Solving Simple Stochastic Tail Games

Hugo Gimbert, Florian Horn

Solving Simple Stochastic Tail Games

Hugo Gimbert[1] and Florian Horn[2]

[1] CNRS, LaBRI, Université Bordeaux 1, Bordeaux, France
hugo.gimbert@labri.fr
[2] IST Austria, Vienna,Austria
florian.horn@ist.ac.at

Abstract. Stochastic games are a natural model for open reactive processes: one player represents the controller and his opponent represents a hostile environment. The evolution of the system depends on the decisions of the players, supplemented by random transitions. There are two main algorithmic problems on such games: computing the values (quantitative analysis) and deciding whether a player can win with probability 1 (qualitative analysis). In this paper we reduce the quantitative analysis to the qualitative analysis: we provide an algorithm for computing values which uses qualitative analysis as a sub-procedure. The correctness proof of this algorithm reveals several nice properties of perfect-information stochastic tail games, in particular the existence of optimal strategies. We apply these results to games whose winning conditions are boolean combinations of mean-payoff and Büchi conditions.

1 Introduction

There is a long tradition of using infinite games to model open reactive processes [BL69,PR89]. The system is represented as a game arena, *i.e.* a graph whose vertices belong either to Eve (controller), Adam (non-deterministic environment), or Random (stochastic evolution). A step of the game consists in moving a token on the arena: when it is in one of Eve's vertices, she chooses its next location among the successors of the current vertex; when it is in one of Adam's vertices, he chooses its next location; when it is in a random vertex, its next location is decided by a fixed random function. The game is played for infinitely many steps and the winner of the play depends on the infinite path of the token: the *winning condition* is the set of infinite plays winning for Eve. Eve and Adam have conflicting interests a play not winning for Eve is winning for Adam, thus given a strategy for Eve, Adam plays a counter-strategy that lowers as much as possible the winning probability of Eve, and vice-versa. Roughly speaking, the *value of the game* is the supremum of winning probabilities that Eve can guarantee, and by Martin's theorem it coincides with the infimum of winning probabilities that Adam can secure (cf. Section 2 for formal definitions).

The computation of *values* of stochastic games is a challenging algorithmic problem, which has received various answers depending on the class of winning conditions considered. The seminal paper by Shapley [Sha53] dealt with discounted games, more recent work focused on games for verification [dA97], while Hoffman-Karp introduced strategy improvement algorithms for mean-payoff [HK66] and Condon focused on reachability games [Con92].

Another appealing algorithmic problem consists in deciding whether Eve or Adam has an *almost-sure strategy* i.e. a strategy which guarantees to win with probability 1. Due to potential applications in verification of open systems, this question was solved for winning conditions used in verification i.e. safety, reachability, Büchi, parity and Müller conditions [CY95,dA97,Cha07c,Hor08].

Although deciding the existence of almost-sure strategies (the qualitative analysis) may seem similar to computing values (the quantitative analysis), these two problems are quite different. For example the qualitative analysis of simple stochastic games can be carried out in polynomial time [dA97] while for the quantitative analysis of these games only exponential time algorithms are known [Con92]. Even worse, for certain classes of stochastic games with partial observation the existence of almost-sure winning strategies is decidable whereas values are not computable [Paz71,CDHR07,BBG08,BGG09].

In this paper, we focus our attention on *simple* stochastic *tail* games. By "simple" we mean that players have perfect-information and take their decisions turn-by-turn, whereas in stochastic games as introduced initially by Shapley [Sha53] players take their decisions concurrently. A *tail* winning condition is such that the winner of a play does not depend on finite prefixes of the play, only the long-term behaviour of the play matters. This class encompasses games for verification and mean-payoff games. From a verification perspective, tail conditions correspond to cases where local glitches are tolerated in the beginning of a run, as long as the specification is met in the long-run, *e.g.* in self-stabilising protocols.

Some interesting results about stochastic games with concurrent moves and tail winning conditions were obtained by Chatterjee. In particular, the *limit-one property* (if at least one vertex has strictly positive value then at least one vertex has value one) was extended from Müller games [dAH00] to the full class of tail games [Cha07a].

Chatterjee also showed that an algorithm for the qualitative analysis (checking existence of almost-sure strategies) can be used for the quantitative analysis (computing values) of stochastic games with Müller and mean-payoff winning conditions [Cha07c]. This was based on the decomposition of the game arena according to its value classes and on the limit-one property.

We provide several new results, both algorithmic and theoretical, about simple stochastic games with tail winning conditions and finitely many vertices.

Our main algorithmic result is a reduction of the quantitative analysis to the qualitative analysis: we design an algorithm for computing values that relies on a procedure for computing almost-sure winning sets. This implies that if almost-sure winning sets of a game G are computable in time $t(G)$ then values of G are computable in time $|\mathcal{Q}_R|! \cdot t(G)$, where $|\mathcal{Q}_R|$ is the number of random vertices (Theorem 14).

The proof of this algorithmic result reveals two properties of simple stochastic tail games which are of independent interest: they are *qualitatively determined* and both players have *optimal strategies*. Qualitative determinacy means that from every initial vertex of a simple stochastic tail game with finitely many vertices, either Eve has an almost-sure winning strategy or Adam has a positive strategy (Theorem 7). Qualitative determinacy is actually a by-product of the proof of a stronger result: in every simple stochastic tail game with finitely many states, both players have *optimal strategies* (Theorem 9), whereas in general only ε-optimal strategies are guaranteed to exist [Mar98].

As an application, we provide algorithms for computing values of games whose winning condition is a boolean combination of a mean-payoff and a Büchi condition (Section 5).

Our results improve previously known results in several aspects. First, to our knowledge no algorithm was ever proposed for the full class of simple stochastic tail games. Moreover, in the special cases of parity and Müller games, our algorithm is more efficient that Chatterjee's algorithm which requires to guess non-deterministically (i.e. to enumerate) the values of the game and a partition of the arena, while our algorithm only requires to guess non-deterministically a permutation of the random vertices. The existence of optimal strategies was only known for simple stochastic games with mean-payoff or Müller conditions, we extended this result to the full class of simple stochastic tail games, using as a main proof tool *reset-strategies*.

The paper is organized as follows. Section 2 recalls the classical notions about simple stochastic games. Section 3 deals with our qualitative determinacy result, and sketches how the proof uses reset-strategies and their properties. In section 4, we show how to compute values of simple stochastic tail games using an algorithm computing almost-sure winning sets. In the same section, we state our main theoretical result: in finite simple stochastic tail games with finitely many states, both players have optimal strategies. Finally, in section 5 we apply these results to prove computability of values of games whose winning conditions are a boolean combination of a Büchi and a mean-payoff condition.

2 Definitions

We recall here several classical notions about simple stochastic games, and refer the reader to [GTW02] and [dA97] for more details.

Arenas and plays. A *simple stochastic arena* $\mathcal{A}$ is a directed graph $(\mathcal{Q}, \mathcal{T})$ without deadlocks, whose vertices are partitioned between Eve's vertices ($\mathcal{Q}_E$, represented as $\bigcirc$'s), Adam's vertices ($\mathcal{Q}_A$, represented as $\square$'s), and random vertices ($\mathcal{Q}_R$, represented as $\triangle$'s), and supplemented by a function $\delta : \mathcal{Q}_R \to \mathcal{D}(\mathcal{Q})$, which is the random law directing the choice of successors in the random vertices (in particular, $\delta(r)(q) > 0 \Leftrightarrow (r, q) \in \mathcal{T}$). A sub-arena $\mathcal{A}_{|B}$ of $\mathcal{A}$ is the restriction of $\mathcal{A}$ to a subset B of $\mathcal{Q}$ such that each controlled vertex of B has a successor in B, and all the successors of random vertices in B belong to B. A *play* ρ *of* $\mathcal{A}$ is an (possibly infinite) path in the graph $(\mathcal{Q}, \mathcal{T})$. The set of infinite plays, denoted by Ω, can naturally be made into a measurable space $(\Omega, \mathcal{O})$ in the following way: $\mathcal{O}$ is the σ-field generated by the cones $\{\Gamma_w \mid w \in \mathcal{Q}^*\}$, where Γ_w is the set of plays extending w.

