Cost Preserving Bisimulations for Probabilistic Automata

Holger Hermanns and Andrea Turrini

Saarland University – Computer Science, Saarbrücken, Germany

Abstract. Probabilistic automata constitute a versatile and elegant model for concurrent probabilistic systems. They are equipped with a compositional theory supporting abstraction, enabled by weak probabilistic bisimulation serving as the reference notion for summarising the effect of abstraction.
This paper considers probabilistic automata augmented with costs. It extends the notions of weak transitions in probabilistic automata in such a way that the costs incurred along a weak transition are captured. This gives rise to cost-preserving and cost-bounding variations of weak probabilistic bisimilarity. Polynomial-time decision algorithms are proposed, that can be effectively used to compute reward-bounding abstractions of Markov decision processes.

1 Introduction

Markov Decision Processes (*MDP*s) are mathematical models widely used in operations research, automated planning, decision support systems and related fields. In the concurrent systems context, they appear in the form of Probabilistic Automata (*PA*s) [18]. *PA*s form the backbone model of successful model checkers such as PRISM [12] enabling the analysis of randomised concurrent systems. They extend classical concurrency models in a simple yet conservative fashion, by enabling probabilistic experiments inside transitions.

As one of the classical concurrency theory manifestations, weak probabilistic bisimilarity is a congruence relation for parallel composition and hiding on *PA*. In other contexts, this has enabled powerful compositional minimisation approaches to combat the state space explosion problem in explicit state verification approaches [6, 10, 15]. With the conception of a polynomial time algorithm for deciding weak probabilistic bisimilarity [11] this avenue can now be followed also in the context of *PA*s and *MDP*s. The decision algorithm follows the usual partition refinement approach. At its core, the decision algorithm needs to check a polynomial number of linear programming (LP) problems. Each of them checks the existence of a specific weak transition. The decision algorithm can be turned into a minimisation algorithm, producing a minimal canonical representation of the *PA* with respect to weak probabilistic bisimilarity [7].

MDP models are usually decorated with cost or rewards structures, with the intention to minimise costs or maximise rewards along the model execution. Likewise, in tools like PRISM, *PA*s appear augmented with cost or reward structures. It is hence a natural question how costs can be embedded into the approach discussed above, and this is what the paper is about.

We propose Cost Probabilistic Automata (*CPA*s), a model where *cost* is any kind of quantity associated with the transitions of the automata, and we aim to minimise the cost. For instance, we can consider as the cost of a transition the power needed to transmit a message, the time spent in the computation modelled by the transition, the (monetary) risk associated with an action, the expense of some work, and so on. Costs for weak transitions are interpreted in line with the vast body of literature on *MDP*s, and we describe how that interpretation can be linked to the weak transition encoding as LP problems.

We then extend weak probabilistic bisimulation to also account for costs. As a strict option, we require weak transition costs to be matched exactly for bisimilar states, inducing *cost-preserving weak probabilistic bisimulation*. As a weaker alternative, we ask them to be bounded from one *PA* to the other, leading to the notion of *minor cost weak probabilistic bisimulation*. We provide polynomial time algorithms for both variations. Finally we present an application of minor cost weak probabilistic bisimulation to a multi-hop wireless communication scenario where the cost structure represents transmission power which in turn depends on physical distances.

Organisation of the paper. After the preliminaries in Section 2, we revisit the LP problem formulation behind weak probabilistic bisimilarity in Section 3 and we present cost probabilistic automata and relative bisimulations in Section 4 together with the wireless channel example. We discuss related work and possible extensions in Section 5 and we conclude the paper in Section 6 with some remarks.

2 Mathematical Preliminaries and Probabilistic Automata

For a set X, denote by $\mathrm{Disc}(X)$ the set of discrete probability distributions over X, and by $\mathrm{SubDisc}(X)$ the set of discrete sub-probability distributions over X. Given $\rho \in \mathrm{SubDisc}(X)$, we denote by $\mathrm{Supp}(\rho)$ the set $\{\, x \in X \mid \rho(x) > 0 \,\}$, by $\rho(\bot)$ the value $1 - \rho(X)$ where $\bot \notin X$, and by δ_x, where $x \in X \cup \{\bot\}$, the *Dirac* distribution such that $\rho(y) = 1$ for $y = x$, 0 otherwise. For a sub-probability distribution ρ, we also write $\rho = \{\, (x, p_x) \mid x \in X \,\}$ where p_x is the probability of x. The lifting $\mathcal{L}(\mathcal{R})$ [14] of a relation $\mathcal{R} \subseteq X \times Y$ is defined as: for $\rho_X \in \mathrm{Disc}(X)$ and $\rho_Y \in \mathrm{Disc}(Y)$, $\rho_X \ \mathcal{L}(\mathcal{R}) \ \rho_Y$ holds if there exists a *weighting function* $w \colon X \times Y \to [0, 1]$ such that (1) $w(x, y) > 0$ implies $x \ \mathcal{R} \ y$, (2) $\sum_{y \in Y} w(x, y) = \rho_X(x)$, and (3) $\sum_{x \in X} w(x, y) = \rho_Y(y)$.

A Probabilistic Automaton (*PA*) $\mathcal{A}$ is a tuple $(S, \bar{s}, \Sigma, D)$, where S is a countable set of *states*, $\bar{s} \in S$ is the *start state*, Σ is a countable set of *actions*, and $D \subseteq S \times \Sigma \times \mathrm{Disc}(S)$ is a *probabilistic transition relation*. The set Σ is divided in two sets $\mathtt{H}$ and $\mathtt{E}$ of internal (hidden) and external actions, respectively; we let s,t,u,v, and their variants with indices range over S; a, b range over actions; and τ range over internal actions. In this work we consider only finite *PA*s, i.e., *PA*s such that S and D are finite.

A Markov Decision Process (*MDP*) $\mathcal{M}$ is a tuple (S, ι, Σ, P, r) that can be considered as a variation of a *PA* with a functional transition relation $P \colon S \times \Sigma \to \mathrm{Disc}(S)$, a start distribution $\iota \in \mathrm{Disc}(S)$ instead of a start state, and additionally a *reward function* or *structure* $r \colon S \times \Sigma \to \mathbb{R}$. In this paper we consider only non-negative rewards, i.e., $r(s, a) \geq 0$ for each $(s, a) \in S \times \Sigma$, but interpret them as *costs*.

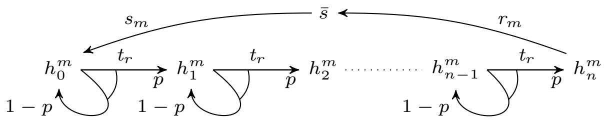

Fig. 1. The wireless communication channel $WCC(n, r, p)$

A transition $tr = (s, a, \mu) \in D$, also denoted by $s \xrightarrow{a} \mu$, is said to *leave* from state s, to be *labelled* by a, and to *lead* to the *target* distribution μ, also denoted by μ_{tr}. We denote by $src(tr)$ the *source* state s and by $act(tr)$ the *action* a. We also say that s enables action a, that action a is enabled from s, and that (s, a, μ) is enabled from s. Finally, we let $D(a) = \{\, tr \in D \mid act(tr) = a \,\}$ be the set of transitions with label a.

Example (A wireless communication channel). As an example of *PA*s, consider a reliable wireless communication channel used to transmit messages belonging to the set Msg from a sender to a receiver. The wireless implementation of the communication channel is depicted in Fig. 1: the *PA* $WCC(n, r, p)$ models a communication that requires n intermediate nodes (hops) to reach the receiver where the probability to transmit correctly the message from each node to the successor is p. Each intermediate node has a transmission radius r, and this parameter will become useful when determining the transmission cost in terms of power consumed. In this *PA*, the message m to transmit is obtained from the sender via the external s_m action and it is delivered to the receiver by using the external action r_m. Internal action t_r models the transmission of the message m from one node to the successor distant at most r, the transmission radius.

The ideal communication channel is modelled by the *PA* $ICC = WCC(0, \infty, 1)$, that is, the automaton that does not require intermediate nodes. Obviously, ICC models a reliable communication channel since the message is delivered with probability 1 just after having received it.

An *execution fragment* of a *PA* $\mathcal{A}$ is a finite or infinite sequence of alternating states and actions $\alpha = s_0 a_1 s_1 a_2 s_2 \ldots$ starting from a state s_0, also denoted by $\mathit{first}(\alpha)$, and, if the sequence is finite, ending with a state denoted by $last(\alpha)$, such that for each $i > 0$ there exists a transition $(s_{i-1}, a_i, \mu_i) \in D$ such that $\mu_i(s_i) > 0$. The *length* of α, denoted by $|\alpha|$, is the number of occurrences of actions in α. If α is infinite, then $|\alpha| = \infty$. Denote by $\mathit{frags}(\mathcal{A})$ the set of execution fragments of $\mathcal{A}$ and by $\mathit{frags}^*(\mathcal{A})$ the set of finite execution fragments of $\mathcal{A}$. An execution fragment α is a *prefix* of an execution fragment α', denoted by $\alpha \leqslant \alpha'$, if the sequence α is a prefix of the sequence α'. The *trace* $trace(\alpha)$ of α is the sub-sequence of external actions of α; we denote by ε the empty trace and we define $trace(a) = a$ for $a \in \mathtt{E}$ and $trace(a) = \varepsilon$ for $a \in \mathtt{H}$.

