MASTER RESEARCH INTERNSHIP

INTERNSHIP REPORT

Stochastic Strategies in Quantitative and Timed Games

Domain: Computer Science and Game Theory

Author:
Julie PARREAUX

Supervisors:
Pierre–Alain REYNIER
Benjamin MONMEGE
Modelisation and Verification - LIS

Abstract: Game theory is well established in the construction of complex reactive systems correct by construction. Two-player games model the interaction between two antagonistic agents on a system: an environment and a controller. The synthesis problem requires the construction of a controller, ensuring the system's specifications regardless how the environment reacts. Game theory models it by the synthesis problem of a winning strategy for the controller.
During the internship, we study the tradeoff between memory and randomness for strategies in two classes of games: quantitative games and weighted timed games. We show that in the context of quantitative games, randomization brings no advantages to the controller. In particular, we build an ε-optimal strategy, when it exists, whose probabilities are parameterized by ε. Also, we characterize and decide in polynomial time the class of quantitative games for which the randomized strategies are optimal. Finally, we give the first elements to extend these results to the timed setting.

Contents

1 Introduction

Verification of complex reactive systems is difficult. Detecting, finding and debugging an error are three difficult sub-problems of verification. *Game theory on graphs* is another approach to this problem designing correct systems by construction. A controller ensuring the system's specifications dealing with a potentially hostile environment must ensure them regardless of the environment's actions. Building such a controller for a given system model is the *synthesis problem*. Games on graphs model the system behaviour by interactions between two antagonistic agents (called players): environment and controller, to solve the synthesis problem of controllers.

A two-player game models the interaction between two players: Eve and Adam, with antagonistic objectives. It includes a graph (generally called an arena) and an objective. A graph specifying the available behaviours of the players gives the rules of the game. An objective specifying the system defines the winner. We assume that games have no ties: there always exists a winner, the specifications are satisfied or not. Given a graph and an objective and a given initial vertex, we want to decide the winner. Also, for the winner, we are looking to see if there exists a way to ensure he wins (generally called a winning strategy). As the system specification is encoded in the objective on a game, the synthesis problem can be approached by the synthesis problem of winning strategy.

There exist many graphs (finite or infinite, with or without weights, probability distributions on transitions) modelling different parameters on these systems. In this internship, we will focus on *quantitative and weighted timed games*, respectively, defined by finite or infinite weighted graphs. Quantitative games can model quantitative parameters such as energy consumption or functional needs of the system. Timed games, that are infinite games, model timed reactive systems or systems with time issues. However, they can be finitely represented with a timed automaton which extends a finite automaton with time. When we add quantitative parameters to these timed automata, we get a weighted timed automata to describe weighted timed games.

Many possible objectives describe the distinct properties of the system. The system seeks, for example, to reach a system's state with the reachability objective, or to avoid a set of states with safety objective. These qualitative objectives, as well as their extensions with the ω-regular objectives, have already been well studied [19]. In addition to these qualitative objectives, more quantitative objectives are useful for selecting a particular strategy among all those which are correct regarding a qualitative objective. Some measures of interest, mainly studied in the literature on quantitative game theory [21], are the average gain (called mean-payoff), the anticipated gain (called discounted-payoff) or the total gain (called total-payoff).

The combination of quantitative and qualitative objectives allows us to select a good strategy from among those valid for the selected metric. One of the simplest combinations consists of shortest-path objectives [7] combining a reachability objective with total-payoff. It requires to reach a state of the system while minimizing the total weight. Another interesting combination is a parity objective (modelling all omega-regular objectives) with mean-payoff. It requires a controller of good average quality over the long-term for any omega-regular objectives. In this internship, we focus on the shortest-path objective for quantitative and weighted timed games.

For a given objective, we will focus on a particular problem: the computation of the *value* allowing a better idea of the interest of a player in a quantitative game. It defines from a vertex the best possible cost of the system execution for a player. We then define the value problem for Adam (respectively Eve) such that for a given vertex, its value is $\leqslant c$ (respectively $\geqslant c$), for c a fixed threshold. When the value is known, we can focus on the *optimal strategy* ensuring to get this value. The existence problem asks if there exists a strategy for Adam (respectively Eve) such that whatever the strategy of the adversary, the system execution has a total weight $\leqslant c$ (respectively $\geqslant c$), for c a fixed threshold. To solve this problem, it usually studies the synthesis problem of an optimal strategy consisting in computing it.

We reduce properties of the strategy by allowing the use or not of memory or randomness. We study the existence problem of an optimal (or winning) strategy by choosing its properties. For most of classical qualitative and quantitative objectives, controller admits strategies without memory nor randomness allowing them to win or play optimally. The situation is more complex when combining qualitative and quantitative objectives. For some of these, at least one of the players may need memory to play optimally. The shortest-path objective is the first example where the controller uses memory to play optimally [7]. Controllers with a combined objective of parity and mean-payoff need memory, and even infinite memory, to play optimally [16].

Randomness in strategies is often crucial in game theory. For example, in matrix games (like rock-paper-scissors) Nash equilibria are only ensured when players can play at random [22]. To our knowledge, in the context of two-player games on graphs without randomness, such strategies have been very little used [14, 17]. Indeed, strategies without randomness, called deterministic, are enough to play optimally or win. However, they have a natural place in stochastic games [9, 14, 15].

In this internship, we will focus on the shortest-path quantitative games where the controller may need (pseudo-polynomial) memory to play optimally. Our goal is then to simulate this memory using randomness, bringing us back to the problem of the stochastic shortest-path already well studied [1, 3, 18]. We then study the implication of using randomness without memory on the value. We, therefore, study the tradeoff between memory and randomness in strategies for shortest-path games with integer weights. Some previous works have studied such a tradeoff for qualitative stochastic games [14] and even in timed games [17].

We are particularly interested in two sets of strategies: deterministic strategies with memory, as well as randomised strategies without memory. Our contribution shows that the controller can reach the same value for both sets of strategies. In particular, we compute a randomised ε-optimal strategy without memory, when it exists, where probabilities are parametrized by ε. Also, we simulate a probabilistic strategy with a deterministic strategy. We also show that for some games, no optimal randomized strategies exist. We characterize, and decide in polynomial time, the class of games admitting an optimal randomized strategy without memory.

We then study how to extend our results in quantitative games to weighted timed games. However, in weighted timed games, the problem of the value is undecidable. Thus, we focus on a class of decidable game: the divergent games. In this class of games, we define the notion of randomized strategies without memory. Then, we conjecture that the value defined by these randomized strategies is the same as that value for deterministic strategies.

In this report, we start by introducing the basic notions of game theory in Section 2. We also focus on basic games like stochastic games or timed games which are an example of infinite games. In Section 3, we introduce quantitative games. After studying their classical objectives, we focus on the shortest-path games and their deterministic strategies or strategies without memory. These different notions allow us to present our main contribution in Section 4. In Section 5, we extend quantitative games to weighted timed games. We study the decidable classes of games for the value problem. Finally, in Section 6, we focus on divergent games. We present a proof scheme allowing to extend the result of Section 4 in the context of these games.

2 Game Theory on Graph

In this section, we formally introduce some notions, as winning strategies, on game theory. Moreover, we present some particular classes of games: stochastic games (Section 2.3) and timed games (Section 2.4). We denote by $\mathbb{Z}$ the set of integers and $\mathbb{Z}_\infty = \mathbb{Z} \cup \{-\infty, +\infty\}$. For a finite set V, we denote by $\Delta(V)$ the set of *distributions* over V, which are mappings $\delta\colon V \to [0,1]$ such that $\sum_{v\in V} \delta(v) = 1$. The support of a distribution δ is the set $\{v \in V \mid \delta(v) > 0\}$, denoted by $\mathrm{supp}(\delta)$. A Dirac distribution is a distribution with singleton support: the Dirac distribution of support $\{v\}$ is denoted by Dirac_v.

2.1 Syntax and Semantics

A two-player game between Adam and Eve on a graph is composed of two elements: an arena (given by a directed graph) and a winning condition called objective. We design a particular game with these two elements.

Definition 1. A *game* on a graph is a tuple $\langle V, E, \mathsf{Win}\rangle$ where $\langle V, E\rangle$ is a directed graph defining the game arena whose transitions are given by the relation $E \subseteq V \times V$, and $\mathsf{Win} \subseteq V^\omega$ is the objective of the game.

Consider a game $\mathcal{G} = \langle V, E, \mathsf{Win}\rangle$. Its arena $\langle V, E\rangle$ can be finite or infinite as in Section 2.4 for an example. For all vertices $v \in V$, we denote $E(v) = \{v' \mid (v, v') \in E\}$ the set of adjacent vertices of v in E. A *play* is an infinite path in its arena $\pi = v_1 v_2 \ldots$ where $v_i \in E(v_{i-1})$, for all $i \in \mathbb{N}^*$. A *finite play* is a play that ends in a deadlock vertex, i.e. a vertex v such that $E(v) = \emptyset$. Without loss of generality, we can assume that all vertices in $\mathcal{G}$ are deadlock-free. Otherwise, a deadlock vertex can be transformed into an absorbent vertex by adding a self-loop. The history is a finite prefix of a play. We denote $\pi[k]$ the history of π of size $k \in \mathbb{N}$.

We consider turn-based semantics based on a single token to describe the choice of players. Its location, representing the current state of the game, gives the current player by the membership of vertex. Then the current player chooses the next location regarding E. Formally, this semantics is defined by a partition of the vertices into two sets: V_{Adam} the Adam set of vertices, and V_{Eve} the one of Eve such as $V = V_{\mathsf{Adam}} \cup V_{\mathsf{Eve}}$ and $V_{\mathsf{Adam}} \cap V_{\mathsf{Eve}} = \emptyset$. In the drawings, we depict V_{Eve} by circles and V_{Adam} by rectangles.

Example 2. *In Figure 1, if the token is in $v_0 \in V_{\mathsf{Adam}}$,* Adam *can choose to go to v_0 or v_1. Likewise, for $v_6 \in V_{\mathsf{Eve}}$,* Eve *can choose to go to v_3 or v_5. A play in this turn-based game is $\pi = v_6 v_3 v_2 (v_0 v_1)^\omega$.*

Now we define Eve's objective in a game $\mathcal{G} = \langle V, E, \mathsf{Win}\rangle$. Its objective $\mathsf{Win} \subseteq V^\omega$ characterizes winning plays for Eve. More precisely, Eve wins under Win with a play π if and only if $\pi \in \mathsf{Win}$. Also, we assume that there is no tie in $\mathcal{G}$: Adam wins when Eve loses. This assumption implies that Adam's objective is the complement of Win.

There exist many objectives, but we only focus on the reachability objective. Under this objective, Eve must reach one target's vertex, since Adam's objective is a safety objective: he must avoid all the target vertices. Formally, let $T \subseteq V$ be the target set and π play in $\mathcal{G}$, Eve wins with π if and only if there exists i such that $v_i \in T$. Otherwise, Adam wins. We denote $\langle V, E, \mathsf{Reach}(T)\rangle$ a reachability game in which T is the target set.

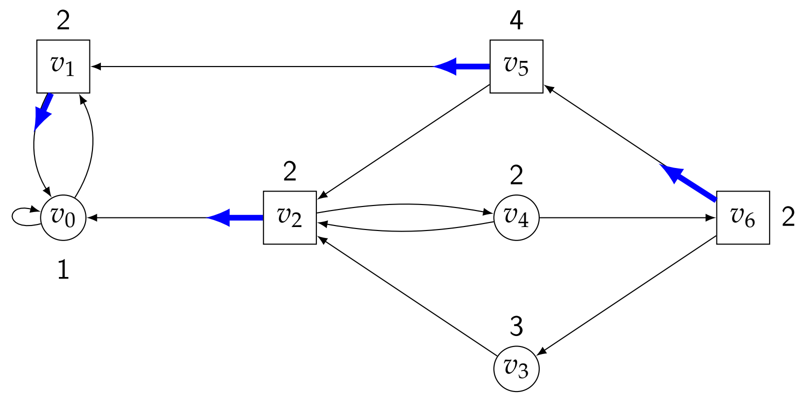


Figure 1: A ω-regular game where colours are in vertices labels.

Example 3. *Consider the reachability game represented in Figure 1 with $T = \{v_0\}$. The play $\pi = v_6v_3v_2(v_0v_1)^\omega$ is winning for* Eve*, but the play $\pi' = v_6v_3v_2(v_4v_2)^\omega$ is winning for* Adam.

Related work on ω-regular objectives ω-regular objectives naturally extend the reachability objective [19]. These objectives have a concise syntax given by some vertex colours. Formally, from a finite set of colours $\mathcal{C}$ being a subset of $\mathbb{N}$, we introduce the colouring matching $\xi : V \to \mathcal{C}$. For example, in Figure 1, $\xi(v_0) = 1$. For a play π in $\mathcal{G}$, the colour of π is an extension of the colouring matching $\xi : V^\omega \to \mathcal{C}^\omega$. The colour of play is a word (over colours) based on the colour of the vertices of play (called *ω-regular objective*). We consider $\mathcal{C} = \{0,1\}$ to express with ω-regular objective, the reachability objective for a target set T. We define $\xi(v) = 1$ if and only if $v \in T$. A play π wins if and only if its colour satisfies $\max(\xi(\pi)) = 1$. A game with an omega-regular objective of this class is a *regular game*.

Example 4. *Consider the regular game in Figure 1 with $\mathcal{C} = \{1,2,3,4\}$. Its colouring function, ξ, is given by the labels on its vertices. We then consider two plays $\pi = v_6v_3v_2(v_0)^\omega$, and $\pi' = (v_2v_4v_6v_3)^\omega$. Their colour is $\xi(\pi) = \xi(v_6v_3v_2(v_0)^\omega) = 231(1)^\omega$, and $\xi(\pi) = \xi((v_2v_4v_6v_3)^\omega) = (2223)^\omega$.*

We present two main ω-regular objectives: the Parity objective and the Müller objective. A *Parity objective* accepts a play π if its colour satisfies a parity condition. For example, we consider $p(\pi) = \min\{c \mid c \text{ appears infinitely often in } \xi(\pi)\}$ which calculates the minimum colour which occurs infinitely often in π. A play π satisfies a parity objective if and only if $p(\pi)$ is even.

Example 5. *Let us consider the regular game in Example 4 with a Parity objective given by the function p described as above.* Adam *wins with the play $\pi = v_6v_3v_2(v_0)^\omega$ because $p(\pi) = \min\{1\} = 1$ and $p(\pi)$ is odd. However,* Eve *wins with $\pi' = (v_2v_4v_6v_3)^\omega$ because $p(\pi') = \min\{2,3\} = 2$ and $p(\pi')$ is even.*

Now, consider $\mathcal{F}$ the set of subsets of $\mathcal{C}$, a Müller's objective accepts a play π if and only if its colour satisfies $\{c \mid c \text{ appears infinitely often in } \xi(\pi)\} \in \mathcal{F}$, i.e. the set colours that occur infinitely often is an element of $\mathcal{F}$.

Example 6. *Consider the regular game in Example 4 with a Müller objective given by $\mathcal{F} = \{\{1,2\},\{1,2,3,4\}\}$.* Eve *wins with the play $\pi = v_6v_3v_2(v_0)^\omega$ because $\{1,2\} \in \mathcal{F}$. However,* Eve *wins with $\pi' = (v_2v_4v_6v_3)^\omega$ because $\{2,3\} \notin \mathcal{F}$.*

2.2 Strategies

To achieve its objective in a game, Eve needs to choose a good vertex for each round. These choices are described with a *strategy*. It is a matching $\chi : V^* V_{\mathsf{Eve}} \to \Delta(V)$ mapping each history of $\mathcal{G}$ on a distribution over vertices giving the probable moves of the token. We assume for all history π and vertex $v \in V_{\mathsf{Eve}}$ that the support of $\chi(\pi v)$ is in $E(v)$. A strategy for Adam denoted ρ is defined analogously. Let π be a play, and σ be a strategy for Eve, respectively for Adam, π is conformed to σ if for all i such that $v_i \in V_{\mathsf{Eve}}$, respectively V_{Adam}, then $\sigma(v_0 v_1 \dots v_i)(v_{i+1}) > 0$. For two strategies for Eve, ρ, and Adam, χ, respectively, and a vertex v, we denote $\mathsf{Play}(v, \chi, \rho)$ the set of plays starting in v and conforming to χ and ρ.

Strategies without randomisation were first introduced. It only chooses one possible vertex to the token's move. Formally, a *deterministic* (or *pure*) strategy assigns at each history a Dirac distribution, i.e., its support is a singleton. For example, a strategy based on an attractor (see Example 8) is deterministic. We denote $\tau : V^* V_{\mathsf{Eve}} \to V$ (respectively $\sigma : V^* V_{\mathsf{Adam}} \to V$) a deterministic strategy for Eve (respectively Adam). We let $\mathsf{d}\mathcal{S}_{\mathsf{Eve}}$ and $\mathsf{d}\mathcal{S}_{\mathsf{Adam}}$ the set of deterministic strategies for players Eve and Adam, respectively.

In general, the whole history of a play is necessary to compute a strategy. However, in a majority of cases, the history is not required. A *memoryless* (or *positional*) strategy assigns for each vertex a unique distribution. In other words, for two distinct histories π and π', and $v \in V_{\mathsf{Eve}}$, a memoryless strategy σ is such that $\sigma(\pi v) = \sigma(\pi v')$. For example, a strategy based on an attractor (see Example 8) is memoryless. We denote $\chi : V_{\mathsf{Eve}} \to V$ (respectively $\rho : V_{\mathsf{Adam}} \to V$) a memoryless strategy for Eve (respectively Adam). We let $\mathsf{m}\mathcal{S}_{\mathsf{Eve}}$ and $\mathsf{m}\mathcal{S}_{\mathsf{Adam}}$ the set of memoryless strategy for players Eve and Adam, respectively.

Sometimes a bit of memory may be needed to change the strategy during the play. There exist several ways to represent this memory. We assume that a deterministic finite automaton defines the strategy.

Definition 7. A *Deterministic Finite Automaton* (DFA) is a tuple $\langle Q, \mathsf{choice}, \mathsf{update}, q_0 \rangle$ where Q is a set of finite states, $\mathsf{update} : Q \times V \to Q$ is a transition relation, $\mathsf{choice} : Q \times V \to \Delta(V)$ is a function to select the probabilistic distribution over adjacent vertices, and q_0 is the initial state.

A *finite-memory* strategy for Eve is based on a DFA $\langle Q, \mathsf{choice}, \mathsf{update}, q_0 \rangle$ that represents its memory. Thus its size is $|Q| + |\mathsf{update}| + |\mathsf{choice}|$. For all Eve's vertices and all states of Q, this strategy applies a chosen distribution and updates its memory. A finite-memory strategy for Adam is analogously defined.

For a game $\langle V, E, \mathsf{Win} \rangle$, χ is a *winning strategy* for Eve, if and only if, for all strategies for Adam ρ and $v \in V$, $\mathsf{Play}(v, \chi, \rho) \subseteq \mathsf{Win}$. For a given vertex v, Eve has a winning strategy χ from v if, for all ρ, $\mathsf{Play}(v, \chi, \rho) \subseteq \mathsf{Win}$. Moreover, a strategy χ for Eve is said *almost surely winning* if, for all strategies ρ for Adam and $v \in V_{\mathsf{Eve}}$, $\mathbb{P}_v^{\rho,\chi}(\mathsf{Win}) = 1$ where $\mathbb{P}$ is a probability measure induced by χ and ρ. We define analogously (almost) winning strategies for Adam.

Deciding on the existence of a winning strategy for each player is a classic problem in game theory. Also, if this strategy exists, we want to compute it (this is the synthesis problem). A strategy based on an attractor is an example of an elementary strategy. It is computed in Example 8 for a reachability game. Further, it is a base element for many objectives.

Example 8. *In a reachability game $\langle V, E, \mathsf{Reach}(T) \rangle$, with $T \subseteq V$ a target set, we compute a winning strategy for* Eve *in polynomial time with an attractor [19]. Intuitively, it defines the set of vertices where*

Eve *is guaranteed to win using the Pre function defined on $X \subseteq V$ a set of vertices. It calculates for X the set of vertices so that whatever* Adam*'s choice,* Eve *reaches X in one step. Formally, Pre is defined for all the subsets of vertices $X \subseteq V$, by*

$$Pre(X) = \{v \in V_{\mathsf{Eve}} \mid \exists v' \in E(v), v' \in X\} \cup \{v \in V_{\mathsf{Adam}} \mid \forall v' \in E(v'), v' \in X\}$$

The Eve *attractor for $T \subseteq V$ is the smallest fixpoint of $X \mapsto Pre(T \cup X)$ iteratively computed as follows in polynomial time. From $X_0 = \emptyset$, it is computed for each iteration with the set $X_{i+1} = Pre(T \cup X_i)$. Intuitively, at each iteration, we add to X_i all* Eve*'s vertices having an exit transition reaching $T \cup X_i$ and all* Adam*'s vertices whose all its exit transitions reach $T \cup X_i$. From each* Eve*'s vertices, the witness to reach X_i defines a winning strategy for* Eve.

Consider the reachability game in Figure 1 with $T = \{v_0\}$. Let $X_0 = \emptyset$, we describe the computation of an attractor for Eve*. For the first iteration, $X_1 = Pre(T \cup X_0) = \{v_1, v_2\}$. We note that $v_2 \in X_1$ as v_2 reaches T with (v_2, v_0), but $v_0 \notin X_1$ because all transitions from v_0 do not reach T: we can take (v_0, v_1). Then we have $X_2 = Pre(T \cup X_1) = \{v_0, v_1, v_2, v_3, v_5\}$, $X_3 = Pre(T \cup X_2) = \{v_0, v_1, v_2, v_3, v_5, v_6\}$, and $X_4 = Pre(T \cup X_3) = V$. In the last iteration, $v_4 \in X_4$ because at this time, all transitions from v_4 reach $T \cup X_3$.* Eve*'s winning strategy, τ^* depicted with blue arrows in Figure 1 is defined as $\tau^*(v_1) = v_0$, $\tau^*(v_2) = v_0$, $\tau^*(v_5) = v_1$, and $\tau^*(v_6) = v_5$.*

Deterministic strategies for regular games Randomized strategies have been first introduced in stochastic games (see Section 2.3) [14], and are not common in regular games: deterministic strategies satisfy already important properties like the existence of winning strategy and determinacy. Thus, we only consider deterministic strategies for Adam and Eve. For all vertices v, $\mathsf{Play}(v, \tau, \sigma)$ is the unique play from v conforming to $\tau \in \mathsf{d}\mathcal{S}_{\mathsf{Eve}}$ and $\sigma \in \mathsf{d}\mathcal{S}_{\mathsf{Adam}}$.

The determination of a game gives a symmetrical argument to reason about it. A game is *determined*, if and only if for all its vertices v, Eve or Adam always has a winning strategy from v. Thus, we can focus on Eve's point of view.