Strategies and measures A *pure strategy* σ *for Eve* is a deterministic way of extending finite plays ending in a vertex of Eve: $\sigma : \mathcal{Q}^*\mathcal{Q}_E \to \mathcal{Q}$ is such that $(q, \sigma(wq)) \in \mathcal{T}$. Strategies can also be defined as *strategies with memory*. Given a (possibly infinite) set of memory states M, a strategy σ with memory M is defined by two functions: a "next-move" function $\sigma^{\mathbf{n}} : (\mathcal{Q}_E \times M) \to \mathcal{Q}$ and a "memory-update" function $\sigma^{\mathbf{u}} : (\mathcal{Q} \times M) \to M$. Notice that any strategy can be represented as a strategy with memory $\mathcal{Q}^*$. A play ρ is *consistent with a strategy* σ if and only if $\forall i, \rho_i \in \mathcal{Q}_E \Rightarrow \rho_{i+1} = \sigma(\rho_0, \ldots, \rho_i)$.

Once an initial vertex q and two strategies σ and τ have been fixed, we can define recursively a probability measure $\mathbb{P}_q^{\sigma,\tau}$ over the cones by $\mathbb{P}_q^{\sigma,\tau}(\Gamma_r) = \mathbf{1}_{q=r}$ and:

$$\forall w \in \mathcal{Q}^*, (r, s) \in \mathcal{Q}^2, \mathbb{P}_q^{\sigma,\tau}(\Gamma_{wrs}) = \begin{cases} \mathbb{P}_q^{\sigma,\tau}(\Gamma_{wr}) \cdot \mathbf{1}_{\sigma(wr)=s} \text{ if } r \in \mathcal{Q}_E \ , \\ \mathbb{P}_q^{\sigma,\tau}(\Gamma_{wr}) \cdot \mathbf{1}_{\tau(wr)=s} \text{ if } r \in \mathcal{Q}_A \ , \\ \mathbb{P}_q^{\sigma,\tau}(\Gamma_{wr}) \cdot \delta(r)(s) \quad \text{if } r \in \mathcal{Q}_R \ . \end{cases}$$

By Carathéodory's extension theorem, there is a unique extension of $\mathbb{P}_q^{\sigma,\tau}$ to $(\Omega, \mathcal{O})$.

Winning conditions and values. A *winning condition* Φ is a Borel set of $(\Omega, \mathcal{O})$. An infinite play is *winning for Eve* if it belongs to Φ, and *winning for Adam* otherwise. Finite plays are not winning for either player. A winning condition Φ is a *tail condition* if the winner of a play does not depend on finite prefixes: $\forall w \in \mathcal{Q}^*, \forall \rho \in \mathcal{Q}^\omega, \rho \in \Phi \Leftrightarrow w\rho \in \Phi$.

The *value of* $q \in \mathcal{Q}$ *with respect to the strategies* σ *and* τ *for Eve and Adam* (or *$\sigma\tau$-value*) is defined by: $v_{\sigma,\tau}(q) = \mathbb{P}_q^{\sigma,\tau}(\Phi)$. The *value of* q *with respect to a strategy* σ *for Eve* (or *σ-value*) is the infimum of its $\{\sigma, \tau\}$-values: $v_\sigma(q) = \inf_\tau v_{\sigma,\tau}(q)$. Symmetrically, the *value of* q *with respect to a strategy* τ *for Adam* (or *τ-value*) is the supremum of its $\{\sigma, \tau\}$-values: $v_\tau(q) = \sup_\sigma v_{\sigma,\tau}(q)$. By the quantitative determinacy of Blackwell games [Mar98], the supremum of the σ-values is equal to the infimum of the τ-values. This common value is called the *value of* q:

$$\mathbf{v}(q) = \sup_\sigma v_\sigma(q) = \inf_\tau v_\tau(q) \ .$$

Almost-sure strategies. A strategy σ for Eve is *almost-surely winning* (or *almost-sure*) from a vertex q if and only if the σ-value of q is one. It is positively winning (or positive) from q if and only if for any strategy τ, the $\{\sigma, \tau\}$-value of q is positive (notice that the σ-value of q may be zero). The *almost-sure region of Eve* (resp. *positive region of Eve*) is the set of vertices from which Eve has an almost-sure (resp. positive) strategy.

3 Qualitative Determinacy of Tail Games

Before we tackle the issues regarding the problem of computing the values of finite simple stochastic tail games, we need some preliminary results about what happens in the regions with value zero and value one —the so-called "qualitative problems". The main result of this section is the following theorem, which establishes the existence of almost-sure strategies in these extremal regions:

Theorem 1. *In any finite simple stochastic tail game, Eve has an almost-sure strategy in the region with value one, and Adam has an almost-sure strategy in the region with value zero.*

This theorem is instrumental in the proof of our next-section's algorithm, and is very interesting on its own, since it states that the *limit* and *almost-sure* winning criteria from [dAH00] are equivalent in finite simple stochastic tail games. It follows directly that the *positive* and *bounded* winning criteria are also equivalent. Using the standard reduction to parity games, these results can be extended to finite simple stochastic ω-regular games.

Our main tool in the proof of Theorem 1 is the notion of *reset-strategy*, for which we need to extend σ-values to finite plays. In a nutshell, the σ-value evaluates how much Eve lost since the beginning of the play, and reset strategies can "reboot" if this loss goes beyond a given threshold.

Definition 2. *The σ-value of a finite play w of length ℓ consistent with σ is the infimum of the $\{\sigma, \tau\}$-values under the assumption that w is a prefix of the play:*

$$v_\sigma(w) = \inf_\tau \mathbb{P}^{\sigma,\tau}_{w_0}(\Phi \mid \rho_0 = w_0, \rho_1 = w_1, \ldots, \rho_{\ell-1} = w_{\ell-1}) \ .$$

Using the σ-values of the prefixes, we can observe how the prospects of the players evolve during a play. In particular, for any positive real number η, we define the event Λ^σ_η, corresponding to the plays where Eve's chances of winning have dropped below η at some point:

$$\Lambda^\sigma_\eta = \{\exists i, v_\sigma(\rho_0 \ldots \rho_i) \leq \eta\} \ .$$

This event has two interesting characteristics: first, if the σ-value of the initial vertex is greater than η, the probability that the ensuing play belongs to Λ^σ_η is bounded away from one (Proposition 3); second, the probability that Adam wins is zero outside of Λ^σ_η (Proposition 4).

Proposition 3. *Let q be a vertex of $\mathcal{Q}$, σ and τ be strategies for Eve and Adam, and $\eta < v_\sigma(q)$ a positive real number. We have:*

$$\mathbb{P}^{\sigma,\tau}_q(\Lambda^\sigma_\eta) \leq \frac{1 - v_\sigma(q)}{1 - \eta} \ .$$

Proposition 4. *Let q be a vertex of $\mathcal{Q}$, σ and τ be a strategy for Adam, and η be a positive real number. We have:*

$$\mathbb{P}^{\sigma,\tau}_q(\Phi \mid \neg\Lambda^\sigma_\eta) = 1 \ .$$

These two results suggest a way to improve σ with a "reset" procedure with respect to a given real number η. Assume that Eve plays σ and, at some point, the σ-value of the prefix drops below η, while the σ-value of the current vertex is greater than η. She can improve her chances to win by forgetting the past, and restart playing σ as if the play just started.

