A multi-objective approach for PH-graphs with applications to stochastic shortest paths

Peter Buchholz[1] · Iryna Dohndorf[1]

Abstract

Stochastic shortest path problems (SSPPs) have many applications in practice and are subject of ongoing research for many years. This paper considers a variant of SSPPs where times or costs to pass an edge in a graph are, possibly correlated, random variables. There are two general goals one can aim for, the minimization of the expected costs to reach the destination or the maximization of the probability to reach the destination within a given budget. Often one is interested in policies that build a compromise between different goals which results in multi-objective problems. In this paper, an algorithm to compute the convex hull of Pareto optimal policies that consider expected costs and probabilities of falling below given budgets is developed. The approach uses the recently published class of PH-graphs that allow one to map SSPPs, even with generally distributed and correlated costs associated to edges, on Markov decision processes (MDPs) and apply the available techniques for MDPs to compute optimal policies.

1 Introduction

Shortest path problems are classical decision problems in computer science and operations research with many practical applications. The goal is to find an in some sense optimal path between a source and a destination node in a weighted directed graph. Weights of edges describe time, costs, gain, or reliability when passing the edge. We use in the sequel the term costs of an edge to describe the quantitative parameter. The

✉ Peter Buchholz
    peter.buchholz@cs.tu-dortmund.de

    Iryna Dohndorf
    iryna.dohndorf@cs.tu-dortmund.de

[1] Informatik IV, TU Dortmund, 44221 Dortmund, Germany

entire version of the problem assumes deterministic costs. However, often costs are not exactly known such that stochastic descriptions of the costs are more appropriate which results in the Stochastic Shortest Path Problem (SSPP). We consider in this paper cases where a decision maker can decide to choose an outgoing edge when reaching a node. This implies that decisions are made adaptively and not a priori as in Nie and Wu (2009). In many practical applications, random variables modeling the edge costs are not independent. For example adjacent roads have similar traffic conditions, interest rates in subsequent periods are correlated and failure rates of related components are dependent. This implies that the introduction of correlation in SSPPs is important for realistic models but makes the problem of specifying parameters and computing optimal paths even harder.

In a stochastic setting with adaptive decisions, a unique optimal path usually does not exist, instead the chosen path depends on the realization of costs on already passed edges. This includes the situation that even with only positive weights, the optimal path may contain cycles. Instead of a path, a policy is defined that specifies for each node the choice of the outgoing edge which may depend on additional information that is available when reaching the node. The term *optimal* is not uniquely defined for stochastic decision problems. One possible interpretation is the minimization (or maximization) of the expected costs to reach the destination from the source. In other settings, it is more appropriate to maximize (or minimize) the probability to reach the destination within a given budget. In traffic networks, the first goal corresponds to the minimization of the expected travel time and the second goal corresponds to the maximization of the probability to meet a deadline. Often one is interested in a compromise solution, i.e., a policy that reaches the destination in a short expected time with a small probability of missing the deadline. This results in a multi-objective optimization problem, with incomparable solutions. Therefore it is important to characterize the set of Pareto optimal solutions or at least a convex hull of this set.

This paper considers SSPPs with correlated edge costs and introduces methods to compute optimal policies for instances of the two mentioned problems. Furthermore, an approach is introduced to approximate the convex hull of Pareto optimal policies, if several goal functions are combined. The problem will be solved in the context of PH-graphs (PHGs) (Buchholz and Felko 2015), a recently published class of stochastic graphs that allows one to include generally distributed edge costs and correlation between adjacent edges. Edge costs are modeled by phase-type distributions (PHDs) (Buchholz et al. 2014; Neuts 1981). PHGs can be mapped on Markov decision processes (MDPs) such that algorithms from MDPs can be adopted to compute optimal policies and a convex hull of Pareto optimal policies.

Related work SSPPs are considered in many different applications ranging from transportation (Barbarosoğlu and Arda 2004; Nikolova and Karger 2008) to computer networks (Nain and Towsley 2016), data migration (Li et al. 2016), social networks (Rezvanian and Reza Meybodi 2016) or finance (Budd 2016; Koenig and Meissner 2015). An enormous number of papers has been published in the area such that we can only highlight a few, most relevant results for our work. Algorithms to compute policies that maximize the probability to reach the destination within a given budget can be found in Fan et al. (2005b) and Samaranayake et al. (2012). Correlation among edge costs is considered in several papers (Fan et al. 2005a; Nain and Towsley 2016;

Samaranayake et al. 2012). The major problem in this context is the compact definition of dependencies and its consideration in optimization algorithms. Often dependencies are defined for the costs of adjacent edges.

The combination of different goal functions results in multi-objective SSPPs (Ji et al. 2011; Zhaowang et al. 2004; Randour et al. 2015; Zhang et al. 2010). Often multi-dimensional costs are considered in this case. Alternatively, the combination of expectation and variance is minimized (Zhaowang et al. 2004; Zhang et al. 2010) or the probability of exceeding the budget is defined as a constraint (Ji et al. 2011). Often heuristic algorithms are applied to approximate the set of Pareto optimal solutions (Ji et al. 2011; Zhaowang et al. 2004; Zhang et al. 2010). Only in some cases correlations are considered, more often independent distributions are assumed (Nikolova et al. 2006).

We analyze the problems in the context of PHGs which have been defined in Buchholz and Felko (2015) and Dohndorf (2017). Distributions in this model class are described by PHDs (Neuts 1981) and correlations between edge costs are introduced using concepts from Markovian arrival processes (MAPs) (Neuts 1979) which have been adopted to adjacent distributions in a graph (Buchholz and Felko 2015; Dohndorf 2017). PHDs and MAPs are commonly used in performance and reliability analysis and methods to determine their parameters from measurements are available (Buchholz et al. 2014). The major advantage of PHGs is that they can be mapped on MDPs in continuous time which implies that the large set of available optimization techniques for MPDs can be adopted for PHGs.

MDPs are widely studied and shortest path problems, where the successor edge is defined by a probability distribution rather than a single edge, can be naturally defined as MDPs (Bertsekas 2007; Puterman 2005). The analysis of continuous time MDPs and the relation between discrete and continuous time MDPs is investigated in Miller (1968), Buchholz et al. (2011), Puterman (2005) and Serfozo (1979). Several approaches for multi-objective optimization exist in the context of MDPs (Chatterjee et al. 2006; Ghosh 1990; Forejt et al. 2011; Wakuta and Togawa 1998; Bouyer et al. 2018). Algorithms to compute or approximate the convex hull of multi-objective MDPs can be found in Roijers et al. (2014b), Roijers et al. (2015) and Roijers et al. (2014a).

Contribution of the paper In this paper we consider in SSPPs with correlated edge costs the combined optimization of different goal functions that take into account the expected costs and the probability to reach the destination with a given budget. It is shown that the weighted sum of these values can be optimized using standard MDP algorithms. Furthermore, the convex hull of Pareto optimal policies according to the goal functions is computed with an algorithm which is an extension of the approach proposed in Roijers et al. (2014b), Roijers et al. (2015) and Roijers et al. (2014a).

Organization of the paper In following section, we introduce the basic definitions and describe solution algorithms. Then, in Sect. 3, it is shown that the weighted sum of the several goal functions can be described by an MDP with a single goal function and a value iteration based approach for optimization of the multi-objective goal function relative to a given weight vector is presented. Afterwards, in Sect. 4, the multi-objective solution algorithm to compute the convex hull of Pareto optimal solutions is described. We present examples from scheduling and vehicular traffic in

Sect. 5 to show the applicability of the proposed solution method and conclude the paper in Sect. 6.

2 PH-graphs

We present PHGs formally and by a simple example. Further details about the model and algorithms to fit the parameters of the included PHDs can be found in Buchholz and Felko (2015) and Dohndorf (2017). After introduction of the basic model, the shortest path problem and the basics for solving the problem are introduced.

A PHG is a directed stochastic graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{P})$ where $\mathcal{V}$ is a finite set of nodes, $\mathcal{E}$ is a finite set of edges, and set $\mathcal{P}$ contains Phase-type distributions (PHDs) characterizing the random variables for edge costs. Set $\mathcal{V}$ contains an initial node $v_{ini} \in \mathcal{V}$ and a destination node $v_{fin} \in \mathcal{V}$, as shown in Fig. 1. We consider paths between v_{ini} and v_{fin} and assume in the sequel that v_{fin} has no outgoing edges. Passing an edge of a PHG induces costs that are described by non-negative random variables. These costs are redrawn from the distribution of the associated random variables if the edge is passed more than once. Costs of adjacent edges can be correlated.

The costs of edge $i \in \mathcal{E}$ are described by a PHD with representation $(\boldsymbol{\pi}_i, \mathbf{D}_i)$ of order n_i (see Fig. 1b). Matrix $\mathbf{D}_i$ is a $n_i \times n_i$ sub-generator (i.e., $\mathbf{D}_i \mathbf{I} \leq \mathbf{0}$, $\mathbf{D}_i(x, y) \geq 0$ for $x \neq y$) which describes an absorbing Markov chain in continu-

(a) Simple PH-graph.

(b) Detailed path $i_1 \rightarrow i_4$.

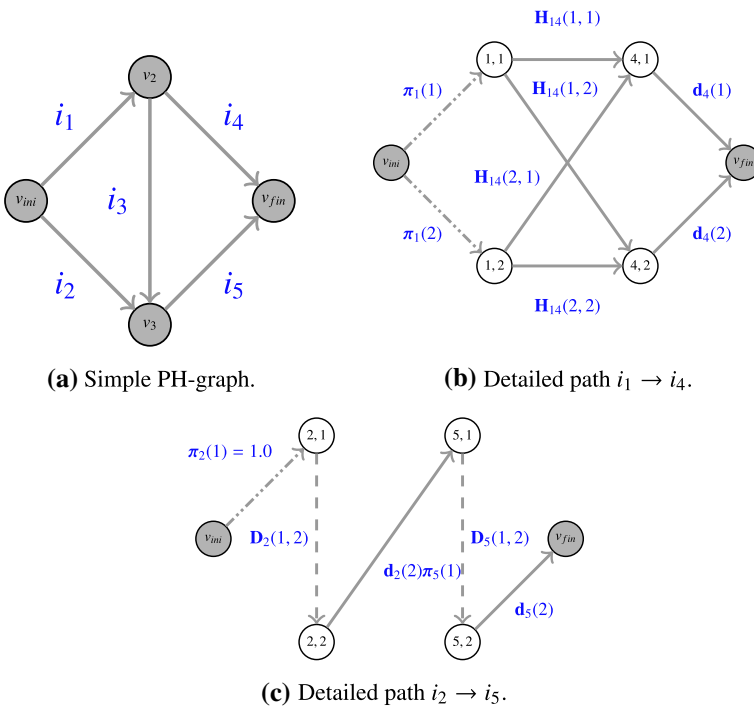**(c)** Detailed path $i_2 \rightarrow i_5$.