A *scheduler* for a *PA* $\mathcal{A}$ is a function $\sigma \colon \mathit{frags}^*(\mathcal{A}) \to \mathrm{SubDisc}(D)$ such that for each $\alpha \in \mathit{frags}^*(\mathcal{A})$, $\sigma(\alpha) \in \mathrm{SubDisc}(\{\, tr \in D \mid src(tr) = last(\alpha) \,\})$. In the *MDP* context, a scheduler is known as *policy* $\pi \colon \mathit{frags}^*(\mathcal{A}) \to \mathrm{Disc}(D)$. Given a scheduler σ and a finite execution fragment α, the distribution $\sigma(\alpha)$ describes how transitions are chosen to move on from $last(\alpha)$. A scheduler σ and a state s induce a probability distribution $\mu_{\sigma,s}$ over execution fragments as follows. The basic measurable events are the cones of finite execution fragments, where the cone of α, denoted by C_α, is the set $\{\, \alpha' \in \mathit{frags}(\mathcal{A}) \mid \alpha \leqslant \alpha' \,\}$. The probability $\mu_{\sigma,s}$ of a cone C_α is defined recursively as follows:

$$\mu_{\sigma,s}(C_\alpha) = \begin{cases} 0 & \text{if } \alpha = t \text{ for a state } t \neq s, \\ 1 & \text{if } \alpha = s, \\ \mu_{\sigma,s}(C_{\alpha'}) \cdot \sum_{tr \in D(a)} \sigma(\alpha')(tr) \cdot \mu_{tr}(t) & \text{if } \alpha = \alpha' a t. \end{cases}$$

Standard measure theoretical arguments ensure that $\mu_{\sigma,s}$ extends uniquely to the σ-field generated by cones. We call the resulting measure $\mu_{\sigma,s}$ a *probabilistic execution fragment* of $\mathcal{A}$ and we say that it is generated by σ from s. Given a finite execution fragment α, we define $\mu_{\sigma,s}(\alpha)$ as $\mu_{\sigma,s}(\alpha) = \mu_{\sigma,s}(C_\alpha) \cdot \sigma(\alpha)(\bot)$, where $\sigma(\alpha)(\bot)$ is the probability of terminating the computation after α has occurred.

We say that there is a *weak combined transition* from $s \in S$ to $\mu \in \mathrm{Disc}(S)$ labelled by $a \in \Sigma$, denoted by $s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$, if there exists a scheduler σ such that the following holds for the induced probabilistic execution fragment $\mu_{\sigma,s}$: (1) $\mu_{\sigma,s}(\mathit{frags}^*(\mathcal{A})) = 1$; (2) for each $\alpha \in \mathit{frags}^*(\mathcal{A})$, if $\mu_{\sigma,s}(\alpha) > 0$ then $\mathit{trace}(\alpha) = \mathit{trace}(a)$; (3) for each state t, $\mu_{\sigma,s}(\{\, \alpha \in \mathit{frags}^*(\mathcal{A}) \mid \mathit{last}(\alpha) = t \,\}) = \mu(t)$. In this case, we say that the weak combined transition $s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$ is induced by σ.

Albeit the definition of weak combined transitions is somewhat intricate, this definition is just the obvious extension of weak transitions on labelled transition systems to the setting with probabilities. See [19] for more details on weak combined transitions.

Example (cont'd). Consider again the *PA* $WCC(2, r, \frac{3}{4})$ and the weak combined transition $h_0^m \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \delta_{h_1^m}$. In order to show that it is actually a weak combined transition of the *PA* $WCC(2, r, \frac{3}{4})$, we have to exhibit a scheduler σ inducing it. It is easy to verify that σ defined as: $\sigma(\alpha) = \delta_{h_0^m \xrightarrow{\tau} \rho}$ if $\mathit{last}(\alpha) = h_0^m$, $\delta_\bot$ otherwise, where $\rho = \{(h_1^m, \frac{3}{4}), (h_0^m, \frac{1}{4})\}$, induces the transition $h_0^m \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \delta_{h_1^m}$. Consider, for instance, the probability of stopping in h_1^m, i.e., the sum of the probability of each finite execution fragment ending with h_1^m, i.e., execution fragments of the form $(h_0^m \tau)^{n+1} h_1^m (\tau h_1^m)^l$ where $l, n \in \mathbb{N}$; it is easy to derive that for $n \in \mathbb{N}$, $\mu_{\sigma,h_0^m}((h_0^m \tau)^{n+1} h_1^m) = (\frac{1}{4})^n \cdot \frac{3}{4} \cdot 1 = (\frac{1}{4})^n \cdot \frac{3}{4}$ and that for $l, n \in \mathbb{N}$, $\mu_{\sigma,h_0^m}((h_0^m \tau)^{n+1} h_1^m (\tau h_1^m)^{l+1}) = 0$. Hence we have that $\mu_{\sigma,h_0^m}(\{\, \alpha \in \mathit{frags}^*(\mathcal{A}) \mid \mathit{last}(\alpha) = h_1^m \,\}) = \mu_{\sigma,h_0^m}(\{\, (h_0^m \tau)^{n+1} h_1^m \mid n \in \mathbb{N} \,\}) + \mu_{\sigma,h_0^m}(\{\, (h_0^m \tau)^{n+1} h_1^m (\tau h_1^m)^{l+1} \mid l, n \in \mathbb{N} \,\}) = \sum_{n \in \mathbb{N}} (\frac{1}{4})^n \cdot \frac{3}{4} + 0 = 1 = \delta_{h_1^m}(h_1^m)$.

Weak probabilistic bisimilarity [18, 19] is of central importance for our considerations.

Definition 1. *Let $\mathcal{A}_1$, $\mathcal{A}_2$ be two PAs. An equivalence relation $\mathcal{R}$ on the disjoint union $S_1 \uplus S_2$ is a* weak probabilistic bisimulation *if, for each pair of states $s, t \in S_1 \uplus S_2$ such that $s \mathrel{\mathcal{R}} t$, if $s \xrightarrow{a} \mu_s$ for some probability distribution μ_s, then there exists μ_t such that $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ and $\mu_s \mathrel{\mathcal{L}(\mathcal{R})} \mu_t$.*

We say that $\mathcal{A}_1$ and $\mathcal{A}_2$ are weak probabilistic bisimilar if there exists a weak probabilistic bisimulation $\mathcal{R}$ on $S_1 \uplus S_2$ such that $\bar{s}_1 \mathrel{\mathcal{R}} \bar{s}_2$ and we say that two states s_1 and s_2 are weak probabilistic bisimilar if $s_1 \mathrel{\mathcal{R}} s_2$. We denote the coarsest weak probabilistic bisimulation, called weak probabilistic bisimilarity, by $\approx$.

Example (cont'd). Consider any instance $WCC(n, r, p)$ and the ideal communication channel ICC. It is quite easy to verify that $ICC \approx WCC(n, r, p)$ for each $n \in \mathbb{N}$, $r \in \mathbb{R}^{\geq 0}$, and $p \in (0, 1]$, where the relation $\mathcal{R}$ justifying $ICC \approx WCC(n, r, p)$ has for each $m \in Msg$ one class containing all h_i^m states and another class containing start states. This means, by transitivity of $\approx$, that $WCC(n,r,p) \approx WCC(n',r',p')$ for each possible value of $n, n' \in \mathbb{N}$, $r, r' \in \mathbb{R}^{\geq 0}$, and $p, p' \in (0,1]$.

We say that there is a *hyper-transition* from $\rho \in \mathrm{Disc}(S)$ to $\mu \in \mathrm{Disc}(S)$ labelled by $a \in \Sigma$, denoted by $\rho \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$, if there exists a family of weak combined transitions $\{s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_s\}_{s \in \mathrm{Supp}(\rho)}$ such that $\mu = \sum_{s \in \mathrm{Supp}(\rho)} \rho(s) \cdot \mu_s$, i.e., for each $t \in S$, $\mu(t) = \sum_{s \in \mathrm{Supp}(\rho)} \rho(s) \cdot \mu_s(t)$. Given $s \stackrel{a}{\longrightarrow} \rho$ and $\rho \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \mu$, we denote by $s \stackrel{a}{\longrightarrow} \rho \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \mu$ the weak combined transition $s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$ obtained by concatenating $s \stackrel{a}{\longrightarrow} \rho$ and $\rho \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \mu$ (cf. [16, Prop. 3.6]).

