Reachability in Stochastic Timed Games

Patricia Bouyer[*] and Vojtěch Forejt[**]

[1] LSV, CNRS & ENS Cachan, France
      bouyer@lsv.ens-cachan.fr
[2] Masaryk University, Brno, Czech Republic
      forejt@fi.muni.cz

Abstract. We define stochastic timed games, which extend two-player timed games with probabilities (following a recent approach by Baier *et al*), and which extend in a natural way continuous-time Markov decision processes. We focus on the reachability problem for these games, and ask whether one of the players has a strategy to ensure that the probability of reaching a fixed set of states is equal to (or below, resp. above) a certain number r, whatever the second player does. We show that the problem is undecidable in general, but that it becomes decidable if we restrict to single-clock $1\frac{1}{2}$-player games and ask whether the player can ensure that the probability of reaching the set is $=1$ (or >0, $=0$).

1 Introduction

Timed systems. Timed automata [1] are a well-established formalism for the modelling and analysis of timed systems. A timed automaton is roughly a finite-state automaton enriched with clocks and clock constraints. This model has been extensively studied, and several verification tools have been developed. To represent interactive or open systems, the model of timed games has been proposed [2], where the system interacts with its environment, and the aim is to build a controller that will guide the system, so that it never violates its specification, whatever are the actions of the environment.

Adding probabilities to timed automata. In [4, 3], a purely probabilistic semantics has been given to timed automata, in which both delays and discrete choices are randomized. The initial motivation of the previous works was not to define a model with real-time and probabilistic features, but rather to propose an alternative semantics to timed automata, following the long-running implementability and robustness paradigm [12, 17, 9]. The idea is that unlikely behaviours should not interfere with the validity of a formula in a timed automaton, and the probabilistic semantics has been proposed to provide a way of measuring the 'size' of sets of runs in a timed automaton. In this context, natural model-checking questions have been considered: (i) 'Does the automaton *almost-surely* satisfy a given ω-regular property?', and (ii) 'Does the automaton satisfy

a given ω-regular property with probability at least p?'. The first problem is decidable for single-clock timed automata [3], but it is open for general timed automata. The second problem is decidable for a subclass of single-clock timed automata [6].

If we consider probabilities no more as a way of providing an alternative semantics to timed automata but rather as part of the model itself, the purely stochastic model defined in [3] can be viewed as an extension of continuous-time Markov chains (CTMCs in short), which have been extensively studied, both by mathematicians [11] and by computer scientists for their role in verification [5, 13].

Stochastic timed games. In real-life systems, pure stochastic models might not be sufficient, and non-determinism and even interaction with an environment might be crucial features (we might think of communication protocols, where messages can be lost, and response delays are probabilistic). In the same way continuous-time Markov decision processes extend CTMCs, we can extend the purely stochastic model of [3] with non-determinism, and even with interaction.

In this paper, we propose the model of *stochastic timed games*, which somehow extend classical timed games with probabilities. In this model, some locations are probabilistic (in some context we could say they represent the nature), and the other locations belong either to player $\Diamond$ or to player $\Box$. We call these locations respectively $\bigcirc$-locations, $\Diamond$-locations, and $\Box$-locations. Following classical terminology in stochastic finite games [8] where the nature is viewed as half a player, those games will be called $2\frac{1}{2}$-*player timed games*, and stochastic timed games with no $\Box$-locations will be called $1\frac{1}{2}$-*player timed games*. Finally, the purely stochastic model of [3] can then be called the $\frac{1}{2}$-*player game model* (there are no $\Diamond$-locations nor $\Box$-locations).

We assume a stochastic timed game is given, and we play the game as follows. At $\Diamond$-locations, player $\Diamond$ chooses the next move (delay and transitions to be taken), at $\Box$-locations, player $\Box$ chooses the next move, and at $\bigcirc$-locations, the environment is purely probabilistic (and the probability laws on delays and transitions are given in the description of the model). Moves for the two players are given by (deterministic) strategies, and given two strategies $\lambda_\Diamond$ (for player $\Diamond$) and $\lambda_\Box$ (for player $\Box$), the play of the game is a probability distribution over the set of runs of the timed automaton. Some natural questions can then be posed:

Qualitative questions: given $r \in \{0, 1\}$, is there a strategy for player $\Diamond$ such that for every strategy for player $\Box$, the probability (under those strategies) of satisfying some reachability property is equal to (resp. less than, resp. more than...) r?

Quantitative questions: given $r \in (0, 1)$, is there a strategy for player $\Diamond$ such that for every strategy for player $\Box$, the probability (under those strategies) of satisfying some reachability property is equal to (resp. no less than, resp. no more than...) r?

On that model, only restricted results have been proven so far, and they only concern the $\frac{1}{2}$-player case: all qualitative questions can be decided (in NLOGSPACE) if we restrict to single-clock models [3], and under a further restriction on the way probabilities are assigned to delays, all quantitative questions can be decided [6].

Our contribution. In this paper, we show the two following results:

– For $1\frac{1}{2}$-player games *with a single clock*, the qualitative questions 'equal to 0' or 'equal to 1' can be solved in PTIME, matching the known PTIME-hardness in

classical Markov decision processes [16], and the qualitative question 'larger than 0' can be solved in NLOGSPACE, matching the NLOGSPACE-hardness of the reachability in finite graphs;
– For $2\frac{1}{2}$-player games, the quantitative questions are undecidable. We will make precise in the core of the paper the classes of models for which this result holds.

