Minimum-Time Reachability in Timed Games[⋆]

Thomas Brihaye[1], Thomas A. Henzinger[2], Vinayak S. Prabhu[3], and Jean-François Raskin[4]

[1] LSV-CNRS & ENS de Cachan; `thomas.brihaye@lsv.ens-cachan.fr`
[2]Department of Computer and Communication Sciences, EPFL; `tah@epfl.ch`
[3]Department of Electrical Engineering & Computer Sciences, UC Berkeley; `vinayak@eecs.berkeley.edu`
[4]Département d'Informatique, Université Libre de Bruxelles; `jraskin@ulb.ac.be`

Abstract. We consider the minimum-time reachability problem in concurrent two-player timed automaton game structures. We show how to compute the minimum time needed by a player to reach a target location against all possible choices of the opponent. We do not put any syntactic restriction on the game structure, nor do we require any player to guarantee time divergence. We only require players to use receptive strategies which do not block time. The minimal time is computed in part using a fixpoint expression, which we show can be evaluated on equivalence classes of a non-trivial extension of the clock-region equivalence relation for timed automata.

1 Introduction

Timed automata [3], finite-state machines enriched with clocks and clock constraints, are a well-established formalism for the modeling and analysis of timed systems. A large number of important and interesting theoretical results have been obtained on problems in the timed automaton framework. In parallel with these theoretical results, efficient verification tools have been implemented and successfully applied to industrially relevant case studies.

Timed automata are models for closed systems, where every transition is controlled. If we want to distinguish between actions of several agents (for instance, a *controller* and an *environment*), we have to consider games on timed automata, also known as *timed automaton games*. In the sequel, we will focus on two-player games. In this context, the *reachability problem* asks whether player-1 has a strategy to force the timed game to reach a target location, no matter how player-2 resolves her choices. These games were first introduced and studied in [23, 20]. In this framework, it is also natural to consider the *minimum-time reachability problem*, which asks for the minimal time required by player-1 to reach a target location, no matter how player-2 resolves her choices. This problem was first posed in [5], where it was shown to be decidable for a restricted class of timed automaton games.

[⋆] This research was supported in part by the NSF grant CCR-0225610 and by the Swiss National Science Foundation.

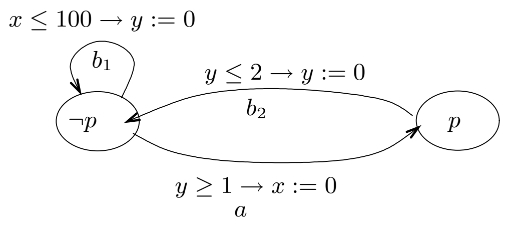
$x \leq 100 \rightarrow y := 0$
b_1
$y \leq 2 \rightarrow y := 0$
b_2
$\neg p$
p
$y \geq 1 \rightarrow x := 0$
a

Fig. 1. A timed automaton game.

Any formalism involving timed systems has to face the problem of *zeno runs*, i.e, runs of the model where time converges. Zeno runs are not physically meaningful. The avoidance of such runs has often been achieved by putting syntactic constraints on the cycles of timed automaton games [21, 5, 16, 6], or by semantic conditions that discretize time [17]. Other works on the existence of controllers [20, 15, 7, 10] have required that time divergence be ensured by the controller — a one-sided, unfair view in settings where the player modeling the environment can also block time.

Recently, a more equitable treatment of zeno runs has been proposed in [13]. This setting formulates a symmetric set-up of the model, where both players are given equally powerful options for updating the state of the game, advancing time, or blocking time. Both players may block time, however, for a player to win for an objective, she must not be *responsible* for preventing time from diverging. It has been shown in [18] that this is equivalent to requiring that the players use only *receptive* strategies [4, 22], which do not prevent time from diverging.

Example. Consider the game depicted in Figure 1. Let edge a be controlled by player-1; the others being controlled by player-2. Suppose we want to know what is the earliest time that player-1 can reach p starting from the state $\langle \neg p, x = 0, y = 0 \rangle$ (i.e., the initial values of both clocks x and y are 0). Player-1 is not able to guarantee time divergence, as player-2 can keep on choosing the edge b_1. On the other hand, we do not want to put any restriction of the number of times that player-2 chooses b_1. Requiring that the players use only non-zeno strategies avoids such unnecessary restrictions, and gives the correct minimum time for player-1 to reach p, namely, 101 time units.

Contribution. We consider the *minimum-time reachability problem (for timed automaton games)*, in the framework of [13, 18]. We present an EXPTIME algorithm to compute the minimum time needed by player-1 to force the game into a target location, with both players restricted to using only receptive strategies (note that reachability in timed automaton games is EXPTIME-complete [17]). The proof technique builds on techniques from [13, 18]. We first show that the minimum time can be obtained by solving a certain μ-calculus fixpoint equation. We then give a proof of termination for the fixpoint evaluation. This requires an important new ingredient: an extension of the clock-region equivalence [3] for timed automata. We show our extended region equivalence classes to be stable with respect to the monotone functions used in the fixpoint equation. Using results from [18], we manage to restrict the fixpoint computation to finitely many regions and thus guarantee termination.

We note that standard clock regions do not suffice for the solution. The minimum-time reachability game has two components: a reachability part that can be handled by discrete arguments based on the clock-region graph; and a minimum-time part that requires minimization within clock regions (cf. [12]). Unfortunately, both arguments are intertwined and cannot be considered in isolation. Our extended regions decouple the two parts in the proofs. We also note that region sequences that correspond to time-minimal runs may in general be required to contain region cycles in which time does not progress by an integer amount; thus a reduction to a loop-free region game, as in [1], is not possible.