3 Weak Transitions as LP Problems Revisited

This section revisits and extends the idea underlying the equivalence of weak transitions and linear programming problems, as developed in [11]. With some inspiration from network flow problems, we were able to see a transition $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ of the *PA* $\mathcal{A}$ as a *flow* where the initial probability mass δ_t flows and splits along internal transitions (and exactly one transition with label a for each stream provided $a \neq \tau$) according to the transition target distributions and the scheduler resolution of the nondeterminism.

The LP problem $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{E})\ \mu$, proposed in [11] to verify the existence of the weak combined transition $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ such that $\mu\ \mathcal{L}(\mathcal{E})\ \mu_t$, assumes that $\mathcal{E}$ is an equivalence relation on S; we can extend it to any relation $\mathcal{R} \subseteq S \times S$ as follows: checking that there exists μ_t such that $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ and $\mu\ \mathcal{L}(\mathcal{R})\ \mu_t$ is equivalent, by properties of $\mathcal{L}(\,\cdot\,)$, to find μ_t and μ'_t such that $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$, $\mu_t\ \mathcal{L}(\mathcal{I})\ \mu'_t$, and $\mu\ \mathcal{L}(\mathcal{R})\ \mu'_t$, where $\mathcal{I}$ is the identity relation on S. Since verifying $\mu\ \mathcal{L}(\mathcal{R})\ \mu'_t$ is itself equivalent [2, Lemma 5.1] to solve a maximum flow problem, such flow problem can be merged with the $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{I})\ \mu'_t$ LP problem, thereby abstracting the actual distribution μ'_t, so as to extend it to a binary relation $\mathcal{R}$, as we formalise in the sequel.

For a *PA* $\mathcal{A} = (S, \bar{s}, \Sigma, D)$ and $\mathcal{R} \subseteq S \times S$, for $a \in \mathtt{E}$, the network $G(t, a, \mu, \mathcal{R}) = (V, E)$ has the set of vertices $V = \{\vartriangle, \blacktriangledown\} \cup S \cup S^{tr} \cup S_a \cup S_a^{tr} \cup S_{\mathcal{R}}$ where $S^{tr} = \{\, v^{tr} \mid tr = v \stackrel{b}{\longrightarrow} \rho \in D, b \in \{a, \tau\}\,\}$, $S_a = \{\, v_a \mid v \in S\,\}$, $S_a^{tr} = \{\, v_a^{tr} \mid v^{tr} \in S^{tr}\,\}$, and $S_{\mathcal{R}} = \{\, s_{\mathcal{R}} \mid s \in S\,\}$ and the set of arcs $E = \{(\vartriangle, t)\} \cup \{\, (v_a, u_{\mathcal{R}}), (u_{\mathcal{R}}, \blacktriangledown) \mid u, v \in S, v\ \mathcal{R}\ u\,\} \cup \{\, (v, v^{tr}), (v^{tr}, v'), (v_a, v_a^{tr}), (v_a^{tr}, v'_a) \mid tr = v \stackrel{\tau}{\longrightarrow} \rho \in D, v' \in \mathrm{Supp}(\rho)\,\} \cup \{\, (v, v_a^{tr}), (v_a^{tr}, v'_a) \mid tr = v \stackrel{a}{\longrightarrow} \rho \in D, v' \in \mathrm{Supp}(\rho)\,\}$. When $a \in \mathtt{H}$, the definition is similar: $V = \{\vartriangle, \blacktriangledown\} \cup S \cup S^{tr} \cup S_{\mathcal{R}}$ and $E = \{(\vartriangle, t)\} \cup \{\, (v, u_{\mathcal{R}}), (u_{\mathcal{R}}, \blacktriangledown) \mid u, v \in S, v\ \mathcal{R}\ u\,\} \cup \{\, (v, v^{tr}), (v^{tr}, v') \mid tr = v \stackrel{\tau}{\longrightarrow} \rho \in D, v' \in \mathrm{Supp}(\rho)\,\}$.

As in [11], this network $G(t, a, \mu, \mathcal{R})$ and the associated maximum flow problem can not be used directly to encode a weak combined transition since it is not possible to force the flow to split proportional to the transition probability distributions. Instead an ordinary LP problem can be derived from the network, which is enriched with additional constraints called *balancing factors*. A balancing factor models a probabilistic choice and ensures a balance between flows that leave a vertex so as to respect the probability values in a probabilistic choice, i.e., when leaving a vertex $v \in S^{tr} \cup S_a^{tr}$.

Definition 2 (cf. [11, Def. 6]). *Given a PA $\mathcal{A}$, $\mathcal{R} \subseteq S \times S$, $\mu \in \mathrm{Disc}(S)$, and $t \in S$, for $a \in \mathtt{E}$ we define the $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ LP problem associated to the network graph*

$(V, E) = G(t, a, \mu, \mathcal{R})$ *as follows:*

$$
\begin{array}{ll}
\max \sum_{(x,y) \in E} -f_{x,y} & \\
\textit{under constraints} & \\
f_{u,v} \geq 0 & \textit{for each } (u, v) \in E \\
f_{\vartriangle,t} = 1 & \\
f_{v_{\mathcal{R}},\blacktriangledown} = \mu(v) & \textit{for each } v \in S_{\mathcal{R}} \\
\sum_{u \in \{\, x \mid (x,v) \in E \,\}} f_{u,v} - \sum_{u \in \{\, y \mid (v,y) \in E \,\}} f_{v,u} = 0 & \textit{for each } v \in V \setminus \{\vartriangle, \blacktriangledown\} \\
f_{v^{tr},v'} - \rho(v') f_{v,v^{tr}} = 0 & \textit{for each } tr = v \xrightarrow{\tau} \rho \in D \textit{ and } v' \in \mathrm{Supp}(\rho) \\
f_{v_a^{tr},v'_a} - \rho(v') f_{v_a,v_a^{tr}} = 0 & \textit{for each } tr = v \xrightarrow{\tau} \rho \in D \textit{ and } v' \in \mathrm{Supp}(\rho) \\
f_{v_a^{tr},v'_a} - \rho(v') f_{v,v_a^{tr}} = 0 & \textit{for each } tr = v \xrightarrow{a} \rho \in D \textit{ and } v' \in \mathrm{Supp}(\rho)
\end{array}
$$

When $a \in \mathtt{H}$, the LP problem $t \overset{\tau}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ associated to $G(t, \tau, \mu, \mathcal{R})$ is defined as above without the last two groups of constraints.

The objective function has no impact on the equivalence of $t \overset{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ and a weak combined transition, since any feasible solution is enough to establish the transition (cf. [11, Thm. 8]). This means that we can use $\min \sum_{(x,y) \in E} f_{x,y}$ as objective function, i.e., a weak transition can also be seen as a minimum cost flow problem plus balancing constraints, so we will in the sequel explore how to use the objective function to compute and minimise the cost of performing a weak combined transition.

4 Cost Probabilistic Automata

As said, in this paper we consider as *cost* any kind of quantity associated with the transitions of the automaton $\mathcal{A}$ that we aim to minimise. We model the cost of the transitions by a function c that assigns to each transition a non-negative real value.

Definition 3. *A* cost probabilistic automaton *(CPA) is a pair* $(\mathcal{A}, c)$ *where* $\mathcal{A}$ *is a probabilistic automaton and* c*, the* transition cost function*, is a total function* $c \colon D \to \mathbb{R}^{\geq 0}$.

4.1 Weak Combined Transition Cost

There are several ways of extending the cost from a single transition to a sequence of transitions, and hence to a weak combined transition. One possibility is to consider the weighted sum of the costs of all involved finite execution fragments. This approach matches the standard interpretation in the operations research literature for expected reward criteria [13].

Definition 4. *Given an MDP* $\mathcal{M} = (S, \iota, \Sigma, P, r)$*, a finite execution fragment* $\alpha = s_1 a_1 \dots s_n a_n s_{n+1} \in \mathit{frags}^*(\mathcal{M})$*, a policy* π*, and the final state reward* $r_s \colon S \to \mathbb{R}$*, let* $\alpha{\downharpoonright}i = s_1 a_1 \dots a_{i-1} s_i$ *be the* i*-prefix of* α*,* $r(\alpha) = \sum_{i=1}^{n} r(s_i, a_i) + r_s(s_{n+1})$*, and* $P^{\pi}(\alpha) = \iota(s_1) \cdot \prod_{i=1}^{n} \pi(\alpha{\downharpoonright}i)(a_i) \cdot P(s_i, a_i)(s_{i+1})$*. Then the* expected total reward with horizon N *is defined as* $\mathbb{E}_N^{\pi} = \sum_{\alpha \in \{\, \alpha \in \mathit{frags}^*(\mathcal{M}) \mid |\alpha| = N \,\}} r(\alpha) \cdot P^{\pi}(\alpha)$.