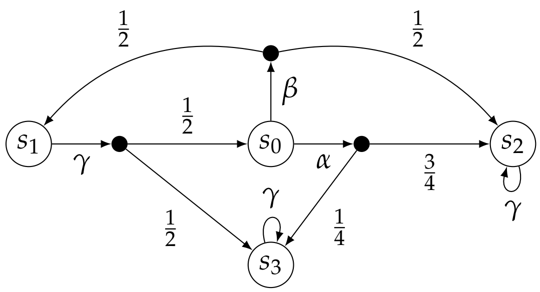
Theorem 9 ([19])**.** *Every game with an ω-regular objective is determined.*

Example 10. *Consider the reachability game in Figure 1 with $T = \{v_1\}$.* Eve*'s attractor computation converges in two iterations when we have added $\{v_5, v_6\}$. Thus,* Eve *has a winning strategy when the game starts from v_5 or v_6. In other cases, by the determinacy result (Theorem 9),* Adam *has a winning strategy.*

2.3 Markov Decision Process: an Example of Stochastic Games

When we do not know precisely the behaviour of one of the players, we can model it with probabilities. The arena will directly contain probabilities modelling its choice. The other player, therefore, plays as before. We present a natural example of stochastic games: the Markov decision process also called one and half-player games [14]. It contains a classical player and a stochastic player.

Definition 11. A *Markov Decision Process* (MDP) is a tuple $\langle S, A, P \rangle$ where S is a finite set of states, A is a finite set of actions, and $P\colon V \times A \to \Delta(V)$ is a partial function mapping some pair of vertices and actions to a distribution of probabilities over the successor vertices.



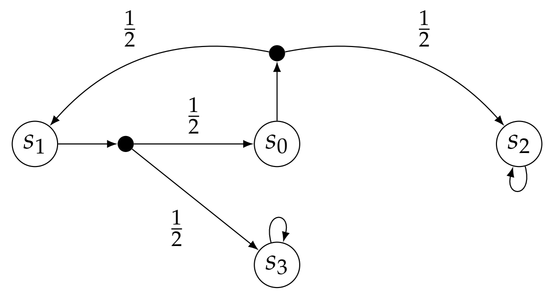


Figure 2: On the left an MDP where the player has two choices on s_0. On the right an MC induced by a memoryless deterministic strategy $\tau(s_0) = \beta$ on the MDP.

Example 12. *The MDP in Figure 2 is defined by* $S = \{s_0, s_1, s_2\}$*, and* $A = \{\alpha, \beta, \gamma\}$*. The partial function is also given by* $P(s_0, \beta)(s_1) = 1/2$*,* $P(s_0, \beta)(s_2) = 1/2$*,* $P(s_0, \alpha)(s_2) = 3/4$*,* $P(s_0, \alpha)(s_3) = 1/4$*,* $P(s_1, \gamma)(s_0) = 1/2$*,* $P(s_1, \beta)(s_0) = 1/2$*,* $P(s_2, \gamma)(s_2) = 1$*, and* $P(s_3, \gamma)(s_3) = 1$*.*

To play in an MDP, the player must choose an action. The final move for the token will then be made by the distribution of the stochastic player. A *play* is, therefore, an infinite sequence $(s_i, \alpha_i) \in (S \times A)^\omega$ such that for all i, $P(s_i, \alpha_i, s_{i+1}) > 0$. For example $\pi = (s_0, \beta)(s_1, \gamma)(s_0, \alpha)((s_0, \gamma))^\omega$ is a play in the MDP in Figure 2. A *strategy* is a mapping $\chi : V^* \to \Delta(A)$ matching all histories to a distribution over actions. We analogously define from two-player games, deterministic, memoryless or finite-memory strategies. For example χ is a deterministic memoryless strategy in the MDP in Figure 2 defined by $\chi(s_0) = \beta$, and $\chi(s_1) = \chi(s_2) = \gamma$. Moreover, we analogously define plays conformed to strategies. In an MDP, when the player has chosen its strategy, there will remain no ' 'choices" to make, and we will obtain a Markov chain. For a given strategy χ and a vertex v, $\mathsf{Play}(v, \chi)$ the set of plays conforming with χ from v induces a Markov chain. Note that $\mathsf{Play}(v, \chi)$ is a finite Markov chain if χ is memoryless.

Definition 13. A *Markov chain* (MC) is a tuple $\langle V, P \rangle$ where V is a set of vertices, and $P \colon V \to \Delta(V)$ is a function mapping each vertex to a distribution of probabilities over the successor vertices.

Example 14. *On the right of Figure 2, we have a MC induced by* χ *where* $\chi(s_0) = \beta$*, and* $\chi(s_1) = \chi(s_2) = \gamma$*. This MC is defined by* $V = \{s_0, s_1, s_2\}$*, and the function is* $P(s_0)(s_1) = 1/2$*,* $P(s_0)(s_2) = 1/2$*,* $P(s_1)(s_0) = 1/2$*,* $P(s_0)(s_3) = 1/2$*,* $P(s_2)(s_2) = 1$*, and* $P(s_3)(s_3) = 1$*.*

To compute the probabilities of reaching a target in an MDP, we begin with the computation for a given strategy χ inducing the MC, $\mathcal{M}^\chi$. From an initial vertex v we let $\mathbb{P}_v^\chi$ denote the probability measure, induced by χ, over the sets of paths in $\mathcal{M}^\chi$. A *property* is any measurable subset of finite or infinite paths in the MC with respect to the standard cylindrical sigma-algebra. For example, we denote by $\mathbb{P}_v^\rho(\diamond T)$ the probability of the set of plays that reach the target set $T \subseteq V$ of the vertices.

Example 15. *Let us compute probability to reach* $\{s_3\}$ *from* s_0 *in the MC in Figure 2. A path reaches* $\{s_3\}$ *if it makes some cycle* (s_1, s_0) *and then go to* s_3 *from* s_1*. Thus, the probability to reach* s_3 *is the probability to take the cycle* $n \in \mathbb{N}^*$ *times before to take the edge* (s_1, s_3)*. We have* $\mathbb{P}_{s_0}^\chi(\diamond s_3) = \sum_n (1/4)^n \times 1/2 = 4/3 \times 1/2 = 2/3$

To compute the probability of a property in an MDP, we use the probabilities induced by a strategy. Minimum or maximum probability of satisfying this property is then studied. These probability measures quantify over the strategies used. Formally for a property P and an initial vertex v, the maximum, respectively minimum, probability of satisfying the property is

$$\mathbb{P}_v^{\max}(P) = \sup_{\chi} \mathbb{P}_v^{\chi}(P) \qquad \text{and} \qquad \mathbb{P}_v^{\min}(P) = \inf_{\chi} \mathbb{P}_v^{\chi}(P)$$

Example 16. *In MDP in Figure 2, $\mathbb{P}_{s_0}^{\max}(\diamond\{s_3\}) = 2/3$ and $\mathbb{P}_{s_0}^{\min}(\diamond\{s_3\}) = 1/4$.*

Minimal reachability probabilities A classic problem in MDP is to compute $\mathbb{P}_v^{\max}(P)$ and $\mathbb{P}_v^{\min}(P)$ for an initial vertex v and a property P with a linear program [2]. We focus on the computation of $\mathbb{P}_v^{\min}(\diamond T)$ where T is a target set. More precisely, we want to characterize when $\mathbb{P}_v^{\min}(\diamond T) = 1$ with properties on finite MC. Note that infinite MC is not useful by the following Lemma.

Lemma 17 ([2])**.** *Let $\langle S, A, P\rangle$ be a finite MDP, $T \subset S$ and $v \in S$. There exists a memoryless strategy χ that minimizes the probabilities of reaching T, i.e. for all states s : $\mathbb{P}_v^{\chi}(\diamond T) = \mathbb{P}_v^{\min}(\diamond T)$.*

In an MC, reachability is almost surely ensured (with probability 1) for vertices included in a *bottom strongly connected component* (bottom SCC) that are (finite) sets of vertices forming a well from which it is not possible to exit. A play in a bottom SSC will, therefore, reach each vertex at least once. Also, a play in an MC will always reach a bottom SSC. Formally, let $\langle V, P\rangle$ be an MC. A subset $T \subseteq V$ is said *strongly connected* if for each $(s, t) \in T \times T$ there exists a finite path $v_0 \dots v_n$ such that $v_0 = s$, $v_n = t$ and for all $0 \leqslant i \leqslant n$, $v_i \in T$. A *strongly connected component* (SCC) denotes a strongly connected set of vertices that no proper subset of T is strongly connected. A bottom strongly connected component T is an SCC from with no vertex outside T is reachable, i.e. for each vertex $t \in T$, $\sum_{s\in T} P(s, t) = 1$.

Example 18. *Bottom SCC in the MC in Figure 2 are $\{s_2\}$ and $\{s_3\}$*

Theorem 19 ([2])**.** *For each vertex v of a finite MC: $\mathbb{P}_v\{\diamond T \mid T \text{ is a bottom SCC}\} = 1$.*

Lemma 17 guarantees that we can limit ourselves to finite MCs to calculate $\mathbb{P}^{\min}(\diamond T)$. We can, therefore, apply Theorem 19 to characterize when $\mathbb{P}^{\min}(\diamond T) = 1$.

Lemma 20 ([2])**.** *Let $\langle S, A, P\rangle$ be a finite MDP, $T \subset S$ and $v \in S$. $\mathbb{P}_v^{\min}(\diamond T) = 1$ if and only if for all χ, all reachable bottom SCC from v in induced MC contains an element of T.*

2.4 Timed Games: an Example of Infinite Games

Timed games are a particular class of infinite games. Their arena is an infinite graph whose vertices include a location, from a finite set of locations, and real values of some 'clocks". However, this graph (and therefore this game) can be finitely represented with a timed automaton. We only study these games with a reachability objective, so we call them reachability timed games.

We store the time on some variables called *clocks*. Let $\mathcal{C}$ be a set of clocks we call *valuation* a function $\nu : \mathcal{C} \to \mathbb{R}_{\geqslant 0}$ such that for all clocks $c \in \mathcal{C}$, $\nu(c)$ is the real value of c. We denote by $V(\mathcal{C})$ the set of valuations of $\mathcal{C}$. For each $c \in \mathcal{C}$, there exist two possible actions on c: let some time elapse and reset c. We suppose that time elapses at the same speed for all clocks in $\mathcal{C}$. If we let $t \in \mathbb{R}_{\geqslant 0}$

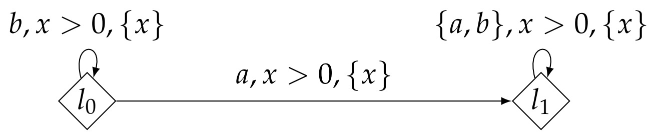


Figure 3: A concurrent timed game where Eve's action is $\{a\}$ and Adam one is $\{b\}$.

units of time elapse, we note, for all $\nu \in V(\mathcal{C})$, $(\nu + t)(c) = \nu(c) + t$. A valuation can reset some clocks of C and for any subset of clocks C, the valuation $\nu[C := 0]$ returns 0 for all $c \in C$, and $\nu(c)$ otherwise. Also, these valuations are subject to guards, which are constraints on some clocks. Let $c, c' \in \mathcal{C}, i \in \mathbb{N}$, and $\bowtie \in \{<, \leqslant, =, \geqslant, >\}$, an elementary constraint on c and c' is of the form $c \bowtie i$ or $c - c' \bowtie i$. A guard on $\mathcal{C}$ is a conjunction of elementary constraints that defines a convex set of clock valuations. We note $\mathsf{Guard}(\mathcal{C})$ this set of guards set on $\mathcal{C}$. This clock extends the notion of finite automaton to define timed automata.

Definition 21. A *timed automaton* is a tuple $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}\rangle$ where L is a finite set of locations, $\mathcal{C}$ is a finite set of clocks, Σ is a finite alphabet, $\mathfrak{d} \subseteq L \times \Sigma \times \mathsf{Guard}(\mathcal{C}) \times L \times 2^{\mathcal{C}}$ is a transitions relation, and $\mathsf{Inv} : L \to \mathsf{Guard}(\mathcal{C})$ assigns at each location an invariant given by a guard.

Example 22. *The timed automaton in Figure 3 is defined by* $L = \{l_0, l_1\}$, $\mathcal{C} = \{x\}$, $\Sigma = \{a, b\}$, $\mathfrak{d} = \{(l_0, a, \{x > 0\}, l_1, \nu[x := 0]), (l_0, b, \{x > 0\}, l_0, \nu[x := 0]), (l_1, a, \{x > 0\}, l_1, \nu[x := 0]), (l_1, b, \{x > 0\}, l_1, \nu[x := 0])\}$*, and* $\mathsf{Inv} : Q \to$ *true.*

For a timed automaton, we define a *configuration* by a location, and a valuation satisfying this invariant of location. For example, $(l_0, 0)$ or $(l_0, 0.5)$ are configurations in the timed automaton of Figure 3. A transition is *available* for a configuration when its valuation satisfies the guard, and the new invariant of location after the resetting clocks. For example, in Figure 3, from $(l_0, 0)$ no transitions are available, but from $(l_0, 0.5)$ both transitions are available. The semantics of a timed automaton is given by its configuration graph explaining all the relationships between the configurations. Note also that these graphs have infinite vertices and exit degrees.

Definition 23. The *graph of configurations* $\langle L \times \mathbb{R}^{\mathcal{C}}_{\geqslant 0}, E\rangle$ of a timed automaton $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}\rangle$ is such that its vertices are the configurations of automaton, and $\mathfrak{d}$ characterizes its transition: for two configurations (l_1, v_1) and (l_2, v_2), $((l_1, v_1), (l_2, v_2)) \in E$, if and only if, there is $a \in \Sigma$, and $g \in \mathsf{Guard}(\mathcal{C})$, such that $(l_1, a, g, l_2, v_2) \in \mathfrak{d}$.

The graph of configurations for a timed automaton is the arena of a *timed game*. In each configuration, players play with two actions: choose a delay that satisfies the invariant of location, and a transition available after this delay. One turn of a play consists of elapsing the chosen delay (it can be null) before making the transition. There exist two partitions of the vertices of arena, the graph of configurations, to choose which player plays: turn-based games, and concurrent games. A turn-based timed arena is based on a timed automaton where L is partitioned into two distinct sets L_{Adam}, and L_{Eve} as already studied before. In a turn-based reachability timed game, attractors can be adapted to work in this timed setting.

An arena *concurrent timed game* is based on a timed automaton where Σ is partitioned into two distinct sets Σ_{Adam}, and Σ_{Eve} containing the actions of each player. In these two sets, we add the action $\perp$ describing a hidden transition to stay in the same location with the same valuation (no reset nor delay). For each configuration (l, ν), we define

$$\Gamma_{\mathsf{Eve}}(l, \nu) = \{(a, t) \mid a \in \Sigma_{\mathsf{Eve}} \text{ labels an available transition from } (l, \nu) \text{ for the delay } t\} \cup \{(\perp, 0)\}$$

the set of Eve's possible choices. The set $\Gamma_{\mathsf{Adam}}(l, \nu)$ for Adam is analogously defined. On each turn, the two players simultaneously choose an element of their Γ: an action and a delay. The player who chooses the shortest delay wins this turn and applies his choice. When players choose the same delay, we use an external strategy, called a scheduler, to choose the turn winner. This game introduces an element of surprise. Note that turn-based games are a particular case of concurrent games, where for all locations, only actions of a single player are present in the location and for which the choice $(\perp, 0)$ is disallowed.

Example 24. *The action of players in the concurrent timed game in Figure 3 are* $\Sigma_{\mathsf{Eve}} = \{a\}$, *and* $\Sigma_{\mathsf{Adam}} = \{b\}$. *We have, for the location* l_0, $\Gamma_{\mathsf{Eve}}(l_0) = \{(a, t) \mid t > 0\} \cup \{(\perp, 0)\}$ *and* $\Gamma_{\mathsf{Adam}}(l_0) = \{(b, t) \mid t > 0\} \cup \{(\perp, 0)\}$.

In timed games, we carefully define the memory used by a strategy. It usually uses an infinitely precise value of clocks (see Example 27), so storing a configuration requires infinite memory. We define a memoryless strategy as a strategy only storing the current configuration. A finite-memory strategy stores a finite number of configurations or uses a DFA. Otherwise, it is an infinite-memory strategy. For example, an infinite-memory strategy uses an extra clock with an infinitely precise valuation, as in Example 25.

Strategies for concurrent reachability timed games We focus on concurrent reachability timed games $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}, \mathsf{Reach}(T)\rangle$ with $T \subseteq L$ the target set. In such games, always choose $(\perp, 0)$ is a natural winning strategy for Adam, the play can not reach T, but a play conformed to it has a finite duration. To avoid this undesirable behaviour, the responsible player winning a turn infinitely-often always loses. We define a *converge-time* play as a play with a finite duration. Otherwise, we call it a *diverge-time* play. For each player, we only consider strategy (called a diverge-time strategy) such that for all converge-time time conforming, this player is not responsible for the converge-time. More precisely, a *diverge-time strategy* is a strategy such that all conformed plays are either diverge-time or only gives a finite number of choices. A winning strategy for Eve is thus a time-divergent strategy such that all conformed plays reach T. We analogously define winning strategies for Adam. In a concurrent reachability timed games, the existence problem of a winning strategy is decidable. However, this strategy may require infinite memory to control the element of surprise of the game [17].

Example 25. *Consider the concurrent reachability timed games in Figure 3 where* $T = \{l_1\}$. *Winning strategies for* Eve *need infinite memory. As long as* Eve*'s choice is not winning, she needs to choose a delay closer to* 0. *For this, she precisely maintains an extra global value of clock to choose the right delay when it is in* l_0. *Formally, if the current configuration is* (l_0, ν) *for a valuation* ν, *it uses a new infinitely precise global clock without reset,* y, *then she chooses* $(\frac{1}{2^{\nu(y)}}, a)$ *at each turn. If the current configuration is* (l_1, ν) *for a valuation* ν, *then she chooses* $(1, a)$. *It is a winning strategy because either* Adam *always chooses* $(\perp, 0)$ *and the conformed play is time-convergent, or* Adam *can choose a delay no null at each turn, and there exists a moment when* Eve *chooses a smaller delay, and the play is time-divergent.*

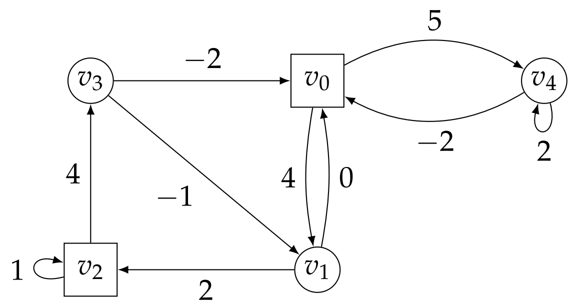


Figure 4: A quantitative game with integer weights.

A randomized strategy that does not requires an additional clock reduces the memory used by uniformly chooses at random a delay in a given interval. A finite-memory divergent-time randomized strategy almost surely achieves Eve's objective in a concurrent reachability timed game.

Theorem 26 ([14]). *Let $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathit{Inv}, \mathit{Reach}(T)\rangle$ be a concurrent reachability timed game with $T \subseteq L$ the target set.* Eve *has a finite-memory randomized strategy χ such that for all locations l where* Eve *can reach T, and all* Adam*'s time-divergent strategies ρ, $\mathbb{P}_l^{\chi,\rho}(\mathit{Reach}(T)) = 1$.*

Example 27. *Consider the concurrent reachability timed game in Figure 3 with $T = \{l_1\}$. By Theorem 26,* Eve *has an almost-winning finite-memory randomized strategy. Let $\tau(l_0, \nu) = (a, Uniform(0, 1 - \nu(x)))$ be an almost-winning strategy that chooses the action a with a delay chosen uniformly at random in the interval $(0, 1 - \nu(x)]$. Let t_j be the delay proposed by* Adam *in the round j under a time-divergent strategy. The probability to never choose* Eve*'s action is* 0 *if $\sum_{j=1}^{\infty} t_j = \infty$ ensuring by the time-divergent strategy for* Adam*. Thus, the probability for choosing the* Eve*'s action is* 1*, and she reaches l_1 with probability* 1*.*

3 Shortest-Path Games

When a player has several winning strategies, we would like to compare them to choose the best one. One way to classify them is to introduce some metrics defined with integer weights. For example, we can model the energy used or created by a robot during a task. The robot then seeks to reach its omega-regular objective by minimizing the energy consumed.

3.1 Quantitative Games

Quantitative games allow us to model and define these different measures. We then consider weighted graphs as arenas. A play $\pi \in V^\omega$ is analogously defined from a classic game. Only its objective adapts to these new measures.

Example 28. *The arena of the quantitative game in Figure 4 depicts weights on transitions, for example, $w(v_3, v_0) = -2$. A run in this game is $v_0 v_1 v_2 v_3 (v_1 v_0)^\omega$.*

An objective in a quantitative game is given by a particular function: the *payoff* mapping a cost at each play on the game. For a given payoff, Eve's objective is to maximize it, and Adam's one is to minimize it. For each vertex v, we define Eve's value denoted $\underline{\mathsf{Val}}_{\mathcal{G}}(v)$) (respectively Adam's value denoted $\overline{\mathsf{Val}}_{\mathcal{G}}(v)$) as the best payoff that Eve (respectively Adam) can guarantee whatever the adversary is doing when the game starts in v. When the game is clear from the context, we can skip $\mathcal{G}$ in the notation.

Definition 29. A *quantitative game* is a tuple $\langle V_{\mathsf{Eve}}, V_{\mathsf{Adam}}, E, w, \mathbf{P}\rangle$ where $V = V_{\mathsf{Eve}} \uplus V_{\mathsf{Adam}}$ is a finite set of vertices partitioned into two sets V_{Eve} and V_{Adam} of Eve and Adam respectively, $E \subseteq V \times V$ is a transition relation, $w : E \to \mathbb{Z}$ is a weight function, and $\mathbf{P}$ is a payoff function.

Under a payoff, we can not use the notion of winning strategy, but we study the optimality of strategies. A strategy is *optimal* for a player if it guarantees its value. In other words, any play from v conforming to an optimal strategy for Eve (respectively for Adam) has a payoff higher (respectively lower) than $\underline{\mathsf{Val}}_{\mathcal{G}}(v)$ (respectively $\overline{\mathsf{Val}}_{\mathcal{G}}(v)$). In this game, as winning strategies do not exist, the determination is defined with the concept of value. A game is *determined* if for all vertices v, $\underline{\mathsf{Val}}(v) = \overline{\mathsf{Val}}(v)$. When a game is determined, we denote $\mathsf{Val}(v)$ the value $\underline{\mathsf{Val}}(v) = \overline{\mathsf{Val}}(v)$ of v, and decide if $\mathsf{Val}(v) > x$ where $x \in \mathbb{Q}$ is a threshold. Generally, solving this problem is done by the existence problem of an optimal strategy for one player. The computation of the optimal strategy, if it exists, allows us to compute a vector containing all the values of the game $(\mathsf{Val}(v))_v$.

Related work on classical payoff There exist many possible payoffs for quantitative games [21]. Each of them expresses a precise property on plays. In particular, quantitative games can express a ω-regular objective whose colours are represented by the weights. For example, the payoff $\mathrm{Sup}(\pi) = \sup w(v_i, v_{i+1})$ expresses the reachability of T when the arena has the weight 1 for every incident edge of T, and 0 otherwise. As Eve would maximize this payoff, we only consider the maximum on a play π, and she wins if her payoff is 1.

Arithmetic operations, as average or sum, also define a payoff. A *mean-payoff* (**MP**) objective computes the cost of a play as an average. It may not be well defined, so

$$\mathbf{MP}(\pi) = \liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} w(v_i, v_{i+1})$$

is the mean-payoff for a play π. A game $\langle V_{\mathsf{Eve}}, V_{\mathsf{Adam}}, E, w, \mathbf{MP}\rangle$ is a mean-payoff game.