Related work. Only special cases of the minimum-time reachability problem have been solved before: [5] restricts attention to the case where every cycle of the timed automaton ensures syntactically that a positive amount of time passes (a.k.a. strong non-zenoness assumption); [2] considers timed automaton games that are restricted to a bounded number of moves; [18] presents an approximate computation of the minimum time (computation of the exact minimum time being left open). The general case for *weighted* timed automaton games (timed automaton games augmented with costs on discrete transitions and cost rates on locations) is undecidable [8]. The recent work of [19] presents a strategy improvement algorithm that computes the minimum time in all timed automaton games, but it does not require strategies to be receptive. Average-reward games in the framework of [13] are considered in [1], but with the durations of time moves restricted to either 0 or 1. The non-game version of the minimum-time reachability problem is solved in [12].

Outline. In Section 2, we recall the definitions of the timed games framework from [13]. The minimum-time reachability problem is defined in Section 3. Section 4 gives an algorithm that computes the minimum time in timed automaton games. The algorithm runs in time exponential in the number of clocks and the size of clock constraints. Proofs can be found in [9]

2 Timed Games

2.1 Timed Game Structures

We use the formalism of [13]. A *timed game structure* is a tuple $\mathcal{G} = \langle S, \Sigma, \sigma, A_1, A_2, \Gamma_1, \Gamma_2, \delta \rangle$ with the following components:

- S is a set of states.
- Σ is a finite set of propositions.
- $\sigma : S \mapsto 2^{\Sigma}$ is the observation map, which assigns to every state the set of propositions that are true in that state.
- A_1 and A_2 are two disjoint sets of actions for players 1 and 2, respectively. We assume that $\perp_i \notin A_i$, and write $A_i^{\perp}$ for $A_i \cup \{\perp_i\}$. We also assume $A_1^{\perp}$

and $A_2^\perp$ to be disjoint. The set of *moves* for player i is $M_i = \mathbb{R}_{\geq 0} \times A_i^\perp$. Intuitively, a move $\langle \Delta, a_i \rangle$ by player i indicates a waiting period of Δ time units followed by a discrete transition labeled with action a_i.

– $\Gamma_i : S \mapsto 2^{M_i} \setminus \emptyset$ are two move assignments. At every state s, the set $\Gamma_i(s)$ contains the moves that are available to player i. We require that $\langle 0, \perp_i \rangle \in \Gamma_i(s)$ for all states $s \in S$ and $i \in \{1, 2\}$. Intuitively, $\langle 0, \perp_i \rangle$ is a time-blocking stutter move.

– $\delta : S \times (M_1 \cup M_2) \mapsto S$ is the transition function. We require that for all time delays $\Delta, \Delta' \in \mathbb{R}_{\geq 0}$ with $\Delta' \leq \Delta$, and all actions $a_i \in A_i^\perp$, we have (1) $\langle \Delta, a_i \rangle \in \Gamma_i(s)$ iff both $\langle \Delta', \perp_i \rangle \in \Gamma_i(s)$ and $\langle \Delta - \Delta', a_i \rangle \in \Gamma_i(\delta(s, \langle \Delta', \perp_i \rangle))$; and (2) if $\delta(s, \langle \Delta', \perp_i \rangle) = s'$ and $\delta(s', \langle \Delta - \Delta', a_i \rangle) = s''$, then $\delta(s, \langle \Delta, a_i \rangle) = s''$.

The game proceeds as follows. If the current state of the game is s, then both players simultaneously propose moves $\langle \Delta_1, a_1 \rangle \in \Gamma_1(s)$ and $\langle \Delta_2, a_2 \rangle \in \Gamma_2(s)$. The move with the shorter duration "wins" in determining the next state of the game. If both moves have the same duration, then one of the two moves is chosen non-deterministically. Formally, we define the *joint destination function* $\delta_{\mathsf{jd}} : S \times M_1 \times M_2 \mapsto 2^S$ by

$$\delta_{\mathsf{jd}}(s, \langle \Delta_1, a_1 \rangle, \langle \Delta_2, a_2 \rangle) = \begin{cases} \{\delta(s, \langle \Delta_1, a_1 \rangle)\} & \text{if } \Delta_1 < \Delta_2; \\ \{\delta(s, \langle \Delta_2, a_2 \rangle)\} & \text{if } \Delta_2 < \Delta_1; \\ \{\delta(s, \langle \Delta_1, a_1 \rangle), \delta(s, \langle \Delta_2, a_2 \rangle)\} & \text{if } \Delta_1 = \Delta_2. \end{cases}$$

The time elapsed when the moves $m_1 = \langle \Delta_1, a_1 \rangle$ and $m_2 = \langle \Delta_2, a_2 \rangle$ are proposed is given by $\mathsf{delay}(m_1, m_2) = \min(\Delta_1, \Delta_2)$. The boolean predicate $\mathsf{blame}_i(s, m_1, m_2, s')$ indicates whether player i is "responsible" for the state change from s to s' when the moves m_1 and m_2 are proposed. Denoting the opponent of player $i \in \{1, 2\}$ by $\sim\! i = 3 - i$, we define

$$\mathsf{blame}_i(s, \langle \Delta_1, a_1 \rangle, \langle \Delta_2, a_2 \rangle, s') \;=\; \big(\Delta_i \leq \Delta_{\sim i} \;\wedge\; \delta(s, \langle \Delta_i, a_i \rangle) = s'\big).$$

A *run* of the timed game structure $\mathcal{G}$ is an infinite sequence $r = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \ldots$ such that $s_k \in S$ and $m_i^k \in \Gamma_i(s_k)$ and $s_{k+1} \in \delta_{\mathsf{jd}}(s_k, m_1^k, m_2^k)$ for all $k \geq 0$ and $i \in 1, 2$. For $k \geq 0$, let $\mathsf{time}(r, k)$ denote the "time" at position k of the run, namely, $\mathsf{time}(r, k) = \sum_{j=0}^{k-1} \mathsf{delay}(m_1^j, m_2^j)$ (we let $\mathsf{time}(r, 0) = 0$). By $r[k]$ we denote the $(k+1)$-th state s_k of r. The run prefix $r[0..k]$ is the finite prefix of the run r that ends in the state s_k; we write $\mathsf{last}(r[0..k])$ for the ending state s_k of the run prefix. Let Runs be the set of all runs of $\mathcal{G}$, and let $\mathsf{FinRuns}$ be the set of run prefixes.