2 Definitions

Timed automata. We assume the classical notions of clocks, clock valuations and guards are familiar to the reader [1]. We write $\mathcal{G}(X)$ for the set of diagonal-free guards over set of clocks X. A *timed automaton* is a tuple $\mathcal{A} = (L, X, E, \mathcal{I})$ such that: (i) L is a finite set of locations, (ii) X is a finite set of clocks, (iii) $E \subseteq L \times \mathcal{G}(X) \times 2^X \times L$ is a finite set of edges, and (iv) $\mathcal{I} : L \to \mathcal{G}(X)$ assigns an invariant to each location. A *state* s of such a timed automaton is a pair $(\ell, v) \in L \times (\mathbb{R}_+)^{|X|}$ (where v is a clock valuation). If $s = (\ell, v)$ is a state and $t \in \mathbb{R}_+$, we write $s + t$ for the state $(\ell, v + t)$. We say that there is a *transition* (t, e) from state $s = (\ell, v)$ to state $s' = (\ell', v')$, we then write $s \xrightarrow{t,e} s'$, if $e = (\ell, g, Y, \ell') \in E$ is such that (i) $v + t \models g$, (ii) for every $0 \leq t' \leq t$, $v + t' \models \mathcal{I}(\ell)$, (iii) $v' = [Y \leftarrow 0](v + t)$, and (iv) $v' \models \mathcal{I}(\ell')$. A *run* in $\mathcal{A}$ is a finite or infinite sequence $\varrho = s_0 \xrightarrow{t_1, e_1} s_1 \xrightarrow{t_2, e_2} s_2 \cdots$ of states and transitions. An edge e is *enabled* in state s whenever there is a state s' such that $s \xrightarrow{0, e} s'$. Given s a state of $\mathcal{A}$ and e an edge, we define $I(s, e) = \{t \in \mathbb{R}_+ \mid s \xrightarrow{t,e} s' \text{ for some } s'\}$ and $I(s) = \bigcup_{e \in E} I(s, e)$. The automaton $\mathcal{A}$ is *non-blocking* if for all states s, $I(s) \neq \emptyset$. For the sake of simplicity, we assume that timed automata are non-blocking.

Stochastic timed games. A *stochastic timed game* is a tuple $\mathcal{G} = (\mathcal{A}, (L_\Diamond, L_\Box, L_\bigcirc), w, \mu)$ where $\mathcal{A} = (L, X, E, \mathcal{I})$ is a timed automaton, $(L_\Diamond, L_\Box, L_\bigcirc)$ is a partition of L into the locations controlled by player $\Diamond, \Box$ and $\bigcirc$, respectively, w is a function that assigns to each edge leaving a location in $L_\bigcirc$ a positive (integral) *weight*, and μ is a function that assigns to each state $s \in L_\bigcirc \times (\mathbb{R}_+)^{|X|}$ a measure over $I(s)$, such that for all such s, $\mu(s)$ satisfies the following requirements:

1. $\mu(s)(I(s)) = 1$;
2. We write χ for the Lebesgue measure. If $\chi(I(s)) > 0$, $\mu(s)$ is equivalent[3] to χ. Furthermore, the choice of the measures should not be too erratic and those measures should evolve smoothly when moving states. We thus require that for every $a < b$, for every s, there is some $\varepsilon > 0$ such that $\mu(s + \delta)((a - \delta, b - \delta))$ is lower-bounded by ε on the set $\{\delta \in \mathbb{R}_+ \mid (a - \delta, b - \delta) \subseteq I(s + \delta)\}$. If $\chi(I(s)) = 0$, the set $I(s)$ is finite, and $\mu(s)$ must be equivalent to the uniform distribution over points of $I(s)$.

Note that these conditions are required, see [3], but can be easily satisfied. For instance, a timed automaton with uniform distributions on bounded sets and with exponential distributions on unbounded intervals (with a smoothly varying rate, see [10]) satisfies these conditions. Also note that we impose no requirements on the representation of the measures. All our results hold regardless of the representation.

---

[3] Measures χ_1 and χ_2 are *equivalent* if for all measurable sets A, $\chi_1(A) = 0 \Leftrightarrow \chi_2(A) = 0$.

In the following, we will say that a timed automaton is *equipped with uniform distributions over delays* if for every state s, $I(s)$ is bounded, and $\mu(s)$ is the uniform distribution over $I(s)$. We will say that the automaton is *equipped with exponential distributions over delays* whenever, for every s, either $I(s)$ has zero Lebesgue measure, or $I(s) = \mathbb{R}_+$ and for every location ℓ, there is a positive rational number α_ℓ such that

$$\mu(s)(I) = \int_{t \in I} \alpha_\ell \cdot e^{-\alpha_\ell t} \, \mathrm{d}t.$$

Intuitively, in a stochastic game, locations in $L_\Diamond$ (resp. $L_\Box$) are controlled by player $\Diamond$ (resp. player $\Box$), whereas locations in $L_\bigcirc$ belong to the environment and behaviours from those locations are governed by probabilistic laws. Indeed, in these locations, both delays and discrete moves will be chosen probabilistically: from s, a delay t is chosen following the *probability distribution over delays* $\mu(s)$; Then, from state $s + t$ an enabled edge is selected following a discrete probability distribution that is given in a usual way with the weight function w: in state $s + t$, the probability of edge e (if enabled) is $w(e)/\left(\sum_{e'}\{w(e') \mid e' \text{ enabled in } s+t\}\right)$. This way of probabilizing behaviours in timed automata has been presented in [4, 3], where all locations were supposed to be probabilistic. We now formalize the *stochastic process* that is defined by a stochastic game, when fixing strategies for the two players.