Example 30. *Consider the mean-payoff game in Figure 4. For the play $\pi = v_0(v_1v_2v_3)^\omega$ the mean-payoff is given by the average on the cycle $(v_1v_2v_3)$, i.e. $\mathbf{MP}(\pi) = 5/3$.*

Theorem 31 ([21])**.** *Mean-payoff games are determined and both players have memoryless deterministic optimal strategies.*

Example 32. *Let the mean-payoff game in Figure 4. By Theorem 31,* Eve *and* Adam *have memoryless deterministic optimal strategies. The strategy τ^* defined by $\tau^*(v_0) = v_1$, and $\tau^*(v_2) = v_3$ is a memoryless optimal strategy for* Eve*. The strategy σ^* defined by $\sigma^*(v_1) = v_2$, $\sigma^*(v_3) = v_1$, and $\sigma^*(v_4) = v_0$ is a memoryless optimal strategy for* Adam*.*

The mean-payoff is a long-term objective, so it is independent of its prefix of play. In economics, for example, the prefix of play is more important than its long-term. The *discounted-payoff* (**DP**) objective can take more account its prefix of play. For a play π and a parameter $\lambda \in (0,1)$,

$$\mathbf{DP}_\lambda(\pi) = (1-\lambda)\sum_{i=0}^{\infty} \lambda^i w(v_i, v_{i-1})$$

is a discounted-payoff objective. A game $\langle V_{\mathsf{Eve}}, V_{\mathsf{Adam}}, E, w, \mathbf{DP}_\lambda\rangle$ is a discounted game with parameter $\lambda \in (0,1)$.

Theorem 33 ([21]). *For all $\lambda \in (0,1)$, discounted games with parameter λ are determined and both players have memoryless deterministic optimal strategies.*

Example 34. *Let the discounted game in Figure 4 with the parameter $\lambda = 0.9$. By Theorem 33,* Eve *and* Adam *have memoryless deterministic optimal strategies. The strategy τ^* defined by $\tau^*(v_0) = v_1$, and $\tau^*(v_2) = v_3$ is an optimal strategy for* Eve*. The strategy σ^* defined by $\sigma^*(v_1) = v_2$, $\sigma^*(v_3) = v_3$, and $\sigma^*(v_4) = v_0$ is an optimal strategy for* Adam*. We note that these strategies are optimal for the corresponding mean-payoff game (see Example 32). When λ is close to 1, the discounted-payoff objective is close to the Mean-payoff objective [21].*

Now, let consider the parameter $\lambda = 0.5$. In this game, only Adam *changes its optimal choice by $\sigma^*(v_1) = v_0$. However, with the parameter $\lambda = 0.1$, the optimal strategy changes for both players: $\tau^*(v_0) = v_4$, $\tau^*(v_2) = v_3$, $\sigma^*(v_1) = v_0$, $\sigma^*(v_3) = v_0$, and $\sigma^*(v_4) = v_0$.*

Another interesting payoff, the *total-payoff* (**TP**) objective computes the cost of a play with a sum. For a play π,

$$\mathbf{TP}(\pi) = \limsup_{n\to\infty} \sum_{i=0}^{n-1} w(v_i, v_{i+1})$$

defines the total-payoff objective. A game $\langle V_{\mathsf{Eve}}, V_{\mathsf{Adam}}, E, w, \mathbf{TP}\rangle$ is a total-payoff game.

Example 35. *Consider the total-payoff game in Figure 4. For the play $\pi = v_0(v_1v_2v_3)^\omega$, the total-payoff is infinite because π is an infinite path. It can be finite only if the play is finite.*

Theorem 36 ([21]). *Total-payoff games are determined and both players have a memoryless deterministic optimal strategy.*

3.2 Shortest-Path Objectives

The shortest-path objective derives from the shortest-path problem on weighted graphs. It consists of reaching a target while minimizing the weight to go there. This problem is extended to the case of two-player games where the second player avoids it or maximizes the weight. This problem is also extended to the case of MDPs.

3.2.1 Shortest-Path Game

Shortest-path games combine a reachability objective with a total-payoff objective. In this case, Adam wants to reach a target with the smallest total-payoff possible. Eve wants to avoid this case: she would not reach the target, and when it is not possible, she reaches it with the greatest total-payoff. Combining qualitative and quantitative objectives, enabling to select a good strategy among the valid ones for the selected metrics, often leads to the need for memory to play optimally. Formally, we define the *shortest-path* payoff (**SP**) for a play π and a target set T by :

$$\mathbf{SP}(\pi) = \begin{cases} +\infty & \text{if } \pi_k \notin T \text{ for all } k \geqslant 0 \\ \mathbf{TP}(\pi[k]) & \text{if } k \geqslant 0 \text{ is the minimal index such that } \pi_k \in T \end{cases}$$

Definition 37. A *shortest-path game* (SPG) is a tuple $\langle V_{\mathsf{Eve}}, V_{\mathsf{Adam}}, E, w, T, \mathbf{SP}\rangle$ where $V = V_{\mathsf{Eve}} \uplus V_{\mathsf{Adam}} \uplus T$ is a finite set of vertices partitioned into the sets V_{Eve} and V_{Adam} of Eve and Adam respectively, and a set T of target vertices, $E \subseteq V \setminus T \times V$ is a set of directed edges, and $w\colon E \to \mathbb{Z}$ is the weight function, associating an integer weight with each edge.

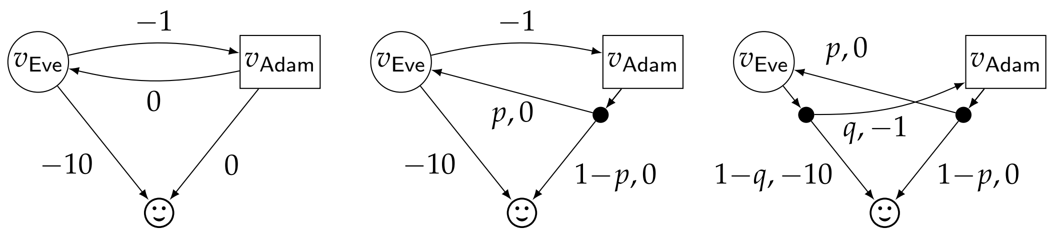


Figure 5: On the left, a shortest-path game, where Adam requires memory to play optimally. In the middle, the Markov Decision Process obtained when letting Adam play at random, with a parametric probability $p \in (0, 1)$. On the right, the Markov Chain obtained when Eve plays along a memoryless randomised strategy, with a parametric probability $q \in [0, 1]$.

Example 38. *The left of Figure 5 represents a shortest-path game where ☺ is the only target.*

In a shortest-path game, without loss of generality, we assume that non-target vertices are deadlock-free, i.e. for all vertices $v \in V \setminus T$, $E(v) \neq \emptyset$. We note that these games are determined [7, Theorem 1].

Related work on this objective A shortest-path game is very close to a total-payoff game. There exists a polynomial-time reduction from a total-payoff game to a shortest-path game. So we can use results on values and strategies computation for shortest-path game to total-payoff game [7]. Moreover, there exists a link between shortest-path gales and mean-payoff games. The computation of the values (with Equation (2)) can be used in a mean-payoff game [21].

In a *divergent shortest-path game*, its arena does not contain null cycles. Their SCCs contain only positive cycles (positive SCC) or only negative cycles (negative SCC). In such a game, the value can be computed in polynomial time. This algorithm starts by identifying and removing the vertices of values $+\infty$ via an attractor to the target for Adam. Then it computes the values by SCC in reverse topological order. In a positive SCC, there exists no vertex of value $-\infty$, the value iteration of (2) converges after $|SCC|$ steps. In a negative SCC, the vertices of $-\infty$ are computed by an attractor to the target for Eve. For the other vertices, the same result is applied as in a positive SCC by multiplying the weights by -1 [11].

3.2.2 Stochastic Shortest-Path Problem

When the adversary's behaviour is modelled by probabilities, the minimization of expectation of a path reaching a target is a variant of the shortest-path problem [9]. This objective has been well studied under the name of *stochastic shortest-path problem* on MDP [3, 18, 1].

The stochastic shortest-path problem in MDP requires adding a cost function in MDP. For a MDP $\langle S, A, P\rangle$ we define a partial function $r : S \times A \times S \to \mathbb{Z}$ mapping a weight for any triplet (s, α, s') such that $P(s, \alpha, s') > 0$. This weight function is naturally transferred to MC induced by a strategy. Given a random variable X on the infinite paths in the MC, we let $\mathbb{E}_v^{\chi}(X)$ the expectation of X for the probability measure $\mathbb{P}_v^{\rho}$. Also, in the case where $X = \mathbf{SP}$, the computation of the expectation in the MC $\langle V, P\rangle$ is given by the Bellman equation. Thus, the vector $(\mathbb{E}_v^{\chi}(\mathbf{SP}))_{v\in V}$ is the only solution to the system of equations

$$\mathbb{E}_v^\chi(\mathbf{SP}) = \begin{cases} 0 & \text{if } v \in T \\ \sum_{v' \in E(v)} P(v,v') \times (w(v,v') + \mathbb{E}_{v'}^\chi(\mathbf{SP})) & \text{if } v \notin T \end{cases} \tag{1}$$

We can extend the computation of the expectation in the MPD to quantify on the strategies to obtain the minimum or maximum expectation of a random variable. Formally, if X is a random variable, we define for a vertex v, the minimal and the maximal expectation by

$$\mathbb{E}_v^{\min}(X) = \min_{\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}} \mathbb{E}_v^\chi(X) \qquad \text{and} \qquad \mathbb{E}_v^{\max}(X) = \max_{\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}} \mathbb{E}_v^\chi(X)$$

Let T be a target set, and v an initial vertex, the stochastic shortest-path problem in an MDP consists of the calculation of $\mathbb{E}_v^{\min}(\diamond T)$, as well as the calculation of optimal strategy ensuring this expectation. When for all $\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}$ $\mathbb{P}_v^\chi(\diamond T) = 1$, [3] assures us that there exists a memoryless deterministic strategy.

Proposition 39 ([3])**.** *Let $\langle S, A, P\rangle$ be an MDP and $T \subseteq S$ such that for all strategies χ and vertices v, $\mathbb{P}_v^\chi(\diamond T) = 1$. Then,*

1. *$\mathbb{E}_v^{\min}(\diamond T)$ satisfies the Bellman equation* (1)*;*
2. *there exists an optimal memoryless strategy.*

In reality, [3] shows that such a strategy exists in a more general framework. Indeed, this result remains true if, for each strategies, either for all v $\mathbb{P}_v^\chi(\diamond T) = 1$ or there exists v such that $\mathbf{SP}$ from v is $+\infty$. In the context of an MDP not satisfying this hypothesis, an optimal finite-memory deterministic strategy exists [1]. It is based on the same principle as optimal deterministic strategies in two-player games (see Section 3.3). Furthermore, they have shown that the computation of $\mathbb{E}$ can be performed in polynomial time.

Theorem 40 ([1])**.** *Given an arbitrary MDP, T a target set, and an initial vertex v, one can compute $\mathbb{E}_v^{\min}(\diamond T)$ in polynomial time.*

3.3 Deterministic Strategies

In this section, we give a formal definition of values obtained by deterministic strategies. We also give some classical results on these values. For this document, let $\mathcal{G} = \langle V_{\mathsf{Eve}}, V_{\mathsf{Adam}}, E, w, T, \mathbf{SP}\rangle$ be a shortest-path game. Recall a deterministic strategy gives a unique choice, its distribution is a Dirac. For all vertices v, we let $\mathsf{Play}(v, \tau, \sigma)$ be the unique play from v conforming to deterministic strategies $\tau : V^* V_{\mathsf{Eve}} \to V$ and $\sigma : V^* V_{\mathsf{Adam}} \to V$ of Eve and Adam, respectively, and we denote by $\mathbf{SP}(\mathsf{Play}(v, \sigma, \tau))$ its payoff. Then, we formally define the value of strategies σ and τ by letting for all v,

$$\mathsf{dVal}^\sigma(v) = \sup_{\tau' \in \mathsf{d}\mathcal{S}_{\mathsf{Eve}}} \mathbf{SP}(\mathsf{Play}(v, \tau', \sigma)) \qquad \text{and} \qquad \mathsf{dVal}^\tau(v) = \inf_{\sigma' \in \mathsf{d}\mathcal{S}_{\mathsf{Adam}}} \mathbf{SP}(\mathsf{Play}(v, \tau, \sigma'))$$

Finally, we formally define the value for Adam and for Eve: for all vertices v,

$$\overline{\mathsf{dVal}}(v) = \inf_{\sigma \in \mathsf{d}\mathcal{S}_{\mathsf{Adam}}} \mathsf{dVal}^\sigma(v) \qquad \text{and} \qquad \underline{\mathsf{dVal}}(v) = \sup_{\tau \in \mathsf{d}\mathcal{S}_{\mathsf{Eve}}} \mathsf{dVal}^\tau(v)$$

Example 41. *We describe the deterministic value of the game in Figure 5. First, let us consider the vertex* v_{Adam} *as initial.* Adam *could directly reach the target, thus leading to a payoff of* 0*. But he can also choose to go to* v_{Eve}*, in which case* Eve *either jumps directly in the target (leading to a beneficial payoff* -10*), or comes back to* v_{Adam}*, but having already capitalized a total payoff* -1*.* Adam *can continue this way* ad libitum *until he is satisfied (at least 10 times) and jumps to the target. This guarantees a value at most* -10 *for* Adam*, when starting in* v_{Adam}*. Reciprocally,* Eve *can guarantee a payoff at least* -10 *by directly jumping into the target when she plays for the first time. Thus, the value is* -10 *when starting from* v_{Adam} *or* v_{Eve}*.*

If we remove the edge from v_{Eve} *to the target (of weight* -10*), we obtain another game in which* $\mathsf{dVal}(v_{\mathsf{Adam}}) = \mathsf{dVal}(v_{\mathsf{Eve}}) = -\infty$ *since* Adam *can decide to turn as long as he wants in the negative cycle, before switching to the target.*

Finally, recall that deterministic strategies $\sigma^\star$ of Adam and $\tau^\star$ of Eve are optimal (respectively, ε-optimal for a positive real number ε) if, for all vertices v: $\mathsf{dVal}^{\sigma^\star}(v) = \mathsf{dVal}(v)$ and $\mathsf{dVal}^{\tau^\star}(v) = \mathsf{dVal}(v)$ (respectively, $\mathsf{dVal}^{\sigma^\star}(v) \leqslant \mathsf{dVal}(v) + \varepsilon$ and $\mathsf{dVal}^{\tau^\star}(v) \geqslant \mathsf{dVal}(v) - \varepsilon$).

Related work on deterministic values Now we present the main results from [7] on which we base our contributions. We will use the iterative calculation resulting from a fixpoint based on an operator to compute the value and the synthesis of an optimal strategy for Adam.

When we assume that weights are non-negative, the principle of Dijkstra algorithm computes an optimal strategy for Adam in polynomial time. Thus, the value problem, i.e. deciding for $v \in V$ if $\mathsf{dVal}(v) < x$, where $x \in \mathbb{Q}$ is a threshold, is solved in polynomial time [21]. However, when we add negative weights, we cannot use the principle of the Bellman–Ford algorithm or those of other graph algorithms. One reason for this is that the optimal strategy for Adam needs memory (see Example 42). However, there exists a pseudo-polynomial time algorithm to compute values and optimal finite (pseudo-polynomial) strategies [7].

Example 42. *In the shortest-path game in Figure 5,* Adam *cannot achieve this deterministic value (computed in Example 41) by playing memoryless, since it either results in a total-payoff* 0 *(directly going to the target) or* Eve *has the opportunity to keep* Adam *in the negative cycle forever, thus never reaching the target. Therefore,* Adam *needs memory to play optimally.*

The value is given by fixpoint computation based on the operator $\mathcal{F}\colon (\mathbb{Z} \cup \{+\infty\})^V \to (\mathbb{Z} \cup \{+\infty\})^V$ defined for all $x = (x_v)_{v\in V} \in (\mathbb{Z} \cup \{+\infty\})^V$ and all vertices $v \in V$ by

$$\mathcal{F}(x)_v = \begin{cases} 0 & \text{if } v \in T \\ \min_{v' \in E(v)}(w(v,v') + x_{v'}) & \text{if } v \in V_{\mathsf{Adam}} \\ \max_{v' \in E(v)}(w(v,v') + x_{v'}) & \text{if } v \in V_{\mathsf{Eve}} \end{cases} \tag{2}$$

Theorem 43 ([7])**.** *Deterministic values of a shortest-path game can be computed in pseudo-polynomial time.*

An optimal strategy for Adam, if it exists, is generally computed by an iterative algorithm of pseudo-polynomial time complexity [7]. We note that an optimal strategy can not exist for any vertex v such that $\mathsf{dVal}(v) = -\infty$. In this case, Adam controls a negative cycle, and it wants to spend as much time as possible in it, which minimizes its payoff. However, to reach its goal, it must exit this negative cycle. It computes *switching strategies* originated from [7], that is a particular kind of deterministic strategies. It is composed of two memoryless deterministic strategies combined using (pseudo-polynomial) memory. Intuitively, the first strategy guarantees the deterministic value for all conforming plays by reaching a negative cycle or giving the shortest-path to the target. The second strategy, based on an attractor, ensures to reach the target. The switching strategy consists of playing along with the first one until eventually switching to the second one when the length of the current finite play is greater than a parameter. It is optimal from vertices of finite value, and they can get a value as low as wanted from vertices of value $-\infty$.

Theorem 44 ([7]). *In a shortest-path game,*

1. Eve *has an optimal deterministic memoryless strategy computable in pseudo-polynomial time.*

2. *For all vertices with a finite deterministic value,* Adam *has an optimal deterministic pseudo-polynomial memory strategy computable in pseudo-polynomial time. For all vertices with deterministic value $-\infty$, there exists a sequence of optimal deterministic finite-memory strategies computable in pseudo-polynomial time such that its value converges to $-\infty$.*

Deterministic values of a shortest-path game can be computed in pseudo-polynomial time.

Example 45. *For all $n \in \mathbb{N}$, let $\sigma^n = (\sigma_1, \sigma_2, \alpha)$ be a switching strategy. In Figure 5, we have $\sigma_1(v_{\mathsf{Adam}}) = v_{\mathsf{Eve}}$, $\sigma_2(v_{\mathsf{Adam}}) = ☺$, and $\alpha = 3(40+n)+1$. Moreover, σ^n is optimal for $n = 10$.*

If we consider the game in Figure 5 without the edge from v_{Eve} to the target (of weight -10), the previous strategy $\sigma^n = (\sigma_1, \sigma_2, \alpha)$ defines a sequence of strategy that his value converge to $-\infty$.

Now, we consider the game in Figure 6. We have $\sigma_1(v_1) = v_0$ and $\sigma_1(v_3) = v_1$, $\sigma_1(v_1) = v_2$ and $\sigma_1(v_3) = ☺$, and $\alpha = 5(60+n)+1$, for all $n \in \mathbb{N}$.

3.4 Memoryless Strategies

The above definitions can be adapted to memoryless (randomized) strategies. To keep the explanations simple, we define only the upper value above, without counting on hypothetical determination results in this context. Once we have fixed a strategy without memory (randomized) $\rho \in \mathsf{m}\mathcal{S}_{\mathsf{Adam}}$, we get an MDP where Eve always has to choose how to react. We denote by $\mathcal{G}^\rho = \langle V, A, P \rangle$ the MDP induced by ρ where V contains the same set of vertices as $\mathcal{G}$, $A = V \cup \{\bot\}$ contains the adjacent vertices in the game as well as an additional action $\bot$ denoting the random choice of ρ, and the distribution P defined by:

- if $v \in V_{\mathsf{Eve}}$, $P(v, v')$ is only defined if $(v, v') \in E$ in which case $P(v, v') = \mathsf{Dirac}_{v'}$, and $P(v, \bot)$ is also undefined;

- if $v \in V_{\mathsf{Adam}}$, $P(v, \bot) = \rho(v)$, and $P(v, v')$ is undefined for all $v' \in V$.

The cost function used for MDPs is trivially transferred from the game.

Example 46. *In Figure 5, a shortest-path game is presented with its MDP in the middle induced by a memoryless strategy for* Adam*. He chooses to go to v_{Eve} with probability $p \in (0,1)$ and to the target vertex with probability $1-p$.*

Another more complex example is given in Figure 6 where the memoryless strategy for Adam *consists, in vertex v_1, to choose successor v_0 with probability $p \in (0,1)$ and successor v_2 with probability $1-p$, and in vertex v_3, to choose successor v_1 with the same probability p and the target vertex with probability $1-p$.*

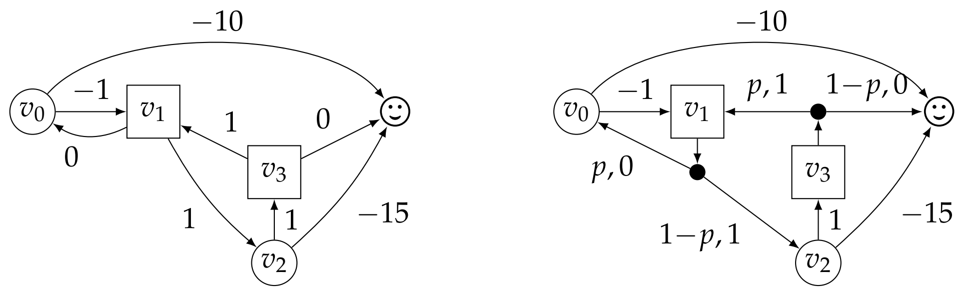


Figure 6: On the left, a more complex example of shortest-path game. On the right, the MDP associated with a randomised strategy of Adam with a parametric probability $p \in (0, 1)$.

In such an MDP, when Eve has chosen her strategy, we obtain an MC. For all MDP $\mathcal{G}^\rho$, and all memoryless strategies $\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}$, we define $\mathcal{G}^{\rho,\chi}$ the MC induced by the strategy χ and the action $\perp$. Formally, $\mathcal{G}^{\rho,\chi} = \langle V, P \rangle$ is defined by V contains the set of vertices of $\mathcal{G}$, and P associates to a vertex $v \in V_{\mathsf{Adam}}$, $P(v) = \rho(v)$ and a vertex $v \in V_{\mathsf{Eve}}$, $P(v) = \chi(v)$.

Example 47. *On the right of Figure 5 is depicted the MC obtained when* Eve *decides to go to* v_{Adam} *with probability* $q \in [0, 1]$ *and to the target vertex with probability* $1 - q$*.*

In $\mathcal{G}^{\rho,\chi}$, we denote by $\mathbb{P}_v^{\rho,\chi}(\diamond T)$ the probability of the set of plays reaching the target set $T \subseteq V$. Likewise, we denote by $\mathbb{E}_v^{\rho,\chi}(\mathbf{SP})$ the expected weight of a path in this MC according to the weights of $\mathcal{G}$. Eve's objective then becomes to maximize the gain in the MDP $\mathcal{G}^\rho$. Consequently, we define the value of the strategy ρ as the worst scenario for Adam:

$$\mathsf{mVal}^\rho(v) = \sup_{\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}} \mathbb{E}_v^{\rho,\chi}(\mathbf{SP})$$

This definition has meaning (otherwise it is worth $+\infty$) only if $\mathbb{P}_v^{\rho,\chi}(\diamond T) = 1$ for all χ, that is to say when the strategy ρ ensures the accessibility of a target vertex with probability 1, whatever the strategy of the opponent. In this case, by denoting P the distribution of MC $\mathcal{G}^{\rho,\chi}$, the vector $(\mathbb{E}_v^{\rho,\chi}(\mathbf{SP}))_{v \in V}$ is the only solution to the system of equations (1).

