Verification and Control of Turn-Based Probabilistic Real-Time Games

Marta Kwiatkowska[1], Gethin Norman[2], and David Parker[3]

[1] Department of Computing Science, University of Oxford, UK
[2] School of Computing Science, University of Glasgow, UK
[3] School of Computer Science, University of Birmingham, UK

Abstract. Quantitative verification techniques have been developed for the formal analysis of a variety of probabilistic models, such as Markov chains, Markov decision process and their variants. They can be used to produce guarantees on quantitative aspects of system behaviour, for example safety, reliability and performance, or to help synthesise controllers that ensure such guarantees are met. We propose the model of turn-based probabilistic timed multi-player games, which incorporates probabilistic choice, real-time clocks and nondeterministic behaviour across multiple players. Building on the digital clocks approach for the simpler model of probabilistic timed automata, we show how to compute the key measures that underlie quantitative verification, namely the probability and expected cumulative price to reach a target. We illustrate this on case studies from computer security and task scheduling.

1 Introduction

Probability is a crucial tool for modelling computerised systems. We can use it to model uncertainty, for example in the operating environment of an autonomous vehicle or a wireless sensor network, and we can reason about systems that use randomisation, from probabilistic routing in anonymity network protocols to symmetry breaking in communication protocols.

Formal verification of such systems can provide us with rigorous guarantees on, for example, the performance and reliability of computer networks [7], the amount of inadvertent information leakage by a security protocol [5], or the safety level of an airbag control system [2]. To do so requires us to model and reason about a range of quantitative aspects of a system's behaviour: probability, time, resource usage, and many others.

Quantitative verification techniques have been developed for a wide range of probabilistic models. The simplest are Markov chains, which model the evolution of a stochastic system over discrete time. Markov decision processes (MDPs) additionally include nondeterminism, which can be used either to model the uncontrollable behaviour of an adversary or to determine an optimal strategy (or policy) for controlling the system. Generalising this model further still, we can add a notion of real time, yielding the model of probabilistic timed automata

(PTAs). This is done in the same way as for the widely used model of timed automata (TAs), adding real-valued variables called clocks to the model. Tools such as PRISM [33] and Storm [21] support verification of a wide range of properties of these different probabilistic models.

Another dimension that we can add to these models and verification techniques is *game-theoretic* aspects. These can be used to represent, for example, the interaction between an attacker and a defender in a computer security protocol [4], between a controller and its environment, or between participants in a communication protocol who have opposing goals [27]. Stochastic multi-player games include choices made by multiple players who can either collaborate or compete to achieve their goals. Tool support for verification of stochastic games, e.g. PRISM-games [41], has also been developed and deployed successfully in a variety of application domains.

In this paper, we consider a modelling formalism that captures all these aspects: probability, nondeterminism, time and multiple players. We define a model called *turn-based probabilistic timed multi-player games* (TPTGs), which can be seen as either an extension of PTAs to incorporate multiple players, or a generalisation of stochastic multi-player games to include time. Building on known techniques for the simpler classes of models, we show how to compute key properties of these models, namely probabilistic reachability (the probability of reaching a set of target states) and expected price reachability (the expected price accumulated before reaching a set of target states).

Existing techniques for PTAs largely fall into two classes: *zone-based* and *digital clocks*, both of which construct and analyse a finite-state abstraction of the model. Zones are symbolic expressions representing sets of clock values. Zone-based approaches for analysing PTAs were first introduced in [39,40,32] and recent work extended them to the analysis of expected time [28] and expected price reachability [34]. The digital clocks approach works by mapping real-valued clocks to integer-valued ones, reducing the problem of solving a PTA to solving a (discrete-time) MDP. This approach was developed for PTAs in [38] and recently extended to the analysis of partially observable PTAs in [46].

In this paper, we show how a similar idea can be used to reduce the verification problem for TPTGs to an equivalent one over (discrete-time) stochastic games. More precisely, for the latter, we use *turn-based stochastic games* (TSGs). We first present the model of TPTGs and give two alternative semantics: one using real-valued clocks and the other using (integer-valued) digital clocks. Then, we prove the correspondence between these two semantics. Next, we demonstrate the application of this approach to two case studies from the domains of computer security and task scheduling. Using a translation from TPTGs to TSGs and the model checking tool PRISM-games, we show that a variety of useful properties can be studied on these case studies. An extended version of this paper, with complete proofs, is available [35].

Related Work. Timed games were introduced and shown to be decidable in [43,6,1]. These games have since been extensively studied; we mention [51,19], where efficient algorithms are investigated, and [47], which concerns the synthe-

sis of strategies that are robust to stochastic perturbation in clock values. Also related is the tool UPPAAL TIGA [8], which allows the automated analysis of reachability and safety problems for timed games.

Priced (or weighted) timed games were introduced in [49,3,11], which extend timed games by associating integer costs with locations and transitions, and optimal cost reachability was shown to be decidable under certain restrictions. The problem has since been shown to be undecidable for games with three or more clocks [17,10]. Priced timed games have recently been extended to allow partial observability [18] and to energy games [14].

Two-player (concurrent) probabilistic timed games were introduced in [24]. The authors demonstrated that such games are not determined (even when all clock constraints are closed) and investigated the complexity of expected time reachability for such games. Stochastic timed games [13,16] are turn-based games where time delays are exponentially distributed. A similar model, based on interactive Markov chains [26], is considered in [15].

2 Background

We start with some background and notation on *turn-based stochastic games* (TSGs). For a set X, let $\mathit{Dist}(X)$ denote the set of discrete probability distributions over X and $\mathbb{R}$ the set of non-negative real numbers.

Definition 1 (Turn-based stochastic multi-player game). *A turn-based stochastic multi-player game (TSG) is a tuple* $\mathsf{G}{=}(\Pi, S, \bar{s}, A, \langle S_i\rangle_{i\in\Pi}, \delta, R)$ *where* Π *is a finite set of players,* S *is a (possibly infinite) set of states,* $\bar{s} \in S$ *is an initial state,* A *is a (possibly infinite) set of actions,* $\langle S_i\rangle_{i\in\Pi}$ *is a partition of the state space,* $\delta : S{\times}A \to \mathit{Dist}(S)$ *is a (partial) transition function and* $R : S{\times}A \to \mathbb{R}$ *is a price (or reward) function.*

The transition function is partial in the sense that δ need not be defined for all state-action pairs. For each state s of a TSG G, there is a set of available actions given by $A(s) \stackrel{\text{def}}{=} \{a \in A \mid \delta(s,a) \text{ is defined}\}$. The choice of which available action is taken in state s is under the control of a single player: the player i such that $s \in S_i$. If player i selects action $a \in A(s)$ in s, then the probability of transitioning to state s' equals $\delta(s,a)(s')$ and a price of $R(s,a)$ is accumulated.

