Perfect-Information Stochastic Games with Generalized Mean-Payoff Objectives[*]

Krishnendu Chatterjee[†] Laurent Doyen[§]
[†] IST Austria [§] CNRS & LSV, ENS Cachan

Abstract

Graph games provide the foundation for modeling and synthesizing reactive processes. In the synthesis of stochastic reactive processes, the traditional model is perfect-information stochastic games, where some transitions of the game graph are controlled by two adversarial players, and the other transitions are executed probabilistically. We consider such games where the objective is the conjunction of several quantitative objectives (specified as mean-payoff conditions), which we refer to as generalized mean-payoff objectives. The basic decision problem asks for the existence of a finite-memory strategy for a player that ensures the generalized mean-payoff objective be satisfied with a desired probability against all strategies of the opponent. A special case of the decision problem is the almost-sure problem where the desired probability is 1. Previous results presented a semi-decision procedure for ε-approximations of the almost-sure problem. In this work, we show that both the almost-sure problem as well as the general basic decision problem are coNP-complete, significantly improving the previous results. Moreover, we show that in the case of 1-player stochastic games, randomized memoryless strategies are sufficient and the problem can be solved in polynomial time. In contrast, in two-player stochastic games, we show that even with randomized strategies exponential memory is required in general, and present a matching exponential upper bound. We also study the basic decision problem with infinite-memory strategies and present computational complexity results for the problem. Our results are relevant in the synthesis of stochastic reactive systems with multiple quantitative requirements.

1 Introduction

Reactive systems are non-terminating processes that interact continually with a changing environment. Since such systems are non-terminating, their behavior is described by infinite sequences of events. The classical framework to

[*]This research was partially supported by Austrian Science Fund (FWF) NFN Grant No S11407-N23 (RiSE/SHiNE), ERC Start grant (279307: Graph Games), Vienna Science and Technology Fund (WWTF) through project ICT15-003, and European project Cassting (FP7-601148).

model reactive systems with controllable and uncontrollable events are games on graphs. In the presence of uncertainties, we have stochastic reactive systems with probability distributions over state changes. The performance requirement on such systems, such as power consumption or latency, can be represented by rewards (or costs) associated to the events of the system, and a quantitative objective that aggregates the rewards of an execution to a single value. In several modeling domains, however, there is not a single objective to be optimized, but multiple, potentially dependent and conflicting goals. For example, in the design of an embedded system, the goal may be to maximize average performance while minimizing average power consumption. Similarly, in an inventory management system, the goal would be to optimize the costs associated to maintaining each kind of product [31, 1]. Thus it is relevant to study stochastic games with multiple quantitative objectives.

Perfect-information stochastic games. A perfect-information stochastic graph game [26], also known as turn-based stochastic game or *$2\frac{1}{2}$-player graph game*, consists of a finite directed graph with three kinds of states (or vertices): player-Max, player-Min, and probabilistic states. The game starts at an initial state, and is played as follows: at player-Max states, player Max chooses a successor state; at player-Min states, player Min (the adversary of player Max) does likewise; and at probabilistic states, a successor state is chosen according to a fixed probability distribution. Thus the result of playing the game forever is an infinite path through the graph. If there are no probabilistic states, we refer to the game as a *2-player graph game*; if there are no player-Min states, we refer to the ($1\frac{1}{2}$-player) game as a Markov decision process (MDP); if there are no probabilistic states and no player-Min states, then the (1-player) game is a standard graph.

The class of 2-player graph games has been used for a long time to synthesize non-stochastic reactive systems [10, 43, 46]: a reactive system and its environment represent the two players, whose states and transitions are specified by the vertices and edges of a game graph. Similarly, MDPs have been used to model stochastic processes without adversary [31, 44]. Consequently, $2\frac{1}{2}$-player graph games, which subsume both 2-player graph games and MDPs, provide the theoretical foundation to model stochastic reactive systems [31, 45].

Mean-payoff objectives. One of the most classical example of quantitative objectives is the mean-payoff objective [31, 44, 33, 29], where a reward is associated to each state and the payoff of a path is the long-run average of the rewards of the path (computed as either $\liminf$ or $\limsup$ of the averages of the finite prefixes to ensure the payoff value always exists). While traditionally the verification and the synthesis problems were considered with Boolean objectives [43, 46, 40], recently quantitative objectives have received a lot of attention [6, 11, 7], as they specify requirements on resource consumption (such as for embedded systems or power-limited systems) as well as performance-related properties.