Since Adam wants to minimise the shortest-path payoff, we finally define the memoryless upper value as

$$\overline{\mathsf{mVal}}(v) = \inf_{\rho \in \mathsf{m}\mathcal{S}_{\mathsf{Adam}}} \mathsf{mVal}^\rho(v)$$

Once again, we say that a memoryless strategy ρ is optimal (respectively, ε-optimal for a positive real number ε) if $\mathsf{mVal}^\rho(v) = \overline{\mathsf{mVal}}(v)$ (respectively, $\mathsf{mVal}^\rho(v) \leqslant \overline{\mathsf{mVal}}(v) + \varepsilon$). To Eve, we only consider optimality and ε-optimality in the MDP $\mathcal{G}^\rho$.

Example 48. *For the game in Figure 5, we let σ and τ the memoryless strategies inducing the MC on the right. Letting $x = \mathbb{E}_{v_{\mathsf{Adam}}}^{\rho,\chi}(\mathbf{TP})$ and $y = \mathbb{E}_{v_{\mathsf{Eve}}}^{\rho,\chi}(\mathbf{TP})$, the system (1) rewrites as*

$$x = (1 - p) \times 0 + p \times y \qquad \textit{and} \qquad y = q \times (-1 + x) + (1 - q) \times (-10)$$

We thus have $x = p(9q - 10)/(1 - pq)$. Two cases happen, depending on the value of p: if $p < 9/10$, then Eve *maximises x by choosing $q = 1$, while she chooses $q = 0$ when $p \geqslant 9/10$. In all cases,* Eve *will therefore play deterministically: if $p < 9/10$, the expected payoff from v_{Adam} will then be $\mathsf{mVal}^\rho(v_{\mathsf{Adam}}) =$*

$-p/(1-p) > -9$; if $p \geqslant 9/10$, it will be $\mathsf{mVal}^\rho(v_{\mathsf{Adam}}) = -10p$. This value is always greater than the optimum -10 that Adam *were able to achieve with memory, since we must keep $1-p>0$ to ensure reaching the target with probability 1. There are no optimal strategies for* Adam*, but an ε-optimal one consisting in choosing probability $p \geqslant 1-\varepsilon/10$. We thus obtain $\overline{\mathsf{mVal}}(v_{\mathsf{Adam}}) = \overline{\mathsf{mVal}}(v_{\mathsf{Eve}}) = -10$ as before.*

The fact that Eve can play optimally with a deterministic strategy in the MDP $\mathcal{G}^\rho$ is not specific to this example. Indeed, in an MDP $\mathcal{G}^\rho$ such that $\mathbb{P}_v^{\rho,\chi}(\diamond T) = 1$ for all χ, Eve cannot avoid reaching the target: she must then ensure the most expensive play possible. Considering the MDP $\tilde{\mathcal{G}}^\rho$ obtained by multiplying all the weights in the graph by -1, the objective of Eve becomes a shortest-path objective. We can then deduce from Proposition 39 that she has an optimal deterministic memoryless strategy. The same applies in the original MDP $\mathcal{G}^\rho$.

Proposition 49. *In the MDP $\mathcal{G}^\rho$ such that $\mathbb{P}_v^{\rho,\chi}(\diamond T) = 1$ for all χ,* Eve *has an optimal deterministic memoryless strategy.*

4 Shortest-Path Games : Memory or Randomisation

The contribution of this internship consists in showing that values are the same when restricting both players to memoryless or deterministic strategies:

Theorem 50. *For all shortest-path games $\mathcal{G}$, for all vertices v, we have $\mathsf{dVal}(v) = \overline{\mathsf{mVal}}(v)$.*

We show this theorem, in the two next sections, by simulating deterministic strategies with memoryless ones and vice versa. We start here by ruling out the case of values $+\infty$. Indeed, $\mathsf{dVal}(v) = +\infty$ characterizes that Adam is not able to reach a target vertex from v with deterministic strategies. This also implies that Adam has no memoryless randomised strategies to ensure reaching the target with probability 1, and thus $\overline{\mathsf{mVal}}(v) = +\infty$. Reciprocally, if $\overline{\mathsf{mVal}}(v) = +\infty$, then Adam has no memoryless strategies to reach the target with probability 1 (since this is the only reason for having a value $+\infty$). Since reachability is a purely qualitative objective, and the arena contains no probabilities, Adam cannot use memory to guarantee to reach the target: therefore, this also means that $\mathsf{dVal}(v) = +\infty$. In the end, we have shown that $\mathsf{dVal}(v) = +\infty$ if and only if $\overline{\mathsf{mVal}}(v) = +\infty$. We thus remove every such vertex, which does not change the values of other vertices in the game.

Example 51. *Consider the shortest-path game in Figure 6, where we remove the edge between v_3 and the target (of weight 0). In this game, all vertices have a deterministic value $+\infty$. In fact, all plays conformed to the strategy for* Eve*, τ, defined by $\tau(v_0) = v_1$ and $\tau(v_2) = v_1$, cannot reach the target. Against this strategy, whatever* Adam *chooses (at random or not), he cannot reach the target. So, the memoryless value is $+\infty$.*

Assumption. From now on, all shortest-path games $\mathcal{G}$ are such that $\mathsf{dVal}(v)$ and $\overline{\mathsf{mVal}}(v)$ are different from $+\infty$, for all vertices v.

4.1 Simulating Deterministic Strategies with Memoryless Strategies

Towards the proof of Theorem 50, we show in this section that, for all shortest-path games $\mathcal{G} = \langle V_{\mathsf{Eve}}, V_{\mathsf{Adam}}, E, w, T, \mathbf{SP}\rangle$ (where no value is $+\infty$) and the vertices $v \in V_{\mathsf{Eve}}$, $\overline{\mathsf{mVal}}(v) \leqslant \mathsf{dVal}(v)$. So, we consider the switching strategies $\sigma = \langle\sigma_1, \sigma_2, \alpha\rangle$. In particular, we use an interesting property of the strategy σ_1. It is such that every finite cyclic play $v_0 v_1 \dots v_k v_0$ conforming to σ_1 has a negative total weight: this is called an *NC-strategy* (for *negative-cycle-strategy*) in [7]. The *fake-value* of σ_1 from a vertex v_0 is defined by

$$\mathsf{fake}^{\sigma_1}(v_0) = \sup\{\mathbf{TP}(v_0 v_1 \cdots v_k) \mid v_k \in T, v_0 v_1 \cdots v_k \text{ conforming to } \sigma_1\}$$

letting $\sup \emptyset = -\infty$: it consists in considering only the games conforming σ_1 which reach the target. The strategy σ_1 is said to be *fake-optimal* if $\mathsf{fake}^{\sigma_1}(v) \leqslant \mathsf{dVal}(v)$ for all vertices v. We then use a more precise result on the NC strategies.

Proposition 52 ([7])**.** *Fix a memoryless strategy σ_2 computed from an attractor computation. Then, there exists a fake-optimal NC-strategy σ_1. Moreover, for all such fake-optimal NC-strategies, and all $n \in \mathbb{N}$, the switching parameter $\alpha = (2W(|V|-1)+n)|V|+1$ defines a switching strategy $\sigma = \langle\sigma_1, \sigma_2, \alpha\rangle$ with a value $\mathsf{dVal}^{\sigma}(v) \leqslant \max(-n, \mathsf{dVal}(v))$, from all initial vertices $v \in V$.*

Definition of a memoryless (randomised) strategy. Let $n \in \mathbb{N}$, we consider the switching strategy $\sigma = \langle\sigma_1, \sigma_2, \alpha\rangle$ described before, of value $\mathsf{dVal}^{\sigma}(v) \leqslant \max(-n, \mathsf{dVal}(v))$, and simulate it with a memoryless (randomised) strategy for Adam, denoted ρ_p, with a parametrised probability $p \in (0,1)$. This new strategy is a probabilistic superposition of the two memoryless deterministic strategies σ_1 and σ_2.

Formally, we define ρ_p on each strongly connected components (SCC) of the arena according to the presence of a negative cycle. In an SCC that does not contain negative cycles, for each vertex $v \in V_{\mathsf{Adam}}$ of the SCC, we let $\rho_p(v) = \mathsf{Dirac}_{\sigma_1(v)}$: Adam chooses to play the first strategy σ_1 of the switching strategy, thus looking for a negative cycle in the next SCCs (in topological order) if any. In an SCC that contains a negative cycle, for each vertex $v \in V_{\mathsf{Adam}}$ of the SCC, we let $\rho_p(v)$ be the distribution of support $\{\sigma_1(v), \sigma_2(v)\}$ that chooses $\sigma_1(v)$ with probability p and $\sigma_2(v)$ with probability $1-p$, except if $\sigma_1(v) = \sigma_2(v)$ in which case we choose it with probability 1. Note that MDPs in Figures 5 and 6 are obtained by applying this strategy ρ_p.

We fix some vertex $v_0 \in V$. In the rest of this section, we prove the following result:

Proposition 53. *For ε small enough and p close enough to 1, $\mathsf{mVal}^{\rho_p,\tau}(v_0) \leqslant \mathsf{dVal}^{\sigma}(v_0) + \varepsilon$.*

This entails the expected result. Indeed, if $\mathsf{dVal}(v_0) \in \mathbb{Z}$, we get (with $n = |\mathsf{dVal}(v_0)|$) that $\mathsf{mVal}^{\rho_p}(v_0) \leqslant \mathsf{dVal}(v_0) + \varepsilon$, and thus $\overline{\mathsf{mVal}}(v_0) \leqslant \mathsf{dVal}(v_0)$ since this holds for all $\varepsilon > 0$. Otherwise, $\mathsf{dVal}(v_0) = -\infty$, and letting n tend towards $+\infty$, we also get $\overline{\mathsf{mVal}}(v_0) = -\infty$.

We first prove that ρ_p is one of the strategies of Adam that guarantee to reach the target with probability 1 in the MDP $\mathcal{G}^{\rho_p}$ no matter how Eve reacts.

Proposition 54. *For all strategies $\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}$, $\mathbb{P}_{v_0}^{\rho_p,\chi}(\diamond T) = 1$.*

Proof. Recall that we designed our arena so that target vertices are the only deadlocks. Thus, by using the characterisation of Lemma 20, $\min_{\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}} \mathbb{P}_{v_0}^{\rho_p,\chi}(\diamond T) = 1$ if and only if for all $\chi \in \mathsf{m}\mathcal{S}_{\mathsf{Eve}}$, all bottom SCCs of the MC $\mathcal{G}^{\rho_p,\chi}$ (the ones from which we cannot exit) consist in a unique target vertex. Suppose in the contrary that Eve has a memoryless strategy χ such that the MC $\mathcal{G}^{\rho_p,\chi}$ contains a bottom SCC $\mathcal{C}$ with no target vertices.

If all vertices of $\mathcal{C}$ belong to Eve, then they all have a successor in $\mathcal{C}$ and therefore there also exists a deterministic memoryless strategy τ' for which all vertices $v \in \mathcal{C}$ are such that $\mathsf{dVal}^{\tau'}(v) =$

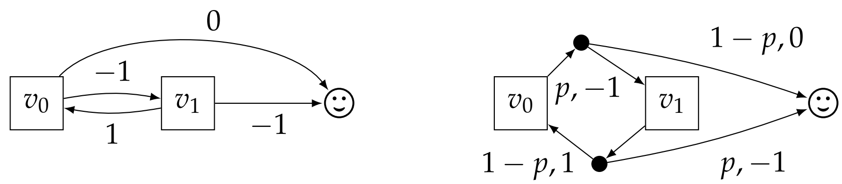


Figure 7: On the left, a game graph with no negative cycles where ρ_p is optimal. The MC obtained when playing a different randomised memoryless strategy.

$+\infty$, and thus $\mathsf{dVal}(v) = +\infty$: this contradicts our hypothesis that all vertices have a deterministic value different from $+\infty$.

Otherwise, for all vertices $v \in V_{\mathsf{Adam}} \cap \mathcal{C}$, since $\mathcal{C}$ is a bottom SCC of $\mathcal{G}^{\rho_p,\mathcal{X}}$, the distribution $\rho_p(v)$ has its support included in $\mathcal{C}$. If $\mathcal{C}$ is included in an SCC of $\mathcal{G}$ with no negative cycles, $\mathrm{supp}(\rho_p(v)) = \{\sigma_1(v)\}$. Playing with $\sigma_1(v)$ in $\mathcal{C}$ will end up in a cycle (since there are no deadlocks) that must be negative, by the hypothesis on σ_1, which is impossible. Thus, $\mathcal{C}$ must be included in an SCC of $\mathcal{G}$ with a negative cycle. Then, $\mathrm{supp}(\rho_p(v)) = \{\sigma_1(v), \sigma_2(v)\} \subseteq \mathcal{C}$, and in particular the attractor strategy is not able to reach a target vertex. Thus, playing the deterministic switching strategy σ will result in not reaching a target vertex either, so that $\mathsf{dVal}(v) = +\infty$ for $v \in V_{\mathsf{Adam}} \cap \mathcal{C}$ which also contradicts our hypothesis. □

We can, therefore, apply Proposition 49. This result is very helpful since it allows us to only consider deterministic memoryless strategies τ to compute, for all initial vertices v_0, $\mathsf{mVal}^{\rho_p}(v_0) = \sup_\tau \mathsf{mVal}^{\rho_p,\tau}(v_0)$. We thus consider such a strategy τ and we now show that $\mathsf{mVal}^{\rho_p,\tau}(v_0) \leqslant \mathsf{dVal}^{\sigma}(v) + \varepsilon$ whenever $p < 1$ is close enough to 1 (in function of $\varepsilon > 0$). By gathering the finite number of lower bounds about p, for all deterministic memoryless strategies of Eve (there are a finite number of such), we obtain a lower bound for p such that $\mathsf{mVal}^{\rho_p}(v_0) \leqslant \mathsf{dVal}^{\sigma}(v_0) + \varepsilon$, as expected to prove Proposition 53.

The case where the whole arena does not contain any negative cycles is easy. In this case, ρ_p chooses the strategy σ_1 with probability 1, by definition since no SCCs contain a negative cycle (this is the only reason why we defined ρ_p as it is, for such SCCs): a play from initial vertex v_0 conforming to ρ_p is thus conforming to σ_1. Since the graph contains no negative cycles and all cycles conforming to σ_1 must be negative, all plays from v_0 conforming to σ_1 reach the target set of vertices, with a total payoff at most $\mathsf{dVal}^{\sigma}(v_0)$. This single play has probability 1 in the MC $\mathcal{G}^{\rho_p,\tau}$, thus $\mathbb{E}_{v_0}^{\rho_p,\tau}(\mathbf{SP}) \leqslant \mathsf{dVal}^{\sigma}(v_0)$, which proves that $\mathsf{mVal}^{\rho_p}(v) \leqslant \mathsf{dVal}^{\sigma}(v_0)$ as expected.

Example 55. *If the definition of ρ_p would not distinguish the SCCs with no negative cycles from the other SCCs, we would not have the optimality of ρ_p as shown before. Indeed, consider the arena on the left of Figure 7, which has no negative cycles. We have $\mathsf{dVal}(v_0) = -2$ and $\mathsf{dVal}(v_1) = -1$. As a switching strategy, we can choose $\sigma_1(v_0) = v_1$, $\sigma_1(v_1) = ☺$, $\sigma_2(v_0) = ☺$ and $\sigma_2(v_1) = v_0$. Then, ρ_p is equal to σ_1 (and thus independent of p), and $\mathsf{mVal}^{\rho_p}(v_0) = -2$ and $\mathsf{mVal}^{\rho_p}(v_1) = -1$. However, if we would have chosen to still mix σ_1 and σ_2, we would obtain a strategy ρ'_p, and the MC on the right of Figure 7. Then, we get $\mathsf{mVal}^{\rho'_p}(v_0) = -2p^2/(1-p(1-p))$ and $\mathsf{mVal}^{\rho'_p}(v_1) = (p^2 - 3p + 1)/(1-p(1-p))$ whose limits are -2 and -1 respectively, when p tends to 1. This strategy ρ'_p would then still be ε-optimal for p close enough to 1.*

Now, suppose that the arena contains negative cycles. We let $c > 0$ be the maximal size of an elementary cycle (that visits a vertex at most once) in $\mathcal{G}$, $w^- > 0$ be the opposite of the maximal weight of an elementary negative cycle in $\mathcal{G}$, and $w^+ \geqslant 0$ be the maximal weight of an elementary non-negative cycle in $\mathcal{G}$ (or 0 if such cycle does not exist).

Example 56. *In the game in Figure 5, we have $c = 2$, $w^- = 1$, and $w^+ = 0$ (since there is no non-negative cycles). In the game in Figure 6, we have $c = 3$, $w^- = 1$, and $w^+ = 3$.*

The difficulty initiates from the possible presence of non-negative cycles too. Indeed, when applying the switching strategy σ, all cycles conforming to σ_1 have a negative weight. This is no longer true with the probabilistic superposition ρ_p, as can be seen in the example of Figure 6. Finding an adequate lower-bound for p requires to estimate $\mathbb{E}_{v_0}^{\rho_p,\tau}(\mathbf{SP})$, by controlling the weight and probability of non-negative cycles, balancing them with the ones of negative cycles. The crucial argument comes from the definition of the superposition ρ_p:

Lemma 57. *All cycles in $\mathcal{G}^{\rho_p,\tau}$ of non-negative total weight contain at least one edge of probability $1-p$.*

Proof. Suppose on the contrary that all edges have probability p or 1, then the cycle is conforming to strategy σ_1, and has, therefore, a negative weight. □

We now partition the set Π of plays starting in v_0, conforming to ρ_p and τ, and reaching the target set of vertices, into subsets $\Pi_{i,\ell}$ regarding the number i of edges of probability $1-p$ they go through, and their length ℓ (we always have $i \leqslant \ell$). The partition is depicted in Figure 8:

- $\Pi_{0,\mathbb{N}}$, depicted in yellow, contains all plays with no edges of probability $1-p$;
- $\Pi_{\geqslant I,\mathbb{N}}$, depicted in green, contains all plays having at least

  $$I = \left\lceil \frac{2w^+}{\gamma W} + \frac{8(w^+ + |V|W)}{\varepsilon} \right\rceil$$

  edges of probability $1-p$ where $\gamma = c\left(1 + \frac{w^+}{w^-}\right) \geqslant 1$;
- $\Pi_{<I,\geqslant L}$, depicted in blue, contains all plays with at most I edges of probability $1-p$, and of length at least $L = I\gamma + \frac{2|\mathsf{dVal}^\sigma(v_0)|+|V|W}{w^-}c + |V|$;
- $\widetilde{\Pi}$, depicted in red, is the rest of the plays.

We let $\gamma_{0,\mathbb{N}}$ (respectively, $\gamma_{<I,\geqslant L}$, $\gamma_{\geqslant I,\mathbb{N}}$, and $\widetilde{\gamma}$) be the previous expectation restricted to plays in $\Pi_{0,\mathbb{N}}$ (respectively, $\Pi_{<I,\geqslant L}$, $\Pi_{\geqslant I,\mathbb{N}}$, $\widetilde{\Pi}$). By linearity of expectation,

$$\mathsf{mVal}^{\rho_p,\tau}(v_0) = \mathbb{E}^{\rho_p,\tau}(\mathbf{SP}) = \gamma_{0,\mathbb{N}} + \gamma_{<I,\geqslant L} + \gamma_{\geqslant I,\mathbb{N}} + \widetilde{\gamma}$$

Proving that $\mathsf{mVal}^{\rho_p,\tau}(v_0) \leqslant \mathsf{dVal}^\sigma(v_0) + \varepsilon$ will be done by showing that, under assumptions on p, yellow and blue zones are such that $\gamma_{0,\mathbb{N}} + \gamma_{<I,\geqslant L} \leqslant \mathsf{dVal}^\sigma(v_0) + \varepsilon/2$, while red and green zones are such that $\gamma_{\geqslant I,\mathbb{N}} + \widetilde{\gamma} \leqslant \varepsilon/2$. Indeed, plays with a large number of non-negative cycles contain a large number of edges of probability $1-p$, by Lemma 57. But if p is made close enough to 1, the probability of this set of plays will be small enough.

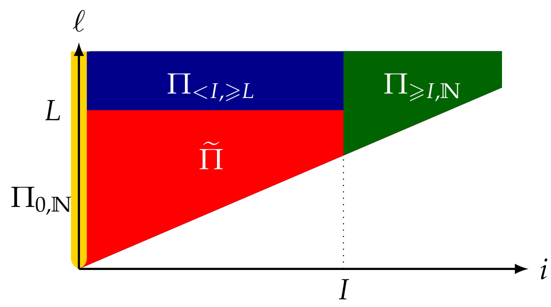


Figure 8: Partition of plays Π.

Example 58. *All zones not necessarily contain any plays, in this case, its expectation is null. For example, the green zone $\Pi_{\geqslant I,\mathbb{N}}$ contains plays only if the arena contains non-negative cycles. Also,* Eve*'s choices, through the chosen strategy, influence the presence of play in zones. For example, in the game in Figure 5, if* Eve *chooses v_{Eve}, then the yellow zone $\Pi_{0,\mathbb{N}}$ is empty, but if* Eve *chooses ☺, then the blue zone $\Pi_{<I,\geqslant L}$ is empty.*

Proof of Proposition 53. All plays of $\Pi_{0,\mathbb{N}}$ reach the target without edges of probability $1-p$, i.e. by conforming to σ_1. By fake-optimality of σ_1, their total payoff is upper-bounded by $\mathsf{dVal}^\sigma(v_0)$. Notice that in case of $\mathsf{dVal}(v_0) = -\infty$, no plays conforming to σ_1 starting in v_0 reach the target, since Adam has the opportunity to stay as long as he wants in negative cycles: thus $\Pi_{0,\mathbb{N}} = \emptyset$ in this case, and $\gamma_{0,\mathbb{N}} = 0$.

All plays of $\Pi_{i,\ell}$, with $1 \leqslant i < I$ and $\ell \geqslant L$, go through i edges of probability $1-p$. By Lemma 57, they contain at most i elementary cycles of non-negative total weight (each of weight is at most w^+). The total length of these cycles is at most ic. Once we have removed these cycles from the play, it remains a play of length at least $\ell - ic$. By a repeated pumping argument, it still contains at least $\left\lfloor \frac{\ell - ic - |V|}{c} \right\rfloor$ elementary cycles, that have all a negative total weight (each has a weight at most $-w^-$). The remaining part, once removed the last negative cycles it contains, has length at most $|V|$, and thus a total payoff at most $|V|W$. In summary, the total payoff of a play in $\Pi_{i,\ell}$ is at most

$$\begin{aligned} iw^+ + \left\lfloor \frac{\ell - ic - |V|}{c} \right\rfloor (-w^-) + |V|W &\leqslant Iw^+ + \frac{L - Ic - |V|}{c}(-w^-) + |V|W \\ &= -2|\mathsf{dVal}^\sigma(v_0)| \leqslant 0 \end{aligned} \tag{3}$$

Let us then consider three cases.