A *strategy* for player $\Diamond$ (resp. player $\Box$) is a function that assigns to every finite run $\varrho = (\ell_0, v_0) \xrightarrow{t_1, e_1} \ldots \xrightarrow{t_n, e_n} (\ell_n, v_n)$ with $\ell_n \in L_\Diamond$ (resp. $\ell_n \in L_\Box$) a pair $(t, e) \in \mathbb{R}_+ \times E$ such that $(\ell_n, v_n) \xrightarrow{t, e} (\ell, v)$ for some (ℓ, v). In order to later be able to measure probabilities of certain sets of runs, we impose the following additional measurability condition on the strategy λ: for every finite sequence of edges $e_1, \ldots, e_n$ and every state s, the function $\kappa : (t_1, \ldots, t_n) \mapsto (t, e)$ such that $\kappa(t_1, \ldots, t_n) = (t, e)$ iff $\lambda(s \xrightarrow{t_1, e_1} s_1 \ldots \xrightarrow{t_n, e_n} s_n) = (t, e)$ is measurable.[4]

A *strategy profile* is a pair $\Lambda = (\lambda_\Diamond, \lambda_\Box)$ where $\lambda_\Diamond$ and $\lambda_\Box$ are strategies for players $\Diamond$ and $\Box$ respectively. Given a stochastic timed game $\mathcal{G}$, a finite run ϱ ending in a state s_0 and a strategy profile $\Lambda = (\lambda_\Diamond, \lambda_\Box)$, we define $Run(\mathcal{G}, \varrho, \Lambda)$ (resp. $Run^\omega(\mathcal{G}, \varrho, \Lambda)$) to be the set of all finite (resp. infinite) runs generated by $\lambda_\Diamond$ and $\lambda_\Box$ after prefix ϱ, *i.e.*, the set of all runs $s_0 \xrightarrow{t_1, e_1} s_1 \xrightarrow{t_2, e_2} \cdots$ in the underlying automaton satisfying the following condition: if $s_i = (\ell, v)$ and $\ell \in L_\Diamond$ (resp. $\ell \in L_\Box$), then $\lambda_\Diamond$ (resp. $\lambda_\Box$) returns (t_{i+1}, e_{i+1}) when applied to $\varrho \xrightarrow{t_1, e_1} s_1 \xrightarrow{t_2, e_2} \ldots \xrightarrow{t_i, e_i} s_i$. Moreover, given a finite sequence of edges $e_1, \ldots, e_n$, we define the *symbolic path* $\pi_\Lambda(\varrho, e_1 \ldots e_n)$ by

$$\pi_\Lambda(\varrho, e_1 \ldots e_n) = \{\varrho' \in Run(\mathcal{G}, \varrho, \Lambda) \mid \varrho' = s_0 \xrightarrow{t_1, e_1} \cdots \xrightarrow{t_n, e_n} s_n, \ t_i \in \mathbb{R}_+\}$$

When Λ is clear from the context, we simply write $\pi(\varrho, e_1 \ldots e_n)$.

We extend the definitions of [3] to stochastic games, and define, given a strategy profile $\Lambda = (\lambda_\Diamond, \lambda_\Box)$ and a finite run ϱ ending in $s = (\ell, v)$, a measure $\mathcal{P}_\Lambda$ over the set $Run(\mathcal{G}, \varrho, \Lambda)$. To that aim, we define $\mathcal{P}_\Lambda$ on symbolic paths initiated in ϱ by $\mathcal{P}_\Lambda(\pi(\varrho)) = 1$ and then inductively as follows:

---

[4] For the purpose of this definition, we define the measurable space on the domain (and codomain) as a product space of measurable spaces of its components (where for real numbers and edges we take the σ-algebra generated by intervals and by set of edges, respectively).

- If $\ell \in L_\Diamond$ (resp. $\ell \in L_\Box$) and $\lambda_\Diamond(\varrho) = (t, e)$ (resp. $\lambda_\Box(\varrho) = (t, e)$), we set

$$\mathcal{P}_\Lambda(\pi(\varrho, e_1 \ldots e_n)) = \begin{cases} 0 & \text{if } e_1 \neq e \\ \mathcal{P}_\Lambda(\pi(\varrho \xrightarrow{t,e} s', e_2 \ldots e_n)) & \text{otherwise (where } s \xrightarrow{t,e} s') \end{cases}$$

- If $\ell \in L_\bigcirc$, we define

$$\mathcal{P}_\Lambda(\pi(\varrho, e_1 \ldots e_n)) = \int_{t \in I(s, e_1)} p(s + t)(e_1) \cdot \mathcal{P}_\Lambda(\pi(\varrho \xrightarrow{t,e_1} s^{t,e_1}, e_2 \ldots e_n)) \, \mathrm{d}\mu(s)(t)$$

where $s \xrightarrow{t,e_1} s^{t,e_1}$ for every $t \in I(s, e_1)$.

These integrals are properly defined thanks to the measurability condition we impose on strategies, and thanks to Fubini's Theorem [19].

Following [3], it is not difficult to see that, given a measurable constraint $\mathcal{C}$ of $\mathbb{R}^n_+$, we can extend this definition to *constrained symbolic paths* $\pi^{\mathcal{C}}_\Lambda(\varrho, e_1 \ldots e_n)$, where $\pi^{\mathcal{C}}_\Lambda(\varrho, e_1 \ldots e_n) = \{\varrho' \in Run(\mathcal{G}, \varrho, \Lambda) \mid \varrho' = s_0 \xrightarrow{t_1, e_1} \cdots \xrightarrow{t_n, e_n} s_n$ and $(t_1, \ldots, t_n) \models \mathcal{C}\}$. We now consider the *cylinder* generated by a constrained symbolic path: an infinite run ϱ'' is in the cylinder generated by $\pi^{\mathcal{C}}_\Lambda(\varrho, e_1 \ldots e_n)$, denoted $\mathsf{Cyl}(\pi^{\mathcal{C}}_\Lambda(\varrho, e_1 \ldots e_n))$, if $\varrho \in Run^\omega(\mathcal{G}, \varrho, \Lambda)$ and there exists $\varrho' \in \pi^{\mathcal{C}}_\Lambda(\varrho, e_1 \ldots e_n)$ which is a prefix of ϱ''. We extend $\mathcal{P}_\Lambda$ to those cylinders in a natural way: $\mathcal{P}_\Lambda(\mathsf{Cyl}(\pi^{\mathcal{C}}_\Lambda(\varrho, e_1 \ldots e_n))) = \mathcal{P}_\Lambda(\pi^{\mathcal{C}}_\Lambda(\varrho, e_1 \ldots e_n))$, and then in a unique way to the σ-algebra Ω^ϱ_Λ generated by those cylinders. Following [3], we can prove the following correctness lemma.

