Stochastic Real-Time Games with Qualitative Timed Automata Objectives*

Tomáš Brázdil, Jan Krčál, Jan Křetínský**, Antonín Kučera, and Vojtěch Řehák

Faculty of Informatics, Masaryk University, Brno, Czech Republic
{brazdil, krcal, kucera, rehak}@fi.muni.cz, jan.kretinsky@in.tum.de

Abstract. We consider two-player stochastic games over real-time probabilistic processes where the winning objective is specified by a timed automaton. The goal of player □ is to play in such a way that the play (a timed word) is accepted by the timed automaton with probability one. Player ◇ aims at the opposite. We prove that whenever player □ has a winning strategy, then she also has a strategy that can be specified by a timed automaton. The strategy automaton reads the history of a play, and the decisions taken by the strategy depend only on the region of the resulting configuration. We also give an exponential-time algorithm which computes a winning timed automaton strategy if it exists.

1 Introduction

In this paper, we study *stochastic real-time games (SRTGs)* which are obtained as a natural game-theoretic extension of *generalized semi-Markov processes (GSMP)* [13, 20, 21] or *real-time probabilistic processes (RTP)* [2]. Intuitively, all of these formalisms model systems which react to certain *events*, such as message receipts, subsystem failures, timeouts, etc. A common characteristic of all events is that they are *delayed* (it takes some time before an initiated event actually occurs) and *concurrent* (there can be several previously initiated events that are currently awaited). For example, if two messages e and e' are sent, it takes some (random) time before they arrive, and one can specify, or approximate, the *densities* f_e, $f_{e'}$ of their arrival times. When e arrives (say, after 20 time units), the system reacts to this event by changing its state, and awaits e' in a new state. The arrival time of e' in the new state is measured from zero again, and its density $f_{e'|20}$ is obtained from $f_{e'}$ by incorporating the condition that e' is delayed for at least 20 time units. That is, $f_{e'|20}(x) = f_e(x + 20)/ \int_{20}^{\infty} f_e(y)\, dy$. Note that if the delays of all events are exponentially distributed, then $f_e = f_{e|b}$ for every $b \in \mathbb{R}_{\geq 0}$, and thus we obtain continuous-time Markov chains (see, e.g., [17]) and continuous-time stochastic games [10, 18] as restricted forms of RTPs and SRTGs, respectively.

Intuitively, a SRTG is a finite graph (see Fig. 1) with three types of nodes—*states* (drawn as large circles), *controls*, where each control can be either *internal* or *adversarial* (drawn as boxes and diamonds, respectively), and *actions* (drawn as small filled

* The authors are supported by the Alexander von Humboldt Foundation (T. Brázdil), the Institute for Theoretical Computer Science, project No. 1M0545 (J. Krčál), Brno Municipality (J. Křetínský), and the Czech Science Foundation, grants No. P202/10/1469 (A. Kučera) and No. 201/08/P459 (V. Řehák).

** On leave at TU München, Boltzmannstr. 3, Garching, Germany.

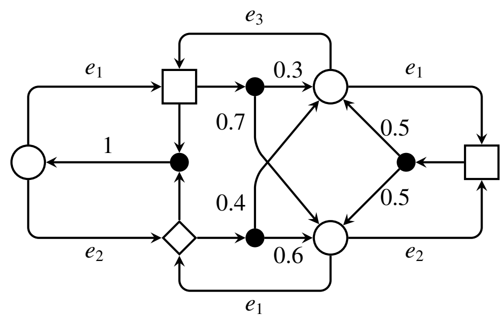


Fig. 1. An example of a stochastic real-time game

circles). In each state s, there is a finite subset $E(s)$ of events *scheduled* in s (the events scheduled in s are those which are "awaited" in a given state; the other events are disabled. Each state s can react to every event of $E(s)$ by entering a designated control c, where player $\Box$ or player $\Diamond$ chooses some of the available actions. Each action is associated with a fixed probability distribution over states. In general, both players can use randomized strategies, which means that they do not necessarily select just a single action but a probability distribution over the available actions, which is multiplied with the distributions associated to actions. Then, the next state is chosen randomly according to the constructed probability distribution, and the play goes on. Whenever a new state s' is entered from a previous state s along a play, each event scheduled in s' is assigned a new *delay* which is chosen randomly according to the corresponding (conditional) density. The state s' then "reacts" to the event with the least delay (under the assumptions adopted in this paper, the probability of assigning the same delay to different events is zero).

Our contribution. In this work we consider SRTGs with *deterministic timed automata (DTA)* objectives. Intuitively, a timed automaton "observes" a play of a given SRTG and checks that certain timing constraints are satisfied. A simple example of a property that can be encoded by a DTA is "whenever a new request is generated, it is either serviced within the next 10 time units, or the system eventually enters a safe state". In this case, we want to setup the internal controls so that the above property holds for almost all plays, no matter what decisions are taken in adversarial controls. Hence, the aim of player $\Box$ is to maximize the probability that a play is accepted by a given timed automaton, while player $\Diamond$ aims at the opposite. By applying the result of [14], we obtain that SRTGs with DTA objectives have a *value*, i.e., $\sup_\sigma \inf_\pi P^{\sigma,\pi} = \inf_\pi \sup_\sigma P^{\sigma,\pi}$, where σ and π range over all strategies of player $\Box$ and player $\Diamond$, and $P^{\sigma,\pi}$ is the probability of all plays satisfying a given DTA objective. This immediately raises the question whether the players have *optimal* strategies which guarantee the equilibrium value against every strategy of the opponent. We show that the answer is *negative*. Then, we concentrate on the *qualitative* variant of the problem, which is perhaps most interesting from the practical point of view. An *almost-sure winning* strategy for player $\Box$ is a strategy such that for every strategy of player $\Diamond$, the probability of all plays satisfying a given DTA objective is equal to one. The main result of this paper is the following: We show that if player $\Box$ has *some* almost-sure winning strategy, then she also has a *DTA* almost-sure winning strategy, which can be encoded by a deterministic timed automaton $\mathcal{A}$ constructible in exponential time. The automaton $\mathcal{A}$ reads the history of a play, and the decision taken by the corresponding DTA strategy depends only on the region of the resulting configuration entered by $\mathcal{A}$.

Our constructions and proofs are combinations of standard techniques (used for timed automata and finite-state games) and some new non-trivial observations that are specific for the considered model of SRTGs. We also adapt some ideas presented in [2] (in particular, we use the concept of δ-separation).

Related work. Continuous-time (semi)Markov chains are a classical and deeply studied model with a mature mathematical theory (see, e.g., [17, 19]). Continuous-time Markov decision processes (CTMDPs) [7, 5, 16] combine probabilistic and non-deterministic choice, but all events are required to be exponentially distributed. Two player games over continuous-time Markov chains were considered only recently [10, 18]. Timed automata [3] were originally introduced as a non-stochastic model with time. Probabilistic semantics of timed automata was proposed in [4, 6], and a more general model of stochastic games over timed automata was considered in [9]. In this paper we build mainly on the previous work about GSMPs [13, 20, 21] and RTPs [2, 1] and interpret timed automata as a model-independent specification language which can express important properties of timed systems. This view is adopted also in [12] where continuous-time Markov chains are checked against timed-automata specifications.

Let us note that our technical treatment of events is somewhat different from the one used for GSMPs and RTPs. Intuitively, in GSMPs (and RTPs), each event is assigned its delay only when it is newly scheduled, and this delay is just updated when moving from state to state (by subtracting the elapsed time) until the event happens or it is disabled. For example, if two messages e and e' are sent, both of them are assigned randomly chosen delays d_e and $d_{e'}$. The smaller of the two delays (say d_e) triggers a transition to the next state, where the delay of $d_{e'}$ is updated by subtracting d_e. Since the current delays of all events are explicitly recorded in the state-space of GSMPs and RTPs, this formalism cannot be directly extended to perfect-information games (the players would "see" the delays assigned to events, i.e., they would know what is going to happen in the future). In our model of SRTGs, we always assign a new random delay to all events that are scheduled in a given control state, but we adjust the corresponding densities (from a "probabilistic" point of view, this approach is equivalent to the one used for GSMPs and RTPs).