Since probabilistic automata are a conservative extension of *MDP*, we extend this notion to weak transition costs by taking into account the resolution of the nondeterminism as induced by a given scheduler.

Definition 5. *Given a CPA* $(\mathcal{A}, c)$*, a state* s*, an action* a*, a probability distribution* μ*, and a scheduler* σ *inducing the weak combined transition* $s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$*, we define the cost* $c_\sigma(s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu)$ *of the weak combined transition* $s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$ *as*

$$c_\sigma(s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu) = \sum_{\alpha \in \mathit{frags}^*(\mathcal{A})} c_\sigma(\alpha) \cdot \mu_{\sigma,s}(\alpha)$$

where $c_\sigma(\alpha) = c_\sigma(\alpha') + \sum_{tr \in D(a)} c(tr) \cdot \widehat{\sigma}(\alpha', t, a, tr)$ *if* $\alpha = \alpha' a t$*,* 0 *otherwise, and where* $\widehat{\sigma} \colon \mathit{frags}^*(\mathcal{A}) \times S \times \Sigma \times D \to \mathbb{R}^{\geq 0}$ *is defined as:*

$$\widehat{\sigma}(\alpha, t, a, tr) = \begin{cases} \dfrac{\sigma(\alpha)(tr) \cdot \mu_{tr}(t)}{\sum_{tr \in D(a)} \sigma(\alpha)(tr) \cdot \mu_{tr}(t)} & \text{if } \sum_{tr \in D(a)} \sigma(\alpha)(tr) \cdot \mu_{tr}(t) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

When the scheduler σ is clear from the context, we just write $c(s \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu)$.

When we restrict Def. 5 to *MDP*s, it coincides with Def. 4:

Proposition 1. *Given an MDP* $\mathcal{M}$*, a policy* π*, and a final state reward* $r \colon S \to \mathbb{R}$ *such that for each* $s \in S$*,* $r(s) = 0$*, it holds that for each* $N \in \mathbb{N}$*,*

$$\mathbb{E}_N^\pi = \sum_{s \in S} \iota(s) \cdot c(s \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \mu)$$

where for each $tr = (s, a, P(s, a)) \in D$ *and* $\alpha \in \mathit{frags}^*(\mathcal{M})$*, we define* $c(tr) = r(s, a)$*,* $\sigma(\alpha)(tr) = \pi(\alpha)(a)$ *if* $|\alpha| < N$*,* 0 *otherwise, and* $s \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \mu$ *is the weak combined transition induced by the scheduler* σ *when all actions are considered as internal.*

Example (cont'd). Consider the *CPA* $(WCC(n, r, p), c)$ where c assigns cost $\mathbf{r}$ to each transition labelled by the internal action t_r; the weak combined transition $h_0^m \stackrel{t_r}{\Longrightarrow}_{\mathrm{C}} \delta_{h_n^m}$ can be seen as the sequence of transitions $h_i^m \stackrel{t_r}{\Longrightarrow}_{\mathrm{C}} \delta_{h_{i+1}^m}$ for $0 \leq i < n$. It is routine to check that each $h_i^m \stackrel{t_r}{\Longrightarrow}_{\mathrm{C}} \delta_{h_{i+1}^m}$ is induced by the scheduler σ_i such that $\sigma_i(\alpha) = \delta_{tr_i^m}$ if $\mathit{last}(\alpha) = h_i^m$, $\delta_\perp$ otherwise, where $tr_i^m = h_i^m \xrightarrow{t_r} \{(h_{i+1}^m, p), (h_i^m, 1-p)\}$. Now, consider the finite execution fragment $\alpha = (h_i^m t_r)^{n+1} h_{i+1}^m$: according to Def. 5, it has cost $c_{\sigma_i}(\alpha) = (n+1) \cdot \mathbf{r}$. The probability $\mu_{\sigma_i, h_i^m}(\alpha)$ of α is $(1-p)^n \cdot p$ while the probability of each $\alpha' \in \mathit{frags}^*(WCC(n, r, p)) \setminus \{\, (h_i^m t_r)^{n+1} h_{i+1}^m \mid n \in \mathbb{N} \,\}$ is 0, thus the cost of the transition $h_i^m \stackrel{t_r}{\Longrightarrow}_{\mathrm{C}} \delta_{h_{i+1}^m}$ as induced by σ_i is $c_{\sigma_i}(h_i^m \stackrel{t_r}{\Longrightarrow}_{\mathrm{C}} \delta_{h_{i+1}^m}) = \sum_{n \in \mathbb{N}} (n+1) \cdot \mathbf{r} \cdot (1-p)^n \cdot p = \mathbf{r} \cdot p \cdot \sum_{n \in \mathbb{N}} (n+1) \cdot (1-p)^n = \frac{\mathbf{r} \cdot p}{1-p} \cdot \sum_{n \in \mathbb{N}} (n+1) \cdot (1-p)^{n+1} = \frac{\mathbf{r} \cdot p}{1-p} \cdot \frac{1-p}{(1-(1-p))^2} = \frac{\mathbf{r}}{p}$, hence $h_0^m \stackrel{t_r}{\Longrightarrow}_{\mathrm{C}} \delta_{h_n^m}$ has cost $n \cdot \frac{\mathbf{r}}{p}$.

By using an equivalent definition of weak transition cost, the transition costs can be encoded in the LP problem as coefficients of the objective function.

Definition 6. *Given a CPA* $(\mathcal{A}, c)$*, a binary relation* $\mathcal{R}$ *on* S*, a probability distribution* $\mu \in \mathrm{Disc}(S)$*, and a state* $t \in S$*, for action* $a \neq \tau$ *we define the min-cost LP problem* $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *associated to the network* $G(t, a, \mu, \mathcal{R})$ *as follows.*

$$
\begin{array}{ll}
\min \sum_{(x,y)\in E} c_f((x,y)) \cdot f_{x,y} & \\
\textit{under constraints} & \\
f_{u,v} \geq 0 & \textit{for each } (u,v) \in E \\
f_{\vartriangle,t} = 1 & \\
f_{v_{\mathcal{R}},\blacktriangledown} = \mu(v) & \textit{for each } v \in S_{\mathcal{R}} \\
\sum_{u\in\{\,x \mid (x,v)\in E\,\}} f_{u,v} - \sum_{u\in\{\,y \mid (v,y)\in E\,\}} f_{v,u} = 0 & \textit{for each } v \in V \setminus \{\vartriangle,\blacktriangledown\} \\
f_{v^{tr},v'} - \rho(v') f_{v,v^{tr}} = 0 & \textit{for each } tr = v \xrightarrow{\tau} \rho \in D \textit{ and } v' \in \mathrm{Supp}(\rho) \\
f_{v_a^{tr},v'_a} - \rho(v') f_{v_a,v_a^{tr}} = 0 & \textit{for each } tr = v \xrightarrow{\tau} \rho \in D \textit{ and } v' \in \mathrm{Supp}(\rho) \\
f_{v_a^{tr},v'_a} - \rho(v') f_{v,v_a^{tr}} = 0 & \textit{for each } tr = v \xrightarrow{a} \rho \in D \textit{ and } v' \in \mathrm{Supp}(\rho)
\end{array}
$$

where $c_f \colon E \to \mathbb{R}^{\geq 0}$ *is a total function defined as follows:*

$$
c_f((x,y)) = \begin{cases} c(tr) & \textit{if } tr = v \xrightarrow{\tau} \rho,\ x = v,\ y = v^{tr}, \\ c(tr) & \textit{if } tr = v \xrightarrow{\tau} \rho,\ x = v_a,\ y = v_a^{tr}, \\ c(tr) & \textit{if } tr = v \xrightarrow{a} \rho,\ x = v,\ y = v_a^{tr}, \\ 0 & \textit{otherwise.} \end{cases}
$$

If $\min_c t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *has an optimal solution* f^o*, then we denote by* $\mathfrak{C}$ *the minimum cost* $\mathfrak{C} = \sum_{(x,y)\in E} c_f((x,y)) \cdot f^o_{x,y}$*.*

When the action a is τ, the min-cost LP problem $\min_c t \xRightarrow{\tau}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ associated to the network $G(t,\tau,\mu,\mathcal{R})$ is defined as above without the last two groups of constraints.