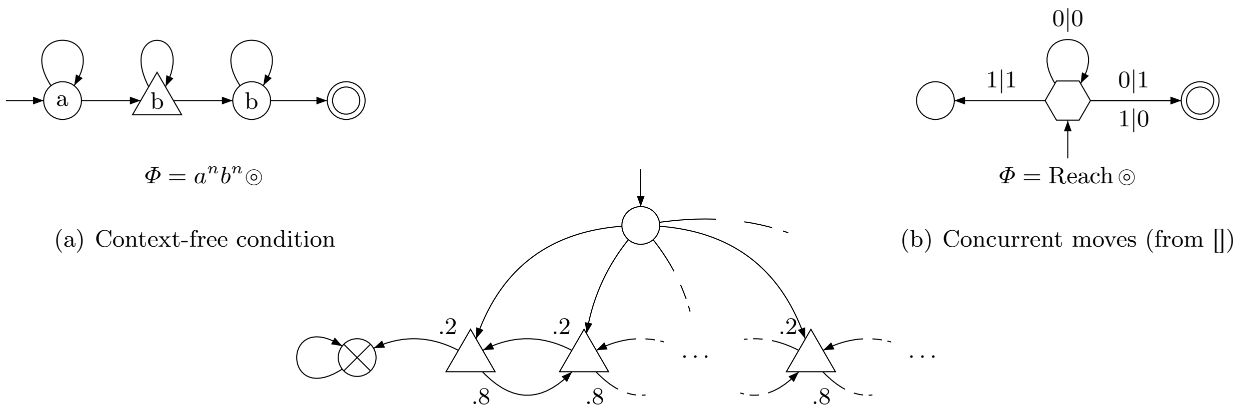
Definition 5. *The strategy σ reset with respect to η, denoted by $\sigma_{\downarrow\eta}$, is a strategy with memory, whose memory states are the plays of $\mathcal{A}$ consistent with σ. Its memory-update and next-move functions are defined as follows:*

$$\sigma^{\mathtt{n}}_{\downarrow\eta}(w, q) = \begin{cases} \sigma(q) & \textit{if } v_\sigma(wq) \leq \eta \textit{ and } v_\sigma(q) > \eta \\ \sigma(wq) & \textit{otherwise} \end{cases}$$

$$\sigma^{\mathtt{u}}_{\downarrow\eta}(w, q) = \begin{cases} q & \textit{if } v_\sigma(wq) \leq \eta \textit{ and } v_\sigma(q) > \eta \\ wq & \textit{otherwise} \end{cases}$$

Proof (of Theorem 1). In order to prove Theorem 1, we just need to choose carefully the initial strategy and the reset trigger. As the arena is finite, we can choose η strictly less than 1 and at the same time strictly greater than the value of any vertex whose value is less than one. Then plays consistent with the reset strategy $\sigma_{\downarrow\eta}$ will stay in the set of vertices with value 1, otherwise a reset occurs just before this set is left. If Eve plays according to σ starting from a vertex with value 1, it follows from Proposition 3 that the probability of one reset when playing is strictly less than 1, hence the number of resets when playing the reset strategy $\sigma_{\downarrow\eta}$ is finite with probability one. Thus, with probability one some suffix of a play has no reset, hence by Proposition 4 this suffix is in Φ. Since Φ is tail, with probability one the whole play itself is won by Eve.

Notice that Theorem 1 cannot be extended to games with context-free conditions, infinite arenas, or concurrent moves: in each of the three games of Figure 1, the value of the initial vertex is one, yet Eve has no almost-sure strategy.



$\Phi = a^n b^n \circledcirc$

(a) Context-free condition

$\Phi = \mathrm{Reach} \circledcirc$

(b) Concurrent moves (from [])

$\Phi = \neg\, \mathrm{Reach} \otimes$

(c) Infinite arena

Fig. 1. In general, limit-sure is not almost-sure

The following theorem is the qualitative counterpart of the limit-one property for concurrent tail games, which states that if at least one vertex has strictly positive value then at least one vertex has value one [Cha07a].

Theorem 6 (Positive-almost property). *In any finite simple stochastic tail game, if Eve has a positive strategy from at least one vertex, she has an almost-sure strategy from at least one vertex.*

As a consequence we obtain qualitative determinacy of simple stochastic tail games.

Theorem 7 (Qualitative determinacy). *In any finite simple stochastic tail game, from any vertex, either Eve has an almost-sure strategy or Adam has a positive strategy, and vice versa.*

Whereas Theorem 1 does not hold for the three games depicted on Figure 1, these three games are qualitatively determined. This gives hope that Theorem 7 may be extended beyond the class of simple stochastic tail games.

4 Computing Values in Tail Games

In recent years, many algorithms were proposed to compute the values of specific classes of finite simple stochastic tail games. These algorithms are often adaptations of algorithms for reachability games which use qualitative algorithms as oracles. For example, one can guess a solution to a set of local consistency equations and use a qualitative algorithm to check necessary and sufficient conditions on the value regions: see [CdAH05] for Rabin games, [Cha07b] for Muller games, and [CHH08] for finitary games. It is also possible to adapt the strategy improvement algorithm of [HK66] when one of the players has positional strategies: see [CJH04] for parity, and [CH06] for Rabin games. Finally, in one-player stochastic tail games (Markov Decision Processes), one can compute first the almost-sure region, and then the values of the reachability game to this region [Cha07a].

In this section, we propose a generic way to compute the values of any tail game, using qualitative oracles and permutation concepts from the algorithm for reachability games of [GH09]. The theoretical complexity of the resulting "meta-algorithm" matches the best known results in each of the cases mentioned before: NP and co-NP membership for quantitative problems in parity games, NP-completeness (resp. co-NP-completeness) in Rabin (resp. Streett) games, and PSPACE-completeness in coloured Muller games. The following theorem unifies and generalises these results:

Theorem 8. *Let $\mathfrak{C}$ be a class of tail conditions. If the problem of deciding whether a vertex is almost-surely winning for Eve in finite simple stochastic $\mathfrak{C}$-games belongs to the complexity class $\mathcal{K}$, then the problem of deciding whether the value of a vertex is greater than $\frac{1}{2}$ in finite simple stochastic $\mathfrak{C}$-games belongs to the classes $\mathtt{NP}^{\mathcal{K}}$ and co-$\mathtt{NP}^{\mathcal{K}}$.*

Furthermore, the permutation algorithm is much more efficient than the "brute-force" approach of guessing the values:

Theorem 9. *Let $\mathfrak{C}$ be a class of tail conditions. If the almost-sure region of Eve in a $\mathfrak{C}$-game $\mathcal{G}$ can be computed in time $t(|\mathcal{G}|)$, then the values of any $\mathfrak{C}$-game $\mathcal{G}$ can be computed in time $|\mathcal{Q}_R+1|!\cdot t(|\mathcal{G}|)$.*

The main idea of our algorithm is that if Adam does not make obvious mistakes, Eve can only hope to win by reaching her almost-sure region. This can only be done through random vertices: there is no vertex of Eve leading to it (it would belong to the almost-sure region), and she cannot hope that Adam will enter it voluntarily (that would be an obvious mistake). The winning condition is then only a tool for Eve to ensure that the token reaches the best possible random vertex: if Adam refuses to comply, he loses with probability one. Consider for example what happens in the game of Figure 2, where the goal of Eve is reaching square vertices infinitely often. By herself, Eve cannot send the token into the top random vertex. However, she can do better than sending the token to the bottom random vertex and win with probability .1. Actually, by repeatedly sending the token to Adam's vertex, she forces him to either send the token to the top vertex or lose the play, and she wins with probability 1.

The behaviour of both players is then determined by their preferences over the random vertices. Furthermore, it is sufficient to consider the cases where Eve and Adam share the same estimation over the respective quality of random vertices, *i.e.*, when their preferences are opposed. These preferences are represented by permutations over the random vertices. In the remainder of the paper, a *permutation* π designates a permutation $(\pi_1, \ldots, \pi_k)$ over the k random vertices, such that $\{\pi_1, \ldots, \pi_k\} = \mathcal{Q}_R$.