Due to space constraints, most of the proofs are omitted and can be found in a full version of this paper [11].

2 Definitions

In this paper, the sets of all positive integers, non-negative integers, real numbers, positive real numbers, and non-negative real numbers are denoted by $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$, respectively.

Let A be a finite or countably infinite set. A *probability distribution* on A is a function $f : A \to \mathbb{R}_{\geq 0}$ such that $\sum_{a \in A} f(a) = 1$. We say that f is *rational* if $f(a)$ is rational for every $a \in A$. The set of all distributions on A is denoted by $\mathcal{D}(A)$. A *σ-field* over a set Ω is a set $\mathcal{F} \subseteq 2^{\Omega}$ that includes Ω and is closed under complement and countable union. A *measurable space* is a pair $(\Omega, \mathcal{F})$ where Ω is a set called *sample space* and $\mathcal{F}$ is a σ-field over Ω whose elements are called *measurable sets*. A *probability measure* over a measurable space $(\Omega, \mathcal{F})$ is a function $\mathcal{P} : \mathcal{F} \to \mathbb{R}_{\geq 0}$ such that, for each countable collection $\{X_i\}_{i \in I}$ of pairwise disjoint elements of $\mathcal{F}$, $\mathcal{P}(\bigcup_{i \in I} X_i) = \sum_{i \in I} \mathcal{P}(X_i)$, and moreover $\mathcal{P}(\Omega) = 1$. A *probability space* is a triple $(\Omega, \mathcal{F}, \mathcal{P})$, where $(\Omega, \mathcal{F})$ is a measurable space and $\mathcal{P}$ is a probability measure over $(\Omega, \mathcal{F})$. We say that a property $A \subseteq \Omega$ holds *for almost all* elements of a measurable set Y if $\mathcal{P}(Y) > 0$, $A \cap Y \in \mathcal{F}$, and $\mathcal{P}(A \mid Y) = 1$.

Let us note that all of the integrals used in this paper should be understood as Lebesgue integrals, although we use Riemann-like notation.

2.1 Stochastic real-time games

Let $\mathcal{E}$ be a finite set of *events*, which are independent of each other. To every $e \in \mathcal{E}$ we associate its *lower bound* $\ell_e \in \mathbb{N}_0$, *upper bound* $u_e \in \mathbb{N} \cup \{\infty\}$, and a *density function* $f_e : \mathbb{R} \to \mathbb{R}$ which is positive on (ℓ_e, u_e) such that $\int_{\ell_e}^{u_e} f_e(x)\, dx = 1$. Further, for every $b \in \mathbb{R}_{\geq 0}$ we also define the *conditional density function* $f_{e|b} : \mathbb{R} \to \mathbb{R}$ as follows:

$$f_{e|b}(x) = \frac{f_e(x+b)}{\left[\int_b^{u_e} f_e(y)\, dy\right]_{\neq 0}}$$

Here $[\cdot]_{\neq 0} : \mathbb{R} \to \mathbb{R}$ is a function which for a given x returns either x or 1 depending on whether $x \neq 0$ or not, respectively. The function f_e defines the density of delaying the event e, i.e., for every time $t \in \mathbb{R}_{\geq 0}$, the probability of delaying e for at most t is equal to $\int_0^t f_e(x)\, dx$. Note that the integral $\int_0^t f_{e|b}(x)\, dx$ is equal to the conditional probability of delaying e for at most $b + t$ under the condition that e is delayed for at least b. Since all events are mutually independent, for every subset $E' \subseteq E$ we have that the conditional probability of delaying all events in E' for at least $b + t$ under the condition that all events in E' are delayed for at least b is equal to $\prod_{e \in E'} \int_t^{\infty} f_{e|b}(x)\, dx$.

Definition 1. *A* stochastic real-time game (SRTG) *is a tuple* $\mathcal{G} = (S, E, C_\Box, C_\Diamond, Act, F, A, \mu_0)$ *where* S *is a finite set of* states*,* $E : S \to 2^{\mathcal{E}}$ *assigns to each* $s \in S$ *the set of events scheduled to occur in* s*,* $C_\Box$ *and* $C_\Diamond$ *are finite disjoint sets of* controls *of player* $\Box$ *and player* $\Diamond$*,* $Act \subseteq \mathcal{D}(S)$ *is a finite set of* actions*,* F *is a* flow function *which to every pair* (s, e)*, where* $s \in S$ *and* $e \in E(s)$*, assigns a control of* $C_\Box \cup C_\Diamond$*,* $A : C_\Box \cup C_\Diamond \to 2^{Act}$ *assigns to each control* c *a non-empty finite set of actions enabled at* c*, and* $\mu_0 \in \mathcal{D}(S)$ *is an* initial *distribution.*

A *stamp* is an element (s, t, e) of $S \times \mathbb{R}_{>0} \times \mathcal{E}$ where $e \in E(s)$. A (computational) *history* of $\mathcal{G}$ is a finite sequence $\mathfrak{h} = (s_0, t_0, e_0), \ldots, (s_n, t_n, e_n)$ of stamps. Intuitively, t_i is the time spent in s_i while waiting for some of the events scheduled in s_i, and e_i is the event that triggered a transition to the next state s_{i+1}. A *strategy* of player $\odot$, where $\odot \in \{\Box, \Diamond\}$, is a measurable function which to every history $(s_0, t_0, e_0), \ldots, (s_n, t_n, e_n)$ such that $F(s_n, e_n) = c \in C_\odot$ assigns a probability distribution over the set $A(c)$ of actions that are enabled at c. The set of all strategies of player $\Box$ and player $\Diamond$ are denoted by Σ and Π, respectively.

Let $(\sigma, \pi) \in \Sigma \times \Pi$. The corresponding *play* of $\mathcal{G}$ is initiated in some $s_0 \in S$ (with probability $\mu_0(s_0)$). Then, each event $e \in E(s_0)$ is assigned a randomly chosen delay $d_e^0 \in \mathbb{R}_{>0}$ according to the density f_e (note that $f_e = f_{e|0}$). Let $t_0 = \min\{d_e^0 \mid e \in E(s_0)\}$ be the minimal delay of all events scheduled in s_0, and let $trigger_0$ be the set of all $e \in E(s_0)$ such that $d_e^0 = t_0$. The event e_0 which "triggers" a transition to the next state is the least element of $trigger_0$ w.r.t. some fixed linear ordering $\preceq$ (note that the probability of assigning the same delay to different events is zero, and hence the choice of $\preceq$ is irrelevant; we need this ordering just to make our semantics well defined). The event e_0 determines a control $c = F(s_0, e_0)$, where the responsible player makes a decision according to her strategy τ, i.e., selects a distribution $\tau(\mathfrak{h})$ over $A(c)$ where $\mathfrak{h} = (s_0, t_0, e_0)$ is the current history. Hence, the next state s_1 is chosen with probability $\sum_{\mu \in A(c)} \tau(\mathfrak{h})(\mu) \cdot \mu(s_1)$. In s_1, we assign a randomly chosen delay d_e^1 to every $e \in E(s_1)$ according to the conditional density $f_{e|b}$, where b is determined as follows: If e was scheduled in the previous state s_0 and $e \neq e_0$, then $b = t_0$; otherwise $b = 0$. The event e_1 is the least event (w.r.t. $\preceq$) with the minimal delay $t_1 = \min\{d_e^1 \mid e \in E(s_1)\}$. The next state s_2 is chosen randomly by combining the strategy of the respective player with the corresponding actions. In general, after entering a state s_i, every $e \in E(s_i)$ is assigned a randomly chosen delay d_e^i according to the conditional density $f_{e|b}$ where b is the total waiting time for e accumulated in the history of the play.