A first straightforward result is that $\min_c t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ is feasible if and only if $t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ is feasible, since the only difference between the two problems is the objective function that does not affect the feasibility of an LP problem:

Proposition 2. *Given a CPA* $(\mathcal{A}, c)$*,* $\mathcal{R} \subseteq S \times S$*,* $a \in \Sigma$*,* $\mu \in \mathrm{Disc}(S)$*, and* $t \in S$*, the minimisation LP problem* $\min_c t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *has a feasible solution* f^* *if and only if* f^* *is a feasible solution of the LP problem* $t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$*.*

Similarly, as generating and checking the existence of a valid solution of the LP problem $t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ is polynomial in $N = \max\{|S|, |D|\}$ (cf. [11, Thm. 7]), the same holds for $\min_c t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$:

Corollary 1. *Given a CPA* $(\mathcal{A}, c)$*,* $\mathcal{R} \subseteq S \times S$*,* $a \in \Sigma$*,* $\mu \in \mathrm{Disc}(S)$*, and* $t \in S$*, generating and checking the existence of a valid solution of the minimisation LP problem* $\min_c t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *is polynomial in* $N = \max\{|S|, |D|\}$*.*

Since $t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ is feasible if and only if there exists a scheduler σ that induces $t \xRightarrow{a}_{\mathrm{C}} \mu_t$ such that $\mu\ \mathcal{L}(\mathcal{R})\ \mu_t$, we may expect a similar result regarding costs, that is, $\min_c t \xRightarrow{a}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ is feasible with optimal value $\mathfrak{C}$ if and only if there exists a scheduler σ that induces $t \xRightarrow{a}_{\mathrm{C}} \mu_t$ such that $\mu\ \mathcal{L}(\mathcal{R})\ \mu_t$ and $c(t \xRightarrow{a}_{\mathrm{C}} \mu_t) = \mathfrak{C}$. But in general it is not possible to obtain such a result since there can be equivalent ways to resolve nondeterminism that induce different costs, thus we can not talk about *the cost* of a weak combined transition, but of the cost of the weak combined transition *as induced by the scheduler* σ. For instance, consider an automaton $\mathcal{A}$ whose transitions are

$tr_1 = \bar{s} \xrightarrow{a} \delta_t$, $tr_2 = \bar{s} \xrightarrow{\tau} \delta_v$, and $tr_3 = v \xrightarrow{a} \delta_t$, each one with cost 1. It is straightforward to check that the scheduler σ_1 such that $\sigma_1(\bar{s}) = \delta_{tr_1}$ and $\sigma_1(\alpha) = \delta_\perp$ for each finite execution fragment $\alpha \neq \bar{s}$ induces the weak combined transition $\bar{s} \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \delta_t$ whose cost is 1; the same transition is induced also by the scheduler σ_2 defined as $\sigma_2(\bar{s}) = \delta_{tr_2}$, $\sigma_2(\bar{s}\tau v) = \delta_{tr_3}$, and $\sigma_2(\alpha) = \delta_\perp$ for each other finite execution fragment α. However the cost as induced by σ_2 is $c_{\sigma_2}(\bar{s} \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \delta_t) = 2 \neq 1 = c_{\sigma_1}(\bar{s} \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \delta_t)$; it is easy to show that $1 \leq c_\sigma(\bar{s} \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \delta_t) \leq 2$ for each scheduler σ inducing $\bar{s} \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \delta_t$. Note that there are uncountably many such schedulers, each one corresponding to a different resolution of the choice between $tr_1 = \bar{s} \xrightarrow{a} \delta_t$ and $tr_2 = \bar{s} \xrightarrow{\tau} \delta_v$: in general, we can denote such choice as the distribution $\{(tr_1, p), (tr_2, 1 - p)\}$ where $p \in [0, 1]$.

The cost given by a scheduler and the value of the objective function of the corresponding LP problem are however related:

Theorem 1. *Given a CPA* $(\mathcal{A}, c)$*,* $\mathcal{R} \subseteq S \times S$*,* $a \in \Sigma$*,* $\mu \in \mathrm{Disc}(S)$*, and* $t \in S$*, consider the* $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *LP problem. The following implications hold:*

1. *If there exists a scheduler* σ *for* $\mathcal{A}$ *that induces* $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ *such that* $\mu\ \mathcal{L}(\mathcal{R})\ \mu_t$*, then* $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *has an optimal solution* f^o *such that* $\mathfrak{C} \leq c(t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t)$*.*
2. *If* $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *has an optimal solution* f^o*, then there exists a scheduler* σ *for* $\mathcal{A}$ *that induces* $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ *such that* $\mu\ \mathcal{L}(\mathcal{R})\ \mu_t$ *and* $c(t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t) = \mathfrak{C}$*.*

As immediate corollaries we have that the cost given by the optimal solution of the $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ LP problem corresponds to the minimum cost induced by any scheduler inducing $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ and that finding such minimum is polynomial.

Corollary 2. *Given a CPA* $(\mathcal{A}, c)$*,* $\mathcal{R} \subseteq S \times S$*,* $a \in \Sigma$*,* $\mu \in \mathrm{Disc}(S)$*, and* $t \in S$ *such that there exists* $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ *with* $\mu\ \mathcal{L}(\mathcal{R})\ \mu_t$*, the LP problem* $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ *has minimum cost* $\mathfrak{C} = \min\{\, c_\sigma(t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t) \mid \sigma \text{ induces } t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t \text{ such that } \mu\ \mathcal{L}(\mathcal{R})\ \mu_t \,\}$*.*

Corollary 3. *Given a CPA* $(\mathcal{A}, c)$*,* $\mathcal{R} \subseteq S \times S$*,* $a \in \Sigma$*,* $\mu \in \mathrm{Disc}(S)$*, and* $t \in S$*, finding* $\min\{\, c_\sigma(t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t) \mid \sigma \text{ induces } t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t \text{ such that } \mu\ \mathcal{L}(\mathcal{R})\ \mu_t \,\}$ *is polynomial in* $N = \max\{|S|, |D|\}$*.*

Extending the above results to hyper-transitions of the *CPA* $(\mathcal{A}, c)$ is straightforward, since we can consider each hyper-transition $\rho \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$ as the weak combined transition $h \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu$ in the *CPA* $(\mathcal{A}', c')$ that is $(\mathcal{A}, c)$ enriched with the fresh state h and the transition $h \xrightarrow{\tau} \rho$ whose cost is set to 0.

4.2 Cost Preserving Bisimulations

We now discuss options for weak bisimulations on *CPA*, so as to ignore internal computations as long as those do not change the visible behaviour of the system. Since a *CPA* is an ordinary *PA* enriched with a cost function, one might consider a naive lifting of *PA* weak probabilistic bisimulation, where two *CPA* are weak probabilistic bisimilar if the underlying *PA* are. However this definition obviously falls too short, since it may relate states with different cost behaviours. For this reason, following [9], we define a restricted notion of weak probabilistic bisimulation where each transition $s \xrightarrow{a} \mu_s$ of the challenging state s has to be matched by the defender state t by enabling a weak combined transition $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$ such that $\mu_s \; \mathcal{L}(\mathcal{R}) \; \mu_t$ as in ordinary weak probabilistic bisimulation, and, in addition, the costs of challenging and defending transitions must agree.

Definition 7. *Given two CPAs $(\mathcal{A}_1, c_1)$ and $(\mathcal{A}_2, c_2)$, an equivalence relation $\mathcal{R}$ on the disjoint union $S_1 \uplus S_2$ is a* weak probabilistic cost-preserving bisimulation *if for each pair of states $s, t \in S_1 \uplus S_2$ such that $s \; \mathcal{R} \; t$, if $s \xrightarrow{a} \mu_s$, then there exists μ_t such that $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t$, $\mu_s \; \mathcal{L}(\mathcal{R}) \; \mu_t$, and $c_d(t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_t) = c_c(s \xrightarrow{a} \mu_s)$ where c_d and c_c are the cost functions of the defender and the challenger CPA, respectively.*

Two CPAs $(\mathcal{A}_1, c_1)$ and $(\mathcal{A}_2, c_2)$ are weak probabilistic cost-preserving bisimilar if there exists a weak probabilistic cost-preserving bisimulation $\mathcal{R}$ on $S_1 \uplus S_2$ such that $\bar{s}_1 \; \mathcal{R} \; \bar{s}_2$. We denote the coarsest weak probabilistic cost-preserving bisimulation by $\approx_c$, called weak probabilistic cost-preserving bisimilarity.

By using this definition of bisimulation, we have that states enabling transitions with different cost are no more bisimilar, since they do not respect cost constraints.

4.3 Cost Bounding Bisimulations

The definition of weak probabilistic cost-preserving bisimulation allows us to relate *CPA*s that have the same behaviour and the same cost. Since our aim is to minimise the cost while preserving the behaviour of a *CPA* $(\mathcal{A}, c)$, we will now relax the cost equality by requiring that the cost of the defender matching transition is at most the cost of the challenger transition. Despite the simplicity of this idea, the formal definition is quite involved since we have to consider properly the cost of internal transitions.