Fig. 1 Simple examples of PH-graph with 3 possible paths between v_{ini} and v_{fin}

ous time where all non-absorbing states are transient states (Buchholz et al. 2014). The closing vector $\mathbf{d}_i = (-\mathbf{D}_i)\,\mathbf{I}$ where $\mathbf{I}$ is the a column vector of 1s of length n_i. Vector $\boldsymbol{\pi}_i \in \mathbb{R}_{\geq 0}^{1 \times n_1}$ is the initial distribution. The costs of edge i corresponds to the time to absorption of the Markov chain described by the PHD. Thus, $E(X_i^j) = j!\boldsymbol{\pi}_i \mathbf{M}_i^j \mathbf{I}$ with $\mathbf{M}_i = (-\mathbf{D}_i)^{-1}$ is the jth moment of the costs, $\sigma_i^2 = E(X_i^2) - (E(X_i^1))^2$ is the variance and $F_i(t) = 1 - \boldsymbol{\pi}_i e^{t\mathbf{D}_i} \mathbf{I}$ is the distribution function. Typical examples of PHDs are Erlang distributions, hyperexponential distributions and hypoexponential distributions. Hyperexponential distributions are used to model random variables with a coefficient of variation (cv) larger than 1. An hyperexponential distribution with only two states allows one to describe random variables with an arbitrary positive first moment and an arbitrary finite cv ≥ 1. Erlang and hypoexponential distribution are used to describe distributions with cv < 1. However, the minimal cv for a distribution with k phases or states equals $1/\sqrt{k}$. Parameters of a PHD can derived from data available for the costs of edges (e.g., data about the time to pass a segment of a road in a traffic model) by fitting empirical moments or by a maximum likelihood approach, corresponding algorithms can be found in the literature (Buchholz et al. 2014). Costs described by PHDs with initial distributions $\boldsymbol{\pi}_i \mathbf{I} = 1$ are strictly positive.

We assume that between two nodes at most one edge exists. For some edge $i \in \mathcal{E}$, $i\bullet \in \mathcal{V}$ and $\bullet i \in \mathcal{V}$ are the starting and terminating node. For some node $v \in \mathcal{V}, \circ v \subseteq \mathcal{E}$ and $v\circ \subseteq \mathcal{E}$ are the sets of edges that start and terminate in node v, respectively. On a possible path through the graph $(i\bullet)\circ$ is the set of possible successors of edge i. Similarly, we define $\circ(\bullet i)$ as the set of possible predecessors of edge i.

If PHDs are assigned to edges, the costs of adjacent edges (i.e., edges $i, j \in \mathcal{E}$ with $j \in (i\bullet)\circ$) are independent random variables. However, the model allows one to introduce correlation among edge costs by making the initial state of the PHD of the following edge dependent on the final state of the PHD from previous edge. Figure 1 shows an example instance of a PH-graph where the network of PHDs for two different paths and the graph model are visualized. For two adjacent edges $i, j \in \mathcal{E}$ we define a $n_i \times n_j$ transfer matrix $\mathbf{H}_{ij}$ with $\mathbf{H}_{ij} \geq \mathbf{0}$ and $\mathbf{H}_{ij}\mathbf{I} = -\mathbf{D}_i\mathbf{I}$. Furthermore, $\mathbf{H}_{ij}$ has to be chosen such that for matrix $\mathbf{P}_{ij} = \mathbf{M}_i \mathbf{H}_{ij}$ the condition $\boldsymbol{\pi}_i \mathbf{P}_{ij} = \boldsymbol{\pi}_j$ with $\mathbf{M}_i = (-\mathbf{D}_i)^{-1}$ holds. $\mathbf{H}_{ij}(x, y)$ equals that rate with which state x of the PHD associated to edge i is left towards phase y of the PHD associated to edge j. The conditions on the row sums of matrix $\mathbf{H}_{ij}$ assures that the exit rates from the states of the PHD for edge i are kept. $\mathbf{P}_{ij}$ is a stochastic matrix and its entry $\mathbf{P}_{ij}(x, y)$ includes the conditional probability of starting in state y of the PHD at edge j if the PHD for edge i ($\in \circ(\bullet j)$) has been entered at state x. The conditions on matrix $\mathbf{P}_{ij}$ assure that the distribution of the costs for edge j are not modified, i.e., are defined by PHD $(\boldsymbol{\pi}_j, \mathbf{D}_j)$. Then

$$Cov(X_i, X_j) = \boldsymbol{\pi}_i \mathbf{M}_i \mathbf{P}_{ij} \mathbf{M}_j \mathbf{I} - (\boldsymbol{\pi}_i \mathbf{M}_i \mathbf{I})(\boldsymbol{\pi}_j \mathbf{M}_j \mathbf{I}) \text{ and } \rho_{X_i, X_j} = \frac{Cov(X_i, X_j)}{\sigma_i \sigma_j}$$

(1)

are the covariance and the correlation coefficient of the costs of edge i and j, and

$$E(X_i^k, X_j^l) = k!l!\boldsymbol{\pi}_i \mathbf{M}_i^k \mathbf{H}_{ij} \mathbf{M}_j^l \mathbf{I} \tag{2}$$

is the joint moment of order k, l between the costs of edge i and j. The computations follow from the analysis of MAPs (Buchholz et al. 2014) and have been derived for PHGs in Dohndorf (2017). Algorithms to compute the entries of matrix $\mathbf{H}_{ij}$ based on available estimates for the covariance or joint moments can be found in Buchholz and Felko (2015) and Dohndorf (2017)

For a given PHG, a sequence of edges $(i_1, \ldots, i_K)$ defines a path which can be analyzed using an absorbing CTMC with $n = \sum_{k=1}^{K} n_{i_k}$ states. The state space of the CTMC contains the states of PHDs corresponding to edges along the path. The path matrix $\mathbf{Q}_{(i_1, \ldots, i_K)}$ is a non-singular matrix, such that quantitative properties of the path can be analyzed from an absorbing Markov chain.

The problem of finding an *optimal* path between v_{ini} and v_{fin} in a PHG can be formulated as a stochastic shortest path problem in an MDP with state space

$$\mathcal{S} = \{(i, x) \mid i \in E, \ x \in \{1, \ldots, n_i\}\} \cup \{(0, 0)\},$$

with an absorbing state $(0, 0)$ associated to v_{fin}. For some state (i, x) with $\mathbf{d}_i(x) > 0$ the choice of a particular successor edge from the set $(i\bullet)\circ$ is associated with an action. For some $u \in (i\bullet)\circ$ define

$$\mathbf{Q}^u((i, x), (j, y)) = \begin{cases} \mathbf{D}_i(x, y) & \text{if } j = i, \ i > 0 \\ \mathbf{H}_{iu}(x, y) & \text{if } j = u, \ u \in (i\bullet)\circ, \ i > 0, \\ \mathbf{d}_i(x) & \text{if } j = 0 \text{ and } y = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

For a vector $\mathbf{u}$ of length $|\mathcal{S}|$ with $\mathbf{u}(i, x) \in (i\bullet)\circ$, (3) defines row (i, x) of matrix $\mathbf{Q}^{\mathbf{u}}$.
Example We consider as a running example the small PHG shown in Fig. 1a. There are three paths $i_1 \rightarrow i_4$, $i_2 \rightarrow i_5$ and $i_1 \rightarrow i_3 \rightarrow i_5$ between v_{ini} and v_{fin}.

Costs of the edges i_1 and i_4 are described by a hyperexponential distribution with two phases and the following description.

$$\boldsymbol{\pi}_1 = \boldsymbol{\pi}_4 = (1/9, 8/9), \ \mathbf{D}_1 = \mathbf{D}_4 = \begin{pmatrix} -0.2 & 0 \\ 0 & -2.0 \end{pmatrix}, \ \mathbf{d}_1 = \mathbf{d}_4 = \begin{pmatrix} 0.2 \\ 2.0 \end{pmatrix}$$

This implies $E(X_1^1) = E(X_4^1) = 1$ and $\sigma_1^2 = \sigma_4^2 = 5$. Costs of the edges i_2 and i_5 are given by Erlang-2 distributions with

$$\boldsymbol{\pi}_2 = \boldsymbol{\pi}_5 = (1, 0), \ \mathbf{D}_2 = \mathbf{D}_5 = \begin{pmatrix} -2.0 & 2.0 \\ 0.0 & -2.0 \end{pmatrix}, \ \mathbf{d}_2 = \mathbf{d}_5 = \begin{pmatrix} 0.0 \\ 2.0 \end{pmatrix}$$

with $E(X_2^1) = E(X_5^1) = 1.0$ and $\sigma_2^2 = \sigma_5^2 = 0.5$. Costs of edge i_3 are exponentially distributed with rate 2, i.e.,

$$\boldsymbol{\pi}_3 = (1), \ \mathbf{D}_3 = (-2), \ \mathbf{d}_3 = (2)$$

and $E(X_3^1) = 0.5, \sigma_3^2 = 0.25$. We assume that only the costs of i_1 and i_4 are correlated by matrix

$$\mathbf{H}_{i_1,i_4} = \begin{pmatrix} 0.16 & 0.04 \\ 0.1 & 1.9 \end{pmatrix} \text{ such that } \mathbf{P}_{i_1,i_4} = \begin{pmatrix} 0.80 & 020 \\ 0.05 & 0.95 \end{pmatrix}$$

which implies a positive correlation with $\rho_{14} = 0.32$. Fig. 1b, c show the states and transitions of the MDP for the paths $i_1 \rightarrow i_4$ and $i_2 \rightarrow i_5$.

PH-graph algorithms for shortest path problems By interpreting a PHG as an MDP with a single absorbing state, decision problems can be naturally defined as shortest path problems in an MDP, even if edge costs are correlated.

For our purpose, it is sufficient to consider deterministic but possibly time-dependent policies. A policy is deterministic, if it only depends on the current state and chooses a unique successor edge to leave immediately after the destination node of the current edge. Let $\mathcal{U}$ be the set of deterministic policies $\mathbf{u} = (u_0, \mathbf{u}_1, \ldots, \mathbf{u}_T)$ containing decision vectors to be chosen at each decision epoch $0, \ldots, T$, i.e., $\mathbf{u}_t(i, x) \in (i\bullet)\circ$ for $t > 0$ is the successor edge that is chosen at time t if edge i is left from state x and $u_0 \in v_{ini}\circ$ is the initial edge selected at v_{ini}. For the computation of optimal policies a discretization scheme is applied. For sufficiently small discretization step $h > 0$, $e^{h \, \mathbf{Q}^{\mathbf{u}_k}} = \mathbf{P}_h^{\mathbf{u}_k} + o(h^2)$ holds, and the stochastic matrix $\mathbf{P}_h^{\mathbf{u}_k}$ is defined as

$$\mathbf{P}_h^{\mathbf{u}_k} = \mathbf{I} + h \, \mathbf{Q}^{\mathbf{u}_k}, \tag{4}$$

which is the transition matrix of the DTMC induced by the decision vector $\mathbf{u}_k$. Thus, for each time step of length h, $e^{h \, \mathbf{Q}^{\mathbf{u}_k}}$ is substituted by $\mathbf{P}_h^{\mathbf{u}_k}$ resulting in a global error in $O(h)$ (Miller 1968). To avoid an overloading of notation, we remove the index h which is fixed during optimization. We use here a discretization approach to analyze the continuous time model, alternatively, one may use a uniformization based computation as in Buchholz et al. (2017) which works similarly but requires a slightly more complex derivation of the resulting equations and computes policies that switch the successor edge at arbitrary times. Define for each edge $i \in v_{ini}\circ$ a vector $\boldsymbol{\phi}_i$ of length $|\mathcal{S}|$ such that $\boldsymbol{\phi}_i(j, x) = \boldsymbol{\pi}_i(x)$ if $j = i$ and 0 otherwise.