To formalize the intuition given above, we define a suitable probability space $(Play, \mathcal{F}, \mathcal{P}_{\mathfrak{h}}^{\sigma,\pi})$ over the set $Play$ of all infinite sequences of stamps, where $\mathfrak{h}$ is a history of steps "performed previously" (the technical convenience of $\mathfrak{h}$ becomes apparent later in Section 3; the definition given below is perhaps easier to understand in the special case when $\mathfrak{h}$ is empty). For the rest of this section, we fix a history $\mathfrak{h} = (s_0, t_0, e_0), \ldots, (s_n, t_n, e_n)$ where $n \in \mathbb{N}_0 \cup \{-1\}$. If $n = -1$, then $\mathfrak{h}$ is empty. A *template* is a finite sequence of the form $B = (s_{n+1}, I_{n+1}, e_{n+1}), \ldots, (s_{n+m}, I_{n+m}, e_{n+m})$ such that $m \geq 1$, $e_i \in E(s_i)$, and I_i is an interval in $\mathbb{R}_{>0}$ for every $n+1 \leq i \leq n+m$. Each such B determines the corresponding *cylinder* $Play(B) \subseteq Play$ consisting of all sequences of the form $(s_{n+1}, t_{n+1}, e_{n+1}), \ldots, (s_{n+m}, t_{n+m}, e_{n+m}), \ldots$ where $t_i \in I_i$ for all $n+1 \leq i \leq n+m$. The σ-field $\mathcal{F}$ is the Borel σ-field generated by all cylinders. For each cylinder $Play(B)$, the probability $\mathcal{P}_{\mathfrak{h}}^{\sigma,\pi}(Play(B))$ is defined in the way described below. Then, $\mathcal{P}_{\mathfrak{h}}^{\sigma,\pi}$ is extended to $\mathcal{F}$ (in the unique way) by applying the extension theorem (see, e.g., [8]).

It remains to show how to define the probability $\mathcal{P}_{\mathfrak{h}}^{\sigma,\pi}(Play(B))$ of a given cylinder $Play(B)$, where $B = (s_{n+1}, I_{n+1}, e_{n+1}), \ldots, (s_{n+m}, I_{n+m}, e_{n+m})$. We put $\mathcal{P}_{\mathfrak{h}}^{\sigma,\pi}(Play(B)) = T_{n+1}$, where the expression T_i is defined inductively for all $n+1 \leq i \leq n+m+1$ as follows:

$$T_i = \begin{cases} \int_{I_i} State_i \cdot Win_i \cdot T_{i+1} \, dt_i & \text{if } n+1 \leq i \leq n+m; \\ 1 & \text{if } i = n+m+1. \end{cases}$$

Observe that T_{n+1} is an expression with m nested integrals. Further, note that when constructing T_{i+1}, we already have $t_0, \ldots, t_i$ at our disposal (each t_i is either fixed in $\mathfrak{h}$, or it is a variable used in some of the preceding integrals).

The subterm $State_i$ corresponds to the probability that s_i is chosen as the next state, assuming that the current history is $(s_0, t_0, e_0), \ldots, (s_{i-1}, t_{i-1}, e_{i-1})$. Hence, we define

- $State_{n+1} = \mu_0(s_{n+1})$ if $\mathfrak{h}$ is empty, otherwise $State_{n+1} = \sum_{\mu \in A(c)} \tau(\mathfrak{h})(\mu) \cdot \mu(s_{n+1})$, where $c = F(s_n, e_n)$, and τ is either σ or π, depending on whether $c \in C_\Box$ or $c \in C_\Diamond$, respectively.
- $State_i = \sum_{\mu \in A(c)} \tau(\mathfrak{h}')(\mu) \cdot \mu(s_i)$, where $n+1 < i \le n+m$, $c = F(s_{i-1}, e_{i-1})$, $\mathfrak{h}' = (s_0, t_0, e_0), \ldots, (s_{i-1}, t_{i-1}, e_{i-1})$, and τ is either σ or π, depending on whether $c \in C_\Box$ or $c \in C_\Diamond$, respectively.

The most complicated part is the definition of Win_i which intuitively corresponds to the probability that the event e_i "wins" the competition among the events scheduled in s_i.

In order to define Win_i, we have to overcome a technical obstacle that the events scheduled in s_i might have been scheduled also in the preceding states. For each $e \in E(s_i)$, let $K(e,i)$ be the minimal index such that $0 \le K(e,i) \le i$ and for all $K(e,i) \le j < i$ we have that $e \in E(s_j)$ and $e \ne e_j$. We put $b(e,i) = t_{K(e,i)} + \cdots + t_{i-1}$. Intuitively, $b(e,i)$ is the total waiting time for e accumulated in the history of the play. Note that if $K(e,i) = i$, then the defining sum of $b(e,i)$ is empty and hence equal to zero. We put

$$Win_i \quad = \quad f_{e_i | b(e_i,i)}(t_i) \quad \cdot \prod_{\substack{e \in E(s_i) \\ e \ne e_i}} \int_{t_i}^{\infty} f_{e|b(e,i)}(x)\, dx.$$

2.2 Deterministic timed automata

Let $\mathcal{X}$ be a finite set of *clocks*. A *valuation* is a function $\nu : \mathcal{X} \to \mathbb{R}_{\ge 0}$. For every valuation ν and every subset $X \subseteq \mathcal{X}$ of clocks, we use $\nu[X := \mathbf{0}]$ to denote the unique valuation such that $\nu[X := \mathbf{0}](x) = 0$ for all $x \in X$, and $\nu[X := \mathbf{0}](x) = \nu(x)$ for all $x \in \mathcal{X} \setminus X$. Further, for every valuation ν and every $\delta \in \mathbb{R}_{\ge 0}$, the symbol $\nu + \delta$ denotes the unique valuation such that $(\nu + \delta)(x) = \nu(x) + \delta$ for all $x \in \mathcal{X}$.

A *clock constraint* (or *guard*) is a finite conjunction of basic constraints of the form $x \bowtie c$, where $x \in \mathcal{X}$, $\bowtie \in \{<, \le, >, \ge\}$, and $c \in \mathbb{N}_0$. For every valuation ν and every clock constraint g we have that ν either does or does not satisfy g, written $\nu \models g$ or $\nu \not\models g$, respectively (the satisfaction relation is defined in the expected way). Sometimes we slightly abuse our notation and identify a guard g with the set of all valuations that satisfy g (for example, we write $g \cap g'$). The set of all guards over $\mathcal{X}$ is denoted by $\mathcal{B}(\mathcal{X})$.