To shed some light on this, consider an automaton $\mathcal{A}_1$ performing three internal steps $\bar{s}_1 \xrightarrow{\tau} \delta_{t_1}$, $t_1 \xrightarrow{\tau} \delta_{u_1}$, and $u_1 \xrightarrow{\tau} \delta_{v_1}$ where each step has cost 5 followed by an external step $v_1 \xrightarrow{a} \delta_{x_1}$ with cost 1 and an automaton $\mathcal{A}_2$ that performs four steps $\bar{s}_2 \xrightarrow{\tau} \delta_{t_2}$, $t_2 \xrightarrow{\tau} \delta_{u_2}$, $u_2 \xrightarrow{\tau} \delta_{v_2}$, and $v_2 \xrightarrow{\tau} \delta_{w_2}$ each with cost 3 followed by an external step $w_2 \xrightarrow{a} \delta_{x_2}$ with cost 1. An external observer is able to recognise that the behaviour of $\mathcal{A}_1$ is more expensive than the one of $\mathcal{A}_2$ since the overall cost is 16 for the former, 13 for the latter. However, from a state-based bisimulation point of view, $\mathcal{A}_2$ is not always cheaper than $\mathcal{A}_1$: let $\{\{\bar{s}_1, \bar{s}_2\}, \{t_1, t_2\}, \{u_1, u_2\}, \{v_1, v_2, w_2\}, \{x_1, x_2\}\}$ be the equivalence classes of $\mathcal{R}$; it is easy to verify that $\mathcal{R}$ is a weak probabilistic bisimulation between $\mathcal{A}_1$ and $\mathcal{A}_2$: when $\mathcal{A}_1$ performs $\bar{s}_1 \xrightarrow{\tau} \delta_{t_1}$ with cost 5, $\mathcal{A}_2$ replies with $\bar{s}_2 \xrightarrow{\tau} \delta_{t_2}$ with cost $3 \leq 5$ and $t_1 \; \mathcal{R} \; t_2$. Note that $\mathcal{A}_2$ can not perform the subsequent transition $t_2 \xrightarrow{\tau} \delta_{u_2}$ since in this case the overall cost is $6 \not\leq 5$. The same happens for transitions $t_1 \xrightarrow{\tau} \delta_{u_1}$ and $u_1 \xrightarrow{\tau} \delta_{v_1}$ that are matched by $t_2 \xrightarrow{\tau} \delta_{u_2}$ and $u_2 \xrightarrow{\tau} \delta_{v_2}$, respectively. Since $\mathcal{A}_1$ now performs $v_1 \xrightarrow{a} \delta_{x_1}$ with cost 1, v_2 is not able to match this transition with a cost at most 1: in order to match the transition, $\mathcal{A}_2$ has to perform both transitions $v_2 \xrightarrow{\tau} \delta_{w_2}$ and $w_2 \xrightarrow{a} \delta_{x_2}$ whose cost is $4 \not\leq 1$.

These considerations indicate that internal challenger transitions should not be considered separately but as a whole, so in order to abstract away from costs of single challenger internal transitions while preserving the overall cost, we consider for the challenger the cost of reaching the border states, i.e., states where the automaton performs an external action or exhibits a different behaviour by changing the current class as induced by the weak bisimulation relation.

Definition 8. *Given a PA $\mathcal{A}$ and an equivalence relation $\mathcal{R}$ over S, we say that a state s is a* border state *if there exists $s \xrightarrow{a} \mu \in D$ such that either $\mu([s]_{\mathcal{R}}) < 1$ or $a \in \mathtt{E}$.*

We denote the set of all border states with respect to $\mathcal{R}$ by $\mathcal{B}(\mathcal{R})$.

Definition 9. *Let $(\mathcal{A}_1, c_1)$ and $(\mathcal{A}_2, c_2)$ be two CPAs. Let $\mathcal{W}$ be an equivalence relation on the disjoint union $S_1 \uplus S_2$ and $\mathcal{C} \subseteq \mathcal{W} \cap S_2 \times S_1$ such that for each $s_2 \in S_2$ there exists $s_1 \in S_1$ such that $s_2 \; \mathcal{C} \; s_1$. Then we say that $(\mathcal{W}, \mathcal{C})$ is a* minor cost weak probabilistic bisimulation *from $(\mathcal{A}_1, c_1)$ to $(\mathcal{A}_2, c_2)$ if $\mathcal{W}$ is a weak probabilistic bisimulation for $\mathcal{A}_1$ and $\mathcal{A}_2$ and for each $s_2 \xrightarrow{a} \mu_2 \in D_2$ and each $s_1 \in S_1$ such that $s_2 \; \mathcal{C} \; s_1$,*

1. *if there exists $\rho_2 \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_2)$ such that $\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho_2$, then there exists $\rho_1 \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_1)$ such that (a) $s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \rho_1$, (b) $\rho_2 \; \mathcal{L}(\mathcal{C}) \; \rho_1$, (c) $c_1(s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \rho_1) \leq c_2(s_2 \xrightarrow{a} \mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho_2)$, and (d) $\min\{ c_2(\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho) \mid \rho \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_2) \} = c_2(\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho_2)$; or*
2. *if there does not exist $\rho_2 \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_2)$ such that $\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho_2$, then there exists $\mu_1 \in \mathrm{Disc}(S_1)$ such that $s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_1$, $\mu_2 \; \mathcal{L}(\mathcal{C}) \; \mu_1$, and $c_1(s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_1) \leq c_2(s_2 \xrightarrow{a} \mu_2)$.*

We say that $(\mathcal{A}_1, c_1)$ is minor cost weak probabilistic bisimilar to $(\mathcal{A}_2, c_2)$ if there exists a minor cost weak probabilistic bisimulation $(\mathcal{W}, \mathcal{C})$ such that $\bar{s}_2 \; \mathcal{C} \; \bar{s}_1$. We denote the coarsest minor cost weak probabilistic bisimulation from $(\mathcal{A}_1, c_1)$ to $(\mathcal{A}_2, c_2)$ by $(\mathcal{A}_1, c_1) \lessapprox_c (\mathcal{A}_2, c_2)$ and we say that $(\mathcal{A}_1, c_1)$ is in minor cost weak probabilistic bisimilarity with $(\mathcal{A}_2, c_2)$.

4.4 The Cost of the Wireless Communication Channel

We now apply the minor cost weak probabilistic bisimulation to the reliable wireless communication channel introduced in Sec. 2 and depicted in Fig. 1. As cost, we consider the function c that assigns cost 1 to transitions labelled by s_m or r_m and cost r^2 to transitions labelled by t_r. We use value 1 to represent a constant power consumption relative to sending/receiving message actions and value r^2 to model the energy, quadratic on the transmission radius, required to transmit a message via wireless.

As a concrete example, consider the two instances $\mathcal{A}_{23} = WCC(2, 3, \frac{1}{2})$ and $\mathcal{A}_{32} = WCC(3, 2, \frac{1}{2})$ of the wireless communication channel connecting sender and receiver that are at distance 6. To avoid name collisions, we rename the states h_j^m of $WCC(3, 2, \frac{1}{2})$ to k_j^m for $0 \leq j \leq 3$. It is easy to verify that the equivalence relation $\mathcal{W}$ whose classes are $\{\bar{s}_{23}, \bar{s}_{32}\}$ and $\{ h_i^m, k_j^m \mid 0 \leq i \leq 2, 0 \leq j \leq 3 \}$ for each $m \in Msg$ justifies $\mathcal{A}_{23} \approx \mathcal{A}_{32}$, so consider the two CPAs $(\mathcal{A}_{23}, c)$ and $(\mathcal{A}_{32}, c)$. We suspect that $(\mathcal{A}_{32}, c) \lessapprox_c (\mathcal{A}_{23}, c)$, but not the reverse, since intuitively $(\mathcal{A}_{23}, c)$ has overall cost 26 for sending and receiving a single message while $(\mathcal{A}_{32}, c)$ has overall cost 38. In order to show $(\mathcal{A}_{32}, c) \lessapprox_c (\mathcal{A}_{23}, c)$, we have to find a suitable relation $\mathcal{C}$ that, together with $\mathcal{W}$, satisfies the conditions of Def. 9. A suitable relation is $\mathcal{C} = \{(\bar{s}_{23}, \bar{s}_{32})\} \cup \bigcup_{m \in Msg} \{ (h_i^m, k_3^m) \mid 0 \leq i \leq 2 \}$: consider the pair $(\bar{s}_{23}, \bar{s}_{32})$ and the only available transition $\bar{s}_{23} \xrightarrow{s_m} \delta_{h_0^m}$. Since $\mathcal{B}(\mathcal{W}) = \{ \bar{s}_{23}, \bar{s}_{32}, h_2^m, k_3^m \mid m \in Msg \}$, the only possible $\rho_{23} \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_{23})$ such that $\delta_{h_0^m} \stackrel{\tau}{\Longrightarrow} \rho_{23}$ is $\rho_{23} = \delta_{h_2^m}$. In order to match such transition, $\bar{s}_{32}$ enables the weak transition $\bar{s}_{32} \stackrel{s_m}{\Longrightarrow}_{\mathrm{C}} \delta_{k_3^m}$ that satisfies $\delta_{h_2^m} \; \mathcal{L}(\mathcal{C}) \; \delta_{k_3^m}$. The last condition we have to verify is that $c(\bar{s}_{32} \stackrel{s_m}{\Longrightarrow}_{\mathrm{C}} \delta_{k_3^m}) \leq$

$c(\bar{s}_{23} \xrightarrow{s_m} \delta_{h_0^m} \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \delta_{h_2^m})$; this constraint is satisfied since $c(\bar{s}_{32} \stackrel{s_m}{\Longrightarrow}_{\mathrm{C}} \delta_{k_3^m}) = 25$ while $c(\bar{s}_{23} \xrightarrow{s_m} \delta_{h_0^m} \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \delta_{h_2^m}) = 37$. It is routine to check the remaining pairs of states, thus $(\mathcal{A}_{32}, c) \lessapprox_c (\mathcal{A}_{23}, c)$.