A *strategy* π_i for player $i \in \{1, 2\}$ is a function $\pi_i : \mathsf{FinRuns} \mapsto M_i$ that assigns to every run prefix $r[0..k]$ a move to be proposed by player i at the state $\mathsf{last}(r[0..k])$ if the history of the game is $r[0..k]$. We require that $\pi_i(r[0..k]) \in \Gamma_i(\mathsf{last}(r[0..k]))$ for every run prefix $r[0..k]$, so that strategies propose only available moves. The results of this paper are equally valid if strategies do not depend on past moves chosen by the players, but only on the past sequence of states and time delays [13]. For $i \in \{1,2\}$, let Π_i be the set of player-i strategies. Given two strategies $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2$, the set of possible *outcomes* of the game starting from a state $s \in S$ is denoted $\mathsf{Outcomes}(s, \pi_1, \pi_2)$: it contains all runs $r = s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \ldots$ such that $s_0 = s$ and for all $k \geq 0$ and $i \in \{1,2\}$, we have $\pi_i(r[0..k]) = m_i^k$.

We distinguish between *physical time* and *game time*. We allow moves with zero time delay, thus a physical time $t \in \mathbb{R}_{\geq 0}$ may correspond to several linearly ordered states, to which we assign the game times $\langle t, 0\rangle, \langle t, 1\rangle, \langle t, 2\rangle, \ldots$ For a run $r \in \mathsf{Runs}$, we define the set of game times as

$$\mathsf{GameTimes}(r) \;=\; \begin{array}{l} \{\langle t,k\rangle \in \mathbb{R}_{\geq 0} \times \mathbb{N} \mid 0 \leq k < |\{j \geq 0 \mid \mathsf{time}(r,j) = t\}|\} \;\cup \\ \{\langle t,0\rangle \mid \mathsf{time}(r,j) \geq t \text{ for some } j \geq 0\}. \end{array}$$

The state of the run r at a game time $\langle t,k\rangle \in \mathsf{GameTimes}(r)$ is defined as

$$\mathsf{state}(r, \langle t,k\rangle) \;=\; \begin{cases} r[j+k] \;\; \text{if } \mathsf{time}(r,j) = t \text{ and for all } j' < j, \mathsf{time}(r,j') < t; \\ \delta(r[j], \langle t - \mathsf{time}(r,j), \bot_i\rangle) \;\; \text{if } \mathsf{time}(r,j) < t < \mathsf{time}(r,j+1) \text{ and} \\ \qquad\qquad r[0..j+1] = r[0..j], \langle m_1^j, m_2^j\rangle, r[j+1] \text{ and} \\ \qquad\qquad \mathsf{blame}_i(r[j], m_1^j, m_2^j, r[j+1]) \end{cases}$$

Note that if r is a run of the timed game structure $\mathcal{G}$, and $\mathsf{time}(r,j) < t < \mathsf{time}(r,j+1)$, then $\delta(r[j], \langle t - \mathsf{time}(r,j), \bot_i\rangle)$ is a state in S, namely, the state that results from $r[j]$ by letting time $t - \mathsf{time}(r,j)$ pass. We say that the run r *visits* a set $X \subseteq S$ at time t if there is a $\tau = \langle t,k\rangle \in \mathsf{GameTimes}(r)$ such that $\mathsf{state}(r,\tau) \in X$. A run r visits a proposition $p \in \Sigma$ if it visits the set S_p defined as $\{s \mid p \in \sigma(s)\}$.

2.2 Timed Automaton Games

Timed automata [3] suggest a finite syntax for specifying infinite-state timed game structures. A *timed automaton game* is a tuple $\mathcal{T} = \langle L, \Sigma, \sigma, C, A_1, A_2, E, \gamma\rangle$ with the following components:

- L is a finite set of locations.
- Σ is a finite set of propositions.
- $\sigma : L \mapsto 2^{\Sigma}$ assigns to every location a set of propositions.
- C is a finite set of clocks. We assume that $z \in C$ for the unresettable clock z, which is used to measure the time elapsed since the start of the game.
- A_1 and A_2 are two disjoint sets of actions for players 1 and 2, respectively.
- $E \subseteq L \times (A_1 \cup A_2) \times \mathsf{Constr}(C) \times L \times 2^{C \setminus \{z\}}$ is the edge relation, where the set $\mathsf{Constr}(C)$ of *clock constraints* is generated by the grammar

$$\theta ::= x \leq d \mid d \leq x \mid \neg\theta \mid \theta_1 \wedge \theta_2$$

  for clock variables $x \in C$ and nonnegative integer constants d. For an edge $e = \langle l, a_i, \theta, l', \lambda\rangle$, the clock constraint θ acts as a guard on the clock values which specifies when the edge e can be taken, and by taking the edge e, the clocks in the set $\lambda \subseteq C \backslash \{z\}$ are reset to 0. We require that for all edges $\langle l, a_i, \theta', l', \lambda' \rangle, \langle l, a_i, \theta'', l'', \lambda'' \rangle \in E$ with $l' \neq l''$, the conjunction $\theta' \wedge \theta''$ is unsatisfiable. This requirement ensures that a state and a move together uniquely determine a successor state.

– $\gamma : L \mapsto \mathsf{Constr}(C)$ is a function that assigns to every location an invariant for both players. All clocks increase uniformly at the same rate. When at location l, each player i must propose a move out of l before the invariant $\gamma(l)$ expires. Thus, the game can stay at a location only as long as the invariant is satisfied by the clock values.