Definition 2. *A* deterministic timed automaton (DTA) *is a tuple* $\mathcal{A} = (Q, \Sigma, \mathcal{X}, \longrightarrow, q_0, T)$*, where* Q *is a nonempty finite set of* locations*,* Σ *is a finite* alphabet*,* $\mathcal{X}$ *is a finite set of* clocks*,* $q_0 \in Q$ *is an* initial location*,* $T \subseteq Q$ *is a set of* target locations*, and* $\longrightarrow \subseteq Q \times \Sigma \times \mathcal{B}(\mathcal{X}) \times 2^{\mathcal{X}} \times Q$ *is an* edge relation *such that for all* $q \in Q$ *and* $a \in \Sigma$ *we have the following:*

1. *the guards are deterministic, i.e., for all edges of the form* (q, a, g_1, X_1, q_1) *and* (q, a, g_2, X_2, q_2) *such that* $g_1 \cap g_2 \ne \emptyset$ *we have that* $g_1 = g_2$*,* $X_1 = X_2$*, and* $q_1 = q_2$*;*
2. *the guards are total, i.e., for all* $q \in Q$*,* $a \in \Sigma$*, and every valuation* ν *there is an edge* (q, a, g, X, q') *such that* $\nu \models g$*.*

A *configuration* of $\mathcal{A}$ is a pair (q, ν), where $q \in Q$ and ν is a valuation. An *infinite timed word* is an infinite sequence $w = c_0 c_1 c_2 \ldots$ where each c_i is either a letter of Σ or a positive real number denoting a time stamp (note that letters and time stamps are not required to alternate in w). The *run* of $\mathcal{A}$ on w is the unique infinite sequence $(q_0, \nu_0)\ c_0\ (q_1, \nu_1)\ c_1\ \dots$ such that q_0 is the initial location of $\mathcal{A}$, $\nu_0 = \mathbf{0}$, and for each $i \in \mathbb{N}_0$ we have that

- if c_i is a time stamp $t \in \mathbb{R}_{\geq 0}$, then $q_{i+1} = q_i$ and $\nu_{i+1} = \nu_i + t$,
- if c_i is an input letter $a \in \Sigma$, then there is a unique edge (q_i, a, g, X, q) such that $\nu_i \models g$, and we require that $q_{i+1} = q$ and $\nu_{i+1} = \nu_i[X := \mathbf{0}]$.

We say that w is *accepted* by $\mathcal{A}$ if the run of $\mathcal{A}$ on w visits a configuration (q, ν) where $q \in T$. Without restrictions, we may assume that each $q \in T$ is *absorbing*, i.e., all of the outgoing edges of q lead back to q.

In this paper, we use DTA for two different purposes. Firstly, DTA are used as a generic *specification language* for properties of timed systems. In this case, a given DTA is constructed so that it accepts the set of all "correct" runs (timed words) of a given timed system. Formally, for a fixed SRTG $\mathcal{G}$ with a set of states S, a finite set Ap of atomic propositions and a labeling $L : S \to 2^{Ap}$, every play $\varrho = (s_0, t_0, e_0), (s_1, t_1, e_1), \dots$ of $\mathcal{G}$ determines a unique infinite timed word $Ap(\varrho) = L(s_0)\, t_0\, L(s_1)\, t_1 \dots$. A DTA $\mathcal{A}$ with alphabet 2^{Ap} then either accepts $Ap(\varrho)$ or not. Intuitively, the automaton $\mathcal{A}$ encodes some desirable property of plays, and the aim of player $\Box$ and player $\Diamond$ is to maximize and minimize the probability of all plays accepted by $\mathcal{A}$, respectively. We denote $Play(\mathcal{A}) \subseteq Play$ the set of all plays ϱ such that $Ap(\varrho)$ is accepted by $\mathcal{A}$. Note that the DTA does not read any information about the events that occurred. However, one can easily encode the information about the last event into the subsequent state by considering copies s_e of each state s for every event e.

Secondly, we use DTA to encode strategies in stochastic real-time games. Here, the constructed DTA "observes" the history of a play, and the decisions taken by the corresponding strategy depend only on the resulting configuration (q, ν). Actually, we require that the decision depends only on the *region* of (q, ν) (see [3] or Section 3.1), which makes DTA strategies finitely representable. Formally, every history $\mathfrak{h} = (s_0, t_0, e_0) \cdots (s_n, t_n, e_n)$ of $\mathcal{G}$ can be seen as a (finite) timed word $s_0, t_0, e_0, \cdots, s_n, t_n, e_n$, where the states and events are seen as letters, and the delays are seen as time stamps. We define DTA strategies as follows.

Definition 3. *A* DTA strategy *is a strategy τ such that there is a DTA $\mathcal{A}$ with alphabet $S \cup \mathcal{E}$ satisfying the following: for every history $\mathfrak{h}$ we have that $\tau(\mathfrak{h})$ is a rational distribution which depends only on the region of (q, ν), where (q, ν) is the configuration entered by $\mathcal{A}$ after reading the word $\mathfrak{h}$.*

3 Results

For the rest of the paper, we fix an SRTG $\mathcal{G} = (S, E, C_\Box, C_\Diamond, Act, F, A, \mu_0)$, a finite set Ap of atomic propositions, a labeling $L : S \to 2^{Ap}$, and a DTA $\mathcal{A} = (Q, 2^{Ap}, \mathcal{X}, \longrightarrow, q_0, T)$.

As observed in [14], the determinacy result for Blackwell games [15] implies determinacy of a large class of stochastic games. This abstract class includes the games studied in this paper, and thus we obtain the following:

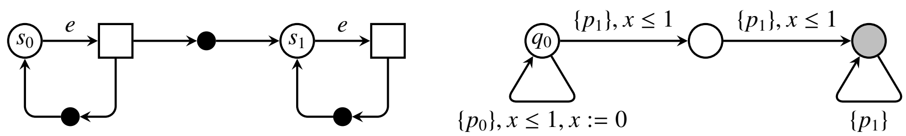


Fig. 2. Player $\Box$ does not have an optimal strategy.

Proposition 1. *Let* $\mathfrak{h}$ *be a history of* $\mathcal{G}$*. Then*

$$\sup_{\sigma\in\Sigma} \inf_{\pi\in\Pi} \mathcal{P}^{\sigma,\pi}_{\mathfrak{h}}(Play(\mathcal{A})) \quad = \quad \inf_{\pi\in\Pi} \sup_{\sigma\in\Sigma} \mathcal{P}^{\sigma,\pi}_{\mathfrak{h}}(Play(\mathcal{A}))$$

The value *of* $\mathcal{G}$ *(with respect to* $\mathfrak{h}$*), denoted by* $val_{\mathfrak{h}}$*, is defined by the above equality.*

The existence of $val_{\mathfrak{h}}$ implies the existence of ε-optimal strategies for both players. However, note that player $\Box$ does not necessarily have an *optimal* strategy which would achieve the outcome $val_{\mathfrak{h}}$ or better against every strategy of player $\Diamond$, even if $val_{\mathfrak{h}} = 1$ and $C_\Diamond = \emptyset$. A simple counterexample is given in Fig. 2. Here f_e is the uniform density on $(0, 1)$ (i.e., $f_e(x) = 1$ for all $x \in (0, 1)$), $Ap = \{p_0, p_1\}$, $L(s_0) = p_0$, $L(s_1) = p_1$, and the only target location is gray. All of the "missing" edges in the depicted DTA (which are needed to satisfy the requirement that the guards are total) lead to a "garbage" location. The initial distribution μ_0 assigns 1 to s_0. Now observe that $val_{\mathfrak{h}} = 1$ (where $\mathfrak{h}$ is the empty history), because for every $\varepsilon > 0$, player $\Box$ can "wait" in s_0 until e is fired so that its delay is smaller than ε (this eventually happens with probability 1), and then she moves to s_1. The probability that e is assigned a delay at most $1 - \varepsilon$ in s_1 is $1 - \varepsilon$, and hence the constructed DFA accepts a play with probability $1-\varepsilon$. However, player $\Box$ has no optimal strategy.