Paths and strategies. A path of a TSG G is a sequence $\pi = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots$ such that $s_i \in S$, $a_i \in A(s_i)$ and $\delta(s_i,a_i)(s_{i+1}){>}0$ for all $i{\geqslant}0$. For a path π, we denote by $\pi(i)$ the $(i{+}1)$th state of the path, $\pi[i]$ the action associated with the $(i{+}1)$th transition and, if π is finite, $\mathit{last}(\pi)$ the final state. The length of a path π, denoted $|\pi|$, equals the number of transitions. For a path π and $k{<}|\pi|$, let $\pi^{(k)}$ be the kth prefix of π. Let $\mathit{FPaths}_{\mathsf{G}}$ and $\mathit{IPaths}_{\mathsf{G}}$ equal the sets of finite and infinite paths starting in the initial state $\bar{s}$.

A strategy for player $i \in \Pi$ is a way of resolving the choice of action in each state under the control of player i, based on the game's execution so far. Formally, a strategy σ for player $i \in \Pi$ is a function $\sigma_i : \{\pi \in \mathit{FPaths}_{\mathsf{G}} \mid$

$last(\pi) \in S_i\} \to Dist(A)$ such that, if $\sigma_i(\pi)(a)>0$, then $a \in A(last(\pi))$. The set of all strategies of player $i \in \Pi$ is represented by Σ^i_{G} (when clear from the context we will drop the subscript G). A strategy for player i is deterministic if it always selects actions with probability 1, and memoryless if it makes the same choice for any paths that end in the same state.

A strategy profile for G takes the form $\sigma=\langle\sigma_i\rangle_{i\in\Pi}$, listing a strategy for each player. We use $FPaths^\sigma$ and $IPaths^\sigma$ for the sets of finite and infinite paths corresponding to the choices made by the profile σ when starting in the initial state. For a given profile σ, the behaviour of G is fully probabilistic and we can define a probability measure $Prob^\sigma$ over the set of infinite paths $IPaths^\sigma$ [30].

Properties. Two fundamental properties of quantitative models are the probability of reaching a set of target states and the expected price accumulated before doing so. For a strategy profile σ and set of target states F of a TSG G, the probability of reaching F and expected price accumulated before reaching F from the initial state $\bar{s}$ under the profile σ are given by the following (again, when it is clear from the context, we will drop the subscript G):

$$\begin{array}{rcl} \mathbb{P}^\sigma_{\mathsf{G}}(F) & \stackrel{\text{def}}{=} & Prob^\sigma(\{\pi \in IPaths^\sigma \mid \pi(i) \in F \text{ for some } i \in \mathbb{N}\}) \\ \mathbb{E}^\sigma_{\mathsf{G}}(F) & \stackrel{\text{def}}{=} & \int_{\pi\in IPaths^\sigma} rew(\pi, F)\,\mathrm{d}Prob^\sigma \end{array}$$

where for any infinite path π:

$$rew(\pi, F) \stackrel{\text{def}}{=} \textstyle\sum_{i=0}^{k_F} R(\pi(i), \pi[i])$$

and $k_F=\min\{k-1 \mid \pi(k) \in F\}$ if $\pi(k) \in F$ for some $k \in \mathbb{N}$ and $k_F=\infty$ otherwise.

To quantify the above properties over the strategies of the players, we consider a coalition $C \subseteq \Pi$ who try to maximise the property of interest, while the remaining players $\Pi\backslash C$ try to minimise it. Formally, we have the following definition:

$$\begin{array}{rcl} \mathbb{P}^C_{\mathsf{G}}(F) & \stackrel{\text{def}}{=} & \sup_{\sigma_1\in\Sigma^1} \inf_{\sigma_2\in\Sigma^2} \mathbb{P}^{\sigma_1,\sigma_2}_{\mathsf{G}^C}(F) \\ \mathbb{E}^C_{\mathsf{G}}(F) & \stackrel{\text{def}}{=} & \sup_{\sigma_1\in\Sigma^1} \inf_{\sigma_2\in\Sigma^2} \mathbb{E}^{\sigma_1,\sigma_2}_{\mathsf{G}^C}(F) \end{array}$$

where G^C is the two-player game constructed from G where the states controlled by player 1 equal $\cup_{i\in C} S_i$ and the states controlled by player 2 equal $\cup_{i\in\Pi\setminus C} S_i$.

The above definition yields the *optimal value* of G if it is *determined*, i.e., if the maximum value that the coalition C can ensure equals the minimum value that the coalition $\Pi \setminus C$ can ensure. Formally, the definition of determinacy and optimal strategies for probabilistic reachability properties of TSGs are given below, and the case of expected reachability is analogous (replacing $\mathbb{P}$ with $\mathbb{E}$).

Definition 2. *For a TSG G, target F and coalition of players C, we say the game G^C is* determined *with respect to probabilistic reachability if:*

$$\sup_{\sigma_1\in\Sigma^1} \inf_{\sigma_2\in\Sigma^2} \mathbb{P}^{\sigma_1,\sigma_2}(F) \;=\; \inf_{\sigma_2\in\Sigma^2} \sup_{\sigma_1\in\Sigma^1} \mathbb{P}^{\sigma_1,\sigma_2}(F)\,.$$

Furthermore, a strategy $\sigma^\star_1 \in \Sigma_1$ is optimal *if $\mathbb{P}^{\sigma^\star_1,\sigma_2}(F) \geqslant \mathbb{P}^C_{\mathsf{G}}(F)$ for all $\sigma_2 \in \Sigma^2$ and strategy $\sigma^\star_2 \in \Sigma_2$ is* optimal *if $\mathbb{P}^{\sigma_1,\sigma^\star_2}(F) \leqslant \mathbb{P}^C_{\mathsf{G}}(F)$ for all $\sigma_1 \in \Sigma^1$.*

As we shall demonstrate, the games we consider are determined with respect to probabilistic and expected reachability, and optimal strategies exist. In particular, finite-state and finite-branching TSGs are determined [31] and efficient techniques exist to approximate optimal values and optimal strategies [20,22]. These techniques underlie the model checking algorithms for logics such as rPATL, defined for TSGs and implemented in the tool PRISM-games [41].

3 Turn-based Probabilistic Timed Multi-Player Games

We now introduce *turn-based probabilistic timed multi-player games* (TPTGs), a framework for modelling systems which allows probabilistic, non-deterministic, real-time and competitive behaviour. Let $\mathbb{T} \in \{\mathbb{R}, \mathbb{N}\}$ be the time domain of either the non-negative reals or natural numbers.

Clocks, valuations and clock constraints. We assume a finite set of *clocks* $\mathcal{X}$. A *clock valuation* is a function $v : \mathcal{X} \to \mathbb{T}$; the set of all clock valuations is denoted $\mathbb{T}^{\mathcal{X}}$. Let $\mathbf{0}$ be the clock valuation that assigns the value 0 to all clocks. For any set of clocks $X \subseteq \mathcal{X}$ and clock valuation $v \in \mathbb{T}^{\mathcal{X}}$, let $v[X{:=}0]$ be the clock valuation such that, for any clock x, we have $v[X{:=}0](x)$ equals 0 if $x \in X$ and $v(x)$ otherwise. Furthermore, for any time instant $t \in \mathbb{T}$, let $v{+}t$ be the clock valuation such that $(v{+}t)(x) = v(x){+}t$ for all $x \in \mathcal{X}$. A closed, diagonal-free clock constraint[4] ζ is a conjunction of inequalities of the form $x \leqslant c$ or $x \geqslant c$, where $x \in \mathcal{X}$ and $c \in \mathbb{N}$. We write $v \models \zeta$ if the clock valuation v satisfies the clock constraint ζ and use $CC(\mathcal{X})$ for the set of all clock constraints over $\mathcal{X}$.