- If $\mathsf{dVal}^\sigma(v_0) \geqslant 0$, we note that all plays in $\Pi_{<I,\geqslant L}$ have a non-positive total payoff, therefore at most $\mathsf{dVal}^\sigma(v_0)$. Thus,

$$\begin{aligned} \gamma_{0,\mathbb{N}} + \gamma_{<I,\geqslant L} &\leqslant \mathsf{dVal}^\sigma(v_0)\mathbb{P}(\Pi_{0,\mathbb{N}}) + \mathsf{dVal}^\sigma(v_0)\mathbb{P}(\Pi_{<I,\geqslant L}) \\ &= \mathsf{dVal}^\sigma(v_0)\big(\mathbb{P}(\Pi_{0,\mathbb{N}}) + \mathbb{P}(\Pi_{<I,\geqslant L})\big) \leqslant \mathsf{dVal}^\sigma(v_0) \end{aligned}$$

- If $\mathsf{dVal}^\sigma(v_0) < 0$ and $\Pi_{<I,\geqslant L} \neq \emptyset$, we have $\gamma_{0,\mathbb{N}} \leqslant 0$ (whatever $\mathsf{dVal}(v_0) = -\infty$ or not). Moreover, a play in $\Pi_{i,\ell}$ goes through i edges of probability $1-p$ and at most ℓ edges of

probability p, other edges having probability 1. So, it has probability at least $(1-p)^i p^\ell$. We can deduce that

$$\gamma_{<I,\geqslant L} \leqslant \sum_{i=1}^{I-1}\sum_{\ell=L}^{\infty}(1-p)^i p^\ell \underbrace{\left(iw^+ + \left\lfloor \frac{\ell - ic - |V|}{c} \right\rfloor (-w^-) + |V|W\right)}_{\leqslant 0 \text{ by } (3)} \leqslant \mathsf{dVal}^\sigma(v_0)$$

the last inequality being true when p is close enough to 1, as shown in Appendix B.

- If $\mathsf{dVal}^\sigma(v_0) < 0$ and $\Pi_{<I,\geqslant L} = \emptyset$, then $\mathsf{dVal}(v_0) \neq -\infty$, since otherwise a play is conformed to σ_1 for L rounds, and then switching to σ_2 for at most $|V| \leqslant I$ rounds, would be in $\Pi_{<I,\geqslant L}$. Thus, $\gamma_{0,\mathbb{N}} + \gamma_{<I,\geqslant L} = \gamma_{0,\mathbb{N}} \leqslant \mathsf{dVal}^\sigma(v_0)\mathbb{P}(\Pi_{0,\mathbb{N}})$. Moreover, by the same argument, all plays in $\Pi_{0,\mathbb{N}}$ are acyclic and their length is at most $|V|$: they go through no edges of probability $1-p$, and thus at most $|V|$ edges of probability p. Therefore, $\mathbb{P}(\Pi_{0,\mathbb{N}}) \geqslant p^{|V|}$, and thus, once again because $\mathsf{dVal}^\sigma(v_0) < 0$, when $p \geqslant (1-\varepsilon/2|\mathsf{dVal}^\sigma(v_0)|)^{1/|V|}$ which is less than 1 for ε small enough,

$$\gamma_{0,\mathbb{N}} + \gamma_{<I,\geqslant L} \leqslant \mathsf{dVal}^\sigma(v_0)p^{|V|} \leqslant \mathsf{dVal}^\sigma(v_0) + \frac{\varepsilon}{2}$$

In all cases, we have $\gamma_{0,\mathbb{N}} + \gamma_{<I,\geqslant L} \leqslant \mathsf{dVal}^\sigma(v_0) + \varepsilon/2$.

To conclude, we now show that $\gamma_{\geqslant I,\mathbb{N}} + \widetilde{\gamma} \leqslant \varepsilon/2$. First, a play of $\Pi_{\geqslant I,\mathbb{N}}$ has a large total payoff, but a low probability to happen, which enables us to control its expected payoff. Indeed, consider a play of $\Pi_{i,\mathbb{N}}$, with $i \geqslant I$. By Lemma 57, it contains at most i elementary cycles of non-negative total weight. The remaining of the play may contain negative cycles, as well as an acyclic part reaching the target in at most $|V|$ steps. The total payoff of the whole play is thus at most $iw^+ + |V|W$. Moreover, $\mathbb{P}(\Pi_{i,\mathbb{N}}) \leqslant (1-p)^i$ since all the plays contain i edges of probability $1-p$. In the overall,

$$\gamma_{\geqslant I,\mathbb{N}} \leqslant \sum_{i=I}^{\infty}(iw^+ + |V|W)(1-p)^i = (1-p)^I\left(\frac{w^+}{p}I + \frac{w^+(1-p)}{p^2} + \frac{|V|W}{p}\right) \leqslant \frac{\varepsilon}{4}$$

where the last inequality holds for p close enough to 1, as shown in Appendix B.

Finally, all plays of $\widetilde{\Pi}$ have a length less than L (and thus a total payoff at most LW) and a number i of edges of probability $1-p$ such that $0 < i < I$. By a similar argument as before, if $p \geqslant LW/(LW+\varepsilon/4)$, we have

$$\widetilde{\gamma} \leqslant \sum_{i=1}^{I} LW(1-p)^i = LW\frac{(1-p)(1-(1-p)^I)}{p} \leqslant LW\frac{1-p}{p} \leqslant \frac{\varepsilon}{4}$$

since $p \mapsto (1-p)/p$ is decreasing on $(0,1)$. □

This ends the proof that for all vertices v, $\overline{\mathsf{mVal}}(v) \leqslant \mathsf{dVal}(v)$. Let us illustrate the computation of the lower-bound on probability p of the memoryless strategy ρ_p in the previously studied examples.

Example 59. *For the game in Figure 5, with initial vertex v_{Adam}, we have $\gamma = 2$. For $\varepsilon = 0.1$, we then have $I = 2400$, and $L = 4903$. The lower-bound on p is then 0.9999995, which gives a value $\mathsf{mVal}^{\rho_p}(v_{\mathsf{Adam}}) = -10p = -9.999995$.*

For the game in Figure 6, with initial vertex v_2, we have $\gamma = 12$. For $\varepsilon = 0.1$, we then have $I = 3121$, and $L = 37730$. The lower-bound on p is then 0.99999998, which gives a value $\mathsf{mVal}^{\rho_p}(v_2) \approx -7.9999996$. We see that the lower-bound are correct, even if they could certainly be made coarser.

4.2 Simulating Memoryless Strategies with Deterministic Strategies

To finish the proof of Theorem 50, we will show that $\mathsf{dVal}(v) \leqslant \overline{\mathsf{mVal}}(v)$, for all vertices v. For a given memoryless strategy ρ ensuring Adam to reach the target set T with probability 1, we build a deterministic strategy σ which guarantees a value $\mathsf{dVal}^\sigma(v) \leqslant \mathsf{mVal}^\rho(v)$ from vertex v. Then, as in the previous section, if $\overline{\mathsf{mVal}}(v)$ is finite, for an ε-optimal memoryless strategy ρ, we get a deterministic strategy such that $\mathsf{dVal}^\sigma(v) \leqslant \overline{\mathsf{mVal}}(v) + \varepsilon$, and thus $\mathsf{dVal}(v) \leqslant \overline{\mathsf{mVal}}(v) + \varepsilon$. We can conclude since this holds for all $\varepsilon > 0$. In case $\overline{\mathsf{mVal}}(v) = -\infty$, if ρ guarantees a value at most $-n$ with $n \in \mathbb{N}$, then so does the deterministic strategy σ, which also ensures that $\mathsf{dVal}(v) = -\infty$.

Definition of the deterministic strategy σ We fix a memoryless strategy ρ, and an initial vertex v_0. Strategy σ will be a switching strategy $\sigma = \langle \sigma_1, \sigma_2, \alpha \rangle$, with σ_2 a deterministic memoryless strategy obtained by an attractor computation towards T, and $\alpha = \max(0, |V|W - \mathsf{mVal}^\rho(v_0)) \times |V| + 1$. In the rest of this section, we will detail how to define σ_1 to obtain the following property:

Proposition 60. *The switching strategy $\sigma = \langle \sigma_1, \sigma_2, \alpha \rangle$ built from the memoryless (randomised) strategy ρ satisfies $\mathsf{dVal}^\sigma(v_0) \leqslant \mathsf{mVal}^\rho(v_0)$.*

We now take care of the construction of σ_1, that selects, for each vertex $v \in V_{\mathsf{Adam}}$ a successor in $\mathrm{supp}(\rho(v))$.Thus, we limit ourselves to the edges present in the MDP $\mathcal{G}^\rho$. For each vertex $v \in V_{\mathsf{Adam}}$ we let $A_v = \mathrm{argmin}_{v' \in \mathrm{supp}(\rho(v))} [w(v, v') + \mathsf{mVal}^\rho(v')]$ the successors of v that minimise the expected value at horizon 1. We let $\mathcal{A}$ be the game obtained from $\mathcal{G}$ by removing all edges $(v, v') \in E$ such that $v' \notin A_v$.

Lemma 61. *(i) Each finite play of $\mathcal{A}$ from a vertex v has a payoff at most $\mathsf{mVal}^\rho(v)$.*
(ii) Each cycle in the game $\mathcal{A}$ has a non-positive total weight.

Proof. We prove the property (i) on finite plays π of $\mathcal{A}$ by induction on the length of π, for all initial vertices v. If π has length 0, this means that $v \in T$, in which case $\mathbf{SP}(\pi) = 0 = \mathsf{mVal}^\rho(v)$. Consider then a play $\pi = v\pi'$ of length at least 1, with π' starting from v', so that $\mathbf{SP}(\pi) = w(v, v') + \mathbf{SP}(\pi')$. By induction hypothesis, $\mathbf{SP}(\pi') \leqslant \mathsf{mVal}^\rho(v')$, so that $\mathbf{SP}(\pi) \leqslant w(v, v') + \mathsf{mVal}^\rho(v')$.

Suppose first that $v \in V_{\mathsf{Eve}}$. By Proposition 49, we know that Eve can play optimally in the MDP $\mathcal{G}^\rho$ with a memoryless deterministic strategy. For each possible memoryless deterministic strategy τ of Eve, we have $\mathsf{mVal}^\rho(u) \geqslant \mathbb{E}_u^{\rho,\tau}(\mathbf{SP})$ for all $u \in V_{\mathsf{Eve}}$, and by the system (1) of equations, letting $u' = \tau(u)$, $\mathbb{E}_u^{\rho,\tau}(\mathbf{SP}) = w(u, u') + \mathbb{E}_{u'}^{\rho,\tau}(\mathbf{SP})$. We thus know that $\mathsf{mVal}^\rho(u) \geqslant w(u, u') + \mathbb{E}_{u'}^{\rho,\tau}(\mathbf{SP})$. By taking a maximum overall memoryless deterministic strategies τ of Eve, Proposition 49 ensures that

$$\forall u \in V_{\mathsf{Eve}} \quad \forall u' \in E(u) \qquad \mathsf{mVal}^\rho(u) \geqslant w(u, u') + \mathsf{mVal}^\rho(u') \tag{4}$$

In particular, $\mathsf{mVal}^\rho(v) \geqslant w(v, v') + \mathsf{mVal}^\rho(v') \geqslant \mathbf{SP}(\pi)$.

If $v \in V_{\mathsf{Adam}}$, then $v' \in A_v$ so that $w(v, v') + \mathsf{mVal}^\rho(v')$ is minimum over all possible successors $v' \in \mathrm{supp}(\rho(v))$. The system (1) of equations implies that, for an optimal strategy χ of Eve,

$$\begin{aligned} \mathsf{mVal}^\rho(v) &= \mathbb{E}_v^{\rho,\chi}(\mathbf{SP}) \sum_{v'' \in E(v)} P(v, v'') \times (w(v, v'') + \mathbb{E}_{v''}^{\rho,\chi}(\mathbf{SP})) \\ &= \sum_{v'' \in \mathrm{supp}(\rho(v))} P(v, v'') \times (w(v, v'') + \mathsf{mVal}^\rho(v'')) \geqslant w(v, v') + \mathsf{mVal}^\rho(v') \end{aligned} \tag{5}$$

so that we also get $\mathsf{mVal}^\rho(v) \geqslant \mathbf{SP}(\pi)$.

We then prove the property (ii) on cycles. Consider thus a cycle $v_1v_2\cdots v_kv_1$ of $\mathcal{A}$, and let $w_1 = w(v_1,v_2), w_2 = w(v_2,v_3),\ldots,w_k = w(v_k,v_1)$ be the sequence of weights of edges. We also let $v_{k+1} = v_1$. We show that $w_1+w_2+\cdots+w_k \leqslant 0$. Let $i \in \{1,2,\ldots,k\}$. If $v_i \in V_{\mathsf{Eve}}$, by (4), $\mathsf{mVal}^\rho(v_i) \geqslant w_i + \mathsf{mVal}^\rho(v_{i+1})$. If $v_i \in V_{\mathsf{Adam}}$, by the reasoning applied above in (5), we also know that $\mathsf{mVal}^\rho(v_i) \geqslant w_i + \mathsf{mVal}^\rho(v_{i+1})$. By summing all the inequalities above, we get

$$\sum_{i=1}^k \mathsf{mVal}^\rho(v_i) \geqslant \sum_{i=1}^k w_i + \sum_{i=1}^k \mathsf{mVal}^\rho(v_i) \qquad \text{i.e.} \quad w_1+w_2+\cdots+w_k \leqslant 0 \qquad \square$$

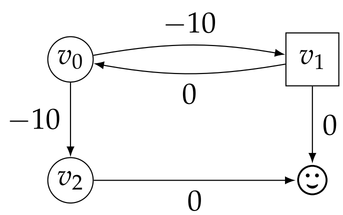
Example 62. *Consider again the game in Figure 7, and the memoryless strategy ρ'_p giving rise to the MDP/MC in Figure 7. Recall that $\mathsf{mVal}^{\rho'_p}(v_0) = -2p^2/(1-p(1-p))$ and $\mathsf{mVal}^{\rho'_p}(v_1) = (p^2-3p+1)/(1-p(1-p))$. Consider p close enough to 1 so that $\mathsf{mVal}^{\rho'_p}(v_0) \leqslant -3/2$ and $\mathsf{mVal}^{\rho'_p}(v_1) \leqslant -1/2$. Then, we have $A_{v_0} = \{v_1\}$ and $A_{v_1} = \{☺\}$. The corresponding game graph $\mathcal{A}$ contains only edges (v_0,v_1) and $(v_1,☺)$, and thus no cycles. The unique finite play from vertex v_0 has payoff $-2 \leqslant \mathsf{mVal}^{\rho'_p}(v_0)$. In particular, the only possible memoryless deterministic strategy σ_1 in $\mathcal{A}$ is optimal in $\mathcal{G}$.*

Contrary to the previous example, not all choices of σ_1 might be good. We thus build one particular memoryless deterministic strategy σ_1 in $\mathcal{A}$, that will be an NC-strategy, i.e. that will follow only negative cycles. By Lemma 61(ii), we naturally have to forbid null cycles. For each vertex $v0$, we let d_v the distance of v to the target given by an attractor to the target in $\mathcal{G}^\rho$. Notice that this may be different from the distance given in the whole arena since some edges are taken with probability 0 in ρ, but still, $d_v < +\infty$ since ρ ensures to reach T with probability 1 (see Example 63). Then, for all vertices $v \in V_{\mathsf{Adam}}$, we let $\sigma_1(v)$ be any vertex v' of A_v that minimises the distance $d_{v'}$.

Example 63. *Consider once again the game in Figure 7, but with a new memoryless strategy ρ''_p defined by $\rho''_p(v_0) = \mathsf{Dirac}_{v_1}$ and $\rho''_p(v_1) = \delta$ such that $\delta(v_0) = 1-p$ and $\delta(☺) = p$, where $p \in (0,1)$. Then, we can check that $\mathsf{mVal}^{\rho''_p}(v_0) = -2$ and $\mathsf{mVal}^{\rho''_p}(v_1) = -1$. Thus, $A_{v_0} = \{v_1\}$ and $A_{v_1} = \{v_0,☺\}$. Not all memoryless deterministic strategies taken in $\mathcal{A}$ are NC-strategies, since it contains the null cycle $v_0v_1v_0$. We thus apply the construction before, using the fact that $d_{☺} = 0$, $d_{v_1} = 1$ and $d_{v_0} = 2$ (since the edge $(v_0,☺)$ is not present in $\mathcal{A}$). Thus, σ_1 is defined by $\sigma_1(v_0) = v_1$ and $\sigma_1(v_1) = ☺$, and is indeed an NC-strategy.*

Lemma 64. *Strategy σ_1 is an NC-strategy, i.e. all cycles of $\mathcal{A}$ conforming with σ_1 have a negative total weight.*

Proof. Let $v_1v_2\cdots v_kv_1$ be a cycle of $\mathcal{A}$ that conforms to σ_1, with v_1 a vertex of minimal distance d_{v_1} among the ones of the cycle. We can choose v_1 such that it belongs to Adam: otherwise, this would contradict the attractor computation in $\mathcal{A}$. By Lemma 61(ii), its total weight is non-positive. Suppose that it is 0. Then, in the proof of Lemma 61(ii), all inequalities $\mathsf{mVal}^\rho(v_i) \geqslant w_i + \mathsf{mVal}^\rho(v_{i+1})$ are indeed equalities. In particular, $\mathsf{mVal}^\rho(v_1) = w_1 + \mathsf{mVal}^\rho(v_2)$. Since $v_2 \in A_{v_1}$, (5) ensures that all successors $v' \in \mathrm{supp}(\rho(v_1))$, $\mathsf{mVal}^\rho(v_1) = w(v_1,v') + \mathsf{mVal}^\rho(v')$. Vertex v_2 has been chosen for $\sigma_1(v_1)$ because it has minimal distance d_{v_2} in all successors $v' \in \mathrm{supp}(\rho(v_1))$. But this is contradicting the fact that v_1 has minimal distance among all vertices of the cycle. $\square$



i	v_0	v_1	v_2
0	$+\infty$	$+\infty$	$+\infty$
1	$+\infty$	0	0
2	-10	0	0
3	-10	-10	0
4	-10	-10	0

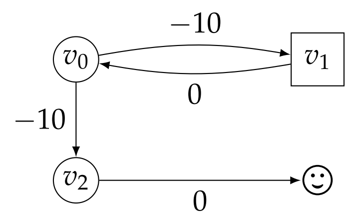


Figure 9: On the left, a shortest-path game for which Adam does not admit an optimal memoryless strategy. In the middle, the computation table of $f^{(i)}$ for all vertex on this game. On the right, the arena of $\mathcal{G}^{A^{(3)}}$ induced by this game.

Proof of Proposition 60. Let π be a play conforming to σ, from vertex v_0. Since σ is a switching strategy, it necessarily reaches T. If σ conforms to σ_1, by Lemma 61(i), it has a payoff $\mathbf{SP}(\pi) \leqslant \mathsf{mVal}^\rho(v_0)$. Otherwise, it is obtained by a switch, and is thus longer than $\alpha = \max(0, |V|W - \mathsf{mVal}^\rho(v_0)) \times |V| + 1$. Then, it contains at least $\max(0, |V|W - \mathsf{mVal}^\rho(v_0))$ elementary cycles, before it switches to the attractor strategy σ_2. Once we remove the cycles, it remains a play of length at most $|V|$, and thus of payoff at most $|V|W$. Since all cycles conforming to σ_1 have a total weight at most -1, by Lemma 64, $\mathbf{SP}(\pi)$ is at most $(-1) \times \max(0, |V|W - \mathsf{mVal}^\rho(v_0)) + |V|W \leqslant \mathsf{mVal}^\rho(v_0)$. □

This concludes the proof of Theorem 50.

4.3 Characterisation of Optimality

All shortest-path games admit an optimal deterministic strategy for both players: however, as we have seen in Example 42, Adam may require memory to play optimally. In this case, we also have seen in Example 48 that Adam does not have an optimal memoryless (randomised) strategy: he only has ε-optimal ones, for all $\varepsilon > 0$. But some shortest-path games indeed admit optimal memoryless strategies for Adam: the strategy ρ_p described in Section 4.1 is indeed optimal in arena not containing negative cycles, for instance. In this section, we characterise the shortest-path games in which Adam admits an optimal memoryless strategy. For sure, Adam does not have an optimal strategy if there is some vertex v of value $\mathsf{dVal}(v) = -\infty$.

Assumption. In this section, we therefore suppose that all shortest-path games are such that $\mathsf{dVal}(v) \neq -\infty$ for all vertices v.

We first recall the deterministic values computations consists of a value iteration based on the operator $\mathcal{F}$ defined by equation (2). We let $f_v^{(0)} = 0$ if $v \in T$ and $+\infty$ otherwise. By monotony of $\mathcal{F}$, the sequence $(f^{(i)} = \mathcal{F}^i(f^{(0)}))_{i \in \mathbb{N}}$ is non-increasing. It is proved to be stationary, and convergent towards $(\mathsf{dVal}(v))_{v \in V}$.

We introduce a new notion, being the most permissive strategy of Adam at each step $i \geqslant 0$ of the computation. It maps each vertex $v \in V_{\mathsf{Adam}}$ to the set $A_v^{(i)} = \{v' \in E(v) \mid w(v,v') + f_{v'}^{(i-1)} = f_v^{(i)}\}$ of vertices that Adam can choose. For each such most permissive strategy $A^{(i)}$, we let $\mathcal{G}^{A^{(i)}}$ be the arena where we remove all edges (v, v') with $v' \notin A_v^{(i)}$.

Example 65. *Consider the game in Figure 5, which we know does not allow an optimal memoryless strategy. The computation of $f^{(i)}$ gives us $f^{(2)}_{v_{\mathsf{Eve}}} = -1$ and $f^{(2)}_{v_{\mathsf{Adam}}} = 0$. This computation has not converged for $i = |V| - 1 = 2$, because $f^{(3)}_{v_{\mathsf{Eve}}} = -1$ and $f^{(3)}_{v_{\mathsf{Adam}}} = -1$. So by Proposition 66,* Adam *does not admit an optimal memoryless strategy.*

Consider the game $\mathcal{G}$ in Figure 9. We iteratively compute, for each vertex, the function f in the middle that converge for $i = |V| - 1 = 3$. Thus, we can deduce $A^{(3)}_v$ for each vertex v. For example, $A^{(3)}_{v_1} = \{v_0\} \neq E(v_1)$ because $f^{(2)}_{v_0} + 0 = -10 = f^{(3)}_{v_0}$, and $f^{(2)}_{☺} + 0 = 0 \neq f^{(3)}_{v_0}$. The game $\mathcal{G}^{A^{(3)}}$ is depicted on the right. In $\mathcal{G}^{A^{(3)}}$, Adam *cannot reach the target. By Proposition 66,* Adam *does not admit an optimal memoryless strategy. In this case, the convergence of $f^{(i)}$ is not sufficient, we need the condition on $\mathcal{G}^{A^{(i)}}$ to conclude the existence of an optimal memoryless strategy.*

Consider the game in Figure 5 such that the output of Adam *is worth -10, so $w(v_{\mathsf{Adam}}, ☺) = -10$. The computation of $f^{(i)}$ gives us $f^{(2)}_{v_{\mathsf{Adam}}} = -10$ and $f^{(2)}_{v_{\mathsf{Eve}}} = -10$. This computation converged for $i = |V| - 1 = 2$, because $f^{(3)}_{v_{\mathsf{Adam}}} = -10$ and $f^{(3)}_{v_{\mathsf{Eve}}} = -10$. Also $A^{(3)}_{v_{\mathsf{Adam}}} = \{☺, v_{\mathsf{Eve}}\}$ so* Adam *can reach the target in $\mathcal{G}^{A^{(2)}}$. So by Proposition 66,* Adam *admits an optimal memoryless strategy.*

Proposition 66. *The following assertions are equivalent:*

1. Adam *has an optimal memoryless deterministic strategy in $\mathcal{G}$ (for* dVal*);*
2. Adam *has an optimal memoryless (randomised) strategy in $\mathcal{G}$ (for $\overline{\mathsf{mVal}}$);*
3. $f^{(|V|-1)}_v = f^{(|V|)}(v) = \mathsf{dVal}(v)$ *for all vertices v (this means that the sequence $(f^{(i)})$ is stationary as soon as step $|V| - 1$), and* Adam *can guarantee to reach T from all vertices in the arena $\mathcal{G}^{A^{(|V|-1)}}$.*

Proof. Implication 1 ⇒ 2 is trivial by the result of Theorem 50.