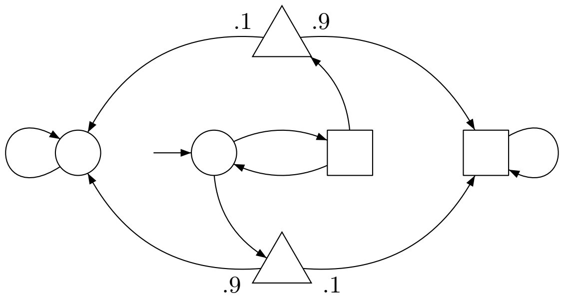
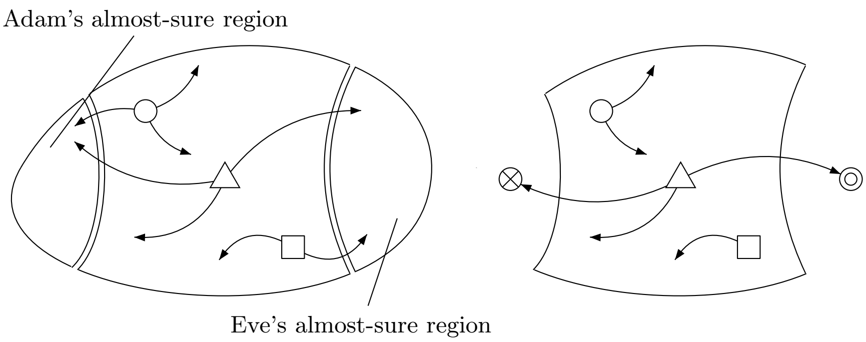


Fig. 2. Winning condition as a tool for Eve

For simplicity (and efficiency), we first *normalise* the games we consider: we compute the almost-sure regions of both players, and merge each of them into a single sink vertex ($\otimes$ for Adam, $\circledcirc$ for Eve). The winning condition is modified accordingly: a play that reaches $\otimes$ is winning for Adam and a play that reaches $\circledcirc$ is winning for Eve. In permutation-based concepts, we often consider the sink and target vertices as random vertices with the implicit assumption that they are respectively the lowest and greatest vertices: $\pi_0 = \otimes$ and $\pi_{k+1} = \circledcirc$.



(a) Original game (b) Normalised game

Fig. 3. Game normalisation

The principle of our algorithm is to search for a *live* (Definition 12) and *self-consistent* (Definition 11) permutation, from which the values can be easily derived. There is always such a permutation, so an exhaustive search yields Theorem 9, while a non-deterministic guess yields Theorem 8.

We first need to determine, for each vertex, the best (with respect to π) random vertex that Eve can ensure to reach. We do so with the help of a qualitative oracle, which computes embedded almost-sure regions for Eve:

Definition 10. *Let $\mathcal{G} = (\mathcal{A}, \Phi)$ be a normalised finite simple stochastic tail game, and π be a permutation over the k random vertices of $\mathcal{A}$. The vertices of $\mathcal{A}$ are partitioned into the π-regions $(W_\pi[0], \ldots, W_\pi[k+1])$ defined as follows:*

- $W_\pi[k+1] = \{\circledcirc\}$;
- *for any $1 \leq i \leq k$, $W_\pi[i]$ is the almost-sure region of Eve in $\mathcal{A}$ with respect to the objective $\Phi \cup \mathrm{Reach}(\cup_{j \geq i}\{\pi_j\})$, minus $\cup_{j>i} W_\pi[j]$;*

– $W_\pi[0] = \{\otimes\}$.

Notice that a random vertex π_i may belong to a region $W_\pi[j]$ with $i < j$ (but not $i > j$). In this case, the region $W_\pi[i]$ is empty. Once the π-regions have been computed, we derive from them a Markov Chain $\mathcal{G}^\pi$, with $k+2$ vertices numbered $0 \dots k+1$: for any $i, j \in [0, k+1]$, the probability of going from π_i to π_j is equal to the probability of going from π_i to $W_\pi[j]$ in $\mathcal{G}$. We denote by $v_\pi[i]$ the value of π_i in $\mathcal{G}^\pi$.

Self-consistency is then a most natural condition, as it simply expresses the adequation between *a priori* preferences, and resulting values:

Definition 11. *Let $\mathcal{G}$ be a finite simple stochastic tail game with k random vertices. A permutation π over $\mathcal{Q}_R$ is self-consistent if for any $1 \leq i \leq j \leq k$, $v_\pi[i] \leq v_\pi[j]$.*

Liveness expresses the intuitive fact that a random vertex should always have a positive probability to progress to a better region (from Eve's point of view) in one step:

Definition 12. *Let $\mathcal{G}$ be a finite simple stochastic tail game with k random vertices. A permutation π over $\mathcal{Q}_R$ is live if for any $1 \leq i \leq k$, $\delta(\pi_i)(\cup_{j>i} W_\pi[j]) > 0$.*

Our interest in self-consistent and live permutations is explained by the following proposition, which the key for the proof of Theorem 9.

Proposition 13. *Let $\mathcal{G}$ be a finite simple stochastic tail game with k random vertices. Then there exists a self-consistent and live permutation in G. Checking that a permutation is live and self-consistent can be achieved with k calls to a procedure computing almost-sure winning sets of games smaller than G. Given a live and self-consistent permutation, values of G are computable time polynomial in the size of G.*

Proof (of Theorem 9). The algorithm enumerates all possible permutations of random vertices, and check whether they are live and self-consistent. Once such a permutation is found, values are computed in polynomial time. According to Proposition 13, this can be achieved with less than $(k+1)!$ calls to the procedure computing almost-sure winning sets.

An important theoretical by-product of this proof follows from the fact that the π-strategies derived from a live and self-consistent permutation are optimal:

Theorem 14. *In any finite simple stochastic tail game, both players have optimal strategies.*

It can also be noted that Eve's strategy is defined as a spatial composition of residually almost-sure strategies, and does not use more memory than its components:

Theorem 15. *Let $\mathfrak{C}$ be a class of tail conditions. If Eve has almost-sure strategies with finite memory M in $\mathfrak{C}$-games, then she also has optimal strategies with memory M in $\mathfrak{C}$-games.*

Note that Theorem 15 does not hold when the winning condition is not a tail condition. Consider for example the (regular) case of weak-parity games, where Eve wins if the lowest visited vertex is even. Both players have positional almost-sure strategies, but optimal strategies may require memory, as in Figure 4.

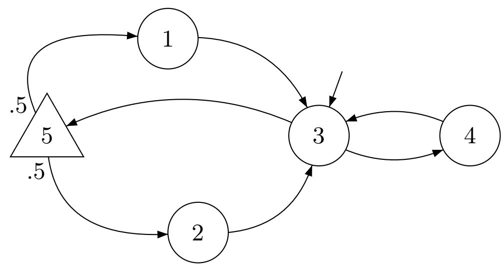


Fig. 4. Optimal strategies require memory in weak parity games

5 New examples of games with computable values

In this section we apply results of the previous section and present an algorithm for computing values of games whose winning condition is a boolean combination of Büchi and mean-payoff conditions.

In verification, Büchi games are popular tools, due to their strong links with temporal logics and their ability to encode accessibility, safety and liveness conditions [GTW02]. In a Büchi game, a subset $B \subseteq \mathcal{Q}$ of the vertices is called the set of Büchi states. The goal of Eve is to visit these vertices infinitely often: $\Phi_{\text{buc}} = \{(q_0, q_1, q_2, \ldots) \in \mathcal{Q}^\omega \mid \exists^\infty n, q_n \in B\}$.