We are now in a position to present the syntax and semantics of TPTGs.

Definition 3 (TPTG syntax). *A* turn-based probabilistic timed multi-player game *(TPTG) is a tuple* $\mathsf{P}{=}(\Pi, L, \bar{l}, \mathcal{X}, Act, \langle L_i \rangle_{i \in \Pi}, inv, enab, prob, r)$ *where:*

- Π *is a finite set of* players*;*
- L *is a finite set of* locations *and* $\bar{l} \in L$ *is an* initial location*;*
- $\mathcal{X}$ *is a finite set of* clocks;
- Act *is a finite set of* actions;
- $\langle L_i \rangle_{i \in \Pi}$ *is a partition of* L*;*
- $inv : L \to CC(\mathcal{X})$ *is an* invariant condition;
- $enab : L{\times}A \to CC(\mathcal{X})$ *is an* enabling condition;
- $prob : L{\times}A \to Dist(2^{\mathcal{X}}{\times}L)$ *is a (partial)* probabilistic transition function;
- $r = (r_L, r_{Act})$ *is a* price structure *where* $r_L : L \to \mathbb{N}$ *is a location price function and* $r_{Act} : L{\times}Act \to \mathbb{N}$ *an action price function.*

As for PTAs [45], a state of a TPTG P is a location-clock valuation pair (l, v) such that the clock valuation satisfies the invariant $inv(l)$. The transition choice in (l, v) is under the control of the player i where $l \in L_i$. A transition is a time-action pair (t, a) which represents letting time t elapse and then performing action a. Time can elapse if the invariant of the current location remains continuously

[4] A constraint is closed if does not contain strict inequalities and diagonal-free if there are no inequalities of the form $x{-}y \sim c$ for $x, y \in \mathcal{X}$, $\sim \in \{<, \leqslant, \geqslant, >\}$ and $c \in \mathbb{N}$.

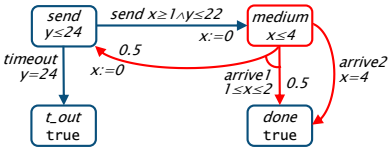


Fig. 1. An example TPTG

satisfied and action a can be performed only if the enabling condition is satisfied. If action a is taken in location l, then the probability of moving to location l' and resetting the set of clocks X equals $prob(l,a)(X,l')$. TPTGs have both location prices, which are accumulated at rate $r_L(l)$ when time passes in location l, and action prices, where $r_{Act}(l,a)$ is accumulated when performing action a in location l. Formally, the semantics of a TPTG is a TSG defined as follows.

Definition 4 (TPTG semantics). *For any time domain $\mathbb{T} \in \{\mathbb{R}, \mathbb{N}\}$ and TPTG $\mathsf{P}=(\Pi, L, \bar{l}, \mathcal{X}, Act, \langle L_i\rangle_{i\in\Pi}, inv, enab, prob, r)$ the semantics of P with respect to the time domain $\mathbb{T}$ is the TSG $[\![\mathsf{P}]\!]_{\mathbb{T}}=(\Pi, S, \bar{s}, A, \langle S_i\rangle_{i\in\Pi}, \delta, R)$ where:*

- $S = \{(l,v) \in L\times\mathbb{T}^{\mathcal{X}} \mid v \models inv(l)\}$ *and* $\bar{s} = (\bar{l}, \mathbf{0})$;
- $A = \mathbb{T}\times Act$;
- $S_i = \{(l,v) \in S \mid l \in L_i\}$ *for* $i \in \Pi$;
- *for any* $(l,v) \in S$ *and* $(t,a) \in A$ *we have* $\delta((l,v),(t,a)) = \mu$ *if and only if* $v+t' \models inv(l)$ *for all* $0 \leqslant t' \leqslant t$, $v+t \models enab(l,a)$ *and for any* $(l',v') \in S$:

$$\mu(l',v') = \sum_{X\subseteq\mathcal{X} \wedge v'=(v+t)[X:=0]} prob(l,a)(X,l')$$

- $R((l,v),(t,a)) = t{\cdot}r_L(l) + r_{Act}(l,a)$ *for all* $(l,v) \in S$ *and* $(t,a) \in A$.

We follow the approach of [29,24,28] and use time-action pairs in the transition function of Definition 4. As explained in [28], this yields a more expressive semantics than having separate time and action transitions.

Example 1. Consider the TPTG in Fig. 1 which represents a simple communication protocol. There are two players: the sender and the medium, with the medium controlling the location *medium* and the sender all other locations. The TPTG has two clocks: x is used to keep track of the time it takes to send a message and y the elapsed time. In the initial location *send*, the sender waits between 1 and 2 time units before sending the message. The message then passes through the medium that can either delay the message for between 1 and 2 time units after which it arrives with probability 0.5, or delay the message for 4 time units after which it arrives with probability 1. If the message does not arrive, then the sender tries to send it again until reaching a timeout after 24 time units.

As for PTAs, in the standard (dense-time) semantics for a TPTG the time domain $\mathbb{T}$ equals $\mathbb{R}$. This yields an infinite state model which is not amenable to verification. One approach that yields a finite state representation used in the case of PTAs is the digital clocks semantics [38]. This is based on replacing the real-valued clocks with clocks taking only values from a bounded set of integers. In order to give the definition for a TPTG P, for any clock x of P we define k_x

to be the greatest constant to which x is compared in the clock constraints of P. This allows us to use bounded clock values since, if the value of the clock x exceeds k_x, then the exact value will not affect the satisfaction of the invariants and enabling conditions of P, and therefore does not influence the behaviour.

Definition 5 (Digital clocks semantics). *The digital clocks semantics of a TPTG* P*, denoted* $[\![\mathsf{P}]\!]_{\mathbb{N}}$*, is obtained from Definition 4, by setting* $\mathbb{T}$ *equal to* $\mathbb{N}$ *and for any* $v \in \mathbb{N}^{\mathcal{X}}$*,* $t \in \mathbb{N}$ *and* $x \in \mathcal{X}$ *letting* $(v{+}t)(x) = \min\{v(x){+}t, k_x{+}1\}$*.*

We restrict our attention to time-divergent (also called non-Zeno) behaviour. More precisely, we only consider strategies for the players that do not generate unrealisable executions, i.e., executions in which time does not advance beyond a certain point. We achieve this by restricting to TPTGs that satisfy the syntactic conditions for PTAs given in [45], derived from results on TAs [50,52]. In addition, we require the following assumptions to ensure the correctness of the digital clocks semantics.

Assumption 1 *For any TPTG* P *: (a) all invariants of* P *are bounded; (b) all clock constraints are closed and diagonal free; (c) all probabilities are rational.*

Regarding Assumption 1(a), in fact bounded TAs are as expressive as standard TAs [9], and this result carries over to TPTGs.