For implication 3 ⇒ 1, consider any memoryless deterministic strategy σ^* that guarantees Adam to reach T from all vertices in the arena $\mathcal{G}^{A^{(|V|-1)}}$. Then, for all vertices v, we show by induction on n, that each play π from v that reaches the target in at most n steps, and conforming to σ^*, has a payoff $\mathbf{SP}(\pi) \leqslant \mathsf{dVal}(v)$. This is trivial for $n = 0$. If $\pi = v\pi'$ with π' starting in v, then

$$\mathbf{SP}(\pi) = w(v, v') + \mathbf{SP}(\pi') \leqslant w(v, v') + \mathsf{dVal}(v') = w(v, v') + f^{(|V|-1)}(v)$$

If $v \in V_{\mathsf{Eve}}$, we have $\mathbf{SP}(\pi) \leqslant w(v, v') + f^{(|V|-1)}(v) \leqslant f^{(|V|)}(v) = \mathsf{dVal}(v)$. If $v \in V_{\mathsf{Adam}}$, since $v' \in A^{(|V|-1)}_v$, $\mathbf{SP}(\pi) = f^{(|V|)}(v) = \mathsf{dVal}(v)$. This ends the proof by induction. To conclude that 1 holds, since σ^* guarantees to reach the target, all plays conforming to it reach the target in less than $|V|$ steps, which proves that $\mathsf{dVal}^{\sigma^*}(v) \leqslant \mathsf{dVal}(v)$, showing that σ^* is optimal.

For implication 1 ⇒ 3, consider an optimal memoryless deterministic strategy σ^*, such that for all v, $\mathsf{dVal}^{\sigma^*}(v) = \mathsf{dVal}(v)$.

First, we show that $f^{(|V|-1)}(v) = \mathsf{dVal}(v)$ for all vertices v. For that, consider the deterministic strategy τ of Eve defined for all finite plays π having $n \leqslant |V|$ vertices, ending in a vertex $v \in V_{\mathsf{Eve}}$, by $\tau(\pi) = v'$ such that $w(v, v') + f^{(|V|-1-n)}_{v'} = f^{(|V|-n)}_v$. For longer finite plays, we define τ arbitrarily. Then, let π be the play from v conforming to σ^* and τ. Since σ^* ensures reaching the target and is memoryless deterministic, π reaches the target in at most $|V| - 1$ steps. Let $\pi = v_0 v_1 v_2 \cdots v_{k-1} v_k$ with $k \leqslant |V|$. Let us show that $\mathbf{SP}(\pi) \geqslant f^{(|V|-1)}_v$. We prove by induction on $0 \leqslant j \leqslant k$ that $\sum_{i=j}^{k-1} w(v_i, v_{i+1}) \geqslant f_{v_j}^{(|V|-1-j)}$. When $j = k-1$, the result is trivial since the sum is $0 = f_{v_k}^{(0)} \geqslant f_{v_k}^{(|V|-1-(k-1))}$. Otherwise, by induction hypothesis $\sum_{i=j}^{k-1} w(v_i, v_{i+1}) \geqslant w(v_j, v_{j+1}) + f_{v_{j+1}}^{(|V|-1-(j+1))}$. If $v_j \in V_{\mathsf{Eve}}$, v_{j+1} is chosen by τ so that $w(v_j, v_{j+1}) + f_{v_{j+1}}^{(|V|-1-(j+1))} = f_{v_j}^{(|V|-1-j)}$. If $v \in V_{\mathsf{Adam}}$, by definition of $\mathcal{F}$, $w(v_j, v_{j+1}) + f_{v_{j+1}}^{(|V|-1-(j+1))} \geqslant f_{v_j}^{(|V|-1-j)}$. We can conclude in all cases, so that $f^{(|V|-1)}(v) = \mathsf{dVal}(v)$ for all vertices v.

Then, we show that Adam can guarantee to reach T from all vertices in the game graph $\mathcal{G}^{A^{(|V|-1)}}$. Let us suppose that this is not the case. Then, there exists a set V' of vertices in which Eve can guarantee to keep Adam for ever, in the game $\mathcal{G}^{A^{(|V|-1)}}$: for all $v' \in V' \cap V_{\mathsf{Adam}}$, $A_{v'}^{(|V|-1)} \subseteq V'$, and for all $v' \in V' \cap V_{\mathsf{Eve}}$, $E(v) \cap V' \neq \emptyset$. Since σ^* guarantees to reach the target, there exists $v \in V' \cap V_{\mathsf{Adam}}$ such that $\sigma^*(v) = v' \notin V'$: then $w(v, v') + \mathsf{dVal}(v') > \mathsf{dVal}(v)$ (here we use that $\mathsf{dVal}(v) = f_v^{(|V|-1)} = f_v^{(|V|)}$). Consider an optimal memoryless deterministic strategy τ^* of Eve in $\mathcal{G}$. Then, the play π from v conforming to σ^* and τ^* starts by taking the edge (v, v') and continues with a play π'. By optimality, we know that $\mathbf{SP}(\pi) = \mathsf{dVal}(v)$ and $\mathbf{SP}(\pi') = \mathsf{dVal}(v')$. However, $\mathbf{SP}(\pi) = w(v, v') + \mathbf{SP}(\pi') = w(v, v') + \mathsf{dVal}(v') > \mathsf{dVal}(v)$ which raises a contradiction.

We finish the proof by showing 2 ⇒ 1. For that, consider an optimal memoryless (randomized) strategy ρ^* for $\overline{\mathsf{mVal}}$. By following the construction of Section 4.2, we build a memoryless deterministic strategy σ_1. Lemma 64 ensures that σ_1 is an NC-strategy so that every cycles conforming to σ_1 has a negative total weight. Let us show that such a negative cycle cannot exist, which will ensure that all plays conforming to σ_1 reach the target, and thus the optimality of σ_1. Suppose that a cycle $v_1 v_2 \cdots v_k v_1$ conforms to σ_1. By following the notations of the proof of Lemma 61(*ii*) we suppose that, v_1 is a Adam's vertex such that its distance d_{v_1} is minimal on the cycle. Note that a such vertex exist, otherwise only Eve has the minimal distance vertices on the cycle and that contradict the attractor notion. As v_1 has a minimal distance, $d_{v_2} \geqslant d_{v_1}$ and there exists $u \in E(v_1)$ such that $d_u < d_{v_1}$, and $\omega(v_1, u) + \mathsf{mVal}^{\rho^*}(u) > \omega_1 + \mathsf{mVal}^{\rho^*}(v_2)$. By (1), $\mathsf{mVal}^{\rho^*}(v_1) > \omega_1 + \mathsf{mVal}^{\rho^*}(v_2)$. By optimality of ρ^*, this rewrites in $\overline{\mathsf{mVal}}(v_1) > \omega_1 + \overline{\mathsf{mVal}}(v_2)$. By Theorem 50, this also rewrites in $\mathsf{dVal}(v_1) > \omega_1 + \mathsf{dVal}(v_2)$. As $v_1 \in V_{\mathsf{Adam}}$, this contradicts the fact that the vector dVal is a fixpoint of $\mathcal{F}$, that conclude the proof. □

This characterisation is testable in polynomial time since it is enough to compute vectors $f^{(|V|-1)}$ and $f^{(|V|)}$, check their equality, compute the sets of sets $A_v^{(|V|-1)}$ (this can be done while computing $f^{(|V|)}$) and check whether Adam can guarantee reaching the target in $\mathcal{G}^{A^{(|V|-1)}}$ by an attractor computation. The proof of implication 3 ⇒ 1 allows one to build an optimal memoryless deterministic strategy in this case.

5 Shortest-Path Timed Games

We want to extend our results to infinite games with weights. One way to consider infinite games is to add time. As in quantitative games, weights allow us to measure the performance of a given strategy. However, strategies in timed games require two choices for each player: the choice of time spent in a location and the next available transition. Weights are therefore added on the transitions and locations in a timed automaton to obtain a priced timed game. Thus, payoffs depend on the transitions taken as well as the delays spent in each location. We will study these games

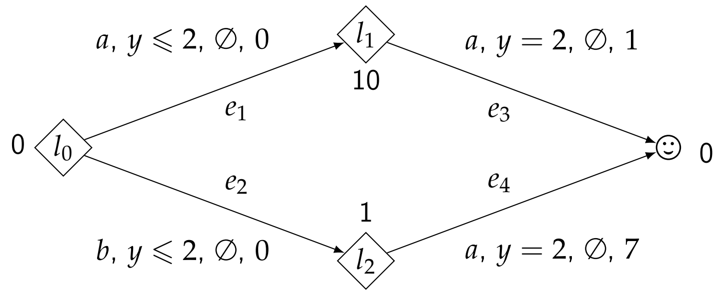


Figure 10: A concurrent shortest-path timed game with a target set $T = \{v_3\}$, Adam's action is $\{a\}$, and Eve's action is $\{b\}$.

with the shortest-path objective. Unfortunately, for the two semantics of the game (concurrent and turn-based), the value problem and the existence problem of an optimal strategy are undecidable.

5.1 Weighted Timed Games with Shortest-Path Objective

A *weighted timed game* is a timed game with weights in its arena. It is finitely represented by a weighted timed automaton that is a timed automaton with weights on theses transitions and locations.

Definition 67. A *weighted timed automaton* is a tuple $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}, w\rangle$ where L is a finite set of locations, $\mathcal{C}$ is a finite set of clocks, Σ is a finite alphabet, $\mathfrak{d} \subseteq L \times \Sigma \times \mathsf{Guard}(\mathcal{C}) \times L \times 2^{\mathcal{C}}$ is a set of transitions, $\mathsf{Inv} : L \to \mathsf{Guard}(\mathcal{C})$ maps an invariant on each locations and $w : L \cup \mathfrak{d} \to \mathbb{Z}$ is a weighted function.

A play in a weighted timed game is an infinite sequence of pairs of locations and delays spend in it, denoted $\pi = (l_i, t_i)_i \in (L, \mathbb{R}_{\geqslant 0})^\omega$. As for a timed game, there exist two semantics for theses games : concurrent games and turn-based games.

Example 68. *The weighted timed automaton in Figure 10 is defined by* $L = \{l_0, l_1, l_2, ☺\}$, $\mathcal{C} = \{y\}$, $\Sigma = \{a, b\}$, $\mathfrak{d} = \{e_1 = (l_0, a, \{y \leqslant 2\}, l_1, \emptyset), e_2 = (l_0, b, \{y \leqslant 2\}, l_2, \emptyset), e_3 = (l_1, a, \{y = 2\}, ☺, \emptyset), e_4 = (l_2, a, \{y = 2\}, ☺, \emptyset)\}$, *and* $\mathsf{Inv} : Q \to \mathit{true}$. *Moreover, weights are defined by* $w(l_0) = w(l_3) = w(e_1) = w(e_2) = 0$, $w(l_1) = 10$, $w(l_2) = w(e_2) = 1$, *and* $w(e_4) = 7$.

The objectives on priced timed games are described by a payoff function as in quantitative games. This function computes the cost of a play regarding the taken transitions, and the time spent in each location. The payoff of a *shortest-path* objective, as in quantitative games, for a set T of vertices and all runs $\pi = (l_i, t_i)_i$, is

$$\mathbf{SP}(\pi) = \begin{cases} +\infty & \text{if } l_k \notin T \text{ for all } k \geqslant 0 \\ \sum_{i=0}^{k-1} w((l_i, l_{i+1})) + t_i w(l_i) & \text{if } k \geqslant 0 \text{ is the minimal index such that } l_k \in T \end{cases}$$

For a shortest-path timed game $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}, w, \mathbf{SP}\rangle$, as in quantitative games, we can define $\overline{\mathsf{Val}}$ and $\underline{\mathsf{Val}}$ the values that Adam and Eve can ensure whatever the adversary plays.

5.2 Concurrent Shortest-Path Timed Games

In a concurrent shortest-path timed game, Σ is partitioned into two sets Σ_{Adam} and Σ_{Eve} as already studied before. In the general case, the value problem and the existence problem of an optimal strategy are undecidable.

Definition 69. A *concurrent shortest-path timed game* is a tuple $\langle L, \mathcal{C}, \Sigma_{\mathsf{Adam}}, \Sigma_{\mathsf{Eve}}, \mathfrak{d}, \mathsf{Inv}, w, \mathbf{SP}\rangle$ where $\Sigma = \Sigma_{\mathsf{Adam}} \uplus \Sigma_{\mathsf{Eve}}$ is the set of action partitioned into two sets : Adam's actions and Eve's ones, and $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}, w, \mathbf{SP}\rangle$ is a shortest-path timed game.

To keep it simple, we assume that the strategies are built on the Γ sets (see Section 2.4). During these games, it is assumed that the action $(\perp, 0)$ is not available.

Example 70. *Consider the concurrent shortest-path timed game in Figure 10 with* $\Sigma_{\mathsf{Adam}} = \{a\}$ *and* $\Sigma_{\mathsf{Eve}} = \{b\}$. Adam*'s value in* $*(l_0, \nu_0)$ *where* $\nu_0 = 0$ *is*

$$\overline{\mathsf{Val}}(l_0, \nu_0) = \inf_{0 \leqslant t \leqslant 2} \max(10(2-t)+1, (2-t)+7) = \inf_{0 \leqslant t \leqslant 2} \max(21-10t, 9-t) = 1$$

The optimal strategy for Adam *in* (l_0, ν_0) *is* $\sigma^*(l_0, \nu_0) = (a, 2)$*. The* Eve*'s value in* l_0 *is*

$$\underline{\mathsf{Val}}(l_0, \nu_0) = \sup_{0 \leqslant t \leqslant 2} \min(21-10t, 9-t) = 9$$

The optimal strategy for Eve *in* $\tau^*(l_0, \nu_0)$ *is* $\tau^*(l_0, \nu_0) = (b, 0)$.

Theorem 71 ([5])**.** *Let $\mathcal{G}$ be a concurrent shortest-path game with non-negative weights.*

1. *Given a threshold $\bowtie c$, the value problem asks whether $\inf_{\sigma \in \mathsf{d}\mathcal{S}_{\mathsf{Adam}}} \mathbf{SP}(\pi) \bowtie c$ where π is conforming with σ. It is undecidable.*

2. *Given a threshold $\bowtie c$ and a vertex v, the existence problem asks whether there is a strategy σ for* Adam *such that for every strategy τ for* Eve*, it holds* $\mathsf{Play}(v, \tau, \sigma) \bowtie c$*. It is undecidable.*

Related work on decidable concurrent shortest-path timed games There exist subclasses of games where the value problem [5] and the existence problem of an optimal strategy are decidable [4]. Theses games verify the non-Zenoness hypothesis to bound the game. Under this hypothesis, the payoff is finite only if the play is finite. Formally, it is given by a technical restriction on the region automaton (see Section 6). For a game under the non-Zenoness hypothesis, the existence problem of an optimal strategy is decidable [4]. If this hypothesis is relaxed slightly, the value problem is still approximable [5].

5.3 Turn-Based Shortest-Path Timed Games

We consider a turn-based shortest-path timed game with arbitrary weights. In a turn-based game, L is partitioned into two sets L_{Adam} and L_{Eve} as already studied before.

Definition 72. A *turn-based shortest-path timed game* is a tuple $\langle L_{\mathsf{Adam}}, L_{\mathsf{Eve}}, T, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}, w, \mathbf{SP}\rangle$ where $L = L_{\mathsf{Adam}} \uplus L_{\mathsf{Eve}} \uplus T$ is the set of locations partitioned into three sets : Adam's actions, Eve's ones and T the target, and $\langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}, w, \mathbf{SP}\rangle$ is a shortest-path timed game.

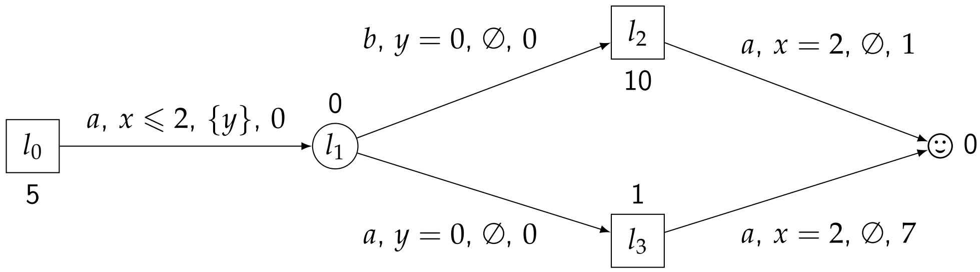


Figure 11: A turn-based shortest path timed game with two clocks.

Example 73. *Consider the turn-based shortest-path timed game in Figure 11. We can compute the value for* Adam *in* (l_0, ν_0) *with* $\nu_0 = (0,0)$ *as*

$$\overline{\mathsf{Val}}(l_0, \nu_0) = \inf_{0 \leqslant t \leqslant 2} \max(5t + 10(2-t) + 1, 5t + (2-t) + 7) = \inf_{0 \leqslant t \leqslant 2} \max(21 - 5t, 9 + 4t) = \frac{43}{3}$$

The optimal strategy for Adam *in* (l_0, ν_0) *is* $\sigma^*(l_0, \nu_0) = (a, \frac{4}{3})$, *and* $\sigma^*(v_2, \nu_1) = \sigma(v_3) = (a, \frac{2}{3})$ *with* $\nu_1 = (4/3, 0)$. *In fact, a play* π *conformed to* σ^* *from* (l_0, ν_0) *admits a cost* $20/3$ *in* (l_1, ν_1). *Then Eve must play with a null delay to* (l_2, ν_1) *and*

$$\mathbf{SP}(\pi) = \frac{20}{3} + 10 \times \frac{2}{3} + 1 = \frac{43}{3}$$

or with a null delay to (l_3, ν_1) *and*

$$\mathbf{SP}(\pi) = \frac{20}{3} + \frac{2}{3} + 7 = \frac{43}{3}$$

Turn-based shortest-path timed games are determined. For these games, the existence problem of an optimal strategy is undecidable for two clocks or more. However, we do not know if it is decidable for only one clock.

Theorem 74 ([8])**.** *The existence of an optimal strategy problem for a turn-based shortest-path timed game is undecidable for two clocks or more.*

Simple priced timed games Simple priced timed games are a subclass of turn-based shortest-path timed games where the existence problem of an optimal strategy and the value problem are decidable [6]. It contains only one clock x that it never reset and its valuation is bounded by 1. Moreover, all guards over x are $0 \leqslant x \leqslant 1$.

Definition 75. A *simple priced timed game* is represented by a tuple $\langle L_{\mathsf{Adam}}, L_{\mathsf{Eve}}, T, \mathcal{C}, \Sigma, \mathfrak{d}, w, \mathbf{SP} \rangle$ where $L = L_{\mathsf{Adam}} \uplus L_{\mathsf{Eve}} \uplus T$, $\mathcal{C} = \{x\}$, $\mathfrak{d} \subseteq L \times \Sigma \times L$ and $w : L \cup \mathfrak{d} \to \mathbb{Z}$. In this game, the total time spent is between 0 and 1. When x reaches 1, the current player must play with a delay null.

The value problem and existence problem of optimal strategy are decidable for those games.

Theorem 76 ([6])**.** *Let* $\mathcal{G}$ *be a simple priced timed game. The value problem can be computed in exponential time. It is solved, for all vertices, by a pair of optimal strategy* σ *for* Adam, *and* τ *for* Eve.

Sketch of the proof. The main idea is to reduce the simple priced timed game on a quantitative game. It is based on urgent locations where the current player must play with a delay null. When all locations in a game are urgent, time cannot pass, and the game is quantitative. For this particular game, compute the value and optimal strategy algorithms for quantitative games work. The idea of the transformation from a simple priced timed game to a quantitative game is the following: at each step, let consider the location with minimal weight to choose the delay as follow Eve would spend minimal time in it, and Adam would spend the maximal time. □

The optimal strategies computed by this algorithm have the same structure as the switching strategies for quantitative games. It is computed in the game where all locations are urgent. There exists an optimal memoryless strategy for Eve. However, an optimal strategy for Adam is finite-memory [6].

Related work on decidable games There exist other subclasses of decidable games. The previous result on simple priced timed games can be extended to games with one clock where the number of resets is bounded, i.e. there are no cycles in the region automaton that contains a reset. Based on the same technique that for the simple priced timed game, the value problem can be solved in exponential time [6]. Let consider some other restrictions to obtain a decidable game.

A *bi-valued priced timed game* has only one clock that is bounded by the greatest constant in the guards. Moreover, we suppose that the weight on the vertices is in $\{-d, 0, d\}$ for $d \in \mathbb{N}$ [8]. Bi-valued priced timed games are determined. The value and the existence of optimal strategy problems are decidable for these games. Moreover, an optimal strategy for Adam uses finite memory and an optimal strategy for Eve may need infinite memory. The optimal strategy can be approximated in pseudo-polynomial time [8].

A *divergent weighted timed game* (see Section 6.1) is a game under a generalisation of the strictly non-Zenoness hypothesis in the case of non-negative weights. In this case, there are no restrictions on the number of clocks of the game. The value problem is in 2EXPTIME and EXPTIME-hard [11]. The existence problem of an optimal strategy is 2EXPTIME [12].

6 Divergent Shortest-Path Timed Games

In this section, we focus on divergent shortest-path timed games for which the value problem is decidable. They are formally defined by the concept of region automaton. In divergent shortest-path timed games, we can define a memoryless (randomized) value to extend our previous result (see Section 4). Thus, we need to introduce randomness into such games.

We denote by $\mathbb{Q}_N = \{n/2^N \mid n \in \mathbb{N}\}$ be a subset of granularity N of $\mathbb{Q}_N$. For all $a \in \mathbb{R}_{>0}$, $\lfloor a \rfloor \in \mathbb{N}$ denotes the integral part of a, and $\mathsf{fract}(a) \in [0, 1)$ its fractional part, such that $a = \lfloor a \rfloor + \mathsf{fract}(a)$.