A *clock valuation* is a function $\kappa : C \mapsto \mathbb{R}_{\geq 0}$ that maps every clock to a non-negative real. The set of all clock valuations for C is denoted by $K(C)$. Given a clock valuation $\kappa \in K(C)$ and a time delay $\varDelta \in \mathbb{R}_{\geq 0}$, we write $\kappa + \varDelta$ for the clock valuation in $K(C)$ defined by $(\kappa + \varDelta)(x) = \kappa(x) + \varDelta$ for all clocks $x \in C$. For a subset $\lambda \subseteq C$ of the clocks, we write $\kappa[\lambda := 0]$ for the clock valuation in $K(C)$ defined by $(\kappa[\lambda := 0])(x) = 0$ if $x \in \lambda$, and $(\kappa[\lambda := 0])(x) = \kappa(x)$ if $x \notin \lambda$. A clock valuation $\kappa \in K(C)$ *satisfies* the clock constraint $\theta \in \mathsf{Constr}(C)$, written $\kappa \models \theta$, if the condition θ holds when all clocks in C take on the values specified by κ.

A *state* $s = \langle l, \kappa \rangle$ of the timed automaton game $\mathcal{T}$ is a location $l \in L$ together with a clock valuation $\kappa \in K(C)$ such that the invariant at the location is satisfied, that is, $\kappa \models \gamma(l)$. Let S be the set of all states of $\mathcal{T}$. In a state, each player i proposes a time delay allowed by the invariant map γ, together either with the action $\perp$, or with an action $a_i \in A_i$ such that an edge labeled a_i is enabled after the proposed time delay. We require that for $i \in \{1, 2\}$ and for all states $s = \langle l, \kappa \rangle$, if $\kappa \models \gamma(l)$, either $\kappa + \varDelta \models \gamma(l)$ for all $\varDelta \in \mathbb{R}_{\geq 0}$, or there exist a time delay $\varDelta \in \mathbb{R}_{\geq 0}$ and an edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ such that (1) $a_i \in A_i$ and (2) $\kappa + \varDelta \models \theta$ and for all $0 \leq \varDelta' \leq \varDelta$, we have $\kappa + \varDelta' \models \gamma(l)$, and (3) $(\kappa + \varDelta)[\lambda := 0] \models \gamma(l')$.

The timed automaton game $\mathcal{T}$ defines the following timed game structure $[\![\mathcal{T}]\!] = \langle S, \varSigma, \sigma^*, A_1, A_2, \varGamma_1, \varGamma_2, \delta \rangle$:

– S is defined above.
– $\sigma^*(\langle l, \kappa \rangle) = \sigma(l)$.
– For $i \in \{1, 2\}$, the set $\varGamma_i(\langle l, \kappa \rangle)$ contains the following elements:
  1. $\langle \varDelta, \perp_i \rangle$ if for all $0 \leq \varDelta' \leq \varDelta$, we have $\kappa + \varDelta' \models \gamma(l)$.
  2. $\langle \varDelta, a_i \rangle$ if for all $0 \leq \varDelta' \leq \varDelta$, we have $\kappa + \varDelta' \models \gamma(l)$, and $a_i \in A_i$, and there exists an edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ such that $\kappa + \varDelta \models \theta$.
– $\delta(\langle l, \kappa \rangle, \langle \varDelta, \perp_i \rangle) = \langle l, \kappa + \varDelta \rangle$, and $\delta(\langle l, \kappa \rangle, \langle \varDelta, a_i \rangle) = \langle l', (\kappa + \varDelta)[\lambda := 0] \rangle$ for the unique edge $\langle l, a_i, \theta, l', \lambda \rangle \in E$ with $\kappa + \varDelta \models \theta$.

2.3 Clock Regions

Timed automaton games can be solved using a region construction from the theory of timed automata [3]. For a real $t \geq 0$, let $\mathsf{frac}(t) = t - \lfloor t \rfloor$ denote the fractional part of t. Given a timed automaton game $\mathcal{T}$, for each clock $x \in C$, let c_x denote the largest integer constant that appears in any clock constraint involving x in $\mathcal{T}$ Two clock valuations $\kappa_1, \kappa_2 \in K(C)$ are *clock-region equivalent*, denoted $\kappa_1 \cong \kappa_2$, if the following three conditions hold:

1. For all $x \in C$, either $\lfloor \kappa_1(x) \rfloor = \lfloor \kappa_2(x) \rfloor$, or both $\lfloor \kappa_1(x) \rfloor > c_x$, $\lfloor \kappa_2(x) \rfloor > c_x$.
2. For all $x, y \in C$ with $\kappa_1(x) \leq c_x$ and $\kappa_1(y) \leq c_y$, we have $\mathsf{frac}(\kappa_1(x)) \leq \mathsf{frac}(\kappa_1(y))$ iff $\mathsf{frac}(\kappa_2(x)) \leq \mathsf{frac}(\kappa_2(y))$.
3. For all $x \in C$ with $\kappa_1(x) \leq c_x$, we have $\mathsf{frac}(\kappa_1(x)) = 0$ iff $\mathsf{frac}(\kappa_2(x)) = 0$.

Two states $\langle l_1, \kappa_1 \rangle, \langle l_2, \kappa_2 \rangle \in S$ are *clock-region equivalent*, denoted $\langle l_1, \kappa_1 \rangle \cong \langle l_2, \kappa_2 \rangle$, iff $l_1 = l_2$ and $\kappa_1 \cong \kappa_2$. It is not difficult to see that $\cong$ is an equivalence relation on S. A *clock region* is an equivalence class with respect to $\cong$. There are finitely many clock regions; more precisely, the number of clock regions is bounded by $|L| \cdot \prod_{x \in C}(c_x + 1) \cdot |C|! \cdot 2^{|C|}$. For a state $s \in S$, we write $[s] \subseteq S$ for the clock region containing s. These clock regions induce a time-abstract bisimulation.