In this paper we consider the existence and effective constructability of *almost-sure* winning strategies for player $\Box$. Formally, a strategy $\sigma \in \Sigma$ is almost-sure winning for a history $\mathfrak{h}$ if for every strategy $\pi \in \Pi$ we have that $\mathcal{P}^{\sigma,\pi}_{\mathfrak{h}}(Play(\mathcal{A})) = 1$. We show the following:

Theorem 1. *Let* $\mathfrak{h}$ *be a history. If player* $\Box$ *has (some) almost-sure winning strategy for* $\mathfrak{h}$*, then she also has a DTA almost-sure winning strategy for* $\mathfrak{h}$*. The existence of a DTA almost-sure winning strategy for* $\mathfrak{h}$ *is decidable in exponential time, and if it exists, it can be constructed in exponential time.*

A proof of Theorem 1 is not immediate and requires several steps. First, in Section 3.1 we construct a *product game* $\mathcal{G}_\mathcal{A}$ of $\mathcal{G}$ and $\mathcal{A}$ and show that $\mathcal{G}_\mathcal{A}$ can be examined instead of $\mathcal{G}$ and $\mathcal{A}$. The existence of a DTA almost-sure winning strategy in $\mathcal{G}_\mathcal{A}$ is analyzed in Section 3.2. Finally, in Section 3.3 we present an algorithm which computes a DTA almost-sure winning strategy if it exists.

3.1 The product game

Intuitively, the product game of $\mathcal{G}$ and $\mathcal{A}$, denoted by $\mathcal{G}_\mathcal{A}$, is constructed by simulating the execution of $\mathcal{A}$ on-the-fly in $\mathcal{G}$. Waiting times for events and clock valuations are represented explicitly in the states of $\mathcal{G}_{\mathcal{A}}$, and hence the state-space of $\mathcal{G}_{\mathcal{A}}$ is uncountable. Still, $\mathcal{G}_{\mathcal{A}}$ is in many aspects similar to $\mathcal{G}$, and therefore we use a suggestive notation compatible with the one used for $\mathcal{G}$. To distinguish among the notions related to $\mathcal{G}$ and $\mathcal{G}_{\mathcal{A}}$, we consistently use the "p-" prefix. Hence, $\mathcal{G}$ has stamps, states, histories, etc., while $\mathcal{G}_{\mathcal{A}}$ has p-stamps, p-states, p-histories, etc.

Let $n = |\mathcal{E}| + |\mathcal{X}|$. The clock values of $\mathcal{A}$ and the delays of currently scheduled events are represented by a *p-vector* $\xi \in \mathbb{R}^n_{\geq 0}$. The set of *p-states* is $S \times Q \times \mathbb{R}^n_{\geq 0}$, and the sets of *p-controls* of player $\Box$ and player $\Diamond$ are $C_\Box \times Q \times \mathbb{R}^n_{\geq 0}$ and $C_\Diamond \times Q \times \mathbb{R}^n_{\geq 0}$, respectively.

The dynamics of $\mathcal{G}_{\mathcal{A}}$ is determined as follows. First, we define a *p-flow* function $F_{\mathcal{A}}$, which to a given p-stamp (s, q, ξ, t, e) assigns the p-control (c, q', ξ'), where $c = F(s, e)$, and q', ξ' are determined as follows. Let $(q, L(s), g, X, q')$ be the unique edge of $\mathcal{A}$ such that the guard g is satisfied by the clock valuation stored in $\xi + t$. We put $\xi' = (\xi +_s t)[(e \cup X) := \mathbf{0}]$. The operator "$+_s t$" adds t to all clocks stored in ξ and to all events scheduled in s, and $(e \cup X) := \mathbf{0}$ resets all clocks of X to zero and assigns zero delay to e. Second, we define the set of p-actions. For every p-control (c, q, ξ) and an action $a \in A(c)$, there is a corresponding p-action which to a given p-state (s', q, ξ'), where $\xi' = \xi[(\mathcal{E} \setminus E(s')) := \mathbf{0}]$, assigns the probability $a(s')$.

A *p-stamp* is an element (s, q, ξ, t, e) of $S \times Q \times \mathbb{R}^n_{\geq 0} \times \mathbb{R}_{>0} \times \mathcal{E}$. Now we define *p-histories* and *p-plays* as sequences of p-stamps. In the game $\mathcal{G}$ we allowed arbitrary sequences of stamps, whereas in the product game we need the automaton part of the product to be consistent with the game part. We say that a p-stamp $x_1 = (s_1, q_1, \xi_1, t_1, e_1)$ is consistent with a p-stamp $x_0 = (s_0, q_0, \xi_0, t_0, e_0)$ if the image of x_0 under the p-flow function is a p-control (c, q_1, ξ') such that $\xi_1 = \xi'[A := \mathbf{0}]$ where A is the set of actions not enabled in s_1.

A *p-history* is a finite sequence of p-stamps $\mathfrak{p} = x_0 \ldots x_n$ such that x_i is consistent with x_{i+1} for all $0 \leq i < n$. A *p-play* is an infinite sequence of p-stamps $x_0 x_1 \ldots$ where each finite prefix $x_0 \ldots x_i$ is a p-history. Each p-history $\mathfrak{p} = (s_0, q_0, \xi_0, t_0, e_0), \ldots, (s_n, q_n, \xi_n, t_n, e_n)$ can be mapped to a unique history $H(\mathfrak{p}) = (s_0, t_0, e_0), \ldots, (s_n, t_n, e_n)$. Note that H is in fact a bijection, because each history induces a unique finite execution of the DTA $\mathcal{A}$ and the consistency condition reflects this unique execution. By the *last p-control* of a p-history $\mathfrak{p}$ we denote the image of the last p-stamp of $\mathfrak{p}$ under the p-flow function.

Region relation. Although the state-space of $\mathcal{G}_{\mathcal{A}}$ is uncountable, we can define a variant of *region relation* over p-histories which has a finite index, and then work with finitely many *regions*.

For a given $x \in \mathbb{R}_{\geq 0}$, we use *frac*$(x)$ to denote the fractional part of x, and *int*(x) to denote the integral part of x. For $x, y \in \mathbb{R}_{\geq 0}$, we say that x and y *agree on integral part* if *int*(x) = *int*(y) and neither or both x, y are integers. A *relevant bound* of a clock x is the largest constant c that appears in all guards. A *relevant bound* of an event e is u_e if $u_e < \infty$, and ℓ_e otherwise. We say that an element $a \in \mathcal{E} \cup \mathcal{X}$ is *relevant* for ξ if $\xi(a) \leq r$ where r is the relevant bound of a. Finally, we put $\xi_1 \approx \xi_2$ if

- for all relevant $a \in \mathcal{E} \cup \mathcal{X}$ we have that $\xi_1(a)$ and $\xi_2(a)$ agree on integral parts;
- for all relevant $a, b \in \mathcal{E} \cup \mathcal{X}$ we have that *frac*$(\xi_1(a)) \leq$ *frac*$(\xi_1(b))$ if and only if *frac*$(\xi_2(a)) \leq$ *frac*$(\xi_2(b))$.

The equivalence classes of $\approx$ are called *time areas*. Now we can define the promised *region relation* $\sim$ on p-histories. Let $\mathfrak{p}_1$ and $\mathfrak{p}_2$ be p-histories such that (c_1, q_1, ξ_1) is the last p-control of $\mathfrak{p}_1$ and (c_2, q_2, ξ_2) is the last p-control of $\mathfrak{p}_2$. We put $\mathfrak{p}_1 \sim \mathfrak{p}_2$ iff $c_1 = c_2$, $q_1 = q_2$ and $\xi_1 \approx \xi_2$. Note that $\sim$ is an equivalence with a finite index. The equivalence classes of $\sim$ are called *regions*. A *target region* is a region that contains such p-histories whose last p-controls have a target location in the second component. The sets of all regions and target regions are denoted by $\mathcal{R}$ and $\mathcal{R}_T$, respectively.

Remark 1. Let us note that the region construction described above can also be applied to configurations of timed automata, where it coincides with the standard region construction of [3].