With the matrices $\mathbf{P}^{\mathbf{u}_k}$, the process becomes a discrete time MDP and standard dynamic programming methods can be applied. We first consider the probability of reaching the absorbing state with a given probability within budget T. Let $N = T/h$ the number of steps of the discrete time process.

Let $\mathbf{g}_N^*$ be a vector of length $|\mathcal{S}|$ with $\mathbf{g}_N^*(i, x) = 1$ if $(i, x) = (0, 0)$ and 0 otherwise. Then the Bellman equations (Puterman 2005) are

$$\mathbf{g}_k^*(i, x) = \max_{u \in (i \bullet) \circ} \left(\sum_{(j,y) \in \mathcal{S}} \mathbf{P}^u((i, x), (j, y)) \mathbf{g}_{k+1}^*(j, y) \right) \qquad \text{and}$$

$$\mathbf{u}_{k+1}^*(i, x) = \arg \max_{u \in (i \bullet) \circ} \left(\sum_{(j,y) \in \mathcal{S}} \mathbf{P}^u((i, x), (j, y)) \mathbf{g}_{k+1}^*(j, y) \right) \tag{5}$$

for all $k = 0, \ldots, N - 1$. Vectors $\mathbf{g}_k^*$ contain the maximal probabilities to reach v_{fin} within $[kh, Nh]$ for each state (i, k) and will be denoted as gain vectors. Computation follows the standard approch to compute policies in stochastic dynamic programming for finite horizon problems (Bertsekas 2005; Puterman 2005). The maximal probability to reach v_{fin} with the given budget is then $G^{\mathbf{u}^*} = \max_{i \in v_{ini} \circ} \boldsymbol{\phi}_i \mathbf{g}_0^*$ and $u_0^* = \arg \max_{i \in v_{ini} \circ} (\boldsymbol{\phi}_i \mathbf{g}_0^*)$. Then u_0^* is the first edge which is chosen when v_{ini} is left at time time $t = 0$ and $\mathbf{u}^* = (u_0^*, \mathbf{u}_1^*, \ldots, \mathbf{u}_N^*)$ is the policy describing the choice of successor edges at times kh with $k = 0, \ldots, N$. An optimization algorithm starts at N and evaluates (5) backwards. The following theorem shows that the maximization problem (5) can also be formulated as a minimization problem which might become necessary if different goal functions are combined.

Theorem 1 *Let* $\mathbf{g}_k^*, \mathbf{u}_{k+1}^*,$ *for* $k = 0, \ldots, N - 1,$ *be the gain and policy vectors computed from* (5) *and let* $\mathbf{g}_k', \mathbf{u}_{k+1}'$ *be the gain and policy vectors resulting from* (5) *with the minimum operator and initial vectors* $\mathbf{g}_N' = \mathbf{I} - \mathbf{g}_N,$ *then* $\mathbf{g}_k' = \mathbf{I} - \mathbf{g}_k^*,$ *decisions can be selected such that* $\mathbf{u}_k^* = \mathbf{u}_k'$ *for* $k = 0, \ldots, N - 1$ *and* $u_0^* = u_0'$ *can be chosen such that* $u_0^* = \arg \max_{i \in v_{ini}} (\boldsymbol{\phi}_i \mathbf{g}_0^*)$ *and* $u_0' = \arg \min_{i \in v_{ini}} (\boldsymbol{\phi}_i \mathbf{g}_0')$ *are identical.*

Proof $\mathbf{g}_N = \mathbf{I} - \mathbf{g}_N'$ holds by assumption. We show that when $\mathbf{g}_k = \mathbf{I} - \mathbf{g}_k'$, then $\mathbf{g}_{k-1} = \mathbf{I} - \mathbf{g}_{k-1}'$ and if $\mathbf{u}(k)$ maximizes the values for $\mathbf{g}_k$, then it minimizes the values for $\mathbf{g}_k'$. The computations for one state $(i, x) \in \mathcal{S}$ equals

$$\begin{aligned}
\mathbf{g}_{k-1}'(i, x) &= \min_{u \in (i \bullet) \circ} \left(\sum_{(j,y) \in \mathcal{S}} \mathbf{P}^u((i, x), (j, y)) \mathbf{g}_k'(j, y) \right) \\
&= \min_{u \in (i \bullet) \circ} \left(\sum_{(j,y) \in \mathcal{S}} \mathbf{P}^u((i, x), (j, y))(1 - \mathbf{g}_k(j, y)) \right) \\
&= 1 - \max_{u \in (i \bullet) \circ} \left(\sum_{(j,y) \in \mathcal{S}} \mathbf{P}^u((i, x), (j, y)) \mathbf{g}_k(j, y) \right) = 1 - \mathbf{g}_{k-1}(i, x)
\end{aligned}$$

The identity holds since $\mathbf{P}^{\mathbf{u}}$ is a stochastic matrix and the same u can be chosen in both cases. The identity of u_0^* and u_0' follows since $\mathbf{g}_0' = \mathbf{I} - \mathbf{g}_0^*$. □

Equation 5 considers the probability of reaching v_{fin} within the budget irrespective of the costs with which the absorbing state is reached. To include costs, define a reward vector $\mathbf{r} \in \mathbb{R}_{\geq 0}^{|\mathcal{S}| \times 1}$ such that $\mathbf{r}(i, x)$ is the reward of staying for one time unit in state (i, x). The following Bellman equations describe the maximization of the accumulated reward in $[0, T = Nh]$ where $\mathbf{f}_N^* = \mathbf{0}$.

$$\mathbf{f}_k^*(i, x) = \max_{u \in (i\bullet)\circ} \left(\mathbf{r}(i, x) + \sum_{(j,y)\in\mathcal{S}} \mathbf{P}^u((i, x), (j, y))\mathbf{f}_{k+1}^*(j, y) \right) \qquad \text{and}$$

$$\mathbf{u}_{k+1}^*(i, x) = \arg \max_{u \in (i\bullet)\circ} \left(\mathbf{r}(i, x) + \sum_{(j,y)\in\mathcal{S}} \mathbf{P}^u((i, x), (j, y))\mathbf{f}_{k+1}^*(j, y) \right) \tag{6}$$

Then $u_0^* = \arg \max_{i \in v_{ini}\circ}(\boldsymbol{\phi}_i \mathbf{f}_0^*)$ is the initial edge, $\mathbf{u}^* = (u_0^*, \mathbf{u}_1^*, \ldots, \mathbf{u}_N^*)$ the policy and $F^{\mathbf{u}^*} = \boldsymbol{\phi}_{u_0^*}\mathbf{f}_0^*$ is the maximal reward. For minimization of the reward, the maximum in the above equation is substituted by a minimum. Often the costs of the edges describe the relevant rewards such that it is sufficient to let $\mathbf{r}(i, x) = 1$ if $(i, x) \neq (0, 0)$ and $\mathbf{r}(0, 0) = 0$. Then $\bar{\mathbf{f}} = \lim_{k\to\infty} \mathbf{f}_k$, computed with (6) and the minimum operator, is the vector of the minimal expected costs to reach v_{fin}. Let $\bar{\mathbf{u}}$ be the corresponding policy which does not depend on the time step k (Bertsekas 2007, Chap. 2) and $\bar{F}^{\bar{\mathbf{u}}}$ the expected costs (i.e., the expected length of the shortest path). Observe that $\bar{\mathbf{u}}$ and $\bar{\mathbf{f}}$ can be computed with well known algorithms for MDPs like policy iteration, value iteration or linear programming (Puterman 2005, Chap. 8). Theorem 1 can be extended to (6) if all states $(i, x) \neq (0, 0)$ have the same reward.

For some policy $\mathbf{u} = (u_0, \mathbf{u}_1, \ldots, \mathbf{u}_N)$ let

$$\mathbf{g}_k^{\mathbf{u}} = \mathbf{P}^{u_{k+1}}\mathbf{g}_{k+1}^{\mathbf{u}}, \ G^{\mathbf{u}} = \boldsymbol{\phi}_{u_0}\mathbf{g}_0^{\mathbf{u}}, \ \mathbf{f}_k^{\mathbf{u}} = \mathbf{r} + \mathbf{P}^{u_{k+1}}\mathbf{f}_{k+1}^{\mathbf{u}} \ \text{and} \ F^{\mathbf{u}} = \boldsymbol{\phi}_{u_0}\mathbf{f}_0^{\mathbf{u}} \tag{7}$$

for $k = 0, \ldots, N - 1$ with $\mathbf{g}_N^{\mathbf{u}}(i, x) = 1$ for $(i, x) = (0, 0)$ and 0 otherwise and $\mathbf{f}_N^{\mathbf{u}} = \mathbf{0}$. Obviously $\mathbf{g}_{k-1}^* = \max_{\mathbf{u}(k:N)}\left(\mathbf{g}_k^{\mathbf{u}(k:N)}\right)$ and $\mathbf{f}_{k-1}^* = \max_{\mathbf{u}(k:N)}\left(\mathbf{f}_k^{\mathbf{u}(k:N)}\right)$, where $\mathbf{u}(k : N) = (\mathbf{u}_k, \ldots, \mathbf{u}_N)$ equals policy $\mathbf{u}$ restricted to the interval $[kh, Nh]$. Then $\mathbf{g}_k^{\mathbf{u}} = \mathbf{g}_k^{\mathbf{u}(k:N)}$ and $\mathbf{f}_k^{\mathbf{u}} = \mathbf{f}_k^{\mathbf{u}(k:N)}$. We denote by $\mathcal{U}^k$ the set of all possible policies $\mathbf{u}(k : N)$, i.e, $\mathbf{u}(k : N) = (\mathbf{u}_k, \ldots, \mathbf{u}_N)$ and $\mathbf{u}_l(i, x) \in (i\bullet)\circ$ for $k \leq l \leq N$. Both vectors $\mathbf{g}_k^{\mathbf{u}}$ and $\mathbf{f}_k^{\mathbf{u}}$ are denoted in the sequel as gain vectors.

Example (continued) We begin with the problem of minimizing the accumulated costs to reach v_{fin} from v_{ini} where all non absorbing states have costs 1. This corresponds to the classical shortest path problem which we solve with policy iteration (Bertsekas 2007, Chap. 3). If we neglect the correlation between the costs of i_1 and i_4, then the expected costs of reaching v_{fin} from v_{ini} are 2 for path $i_1 \to i_4$ and path $i_2 \to i_5$, whereas path $i_1 \to i_3 \to i_5$ has costs of 2.5. With correlation the optimal policy starts with i_1 and chooses i_4 if i_1 has been passed via phase 2, if i_1 has been passed via phase 1, then i_3 followed by i_5 is chosen as successor. This policy results in expected costs of 1.811.