In simple stochastic Büchi games, almost-sure sets can be computed in quadratic time [dA97] and values of Büchi games in exponential time [dAH00].

Mean-payoff games arise from economic modeling and have been extensively studied in classical game theory [MN81]. In a game equipped with the mean-payoff condition Φ_{mean}, each vertex q is labelled with a reward $r(q) \in \mathbb{R}$ and the goal of Eve is to maximize the probability that the average value of rewards is positive:

$$\Phi_{\text{mean}} = \left\{ (q_0, q_1, q_2, \ldots) \in \mathcal{Q}^\omega \mid \limsup_n \frac{1}{n+1}(r(q_0) + r(q_1) + \cdots + r(q_n)) \geq 0 \right\} .$$

Remark that this definition slightly differs from the usual notion of mean-payoff game, where Eve wants to maximize the expected average value of rewards, while in a Φ_{mean}-game Eve wants to maximize the probability that this average value is positive. For that reason, we cannot make use of the classical algorithms for mean-payoff games. However, results of [Gim07,Gim06] imply that players have positional (memoryless) optimal strategies in Φ_{mean}-games, hence a standard strategy enumeration algorithm can be used to compute values of Φ_{mean}-games in exponential time. The almost-sure set is exactly the set of vertices with value 1, according to Theorem 1.

Theorem 16. *Values of games equipped with winning conditions $\Phi_{mean} \cap \Phi_{buc}$ and $\Phi_{mean} \cup \Phi_{buc}$ are computable in exponential time.*

At first, computing values of $\Phi_{\text{mean}} \cap \Phi_{\text{buc}}$-games and $\Phi_{\text{mean}} \cup \Phi_{\text{buc}}$-games seems pretty complicated, because in these games players do not have positional (memoryless) optimal strategies, in contrary to Φ_{buc}-games and Φ_{mean}-games.

However, since $\Phi_{\text{mean}} \cap \Phi_{\text{buc}}$ and $\Phi_{\text{mean}} \cup \Phi_{\text{buc}}$ are tail winning conditions, we can make use of the algorithm described by Theorem 9, and it is enough to design algorithms for computing *almost-sure regions* in $\Phi_{\text{mean}} \cap \Phi_{\text{buc}}$ and $\Phi_{\text{mean}} \cup \Phi_{\text{buc}}$ games, which is quite easy.

For the winning-condition $\Phi_{\text{mean}} \cap \Phi_{\text{buc}}$, computation of the almost-sure region is achieved by Algorithm 1, which computes the largest sub-game where Eve wins both the Φ_{mean} and Φ_{buc} games.

Input: A game $G = (\mathcal{Q}, \mathcal{T})$ and $r : \mathcal{Q} \to \mathbb{R}$ and $B \subseteq \mathcal{Q}$.
Output: The almost-sure region R of Eve in $(G, \Phi_{\text{buc}} \cap \Phi_{\text{mean}})$

$R \leftarrow \mathcal{Q}$;
repeat
  let R' be the almost-sure region in the Φ_{mean}-game induced by R
  let R'' be the almost-sure region in the Φ_{buc}-game induced by R'
  if $R = R''$ **then**
    return R;
  else
    $R \leftarrow R''$;

Algorithm 1: Algorithm for computing almost-sure region for $\Phi_{\text{buc}} \cap \Phi_{\text{mean}}$.

For the winning-condition $\Phi_{\text{mean}} \cup \Phi_{\text{buc}}$, computation of the almost-sure region is achieved by Algorithm 2. This algorithm uses reachability games, whose winning condition is $\Phi_{\text{reach}} = \{(q_0, q_1, q_2, \ldots) \in \mathcal{Q}^\omega \mid \exists n \in \mathbb{N}, q_n \in B\}$ and computes the largest sub-game of G where from every vertex Eve wins either positively the Φ_{reach} game or wins almost-surely the Φ_{mean} game.

Input: A game $G = (\mathcal{Q}, \mathcal{T})$ and $r : \mathcal{Q} \to \mathbb{R}$ and $B \subseteq \mathcal{Q}$.
Output: The almost-sure region R of Eve in $(G, \Phi_{\text{buc}} \cup \Phi_{\text{mean}})$

$R \leftarrow \mathcal{Q}$;
repeat
  let R' be the almost-sure region of Adam in the Φ_{reach}-game induced by R;
  let R'' be the positive region of Adam in the Φ_{mean}-game induced by R';
  if $R'' = \emptyset$ **then**
    return R;
  else
    remove R'' from R;

Algorithm 2: Algorithm for computing almost-sure region for $\Phi_{\text{buc}} \cup \Phi_{\text{mean}}$.

6 Conclusion

We have shown that the computation of values of simple stochastic games with a tail winning condition reduces to the computation of almost-sure winning sets in games with the same winning condition. Moreover, we have proved the qualitative determinacy of stochastic tail games and the existence of optimal strategies in these games. Based on these results, we have described an algorithm for computing values of games whose winning condition is a boolean condition of mean-payoff and Büchi condition.

This work opens two interesting research directions. First, finding classes of tail winning conditions that are of interest for verification and/or economics and whose values are computable. Second, checking whether qualitative determinacy holds for other classes of stochastic games: concurrent games, pushdown games or games on vector addition systems.

References

[BBG08] Christel Baier, Nathalie Bertrand, and Marcus Groesser. On Decision Problems for Probabilistic Büchi Automata. In *Proceedings of the 11th International Conference on the Foundations of Software Science and Computational Structures, FoSSaCS' 08*, volume 4972 of *Lecture Notes in Computer Science*, pages 287–301. Springer-Verlag, 2008.

[BGG09] Nathalie Bertrand, Hugo Gimbert, and Blaise Genest. Qualitative Determinacy and Decidability of Stochastic Games with Signals. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS'09*. IEEE Computer Society, 2009.

[BL69] J. Richard Büchi and Lawrence H. Landweber. Solving Sequential Conditions by Finite-State Strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.

[CdAH05] Krishnendu Chatterjee, Luca de Alfaro, and Thomas A. Henzinger. The Complexity of Stochastic Rabin and Streett Games. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, ICALP'05*, volume 3580 of *Lecture Notes in Computer Science*, pages 878–890. Springer-Verlag, 2005.

[CDHR07] Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Algorithms for Omega-regular Games of Incomplete Information. *Logical Methods in Computer Science*, 3(3), 2007.

[CH06] Krishnendu Chatterjee and Thomas A. Henzinger. Strategy Improvement for Stochastic Rabin and Streett Games. In *Proceedings of the 17th International Conference on Concurrency Theory, CONCUR'06*, volume 4137 of *Lecture Notes in Computer Science*, pages 375–389. Springer-Verlag, 2006.

[Cha07a] Krishnendu Chatterjee. Concurrent Games with Tail Objectives. *Theoretical Computer Science*, 388(1–2):181–198, 2007.

[Cha07b] Krishnendu Chatterjee. Stochastic Müller Games are PSPACE-Complete. In *Proceedings of the 27th International Conference on the Foundations of Software Technology and Theoretical Computer Science, FSTTCS'07*, volume 4855 of *Lecture Notes in Computer Science*, pages 436–448. Springer-Verlag, 2007.

[Cha07c] Krishnendu Chatterjee. *Stochastic ω-regular Games*. PhD thesis, University of California at Berkeley, 2007.

[CHH08] Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Stochastic Finitary Games. Technical Report 2008–002, Laboratoire d'Informatique Algorithmique: Fondements et Applications, CNRS UMR 7089, 2008.

[CJH04] Krishnendu Chatterjee, Marcin Jurdziński, and Thomas A. Henzinger. Quantitative Stochastic Parity Games. In *Proceedings of the 15th Symposium on Discrete Algorithms, SODA'04*, pages 121–130. Society for Industrial and Applied Mathematics, 2004.