Lemma 1. *Let $\mathcal{G}$ be a stochastic timed game. For every strategy profile Λ, for every finite run ϱ, $\mathcal{P}_\Lambda$ is a probability measure over $(Run^\omega(\mathcal{G}, \varrho, \Lambda), \Omega^\varrho_\Lambda)$.*

Example 2. Consider the following game $\mathcal{G}$:
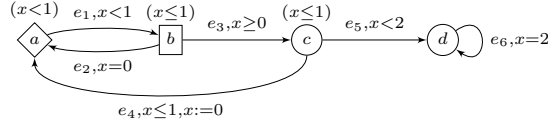
Suppose the game is equipped with uniform distributions over delays and over edges, and consider a strategy profile $\Lambda = (\lambda_\Diamond, \lambda_\Box)$ such that strategy $\lambda_\Diamond$ assigns $(0.5, e_1)$ to each run ϱ ending in state (a, v) if $v \leq 0.5$ and $(0, e_1)$ otherwise, and such that strategy $\lambda_\Box$ assigns $(0, e_3)$ to each run ending in (b, v). If $\varrho = (a, 0) \xrightarrow{0.5, e_1} (b, 0.5) \xrightarrow{0, e_3} (c, 0.5)$, $\mathcal{P}_\Lambda(\pi(\varrho, e_4 e_1 e_3 e_4)) = \frac{1}{36}$. $\lrcorner$

The reachability problem. In this paper we study *the reachability problem* for stochastic games, which is stated as follows. Given a game $\mathcal{G}$, an initial state s, a set of locations A, a comparison operator $\sim \in \{<, \leq, =, \geq, >\}$ and a rational number $r \in [0, 1]$, decide whether there is a strategy $\lambda_\Diamond$ for player $\Diamond$, such that for every strategy $\lambda_\Box$ for player $\Box$, if $\Lambda = (\lambda_\Diamond, \lambda_\Box)$, $\mathcal{P}_\Lambda\{\varrho \in Run(\mathcal{G}, s, \Lambda) \mid \varrho \text{ visits } A\} \sim r$. In that case, we say that $\lambda_\Diamond$ is a *winning strategy* from s for the *reachability objective* $\mathsf{Reach}_{\sim r}(A)$.

A special case is when $r \in \{0, 1\}$, and the problem is called the *qualitative reachability problem*. In all other cases, we speak of the *quantitative reachability problem*.

Example 3. Consider again the stochastic timed game $\mathcal{G}$ of Example 2 together with the qualitative reachability objective $\mathsf{Reach}_{=1}(\{d\})$. Player $\Diamond$ has a winning strategy $\lambda_\Diamond$ for that objective from state $(a, 0)$, which is defined as follows: $\lambda_\Diamond(\varrho) = (0.5, e_1)$ for all runs ϱ ending in state $(a, 0)$. On the other hand, player $\Diamond$ has no winning strategy from $(a, 0)$ for the quantitative objective $\mathsf{Reach}_{<0.9}(\{c\})$. ⌟

The region automaton abstraction. The well-known region automaton construction is a finite abstraction of timed automata which can be used for verifying many properties like ω-regular untimed properties [1]. In this paper, we will only use this abstraction in the context of single-clock timed automata, where the original abstraction can be slightly improved [14]. Furthermore, we will still interpret this abstraction as a timed automaton, as it is done in [3].

Let $\mathcal{A}$ be a single-clock timed automaton, and $\Gamma = \{0 = \gamma_0 < \gamma_1 < \cdots < \gamma_p\}$ be the set of constants that appear in $\mathcal{A}$ (plus the constant 0). We define the set $R_\mathcal{A}$ of regions in $\mathcal{A}$ as the set of intervals of the form $[\gamma_i; \gamma_i]$ (with $0 \leq i \leq p$), or $(\gamma_{i-1}; \gamma_i)$ (with $1 \leq i \leq p$) or $(\gamma_p; +\infty)$. Assuming $\mathcal{A} = (L, \{x\}, E, \mathcal{I})$, the *region automaton* of $\mathcal{A}$ is the timed automaton $\mathcal{R}(\mathcal{A}) = (Q, \{x\}, T, \kappa)$ such that $Q = L \times R_\mathcal{A}$, $\kappa((\ell, r)) = \mathcal{I}(\ell)$, and $T \subseteq (Q \times R_\mathcal{A} \times E \times 2^X \times Q)$ is such that $(\ell, r) \xrightarrow{r'', e, Y} (\ell', r')$ is in T iff there exists $e \in E$, $v \in r$, $\tau \in \mathbb{R}_+$ such that $v + \tau \in r''$, $(\ell, v) \xrightarrow{\tau, e} (\ell', v')$, and $v' \in r'$.