For the maximization of the probability to reach v_{fin} with a given budget a discretization $h = 0.05$ is chosen and the policy, which is time-dependent, is computed with (5). If the budget is ≥ 3.2, then path $i_2 \to i_5$ is selected. The small variance of the Erlang distributions assures that it is unlikely to exceed the budget but it is also unlikely to pass the edges with low costs. With a budget ≤ 3.15 the optimal paths starts with i_1, if i_1 is left from phase 1 with a budget ≤ 2.1, then i_3 followed by i_5 is selected, otherwise i_4 is chosen as next edge. In this case, first the edge with hyperexponentially distributed costs is chosen. The hyperexponential distribution has a large variance. If i_1 is left after a small number of steps (i.e., with a large budget), it is the best choice

to exploit the positive correlation and choose i_4 as the next edge. Otherwise, if i_1 is left with a large number of steps (i.e, with a small remaining budget), the positive correlation of the costs at edge i_4 would have a negative effect and it is better to choose the path over i_3 and i_5 with uncorrelated costs.

3 Optimization of the scalarized multi-objective problem

In many settings, one is interested in policies which behave well with respect to different measures, e.g., the probability to reach the destination with a given budget should be high but the expected length of the shortest path should not be too long. One way to compute such policies is to minimize or maximize the weighted sum of the goal functions. In this section, we show that this problem can often be mapped on an MDP with a single goal function. In the following section it is shown how the convex hull of all gain vectors that are optimal according to some preference vector including the weights is computed.

We consider the convex linear combination of C goal functions. Let $T_c > 0$ and $T_c \geq T_{c-1}$ ($1 < c \leq C$) be the time bounds of the different goal functions and $\mathbf{r}^c \geq \mathbf{0}$ the corresponding reward vectors, if a reward vector is required for goal function c. We assume that $N_c = T_c/h$ are integers. The following two sets are needed $\mathcal{N}_k = \{c | N_c = k\}$ and $\mathcal{N}_{\geq k} = \cup_{k \leq l \leq N_C} \mathcal{N}_l$. The resulting method is commonly denoted as weighting factor method (White [1982]). Since a convex linear combination of the goal functions is considered let $\mathbf{w} \in \mathbb{R}_{>0}^{1,C}$ be the preference vector with $\mathbf{w} \, \mathbf{I} = 1$. For notational convenience we define $\omega_k = \sum_{c \in \mathcal{N}_{\geq k}} \mathbf{w}(c)$. Let $\delta_c = 1$ if goal c computes a probability according to (5) and $\delta_c = 0$ if goal c computes a reward according to (6).

Let $\mathbf{u} = (u_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N_C})$ be some policy vector and let $\mathbf{h}_k^{\mathbf{u}}$ for $k = 0, \ldots, N_C$ be vectors of length $|\mathcal{S}|$ that store the costs of the sum of the different goal functions multiplied with the weights from the preference vector. Define a vector $\mathbf{h}_{ini} \in \mathbb{R}^{|\mathcal{S}|,1}$ with $\mathbf{h}_{ini}(0, 0) = 1$ and $\mathbf{h}_{ini}(i, x) = 0$ for $(i, x) \neq (0, 0)$. The costs for the composed goal function are then computed from

$$\mathbf{h}_k^{\mathbf{u}} = \frac{\omega_{k+1}}{\omega_k} \left(\sum_{c \in \mathcal{N}_{\geq k+1} \wedge \delta_c = 0} \frac{\mathbf{w}(c)}{\omega_{k+1}} \mathbf{r}^c + \mathbf{P}^{\mathbf{u}_{k+1}} \mathbf{h}_{k+1}^{\mathbf{u}} \right) + \frac{\sum_{c \in \mathcal{N}_k} \delta_c \mathbf{w}(c)}{\omega_k} \mathbf{h}_{ini} \qquad (8)$$

with

$$\mathbf{h}_{N_C}^{\mathbf{u}} = \frac{\sum_{c \in \mathcal{N}_{N_C}} \delta_c \mathbf{w}(c)}{\omega_{N_C}} \mathbf{h}_{ini}. \qquad (9)$$

For c with $\delta_c = 1$ let $\mathbf{f}_k^{\mathbf{u},c}$ and for $\delta_c = 0$ let $\mathbf{g}_k^{\mathbf{u},c}$ ($0 \leq k \leq N_c$) be the gain vectors for goal function c under policy $\mathbf{u}$. The following theorem shows the correspondence between the costs for the different goal functions and $\mathbf{h}_k^{\mathbf{u}}$.

Theorem 2 *For $k = 0, \ldots, N_C$ the relation*

$$\mathbf{h}_k^{\mathbf{u}} = \sum_{c \in \mathcal{N}_{\geq k} \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{g}_k^{\mathbf{u},c} + \sum_{c \in \mathcal{N}_{\geq k} \wedge \delta_c = 0} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{f}_k^{\mathbf{u},c}$$

holds.

Proof We prove the results by induction starting with $k = N_C$. Observe that for all policies $\mathbf{u}$ and c with $\delta_c = 1$, $\mathbf{g}_{N_c}^{\mathbf{u},c} = \mathbf{h}_{ini}$ and for c with $\delta_c = 0$ $\mathbf{f}_{N_C}^{\mathbf{u}} = \mathbf{0}$. Thus, we have

$$\mathbf{h}_{N_C}^{\mathbf{u}} = \frac{\sum_{c \in \mathcal{N}_{N_C}} \delta_c \mathbf{w}(c)}{\omega_{N_C}} \mathbf{h}_{ini} = \sum_{c \in \mathcal{N}_{N_C} \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_{N_C}} \mathbf{h}_{ini} = \sum_{c \in \mathcal{N}_{N_C} \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_{N_C}} \mathbf{g}_{N_C}^{\mathbf{u},c}$$

Now assume that the theorem has been proved for $k + 1$, we show that it also holds for k.

$$\begin{aligned}
\mathbf{h}_k^{\mathbf{u}} &= \frac{\omega_{k+1}}{\omega_k} \left(\sum_{c \in \mathcal{N}_{\geq k+1} \wedge \delta_c = 0} \frac{\mathbf{w}(c)}{\omega_{k+1}} \mathbf{r}^c + \mathbf{P}^{\mathbf{u}_{k+1}}(i) \mathbf{h}_{k+1}^{\mathbf{u}} \right) + \frac{\sum_{c \in \mathcal{N}_k} \delta_c \mathbf{w}(c)}{\omega_k} \mathbf{h}_{ini} \\
&= \sum_{c \in \mathcal{N}_{\geq k+1} \wedge \delta_c = 0} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{r}^c + \sum_{c \in \mathcal{N}_{\geq k+1} \wedge \delta_c = 0} \frac{\mathbf{w}(c)}{\omega_{k+1}} \mathbf{P}^{\mathbf{u}_{k+1}}(i) \mathbf{f}_{k+1}^{\mathbf{u},c} + \\
&\quad \sum_{c \in \mathcal{N}_{\geq k+1} \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{P}^{\mathbf{u}_{k+1}}(i) \mathbf{g}_{k+1}^{\mathbf{u},c} + \sum_{c \in \mathcal{N}_k \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_{k+1}} \mathbf{g}_k^{\mathbf{u}} \\
&= \sum_{c \in \mathcal{N}_{\geq k+1} \wedge \delta_c = 0} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{f}_{k+1}^{\mathbf{u},c} + \sum_{c \in \mathcal{N}_{\geq k} \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{g}_k^{\mathbf{u},c}
\end{aligned}$$

which proves the theorem because $\mathbf{f}_k^{\mathbf{u},c} = \mathbf{0}$ for $k = N_c$. □

To compute an optimal policy, assume that the partial policy $\mathbf{u}^*(k+1 : N)$ and the vectors $\mathbf{h}_{k+1}^{\mathbf{u}(k+1:N)}, \ldots, \mathbf{h}_{N_C}$ are available. The decision vector at k then equals

$$\mathbf{u}_k^*(i, x) = \arg \max_{u \in \mathcal{U}(i)} \left(\sum_{(j,y) \in \mathcal{S}} \mathbf{P}^u((i, x)(j, y)) \mathbf{h}_k^{\mathbf{u}^*(k+1:N)} \right), \tag{10}$$

where $\mathbf{u}^*(k : N) = (\mathbf{u}_k^*, \mathbf{u}^*(k+1 : N))$ and $u_0^* = \arg \max_{i \in v_{ini}} \circ \left(\boldsymbol{\phi}_i \mathbf{h}_0^{\mathbf{u}(1:N)} \right)$, resulting in policy $\mathbf{u}^* = (u_{ini}^*, \mathbf{u}_1^*, \ldots, \mathbf{u}_N^*)$.

Algorithm 1 Algorithm to compute an optimal policy and the costs for a preference vector.

1: Initialize $\mathbf{h}_{N_C}$ using (9) and $\mathbf{u}^{N_C+1} = \emptyset$;
2: **for** $k = N_C$ downto 1 **do**
3:     Compute $\mathbf{u}_k$ using (10);
4:     $\mathbf{u}(k : N_C) = (\mathbf{u}_k, \mathbf{u}(k + 1 : N_C))$;
5:     Compute $\mathbf{h}_{k-1}$ using (8) and policy vector $\mathbf{u}_k$;
6: **end for**
7: $u_0 = \arg\max_{i \in v_{ini} \circ} (\boldsymbol{\phi}_i \mathbf{h}_0)$;
8: $H^* = \boldsymbol{\phi}_{u_0} \mathbf{h}_0^{\mathbf{u}(1:N_C)}$, $\mathbf{u}^* = (u_0, \mathbf{u})$;

Theorem 3 *For vectors $\mathbf{h}_k$ computed with Algorithm 1 the following relation holds*

$$\mathbf{h}_k = \max_{\mathbf{u} \in \mathcal{U}^k} \left(\sum_{c \in \{1,...,C\} \wedge \delta_c = 0} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{f}_k^{\mathbf{u},c} + \sum_{c \in \{1,...,C\} \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_k} \mathbf{g}_k^{\mathbf{u},c} \right)$$

Proof According to Theorem 2 vector $\mathbf{h}_k$ contains the sum of costs multiplied with the weights from the preference vector for the different goal functions under policy $\mathbf{u}(k : N_C)$. For $k = N_C$ the vector is given by the initial vector which is independent of any decision. Equation 10 selects decisions that maximize the sum of costs in step k multiplied with the preference values. Thus, we can argue recursively that in each step decisions are chosen that maximize the vector $\mathbf{h}$. Since the Bellman optimality criterion holds, the result holds.   □

It is, of course, possible to substitute in (10) the maximum operator by a minimum to compute minimal costs. Using Theorem 1 it is also possible to transform a minimization problem in a maximization problem and vice versa. Thus, it is possible to combine minimization and maximization in a single problem. E.g., to find a policy with a large probability to meet a deadline and a small expected length of the shortest path. The combination of minimization and maximization reaches its limits, if we have a minimization problem with edge dependent rewards and a maximization problem with edge dependent rewards which cannot be combined.