[Con92] Anne Condon. The Complexity of Stochastic Games. *Information and Computation*, 96(2):203–224, 1992.

[CY95] Costas Courcoubetis and Mihalis Yannakakis. The Complexity of Probabilistic Verification. *Journal of the ACM*, 42(4):857–907, 1995.

[dA97] Luca de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997.

[dAH00] Luca de Alfaro and Thomas A. Henzinger. Concurrent ω-regular Games. In *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science, LICS'00*, pages 141–154. IEEE Computer Society, 2000.

[Dur96] Richard Durett. *Probability Theory and Examples*. Duxbury Press, 1996.

[GH09] Hugo Gimbert and Florian Horn. Solving Simple Stochastic Games with Few Random Vertices. *Logical Methods in Computer Science*, 5(2):9th, 2009.

[Gim06] Hugo Gimbert. *Jeux Positionels*. PhD thesis, Université Paris VII – Denis Diderot, 2006.

[Gim07] Hugo Gimbert. Pure Stationary Optimal Strategies in Markov Decision Processes. In *Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science, STACS'07*, volume 4393 of *Lecture Notes in Computer Science*, pages 200–211. Springer-Verlag, 2007.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[HK66] Alan J. Hoffman and Richard M. Karp. On Nonterminating Stochastic Games. *Management Science*, 12(5):359–370, 1966.

[Hor08] Florian Horn. *Random Games*. PhD thesis, Université Paris 7 and RWTH Aachen, 2008.

[Mar98] Donald A. Martin. The Determinacy of Blackwell Games. *Journal of Symbolic Logic*, 63(4):1565–1581, 1998.

[MN81] J.F. Mertens and A. Neyman. Stochastic games. *International Journal of Game Theory*, 10:53–56, 1981.

[Paz71] Arturo Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.

[PR89] Amir Pnueli and Roni Rosner. On the Synthesis of a Reactive Module. In *Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages, POPL'89*, pages 179–190, 1989.

[Sha53] Lloyd S. Shapley. Stochastic Games. In *Proceedings of the National Academy of Science of the USA*, volume 39, pages 1095–1100, 1953.

A Proofs for Section 3

Proof (of Proposition 3). For any finite play u such that $v_\sigma(u) \leq \eta$, we define a strategy τ_u such that $v_{\sigma,\tau_u}(u) \leq \eta$. Consider now the strategy θ, defined by:

- if for any prefix u of x, $v_\sigma(u) > \eta$, $\theta(x) = \tau(x)$;
- if u is the shortest prefix of x such that $v(u) \leq \eta$, $\theta(x) = \tau_u(x)$.

It is clear that $\mathbb{P}_q^{\sigma,\tau}(\Lambda_\eta^\sigma) = \mathbb{P}_q^{\sigma,\theta}(\Lambda_\eta^\sigma)$, and that $\mathbb{P}_q^{\sigma,\theta}(\Phi \mid \Lambda_\eta^\sigma) \leq \eta$. As $\mathbb{P}_q^{\sigma,\theta}(\Phi) \geq \nu$, we get:

$$\nu \leq \eta \cdot \mathbb{P}_q^{\sigma,\tau}(\Lambda_\eta^\sigma) + (1 - \mathbb{P}_q^{\sigma,\tau}(\Lambda_\eta^\sigma)) \enspace .$$

Proposition 3 follows.

Proof (of Proposition 4). For any integer n, we define the function φ_n, from Ω to $[0,1]$ by $\varphi_n(\rho) = v_{\sigma,\tau}(\rho_0 \dots \rho_n)$. By Levy's law [Dur96],

$$\mathbb{P}_q^{\sigma,\tau}(\lim_{n\to\infty} \mathbb{E}_q^{\sigma,\tau} \varphi_n = 1_\Phi) = 1 \enspace .$$

Now, if $\rho \not\models \Lambda_\eta^\sigma$, we get,

$$\forall n, \varphi_n(\rho) = v_{\sigma,\tau}(\rho_0 \dots \rho_n) \geq v_\sigma(\rho_0 \dots \rho_n \geq \eta \enspace ,$$

so $\lim_{n\to\infty} \varphi_n(\rho) \neq 0$, $\mathbb{P}_q^{\sigma,\tau}(\Phi \mid \neg\Lambda_\eta^\sigma) = 1$, and Proposition 4 follows.

We define some shorthand notation to simplify the manipulation of reset-related events:

$$\begin{aligned}
\mathcal{R}_\eta^i &= \{\rho \in \Omega^{\sigma\downarrow\eta} \mid \text{ there are } i \text{ resets in } \rho\} \enspace , \\
\mathcal{R}_\eta^\infty &= \bigcap_{i\in\mathbb{N}} \mathcal{R}_\eta^i \enspace .
\end{aligned}$$

We can now prove two intermediate propositions about the behaviour of reset strategies:

Proposition 17. *Let q be a vertex of $\mathcal{Q}$, σ be a strategy for Eve, and τ be a strategy for Adam. We have:*

$$\mathbb{P}_q^{\sigma\downarrow\eta,\tau}(\mathcal{R}_\eta^\infty) = 0 \enspace .$$

Proof. Let $\nu = \min\{v_\sigma(s) \mid s \in \mathcal{Q} \wedge v_\sigma(s) > \eta\}$. The key observation is that:

$$\forall i, \mathbb{P}_q^{\sigma\downarrow\eta,\tau}(\mathcal{R}_\eta^{i+1} \mid \mathcal{R}_\eta^i) \leq \frac{1-\nu}{1-\eta} \enspace . \tag{1}$$

Indeed, after the ith reset, the token is in a vertex whose σ-value is greater than η (and thus greater or equal than ν), and Eve plays σ as if the play just started. Thus, by Proposition 3, the probability that the σ-value of the finite play in memory will ever drop below η is at most $\frac{1-\nu}{1-\eta}$, and (1) follows. This completes the proof of Proposition 17.

Proposition 18. *Let q be a vertex of $\mathcal{Q}$, σ be a strategy for Eve, and τ be a strategy for Adam. We have:*

$$\mathbb{P}_q^{\sigma\downarrow\eta,\tau}(\Phi \mid \exists i, \forall j \geq i, v_\sigma(\rho_j) > \eta) = 1 \enspace .$$

Proof. By Proposition 17, $\mathbb{P}_q^{\sigma_{\downarrow\eta},\tau}(\mathcal{R}_\eta^\infty) = 0$, so we can consider only the plays with a finite number of resets. Let us consider the "final" memory after the play: it is a play consistent with σ which does not verify Λ_η^σ. By Proposition 4 it is winning for Eve with probability one, and Proposition 18 follows from the fact that Φ is a tail condition.

Proof (of Theorem 1). Let us show first how to compute an almost-sure strategy for Eve from the vertices with value one. As $\mathcal{Q}$ is finite, we can choose a real number η such that $\forall q, \mathbf{v}(q) < 1 \rightarrow \mathbf{v}(q) < \eta$ and a strategy σ such that $\forall q, \mathbf{v}(q) = 1 \rightarrow v_\sigma(q) > \eta$. The proof consists then in showing that $\sigma_{\downarrow\eta}$ is almost-sure from any vertex with value one. Neither Adam nor Random can leave the region with value one, and by Definition 5, Eve does not: she could leave only if the value of the prefix was below η, and she would sooner reset her memory. So, for any play ρ starting in the region with value one and consistent with $\sigma_{\downarrow\eta}$, $\forall i, v_\sigma(\rho_i) > \eta$, and by Proposition 18, $\mathbb{P}_q^{\sigma_{\downarrow\eta},\tau}(\Phi) = 1$. An almost-sure strategy for Adam from the vertices with value zero can be built in the same way. This completes the proof of Theorem 1

Proof (of Theorem 6). By Theorem 1, each vertex in the positive region of Eve has a positive value. As $\mathcal{Q}$ is finite, we can choose a real number ν such that $\forall q \in \mathcal{Q}, \nu < \mathbf{v}(q)$, and a strategy σ such that $\forall q \in \mathcal{Q}, v_\sigma(q) > \nu$. Let η be a real number such that $\eta < \nu$. For any play ρ of $\mathcal{A}$, $\forall i, v_\sigma(\rho_i) > \eta$, so Proposition 18 yields the almost sureness of $\sigma_{\downarrow\eta}$. The second equation follows by duality, and Theorem 6 follows.