To facilitate higher-level modelling, PTAs can be extended with parallel composition, discrete variables, urgent transitions and locations and resetting clocks to integer values [45]. We can extend TPTGs in a similar way, and will use these constructs in Section 5.

4 Correctness of the Digital Clocks Semantics

We now show that, under Assumption 1, optimal probabilistic and expected price reachability values agree under the digital and dense-time semantics. As for PTAs [45], by modifying the TPTG under study, we can reduce time-bounded probabilistic reachability properties to probabilistic reachability properties and both expected time-bounded cumulative price properties and expected time-instant price properties to expected reachability properties. In each case the modifications to the TPTG preserve Assumption 1, and therefore the digital clocks semantics can also be used to verify these classes of properties.

For the remainder of this section, we fix a TPTG P, coalition of players C and set of target locations $F \subseteq L$, and let $F_{\mathbb{T}} = \{(l, v) \in F \times \mathbb{T}^{\mathcal{X}} \mid v \models inv(l)\}$ for $\mathbb{T} \in \{\mathbb{R}, \mathbb{N}\}$. We have omitted the proofs that closely follow those for PTAs [38]. The missing proofs can be found in [35].

We first present results relating to the determinacy and existence of optimal strategies for the games $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$ and $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$ and a correspondence between the strategy profiles of $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$ and $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$.

Proposition 1. *For any TPTG* P *satisfying Assumption 1, the games* $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$ *and* $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$ *are determined and have optimal strategies for both probabilistic and expected price reachability properties.*

Proof. In the case of $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$, the result follows from [23] and Assumption 1, i.e. since all clock constraints are closed. Considering $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$, the result follows from the fact that the game has a finite state space and is finitely branching [31]. □

Proposition 2. *For any strategy profile σ' of $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$, there exists a strategy profile σ of $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$ such that $\mathbb{P}^{\sigma}(F_{\mathbb{R}}) = \mathbb{P}^{\sigma'}(F_{\mathbb{N}})$ and $\mathbb{E}^{\sigma}(F_{\mathbb{R}}) = \mathbb{E}^{\sigma'}(F_{\mathbb{N}})$.*

Using the ϵ-digitization approach of TAs [25], which has been extended to PTAs in [38], the following theorem follows, demonstrating the correctness of the digital clocks semantics for probabilistic reachability properties.

Theorem 1. *For any TPTG P satisfying Assumption 1, coalition of players C and set of locations $F \subseteq L$: $\mathbb{P}^C_{[\![\mathsf{P}]\!]_{\mathbb{R}}}(F_{\mathbb{R}}) = \mathbb{P}^C_{[\![\mathsf{P}]\!]_{\mathbb{N}}}(F_{\mathbb{N}})$.*

For expected price reachability properties, we extend the approach of [38], by first showing that, for any fixed (dense-time) profile σ of $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$ and $n \in \mathbb{N}$, there exist profiles of $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$ whose expected price of reaching the target locations F within n transitions that provide lower and upper bounds for that of σ.

For $\mathbb{T} \in \{\mathbb{R}, \mathbb{N}\}$, profile σ of $[\![\mathsf{P}]\!]^C_{\mathbb{T}}$, and finite path $\pi \in \mathit{FPaths}^{\sigma}$ we inductively define the values $\langle \mathbb{E}^{\sigma}_n(\pi, F_{\mathbb{T}})\rangle_{n\in\mathbb{N}}$ which equal the expected price, under the profile σ, of reaching the target $F_{\mathbb{T}}$ after initially performing the path π within n steps. To ease presentation we only give the definition for deterministic profiles.

Definition 6. *For $\mathbb{T} \in \{\mathbb{R}, \mathbb{N}\}$, strategy profile $\sigma{=}(\sigma_1, \sigma_2)$ of $[\![\mathsf{P}]\!]^C_{\mathbb{T}}$ and finite path π of the profile let $\mathbb{E}^{\sigma_1,\sigma_2}_0(\pi, F_{\mathbb{T}}) = 0$ and for any $n \in \mathbb{N}$, if $\mathit{last}(\pi){=}(l, v) \in S_i$ for $1{\leqslant}i{\leqslant}2$, $\sigma_i(\pi){=}(t, a)$ and $\mu = P_{[\![\mathsf{P}]\!]_{\mathbb{T}}}((l, v), (t, a))$, then:*

$$\mathbb{E}^{\sigma}_{n+1}(\pi, F_{\mathbb{T}}) = \begin{cases} 0 & \text{if } (l, v) \in F_{\mathbb{T}} \\ r_L(l){\cdot}t + r_{Act}(l, a) + \sum\limits_{s' \in S} \mu(s') \cdot \mathbb{E}^{\sigma}_n(\pi \xrightarrow{t,a} s', F_{\mathbb{T}}) & \text{otherwise.} \end{cases}$$

We require the following properties of these expected price reachability properties. These then allow us to prove the correctness of the digital clocks semantics for expected price reachability properties (Theorem 2 below).

Lemma 1. *For $\mathbb{T} \in \{\mathbb{R}, \mathbb{N}\}$ and profile σ of $[\![\mathsf{P}]\!]^C_{\mathbb{T}}$, the sequence $\langle \mathbb{E}^{\sigma}_n(F_{\mathbb{T}})\rangle_{n\in\mathbb{N}}$ is non-decreasing and converges to $\mathbb{E}^{\sigma}(F_{\mathbb{T}})$, and, for any player 1 strategy σ_1 of $[\![\mathsf{P}]\!]^C_{\mathbb{T}}$, the sequence of functions $\mathbb{E}^{\sigma_1,\cdot}_n(F_{\mathbb{T}}) : \Sigma^2 \to \mathbb{R}$ converges uniformly. Furthermore, for any player 1 strategy σ_1, the sequence $\langle \inf_{\sigma_2\in\Sigma^2} \mathbb{E}^{\sigma_1,\sigma_2}_n(F_{\mathbb{T}})\rangle_{n\in\mathbb{N}}$ is non-decreasing and converges to $\inf_{\sigma_2\in\Sigma^2} \mathbb{E}^{\sigma_1,\sigma_2}(F_{\mathbb{T}})$, and the sequence of functions $\inf_{\sigma_2\in\Sigma^2} \mathbb{E}^{\cdot,\sigma_2}_n(F_{\mathbb{T}}) : \Sigma^1 \to \mathbb{R}$ converges uniformly.*

Proof. In each case, proving that the sequence is non-decreasing and converges follows from Definition 6. Uniform convergence follows from showing the set of strategies for players is compact and using the fact that the sequences are non-decreasing and converge pointwise [48, Theorem 7.13]. In the case when $\mathbb{T}{=}\mathbb{N}$, compactness follows from the fact the action set is finite, while for $\mathbb{T}{=}\mathbb{R}$ we must restrict to PTAs for which all invariants are bounded (Assumption 1) to ensure the action set is compact. □