Algorithm 1 can only be applied for finite horizons of all goal functions. Now assume that we have C goal functions with finite horizons N_c ($c = 1, \ldots, C$) and B additional goal functions, numbered $C + 1, \ldots, C + B$ with an infinite horizon. We assume that all infinite horizon goal functions have a finite optimal solution, i.e., they are minimization problems or they are maximization problems in an acyclic graph. This implies that only problems with accumulated rewards (Eq. 6) are of interest since the probability of eventually reaching v_{fin} from v_{ini} is 1 in the mentioned settings. The B goal functions with an infinite horizon can then be combined by defining a new reward vector

$$\bar{\mathbf{r}} = \sum_{b=C+1}^{C+B} \frac{\mathbf{w}(b)}{\vartheta_B} \mathbf{r}^b$$

where $\vartheta_B = \sum_{b=C+1}^{C+B} \mathbf{w}(b)$. For the MDP with reward vector $\bar{\mathbf{r}}$, the shortest (or longest) path can be computed (White 1982), resulting in a vector with the optimal costs and an optimal policy for the weighted sum of costs. Let $\bar{\mathbf{f}}$ be the resulting costs and $\bar{\mathbf{u}}$ be the resulting policy which only depends on the state but not on the time. $\bar{\mathbf{f}}$ and $\bar{\mathbf{u}}$ are computed using one of the available methods to compute optimal policies for average reward MDPs over infinite horizons (Puterman 2005, Chap. 8). To solve the entire problem $\mathbf{h}_{N_C}$ is then initialized with

$$
\mathbf{h}_{N_C} = \sum_{c \in \mathcal{N}_C \wedge \delta_c = 1} \frac{\mathbf{w}(c)}{\omega_{N_C}} \mathbf{h}_{ini} + \sum_{b=C+1}^{C+B} \frac{\mathbf{w}(b)}{\omega_{N_C}} \bar{\mathbf{f}}.
$$

Observe that $\omega_{N_c} = \sum_{c \mathcal{N}_{N_C}} \mathbf{w}(c) + \sum_{C+1}^{C+B} \mathbf{w}(c)$ in this case. By adding $\bar{\mathbf{f}}$ multiplied with the relative weights for goal functions without a time bound, the approach takes care of the situation that after N_C steps v_{fin} has not been reached but the expected accumulated reward to reach v_{fin} is considered in some measures (i.e., $B > 0$). Then $\bar{\mathbf{u}}$ is the best policy at step $k > N_C$ and $\bar{\mathbf{f}}(i, x)$ are the remaining costs to reach v_{fin}. Then Algorithm 1 can be applied to compute the solution of the complete problem.

Example (continued) We first consider the computation a policy that maximizes the probability to reach v_{fin} with $T_1 = 2$, $T_2 = 3$ and $T_3 = 4$ ($N_c = 40, 60, 80$ for $c = 1, 2, 3$). We begin with preference vector $\mathbf{w} = (1/3, 1/3, 1/3)$ that results in a policy that start with i_1 and selects i_3 afterwards if i_1 is left from phase 1 and the budget is 3 or below 2.2, otherwise i_4 is selected as subsequent edge.

4 Solving multi-objective PH-graphs

For a given preference vector, a dynamic programming approach which computes an optimal piecewise constant policy has been introduced in the previous section. However, often the preference vector $\mathbf{w}$ is not known in advance or cannot be defined (see Roijers et al. 2013 for some examples). Then it is important to know the set of policies that are optimal for some preference vector. We slightly extend the notation and define $\mathbf{u}_{\mathbf{w}}^*$ as an optimal policy computed for preference vector $\mathbf{w}$, $H_{\mathbf{w}}^*$ are the corresponding costs. All values are computed in Algorithm 1. For some policy $\mathbf{u}$ define $G_{\mathbf{u}}^c = \boldsymbol{\phi}_{u_0} \mathbf{f}_0^{\mathbf{u},c}$ if $\delta_c = 0$ and $G_{\mathbf{u}}^c = \boldsymbol{\phi}_{u_0} \mathbf{g}_0^{\mathbf{u},c}$ if $\delta_c = 1$. Obviously $H_{\mathbf{w}}^* = \sum_{c=1}^{C} \mathbf{w}(c) G_{\mathbf{u}^*}^c$.

The following set of policies is defined.

$$
\mathcal{CU} = \left\{ \mathbf{u} \mid \mathbf{u} \in \mathcal{U} \wedge \exists \mathbf{w} \, \forall \mathbf{u}' \in \mathcal{U} : H_{\mathbf{w}}^{\mathbf{u}} \geq H_{\mathbf{w}}^{\mathbf{u}'} \right\}
$$

The set is often referred as a convex coverage set - a convex hull of Pareto optimal policies (Roijers et al. 2013). Set $\mathcal{CU}$ as it is defined might still contain redundant policies, i.e., a policy is redundant, if it can be removed from the set and the required condition still holds. Thus, it is sufficient to define set $\mathcal{CU}$ such that for each $\mathbf{w}$ a policy $\mathbf{u} \in \mathcal{CU}$ exists such that $H_{\mathbf{w}}^{\mathbf{u}} \geq H_{\mathbf{w}}^{\mathbf{u}'}$ for all $\mathbf{u}' \in \mathcal{U}$. An approximation of this set can

often be computed with an acceptable effort, as shown in the sequel of this section whereas the computation of the whole set of Pareto optimal policies is a non-trivial task with a complexity strongly depending on the dimension of the preference vector and the number of existing policies (Chatterjee et al. 2006).

Computation of optimal policies The iterative method proposed in Roijers et al. (2014b) finds optimal policy for some preference vectors $\mathbf{w}$ by solving a finite number of linearly scalarized problems using the Algorithm 1 as a subroutine. Let $\mathbf{w}^1, \ldots, \mathbf{w}^L$ be a set of preference vectors for which policies have been computed (i.e., for which Algorithm 1 has been used). Let $\mathbf{u}^1, \ldots, \mathbf{u}^J$ be the corresponding set of policies. Observe that $J \leq L$ but not necessarily $J = L$ because some preference vectors may result in the same policy. Define

$$\boldsymbol{\theta}_j = \left(G^1_{\mathbf{u}^j}, \ldots, G^C_{\mathbf{u}^j} \right) \text{ and } \boldsymbol{\Theta} = \begin{pmatrix} \boldsymbol{\theta}_1 & -1 \\ \vdots & \vdots \\ \boldsymbol{\theta}_J & -1 \end{pmatrix}$$

as the result vectors of the analyzed policies and the matrix of these result vectors. Then let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_C, \xi_{C+1})^T$ be a vector. Any non-negative solution of

$$\boldsymbol{\Theta}\boldsymbol{\xi} \leq \mathbf{0} \text{ and } \sum_{c=1}^{C} \xi_c = 1, \boldsymbol{\xi} \geq \mathbf{0}, \tag{11}$$

defines an upper bound ξ_{C+1} for the costs resulting from the available policies $\mathbf{u}^1, \ldots, \mathbf{u}^J$ under some preference vector. This can be seen by considering $\boldsymbol{\theta}_j(\xi_1, \ldots, \xi_C)^T$. For some preference vector $(\xi_1, \ldots, \xi_C)$ the value equals the costs under policy $\mathbf{u}^j$ and ξ_{C+1} has to be chosen larger than or equal to this value to obtain a value ≤ 0. A solution of (11) with $\boldsymbol{\theta}_j(\xi_1, \ldots, \xi_C)^T = \xi_{C+1}$ for some j is a minimal solution that cannot be reduced by any of the policies $\mathbf{u}^1, \ldots, \mathbf{u}^J$.

We can define a second polyhedron to bound possible costs from above. Define

$$\boldsymbol{\Psi} = \begin{pmatrix} \mathbf{w}^1(1) & \cdots & \mathbf{w}^1(C) \\ \vdots & \vdots & \vdots \\ \mathbf{w}^L(1) & \cdots & \mathbf{w}^L(C) \end{pmatrix} \text{ and } \boldsymbol{\chi} = \begin{pmatrix} H^*_{\mathbf{w}^1} \\ \vdots \\ H^*_{\mathbf{w}^L} \end{pmatrix}$$

as the matrix of already analyzed preference vectors and the vector of resulting costs. Let $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_C)^T$ be a vector of potential costs of the goal functions $c = 1, \ldots, C$ (i.e., upper bounds for $G^c_{\mathbf{u}}$ under some unknown policy $\mathbf{u}$). An upper bound for the costs is then given by

$$\boldsymbol{\Psi}\boldsymbol{\psi} \leq \boldsymbol{\chi}. \tag{12}$$

This can be seen by considering $(\mathbf{w}^l(1), \ldots, \mathbf{w}^l(C))\boldsymbol{\psi}$ which has to be smaller than or equal to $H^*_{\mathbf{w}^l}$ which is the optimal value for this preference vector.

Equation 11 defines the lower surface for preference vectors and the corresponding costs. Equation 12 defines the upper surface for the costs of the different goal functions. To compute the preference vector that results in the largest difference between lower and upper bound of the resulting costs, the following problem has to be solved.

$$\max_{\boldsymbol{\psi}, \boldsymbol{\xi}} \left((\boldsymbol{\psi}^T, -1) \, \boldsymbol{\xi} \right)$$

under the constraints $\boldsymbol{\Psi}\boldsymbol{\psi} \leq \boldsymbol{\chi}$, $\boldsymbol{\Theta}\boldsymbol{\xi} \leq \mathbf{0}$ and $\sum_{c=1}^{C} \xi_c^j = 1$, $\boldsymbol{\psi}, \boldsymbol{\xi} \geq \mathbf{0}$ (13)

Theorem 4 *The optimization problem* (13) *computes the maximal difference between a solution which can be derived with the policies* $\mathbf{u}^1, \ldots, \mathbf{u}^J$ *and a hypothetical solution that results from a preference vector that respects the optimal solutions that have been computed for the preference vectors* $\mathbf{w}^1, \ldots, \mathbf{w}^L$.

Proof Vector $\boldsymbol{\psi}$ contains potential costs for the goal functions $1, \ldots, C$. These costs have to observe $\boldsymbol{\Psi}\boldsymbol{\psi} \leq \boldsymbol{\chi}$ because otherwise the already computed optimal solution for the preference vectors $\mathbf{w}^1, \ldots, \mathbf{w}^L$ would no longer be optimal. The first C elements of vector $\boldsymbol{\xi}$ define a preference vector and the last element a value for the weighted sum of costs resulting from the preference vector multiplied with the costs for the different policies. Since we are computing a maximum, the accumulated costs have to be at least as large as the costs resulting from one of the already analyzed policies which implies that for the solution the relation $\geq \mathbf{0}$ has to hold. The goal function can then be interpreted as $\sum_{c=1}^{C} \boldsymbol{\xi}(c)\boldsymbol{\psi}^j(c) - \boldsymbol{\psi}^j(C+1)$, the difference between the upper bound for the costs under the chosen weight vector and the costs for some policy under this weight vector. By choosing the maximum among all already analyzed policies, this results in the computation of a preference vector belonging to the maximal difference between a point belonging to a preference vector on the upper surface and a point belonging to the same preference vector on the lower surface. $\square$

Equation 13 is a bilinear optimization problem (Nahapetyan 2009) which is not convex. However, if we fix $\boldsymbol{\psi}$ or $\boldsymbol{\xi}$, the problem becomes a linear program. Since the two vector have no joint constraints, the optimal solution $(\boldsymbol{\psi}^*, \boldsymbol{\xi}^*)$ consists of extreme points of the polyhedra defined by the constraints. The solution algorithm proposed in Roijers et al. (2014b) computes the extreme points for polyhedron $\boldsymbol{\Theta}\boldsymbol{\xi} \leq \mathbf{0}$ and solves the linear program for the preference vectors resulting from these extreme points. However, it seems to be preferable to solve (13) by one of the available solvers for bilinear problems which use cutting planes or branch and bound methods (Nahapetyan 2009; Sherali and Alameddine 1992).