3 The Minimum-Time Reachability Problem

Given a state s and a target proposition $p \in \Sigma$ in a timed game structure $\mathcal{G}$, the *reachability* problem is to determine whether starting from s, player-1 has a strategy for visiting the proposition p. We must make sure that player-2 does not prevent player-1 from reaching a target state by blocking time. We also require player-1 to not block time as it can lead to physically unmeaningful plays. These requirements can be achieved by requiring strategies to be *receptive* [22, 4]. Formally, we first define the following two sets of runs:

- $\mathsf{Timediv} \subseteq \mathsf{Runs}$ is the set of all time-divergent runs. A run r is *time-divergent* if $\lim_{k \to \infty} \mathsf{time}(r, k) = \infty$.
- $\mathsf{Blameless}_i \subseteq \mathsf{Runs}$ is the set of runs in which player i is responsible only for finitely many transitions. A run $s_0, \langle m_1^0, m_2^0 \rangle, s_1, \langle m_1^1, m_2^1 \rangle, \ldots$ belongs to the set $\mathsf{Blameless}_i$, for $i = \{1, 2\}$, if there exists a $k \geq 0$ such that for all $j \geq k$, we have $\neg\, \mathsf{blame}_i(s_j, m_1^j, m_2^j, s_{j+1})$.

A strategy π_i for player $i \in \{1, 2\}$ is *receptive* if for all opposing strategies $\pi_{\sim i}$, and all states $s \in S$, $\mathsf{Outcomes}(s, \pi_1, \pi_2) \subseteq \mathsf{Timediv} \cup \mathsf{Blameless}_i$. Thus, no what matter what the opponent does, a receptive player-i strategy should not be responsible for blocking time. Strategies that are not receptive are not physically meaningful (note that receptiveness is not sufficient for a strategy to be physically meaningful, see [11]). For $i \in \{1, 2\}$, let Π_i^R be the set of player-i receptive strategies. A timed game structure is *well-formed* if both players have receptive strategies. We restrict our attention to well-formed timed game structures. Well-formedness of timed automaton games can be checked for (see [18]).

We say player-1 *wins* for the reachability objective p at state s, denoted $s \in \langle\langle 1 \rangle\rangle \Diamond p$, if he has a receptive strategy π_1 such that for all player-2 receptive strategies π_2, we have that all runs $r \in \mathsf{Outcomes}(s, \pi_1, \pi_2)$ visit p.

Equivalently [18], we can define player-1 to be winning for the reachability objective p at state s if he has a strategy π_1 such that for all player-2 strategies π_2, for all runs $r \in \mathsf{Outcomes}(s, \pi_1, \pi_2)$:

– if $r \in \mathsf{Timediv}$, then r visits the proposition p;
– if $r \notin \mathsf{Timediv}$, then $r \in \mathsf{Blameless}_1$.

The *minimum-time reachability problem* is to determine the minimal time in which a player can force the game into a set of target states, using only receptive strategies. Formally, given a timed game structure $\mathcal{G}$, a target proposition $p \in \Sigma$, and a run r of $\mathcal{G}$, let

$$T_{\text{visit}}(\mathcal{G}, r, p) = \begin{cases} \infty & \text{if } r \text{ does not visit } p; \\ \inf\{t \in \mathbb{R}_{\geq 0} \mid p \in \sigma(\mathsf{state}(r, \langle t, k\rangle)) \text{ for some } k\} & \text{otherwise.} \end{cases}$$

The *minimal time* for player-1 to force the game from a start state $s \in S$ to a visit to p is then

$$T_{\min}(\mathcal{G}, s, p) = \inf_{\pi_1 \in \Pi_1^R} \sup_{\pi_2 \in \Pi_2^R} \sup_{r \in \mathsf{Outcomes}(s, \pi_1, \pi_2)} T_{\text{visit}}(\mathcal{G}, r, p)$$

We omit $\mathcal{G}$ when clear from the context.

4 Solving for Minimum-Time Reachability

We restrict our attention to well-formed timed automaton games. The definition of $T_{\min}$ quantifies strategies over the set of receptive strategies. Our algorithm will instead work over the set of *all* strategies. Theorem 1 presents this reduction. We will then present a game structure for the timed automaton game $\mathcal{T}$ in which $\mathsf{Timediv}$ and $\mathsf{Blameless}_1$ can be represented using Büchi and co-Büchi constraints. This builds on the framework of [13] in which a run satisfies the reachability objective p for player-1 iff it belongs in $(\mathsf{Timediv} \cap \mathsf{Reach}(p)) \cup (\neg\, \mathsf{Timediv} \cap \mathsf{Blameless}_1)$, where $\mathsf{Reach}(p)$ denotes the set of runs which visit p. In addition, our game structure will also have a backwards running clock, which will be used in the computation of the minimum time, using a μ-calculus algorithm on *extended regions*.

4.1 Allowing Players to Use All Strategies

To allow quantification over all strategies, we first modify the payoff function T_{visit}, so that players are maximally penalised on zeno runs:

$$T_{\text{visit}}^{\text{UR}}(r, p) = \begin{cases} \infty & \text{if } r \notin \mathsf{Timediv} \text{ and } r \notin \mathsf{Blameless}_i; \\ \infty & \text{if } r \in \mathsf{Timediv} \text{ and } r \text{ does not visit } p; \\ 0 & \text{if } r \notin \mathsf{Timediv} \text{ and } r \in \mathsf{Blameless}_i; \\ \inf\{t \in \mathbb{R}_{\geq 0} \mid p \in \sigma(\mathsf{state}(r, \langle t, k\rangle)) \text{ for some } k\} & \text{otherwise.} \end{cases}$$

It turns out that penalizing on zeno-runs is equivalent to penalising on non-receptive strategies:

Theorem 1. *Let s be a state and p a proposition in a well-formed timed game structure $\mathcal{G}$. Then:*

$$T_{\min}(s,p) = \inf_{\pi_1 \in \Pi_1} \sup_{\pi_2 \in \Pi_2} \sup_{r \in \mathsf{Outcomes}(s,\pi_1,\pi_2)} T^{\mathrm{UR}}_{\mathrm{visit}}(r,p)$$

4.2 Reduction to Reachability with Büchi and co-Büchi Constraints

We now decouple reachability from optimizing for minimal time, and show how reachability with time divergence can be solved for, using an appropriately chosen μ-calculus fixpoint.