Lemma 2. *For any strategy profile σ of $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$ and $n \in \mathbb{N}$, there exist strategy profiles σ^{lb} and σ^{ub} of $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$ such that: $\mathbb{E}^{\sigma^{lb}}_n(F_{\mathbb{N}}) \leqslant \mathbb{E}^{\sigma}_n(F_{\mathbb{R}}) \leqslant \mathbb{E}^{\sigma^{ub}}_n(F_{\mathbb{N}})$.*

Theorem 2. *For any TPTG P satisfying Assumption 1, coalition of players C and set of locations $F \subseteq L$: $\mathbb{E}^C_{[\![\mathsf{P}]\!]_{\mathbb{R}}}(F_{\mathbb{R}}) = \mathbb{E}^C_{[\![\mathsf{P}]\!]_{\mathbb{N}}}(F_{\mathbb{N}})$.*

Proof. Consider any $n \in \mathbb{N}$. Using Lemma 2 it follows that, for any profile $\sigma{=}(\sigma_1, \sigma_2)$ of $[\![\mathsf{P}]\!]_{\mathbb{R}}$, there exist profiles $\sigma^{lb}{=}(\sigma_1^{lb}, \sigma_1^{lb})$ and $\sigma^{ub}{=}(\sigma_1^{ub}, \sigma_1^{ub})$ of $[\![\mathsf{P}]\!]^C_{\mathbb{N}}$ such that:

$$\mathbb{E}^{\sigma_1^{lb},\sigma_2^{lb}}_n(F_{\mathbb{N}}) \leqslant \mathbb{E}^{\sigma_1,\sigma_2}_n(F_{\mathbb{R}}) \leqslant \mathbb{E}^{\sigma_1^{ub},\sigma_2^{ub}}_n(F_{\mathbb{N}})\,.$$

On the other hand, using the construction in the proof of Proposition 2, for any profile $\sigma'{=}(\sigma'_1, \sigma'_2)$ of $[\![\mathsf{P}]\!]_{\mathbb{N}}$, there exists a profile $\sigma{=}(\sigma_1, \sigma_1)$ of $[\![\mathsf{P}]\!]^C_{\mathbb{R}}$ such that:

$$\mathbb{E}^{\sigma_1,\sigma_2}_n(F_{\mathbb{N}}) = \mathbb{E}^{\sigma'_1,\sigma'_2}_n(F_{\mathbb{R}})\,.$$

Combining these results with Proposition 1 it follows that:

$$\sup_{\sigma'_1 \in \Sigma^1_{[\![\mathsf{P}]\!]^C_{\mathbb{N}}}} \inf_{\sigma'_2 \in \Sigma^2_{[\![\mathsf{P}]\!]^C_{\mathbb{N}}}} \mathbb{E}^{\sigma'_1,\sigma'_2}_n(F_{\mathbb{N}}) = \sup_{\sigma_1 \in \Sigma^1_{[\![\mathsf{P}]\!]^C_{\mathbb{R}}}} \inf_{\sigma_2 \in \Sigma^2_{[\![\mathsf{P}]\!]^C_{\mathbb{R}}}} \mathbb{E}^{\sigma_1,\sigma_2}_n(F_{\mathbb{R}})\,.$$

Since $n \in \mathbb{N}$ was arbitrary, we have:

$$\lim_{n\to\infty} \sup_{\sigma'_1 \in \Sigma^1_{[\![\mathsf{P}]\!]^C_{\mathbb{N}}}} \inf_{\sigma'_2 \in \Sigma^2_{[\![\mathsf{P}]\!]^C_{\mathbb{N}}}} \mathbb{E}^{\sigma'_1,\sigma'_2}_n(F_{\mathbb{N}}) = \lim_{n\to\infty} \sup_{\sigma_1 \in \Sigma^1_{[\![\mathsf{P}]\!]^C_{\mathbb{R}}}} \inf_{\sigma_2 \in \Sigma^2_{[\![\mathsf{P}]\!]^C_{\mathbb{R}}}} \mathbb{E}^{\sigma_1,\sigma_2}_n(F_{\mathbb{R}})$$

and hence using Lemma 1 it follows that:

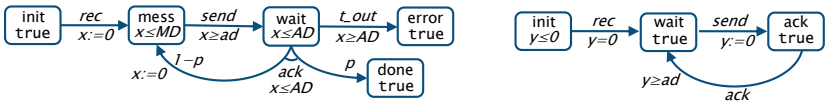
$$\sup_{\sigma'_1 \in \Sigma^1_{[\![\mathsf{P}]\!]^C_{\mathbb{N}}}} \inf_{\sigma'_2 \in \Sigma^2_{[\![\mathsf{P}]\!]^C_{\mathbb{N}}}} \mathbb{E}^{\sigma'_1,\sigma'_2}_n(F_{\mathbb{N}}) = \sup_{\sigma_1 \in \Sigma^1_{[\![\mathsf{P}]\!]^C_{\mathbb{R}}}} \inf_{\sigma_2 \in \Sigma^2_{[\![\mathsf{P}]\!]^C_{\mathbb{R}}}} \mathbb{E}^{\sigma_1,\sigma_2}_n(F_{\mathbb{R}})\,.$$

The fact that the limit can move inside the sup and inf operators on both sides of the inequality follows from the uniform convergence results of Lemma 1. □

5 Case Studies

In this section, we apply our approach to two case studies, a security protocol and a scheduling problem, both of which have been previously modelled as PTAs [38]. In both case studies, by working with games we are able to give more realistic models that overcome the limitations of the earlier PTA models. We specify the finite-state TSG digital clocks semantic models of the case studies using the PRISM language and employ the PRISM-games tool [41] to perform the analysis. Using PRISM-games we are not only able to find optimal probabilistic and expected reachability values, but also synthesise optimal strategies for the players. PRISM files for the case studies are available from [53].

Non-repudiation protocol. Markowitch and Roggeman's non-repudiation protocol for information transfer [44] is designed to allow an originator O to transfer information to a recipient R while guaranteeing non-repudiation, that is, neither O nor R can deny that they participated in the transfer.



(a) Originator (b) Honest recipient

Fig. 2. PTAs used to model the non-repudiation protocol

Randomisation is fundamental to the protocol as, in the initialisation step, O randomly selects a positive integer N that is never revealed to R during execution. Timing is also fundamental as, to prevent R potentially gaining an advantage, if O does not receive an acknowledgement within a specific time-out value (denoted AD), the protocol is stopped and O states R is trying to cheat. In previous PTA models of the protocol [42,45] the originator O had fixed behaviour, while the choices of a (malicious) recipient R included varying the delay between receiving a message from O and sending an acknowledgement. By modelling the protocol as a two-player game we can allow both O and R to make choices which can depend on the history, i.e., the previous behaviour of the parties. The game is naturally turn-based since, in each round, first O sends a message after a delay of their choosing and, after receiving this message, R can respond with an acknowledgement after a delay of their choosing.