Strategies in the product game. Note that every pair of strategies $(\sigma, \pi) \in \Sigma \times \Pi$ defined for the original game $\mathcal{G}$ can also be applied in the constructed product game $\mathcal{G}_{\mathcal{A}}$ (we just ignore the extra components of p-stamps). By re-using the construction of Section 2.1, for every p-history $\mathfrak{p}$ and every pair of strategies $(\sigma, \pi) \in \Sigma \times \Pi$, we define a probability measure $\mathcal{P}_{\mathfrak{p}}^{\sigma,\pi}$ on the Borel σ-field $\mathcal{F}$ over the p-plays in $\mathcal{G}_{\mathcal{A}}$ (the details are given in [11]).

For every $\mathcal{S} \subseteq \mathcal{R}$, let *Reach*$(\mathcal{S})$ be the set of all p-plays that visit a region of $\mathcal{S}$ (i.e., some prefix of the p-play belongs to some $r \in \mathcal{S}$). We say that a strategy $\sigma \in \Sigma$ is *almost-sure winning* in $\mathcal{G}_{\mathcal{A}}$ for a p-history $\mathfrak{p}$ if for every $\pi \in \Pi$ we have that $\mathcal{P}_{\mathfrak{p}}^{\sigma,\pi}(\mathit{Reach}(\mathcal{R}_T)) = 1$. The relationship between almost-sure winning strategies in $\mathcal{G}$ and $\mathcal{G}_{\mathcal{A}}$ is formulated in the next proposition.

Proposition 2. *Let $\sigma \in \Sigma$ and $\mathfrak{p}$ be a p-history. Then σ is almost-sure winning for $\mathfrak{p}$ in $\mathcal{G}_{\mathcal{A}}$ iff σ is almost-sure winning for $H(\mathfrak{p})$ in $\mathcal{G}$.*

Another observation about strategies in $\mathcal{G}_{\mathcal{A}}$ which is heavily used in the next sections concerns strategies that are *constant on regions*. Formally, a strategy $\tau \in \Sigma \cup \Pi$ is constant on regions if for all p-histories $\mathfrak{p}_1$ and $\mathfrak{p}_2$ such that $\mathfrak{p}_1 \sim \mathfrak{p}_2$ we have that $\tau(\mathfrak{p}_1) = \tau(\mathfrak{p}_2)$.

Proposition 3. *Every strategy $\tau \in \Sigma \cup \Pi$ which is constant on regions is a DTA strategy.*

Proof (Sketch). We transform τ into a DTA $A_{\mathcal{G}_{\mathcal{A}}}$ whose regions are in one-to-one correspondence with the regions of $\mathcal{G}_{\mathcal{A}}$. The automaton $A_{\mathcal{G}_{\mathcal{A}}}$ reads a sequence of stamps of $\mathcal{G}$ and simulates the behavior of $\mathcal{G}_{\mathcal{A}}$. It has a special clock for every clock of $\mathcal{A}$ and every event of $\mathcal{E}$, and uses its locations to store also the current state of the game. The details are given in [11]. □

Note that due to Proposition 3, every strategy constant on regions can be effectively transformed into a DTA strategy.

3.2 Almost-sure winning strategies

In this section, we outline a proof of the following theorem:

Theorem 2. *Let* $\mathfrak{p}$ *be a p-history. If there is a strategy* $\sigma \in \Sigma$ *which is almost-sure winning in* $\mathcal{G}_{\mathcal{A}}$ *for* $\mathfrak{p}$*, then there is a DTA strategy* $\sigma^* \in \Sigma$ *which is almost-sure winning for* $\mathfrak{p}$.

Note that due to Proposition 3, it suffices to show that there is an almost-sure winning strategy in $\mathcal{G}_{\mathcal{A}}$ for $\mathfrak{p}$ which is constant on regions.

Observe that if $\sigma \in \Sigma$ is an almost-sure winning strategy in $\mathcal{G}_{\mathcal{A}}$ for $\mathfrak{p}$, then for every $\pi \in \Pi$ the plays of $\mathcal{G}_{\mathcal{A}}$ may visit only regions from which it is still possible to visit a target region. Hence, a good candidate for an almost-sure winning DTA strategy in $\mathcal{G}_{\mathcal{A}}$ for $\mathfrak{p}$ is a strategy which never leaves this set of "safe" regions. This motivates the following definition (in the rest of this section we often write $\mathfrak{p} \in \mathcal{S}$, where $\mathfrak{p}$ is a p-history and $\mathcal{S}$ a set of regions, to indicate that $\mathfrak{p} \in \bigcup_{r \in \mathcal{S}} r$).

Definition 4. *A DTA strategy* $\sigma \in \Sigma$ *is a* candidate on a set of regions $\mathcal{S} \subseteq \mathcal{R}$ *if for every* $\pi \in \Pi$ *and every p-history* $\mathfrak{p} \in \mathcal{S}$ *we have that* $\mathcal{P}_{\mathfrak{p}}^{\sigma,\pi}(\mathit{Reach}(\mathcal{R} \setminus \mathcal{S})) = 0$ *and* $\mathcal{P}_{\mathfrak{p}}^{\sigma,\pi}(\mathit{Reach}(\mathcal{R}_T)) > 0$.

In the following, we prove Propositions 4 and 5 that together imply Theorem 2.

Proposition 4. *Let* σ *be an almost-sure winning strategy in* $\mathcal{G}_{\mathcal{A}}$ *for a p-history* $\mathfrak{p}_0$*. Then there is a set* $\mathcal{S} \subseteq \mathcal{R}$ *and a DTA strategy* σ^* *such that* $\mathfrak{p}_0 \in \mathcal{S}$ *and* σ^* *is a candidate on* $\mathcal{S}$.

Proof (Sketch). We define $\mathcal{S}$ as the set of all regions reached with positive probability in an arbitrary play where player $\Box$ uses the strategy σ and player $\Diamond$ uses some $\pi \in \Pi$. For every action a, let $\mathit{p\text{-}hist}_a$ be the set of all p-histories where σ assigns a positive probability to a. For every region $r \in \mathcal{S}$, we denote by A_r the set of all $a \in \mathit{Act}$ for which there is $\pi \in \Pi$ such that $\mathcal{P}_{\mathfrak{p}_0}^{\sigma,\pi}(\mathit{p\text{-}hist}_a \cap r) > 0$.

- Firstly, we show that every DTA strategy σ' that selects only the actions of A_r in every $r \in \mathcal{S}$ has to satisfy $\mathcal{P}_{\mathfrak{p}}^{\sigma',\pi}(\mathit{Reach}(\mathcal{R} \setminus \mathcal{S})) = 0$ for all $\pi \in \Pi$ and $\mathfrak{p} \in \mathcal{S}$. To see this, realize that when we use only the actions of A_r, we do not visit (with positive probability) any other regions than we did with σ. Hence, we stay in $\mathcal{S}$ almost surely.
- Secondly, we prove that from every p-history in $\mathcal{S}$, we can reach a target region with positive probability. We proceed in several steps.
  - Let us fix a region $r \in \mathcal{S}$. Realize that then there is a p-history $\mathfrak{p} \in r$ for which σ is almost-sure winning (since σ is almost-sure winning and for every $r \in \mathcal{S}$ there is $\pi \in \Pi$ such that r is visited with positive probability, there must be a p-history $\mathfrak{p} \in r$ for which σ is almost-sure winning). In particular, $\mathcal{P}_{\mathfrak{p}}^{\sigma,\pi}(\mathit{Reach}(\mathcal{R}_T)) > 0$ for every $\pi \in \Pi$. We show how to transform σ into a DTA strategy σ' such that $\mathcal{P}_{\mathfrak{p}}^{\sigma',\pi}(\mathit{Reach}(\mathcal{R}_T)) > 0$.

