA, 15-18 September 2010*, pages 209–218, 2010.

[132] M. F. Neuts. A Versatile Markovian Point Process. *Journal of Applied Probability*, 16:764–779, 1979.

[133] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. Johns Hopkins University Press, 1981.

[134] M. F. Neuts and K. S. Meier. On the use of phase type distributions in reliability modelling of systems with two components. *Operations-Research-Spektrum*, 2(4):227–234, 1981.

[135] E. Nikolova, M. Brand, and D. R. Karger. Optimal Route Planning under Uncertainty. In *ICAPS*, pages 131–141, 2006.

[136] E. Nikolova and D. R. Karger. Route Planning under Uncertainty: The Canadian Traveller Problem. In *AAAI*, pages 969–974, 2008.

[137] C. A. O'Cinneide. On Non-Uniqueness of Representations of Phase-Type Distributions. *Stochastic Models*, 5:247–259, 1989.

[138] C. A. O'Cinneide. Phase-type distributions: open problems and a few properties. *Stochastic Models*, 15(4):731–757, 1999.

[139] H. Okamura, T. Dohi, and K. S. Trivedi. Markovian Arrival Process Parameter Estimation with Group Data. *IEEE/ACM Trans. Netw.*, 17(4):1326–1339, 2009.

[140] H. Okamura, T. Dohi, and K. S. Trivedi. Improvement of Expectation-Maximization Algorithm for Phase-Type Distributions with Grouped and Truncated Data. *Applied Stochastic Models in Business and Industry*, 29(2):141–156, 2012.

[141] A. Panchenko. *Modelling of Network Processes by Means of Markovian Arrival Processes*. PhD thesis, Technische Universität Dresden, Fakultät Informatik, 2007.

[142] C. Papadimitriou and J. N. Tsitsiklis. The Complexity of Markov Decision Processes. *Math. Oper. Res.*, 12(3):441–450, aug 1987.

[143] C. H. Papadimitriou and M. Yannakakis. Shortest Paths Without a Map. *Theor. Comput. Sci.*, 84(1):127–150, 1991.

[144] M. J. Parks. *A Study of Algorithms to Compute the Matrix Exponential*. PhD thesis, Mathematics Department, University of California, 1994.

[145] G. H. Polychronopoulos and J. N. Tsitsiklis. Stochastic Shortest Path Problems with Recourse. *Networks*, 27(2):133–143, 1996.

[146] M. Pourakbar, J. B. G. Frenk, and R. Dekker. End-of-life inventory decisions for consumer electronics service parts. *Production and Operations Management*, 21(5):889–906, 2012.

[147] R. Pulungan. *Reduction of Acyclic Phase-Type Representations*. PhD thesis, Universitaet des Saarlandes, Saarbruecken, Germany, 2009.

[148] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

[149] Q. Qiu, Q. Qu, and M. Pedram. Stochastic Modeling of a Power-managed System-construction and Optimization. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 20(10):1200–1217, 2006.

[150] A. Ralston and P. Rabinowitz. *A First Course in Numerical Analysis*. Dover books on mathematics. Dover Publications, 2001.

[151] A. Reibman, R. Smith, and K. Trivedi. Markov and Markov reward model transient analysis: An overview of numerical approaches. *European Journal of Operational Research*, 40(2):257–267, 1989.

[152] A. Reibman and K. Trivedi. Numerical Transient Analysis of Markov Models. *Computers & Operations Research*, 15(1):19–36, Jan. 1988.

[153] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003. http://www.google.com/url?q=http://www-users.cs.umn.edu/ saad/IterMethBook_2ndEd.pdf.

[154] P. Salvador, A. Pacheco, and R. Valadas. Modeling IP Traffic: Joint Characterization of Packet Arrivals and Packet Sizes Using BMAPs. *Computer Networks*, 44(3):335–352, 2004.

[155] M. Schäl. On Discrete-time Dynamic Programming in Insurance: Exponential Utility and Minimizing the Ruin Probability. *Scandinavian Actuarial Journal*, 3:189–210, 2004.

[156] M. Schäl. Control of Ruin Probabilities by Discrete-time Investments. *Mathematical Methods of Operational Research*, 62:141–158, 2005.

[157] B. Sengupta. Markov processes whose steady state distribution is matrix-exponential with an application to the gi/ph/1 queue. *Advances in Applied Probability*, 21(1), march 1989.

[158] J. Shanthikumar. Bilateral Phase Type Distribution. *Naval Research Logistics Quarterly*, 32(March 1985):119–136, 1985.

[159] C. E. Sigal, A. A. B. Pritsker, and J. J. Solberg. The stochastic shortest route problem. *Operations Research*, 1122-1129(5), 1980.

[160] R. Simmons and S. Koenig. Probabilistic Robot Navigation in Partially Observable Environments. In *In Proceedings of IJCAI-95*, pages 1080–1087. IJCAI, Inc, 1995.

[161] M. Steffensen. Quadratic Optimization of Life and Pension Insurance Payments. *Astin Bulletin*, 36(1):245, 2006.

[162] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[163] W. J. Stewart. *Probability, Markov Chains, Queues, and Simulation*. Princeton University Press, 2009.

[164] D. Sudholt and C. Thyssen. A simple ant colony optimizer for stochastic shortest path problems. *Algorithmica*, 64(4):643–672, 2012.

[165] M. Telek and G. Horváth. A Minimal Representation of Markov Arrival Processes and a Moments Matching Method. *Perform. Eval.*, 64(9-12):1153–1168, 2007.

[166] M. Telek et al. BuTools - Program Packages. webspn.hit.bme.hu/˜butools/.

[167] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *Journal of Robotics Research*, 19(11), 2000.

[168] A. Thümmler, P. Buchholz, and M. Telek. A Novel Approach for Phase-Type Fitting with the EM Algorithm. *IEEE Trans. Dep. Sec. Comput.*, 3(3):245–258, 2006.

[169] K. S. Trivedi, J. K. Muppala, S. P. Woolet, and B. R. Haverkort. Composite Performance and Dependability Analysis. *Performance Evaluation*, 14(2-3):197–215, 1992.

[170] P. Tseng. Solving H-horizon, Stationary Markov Decision Problems in Time Proportional to Log(H). *Oper. Res. Lett.*, 9(5):287–297, sep 1990.

[171] S. Uppoor, Ó. Trullols-Cruces, M. Fiore, and J. M. Barceló-Ordinas. Generation and Analysis of a Large-Scale Urban Vehicular Mobility Dataset. *IEEE Trans. Mob. Comput.*, 13(5):1061–1075, 2014.

[172] S. T. Waller and A. K. Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227, 2002.

[173] H. Wang, K. Ma, and F. Blaabjerg. Design for reliability of power electronic systems. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pages 33–44. IEEE, 2012.

[174] M. P. Wellman, M. Ford, and K. Larson. Path Planning under Time-Dependent Uncertainty. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 532–539, 1995.

[175] C. Wenner. Hardness Results for the Shortest Path Problem under Partial Observability, 2009. See http://cwenner.net/papers/sctp.pdf.

[176] K. Wickwire. Mathematical Models for the Control of Pests and Infectious Diseases: A Survey. *Theoretical Population Biology*, 11(2):182–238, 1977.

[177] C. F. J. Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[178] Y. Ye. The Simplex and Policy-Iteration Methods are Strongly Polynomial for the Markov Decision Problem with a Fixed Discount Rate, 2010.

List of Figures

List of Tables

List of Algorithms