We first consider an 'honest' version of the protocol where both O and R can choose delays for their messages but do follow the protocol (i.e., send messages and acknowledgements before timeouts occur). The component PTA models for O and R are presented in Fig. 2. In the PTA for O, the message delay is between md=2 and MD=9 time units, while the acknowledgement delay is at least ad=1 time units and AD=5 is the timeout value. In addition, the probabilistic choice of N is made using a geometric distribution with parameter $p \in (0, 1]$. The parallel composition of these two components then gives the TPTG model of the protocol by assigning control of locations to either O or R, based on which party decides on the delay. There is a complication in the location where O is waiting and R is sending an acknowledgement, as the delay before sending the acknowledgement controlled by R, while if the timeout is reached O should end the protocol. However, since O's behaviour is deterministic in this location, we assign this location to be under the control of R, but add the constraints that an acknowledgement can only be sent before the timeout is reached. If O's behaviour was not deterministic, then a turn-based model would not be sufficient and the protocol would need to be modelled as a concurrent game.

We also consider two 'malicious' versions of the protocol, one in which R is allowed to guess which is the last message (malicious version 1) and a version further extended by giving R additional power through a probabilistic decoder that can decode a message with probability 0.25 before O will timeout (malicious version 2). The TPTG models follow the same structure as that for the 'honest' version, requiring that, in the locations where O is waiting for an acknowledgement, once the timeout has been reached the only possible behaviour is for the protocol to terminate and O states R is trying to cheat.

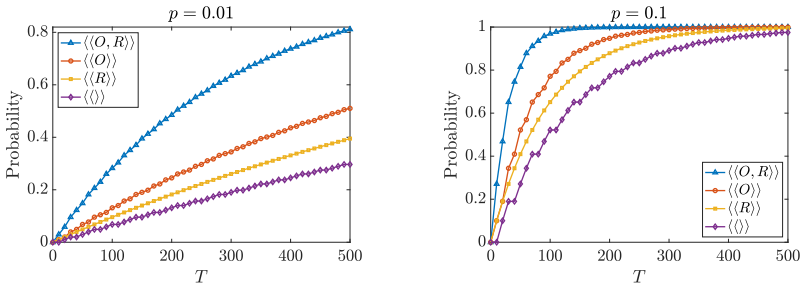


Fig. 3. Max. probability the protocol terminates successfully by time T (honest version)

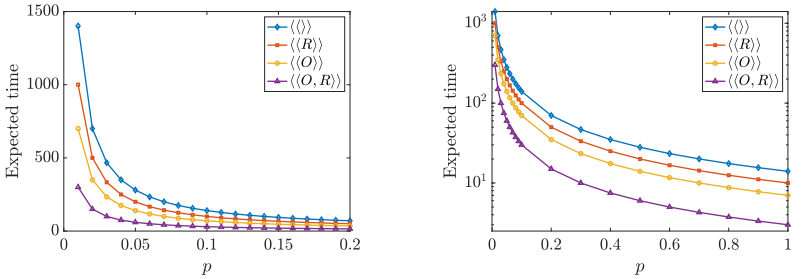


Fig. 4. Min. expected time until the protocol terminates successfully (honest version)

For the 'honest' version, Figures 3 and 4 present results when different coalitions try to maximise the probability the protocol terminates successfully by time T when p=0.01 and p=0.1 and minimise expected time for successful termination as the parameter p varies. More precisely, we consider the coalition of both players ($\langle\langle O, R\rangle\rangle$), a single player ($\langle\langle O\rangle\rangle$ or $\langle\langle R\rangle\rangle$) and the empty coalition ($\langle\langle\rangle\rangle$). Using a PTA model, only the first and last cases could be considered. As can be seen, both parties have some control over the time it takes for the protocol to complete and O has greater power as it can delay messages longer than R (if R delays too long then O will terminate the protocol stating R is cheating).

In the case of the versions with a malicious recipient, in Figures 5 and 6 we have plotted the maximum probability the recipient gains information by time T for versions 1 and 2 respectively. We have included the cases where O works against R ($\langle\langle R\rangle\rangle$) and where they collaborate ($\langle\langle O, R\rangle\rangle$). As we can see, although O cannot reduce the probability of R obtaining information, it can to some extent increase the time it takes R to obtain this information.

Processor Task Scheduling. This case study is based on the task-graph scheduling problem from [12]. The task-graph is given in Fig. 7 and is for evaluating the expression $D\times(C\times(A+B))+((A+B)+(C\times D))$ where each multiplication and addition is evaluated on one of two processors, P_1 and P_2. The time and energy required to perform these operations is different, with P_1 being faster than P_2 while consuming more energy as detailed below.

– Time and energy usage of P_1: $[0, 2]$ picoseconds for addition, $[0, 3]$ picoseconds multiplication, 10 Watts when idle and 90 Watts when active.

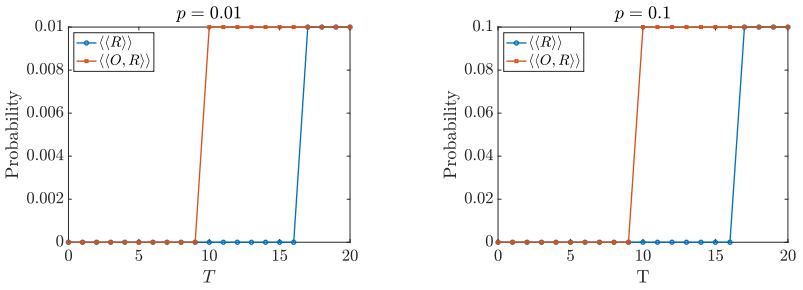


Fig. 5. Maximum probability R gains information by time T (malicious version 1)

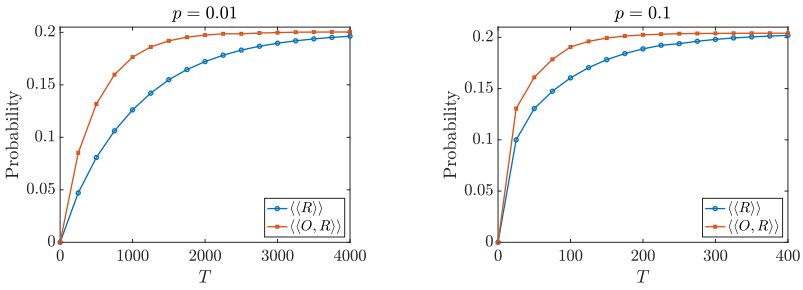


Fig. 6. Maximum probability R gains information by time T (malicious version 2)

– Time and energy usage of P_2: $[0, 5]$ picoseconds for addition, $[0, 7]$ picoseconds multiplication, 20 Watts when idle and 30 Watts when active.

A (non-probabilistic) TA model is considered in [12], which is the parallel composition of a TA for each processor and for the scheduler. Previously, in [45], we extended this model by adding probabilistic behaviour to give a PTA. However, the execution time of the processors had to remain fixed since the nondeterminism was under the control of the scheduler, and therefore the optimal scheduler would always choose the minimum execution time for each operation. By moving to a TPTG model, we can allow the execution times to be under the control of a separate player (the environment). We further extend the model by allowing the processors P_1 and P_2 to have at most k_1 and k_2 faults respectively. We assume that the probability of any fault causing a failure is p and faults can happen at any time a processor is active, i.e., the time the faults occur is under the control of the environment. Again, we could not model this extension with a PTA, since the scheduler would then be in control of when the faults occurred, and therefore could decide that no faults would occur.