B Proofs for Section 4

For a given permutation π, we define the π-strategies for both players:

Eve's π-strategy σ_π is a spatial combination of almost-sure strategies: in $W_\pi[i]$, she plays an almost-sure strategy with respect to the objective $\mathcal{W} \vee \text{Reach}(\cup_{j\geq i}\{\pi_j\})$.

Adam's π-strategy τ_π is a spatial combination of reset strategies: in $W_\pi[i]$, he plays a bounded strategy of value η with respect to the objective $\mathcal{W} \vee \text{Reach}(\cup_{j\geq i}\{\pi_j\}))$, which is reset when the value of the prefix drops below $\frac{\eta}{2}$. By Proposition 18, if any region is visited infinitely often, Adam wins with probability one, and Proposition 19 follows:

Proposition 19. *Let π be a permutation, and τ_π be the corresponding π-strategy for Adam. For any initial vertex q and strategy σ of Eve, we have:*

$$\mathbb{P}_q^{\sigma,\tau_\pi}(\neg\Phi \vee \text{Reach} \circledcirc) = 1$$

Proposition 20. *Let π be a live permutation, and σ_π be the corresponding π-strategy for Eve. For any strategy τ of Adam, we have:*

$$\mathbb{P}_q^{\sigma_\pi,\tau}(\Phi \vee \text{Reach} \otimes) = 1$$

Proof. Let q be a vertex of $\mathcal{Q}$, τ be a strategy for Adam, and $\texttt{Stuck}(i)$ be the event "$\text{Inf}(\rho) \cap W_\pi[i] \neq \emptyset \wedge \text{Inf}(\rho) \cap \{\pi_i, \ldots, \pi_k\} = \emptyset$". By definition of σ_π, for any $1 \leq i \leq k$, we have $\mathbb{P}_q^{\sigma_\pi,\tau}(\texttt{Stuck}(i) \wedge \neg\Phi) = 0$. By the liveness property, for any $1 \leq i \leq k$, we have $\mathbb{P}_q^{\sigma_\pi,\tau}(\pi_i \in \text{Inf}(\rho) \wedge \text{Inf}(\rho) \cap \cup_{j>i} W_\pi[j] = \emptyset) = 0$. Proposition 20 follows.

Proposition 21. *There is a live permutation consistent with the values of $\mathcal{G}$.*

Proof. The permutation is chosen starting from π_k, and going down to π_1. At each step, the vertex π_i is chosen among the ones such that:

- $\mathbf{v}(\pi_i) = \max\{\mathbf{v}(q) \mid q \in \mathcal{Q}_R \setminus \{\pi_{i+1}, \ldots, \pi_k\}\}$
- $\delta(\pi_i)(\cup_{j>i} W_\pi[i]) > 0$

There is always such a vertex: otherwise, the set X of vertices whose value is maximal in $\mathcal{Q} \setminus \cup_{j>i} W_\pi[j]$ would be a trap for Adam, and the vertices of X have value 1, in contradiction with the "normalised" hypothesis.

Proposition 22. *Let π be a self-consistent permutation, and i and j be two integers such that $i < j$ and $\pi_i \in W_\pi[j]$. Then for all ℓ such that $\delta(\pi_i)(W_\pi[\ell]) > 0$, $\mathfrak{v}_\pi[i] = \mathfrak{v}_\pi[j] = \mathfrak{v}_\pi[\ell]$.*

Proof. As $\pi_i \in W_\pi[j]$, $\delta(\pi_i)(W_\pi[\ell]) > 0 \Rightarrow \ell \geq j$. By self-consistency, $\ell \geq j \Rightarrow \mathfrak{v}_\pi[\ell] \geq \mathfrak{v}_\pi[j]$, so $\mathfrak{v}_\pi[i] \geq \mathfrak{v}_\pi[j]$. But, again by self-consistency, $\mathfrak{v}_\pi[i] \leq \mathfrak{v}_\pi[j]$. So $\mathfrak{v}_\pi[i] = \mathfrak{v}_\pi[j]$, and, $\delta(\pi_i)(W_\pi[\ell]) > 0 \Rightarrow \mathfrak{v}_\pi[i] = \mathfrak{v}_\pi[\ell]$. Proposition 22 follows.

Lemma 23. *There is a live and self-consistent permutation.*

Proof. The first part of this proof was to show that there is a live permutation π consistent with the values of the game (Proposition 21). The point is now to prove that the π-values *are* the values of $\mathcal{G}$. These values are constant over the π-regions:

$$q \in \mathrm{Win}_E^{\Phi \vee \mathrm{Reach}\, \mathtt{X},1}(\mathcal{A}) \Rightarrow \mathbf{v}(q) \geq \min\{\mathbf{v}(q) \mid q \in X\}$$

$$q \notin \mathrm{Win}_E^{\Phi \vee \mathrm{Reach}\, \mathtt{X},1}(\mathcal{A}) \Rightarrow \mathbf{v}(q) \leq \max\{\mathbf{v}(r) \mid r \in \mathcal{Q}_R \setminus X\}$$

Thus, the relations between the values of the π-regions which follow from (**??**) are exactly the relations between the values of the vertices in $\mathfrak{G}^\pi$. So $\mathbf{v} = \mathfrak{v}_\pi$, and Lemma 23 follows.

Lemma 24. *If π is a live and self-consistent permutation, then the π-strategies are optimal and $\mathfrak{v}_\pi = \mathbf{v}$.*

Proof. We fix an initial vertex q and prove independently that $v_{\sigma_\pi}(q) \geq \mathfrak{v}_\pi(q)$ and $v_{\tau_\pi}(q) \leq \mathfrak{v}_\pi(q)$. Let τ be a strategy for Adam. We define an "expected π-value" function f by $f(n) = \sum_{s \in \mathcal{Q}} \mathfrak{v}_\pi(s) \cdot \mathbb{P}_q^{\sigma_\pi,\tau}(\rho_n = s)$. This function is waxing:

- a move of Eve consistent with σ_π remains in the same π-region;
- a move of Adam sends the token to a vertex with greater or equal π-value (self-consistency);
- the value of a random vertex π_i such that $\pi_i \in W_\pi[i]$ is the average value of its successors;
- a random vertex π_i such that $\pi_i \in W_\pi[j]$ and $i < j$ sends the token to a vertex with equal π-value (Proposition 22).

Thus, $f(n) \leq f(n+1)$. Furthermore, as $f(n) \leq 1 - \mathbb{P}_q^{\sigma_\pi,\tau}(\rho_n = \otimes)$, we get $\lim f(n) \leq 1 - \mathbb{P}_q^{\sigma_\pi,\tau}(\mathrm{Reach}\,\otimes)$. By Proposition 20, $\mathbb{P}_q^{\sigma,\tau_\pi}(\mathrm{Reach}\,\otimes) = 1 - v_{\sigma_\pi,\tau}(q)$, so $\mathfrak{v}_\pi(q) = f(0) \leq \lim_{n\to\infty} f(n) \leq v_{\sigma_\pi,\tau}(q)$. As τ is an arbitrary strategy for Adam, we get $v_{\sigma_\pi} \geq \mathfrak{v}_\pi$.