In the case of single-clock timed automata, the above automaton $\mathcal{R}(\mathcal{A})$ has size polynomial in the size of $\mathcal{A}$ (the number of regions is polynomial), and the reachability problem is indeed NLOGSPACE-complete in single-clock timed automata [14]. In the following, we will assume w.l.o.g. that timed automata are given in their region automaton form. Hence, to every location of this automaton will be associated a single region in which the valuation will be when arriving in that location.

3 Qualitative reachability in single-clock $1\frac{1}{2}$-player games

In this section we restrict to single-clock $1\frac{1}{2}$-player games, *i.e.*, stochastic games with a single clock, and with no locations for player $\Box$. Furthermore, we focus on the qualitative reachability problems.

Optimal strategies may not exist. Indeed, it may be the case that for every $\varepsilon > 0$, there is a strategy achieving the (reachability) objective with probability at least $1 - \varepsilon$ (resp. at most ε), while there is no strategy achieving the objective with probability 1 (resp. 0). In this case, we speak about ε-*optimal strategies*. For instance, consider the following game, where we assume uniform distributions over delays.
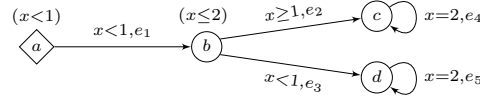
Assuming that the objective is to reach location c (resp. d) from $(a, 0)$, one can check that by taking the edge e_1 close enough to time 1, the probability of reaching c (resp. d)

can be arbitrary close to 1 (resp. 0), while there is no strategy that could ensure reaching c (resp. d) with probability 1 (resp. 0).

In this paper we will ask whether there are strategies that precisely achieve a qualitative objective (like equal to 1, or equal to 0), and we leave for future work the interesting but difficult question whether we can approximate arbitrarily these objectives or not.

Decidability of the existence of optimal strategies. We now turn to one of the two main theorems of this paper, whose proof will be developed.

Theorem 4. *Given a single-clock $1\frac{1}{2}$-player timed game $\mathcal{G}$, $s = (\ell, 0)$ a state of $\mathcal{G}$, and A a set of locations of $\mathcal{G}$, we can decide in PTIME whether there is a strategy achieving the objective $\mathsf{Reach}_{=1}(A)$ (or $\mathsf{Reach}_{=0}(A)$). We can decide in NLOGSPACE whether there is a strategy achieving the objective $\mathsf{Reach}_{>0}(A)$. These complexity upper bounds are furthermore optimal.*

For the rest of the section, we assume that $\mathcal{G}$ is a single-clock $1\frac{1}{2}$-player timed game with the underlying automaton being a region automaton. We also fix a set A of locations. Computing winning states for the objectives $\mathsf{Reach}_{=0}(A)$ and $\mathsf{Reach}_{>0}(A)$ can be performed using a simple fixpoint algorihm (in fact, the problem can be reduced to the similar problem for discrete-time Markov decision processes [18]). [5]

Proposition 5. *We can compute in PTIME (resp. NLOGSPACE) the set of states from which player $\Diamond$ has a strategy to achieve the objective $\mathsf{Reach}_{=0}(A)$ (resp. $\mathsf{Reach}_{>0}(A)$). Furthermore, this set of states is closed by region (i.e., if (ℓ, v) is winning, then for every v' in the same region as v, (ℓ, v') is winning).*

The case of the objective $\mathsf{Reach}_{=1}(A)$ requires more involved developments, but a proposition identical to the previous one can however be stated.

Proposition 6. *We can compute in PTIME the set of states from which player $\Diamond$ has a strategy to achieve the objective $\mathsf{Reach}_{=1}(A)$. Furthermore, this set of states is closed by region.*

The restriction to single-clock games yields the following important property: resetting the unique clock somehow resets the history of the game, the target state is then uniquely determined by the target location. Hence, we will first focus on games where the clock is never reset, and then decompose the global game w.r.t. the resetting transitions and solve the different non-resetting parts separately and glue everything together.

We first focus on games without any resets, and consider a more complex objective: given two sets of locations A and B such that $B \subseteq A$, we say that the strategy λ achieves the objective $\mathsf{ExtReach}(A, B)$ if it achieves both $\mathsf{Reach}_{=1}(A)$ and $\mathsf{Reach}_{>0}(B)$. We can prove (using another fixpoint algorithm):

---

[5] All propositions in this section make use of the fact that we can remove w.l.o.g. certain "negligible" edges from the game effectively. An edge e of $\mathcal{G}$ is said to be *negligible* if it starts from some $\bigcirc$-location ℓ and if it is constrained by some punctual constraint, whereas there is another edge leaving ℓ labelled with a non-punctual constraint.

Lemma 7. *We assume that the clock is never reset in $\mathcal{G}$. Let A and $B \subseteq A$ be two sets of locations of $\mathcal{G}$. We can compute in PTIME the set of states from which player $\Diamond$ has a strategy to achieve the objective $\mathsf{ExtReach}(A, B)$. Furthermore, this set of states is closed by region.*

Now we show how we can use Lemma 7 to solve the games for the objective $\mathsf{Reach}_{=1}(A)$. This lemma heavily relies on the specific properties of single-clock timed automata that we have mentioned earlier. Somehow to solve the objective $\mathsf{Reach}_{=1}(A)$, we will enforce moving from one resetting transition to another one, always progressing towards A. This is formalized as follows.