Lemma 1 ([18]). *Given a state s, and a proposition p of a well-formed timed automaton game $\mathcal{T}$, 1)we can determine if $T_{\min}(s,p) < \infty$, and 2) If $T_{\min}(s,p) < \infty$, then $T_{\min}(s,p) < M = 8|L| \cdot \prod_{x \in C}(c_x + 1) \cdot |C+1|! \cdot 2^{|C|}$. This upper bound is the same for all $s' \cong s$.*

Let M be the upper bound on $T_{\min}(s,p)$ as in Lemma 1 if $T_{\min}(s,p) < \infty$, and $M = 1$ otherwise. For a number N, let $\mathbb{R}_{[0,N]}$ and $\mathbb{R}_{[0,N)}$ denote $\mathbb{R} \cap [0,N]$ and $\mathbb{R} \cap [0,N)$ respectively. We first look at the enlarged game structure $\widehat{[\![\mathcal{T}]\!]}$ with the state space $\widehat{S} = S \times \mathbb{R}_{[0,1)} \times (\mathbb{R}_{[0,M]} \cup \{\bot\}) \times \{\text{TRUE}, \text{FALSE}\}^2$, and an augmented transition relation $\widehat{\delta} : \widehat{S} \times (M_1 \cup M_2) \mapsto \widehat{S}$. In an augmented state $\langle s, \mathfrak{z}, \beta, \mathit{tick}, \mathit{bl}_1\rangle \in \widehat{S}$, the component $s \in S$ is a state of the original game structure $[\![\mathcal{T}]\!]$, $\mathfrak{z}$ is value of a fictitious clock z which gets reset every time it hits 1, β is the value of a fictitious clock which is running *backwards*, *tick* is true iff the last transition resulted in the clock z hitting 1 (so *tick* is true iff the last transition resulted in $\mathfrak{z} = 0$), and bl_1 is true if player-1 is to blame for the last transition.

Formally, $\langle s', \mathfrak{z}', \beta', \mathit{tick}', \mathit{bl}_1'\rangle = \widehat{\delta}(\langle s, \mathfrak{z}, \beta, \mathit{tick}, \mathit{bl}_1\rangle, \langle \Delta, a_i\rangle)$ iff

1. $s' = \delta(s, \langle \Delta, a_i\rangle)$
2. $\mathfrak{z}' = (\mathfrak{z} + \Delta) \bmod 1$;
3. $\beta' = \beta \ominus \Delta$, where we define $\beta \ominus \Delta$ as $\beta - \Delta$ if $\beta \neq \bot$ and $\beta - \Delta \geq 0$, and $\bot$ otherwise ($\bot$ is an absorbing value for β).
4. $\mathit{tick}' = \text{TRUE}$ if $\mathfrak{z} + \Delta \geq 1$, and FALSE otherwise
5. $\mathit{bl}_1 = \text{TRUE}$ if $a_i \in A_1^{\bot}$ and FALSE otherwise.

Each run r of $[\![\mathcal{T}]\!]$, and values $\mathfrak{z} \in \mathbb{R}_{\geq 0}, \beta \leq M$ can be mapped to a corresponding unique run $\widehat{r}_{\mathfrak{z},\beta}$ in $\widehat{[\![\mathcal{T}]\!]}$, with $\widehat{r}_{\mathfrak{z},\beta}[0] = \langle r[0], \mathfrak{z}, \beta, \text{FALSE}, \text{FALSE}\rangle$. Similarly, each run $\widehat{r}$ of $\widehat{[\![\mathcal{T}]\!]}$ can be projected to a unique run $\widehat{r} \downarrow \mathcal{T}$ of $[\![\mathcal{T}]\!]$. It can be seen that the run r is in $\mathsf{Timediv}$ iff *tick* is true infinitely often in $\widehat{r}_{\mathfrak{z},\beta}$, and that the set $\mathsf{Blameless}_1$ corresponds to runs along which bl_1 is true only finitely often.

Lemma 2. *Given a timed game structure $[\![\mathcal{T}]\!]$, let $\widehat{X}_p = S_p \times \mathbb{R}_{[0,1)} \times \{0\} \times \{\text{TRUE}, \text{FALSE}\}^2$.*

1. *For a run r of the timed game structure $[\![\mathcal{T}]\!]$, let $T_{\mathrm{visit}}(r,p) < \infty$. Then, $T_{\mathrm{visit}}(r,p) = \inf\{\beta \mid \beta \in \mathbb{R}_{[0,M]}$ and $\widehat{r}_{0,\beta}$ visits the set $\widehat{X}_p\}$.*

2. Let $T_{\min}(s,p) < \infty$. Then,
$T_{\min}(s,p) = \inf \left\{ \beta \mid \beta \in \mathbb{R}_{[0,M]} \text{ and } \langle s, 0, \beta, \text{FALSE}, \text{FALSE} \rangle \in \langle\langle 1 \rangle\rangle \Diamond \widehat{X}_p \right\}$

3. If $T_{\min}(s,p) = \infty$, then for all β, we have $\langle s, 0, \beta, \text{FALSE}, \text{FALSE} \rangle \notin \langle\langle 1 \rangle\rangle \Diamond \widehat{X}_p$.

The rechability objective can be reduced to a parity game: each state in $\widehat{S}$ is assigned an index $\Omega : \widehat{S} \mapsto \{0,1\}$, with $\Omega(\widehat{s}) = 1$ iff $\widehat{s} \notin \widehat{X_p}$; and $tick \vee bl_1 = \text{TRUE}$. We also modify the game structure so that the states in $\widehat{X_p}$ are absorbing.