    Let us first consider one-player games, i.e., the situation when $C_\Diamond = \emptyset$. Then there must be a sequence of regions $r_0, \ldots, r_n$ visited on the way from $\mathfrak{p}$ to a target, selecting some actions $a_0, \ldots, a_{n-1}$. We fix these actions for the respective regions (if some region is visited several times, we fix the last action taken) and thus obtain the desired DTA strategy σ'.

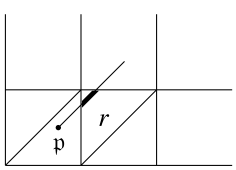
    In the general case of two-player games, we have to consider a tree of regions and actions instead of a single sequence, because every possible behaviour of the opponent in the first n steps has to be taken into account.

– Then we prove that for *every* $\mathfrak{p}' \in r$ we have that $\mathcal{P}^{\sigma',\pi}_{\mathfrak{p}'}(Reach(\mathcal{R}_T)) > 0$ for every $\pi \in \Pi$. For the p-histories $\mathfrak{p}, \mathfrak{p}' \in r$, consider the probability that taking an action a results in reaching a given region in one step. These probabilities are either both positive or both zero. This one-step qualitative equivalence is then extended to arbitrary many steps. Hence, $\mathcal{P}^{\sigma',\pi}_{\mathfrak{p}'}(Reach(\mathcal{R}_T)) > 0$.
– Let us now drop the fixed region r. We need to "stitch" the DTA strategies described above for each region into one DTA strategy σ^*. We construct σ^* as follows. In the first step, we take an arbitrary region reachable with positive probability (e.g., the initial one containing $\mathfrak{p}_0$) and fix the decisions in the regions $r_0, \ldots, r_n$ (where $r_n \in \mathcal{R}_T$) discussed above. Let us denote this set of regions by F_1. In the second step, we take an arbitrary region $v \in \mathcal{S} \setminus F_1$. Again, we take a sequence of regions $r'_0, \ldots, r'_{n'}$ ending in $\mathcal{R}_T \cup F_1$. We fix the actions in these regions accordingly and get a set F_2. We repeat this step until $F_k = \mathcal{S}$. In the other regions, σ^* is defined arbitrarily requiring only it is constant on each region. □

Proposition 5. *If a DTA strategy σ^* is a candidate on a set of regions $\mathcal{S} \subseteq \mathcal{R}$, then for every $\mathfrak{p} \in \mathcal{S}$ and every $\pi \in \Pi$ we have that $\mathcal{P}^{\sigma^*,\pi}_{\mathfrak{p}}(Reach(\mathcal{R}_T)) = 1$.*

Note that we are guaranteed that for every p-history in every region in $\mathcal{S}$, the probability of reaching a target is positive. However, it can be arbitrarily small. Therefore, even if we pass through these regions infinitely often and never leave them, it is not clear that we eventually reach a target almost surely. This would be guaranteed if the probabilities were bounded from below by a positive constant.

Remark 2. If we considered the restricted case of one-player games with bounded intervals and exponentially distributed unbounded events, we can already easily prove that σ^* is winning using [3] as follows. Fixing σ^* resolves all non-determinism and yields a system of the type considered by [3]. Since we are guaranteed the positive probability of reaching the target, we may apply Lemma 3 of [3]. However, in the setting of two-player games, we cannot use this argument directly and some (non-trivial) changes are required.



Intuitively, the reason why the probabilities of reaching a target are generally not bounded from below is that when the fractional parts of the clocks are too close, the probability of reaching a given region may approach zero. The figure on the left shows the region graph of a system with two clocks and a single state. There is also a single event, which is positive on $(0, 1)$ and its associated clock is not depicted. Now observe that if $\mathfrak{p}$ comes closer and closer to the diagonal, the probability that the (only) event happens in the region r is smaller and smaller.

Nevertheless, we can bound the probabilities if we restrict ourselves to a smaller set of positions. We define *δ-separated* parts of regions, where the differences of p-clocks are at least δ (and hence we are at least δ-away from the boundary of the region) or zero due to a synchronization of the clocks of the original automaton. Being away from the boundary by a fixed δ then guarantees that we reach the next region with a probability bounded from below.

Definition 5. *Let $\delta > 0$. We say that a set $D \subseteq \mathbb{R}_{\geq 0}$ is δ-separated if for every $x, y \in D$ either $frac(x) = frac(y)$ or $|frac(x) - frac(y)| > \delta$. Further, we say that a p-history with the last p-control (s, q, ξ) is δ-separated if the set $\{0\} \cup \{\xi(a) \mid a \in \mathcal{E} \cup \mathcal{X}, a \text{ is relevant for } \xi\}$ is δ-separated.*

Now we prove that the probabilities of reaching a target region are bounded from below if we start in a δ-separated p-history.

Proposition 6. *Let σ^* be a DTA strategy candidate on a set of regions $\mathcal{S}$. For every $\delta > 0$ there is $\varepsilon > 0$ such that for every δ-separated p-history $\mathfrak{p} \in \mathcal{S}$ and every strategy π we have that $\mathcal{P}_{\mathfrak{p}}^{\sigma^*,\pi}(Reach(\mathcal{R}_T)) > \varepsilon$.*

Proof (Sketch). We prove that for every $\delta > 0$ there is $\varepsilon > 0$ such that starting in a δ-separated p-history, the probability of reaching a target in at most $|\mathcal{R}|$ steps is greater than ε. For this we use the observation that after performing one step from a δ-separated p-history, we end up (with a probability bounded from below) in a δ'-separated p-history. This can be generalized to an arbitrary (but fixed) number of steps. Now it suffices to observe that for every $\pi \in \Pi$ and a δ-separated p-history $\mathfrak{p}$ there is a sequence of regions $r_1, \ldots, r_k$ with $k \leq |\mathcal{R}|$, such that $\mathfrak{p} \in r_1$, $r_k \in \mathcal{R}_T$, and the probability of reaching r_{i+1} from r_i in one step using σ^* and π is positive. □

Nevertheless, there is a non-zero probability of falling out of safely separated parts of regions. To finish the proof of Proposition 5, we need to know that we pass through δ-separated p-histories infinitely often almost surely (since the probability of reaching a target from δ-separated p-histories is bounded from below by Proposition 6, a target is eventually visited with probability one). For this, it suffices to prove that we eventually return to a δ-separated part almost surely. Hence, the following proposition makes our proof complete.

Proposition 7. *There is $\delta > 0$ such that for every DTA strategy $\sigma \in \Sigma$ and every $\pi \in \Pi$, a δ-separated p-history is reached almost surely from every p-history $\mathfrak{p}$.*

Proof (Sketch). We prove that there are $n \in \mathbb{N}$, $\delta > 0$, and $\varepsilon > 0$ such that for every p-history $\mathfrak{p}$ and every $\pi \in \Pi$, the probability of reaching a δ-separated p-history in n steps is greater than ε. Then, we just iterate the argument. □

3.3 The algorithm

In this section, we show that the existence of a DTA almost-sure winning strategy is decidable in exponential time, and we also show how to compute such a strategy if it exists. Due to Proposition 2, this problem can be equivalently considered in the setting of the product game $\mathcal{G}_{\mathcal{A}}$. Due to Proposition 3, an almost-sure winning DTA strategy can be constructed as a strategy that is constant on every region of $\mathcal{G}_{\mathcal{A}}$. We show that this problem can be further reduced to the problem of computing wining strategies in a *finite* stochastic game $\mathcal{G}^{\mathcal{A}}$ with reachability objectives induced by the product game $\mathcal{G}_{\mathcal{A}}$. Note that the game $\mathcal{G}^{\mathcal{A}}$ can be solved by standard methods (e.g., by computing the attractor of a target set). First, we define the game $\mathcal{G}^{\mathcal{A}}$ and show how to compute it. The complexity discussion follows.