As explained in [12], an optimal schedule for a game model in which delays can vary does not yield a simple assignment of tasks to processors at specific times as presented in [45] for PTAs, but instead it is an assignment that also has as input when previous tasks were completed and on which processors.

In Fig. 8 we present both the original TA model for processor P_1, in which the execution time is non-deterministic, and the extended PTA, which allows k_1 faults and where the probability of a fault causing a failure equals p. The PTA includes an integer variable *faults* and the missing enabling conditions equal

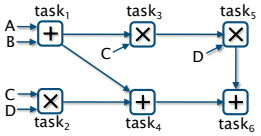
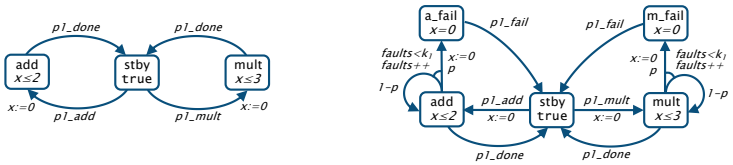


Fig. 7. Task graph for computing $D\times(C\times(A+B))+((A+B)+(C\times D))$



(a) Processor P_1 (b) Faulty version of processor P_1

Fig. 8. PTAs for the task-graph scheduling case study

`true`. To specify the automaton for the scheduler and ensure that we can then build a turn-based game, we restrict the scheduler so that it decides what tasks to schedule initially and immediately after a task ends, then passes control to the environment, which decides the time for the next active task to end.

In Fig. 9 we have plotted both the optimal expected time and energy when there are different number of faults in each processor as the parameter p (the probability of a fault causing a failure) varies. As would be expected, both the optimal expected time and energy consumption increases both as the number of faults increases and the probability that a fault causes a failure increases.

Considering the synthesised optimal schedulers for the expected time case, when $k_1{=}k_2{=}1$ and $p{=}1$, the optimal approach is to just use the faster processor P_1 and the expected time equals 18.0. The optimal strategy for the environment, i.e., the choices that yield the worst-case expected time, against this scheduler is to delay all tasks as long as possible and cause a fault when a multiplication task is just about to complete on P_1 (recall P_2 is never used under the optimal scheduler). A multiplication is chosen as this takes longer (3 picoseconds) than an addition task (2 picoseconds). These choices can be seen through the fact that 18.0 is the time for 4 multiplications and 3 additions to be performed on P_1, while the problem requires 3 multiplications and 3 additions. As soon as the probability of a fault causing a failure is less than 1, the optimal scheduler does use processor P_2 from the beginning by initially scheduling $task_1$ on process P_1 and $task_2$ on processor P_2 (which is also optimal when no faults can occur).

In the case of the expected energy consumption, the optimal scheduler uses both processes unless one has 2 or more faults than the other and there is only a small chance that a fault will cause a failure. For example, if P_1 has 3 faults and P_2 has 1 fault, then P_1 is only used by the optimal scheduler when the probability of a failure causing a fault is approximately 0.25 or less.

6 Conclusions

We have introduced turn-based probabilistic timed games and shown that digital clocks are sufficient for analysing a large class of such games and performance

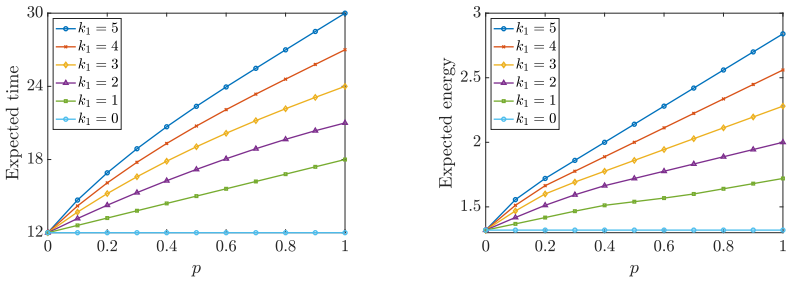


Fig. 9. Minimum expected time and energy to complete all tasks (k_2=k_1)

properties. We have demonstrated the feasibility of the approach through two case studies. However, there are limitations of the method since, in particular, as for PTAs [38], the digital clocks semantics does not preserve stopwatch properties or general (nested) temporal logic specifications.

We are investigating extending the approach to concurrent probabilistic timed games. However, since such games are not determined for expected reachability properties [24], this is not straightforward. One direction is to find a class of games which are determined. If we are able to find such a class, then the extension of PRISM-games to concurrent stochastic games [36] could be used to verify this class. Work on finite-state concurrent stochastic games has recently been extended to the case when players have distinct objectives [37] and considering such objectives in the real-time case is also a direction of future research.

Another direction of future research is to formulate a zone-based approach for verifying probabilistic timed games. For the case of probabilistic reachability, this appears possible through the approach of [32] for PTAs. However, it is less clear that the techniques for expected time [28] and expected prices [34], and temporal logic specifications [40] for PTAs, can be extended to TPTGs. Finally, we mention that, although the PRISM language models used in Section 5 were built by hand, in future we plan to automate this procedure, extending the one already implemented in PRISM [33] for PTAs.

Acknowledgements. This work is partially supported by the EPSRC Programme Grant on Mobile Autonomy and the PRINCESS project, under the DARPA BRASS programme (contract FA8750-16-C-0045).

References

1. de Alfaro, L., Faella, M., Henzinger, T., Majumdar, R., Stoelinga., M.: The element of surprise in timed games. In: Proc. CONCUR'03, LNCS 2761. Springer (2003)
2. Aljazzar, H., Fischer, M., Grunske, L., Kuntz, M., Leitner, F., Leue, S.: Safety analysis of an airbag system using probabilistic FMEA and probabilistic counter examples. In: Proc. QEST'09. IEEE (2009)
3. Alur, R., Mikhail, B., Madhusudan, P.: Optimal reachability for weighted timed games. In: Proc. ICALP'04, LNCS 3142. Springer (2004)
4. Alvim, M., Chatzikokolakis, K., Kawamoto, Y., Palamidessi, C.: A game-theoretic approach to information-flow control via protocol composition. Entropy **20**(5) (2018)