Lemma 8. *If ℓ_{in} is a location of $\mathcal{G}$, the following two statements are equivalent:*

1. *There is a strategy λ from $(\ell_{in}, 0)$ that achieves the objective $\mathsf{Reach}_{=1}(A)$.*
2. *Writing L_0 for the set of locations which are targets of resetting transitions (w.l.o.g. we assume $\ell_{in} \in L_0$ and $A \subseteq L_0$), there is a set $R \subseteq L_0 \times 2^{L_0} \times L_0$ such that:*
   (a) *There is $(\ell_{in}, S, k) \in R$ for some $S \subseteq L_0$ and $k \in S$;*
   (b) *Whenever $\ell \in S \setminus A$ for some $(\ell', S, k) \in R$, then $(\ell, S', k') \in R$ for some $S' \subseteq L_0$ and $k' \in S'$;*
   (c) *For each $(\ell, S, k) \in R$, there is a strategy that achieves $\mathsf{ExtReach}(S, \{k\})$ from $(\ell, 0)$ without resetting the clock (except for the last move to S);*
   (d) *For each $(\ell, S, k) \in R$, there is a sequence $k_1 k_2 \ldots k_n$ such that $k_1 = \ell$, $k_n \in A$, and for every $1 \le i < n$, there exist $S_{i+1} \subseteq L_0$ and $k_{i+1} \in S_{i+1}$ such that $(k_i, S_{i+1}, k_{i+1}) \in R$.*

Moreover, the set R has polynomial size and can be computed in polynomial time.

We define some vocabulary before turning to the proof. If such an above relation R exists, we write $L_R = \{\ell \in L_0 \mid \exists S \subseteq L_0 \text{ and } k \in S \text{ s.t. } (\ell, S, k) \in R\}$. For every $\ell \in L_R$, we call the *distance to A* from ℓ the smallest integer n such that there is a chain leading to A, as in condition *2d*. For every $\ell \in L_R$, the distance to A is a natural number. Furthermore, for every $\ell \in L_R$, there is $(\ell, S, k) \in R$ such that the distance to A from k is (strictly) smaller than the distance to A from ℓ.

Proof (sketch). We only justify the implication *2. $\Rightarrow$ 1.*, which gives a good intuition for the construction. We start by fixing some $(\ell, S, k) \in R$ for every $\ell \in L_R$ such that the distance to A from k is (strictly) smaller than the distance to A from ℓ. Let λ_ℓ be a (fixed) strategy that achieves the objective $\mathsf{ExtReach}(S, \{k\})$ from state $(\ell, 0)$. From these strategies we construct a strategy λ that achieves $\mathsf{Reach}_{=1}(A)$ from $(\ell_{in}, 0)$ as follows.

We let $\varrho = s_0 \xrightarrow{t_1, e_1} s_1 \xrightarrow{t_2, e_2} \cdots \xrightarrow{t_{n-1}, e_{n-1}} s_n$ be a finite run in $\mathcal{G}$, and set ϱ' as the longest suffix of ϱ which does not reset the clock. ϱ' starts in some state $s_i = (\ell_i, 0)$. If $\ell_i \in L_R$, we define $\lambda(\varrho)$ as $\lambda_{\ell_i, S_i, k_i}(\varrho')$, and otherwise we define it in an arbitrary manner (but the set of runs for which we will need to define the strategy in an arbitrary manner has probability 0). The intuitive meaning of the definition is depicted in the following picture.
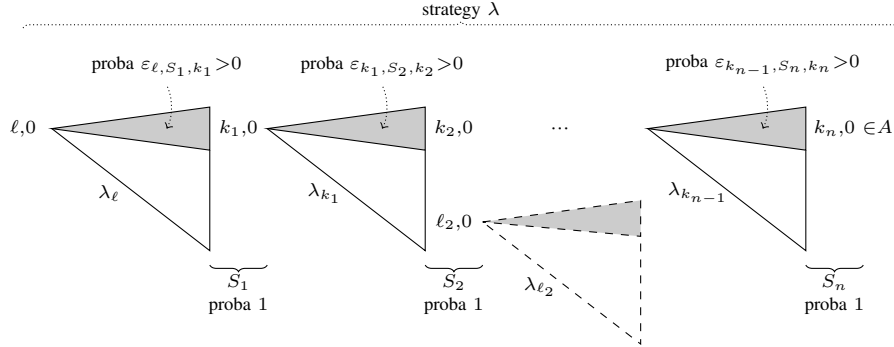
It can be proven using standard tools of probability theory that λ achieves the objective. Indeed, if $\ell \in L_R$ and (ℓ, S, k) has been selected, from state $(\ell, 0)$, runs generated by λ almost-surely resets the clock, reaching states $(\ell', 0)$ with $\ell' \in S$, and with positive probability, say $\varepsilon_{\ell, S, k} > 0$, reach state $(k, 0)$. Furthermore, the distance to A from k is smaller than that from ℓ. Now, due to condition *2d*, the probability to reach A from $(\ell, 0)$ is at least the positive product $\varepsilon_{\ell, S_1, k_1} \cdot \varepsilon_{k_1, S_2, k_2} \cdots \varepsilon_{k_{n-1}, S_n, k_n}$. Hence, there exists $\varepsilon > 0$ such that the probability to reach A from any $(\ell, 0)$ (such that there is some $(\ell, S, k) \in R$) is at least ε. We can now conclude, by saying that from any $(\ell, 0)$ with $\ell \in L_R$, the probability to reach $\{(\ell', 0) \mid \ell' \in L_R\}$ is 1. Hence, with probability 1 we reach A under strategy λ. ⌟

4 Quantitative reachability in $2\frac{1}{2}$-player games

In this section we present the second main theorem of this paper.