Algorithm 2 computes an approximation of the set $\mathcal{CU}$. New policies and vectors with costs are added until the difference between upper and lower bound falls beyond a threshold ϵ. The effort of the proposed algorithm depends on the number of states of the MDP, the size of $\mathcal{CU}$ which can be exponential in the problem size and the number of preference vectors that have to be analyzed. In each iteration a bilinear optimization problem (Eq. 13) and an MDP problem (Eqs. 8 and 9) have to be solved. If the number of different goal functions is not too large, the computation of an optimal policy for the MDP requires most of the time. For larger number of different goal functions,

Algorithm 2 Approximation of the set $\mathcal{CU}$.

1: Initialize $\mathbf{w}^1 = (1, 0, \ldots, 0), \ldots, \mathbf{w}^C = (0, \ldots, 0, 1)$ and set $L = C$;
2: Compute for $\mathbf{w}^l$ optimal policies $\mathbf{u}^j$ and vectors with costs using Algorithm 1 ($l = 1, \ldots, L, j = 1, \ldots, J, J \leq L$);
3: Compute $G_{\mathbf{u}^j}^c$ for $c = 1, \ldots, C, j = 1, \ldots, J$;
4: **while** the maximal value resulting from (13) $> \epsilon$ **do**
5:     Add preference vector $\mathbf{w}^{L+1} = (\boldsymbol{\xi}(1), \ldots, \boldsymbol{\xi}(C))$ and set $L = L + 1$;
6:     For preference vector $\mathbf{w}^L$ compute an optimal policy $\mathbf{u}$ and costs $H_{\mathbf{w}^L}^*$ using (8,9);
7:     **if** $\mathbf{u} \notin \{\mathbf{u}^1, \ldots, \mathbf{u}^J\}$ **then**
8:         Add $\mathbf{u}$ as $\mathbf{u}^{J+1}$ and set $J = J + 1$;
9:         Compute $G_{\mathbf{u}^J}^1, \ldots, G_{\mathbf{u}^J}^C$;
10:     **end if**
11: **end while**
12: **return** $\mathbf{w}^1, \ldots, \mathbf{w}^L$ and $\mathbf{u}^1, \ldots, \mathbf{u}^J$;
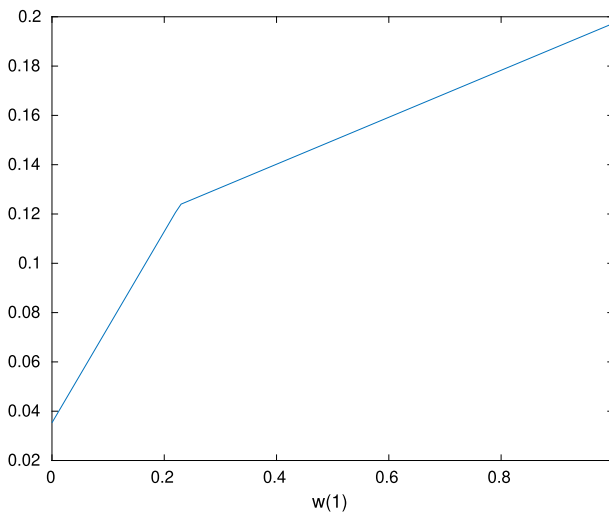
Fig. 2 Convex hull of the Pareto optimal policies for the weighted sum of the minimal probabilities to reach the destination with budget 2 and 4

beyond 5–10, the number of policies in $\mathcal{CU}$ often grows quickly and makes the problem practically infeasible for smaller values of ϵ.

Example (continued) We consider the maximization of the weighted sum of the probabilities to reach the destination with a budget of 2 and 4. Figure 2 shows the convex hull for the simple example and the probabilities of not reaching the destination with budget 2 and 4. For this simple example only two different policies have to be considered.

5 Examples

In this section, we illustrate the applicability and effectiveness of the proposed multi-objective approaches with two application examples.

Scheduling problem in the cloud Scheduling with stochastic task execution times is of interest in many applications and has received significant attention over the years. Within soft real-time systems, Markovian analysis techniques become more and more popular, e.g. Markovian methods have been applied to model the system using GSPNs and CTMCs and to analyze the probability of task deadline misses (Manolache et al. 2002), or to analyze the average system utilization (Diaz et al. 2002).

Consider a scheduling problem on a single machine in a sequence where task execution times are stochastic and possible correlations between execution time distributions can occur. In practice, the stochasticity of task execution time is motivated due to the large number of environmental run-time parameters which can depend on the application, the hardware and software platform. Typically, the key factors influencing the execution time are determined by the hardware-architecture of the processing units, the program structure, the level of parallelism, the data interchange, e.g. database and web accesses etc. However, in the real-time field, the task execution times can be correlated, e.g. due to the amount of data to be processed. Obviously, the existing dependencies between execution time distributions should be accurately considered when the tasks have to be executed in a sequence.

We propose the PH-graph model to model stochastic execution times of tasks in a uniprocessor system and to analyze scheduling of tasks with time slot constraints in a cloud service environment. Reserving a time unit for job processing on the cloud is typically associated with some costs. Cloud users anticipate completing their tasks within a reserved time slot which is much cheaper than on-demand pricing options when instances are charged on a pay-as-used basis. However, the execution of tasks is linked to economic benefits, e.g. number of orders processed, number of verified transactions, SLA-awareness, load balancing. Hence, one is dealing with the optimization of these conflicting criteria simultaneously. The goal is to find a compromise between tight time deadline compliance and reward maximization which is linked to a profit associated with the job running time.

Consider three tasks T_1, T_2 and T_3 running in a sequence within a cloud service environment. Execution times of these tasks on a single processor are modeled by the order 2 hyper-exponential and hypoexponential PHDs PH_{T_1}, PH_{T_2}, and PH_{T_3} with the following representation

$$\pi_{T_1} = (0.5, 0.5), \ \mathbf{D}_{T_1} = \begin{pmatrix} -10 & 0 \\ 0 & -0.25 \end{pmatrix}, \ \mathbf{D}_{T_2} = \begin{pmatrix} -14 & 0 \\ 0 & -0.11 \end{pmatrix}, \ \mathbf{D}_{T_3} = \begin{pmatrix} -5.24 & 5.24 \\ 0 & -0.789 \end{pmatrix},$$

where $\pi_{T_2} = \pi_{T_3} = \pi_{T_1}$. The means and variances of the execution times are shown in Table 1.

Table 1 First conditional moment and variance of PHDs describing task execution time

PHD	μ	σ^2
$(\pi_{T_1}, \mathbf{D}_{T_1})$	2.05	11.80
$(\pi_{T_2}, \mathbf{D}_{T_2})$	4.53	60.43
$(\pi_{T_3}, \mathbf{D}_{T_3})$	1.36	1.63

(a) Task graph

(b) PH-paths representing two scheduling policies $T_1 T_2 T_3$ and $T_1 T_3 T_2$
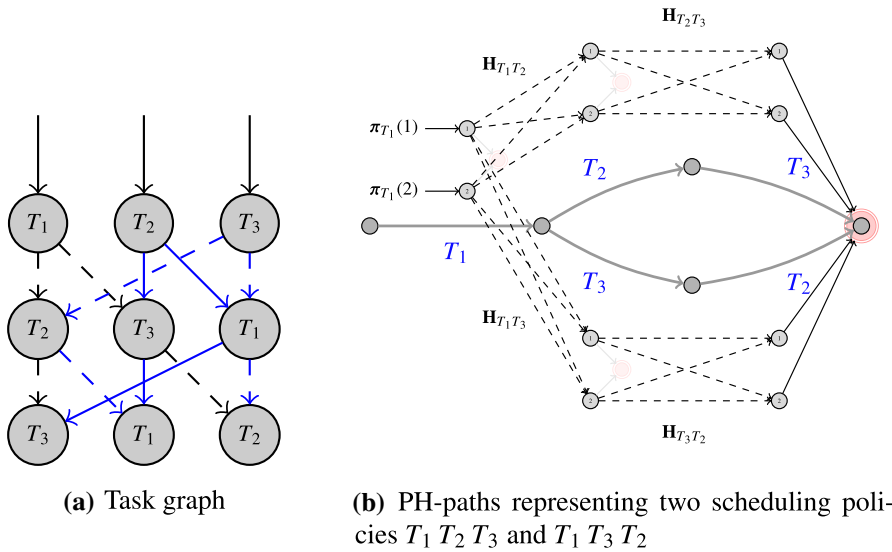
Fig. 3 PH-paths used to model stochastic task scheduling

Furthermore, the data processing and output performed by the task T_1 influence the input of the task T_2 and thus its execution time. The scenario is realistic when both tasks are working on the same data trace. We assume that the distribution of the execution time of the task T_1 is correlated with the distribution of the execution time of T_2 by the matrix

$$\mathbf{H}_{T_1 T_2} = \begin{pmatrix} 10 & 0 \\ 0 & 0.25 \end{pmatrix}$$

which results in a positive correlation $\rho = 0.325$. The execution times of the remaining tasks are all independent. However, there are different possibilities to schedule the tasks within a predefined time slot. The task graph and two corresponding PH-paths are visualized in Fig. 3.

The analysis of the scheduling policies in the system is shown in Fig. 4 where varying time slots for the tasks completions have been considered.

Consider two scheduling policies $T_1 T_2 T_3$ and $T_1 T_3 T_2$. The effect of correlation between execution times of the task T_1 and the task T_2 can be shown in Fig. 5 where conditional transient probabilities and conditional accumulated reward of the remaining schedules $T_2 T_3$ and $T_3 T_2$ in dependence of the realized execution time of the task T_1, which is the first task in sequence, have been analyzed, such that different scheduling policies are optimal for different optimization goals.