5. Alvim, M., Chatzikokolakis, K., Palamidessi, C., Smith, G.: Measuring information leakage using generalized gain functions. In: Proc. CSF'12. IEEE (2012)
6. Asarin, E., Maler, O., Pnueli, A., Sifakis, J.: Controller synthesis for timed automata. In: Proc. SSC'98. Elsevier (1998)
7. Baier, C., Haverkort, B., Hermanns, H., Katoen, J.P.: Performance evaluation and model checking join forces. CACM **53**(9) (2010)
8. Behrmann, G., Cougnard, A., David, A., Fleury, E., Larsen, K., Lime, D.: UPPAAL-Tiga: Time for playing games! In: Proc. CAV'07. Springer (2007)
9. Behrmann, G., Fehnker, A., Hune, T., Larsen, K., Pettersson, P., Romijn, J., Vaandrager, F.: Minimum-cost reachability for priced timed automata. In: Proc. HSCC'01, LNCS 2034. Springer (2001)
10. Bouyer, P., Brihaye, T., Markey, N.: Improved undecidability results on weighted timed automata. IPL **98** (2006)
11. Bouyer, P., Cassez, F., Fleury, E., Larsen, K.: Optimal strategies in priced timed game automata. In: Proc. FSTTCS'04, LNCS 3328. Springer (2004)
12. Bouyer, P., Fahrenberg, U., Larsen, K., Markey, N.: Quantitative analysis of real-time systems using priced timed automata. Comm. ACM **54**(9) (2011)
13. Bouyer, P., Forejt, V.: Reachability in stochastic timed games. In: Proc. ICALP'09, LNCS 5556. Springer (2009)
14. Bouyer, P., Markey, N., Randour, M., Larsen, K., Laursen, S.: Average-energy games. Acta Informatica **55**(2) (2018)
15. Brázdil, T., Hermanns, H., Krcál, J., Kretínský, J., Rehák, V.: Verification of open interactive Markov chains. In: Proc. FSTTCS'12, LIPIcs 18 (2012)
16. Brázdil, T., Krcál, J., Kretínský, J., Kucera, A., Rehák, V.: Stochastic real-time games with qualitative timed automata objectives. In: Proc. CONCUR'10 (2010)
17. Brihaye, T., Bruyère, V., Raskin, J.: On optimal timed strategies. In: Proc. FORMATS'05, LNCS 3829. Springer (2005)
18. Cassez, F., David, A., Larsen, K., Lime, D., Raskin, J.F.: Timed control with observation based and stuttering invariant strategies. In: Proc. ATVA'07 (2007)
19. Cassez, F., David, D., Fleury, E., Larsen, K., Lime, D.: Efficient on-the-fly algorithms for the analysis of timed games. In: Proc. CONCUR'05, LNCS 3653. Springer (2005)
20. Condon, A.: On algorithms for simple stochastic games. Advances in computational complexity theory, DIMACS Series in DMTCS **13** (1993)
21. Dehnert, C., Junges, S., Katoen, J.P., Volk, M.: A storm is coming: A modern probabilistic model checker. In: Proc. CAV'17, LNCS 10427. Springer (2017)
22. Filar, J., Vrieze, K.: Competitive Markov Decision Processes. Springer (1997)
23. Forejt, V., Kwiatkowska, M., Norman, G., Trivedi, A.: Expected reachability-time games. In: Proc. FORMATS'10, LNCS 6246. Springer (2010)
24. Forejt, V., Kwiatkowska, M., Norman, G., Trivedi, A.: Expected reachability-time games. TCS **631** (2016)
25. Henzinger, T., Manna, Z., Pnueli, A.: What good are digital clocks? In: Proc. ICALP'92, LNCS 623. Springer (1992)
26. Hermanns, H.: Interactive Markov Chains and the Quest for Quantified Quality. LNCS 2428. Springer (2002)
27. van der Hoek, W., Wooldridge, M.: Model checking cooperation, knowledge, and time - A case study. Research In Economics **57**(3) (2003)
28. Jovanovic, A., Kwiatkowska, M., Norman, G., Peyras, Q.: Symbolic optimal expected time reachability computation and controller synthesis for probabilistic timed automata. TCS **669** (2017)

29. Jurdziński, M., Kwiatkowska, M., Norman, G., Trivedi, A.: Concavely-priced probabilistic timed automata. In: Proc. CONCUR'09, LNCS 5710. Springer (2009)
30. Kemeny, J., Snell, J., Knapp, A.: Denumerable Markov Chains. Springer (1976)
31. Krčál, J.: Determinacy and optimal strategies in stochastic games. Master's thesis, School of Informatics, Masaryk University, Brno (2009)
32. Kwiatkowska, M., Norman, G., Parker, D.: Stochastic games for verification of probabilistic timed automata. In: Proc. FORMATS'09. Springer (2009)
33. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Proc. CAV'11, LNCS 6806. Springer (2011)
34. Kwiatkowska, M., Norman, G., Parker, D.: Symbolic verification and strategy synthesis for linearly-priced probabilistic timed automata. In: Models, Algorithms, Logics and Tools, LNCS 10460. Springer (2017)
35. Kwiatkowska, M., Norman, G., Parker, D.: Verification and control of turn-based probabilistic real-time games (2019). arXiv:1906.09142
36. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Automated verification of concurrent stochastic games. In: Proc. QEST'18, LNCS 11024. Springer (2018)
37. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Equilibria-based probabilistic model checking for concurrent stochastic games. In: Proc. FM'19, LNCS. Springer (2019). To appear
38. Kwiatkowska, M., Norman, G., Parker, D., Sproston, J.: Performance analysis of probabilistic timed automata using digital clocks. FMSD **29** (2006)
39. Kwiatkowska, M., Norman, G., Segala, R., Sproston, J.: Automatic verification of real-time systems with discrete probability distributions. TCS **282** (2002)
40. Kwiatkowska, M., Norman, G., Sproston, J., Wang, F.: Symbolic model checking for probabilistic timed automata. IC **205**(7) (2007)
41. Kwiatkowska, M., Parker, D., Wiltsche, C.: PRISM-games: Verification and strategy synthesis for stochastic multi-player games with multiple objectives. STTT **20**(2) (2018)
42. Lanotte, R., Maggiolo-Schettini, A., Troina, A.: Automatic analysis of a non-repudiation protocol. In: Proc. QAPL'04, ENTCS 112 (2005)
43. Maler, O., Pnueli, A., Sifakis, J.: On the synthesis of discrete controllers for timed systems. In: Proc. STACS'95, LNCS 900. Springer (1995)
44. Markowitch, O., Roggeman, Y.: Probabilistic non-repudiation without trusted third party. In: Proc. Workshop Security in Communication Networks (1999)
45. Norman, G., Parker, D., Sproston, J.: Model checking for probabilistic timed automata. FMSD **43**(2) (2013)
46. Norman, G., Parker, D., Zou, X.: Verification and control of partially observable probabilistic systems. RTS **53**(3) (2017)
47. Oualhadj, Y., Reynier, P.A., Sankur, O.: Probabilistic robust timed games. In: Proc. CONCUR'14, LNCS 8704. Springer (2014)
48. Rudin, W.: Principles of Mathematical Analysis, 3rd edn. McGraw-Hill (1976)
49. S. La Torre, S., Mukhopadhyay, S., Murano, A.: Optimal-reachability and control for acyclic weighted timed automata. In: Proc. TCS'02. Kluwer (2002)
50. Tripakis, S.: Verifying progress in timed systems. In: Proc. ARTS'99, LNCS 1601. Springer (1999)
51. Tripakis, S., Altisen, K.: On-the-fly controller synthesis for discrete and dense-time systems. In: Proc. FM'99, LNCS 1708. Springer (1999)
52. Tripakis, S., Yovine, S., Bouajjan, A.: Checking timed Büchi automata emptiness efficiently. FMSD **26**(3) (2005)
53. Supporting material. www.prismmodelchecker.org/files/tptgs/