Theorem 9. *Given a $2\frac{1}{2}$-player timed game $\mathcal{G}$, $s = (\ell, 0)$ a state of $\mathcal{G}$, and A a set of locations of $\mathcal{G}$, we cannot decide whether there is a winning strategy from s for achieving the objective* $\mathsf{Reach}_{\sim \frac{1}{2}}(A)$ *(for $\sim \in \{<, \leq, =, \geq, >\}$). This result holds for games with three clocks.*

We will prove this theorem by reduction from the halting problem for two-counter machines. The reduction has been inspired by recent developments in weighted timed systems [7]. A *two-counter machine* $\mathcal{M}$ is a finite set of labeled instructions $1{:}\mathsf{inst}_1, \ldots, n{-}1{:}\mathsf{inst}_{n-1}, n{:}\mathsf{stop}$ where each inst_i is of the form "$c_j := c_j + 1$; *goto* k" or "*if* $c_j = 0$ *then goto* k; *else* $c_j := c_j - 1$; *goto* ℓ". Here, $j \in \{1, 2\}$ and $k, \ell \in \{1, \ldots, n\}$). A configuration of $\mathcal{M}$ is a triple $[\mathsf{inst}_i, d_1, d_2]$ where inst_i is the instruction to be executed and $d_1, d_2 \in \mathbb{N}$ are the current values of the counters c_1 and c_2. A computational step $\rightsquigarrow$ between configurations is defined in the expected way. A computation is a (finite or infinite) sequence $\alpha_1, \alpha_2, \ldots$ where $\alpha_1 = [\mathsf{inst}_1, 0, 0]$ and for all i, $\alpha_i \rightsquigarrow \alpha_{i+1}$. A halting computation is a finite computation that ends in the instruction stop. The halting problem asks, given a two counter machine, whether there is a halting computation. This problem is known to be undecidable [15].

Let $\mathcal{M}$ be a two-counter machine. We construct a timed game $\mathcal{G}$ with three clocks and uniform distributions over delays, and a set of (black) locations A such that player

$\Diamond$ has a strategy to reach A with probability $\frac{1}{2}$ iff $\mathcal{M}$ has a halting computation. In $\mathcal{G}$, player $\Diamond$ will simulate a computation of $\mathcal{M}$ and the values of both counters will be represented as the value of one clock. More precisely, if the values of the two counters are p_1 and p_2 respectively, the correct representation will be $\frac{1}{2^{p_1} \cdot 3^{p_2}}$. At the same time, player $\Box$ will be allowed to check that the representation is correct and "faithful" (*i.e.* that if an instruction of $\mathcal{M}$ is simulated, then the value of the counter is changed appropriately). Due to the lack of space, we are unable to provide the whole construction here, we only give a brief insight by defining some gadgets that we use in the construction.

Note that in the gadgets, the letters x and y are clock variables. Later on, we will build up a game by instantiating the clock variables with real clocks. In all the gadgets, unless specified, the weight of each edge is 1 (so that when two edges are concurrently enabled, they are equally probable). From all states, the set of possible delays is bounded, hence we assume uniform distribution over delays everywhere. Finally, v_0 denotes the valuation assigning x_0 (resp. y_0, 0) to x (resp. y, u).

First, we define gadgets check_succ$_1(x, y)$ and check_succ$_2(x, y)$. These gadgets are used for testing whether the values of clocks x and y are of the form α and $\frac{\alpha}{2}$ for some $\alpha \in [0, 1]$ (in the case of check_succ$_1(x, y)$), or α and $\frac{\alpha}{3}$ for some $\alpha \in [0, 1]$ (in the case of check_succ$_2(x, y)$). We will later use these gadgets to check that an increment or a decrement of the counters has been faithfully made. The gadget check_succ$_1(x, y)$ has only probabilistic locations and has the following structure:
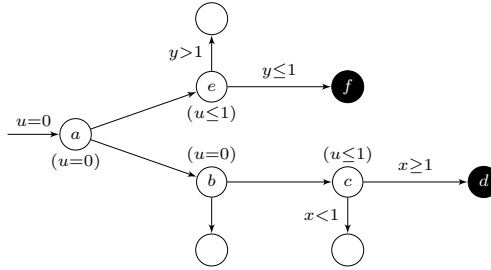

We claim that in gadget check_succ$_1(x, y)$, the probability of reaching the black locations from (a, v_0) is $\frac{1}{2}$ iff $x_0 = 2y_0$, because the probability of reaching one of the black locations is $\frac{1}{2}(1 - y_0) + \frac{1}{4}x_0$. The gadget check_succ$_2(x, y)$ can be created from check_succ$_1(x, y)$ by changing the weights of the edges.

Next, we define gadgets check_zero$_1(x, y)$ and check_zero$_2(x, y)$ that are used for testing that the value stored in clock x is $\frac{1}{3^p}$ for some $p \geq 0$ in the case of check_zero$_1(x, y)$, or $\frac{1}{2^p}$ for some $p \geq 0$ in the case of check_zero$_2(x, y)$. These gadgets will later be used to check that the value of the first or second counter is zero. The gadget check_zero$_1(x, y)$ has the following structure:
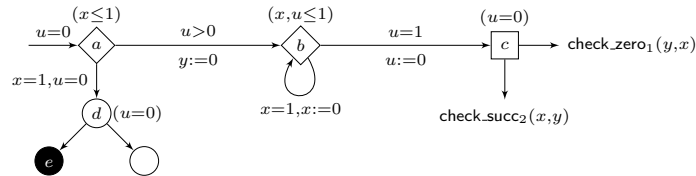
We claim that in the gadget $\mathsf{check\_zero}_1(x,y)$, player $\Diamond$ has a strategy from (a, v_0) for reaching the black locations with probability $\frac{1}{2}$ iff there is some integer $p \geq 0$ such that $x_0 = \frac{1}{3^p}$. The idea is that the value x_0 is of the required form iff it is possible to iteratively multiply its value by 2 until we eventually reach the value 1 (in which case we can take the edge down to d from a). The fact that we multiply by 2 is ensured by the gadget $\mathsf{check\_succ}_2(x,y)$. The gadget $\mathsf{check\_zero}_2(x,y)$ can be defined similarly and the precise definition is omitted here.