Lemma 3. *For the timed game $\widehat{[\![\mathcal{T}]\!]}$ with the reachability objective $\widehat{X}_p$, the state $\widehat{s} = \langle s, 0, \beta, \text{FALSE}, \text{FALSE} \rangle \in \langle\langle 1 \rangle\rangle \Diamond \widehat{X}_p$ iff player-1 has a strategy π_1 such that for all strategies π_2 of player-2, and all runs $\widehat{r}_{0,\beta} \in \mathsf{Outcomes}(\widehat{s}, \pi_1, \pi_2)$, the index 1 does not occur infinitely often in $\widehat{r}_{0,\beta}$.*

The fixpoint formula for solving the parity game in Lemma 3 is given by (as in [14]),

$$Y = \mu Y \nu Z \left[(\Omega^{-1}(1) \cap \mathsf{CPre}_1(Y)) \cup (\Omega^{-1}(0) \cap \mathsf{CPre}_1(Z)) \right]$$

The fixpoint expression uses the variables $Y, Z \subseteq \widehat{S}$ and the *controllable predecessor operator*, $\mathsf{CPre}_1 : 2^{\widehat{S}} \mapsto 2^{\widehat{S}}$, defined formally by $\mathsf{CPre}_1(X) \equiv \{\widehat{s} \mid \exists m_1 \in \Gamma_1(\widehat{s})\ \forall m_2 \in \Gamma_2(\widehat{s})\ (\widehat{\delta}_{\mathsf{jd}}(\widehat{s}, m_1, m_2) \subseteq X)\}$. Intuitively, $\widehat{s} \in \mathsf{CPre}_1(X)$ iff player 1 can force the augmented game from $\widehat{s}$ into X in one move.

4.3 Termination of the Fixpoint Iteration

We prove termination of the μ-calculus fixpoint iteration by demonstrating that we can work on a finite partition of the state space. Let an equivalence relation $\cong_e$ on the states in $\widehat{S}$ be defined as: $\langle \langle l^1, \kappa^1 \rangle, \mathfrak{z}^1, \beta^1, tick^1, bl_1^1 \rangle \cong_e \langle \langle l^2, \kappa^2 \rangle, \mathfrak{z}^2, \beta^2, tick^2, bl_1^2 \rangle$ iff

1. $l^1 = l^2$, $tick^1 = tick^2$, and $bl^1 = bl^2$.
2. $\widehat{\kappa^1} \cong \widehat{\kappa^2}$ where $\widehat{\kappa^i} : C \cup \{z\} \mapsto \mathbb{R}_{\geq 0}$ is a clock valuation such that $\widehat{\kappa^i}(c) = \kappa^i(c)$ for $c \in C$, $\widehat{\kappa^i}(z) = \mathfrak{z}^i$,and $c_z = 1$ (c_z is the maximum value of the clock z in the definition of $\cong$) for $i \in \{1,2\}$.
3. $\beta^1 = \bot$ iff $\beta^2 = \bot$.
4. If $\beta^1 \neq \bot, \beta^2 \neq \bot$ then
   - $\lfloor \beta^1 \rfloor = \lfloor \beta^2 \rfloor$
   - $\mathsf{frac}(\beta^1) = 0$ iff $\mathsf{frac}(\beta^2) = 0$.
   - For each clock $x \in C \cup \{z\}$ with $\kappa^1(x) \leq c_x$ and $\kappa^2(x) \leq c_x$, we have $\mathsf{frac}(\kappa^1(x)) + \mathsf{frac}(\beta^1) \sim 1$ iff $\mathsf{frac}(\kappa^2(x)) + \mathsf{frac}(\beta^2) \sim 1$ with $\sim \in \{<, =, >\}$.

The number of equivalence classes induced by $\cong_e$ is again finite $\left(O\left((|L| \cdot \prod_{x \in C} (c_x + 1) \cdot |C+1|! \cdot 2^{|C|})^2 \cdot |C| \right)\right)$. We call each equivalence class an *extended region*. An extended region Y of $\widehat{[\![\mathcal{T}]\!]}$ can be specified by the tuple $\langle l, tick, bl_1, h, \mathcal{P}, \beta_i, \beta_f, C_<, C_=, C_> \rangle$ where for a state $\widehat{s} = \langle \langle l, \kappa \rangle, \mathfrak{z}, \beta, tick, bl_1 \rangle$,

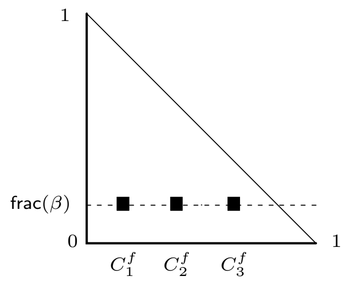

Fig. 2. An extended region with $C_< = C \cup \{z\}, C_= = \emptyset, C_> = \emptyset$

- $l, tick, bl_1$ correspond to $l, tick, bl_1$ in $\widehat{s}$.
- h is a function which specifies the integer values of clocks: $h(x) = \lfloor \kappa(x) \rfloor$ if $\kappa(x) < C_x + 1$, and $h(x) = C_x + 1$ otherwise.
- $\mathcal{P} \subseteq 2^{C \cup \{z\}}$ is a partition of the clocks $\{C_0, \ldots, C_n \mid \uplus C_i = C \cup \{z\}, C_i \neq \emptyset$ for $i > 0\}$, such that 1)for any pair of clocks x, y, we have $\mathsf{frac}(\kappa(x)) < \mathsf{frac}(\kappa(y))$ iff $x \in C_j, y \in C_k$ for $j < k$; and 2)$x \in C_0$ iff $\mathsf{frac}(\kappa(x)) = 0$.
- $\beta_i \in \mathbb{N} \cap \{0, \ldots, M\} \cup \{\perp\}$ indicates the integral value of β.
- $\beta_f \in \{\text{TRUE}, \text{FALSE}\}$ indicates whether the fractional value of β is greater than 0, $\beta_f = \text{TRUE}$ iff $\beta \neq \perp$ and $\mathsf{frac}(\beta) > 0$.
- For a clock $x \in C \cup \{z\}$ and $\beta \neq \perp$, we have $\mathsf{frac}(\kappa(x)) + \mathsf{frac}(\beta) \sim 1$ iff $x \in C_\sim$ for $\sim \in \{<, =, >\}$.