Now, assume $(\mathcal{A}_{23}, c) \lessapprox_c (\mathcal{A}_{32}, c)$: by definition, it must hold that $\bar{s}_{32} \ \mathcal{C} \ \bar{s}_{23}$, so consider the transition $\bar{s}_{32} \xrightarrow{s_m} \delta_{k_0^m}$. For sure k_3^m and h_2^m are border states, as well as $\bar{s}_{32}$ and $\bar{s}_{23}$. Moreover, $\bar{s}_{32}$ and $\bar{s}_{23}$ can not be related by $\mathcal{W}$ to any other state as they are the only states performing s_m. Suppose that these are the only border states; this implies that $\bar{s}_{32} \xrightarrow{s_m} \delta_{k_0^m}$ has to be extended to $\bar{s}_{32} \xrightarrow{s_m} \delta_{k_0^m} \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \delta_{k_3^m}$ whose cost is 25. The only possibility for $\bar{s}_{23}$ to match such transition while respecting the cost constraint is to perform the weak combined transition $\bar{s}_{23} \stackrel{s_m}{\Longrightarrow}_{\mathrm{C}} \delta_{h_i^m}$ with $i = 0$ or $i = 1$ and $k_3^m \ \mathcal{C} \ h_i^m$. Note that we can not use $\bar{s}_{23} \stackrel{s_m}{\Longrightarrow}_{\mathrm{C}} \delta_{h_2^m}$ since its cost is $37 \nleq 25$. Independently on the chosen i, since $k_3^m \ \mathcal{C} \ h_i^m$ and $k_3^m \xrightarrow{r_m} \delta_{\bar{s}_{32}}$, h_i^m has to perform the weak combined transition $h_i^m \stackrel{r_m}{\Longrightarrow}_{\mathrm{C}} \delta_{\bar{s}_{23}}$ whose cost is $1 + 18 \cdot (2 - i) \nleq 1$, so the condition is not satisfied. By applying the same approach to the case where we consider other states as border states, we can derive a similar failure, thus there does not exist any suitable cost relation $\mathcal{C}$ with $\bar{s}_{32} \ \mathcal{C} \ \bar{s}_{23}$, hence $(\mathcal{A}_{23}, c) \not\lessapprox_c (\mathcal{A}_{32}, c)$.

4.5 Decision Procedure

In order to algorithmically decide whether $(\mathcal{A}_1, c_1) \lessapprox_c (\mathcal{A}_2, c_2)$, we extend the polynomial decision procedure QUOTIENT that establishes whether $\mathcal{A}_1 \approx \mathcal{A}_2$ holds [11], to the MINORCOST algorithm depicted in Fig. 2 that computes $(\mathcal{W}, \mathcal{C})$ justifying $(\mathcal{A}_1, c_1) \lessapprox_c (\mathcal{A}_2, c_2)$: we first compute $\mathcal{W} = \text{QUOTIENT}(\mathcal{A}_1, \mathcal{A}_2)$ and then we consider as candidate cost relation $\mathcal{C} = \mathcal{C}'$ all pairs $s_2 \ \mathcal{W} \ s_1$ with $s_2 \in S_2$ and $s_1 \in S_1$. In the main loop of MINORCOST we repeatedly refine $\mathcal{C}$ by removing all pairs that do not satisfy the conditions of Def. 9: if a check fails, we remove the offending pair (s_2, s_1) from $\mathcal{C}'$.

On termination of the loop, $\mathcal{C}$ contains only pairs satisfying Def. 9, so deciding whether $(\mathcal{A}_1, c_1) \lessapprox_c (\mathcal{A}_2, c_2)$ reduces to check whether $\bar{s}_2 \ \mathcal{C} \ \bar{s}_1$ and whether for each $s_2 \in S_2$ there exists $s_1 \in S_1$ such that $s_2 \ \mathcal{C} \ s_1$.

Given two *CPA*s $(\mathcal{A}_1, c_1)$ and $(\mathcal{A}_2, c_2)$, let $N = \max\{|S_1 \uplus S_2|, |D_1 \uplus D_2|\}$. Computing $\mathcal{W} = \text{QUOTIENT}(\mathcal{A}_1, \mathcal{A}_2)$ is polynomial in N (cf. [11, Thm. 11]), say $P(N)$; in the worst case, that occurs when we remove all pairs from $\mathcal{C}$, the main loop of MINORCOST is performed at most N^2 times; according to Thm. 1 and its corollaries, finding $\rho_2 \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_2)$ such that $\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho_2$ and $c_2(\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho_2) = \min\{\, c_2(\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho) \mid \rho \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_2) \,\}$ is polynomial in N, say $R(N)$, by solving the LP problem $\min_c \mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{B}) \ \delta_{b_2}$ where $b_2 \in \mathcal{B}(\mathcal{W}) \cap S_2$ and $\mathcal{B}$ is the reflexive, symmetric, and transitive closure of $\mathcal{B}(\mathcal{W})$. Similarly, $R(N)$ is also the complexity of either finding $\rho_1 \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_1)$ such that $s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \rho_1$, $\rho_2 \ \mathcal{L}(\mathcal{C}) \ \rho_1$, and $c_1(s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \rho_1) \leq c_2(s_2 \xrightarrow{a} \mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \rho_2)$, or finding $\mu_1 \in \mathrm{Disc}(S_1)$ such that $s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_1$, $\mu_2 \ \mathcal{L}(\mathcal{C}) \ \mu_1$, and $c_1(s_1 \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \mu_1) \leq c_2(s_2 \xrightarrow{a} \mu_2)$. This implies that the total complexity of MINORCOST is $P(N) + N^2 \cdot 2R(N)$.

Theorem 2. *Given two CPAs $(\mathcal{A}_1, c_1)$ and $(\mathcal{A}_2, c_2)$, checking $(\mathcal{A}_1, c_1) \lessapprox_c (\mathcal{A}_2, c_2)$ is polynomial in $N = \max\{|S_1 \uplus S_2|, |D_1 \uplus D_2|\}$.*

Regarding weak probabilistic cost-preserving bisimulation, the algorithm is actually simpler, since in order to check for the existence of weak combined transitions with a

MinorCost$((\mathcal{A}_1, c_1), (\mathcal{A}_2, c_2))$

$\mathcal{W} =$ Quotient$(\mathcal{A}_1, \mathcal{A}_2)$
$\mathcal{C}' = \mathcal{W} \cap (S_2 \times S_1)$
repeat
  $\mathcal{C} = \mathcal{C}'$
  for all $(s_2, a, \mu_2) \in D_2$ and $s_1 \in S_1$ s. t. $s_2\ \mathcal{C}'\ s_1$ **do**
    if there exists $\bar{\rho}_2 \in \mathrm{Disc}(\mathcal{B}(\mathcal{W}) \cap S_2)$ such that $\mu_2 \stackrel{\tau}{\Longrightarrow}_{\mathrm{C}} \bar{\rho}_2$ **then**
      if Condition 1 of Def. 9 is not satisfied **then** $\mathcal{C}' = \mathcal{C}' \setminus \{(s_2, s_1)\}$
    else
      if Condition 2 of Def. 9 is not satisfied **then** $\mathcal{C}' = \mathcal{C}' \setminus \{(s_2, s_1)\}$
until $\mathcal{C}' \neq \mathcal{C}$
return $(\mathcal{W}, \mathcal{C})$

Quotient$(\mathcal{A}_1, \mathcal{A}_2)$

$\mathcal{W} = \{S_1 \uplus S_2\}$;
$(\mathcal{C}, a, \mu) =$ FindSplit$(\mathcal{W})$;
while $\mathcal{C} \neq \emptyset$ **do**
  $\mathcal{W} =$ Refine$(\mathcal{W}, (\mathcal{C}, a, \mu))$;
  $(\mathcal{C}, a, \mu) =$ FindSplit$(\mathcal{W})$;
return $\mathcal{W}$

FindSplit$(\mathcal{W})$

for all $(s, a, \mu) \in D = D_1 \uplus D_2$ **do**
  for all $t \in [s]_{\mathcal{W}}$ **do**
    if $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{W})\ \mu$ has no solution **then**
      return $([s]_{\mathcal{W}}, a, \mu)$
return $(\emptyset, \tau, \delta_{\bar{s}})$

Fig. 2. Minor cost weak bisimulation decision procedure

given cost **c**, it is enough to add the new constraint $\sum_{(x,y)\in E} c_f((x,y)) \cdot f_{x,y} = \mathbf{c}$ to the $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ LP problem. This allows us to check in polynomial time whether two *CPA*s are weak probabilistic cost-preserving bisimilar: we compute Quotient on the two *CPA*s where we have replaced in procedure FindSplit the test for feasibility of $t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{W})\ \mu$ with the test for feasibility of $\min_c t \stackrel{a}{\Longrightarrow}_{\mathrm{C}} \diamond \mathcal{L}(\mathcal{R})\ \mu$ extended with the constraint $\sum_{(x,y)\in E} c_f((x,y)) \cdot f_{x,y} = c(s \stackrel{a}{\longrightarrow} \mu_s)$.