The multi-objective problem is analyzed in order to compute scheduling policies with high weighted probability to satisfy a tight time deadline and high economical profit constraint simultaneously. The resulting PH-graph has 36 states. Results are presented in Table 2 and in Fig. 6 for a very tight time deadline $T = 2.5$, and in Fig. 7 for a larger time deadline $T = 12$. The discretization parameter is set to a small

Reachability probability for different time horizon value T
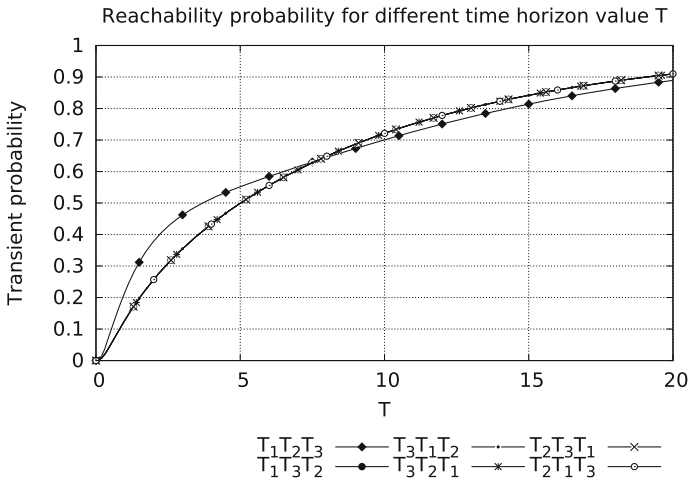
(a) The probability of completing all tasks in the time slot T

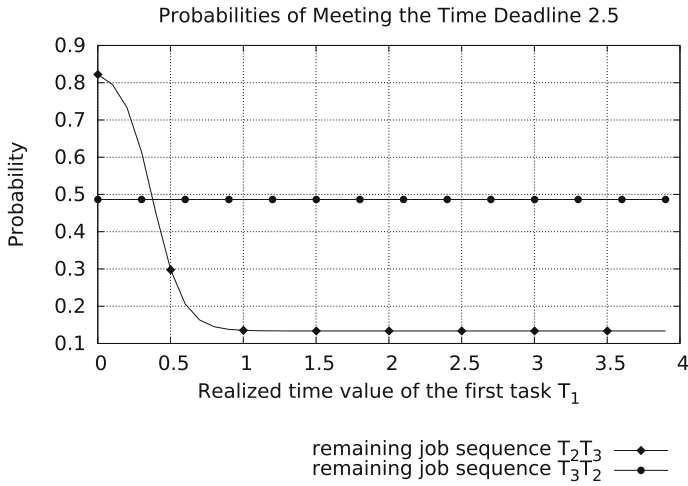Accumulated reward for different time horizon value T
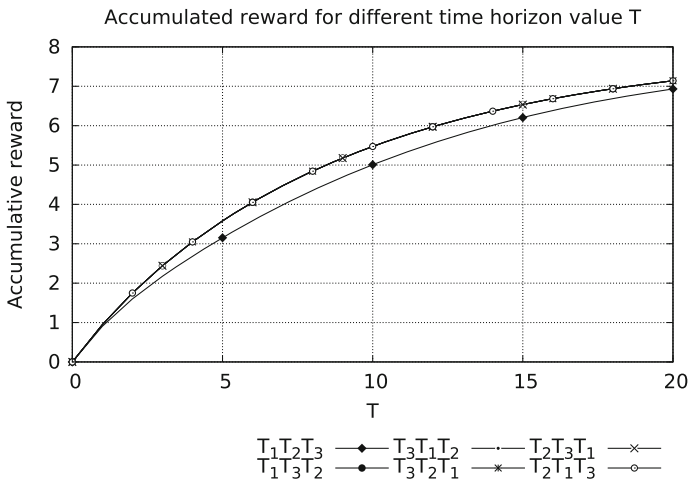
(b) The expected accumulated reward in the time interval T

Fig. 4 Probability of reaching the absorbing state and the expected accumulated reward over the finite horizon

value $h = 0.0001$ and the approximation of the set $\mathcal{CU}$ is performed for time horizon $T = 2.5$ and $T = 12$. In the former case the set $\mathcal{CU}$ contains 4 policies and in the latter 2 policies.

Shortest paths in random road networks (Eisenstat 2010) proposed a random road network generation method based on quadtrees. These quadtree models show many useful features which are essential in modeling and analysis of real road networks and relevant algorithms. In particular, quadtree based road networks represent planar, self-similar and multi-scale-disperse graphs with reasonable variations in density of

Probabilities of Meeting the Time Deadline 2.5

(a) Conditional transient probability

Accumulated Reward for the Time Horizon 2.5
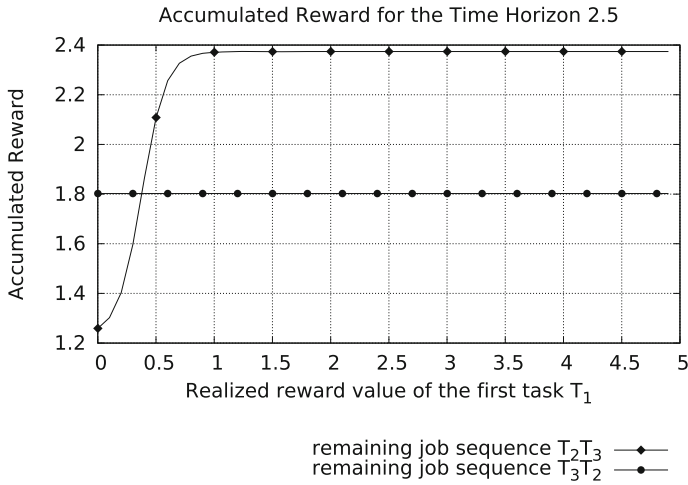
(b) Conditional cumulative reward

Fig. 5 Probability of completing the execution of tasks $T_2\,T_3$ and $T_3\,T_2$ within the short time slot $T = 2.5$ depending on the realized execution time of the task T_1, which is the first task in sequence. The accumulated reward of the remaining tasks $T_2\,T_3$ and $T_3\,T_2$ is visualized in Fig. 5b

Table 2 Pareto optimal policies computed for a short time horizon $T = 2.5$	Policy $\mathbf{u}^j, \boldsymbol{\theta}_j$	Weight vector $\mathbf{w}$	$H_{\mathbf{w}}^{\mathbf{u}^j}$
	(1.2695, 0.7475)	(0, 1)	0.7476
	(2.3304, 0.1576)	(0.3775, 0.6225)	0.9778
	(2.4869, 0.0179)	(0.5742, 0.4258)	1.4357
	(2.4987, 0.0021)	(1, 0)	2.4987
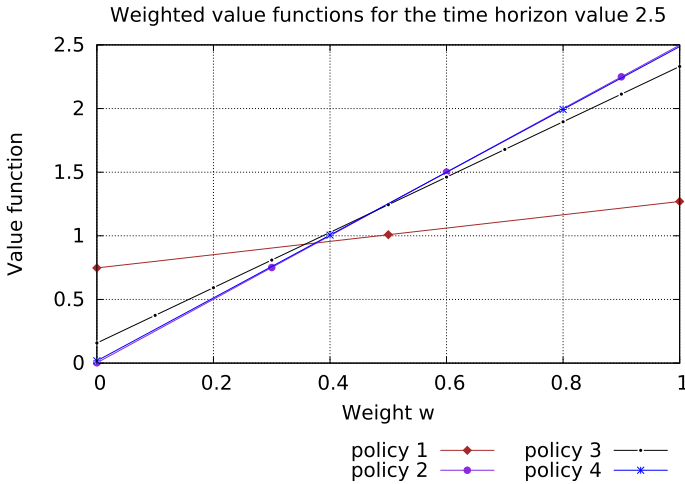
Fig. 6 The set of scalarized value functions for Pareto optimal solutions obtained in the short time slot T = 2.5, which is the time less than the expected path length in the PH-graph
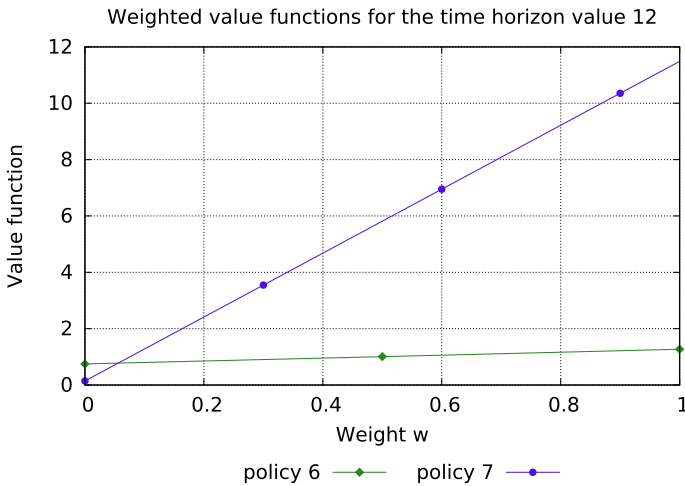

Fig. 7 The set of scalarized value functions for Pareto optimal solutions obtained in the time slot T = 12

modeled roads and in the number of road intersections. Each node in the quadtree represents a square embedded in a plane. In order to resemble realistic road networks each node could have up to n children which represents the division of the square into quadrants of equal size. Squares are added to the quadtree by recursively dividing squares into quadrants, i.e. adding the child nodes. The subdivision can be performed according to some predefined parameters, like distribution of road intersections, number of peripheral or high way roads and amount of sprawl.

We used this method to generate a random road instance which describes some realistic road network. We generated the desired number of squares such that each path from the leftmost top node v_{ini} to the rightmost bottom node v_{fin} contains 32

Fig. 8 A random quadtree road network instance. The mean weight of each possible path from v_{ini} to v_{fin} is given by $\mu = 30.55$
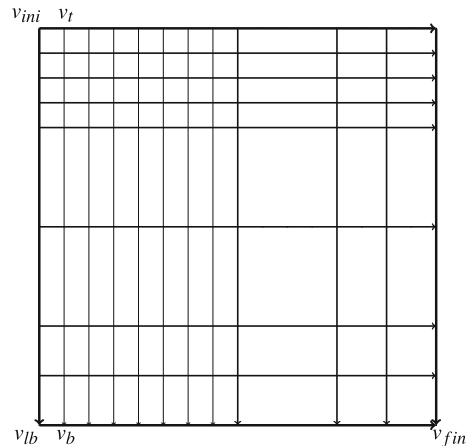

edges. Figure 8 shows the generated quadtree to model some realistic roads. Each edge in the quadtree forms a road connecting two vertices in the square. Edges are directed from top to bottom and from left to right and are parameterized with Phase-type distributions introduced in the running example in Fig. 1a. We used the hyperexponential distribution to describe the weights of the path running from the initial node v_{ini} to the leftmost bottom node v_{lb} and from the node v_{lb} to the final node v_{fin}. Additionally, the path running from the node v_t to the node v_b is also modeled using hyperexponential PHD. We assume that costs of edges along these paths are correlated. Costs of the remaining edges are given by the introduced Erlang-2 distribution and are uncorrelated. Even though the generated quadtree road network assures a relatively small node degree, each road intersection doubles the number of strategies. We computed results for the maximization of the weighted sum of the probabilities to reach the node v_{fin} with a budget slightly below the expected path length, i.e. $T = 30$, and $T = 42$.

For the weighted maximization problem 7 different strategies have been computed for $\epsilon = 0.01$ within 60.27 s using a MATLAB implementation of the algorithms on a PC with 3.2 GHz processor. Figure 9 shows the corresponding convex hull and the probabilities of not reaching the final node within the specified time bounds and Table 3 summarizes the results for different optimal strategies. The required time to compute the approximation of the Pareto optimal policies in quadtrees indicates that small road networks can be analyzed fairly quickly and that PH-graphs have a potential to be applied in online navigation scenarios in cities.