Pictorially, the relationship between $\widehat{\kappa}$ and β can be visualised as in Fig. 2. The figure depicts an extended region for $C_0 = \emptyset, \beta_i \in \mathbb{N} \cap \{0, \ldots, M\}, \beta_f = \text{TRUE}, C_< = C \cup \{z\}, C_= = \emptyset, C_> = \emptyset$. The vertical axis is used for the fractional value of β. The horizontal axis is used for the fractional values of the clocks in C_i. Thus, given a disjoint partition $\{C_0, \ldots, C_n\}$ of the clocks, we pick $n+1$ points on a line parallel to the horizontal axis, $\{\langle C_0^f, \mathsf{frac}(\beta)\rangle, \ldots, \langle C_n^f, \mathsf{frac}(\beta)\rangle\}$, with C_i^f being the fractional value of the clocks in the set C_i at $\widehat{\kappa}$.

Lemma 4. *Let $X \subseteq \widehat{S}$ consist of a union of extended regions in a timed game structure $\widehat{[\![\mathcal{T}]\!]}$. Then $\mathsf{CPre}_1(X)$ is again a union of extended regions.*

Lemma 4 demonstrates that the sets in the fixpoint computation of the μ-calculus algorithm which computes winning states for player-1 for the reachability objective $\widehat{X}_p$ consist of unions of extended regions. Since the number of extended regions is finite, the algorithm terminates.

Theorem 2. *For a state s and a proposition p in a timed automaton game $\mathcal{T}$,*

1. *The minimum time for player-1 to visit p starting from s (denoted $T_{\min}(s,p)$) is computable in time $O\left((|L| \cdot \prod_{x \in C}(c_x + 1) \cdot |C+1|! \cdot 2^{|C|})^2 \cdot |C|\right)$.*
2. *For every region R of $[\![\mathcal{T}]\!]$, either there is a constant $d_R \in \mathbb{N} \cup \{\infty\}$ such that for every state $s \in R$, we have $T_{\min}(s,p) = d_R$, or there is an integer constant d_R and a clock $x \in C$ such that for every state $s \in R$, we have $T_{\min}(s,p) = d_R - \mathsf{frac}(\kappa(x))$, where $\kappa(x)$ is the value of the clock x in s.*

References

1. B. Adler, L. de Alfaro, and M. Faella. Average reward timed games. In *FORMATS 05*, LNCS 3829, pages 65–80. Springer, 2005.
2. R. Alur, M. Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *ICALP 04*, LNCS 3142, pages 122–133. Springer, 2004.
3. R. Alur and D. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
4. R. Alur and T. Henzinger. Modularity for timed and hybrid systems. In *CONCUR 97*, LNCS 1243, pages 74–88. Springer, 1997.
5. E. Asarin and O. Maler. As soon as possible: Time optimal control for timed automata. In *HSCC 99*, LNCS 1569, pages 19–30. Springer, 1999.
6. P. Bouyer, F. Cassez, E. Fleury, and K. Larsen. Optimal strategies in priced timed game automata. In *FSTTCS 04*, LNCS 3328, pages 148–160. Springer, 2004.
7. P. Bouyer, D. D'Souza, P. Madhusudan, and A. Petit. Timed control with partial observability. In *CAV 03*, LNCS 2725, pages 180–192. Springer, 2003.
8. T. Brihaye, V. Bruyère, and J. Raskin. On optimal timed strategies. In *FORMATS 05*, LNCS 3829, pages 49–64. Springer, 2005.
9. T. Brihaye, T. Henzinger, V. Prabhu, and J. Raskin. Minimum-time reachability in timed games. *UC Berkeley Tech. Report, UCB/EECS-2007-47*, 2007.
10. F. Cassez, A. David, E. Fleury, K. Larsen, and D. Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *CONCUR 05*, LNCS 3653, pages 66–80. Springer, 2005.
11. F. Cassez, T. Henzinger, and J.-F. Raskin. A comparison of control problems for timed and hybrid systems. In *HSCC 02*, LNCS 2289, pages 134–148. Springer, 2002.
12. C. Courcoubetis and M. Yannakakis. Minimum and maximum delay problems in real-time systems. *Formal Methods in System Design*, 1(4):385–415, 1992.
13. L. de Alfaro, M. Faella, T. Henzinger, R. Majumdar, and M. Stoelinga. The element of surprise in timed games. In *CONCUR 03*, LNCS 2761, pages 144–158. Springer, 2003.
14. L. de Alfaro, T. Henzinger, and R. Majumdar. From verification to control: Dynamic programs for omega-regular objectives. In *LICS 01*, pages 279–290. IEEE Computer Society Press, 2001.
15. D. D'Souza and P. Madhusudan. Timed control synthesis for external specifications. In *STACS 02*, LNCS 2285, pages 571–582. Springer, 2002.
16. M. Faella, S. L. Torre, and A. Murano. Dense real-time games. In *LICS 02*, pages 167–176. IEEE Computer Society, 2002.
17. T. Henzinger and P. Kopke. Discrete-time control for rectangular hybrid automata. *Theoretical Computer Science*, 221:369–392, 1999.
18. T. Henzinger and V. Prabhu. Timed alternating-time temporal logic. In *FORMATS 06*, LNCS 4202, pages 1–17. Springer, 2006.
19. M. Jurdziński and A. Trivedi. Reachability-time games on timed automata. In *ICALP 07*, LNCS. Springer, 2007.
20. O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems (an extended abstract). In *STACS 95*, pages 229–242, 1995.
21. A. Pnueli, E. Asarin, O. Maler, and J. Sifakis. Controller synthesis for timed automata. In *Proc. System Structure and Control*. Elsevier, 1998.
22. R. Segala, R. Gawlick, J. Søgaard-Andersen, and N. Lynch. Liveness in timed and untimed systems. *Inf. Comput.*, 141(2):119–171, 1998.

23. H. Wong-Toi and G. Hoffmann. The control of dense real-time discrete event systems. In *Proc. of 30th Conf. Decision and Control*, pages 1527–1528, 1991.