6.1 Regions and Region Automaton

Regions are a finite partition of $\mathbb{R}_{\geqslant 0}^{\mathcal{C}}$ to abstract the semantics of a timed automaton as a finite state system. They are induced by an equivalence relation on valuations. In particular, in each region, a transition is available for all valuations: a guard is satisfied or not. Time has the same effect on all valuations for each region. For all valuations in a region, all valuations reach the same region

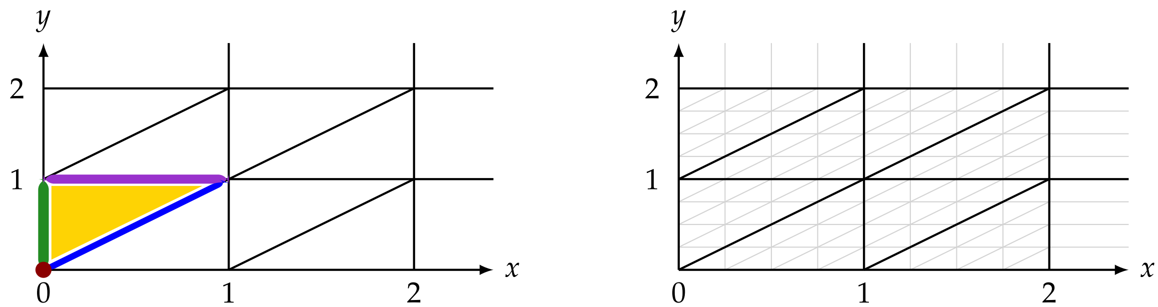


Figure 12: On the left, regions for two clocks where the maximal constant is 2. On the right, regions with a granularity $1/2^2$ for two clocks where the maximal constant is 2.

when delay pass. This remains true for clocks resets, they reach a region where reset clocks have a null value. To ensure that the partition is finite, we use an upper bound on the value of the clocks. This constant is defined as the maximum constant appearing in $\mathsf{Guard}(\mathcal{G})$ of the timed automaton $\mathcal{G}$.

Definition 77. Let $\mathcal{C}$ be a set of clocks, $N \in \mathbb{N}^*$ be a granularity, and M be a maximal constant. $1/2^N$*-regions* are subsets of valuations r characterised by a valuation $\imath \in \mathbb{R}^{\mathcal{C}}_{\geqslant 0, \leqslant M}$ called the integral part of r such that $\imath(c) \in \mathbb{Q}_N$ for every $c \in \mathcal{C}$, and an ordered partition $R_0 \uplus R_1 \uplus \cdots \uplus R_m$ splitting $\mathcal{C}$ into $m+1$ subsets. The ordered partition is denoted $0 = R_0 < R_1 < \cdots < R_m$, where R_0 can be empty but $R_i \neq \emptyset$ for $1 \leqslant i \leqslant m$.

We denote by $\mathsf{Reg}_N(\mathcal{C}, M)$ the set of $1/2^N$ regions over $\mathcal{C}$ with a maximal constant M. The regions with a granularity 0 are the classical regions [10]. A valuation ν in $\mathbb{R}^{\mathcal{C}}_{\geqslant 0}$ belongs to a $1/2^N$ region r if

- for all $c \in \mathcal{C}$, $\imath(c)2^N = \lfloor \nu(c)2^N \rfloor$;
- for all $c \in R_0$, $\mathsf{fract}(\nu(c)2^N) = 0$;
- for all $0 \leqslant i \leqslant m$, for all $c, c' \in R_i$, $\mathsf{fract}(\nu(c)2^N) = \mathsf{fract}(\nu(c')2^N)$.
- for all i, j such that $0 \leqslant i < j \leqslant m$, for all $c \in R_i$ and all $c' \in R_j$, $\mathsf{fract}(\nu(c)2^N) < \mathsf{fract}(\nu(c')2^N)$.

Example 78. *The left of Figure 12 represents regions over two clocks with granularity $N = 0$ and a maximal constant $M = 2$. The red region contains only one valuation $(0,0)$ and is characterized by $\imath = (0,0)$ and $0 = \{x, y\}$. The set of valuations $\{(0,y) \mid 0 < y < 1\}$ is the green region. It is characterized by $\imath = (0,0)$ and $0 = \{x\} < \{y\}$. The set of valuations $\{(x,1) \mid 0 < x < 1\}$ is the purple region. It is characterized by $\imath = (0,1)$ and $0 = \{y\} < \{x\}$. The set of valuation $\{(x,y) \mid 0 < x < 1\}$ is the blue region. It is characterized by $\imath = (0,0)$ and $0 < \{x,y\}$. A valuation in a yellow region is for example $(0.5, 0.6)$. It is characterised by $\imath = (0,0)$ and $0 < \{x\} < \{y\}$. The right of Figure 12 represents regions over two clocks with granularity $N = 2$ and a maximal constant $M = 2$.*

A region automaton is a finite automaton that is built by labelling configurations with their region and collapsing configurations with the same location and region. It abstracts paths in a timed automaton by a sequence of regions alternating between time elapsing and taking transitions. This automaton accepts the untimed language of a timed automaton by simulating its execution.

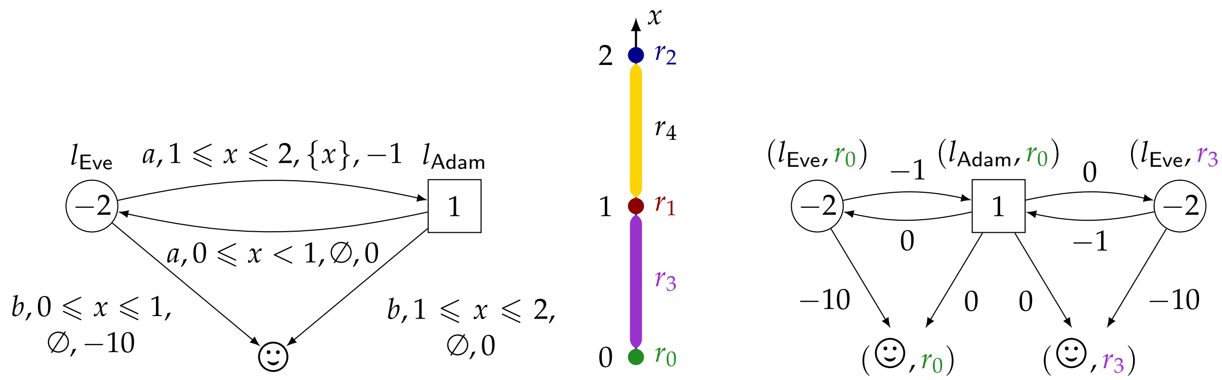


Figure 13: On the left, a divergent shortest-path timed game. In the middle, regions for one clock to the game. On the right, a partial region automaton from the game.

Definition 79. A *region automaton* of a timed automaton $\mathcal{A} = \langle L, \mathcal{C}, \Sigma, \mathfrak{d}, \mathsf{Inv}\rangle$ is a finite automaton $\mathcal{R}(\mathcal{A}) = \langle L \times \mathsf{Reg}_N(\mathcal{C}, M), \mathcal{C}, \Sigma, \mathfrak{d}', \mathsf{Inv}\rangle$ whose locations are labelled by regions and such that $\mathfrak{d}'$ contains all transitions $((l,r), r'', e, (l', r'))$ such that $e = (l, a, g, l', C)$ is a transition of $\mathcal{A}$, r'' is a time successor of r (i.e. there exists $v \in r$, $v'' \in r''$, and $d > 0$ such that $v'' = v + d$), r'' satisfies g, and $r''[C := 0] = r'$.

Example 80. *The right of Figure 13 depicts a partial region automaton from the game on the left. It is the reachable states from the initial configuration $(l_{\mathsf{Adam}}, 0)$ depicted by (l_{Adam}, r_3).*

A region automaton allows us to build two types of games: a timed game and an finite game. A timed game on the region automaton denoted $\mathcal{R}(\mathcal{G})$ where transitions are labelling by guards. It is another way to represent $\mathcal{G}$ by giving more information: the regions. It corresponds to a classical timed game labelling (and duplicating) the vertices (and transitions) by the regions. Let $\mathcal{G}$ be a game in the left of Figure 13, a play in $\mathcal{R}(\mathcal{G})$ is given by $(((l_{\mathsf{Adam}}, r_0,), 0.5)((l_{\mathsf{Eve}}, r_3), 1))^*((l_{\mathsf{Eve}}, r_3), 0)(☺, r_3)$.

A second game denoted $\Gamma_N(\mathcal{G})$, can be built using a region automaton. In this game, players play at the *corners* of the regions [10]. A valuation v is at the corner of a region $1/2^N\, r$ if it belongs to the topological closure of r and its coordinates are in $\mathbb{Q}_N$. Each region has $|\mathcal{C}| + 1$ corners maximum. For example, consider regions in the middle of Figure 13. Corner of r_0 is 0, its unique valuation. Corners of r_3 are 0 and 1. In such a game the vertices are tuples (l, r, v) containing a location l, a region r and a corner v of r. The transitions are then described by the existence of a delay and an available transition between two vertices. For a fine granularity, this game can be untimed by removing the guards.

6.2 Divergent Games and Deterministic Values

The class of divergent shortest-path timed games is defined by a restriction over the total weight of a play. In these games, cycles having a total weight as close to 0 as wanted are prohibited. For example, the shortest-path timed game in the left of Figure 13 is divergent. To formalize this condition, we use the region automaton.

Definition 81. A shortest-path timed game $\mathcal{G}$ is *divergent* if every finite play π in $\mathcal{G}$ following a cycle in the region automaton $\mathcal{R}(\mathcal{G})$ satisfies $\mathbf{TP}(\pi) \notin (-1, 1)$.

Recall, in a timed game, for all configurations (l, ν), a deterministic strategy for Adam makes a choice in

$$\Gamma_{\mathsf{Adam}}(l, \nu) = \{(a, t) \mid a \in \Sigma \text{ labels an available transition from } (l, \nu) \text{ for the delay } t\}$$

Formally, a deterministic strategy for Adam is a matching $\sigma : (L \times \mathbb{R}^{\mathcal{C}})^*(L_{\mathsf{Adam}} \times \mathbb{R}^{\mathcal{C}}) \to \Gamma_{\mathsf{Adam}}$. A deterministic strategy for Eve is analogously defined.

As for quantitative games, we formally define the value of strategies σ and τ by letting for all configurations (l, ν),

$$\mathsf{dVal}^{\sigma}(l, \nu) = \sup_{\tau' \in \mathsf{d}\mathcal{S}_{\mathsf{Eve}}} \mathbf{SP}(\mathsf{Play}((l, \nu), \tau', \sigma)) \qquad \text{and} \qquad \mathsf{dVal}^{\tau}(l, \nu) = \inf_{\sigma' \in \mathsf{d}\mathcal{S}_{\mathsf{Adam}}} \mathbf{SP}(\mathsf{Play}((l, \nu), \tau, \sigma'))$$

Finally, the value for Adam and for Eve is : for all vertices (l, ν),

$$\overline{\mathsf{dVal}}(l, \nu) = \inf_{\sigma \in \mathsf{d}\mathcal{S}_{\mathsf{Adam}}} \mathsf{dVal}^{\sigma}(l, \nu) \qquad \text{and} \qquad \underline{\mathsf{dVal}}(l, \nu) = \sup_{\tau \in \mathsf{d}\mathcal{S}_{\mathsf{Eve}}} \mathsf{dVal}^{\tau}(l, \nu)$$

Moreover, a divergent shortest-path timed game is determined and we note for all configurations (l, ν), $\mathsf{dVal}(l, \nu) = \overline{\mathsf{dVal}}(l, \nu) = \underline{\mathsf{dVal}}(l, \nu)$ the common value.

Example 82. *Consider the divergent shortest-path game on the left of Figure 13. First, let us consider the configuration $(l_{\mathsf{Adam}}, 0)$ as initial.* Adam *could directly reach the target, thus leading to a payoff of* 1. *But he can also choose to go to l_{Eve} with a delay* 0*, in which case* Eve *either jumps directly in the target (leading to a beneficial payoff* -11*), or comes back to l_{Adam}, but having already capitalized a total payoff* -3*.* Adam *can continue this way* ad libitum *until he is satisfied (at least 4 times) and jumps to the target. This guarantees a value at most* -10 *for* Adam*, when starting in $(l_{\mathsf{Adam}}, 0)$. At each turn, he is in the configuration $(l_{\mathsf{Adam}}, 0)$. Reciprocally,* Eve *can guarantee a payoff at least* -10 *by directly jumping into the target when she plays for the first time. Thus, the value is* -10 *when starting from $(l_{\mathsf{Adam}}, 0)$.*

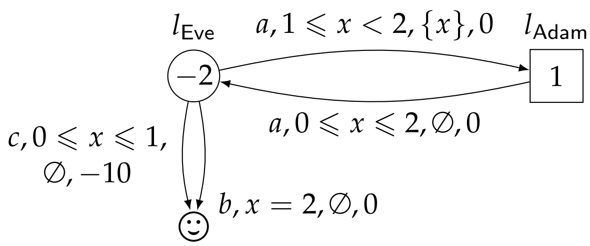
The value is given by fixpoint computation based on the operator $\mathcal{F}_{temp}$ that summarises a turn of the game [10]. It is based on *value function*, a mapping from a configuration $L \times \mathbb{R}_{\geqslant 0}$ to a value in $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$. Formally $\mathcal{F}_{temp} \colon ((L \times \mathbb{R}^{\mathcal{C}}_{\geqslant 0} \to \overline{\mathbb{R}}) \to (L \times \mathbb{R}^{\mathcal{C}}_{\geqslant 0} \to \overline{\mathbb{R}})$ defined for all $x = (x_{(l,\nu)})_{(v,\nu) \in L \times \in \mathbb{R}^{\mathcal{C}}_{\geqslant 0}} \in (L \times R^{\mathcal{C}}_{\geqslant 0} \to \overline{\mathbb{R}})$ and all configurations $(l, \nu) \in L \times R^{\mathcal{C}}_{\geqslant 0}$ by

$$\mathcal{F}_{temp}(x)_{(l,\nu)} = \begin{cases} 0 & \text{if } v \in T \\ \inf_{(l,\nu) \xrightarrow{a,t} (l',\nu')} (w(l, l') + t w(l) + x_{(l',\nu')}) & \text{if } v \in L_{\mathsf{Adam}} \\ \sup_{(l,\nu) \xrightarrow{a,t} (l',\nu')} (w(l, j') + t w(l) + x_{(l',\nu')}) & \text{if } v \in L_{\mathsf{Eve}} \end{cases} \tag{6}$$

where $(l, \nu) \xrightarrow{a,t} (l', \nu')$ is an available transition from (l, ν) labelled by action a and delay t.

6.3 How to Play at Random in Timed Games?

To define a randomized strategy for Adam, we need to define a distribution over all possible actions Γ_{Adam}. Formally, a memoryless strategy for Adam is a mapping $\rho : (L_{\mathsf{Adam}} \times \mathbb{R}^{\mathcal{C}}_{\geqslant 0}) \to$



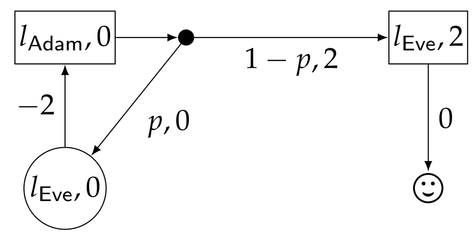


Figure 14: On the left, a divergent shortest-path timed game from a configuration need two delays to play optimally. On the right, a MC induced by strategies with a finite support for distributions.

$\Delta(\Gamma_{\text{Adam}})$ matching each configuration of Adam to a distribution on Adam's possible choices. Thus, the stochastic model induced is an infinite (with an infinite degree) MDP or MC. The notion of value is more subtle to define. However, we hope to succeed in defining the value with integrals as in the models presented in [13, 18].

In this internship, we limit, for each configuration (l, ν), the distribution support to finite support. In other words, we restrict $\Gamma_{\text{Adam}}(l, \nu)$ to a finite set of possible choices. However, this restriction allows for several delays per transition. Formally, a memoryless strategy for Adam is defined by $\rho : (L_{\text{Adam}} \times \mathbb{R}^{\mathcal{C}}_{\geqslant 0}) \to \Delta(\Gamma'_{\text{Adam}})$ where Γ'_{Adam} is a finite subset of Γ_{Adam}. A memoryless strategy for Eve is defined analogously.

Example 83. *Consider the divergent shortest-path game in the left of Figure 14 and an initial configuration $(l_{\text{Adam}}, 0)$. The deterministic value for $(l_{\text{Adam}}, 0)$ is -10 ensured by an optimal switching strategy $\sigma = \langle \sigma_1, \sigma_2, \alpha \rangle$. It is defined by $\sigma_1(l_{\text{Adam}}, 0) = (a, 0)$ and $\sigma_2(l_{\text{Adam}}, 0) = (a, 2)$ to ensure that* Eve *chooses the target. The MC on the right induced by ρ defined by $\rho(l_{\text{Adam}}, 0)(a, 0) = p$, $\rho(l_{\text{Adam}}, 0)(b, 2) = 1 - p$, and τ defined by $\tau(l_{\text{Eve}}, 0) = (a, 1)$ and $\tau(l_{\text{Eve}}, 0) = (b, 0)$. If we suppose that each transition has only one delay, $p = 0$ or $p = 1$ in ρ, then total-weight of conformed play is* 1 *or $+\infty$ that is greater than -10.*

However, even finite support for distribution for each configuration, the MC induced by two memoryless strategies can be infinite. This MC can have an infinite set of vertices if strategies build an infinite set of reachable configurations. However, this MC fas finite degree. The definition of value then remains analogous to quantitative games.

Example 84. *Consider the divergent shortest-path game depicted on the left of Figure 15. Consider the memoryless strategy for* Adam*, ρ, such that $\rho(l_{\text{Adam}}, x)(a, 0) = p_1$, $\rho(l_{\text{Adam}}, x)(b, 2 - x) = p_2$, and $\rho(l_{\text{Adam}}, x)(c, 0) = p_3$ for $0 \leqslant x < 1$, and $p_1 + p_2 + p_3 = 1$, and $\rho(l_{\text{Adam}}, x)(b, 2 - x) = 1$ for $1 \leqslant x \leqslant 2$. The memoryless strategy for* Eve*, χ, is defined by $\chi(l_{\text{Eve}}, 0)(a, 1/2) = 1$, $\chi(l_{\text{Eve}}, (2^n - 1)/2^n)(a, 1/2^{n+1}) = 1$, for all $n \in \mathbb{N}^*$. A play conformed to these strategies is $(l_{\text{Adam}}, 0)(l_{\text{Eve}}, 0)(l_{\text{Adam}}, 1/2)(l_{\text{Eve}}, 1/2)(l_{\text{Adam}}, 3/4)(l_{\text{Eve}}, 3/4)\ldots$ until* Adam *switch to the target after some reset of clocks with probability* 1.

In $\mathcal{G}^{\rho,\chi}$ induced by memoryless strategies for Adam and Eve, we denote by $\mathbb{P}^{\rho,\chi}_v(\diamond T)$ the probability of the set of plays that reach the target set $T \subseteq V$ of vertices. Similarly, $\mathbb{E}^{\rho,\chi}_v(\mathbf{SP})$ is the expected weight of a path in this MC, weights being the ones taken from $\mathcal{G}$. We can define the value with an expectation over the MC. We therefore define the value of strategy ρ as the worst-case scenario for him:

$$\mathsf{mVal}^{\rho}(v) = \sup_{\chi \in \mathsf{m}\mathcal{S}_{\text{Eve}}} \mathbb{E}^{\rho,\chi}_v(\mathbf{SP})$$

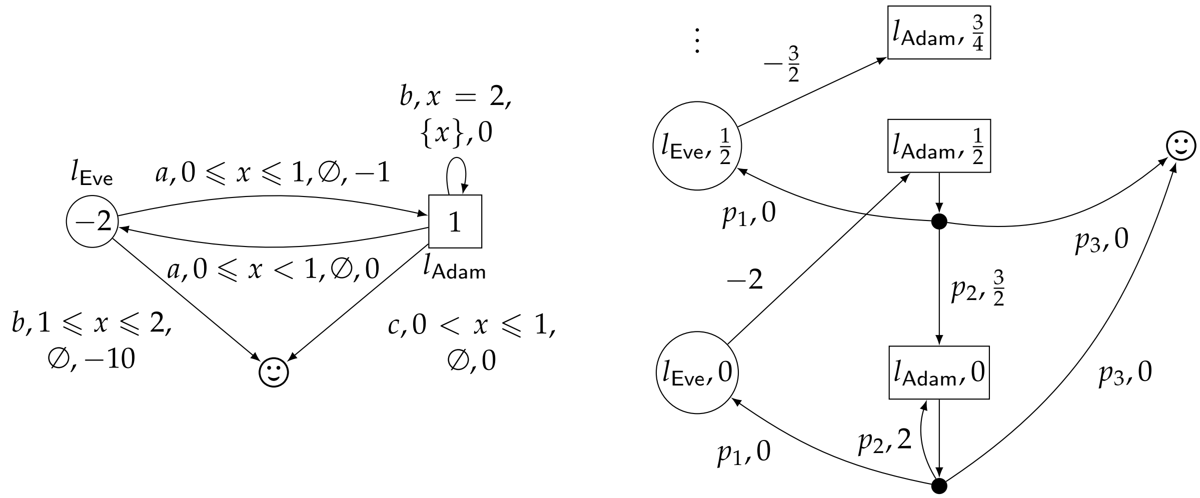


Figure 15: On the left, a divergent shortest-path timed game that can induce an infinite MC. On the right, a partial MC obtained with a strategy with a distribution of finite support.

This definition only makes sense (otherwise it is $+\infty$) if $\mathbb{P}_v^{\rho,\chi}(\diamond T) = 1$ for all χ, i.e. if strategy ρ ensures the reachability of a target vertex with probability 1, no matter how the opponent plays. In this case, letting P the probability mapping defining the MC $\mathcal{G}^{\rho,\chi}$, the vector $(\mathbb{E}_v^{\rho,\chi}(\mathbf{SP}))_{v \in V}$ is the only solution to the system of equations

$$\mathbb{E}_l^{\rho,\chi}(\mathbf{TP}) = \begin{cases} 0 & \text{if } v \in T \\ \sum_{(a,t) \in \Gamma_{\mathsf{Adam}}(l)} P(l,l') \times (w(l,l') + tw(l) + \mathbb{E}_{l'}^{\rho,\chi}(\mathbf{SP})) & \text{if } l \in L_{\mathsf{Adam}} \\ \sum_{(a,t) \in \Gamma_{\mathsf{Eve}}(l)} P(l,l') \times (w(l,l') + tw(l) + \mathbb{E}_{l'}^{\rho,\chi}(\mathbf{SP})) & \text{if } l \in L_{\mathsf{Eve}} \end{cases} \tag{7}$$

where l' is the location reached by the available transition labelled with a for the delay t.

Since Adam wants to minimise the shortest-path payoff, we finally define the memoryless upper value as

$$\overline{\mathsf{mVal}}(v) = \inf_{\rho \in \mathsf{m}\mathcal{S}_{\mathsf{Adam}}} \mathsf{mVal}^{\rho}(v)$$

6.4 Memoryless Values

The next contribution will consist in showing that values in divergent shortest-path timed games are the same when restricting both players to memoryless or deterministic strategies:

Conjecture 85. *For all divergent shortest-path timed games $\mathcal{G}$, for all configurations (l, ν), we have* $\mathsf{dVal}(l,\nu) = \overline{\mathsf{mVal}}(l,\nu)$.