Theorem 3. *Given two CPAs $(\mathcal{A}_1, c_1)$ and $(\mathcal{A}_2, c_2)$, checking $(\mathcal{A}_1, c_1) \approx_c (\mathcal{A}_2, c_2)$ is polynomial in $N = \max\{|S_1 \uplus S_2|, |D_1 \uplus D_2|\}$.*

5 Discussion

This section puts our work in the context of related work and also discusses other options to follow.

Givan, Dean and Greig [9] have introduced the idea of strong bisimilarity for *MDP*s with state and transition costs, together with algorithms for minimisation to the quotient model. The minimisation with respect to weak probabilistic bisimulation on *PA* has lately been discussed [7], and it remains to be investigated how the minimisation can be applied for the minor cost approach meaningfully. For the cost-preserving bisimilarity, the adaptations are straightforward, so we can indeed minimise with respect to weak transition costs.

Since *CPA*s are basically *MDP*s with transition costs only, it is interesting to discuss how state costs can be handled. Indeed it is possible to turn state costs to transition costs by moving them on incoming or outgoing transitions. The concrete choice makes a difference, because the labels of incoming and outgoing transitions generally differ. If already transitions costs were present prior to the move, we end up with a second cost structure. Multiple cost structures can indeed also be integrated into our setting rather easily, one just needs to take the minor cost for all structures in the decision problem.

For *MDP*s, multiple reward structures have been investigated [8] in the context of model checking, and our approach naturally combines with that. Chatterjee, Majumdar, and Henzinger [5] investigated them in a setting with discounting. In fact, our polynomial time LP approach can be extended to compute the minimum cost of discounted weak combined transitions, if we can assume a polynomially bounded number of internal steps. Conversely, one can compute an upper bound on discounted but non-polynomially bounded weak combined transitions in polynomial time.

If discounting is integrated into the weak bisimulation definitions we propose, this however induces difficult-to-grasp equalities. This is because sequences of internal transitions of different length are abstracted away by weak bisimilarity, but they would imply different discounts. For similar reasons, our cost model does by itself not talk about traces. As long as internal transitions carry nonzero costs, the definition of the cost of a weak trace is not obvious. Even if two executions fragments have the same trace, i.e., the same sequence of visible actions, different execution fragments usually have different costs when they involve different internal transitions, in particular after the last external action of the trace. Moreover, even if the execution fragment does not involve internal transitions, it can have different costs as resulting by the resolution of probabilistic and nondeterministic choices, the latter performed by the scheduler.

Still, cost-preserving bisimilarity implies equal trace costs, and if $(\mathcal{A}_1, c_1)$ is in minor cost weak probabilistic bisimilarity with $(\mathcal{A}_2, c_2)$, then the trace costs of $(\mathcal{A}_1, c_1)$ are bounded from above by $(\mathcal{A}_2, c_2)$. Trace costs appear central in many cost related formalisms not involving probabilities, such as weighted timed and energy automata [3, 17], though without (internal) actions playing a dedicated role here, so it is worth to investigate trace costs in the *CPA* model as well.

While minor cost weak bisimilarity is implicitly asymmetric, we have still formulated it as an equivalence relation. The case study has demonstrated that this approach is undoubtedly useful. Yet, it seems worthwhile to also take inspiration from simulation and simulation distance approaches [1, 4] in this matter.

6 Concluding Remarks

In this paper we have presented the extension of Probabilistic Automata to Cost Probabilistic Automata and we have proposed two cost related weak probabilistic bisimulations: a cost preserving bisimulation and a cost bounding variation, minor cost weak probabilistic bisimulation, where the defender matches a transition with a cost that is bounded by at most the cost of the challenger.

Moreover we have shown how to compute in polynomial time the minimum cost for each transition, and hence to decide the two relations. Since the *CPA* model encompasses *MDP*s, the results apply readily to these models as well. In the future we plan to investigate how the compositionality properties of weak probabilistic bisimilarity extend from *PA* to *CPA*. With this, we aim to arrive at compositional construction and minimisation techniques that can be rolled out to operations research, automated planning, and decision support applications.

Acknowledgements. This work is supported by the DFG/NWO bilateral research programme ROCKS, by the DFG as part of the SFB/TR 14 AVACS, by the EU FP7 Programme under grant agreement no. 295261 (MEALS) and 318490 (SENSATION). Andrea Turrini has received support from the Cluster of Excellence "Multimodal Computing and Interaction" (MMCI), part of the German Excellence Initiative.

References

1. Avni, G., Kupferman, O.: Making weighted containment feasible: A heuristic based on simulation and abstraction. In: CONCUR. LNCS, vol. 7454, pp. 84–99 (2012)
2. Baier, C., Engelen, B., Majster-Cederbaum, M.: Deciding bisimilarity and similarity for probabilistic processes. J. Computer and Systes Science 60(1), 187–231 (2000)
3. Bouyer, P., Fahrenberg, U., Larsen, K.G., Markey, N., Srba, J.: Infinite runs in weighted timed automata with energy constraints. In: FORMATS. LNCS, vol. 5215, pp. 33–47 (2008)
4. Černý, P., Henzinger, T.A., Radhakrishna, A.: Simulation distances. TCS 413(1), 21–35 (2012)
5. Chatterjee, K., Majumdar, R., Henzinger, T.A.: Markov decision processes with multiple objectives. In: STACS. LNCS, vol. 3884, pp. 325–336 (2006)
6. Chehaibar, G., Garavel, H., Mounier, L., Tawbi, N., Zulian, F.: Specification and verification of the PowerScale™ bus arbitration protocol: An industrial experiment with LOTOS. In: FORTE. pp. 435–450 (1996)
7. Eisentraut, C., Hermanns, H., Schuster, J., Turrini, A., Zhang, L.: The quest for minimal quotients for probabilistic automata. In: TACAS. LNCS, vol. 7795, pp. 14–33 (2013)
8. Etessami, K., Kwiatkowska, M., Vardi, M.Y., Yannakakis, M.: Molti-objective model checking of Markov decision processes. Logical Methods in Computer Science 4(8), 1–21 (2008)
9. Givan, R., Dean, T., Greig, M.: Equivalence notions and model minimization in Markov decision processes. Artificial Intelligence 147(1-2), 163–223 (2003)
10. Hermanns, H., Katoen, J.P.: Automated compositional Markov chain generation for a plain-old telephone system. Science of Computer Programming 36(1), 97–127 (2000)
11. Hermanns, H., Turrini, A.: Deciding probabilistic automata weak bisimulation in polynomial time. In: FSTTCS. pp. 435–447 (2012)
12. Hinton, A., Kwiatkowska, M., Norman, G., Parker, D.: PRISM: A tool for automatic verification of probabilistic systems. In: TACAS. LNCS, vol. 3920, pp. 441–444 (2006)
13. Howard, R.A.: Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes. Dover Publications (2007)
14. Jonsson, B., Larsen, K.G.: Specification and refinement of probabilistic processes. In: LICS. pp. 266–277 (1991)
15. Katoen, J.P., Kemna, T., Zapreev, I.S., Jansen, D.N.: Bisimulation minimisation mostly speeds up probabilistic model checking. In: TACAS. LNCS, vol. 4424, pp. 76–92 (2007)
16. Lynch, N.A., Segala, R., Vaandrager, F.W.: Observing branching structure through probabilistic contexts. SIAM J. on Computing 37(4), 977–1013 (2007)
17. Quaas, K.: Wighted timed MSO logics. In: Developments in Language Theory. LNCS, vol. 5583, pp. 419–430 (2009)
18. Segala, R.: Modeling and Verification of Randomized Distributed Real-Time Systems. Ph.D. thesis, MIT (1995)
19. Segala, R.: Probability and nondeterminism in operational models of concurrency. In: CONCUR. LNCS, vol. 4137, pp. 64–78 (2006)