6 Conclusions

In this paper, we present a multi-objective approach for solving multiple criteria PH-graphs to analyze Stochastic Shortest Path Problems with several goal functions. Based on available approaches for MDPs, we develop a value iteration based algorithm for finite horizon MDPs to compute the optimal policy for the weighted combination of reachability probabilities with given budgets and cumulative rewards in stochastic
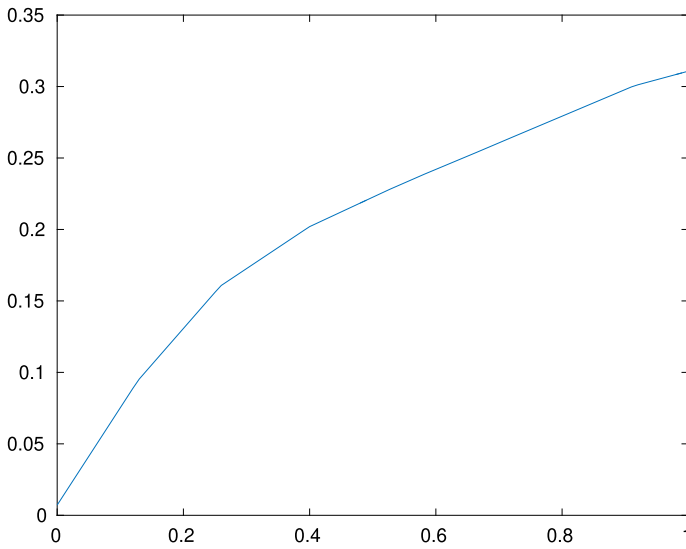
Fig. 9 Convex hull of the Pareto optimal policies for the weighted sum of the minimal probabilities to reach the destination v_{fin} with budget 30 and 42

Table 3 Pareto optimal policies computed for the maximization of the weighted sum of the probabilities to reach the destination in the quadtree with a budget of 30 and 42

Policy $\mathbf{u}^j$, $\boldsymbol{\theta}_j$	Weight vector $\mathbf{w}$	$H_{\mathbf{w}}^{\mathbf{u}^j}$
(0.6892, 0.8086)	(1, 0)	0.6892
(0.3120, 0.9928)	(0, 1)	0.9928
(0.6214, 0.9157)	(0.3281, 0.6719)	0.8191
(0.6835, 0.8698)	(0.6122, 0.3878)	0.7557
(0.4646, 0.9704)	(0.1995, 0.8005)	0.8695
(0.6745, 0.8803)	(0.4880, 0.5120)	0.7799
(0.4646, 0.9704)	(0.1983, 0.8017)	0.8701

graphs with correlated edge weights. To characterize the set of policies that are optimal with respect to some preference vector including weights for the different goals, an algorithm is introduced that approximates the convex hull of Pareto optimal policies. This algorithm, that is based on earlier work in the context of sequential decision making, repeatedly solves a bilinear optimization problem and applies then the value iteration based approach for some weight vector to reduce an upper bound for the approximation error in each step. For small bounds for the approximation error, the algorithm computes a precise approximation of the convex hull of Pareto optimal policies.

In this paper a simple discretization approach is used to solve the continuous time models. It is also possible to solve these models with recently developed methods based on uniformization resulting in adaptive decision points. The corresponding algorithms can be easily integrated in the developed framework. In future work, multi-objective PH-graphs should be extended by decisions which are based on edges in the PH-graph

and not on states in the corresponding MDP. Then, the number of implemented policies can be reduced but the same MDP algorithms can still be used in order to compute Pareto optimal policies.

References

Barbarosoğlu G, Arda Y (2004) A two-stage stochastic programming framework for transportation planning in disaster response. J Oper Res Soc 55(1):43–53

Bertsekas DP (2005) Dynamic programming and optimal control, vol I. Athena Scientific, Belmont

Bertsekas DP (2007) Dynamic programming and optimal control, vol II. Athena Scientific, Belmont

Bouyer P, González M, Markey N, Randour M (2018) Multi-weighted Markov decision processes with reachability objectives. In: Andrea O, Martin Z (eds) Proceedings ninth international symposium on games, automata, logics, and formal verification, GandALF 2018, Saarbrücken, Germany, 26–28th September 2018, volume 277 of EPTCS, pp 250–264

Buchholz P, Felko I (2015) PH-graphs for analyzing shortest path problems with correlated traveling times. Comput OR 59:51–69

Buchholz P, Hahn EM, Hermanns H, Zhang L (2011) Model checking algorithms for CTMDPs. In: CAV 2011, USA, pp 225–242

Buchholz P, Kriege J, Felko I (2014) Input modeling with phase-type distributions and Markov models: theory and applications. Springer briefs in mathematics. Springer, Berlin

Buchholz P, Dohndorf I, Scheftelowitsch D (2017) Optimal decisions for continuous time Markov decision processes over finite planning horizons. Comput OR 77:267–278

Budd JK (2016) Modelling credit card usage for individual card-holders. Ph.D. thesis, The University of Melbourne, Melbourne, Australia

Chatterjee K, Majumdar R, Henzinger TA (2006) Markov decision processes with multiple objectives. In: Bruno D, Wolfgang T (eds) STACS 2006, 23rd annual symposium on theoretical aspects of computer science, Marseille, France, February 23–25, 2006, proceedings, volume 3884 of lecture notes in computer science, pp 325–336. Springer

Diaz JL, Garcia DF, Kim K, Lee C-G, Lo BL, Lopez JM, Min SL, Mirabella O (2002) Stochastic analysis of periodic real-time systems. In: 23rd IEEE real-time systems symposium, 2002. RTSS 2002, pp 289–300

Dohndorf I (2017) Stochastic graph models with phase type distributed edge weights. Ph.D. thesis, TU Dortmund, Dortmund, Germany

Eisenstat D (2010) Random road networks: the quadtree model. CoRR, arXiv:1008.4916

Fan YY, Kalaba RE, Moore JE (2005a) Shortest paths in stochastic networks with correlated link costs. Comput Math Appl 49(9–10):1549–1564

Fan YY, Kalaba RE, Moore JE II (2005b) Arriving on time. J Optim Theory Appl 127(3):497–513

Forejt V, Kwiatkowska MZ, Norman G, Parker D, Qu H (2011) Quantitative multi-objective verification for probabilistic systems. In: Abdulla PA, Leino KRM (eds) Tools and algorithms for the construction and analysis of systems: 17th international conference, TACAS 2011, volume 6605 of lecture notes in computer science, pp 112–127. Springer

Ghosh MK (1990) Markov decision processes with multiple costs. Op. Res. Lett. 9(4):257–260

Ji Z, Kim YS, Chen A (2011) Multi-objective alpha-reliable path finding in stochastic networks with correlated link costs. Expert Syst Appl 38(3):1515–1528

Koenig M, Meissner J (2015) Value-at-risk optimal policies for revenue management problems. Int J Prod Econ 166:11–19

Li G, Zhang Q, Feng Z (2016) Research on optimal path of data migration among multisupercomputer centers. Sci Program 2016:5018213:1–5018213:8

Manolache S, Eles P, Peng Z (2002) Schedulability analysis of multiprocessor real-time applications with stochastic task execution times. In: Pileggi LT, Kuehlmann A (eds) Proceedings of the 2002 IEEE/ACM international conference on computer-aided design, ICCAD 2002, San Jose, California, USA, November 10–14, 2002. ACM/IEEE Computer Society, pp 699–706

Miller BL (1968) Finite state continuous time Markov decision processes with a finite planning horizon. SIAM J Control 6(2):266–280

Nahapetyan AG (2009) Bilinear programming. In: Floudas CA, Pardalos PM (eds) Encyclopedia of optimization, 2nd edn. Springer, Berlin, pp 279–282

Nain P, Towsley D (2016) File dissemination in dynamic graphs: the case of independent and correlated links in series. TOMPECS 2(1):4:1–4:23

Neuts MF (1979) A versatile Markovian point process. J Appl Probab 16:764–779

Neuts Marcel F (1981) Matrix geometric solutions in stochastic models: an algorithmic approach. Dover Publications Inc., New York

Nie YM, Wu X (2009) Reliable a priori shortest path problem with limited spatial and temporal dependencies. In: Lam W, Wong S, Lo H (eds) Transportation and traffic theory 2009: Golden Jubilee, chapter 9. Springer, Boston, pp 169–195

Nikolova E, Karger DR (2008) Route planning under uncertainty: the canadian traveller problem. In: Fox D, Gomes CP (eds) Proceedings of the twenty-third AAAI conference on artificial intelligence, AAAI 2008, Chicago, Illinois, USA, July 13–17, 2008. AAAI Press, pp 969–974

Nikolova E, Brand M, Karger DR (2006) Optimal route planning under uncertainty. In: Long D, Smith SF, Borrajo D, McCluskey L (eds) Proceedings of the sixteenth international conference on automated planning and scheduling, ICAPS 2006, Cumbria, UK, June 6–10, 2006. AAAI, pp 131–141

Puterman ML (2005) Markov decision processes. Wiley, New York

Randour M, Raskin JF, Sankur O (2015) Variations on the stochastic shortest path problem. In: Souza DD, Lal A, Larsen KG (eds) Verification. Model checking, and abstract interpretation (VMCAI). Volume 8931 of lecture notes in computer science. Springer, Berlin, pp 1–18

Rezvanian A, Reza Meybodi M (2016) Stochastic graph as a model for social networks. Comput Hum Behav 64:621–640

Roijers DM, Vamplew P, Whiteson S, Dazeley R (2013) A survey of multi-objective sequential decision-making. J Artif Intell Res 48:67–113

Roijers DM, Scharpff J, Spaan MTJ, Oliehoek FA, de Weerdt M, Whiteson S (2014a) Bounded approximations for linear multi-objective planning under uncertainty. In: Chien SA, Do MB, Fern A, Ruml W (eds) Proceedings of the twenty-fourth international conference on automated planning and scheduling, ICAPS 2014, Portsmouth, New Hampshire, USA, June 21–26, 2014

Roijers DM, Whiteson S, Oliehoek FA (2014b) Linear support for multi-objective coordination graphs. In: AAMAS '14, Paris, France, pp 1297–1304

Roijers DM, Whiteson S, Oliehoek FA (2015) Computing convex coverage sets for faster multi-objective coordination. J Artif Intell Res 52:399–443

Samaranayake S, Blandin S, Bayen A (2012) A tractable class of algorithms for reliable routing in stochastic networks. Transp Res C Emerg Technol 20(1):199–217

Serfozo RF (1979) An equivalence between continuous and discrete time Markov decision processes. Oper Res 27(3):616–620

Sherali HD, Alameddine A (1992) A new reformulation-linearization technique for bilinear programming problems. J Glob Optim 2(4):379–410

Wakuta K, Togawa K (1998) Solution procedures for multi-objective Markov decision processes. Optimization 43(1):29–46

White DJ (1982) Multi-objective infinite-horizon discounted Markov decision processes. J Math Anal Appl 89(2):639–647

Zhang Y, Jun Y, Wei G, Lenan W (2010) Find multi-objective paths in stochastic networks via chaotic immune PSO. Expert Syst Appl 37(3):1911–1919

Zhaowang J, Anthony C, Subprasom K (2004) Finding multi-objective paths in stochastic networks: a simulation-based genetic algorithm approach. In: Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753), vol 1, pp 174–180