The product $\mathcal{G}_{\mathcal{A}}$ induces a game $G^{\mathcal{A}}$ whose vertices are the regions of $\mathcal{G}_{\mathcal{A}}$ as follows. Player $\odot$, where $\odot \in \{\Box, \Diamond\}$, plays in regions $(c, q, [\xi]_{\approx})$ [1] where $c \in C_{\odot}$. In a region $r = (c, q, [\xi]_{\approx})$, she chooses an arbitrary action $a \in A(c)$ and this action a leads to a stochastic vertex $(r, a) = ((c, q, [\xi]_{\approx}), a)$. From this stochastic vertex there are transitions to all regions $r' = (c', q', [\xi']_{\approx})$, such that r' is reachable from all $\mathfrak{p} \in r$ in one step using action a with some positive probability in the product $\mathcal{G}_{\mathcal{A}}$. One of these probabilistic transitions is taken at random according to the uniform distribution. From the next region the play continues in the same manner. Player $\Box$ tries to reach the set $\mathcal{R}_T$ of target regions (which is the same as in the product game) and player $\Diamond$ tries to avoid it. We say that a strategy σ of player $\Box$ is almost-sure winning for a vertex v if she reaches $\mathcal{R}_T$ almost surely when starting from v and playing according to σ.

At first glance, it might seem surprising that we set all probability distributions in $G^{\mathcal{A}}$ as uniform. Note that in different parts of a region r, the probabilities of moving to r' are different. However, as noted in the sketch of proof of Proposition 4, they are all positive or all zero. Since we are interested only in *qualitative* reachability, this is sufficient for our purposes.

Moreover, note that since we are interested in non-zero probability behaviour, there are no transitions to regions which are reachable only with zero probability (such as when an event occurs at an integral time).

We now prove that the reduction is correct. Observe that a strategy for the product game $\mathcal{G}_{\mathcal{A}}$ which is constant on regions induces a unique positional strategy for the game $G^{\mathcal{A}}$, and vice versa. Slightly abusing the notation, we consider these strategies to be strategies in both games.

Proposition 8. *Let $\mathcal{G}$ be a game and $\mathcal{A}$ a deterministic timed automaton. For every p-history $\mathfrak{p}$ in a region r, we have that*

- *a positional strategy σ is almost-sure winning for r in $G^{\mathcal{A}}$ iff it is almost-sure winning for $\mathfrak{p}$ in $\mathcal{G}_{\mathcal{A}}$,*
- *player $\Box$ has an almost-sure winning strategy for r in $G^{\mathcal{A}}$ iff player $\Box$ has an almost-sure winning strategy for $\mathfrak{p}$ in $\mathcal{G}_{\mathcal{A}}$.*

The algorithm constructs the regions of the product $\mathcal{G}_{\mathcal{A}}$ and the induced game graph of the game $G^{\mathcal{A}}$ (see [11]). Since there are exponentially many regions (w.r.t. the number of clocks and events), the size of $G^{\mathcal{A}}$ is exponential in the size of $\mathcal{G}$ and $\mathcal{A}$. As we already noted, two-player stochastic games with qualitative reachability objectives are easily solvable in polynomial time, and thus we obtain the following:

Theorem 3. *Let $\mathfrak{h}$ be a history. The problem whether player $\Box$ has a (DTA) almost-sure winning strategy for $\mathfrak{h}$ is solvable in time exponential in $|\mathcal{G}|$ and $|\mathcal{A}|$, and polynomial in $|\mathfrak{h}|$. A DTA almost-sure winning strategy is computable in exponential time if it exists.*

4 Conclusions and Future Work

An interesting question is whether the positive results presented in this paper can be extended to more general classes of objectives that can be encoded, e.g., by determin-

[1] Note that a region is a set of p-histories such that their last p-controls share the same control c, location q, and equivalence class $[\xi]_{\approx}$. Hence, we can represent a region by a triple $(c, q, [\xi]_{\approx})$.

istic timed automata with ω-regular acceptance conditions. Another open problem are algorithmic properties of ε-optimal strategies in stochastic real-time games.

References

1. R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking for probabilistic real-time systems. In *Proceedings of ICALP'91*, volume 510 of *LNCS*, pages 115–136. Springer, 1991.
2. R. Alur, C. Courcoubetis, and D.L. Dill. Verifying automata specifications of probabilistic real-time systems. In *Real-Time: Theory in Practice*, volume 600 of *LNCS*, pages 28–44. Springer, 1992.
3. R. Alur and D. Dill. A theory of timed automata. *TCS*, 126(2):183–235, 1994.
4. C. Baier, N. Bertrand, P. Bouyer, T. Brihaye, and M. Größer. Almost-sure model checking of infinite paths in one-clock timed automata. In *Proceedings of LICS 2008*, pages 217–226. IEEE, 2008.
5. C. Baier, H. Hermanns, J.-P. Katoen, and B.R. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *TCS*, 345:2–26, 2005.
6. N. Bertrand, P. Bouyer, T. Brihaye, and N. Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics. In *Proceedings of 5th Int. Conf. on Quantitative Evaluation of Systems (QEST'08)*, pages 55–64. IEEE, 2008.
7. D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2007.
8. P. Billingsley. *Probability and Measure*. Wiley, 1995.
9. P. Bouyer and V. Forejt. Reachability in stochastic timed games. In *Proceedings of ICALP 2009*, volume 5556 of *LNCS*, pages 103–114. Springer, 2009.
10. T. Brázdil, V. Forejt, J. Krčál, J. Křetínský, and A. Kučera. Continuous-time stochastic games with time-bounded reachability. In *Proceedings of FST&TCS 2009*, volume 4 of *LIPIcs*, pages 61–72. Schloss Dagstuhl, 2009.
11. T. Brázdil, J. Krčál, J. Křetínský, A. Kučera, and V. Řehák. Stochastic real-time games with qualitative timed automata objectives. Technical report FIMU-RS-2010-05, Faculty of Informatics, Masaryk University, 2010.
12. T. Chen, T. Han, J.-P. Katoen, and A. Mereacre. Quantitative model checking of continuous-time Markov chains against timed automata specifications. In *Proceedings of LICS 2009*, pages 309–318. IEEE, 2009.
13. P.J. Haas and G.S. Shedler. Regenerative generalized semi-Markov processes. *Stochastic Models*, 3(3):409–438, 1987.
14. A. Maitra and W. Sudderth. Finitely additive stochastic games with Borel measurable pay-offs. *Int. Jour. of Game Theory*, 27:257–267, 1998.
15. D.A. Martin. The determinacy of Blackwell games. *Journal of Symbolic Logic*, 63(4):1565–1581, 1998.
16. M. Neuhäußer, M. Stoelinga, and J.-P. Katoen. Delayed nondeterminism in continuous-time Markov decision processes. In *Proceedings of FoSSaCS 2009*, volume 5504 of *LNCS*, pages 364–379. Springer, 2009.
17. J.R. Norris. *Markov Chains*. Cambridge University Press, 1998.
18. M. Rabe and S. Schewe. Optimal time-abstract schedulers for CTMDPs and Markov games. In *Eighth Workshop on Quantitative Aspects of Programming Languages*, 2010.
19. S.M. Ross. *Stochastic Processes*. Wiley, 1996.
20. R. Schassberger. Insensitivity of steady-state distributions of generalized semi-Markov processes. *Advances in Applied Probability*, 10:836–851, 1978.
21. W. Whitt. Continuity of generalized semi-Markov processes. *Mathematics of Operations Research*, 5(4):494–501, 1980.