In this section, we give our intuitions towards the proof of this conjecture. To simplify, we will only consider configurations with finite values. Moreover, by the same argument as in quantitative games (see Section 4), we can rule out the configurations with values $+\infty$. For configurations (l, ν) with value $-\infty$, we need to show that we can ensure value as low as we want. To this end, we use the same construction as in quantitative games where the value is replaced by $\max(-n, \mathsf{dVal}(l,\nu))$ or $\max(-n, \overline{\mathsf{mVal}}(l,\nu))$ for all $n \in \mathbb{N}$.

$$
\begin{array}{ccccccc}
\mathsf{dVal}_{\mathcal{G}} & \overset{\varepsilon}{=} & \mathsf{dVal}^{P}_{\mathcal{G}} & \underset{?}{\overset{\varepsilon}{=}} & \mathsf{dVal}^{P'}_{\Gamma_N(\mathcal{G})} & \overset{\varepsilon}{=} & \mathsf{dVal}_{\Gamma_N(\mathcal{G})} \\
 & & & & & & \| \text{ (Theorem 50)} \\
\mathsf{mVal}_{\mathcal{G}} & & \xleftarrow[\text{memoryless random strategy}]{\hspace{6cm}} & & & & \mathsf{mVal}_{\Gamma_n(\mathcal{G})}
\end{array}
$$

Figure 16: Scheme of proof. We write $x \overset{\varepsilon}{=} y$ to denote $|x - y| \leqslant \varepsilon$.

Let (l, ν) be an initial configuration in a divergent shortest-path timed game $\mathcal{G}$. To prove that $\mathsf{dVal}(l, \nu) \geqslant \overline{\mathsf{mVal}}(l, \nu)$ we use some approximations of the deterministic value, depicted in Figure 16, to approximate the deterministic value in $\mathcal{G}$ with a deterministic value in a finite game, $\Gamma_N(\mathcal{G})$. Thus, by Theorem 50, we obtain an approximation of the deterministic value in $\mathcal{G}$ with a memoryless strategy in $\Gamma_N(\mathcal{G})$. Finally, we obtain the memoryless value in $\mathcal{G}$ with a memoryless randomized strategy built from an ε-optimal memoryless strategy in $\Gamma_N(\mathcal{G})$.

We present this approximation sequence. The deterministic value in a divergent shortest-path timed game $\mathcal{G}$ is computed by a fixpoint of $\mathcal{F}_{temp}$ (6). There is, therefore, $P \in \mathbb{N}$ such that the P-th iteration of $\mathcal{F}_{temp}$ approaches $\mathsf{dVal}_{\mathcal{G}}(l, \nu)$. We denote by $\mathsf{dVal}^{P}_{\mathcal{G}}(l, \nu)$ the value computed by the P-th iteration of $\mathcal{F}_{temp}$ and $\mathsf{dVal}_{\mathcal{G}}(l, \nu) \overset{\varepsilon}{=} \mathsf{dVal}^{P}_{\mathcal{G}}(l, \nu)$. For the last approximation, we consider $\Gamma_N(\mathcal{G})$ a finite game induced by $\mathcal{G}$. Let $\mathsf{dVal}^{P'}_{\Gamma_N(\mathcal{G})}(l, \nu)$ the value induced by the P'-th iteration of $\mathcal{F}$ (2). By the same approximation on the fixpoints (see Section 3.3), there exists $P' \in \mathbb{N}$ such that $\mathsf{dVal}_{\Gamma_N(\mathcal{G})}(l, \nu) \overset{\varepsilon}{=} \mathsf{dVal}^{P'}_{\Gamma_N(\mathcal{G})}(l, \nu)$. To conclude this approximation sequence, we need to prove $\mathsf{dVal}^{P}_{\mathcal{G}}(l, \nu) \overset{\varepsilon}{=} \mathsf{dVal}^{P'}_{\Gamma_N(\mathcal{G})}(l, \nu)$. We need to find the good $N \in \mathbb{N}$ such that the value in the timed game is ε-close to the value in the finite game. We want to build the finite game with an iterative calculation of $\mathcal{F}_{temp}$ so that each iteration on $\mathcal{F}_{temp}$ to a region $1/2^N$ is an epsilon-approximation. Thus the subdivision on the result of the P-th iteration is our finite game.

Reciprocally, we will use the same proof technique as in Section 4.2: simulating a randomized memoryless strategy with a finite-memory deterministic strategy. The deterministic strategy is based on the value induced by the randomized memoryless strategy in each vertex. In a quantitative game, the cycles of value zero induce an additional difficulty: null cycle can not reach by a switching strategy. To address this difficulty, we use a distance based on an attractor to the target. In the case of divergent shortest-path timed games, such cycles do not exist by assumption, so this additional difficulty should not occur.

7 Conclusion

In this internship, we studied some games on graphs: quantitative games and weighted timed games with a shortest-path objective. These games are based on the same formalism: an arena (a finite or infinite weighted graph) and an objective combining a qualitative objective (reachability) and a quantitative objective (total-payoff). In the context of these games, we solve the value problem giving the interest of a player regarding the type of strategies used. We studied the tradeoff between memory and randomness in the context of computing the optimal values.

In quantitative games, we have shown that Adam guarantees the same value when he is limited to deterministic strategies or randomized memoryless strategies. For this, we simulate deterministic strategies with an ε-optimal randomized memoryless strategy where probabilities are parametrized by ε. Reciprocally, we simulate a randomized memoryless strategy with a switching strategy. We also studied the existence of optimal memoryless strategies, which appears to be equivalent to the existence of optimal memoryless deterministic strategies, and testable in polynomial time.

In shortest-path timed games, the value problem and the existence of an optimal strategy problem are both undecidable for at least two clocks. We, therefore, focus on a class of games for which these problems are decidable: divergent shortest-path timed games. For these games, we conjecture that Adam can guarantee the same value with deterministic or memoryless strategies but with randomness. We give a proof scheme using as a black-box the result obtained in finite quantitative games.

Future works

Many questions remain open following this internship. We detail some of them here.

In quantitative games, we could also define more general lower and upper values $\underline{\mathsf{Val}}(v)/\overline{\mathsf{Val}}(v)$ when we let Adam and Eve play unrestricted strategies (randomized and with memory). The results of Blackwell's determination [20] imply that, for such unrestricted strategies, the shortest-path games are always determined so that $\overline{\mathsf{Val}}(v) = \underline{\mathsf{Val}}(v) = \mathsf{Val}(v)$. A natural question is whether $\mathsf{Val}(v) = \mathsf{dVal}(v)$ The reasoning of Section 4.2 only used the vector of values $(\mathsf{mVal}^{\rho}(v))_{v \in V}$ to define the deterministic switching strategy σ, without using anywhere that ρ is memoryless. We have, therefore, shown that $\mathsf{dVal}(v) \leqslant \mathsf{Val}(v)$. Conversely, however, the proof in Section 4.1 cannot be directly translated if we allow Adam to use memory and randomization. In particular, we no longer know how Eve can react, which could break the result of Proposition 54.

Also, in these games, the value is computed in pseudo-polynomial time [7]. We hope that randomized strategies give us a polynomial-time algorithm to compute the value. Indeed, if we succeeded in computing an epsilon-optimal strategy in polynomial time for ε small enough, arounding procedure would them allow us to get the exact values in polynomial time.

In divergent shortest-path timed games, we want to prove the conjecture 85. For that, we need to check that the value of a timed game is well approximable by the value of a finite game. Also, we wish to study the generalization of the concept of value when the distribution used in the strategies is more general, than just finite support. However, defining the concept of value for such strategies is more challenging in particular because of the necessary mathematical tools (integrals) but also by the intuition that this concept carries.

Acknowledgement

I thank all the people who allowed me to do my internship in the best possible conditions, despite the exceptional situation of the past few months.
I would especially like to thank my supervisors, Pierre-Alain Reynier and Benjamin Monmege, for their welcome, their time and their help throughout this internship.
I also wanted to thank the people with whom I shared my office as well as the laboratory doctoral students for their hospitality and support throughout this internship.
I will finish with a special thank you for Nathalie Bertrand who allowed me to do this internship. Her help and support were of great importance to me.

References

[1] Christel Baier, Nathalie Bertrand, Clemens Dubslaff, Daniel Gburek, and Ocan Sankur. Stochastic shortest paths and weight-bounded properties in markov decision processes. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 86–94. ACM, 2018.

[2] Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.

[3] Dimitri P. Bertsekas and John N. Tsitsiklis. An analysis of stochastic shortest path problems. *Math. Oper. Res.*, 16(3):580–595, 1991.

[4] Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim Guldstrand Larsen. Optimal strategies in priced timed game automata. In *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science, 24th International Conference, Chennai, India, December 16-18, 2004, Proceedings*, pages 148–160, 2004.

[5] Patricia Bouyer, Samy Jaziri, and Nicolas Markey. On the value problem in weighted timed games. In *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, pages 311–324, 2015.

[6] Thomas Brihaye, Gilles Geeraerts, Axel Haddad, Engel Lefaucheux, and Benjamin Monmege. Simple priced timed games are not that simple. In *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*, pages 278–292, 2015.

[7] Thomas Brihaye, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege. Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games. *Acta Informatica*, 54, 07 2016.

[8] Thomas Brihaye, Gilles Geeraerts, Shankara Narayanan Krishna, Lakshmi Manasa, Benjamin Monmege, and Ashutosh Trivedi. Adding negative prices to priced timed games. In *CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings*, pages 560–575, 2014.

[9] Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-Francois Raskin. Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games. *Leibniz International Proceedings in Informatics, LIPIcs*, 25, 03 2014.

[10] Damien Busatto-Gaston. *Symbolic controller synthesis for timed systems: robustness and optimality*. Theses, Aix Marseille Université, December 2019.

[11] Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. Optimal reachability in divergent weighted timed games. In *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, pages 162–178, 2017.

[12] Damien Busatto-Gaston, Benjamin Monmege, and Pierre-Alain Reynier. Symbolic approximation of weighted timed games. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, pages 28:1–28:16, 2018.

[13] Yuliya Butkova, Ralf Wimmer, and Holger Hermanns. *Long-Run Rewards for Markov Automata*. 01 2017.

[14] Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. Trading memory for randomness. In *Proceedings of the The Quantitative Evaluation of Systems, First International Conference*, QEST '04, pages 206–217, Washington, DC, USA, 2004. IEEE Computer Society.

[15] Krishnendu Chatterjee, Laurent Doyen, Hugo Gimbert, and Youssouf Oualhadj. Perfect-information stochastic mean-payoff parity games. pages 210–225, 2014.

[16] Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdziński. Mean-payoff parity games. In *Proceedings of the 20th Annual Symposium on Logic in Computer Science (LICS'05)*, pages 178–187. IEEE Computer Society Press, 2005.

[17] Krishnendu Chatterjee, Thomas A. Henzinger, and Vinayak S. Prabhu. Trading infinite memory for uniform randomness in timed games. In *Hybrid Systems: Computation and Control, 11th International Workshop, HSCC 2008, St. Louis, MO, USA, April 22-24, 2008. Proceedings*, pages 87–100, 2008.

[18] Luca de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In Jos C. M. Baeten and Sjouke Mauw, editors, *CONCUR'99 Concurrency Theory*, pages 66–81, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[19] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research*. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[20] Donald A. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.

[21] Benjamin Monmege. *Games on Graphs*, chapter 4 : Payoffs. Unpublished, 2019.

[22] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

A Sketch of Proofs for Quantitative Games

Theorem 86 ([7]). *Deterministic values of a shortest-path game can be computed in pseudo-polynomial time.*

Sketch of the proof. We present the main idea of this algorithm. It iteratively computes in pseudo-polynomial time the deterministic value as a fixpoint of $\mathcal{F}$. However this computation does not converge when vertices have an infinite value.

Find vertices v such that $\mathsf{dVal}(v) = +\infty$ By definition of the payoff, a vertex has a value $+\infty$ if and only if Adam cannot reach the target from it. A classical attractor technique with the same equations as in Example 8 computes the set of vertices $V_{+\infty} = \{v \in V \mid \mathsf{dVal}(v) = +\infty\}$.

Find vertices v such that $\mathsf{dVal}(v) = -\infty$ A vertex has a deterministic value $-\infty$ if and only if Adam controls a negative cycle. They are exactly those whose a negative value in a mean-payoff game defined on the same arena. With this equivalence between the shortest-path and the mean-payoff games, a threshold can be deduced. When the iterative computed value is less than this threshold, its value is necessary $-\infty$. □

Theorem 87 ([7]). *In a shortest-path game,*

1. Eve *has an optimal deterministic memoryless strategy computable in pseudo-polynomial time.*
2. *For all vertices with a finite deterministic value,* Adam *has an optimal deterministic pseudo-polynomial memory strategy computable in pseudo-polynomial time. For all vertices with deterministic value $-\infty$, there exists a sequence of optimal deterministic finite-memory strategies computable in pseudo-polynomial time such that its value converges to $-\infty$.*

Deterministic values of a shortest-path game can be computed in pseudo-polynomial time.

Sketch of the proof. Let give us the main argument to compute the strategy for Eve (item 1). For all vertices v such that $\mathsf{dVal}(v) = +\infty$, the strategy for Eve is an attractor where the target is $V_{+\infty}$ the vertices set with a deterministic value $+\infty$. For all vertices v such that $\mathsf{dVal}(v) = -\infty$ all strategies for Eve are equally bad. For other vertices, Eve's optimal strategy τ^* is $\tau^*(\pi v) = \mathrm{argmax}_{v' \in E(v)}\{w(v,v') + \mathsf{dVal}(v')\}$. It is memoryless and its optimality is proved by induction on the play's size.

Now, we give the main argument to compute the strategy for Adam (item 2). For all vertices v such that $\mathsf{dVal}(v) = +\infty$, all strategies for Adam are equally bad. Otherwise, Adam builds a switching strategy that is a combination of two memoryless strategies. The first one is computed with the same technique as for Eve: $\sigma_1(\pi v) = \mathrm{argmin}_{v' \in E(v)}\{w(v,v') + \mathsf{dVal}(v')\}$ is a memoryless strategy. It reaches a negative cycle or a shortest-path to the target. The second one, computed in polynomial time, σ_2, is a classical attractor for Adam to reach the target. With these two strategies, we define σ^n such that it is conformed to σ_1 until the size of the current play is at least $\alpha = (2W(|V|-1)+n)|V|+1$, next it is conformed to σ_2. By induction on n, authors prove that the total-weight of a play from v consistent with σ^n is at least $\max(-n, \mathsf{dVal}(v))$. If $\mathsf{dVal}(v) > -\infty$, there exist n such that σ^n is optimal. Otherwise, the sequence of $(\sigma^n)_n$ has a value to converge to $-\infty$. □

B Computations for Proof of Proposition 53

B.1 Computations for $\gamma_{0,\mathbb{N}} + \gamma_{<I,\geqslant L} \leqslant \mathsf{dVal}^\sigma(v_0)$

When $\mathsf{dVal}^\sigma(v_0) < 0$ and $\Pi_{<I,\geqslant L} \neq \emptyset$, it remains to show under which conditions over p,

$$S = \sum_{i=1}^{I-1} \sum_{\ell=L}^{\infty} (1-p)^i p^\ell \left(i w^+ + \left\lfloor \frac{\ell - ic - |V|}{c} \right\rfloor (-w^-) + |V|W \right) \leqslant \mathsf{dVal}^\sigma(v_0)$$

Upper-bounding $\left\lfloor \frac{\ell-ic-|V|}{c} \right\rfloor (-w^-)$ by $\left(\frac{\ell-ic-|V|}{c} - 1 \right)(-w^-) = \frac{\ell-ic-|V|-c}{c}(-w^-)$, we can split the double sum S in three parts:

$$S = (w^+ + w^-) \underbrace{\sum_{i=1}^{I-1} \sum_{\ell=L}^{\infty} (1-p)^i p^\ell i}_{S_1} \quad - \quad \frac{w^-}{c} \underbrace{\sum_{i=1}^{I-1} \sum_{\ell=L}^{\infty} (1-p)^i p^\ell \ell}_{S_2}$$
$$+ \quad \left(\frac{-|V| - c}{c}(-w^-) + |V|W \right) \underbrace{\sum_{i=1}^{I-1} \sum_{\ell=L}^{\infty} (1-p)^i p^\ell}_{S_3}$$

Using the fact that $L \geqslant 2$ ($L = I\gamma + \frac{2|\mathsf{dVal}^\sigma(v_0)|+|V|W}{w^-}c + |V| > |V| > 1$ otherwise, for the unique $v \in V$, $\mathsf{dVal}(v) = 0$ or $+\infty$ regarding $v \in T$ or not), we have

$$S_1 \leqslant \sum_{i=1}^{\infty} i(1-p)^i \sum_{\ell=2}^{\infty} p^\ell = \frac{1-p}{p^2} \times \frac{p^2}{1-p} = 1$$

$$S_3 \leqslant \sum_{i=1}^{\infty} (1-p)^i \sum_{\ell=1}^{\infty} p^\ell = \frac{1-p}{p} \times \frac{p}{1-p} = 1$$

and

$$\begin{aligned}
S_2 &= \sum_{i=1}^{I-1} (1-p)^i \sum_{\ell=L}^{\infty} p^\ell \ell \\
&= (1-p)\frac{1-(1-p)^{I-1}}{p} \times \frac{p^L(-Lp+L+p)}{(1-p)^2} \\
&= \frac{(1-(1-p)^{I-1})p^{L-1}(-Lp+L+p)}{1-p} \\
&\geqslant \frac{(1-(1-p)^{I-1})p^L}{1-p} && \text{(since } -Lp+L \geqslant 0\text{)} \\
&\geqslant \frac{1}{4(1-p)} && \text{(since } p \geqslant \tfrac{1}{2^{1/L}} \geqslant \tfrac{1}{2} \text{ and } 1-(1-p)^{I-1} \geqslant \tfrac{1}{2} \text{ by } 0 \leqslant I\text{)}
\end{aligned}$$

Therefore, we obtain

$$S \leqslant (w^+ + w^-) - \frac{w^-}{c}\frac{1}{4(1-p)} + \left(\frac{-|V| - c}{c}(-w^-) + |V|W \right)$$

The right term goes towards $-\infty$ when $p \to 1$. In particular, when

$$p \geqslant 1 - \frac{w^-}{4(cw^+ + 2cw^- + |V|w^- + c|V|W - \mathsf{dVal}^\sigma(v_0)c)}$$

we obtain

$$S \leqslant \mathsf{dVal}^\sigma(v_0)$$

B.2 Computations for $\gamma_{\geqslant I,\mathbb{N}} \leqslant \varepsilon/4$

It remains to show that

$$(1-p)^I \left(\frac{w^+}{p} I + \frac{w^+(1-p)}{p^2} + \frac{|V|W}{p}\right) \leqslant \frac{\varepsilon}{4}$$

We let here $\delta = \frac{2|\mathsf{dVal}^\sigma(v_0)|+|V|W}{w^-}c + |V|$ so that $L = I\gamma + \delta$. Since, $p \geqslant LW/(LW + \varepsilon/4) = (I\gamma W + \delta W)/(I\gamma W + \delta W + \varepsilon/4)$,

$$1 - p \leqslant \frac{\varepsilon/4}{I\gamma W + \delta W + \varepsilon/4} = \frac{1}{4I\gamma W/\varepsilon + 4\delta W/\varepsilon + 1}$$

By also using that $p \geqslant 1/2 \geqslant 1/4$, thus $1/p \leqslant 4$ and $1/p^2 \leqslant 4$, we obtain

$$\gamma_{\geqslant I,\mathbb{N}} \leqslant \left(\frac{1}{4I\gamma W/\varepsilon + 4\delta W/\varepsilon + 1}\right)^I (4w^+ I + 4(w^+ + |V|W))$$

The value $4I\gamma W/\varepsilon + 4\delta W/\varepsilon + 1$ being greater than 1, we can write

$$\gamma_{\geqslant I,\mathbb{N}} \leqslant \left(\frac{1}{4I\gamma W/\varepsilon + 4\delta W/\varepsilon + 1}\right)^{I-1} \left(4w^+ \frac{I}{4I\gamma W/\varepsilon + 4\delta W/\varepsilon + 1} + 4(w^+ + |V|W)\right)$$

Since $x/(ax+b) \leqslant 1/a$ whenever $a, x, b \geqslant 0$, we have $\frac{I}{4I\gamma W/\varepsilon + 4\delta W/\varepsilon + 1} \leqslant \frac{\varepsilon}{4\gamma W}$. Moreover, $\frac{4I\gamma W}{\varepsilon} + \frac{4\delta W}{\varepsilon} + 1 > \frac{I\gamma W}{2\varepsilon}$ and thus

$$\gamma_{\geqslant I,\mathbb{N}} \leqslant \left(\frac{2\varepsilon}{I\gamma W}\right)^{I-1} \left(\frac{\varepsilon w^+}{\gamma W} + 4(w^+ + |V|W)\right)$$

But

$$\left(\frac{2\varepsilon}{I\gamma W}\right)^{I-1} \left(\frac{\varepsilon w^+}{\gamma W} + 4(w^+ + |V|W)\right) \leqslant \frac{\varepsilon}{4}$$

if and only if

$$\left(\frac{I\gamma W}{2\varepsilon}\right)^{I-1} \geqslant \frac{4w^+}{\gamma W} + \frac{16(w^+ + |V|W)}{\varepsilon} \geqslant \frac{2w^+}{\gamma W} + \frac{8(w^+ + |V|W)}{\varepsilon}$$

if and only if

$$(I-1)\ln\left(\frac{I\gamma W}{2\varepsilon}\right) \geqslant \ln\left(\frac{2w^+}{\gamma W} + \frac{8(w^+ + |V|W)}{\varepsilon}\right) = \ln\left(\frac{\xi\gamma W}{2\varepsilon}\right)$$

where $\xi = \frac{4\varepsilon w^+}{\gamma^2 W^2} + \frac{16(w^+ + |V|W)}{\gamma W}$. Consider ε small enough so that $\gamma W/2\varepsilon \geqslant 1$ and $\xi\gamma W/2\varepsilon \geqslant 2$ (the two terms tend to $+\infty$ when ε tends to 0). Then, $(I-1)\ln\left(\frac{I\gamma W}{2\varepsilon}\right) \geqslant (I-1)\ln(I)$, and it is sufficient to prove that

$$(I-1)\ln(I) \geqslant \ln\left(\frac{\xi\gamma W}{2\varepsilon}\right)$$

Since the mapping $I \mapsto (I-1)\ln(I)$ is increasing, and $I \geqslant \frac{\xi\gamma W}{2\varepsilon}$ (by definition),

$$(I-1)\ln(I) \geqslant \left(\frac{\xi\gamma W}{2\varepsilon} - 1\right)\ln\left(\frac{\xi\gamma W}{2\varepsilon}\right) \geqslant \ln\left(\frac{\xi\gamma W}{2\varepsilon}\right)$$

B.3 Lower Bound over p

If we gather all the lower bounds over p that we need in the proof, we get that:

- if $\mathsf{dVal}^\sigma(v_0) \geqslant 0$, we must have

$$p \geqslant \max\left(\frac{LW}{LW + \varepsilon/4}, \frac{1}{2}\right)$$

- if $\mathsf{dVal}^\sigma(v_0) < 0$, we must have

$$\max\left(\frac{LW}{LW + \varepsilon/4}, \frac{1}{2^{1/L}}, \left(1 - \frac{\varepsilon}{2|\mathsf{dVal}^\sigma(v_0)|}\right)^{\frac{1}{|V|}}, \right.$$
$$\left. 1 - \frac{w^-}{4(cw^+ + 2cw^- + |V|w^- + c|V|W + |\mathsf{dVal}^\sigma(v_0)|c)}\right)$$

  with ε small enough so that this bound is less than 1.