Finally for each instruction inst_i we create a gadget $g_i(x,y)$, that will simulate the instruction, with the encoding of the counters given earlier. For instance, if inst_i is of the form '$c_j := c_j + 1$; goto k', then $g_i(x,y)$ is of the form:
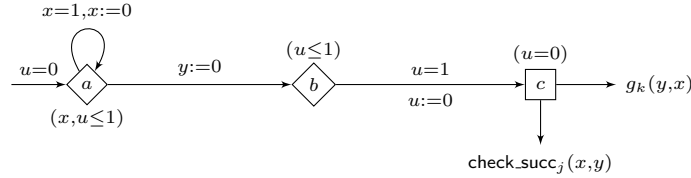
Player $\Diamond$ chooses at which time the transition from a to b is taken, and it should be such that the value of y when entering c is properly linked with the value of x so that the incrementation of counter c_j has been simulated. The gadget for a decrementation follows similar ideas, but is a bit more technical, that is why we omit it here.

Remark 10. The above reduction is for a reachability objective of the form $\mathsf{Reach}_{=\frac{1}{2}}(A)$. However, we can twist the construction and have the reachability objective $\mathsf{Reach}_{\sim\frac{1}{2}}(A)$ (for any $\sim\, \in \{<, \leq, \geq, >\}$). Also, the construction can be twisted to get the following further undecidability results:

1. The value $\frac{1}{2}$ in the previous construction was arbitrary, and the construction could be modified so that it would work for any rational number $r \in (0, 1)$.
2. Instead of assuming uniform distributions over delays, one can assume unbounded intervals and exponential distributions over delays: it only requires one extra clock in the reduction.

5   Conclusion

In this paper, we have defined stochastic timed games, an extension of two-player timed games with stochastic aspects. This $2\frac{1}{2}$-player model can also be viewed as an extension of continuous-time Markov decision processes, and also of the purely stochastic model proposed in [3]. On that model, we have considered the qualitative and quantitative reachability problems, and have proven that the qualitative reachability problem can be decided in single-clock $1\frac{1}{2}$-player model, whereas the quantitative reachability problem is undecidable in the (multi-clock) $2\frac{1}{2}$-player model. This leaves a wide range of open problems. Another challenge is the computation of approximate almost-surely winning strategies (that is for every $\varepsilon > 0$, a strategy for player $\Diamond$ which ensures the reachability objective with probability larger than $1 - \varepsilon$). Finally, more involved objectives (like ω-regular or parity objectives) should be explored in the context of $1\frac{1}{2}$-player timed games.

11

References

1. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
2. E. Asarin, O. Maler, A. Pnueli, and J. Sifakis. Controller synthesis for timed automata. In *Proc. IFAC Symposium on System Structure and Control*, pages 469–474. Elsevier Science, 1998.
3. C. Baier, N. Bertrand, P. Bouyer, T. Brihaye, and M. Größer. Almost-sure model checking of infinite paths in one-clock timed automata. In *Proc. 23rd Annual Symposium on Logic in Computer Science (LICS'08)*, pages 217–226. IEEE Computer Society Press, 2008.
4. C. Baier, N. Bertrand, P. Bouyer, Th. Brihaye, and M. Größer. Probabilistic and topological semantics for timed automata. In *Proc. 27th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'07)*, volume 4855 of *Lecture Notes in Computer Science*, pages 179–191. Springer, 2007.
5. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):524–541, 2003.
6. N. Bertrand, P. Bouyer, Th. Brihaye, and N. Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics. In *Proc. 5th International Conference on Quantitative Evaluation of Systems (QEST'08)*. IEEE Computer Society Press, 2008.
7. P. Bouyer, Th. Brihaye, and N. Markey. Improved undecidability results on weighted timed automata. *Information Processing Letters*, 98(5):188–194, 2006.
8. K. Chatterjee, M. Jurdziński, and Th. A. Henzinger. Simple stochastic parity games. In *Proc. 17th International Workshop on Computer Science Logic (CSL'03)*, volume 2803 of *Lecture Notes in Computer Science*, pages 100–113. Springer, 2003.
9. M. De Wulf, L. Doyen, N. Markey, and J.-F. Raskin. Robust safety of timed automata. *Formal Methods in System Design*, 2008. To appear.
10. J. Desharnais and P. Panangaden. Continuous stochastic logic characterizes bisimulation of continuous-time Markov processes. *Journal of Logic and Algebraic Programming*, 56:99–115, 2003.
11. W. Feller. *An Introduction to Probability Theory and Its Applications*. John Wiley & Sons, 1968.
12. V. Gupta, Th. A. Henzinger, and R. Jagadeesan. Robust timed automata. In *Proc. International Workshop on Hybrid and Real-Time Systems (HART'97)*, volume 1201 of *Lecture Notes in Computer Science*, pages 331–345. Springer, 1997.
13. H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle. A tool for model-checking Markov chains. *International Journal on Software Tools for Technology Transfer*, 4:153–172, 2003.
14. F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model checking timed automata with one or two clocks. In *Proc. 15th International Conference on Concurrency Theory (CONCUR'04)*, volume 3170 of *Lecture Notes in Computer Science*, pages 387–401. Springer, 2004.
15. M. Minsky. *Computation: Finite and Infinite Machines*. Prentice Hall International, 1967.
16. C. H. Papadimitriou and J. N. Tsitsiklis. On stochastic scheduling with in-tree precedence constraints. *SIAM Journal on Computing*, 16(1):1–6, 1987.
17. A. Puri. Dynamical properties of timed automata. In *Proc. 5th International Symposium on Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'98)*, volume 1486 of *Lecture Notes in Computer Science*, pages 210–227. Springer, 1998.
18. M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.
19. W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1966.