Likewise, for a strategy σ for Eve, we define the function g by $g(n) = \sum_{s \in \mathcal{Q}} \mathfrak{v}_\pi(s) \cdot \mathbb{P}_q^{\sigma,\tau_\pi}(\rho_n = s)$. This function is waning:

- a move of Eve sends the token to a vertex with lower or equal π-value (self-consistency);
- a move of Adam consistent with τ_π remains in the same π-region;

– the value of a random vertex π_i such that $\pi_i \in W_\pi[i]$ is the average value of its successors;
– a random vertex π_i such that $\pi_i \in W_\pi[j]$ and $i < j$ sends the token to a vertex with equal π-value (Proposition 22).

Thus, $g(n) \geq g(n+1)$. Furthermore, as $g(n) \geq \mathbb{P}_q^{\sigma,\tau_\pi}(\rho_n = \circledcirc)$, we get $\lim g(n) \geq \mathbb{P}_q^{\sigma,\tau_\pi}(\text{Reach}\circledcirc)$. By Proposition 19, $\mathbb{P}_q^{\sigma,\tau_\pi}(\text{Reach}\circledcirc) = v_{\sigma,\tau_\pi}(q)$, so $\mathfrak{v}_\pi(q) = g(0) \geq \lim_{n\to\infty} g(n) \geq v_{\sigma,\tau_\pi}(q)$. As σ is an arbitrary strategy for Eve, we get $v_{\tau_\pi} \leq \mathfrak{v}_\pi$.

It follows that $v_{\sigma_\pi} = v_{\tau_\pi} = \mathfrak{v}_\pi$, so σ_π and τ_π are optimal strategies, and $\mathfrak{v}_\pi = \mathbf{v}$. This concludes the proof of Lemma 24.

C Proofs for Section 5

Proof (of Theorem 16). According to Theorem 9, it is enough to provide exponential time algorithms for computing almost-sure sets of $\Phi_{\text{mean}} \cup \Phi_{\text{buc}}$-games and $\Phi_{\text{mean}} \cap \Phi_{\text{buc}}$-games. This is a consequence of the following propositions.

Proposition 25. *Algorithm 1 terminates and is correct.*

Proof. First we have to explain what is the "Φ_{mean}-game induced by the subset" $R \subseteq \mathcal{Q}$, since this term is used in line 3 of Algorithm 1. This notion defined only if R is a *trap for Adam*, in the following sense.

Definition 26. *Let $G = (\mathcal{Q}, \mathcal{T})$ be a game. A subset $R \subseteq \mathcal{Q}$ is a trap for Adam if every random vertex $q \in R \cap \mathcal{Q}_R$ and every Adam vertices $q \in R \cap \mathcal{Q}_A$ have all their successors in R and every Eve vertex $q \in R \cap \mathcal{Q}_E$ has at least one successor in R. In this case the game on the arena $(R, \mathcal{T} \cap R \times R)$ is called the game induced by R and denoted $G[R]$.*

Since Φ_{mean} and Φ_{buc} are tail winning conditions, the almost-sure winning sets of Eve in Φ_{mean}-games and Φ_{buc}-games are traps for Adam, hence the following invariant of Algorithm 1:

(A) The sets R, R' and R'' are traps for Adam.

Thus it makes sense to speak about the game arena $G[R]$.

To prove that Algorithm 1 is correct, we prove that the almost-sure winning set of Eve for the $\Phi_{\text{mean}} \cap \Phi_{\text{buc}}$-game is the largest set $R \subseteq \mathcal{Q}$ such that R is a trap for Adam and Eve wins almost-surely both the Φ_{mean} and the Φ_{buc} games in $G[R]$. This property is stable by union hence the existence of such a largest set R.

We prove that R is contained in the almost-sure region for Eve in the $\Phi_{\text{mean}} \cup \Phi_{\text{buc}}$-game. According to [Gim07], in $G[R]$ Eve has two positional optimal strategies $\sigma_B : R \to R$ and $\sigma_M : R \to R$ that are almost-surely winning for the Büchi and mean-payoff games respectively. To win almost-surely the $\Phi_{\text{buc}} \cap \Phi_{\text{mean}}$-game, Eve can alternate between σ_B and σ_M. Since R is finite, there exists a probability $p > 0$ that using σ_B Eve will reach a Büchi vertex with probability more than p in less than $|R|$ steps, whatever be the initial vertex $q \in R$. Then it is straightforward to check that the following strategy is almost-surely winning for the $\Phi_{\text{buc}} \cap \Phi_{\text{mean}}$-game: play σ_M for 1 step, play σ_B for $|R|$ steps, play σ_M for 2 step, play σ_B for $|R|$ steps, play σ_M for 3 step, play σ_B for $|R|$ steps and so on... Actually this strategy both guarantees to visit Büchi vertices infinitely often and that the expected mean value of rewards is positive, as guaranteed by σ_M.

We prove that the almost-sure region for Eve in the $\Phi_{\text{mean}} \cup \Phi_{\text{buc}}$-game is contained in R. This property actually holds not only for R but for R' and R'', at every step of Algorithm 1. This is clear because if Eve can win the $\mathit{\Phi}_{\text{mean}} \cap \mathit{\Phi}_{\text{buc}}$-game almost-surelythen a fortiori she should win both the $\mathit{\Phi}_{\text{mean}}$ and the $\mathit{\Phi}_{\text{buc}}$ games almost-surely.

Proposition 27. *Algorithm 2 terminates and is correct.*

Proof. It is easy to establish that:

(A) The set R' is a trap for Eve in $G[R]$ and the set $R - R''$ is a trap for Adam,

thus it makes sense to speak about the games induced by R and R'.

To prove that Algorithm 2 is correct, we prove that the almost-sure winning set of Eve for the $\mathit{\Phi}_{\text{mean}} \cup \mathit{\Phi}_{\text{buc}}$-game is the largest set $R \subseteq \mathcal{Q}$ such that R is a trap for Adam and, denoting R' the almost-sure region for Adam for the $\mathit{\Phi}_{\text{reach}}$-game in $G[R]$, Eve wins almost-surely the $\mathit{\Phi}_{\text{mean}}$-game in $G[R']$. This property is stable by union hence the existence of such a largest set R.

We prove that R is contained in the almost-sure region for Eve in the $\mathit{\Phi}_{\text{mean}} \cup \mathit{\Phi}_{\text{buc}}$-game. Again we construct an almost-sure strategy which consists in switching between two positional strategies. When the play is in R', Eve plays her positional strategy almost-sure for the $\mathit{\Phi}_{\text{mean}}$-game. When the play is outside R', Eve plays for $|R|$ steps her positional strategy for attracting the play in a Büchi vertex with positive probability. Under the conditional hypothesis that the play stays ultimately trapped in R' then Eve wins the $\mathit{\Phi}_{\text{mean}}$-game with probability 1. Under the conditional hypothesis that the play reaches $R - R'$ infinitely often then with probability 1 the play visits Büchi vertices infinitely often, hence Eve wins the $\mathit{\Phi}_{\text{buc}}$-game with probability 1. This proves that Eve is almost-surely winning the $\mathit{\Phi}_{\text{mean}} \cup \mathit{\Phi}_{\text{buc}}$-game on R.

We prove that the almost-sure region for Eve in the $\mathit{\Phi}_{\text{mean}} \cup \mathit{\Phi}_{\text{buc}}$-game is contained in R. This is because every set R'' which is removed from R is such that R'' is a trap for Eve in $G[R]$ and moreover Adam wins the $\mathit{\Phi}_{\text{mean}} \cup \mathit{\Phi}_{\text{buc}}$-game with positive probability in $G[R'']$. Hence at every step of the program, $\mathcal{Q} - R$ is contained in the positive region for Adam in the $\mathit{\Phi}_{\text{mean}} \cup \mathit{\Phi}_{\text{buc}}$-game and the almost-sure region for Eve in the $\mathit{\Phi}_{\text{mean}} \cup \mathit{\Phi}_{\text{buc}}$-game is contained in R.