Various semantics for multiple quantitative objectives. The two classical semantics for quantitative objectives are as follows [8]: the first is the expectation semantics, which is a probabilistic average of the quantitative objective over the executions of the system; and the second is the satisfaction semantics, which consider the probability of the set of executions where the quantitative objective is at least a required threshold value ν. The expectation objective is relevant in situations where we are interested in the "average" behaviour of many instances of a given system, while the satisfaction objective is useful for analyzing and optimizing the desired executions, and is more relevant for the design of critical stochastic reactive systems (see [8] for a more detailed discussion). For example, consider one mean-payoff objective that specifies the set of executions where the average power consumption is at most 5 units, and another mean-payoff objective that specifies the set of executions where the average latency is at most 10 units. A multiple objective asks to *satisfy* both, i.e., their conjunction. We refer to such objectives (i.e., conjunction of multiple mean-payoff objectives) as *generalized mean-payoff objectives*[1]. The goal of player Max is to maximize the probability of satisfaction of the generalized mean-payoff objective while player Min tries to minimize this probability, i.e., the game is zero-sum. Concrete applications of $2\frac{1}{2}$-player graph games with generalized mean-payoff objectives have been considered, such as best-effort synthesis where the goal is to minimize the violation of several incompatible specifications [12], real-time scheduling algorithms with requirements on the utility and energy consumption [21], and electric power distribution in an avionics application [4]. In particular, for the real-world avionics application in [4], both two adversarial players, stochastic transitions, as well as multiple mean-payoff objectives are required, i.e., the application can be modeled as $2\frac{1}{2}$-player graph games with generalized mean-payoff objectives, but not in a strict subclass.

Computational questions. In this work, we consider $2\frac{1}{2}$-player graph games with generalized mean-payoff objectives in the satisfaction semantics. A strategy for a player is a recipe that given the history of interaction so far (i.e., the sequence of states) prescribes the next move. The basic decision problem asks, given a $2\frac{1}{2}$-player graph game, a generalized mean-payoff objective, and a probability threshold α, whether there exists a strategy for player Max to ensure the objective be satisfied with probability at least α against all strategies of player Min. Since strategies in games correspond to implementations of controllers for reactive systems, a particularly relevant question is to ask for the existence of a finite-memory strategy in the basic decision problem, instead of an arbitrary strategy. Moreover, an important special case of the basic decision problem is the almost-sure problem, where the probability threshold α is equal to 1.

Previous results. We summarize the main previous results for MDPs, 2-player graph games, and $2\frac{1}{2}$-player graph games, with generalized mean-payoff objectives.

1. *MDPs.* The basic decision problem for generalized mean-payoff objectives in MDPs with infinite-memory strategies can be solved in polynomial time [8]. The problem under finite-memory strategies has not been

[1] In the verification literature, conjunction of reachability, Büchi, and parity objectives, are referred to as generalized reachability, generalized Büchi, and generalized parity objectives, respectively, and generalized mean-payoff objectives naming is for consistency.

addressed yet.

2. *2-player games.* The following results are known [48]: the basic decision problem for generalized mean-payoff objectives in 2-player graph games, both under finite-memory and infinite-memory strategies, is coNP-complete; moreover, for infinite-memory strategies if the mean-payoff objective is defined as the limit supremum of the averages (rather than limit infimum of the average), then the problem is in NP $\cap$ coNP.
3. *$2\frac{1}{2}$-player games.* The almost-sure problem for generalized mean-payoff objectives in $2\frac{1}{2}$-player graph games under finite-memory strategies was considered in [4], and a semi-algorithm (or semi-decision procedure) was presented for approximations of the problem.
4. *Memory of strategies.* Infinite-memory strategies are strictly more powerful than finite-memory strategies, even in 1-player graph games thus also in MDPs and 2-player graph games: there are games where an infinite-memory strategy can ensure the objective with probability 1 while all finite-memory strategies fail to do so[2] [48].

Our contributions. The previous results suggest that $2\frac{1}{2}$-player graph games with generalized mean-payoff objectives are considerably more complicated than 2-player graph games as well as MDPs, as even the decidability of the almost-sure problem was open for $2\frac{1}{2}$-player graph games for finite-memory strategies (the previous result neither gives an exact algorithm, nor establishes decidability for approximation). In this work we present a complete picture of decidability as well as computational complexity. Our results are as follows:

1. *MDPs.* First we study the generalized mean-payoff problem under finite-memory strategies in MDPs. We present a polynomial-time algorithm, and show that with randomization, memoryless strategies (which do not depend on histories but only on the current state) are sufficient, i.e., for finite-memory optimal strategies no memory is required.
2. *$2\frac{1}{2}$-player games.* For $2\frac{1}{2}$-player graph games with generalized mean-payoff objectives we show that: (1) the basic decision problem is coNP-complete under finite-memory strategies (significantly improving the known semi-decidability result for approximation of the almost-sure problem [4]), and moreover, the same complexity holds for the almost-sure problem; and (2) under infinite-memory strategies, the computational complexity results coincide with the special case of 2-player graph games.
3. *Memory of strategies.* Under finite-memory strategies, in contrast to MDPs where we show with randomization no memory is required, we establish an exponential lower bound (even with randomization) for memory required in $2\frac{1}{2}$-player graph games with generalized mean-payoff objectives. We also present a matching upper bound showing that exponential memory is sufficient.

[2] However, in some variants of the decision problem (such as requiring the mean-payoff value, computed as the $\liminf$ of the averages of the finite prefixes, be strictly greater than a threshold ν) finite-memory strategies are as powerful as infinite-memory strategies in MDPs [25].

Key technical insights. We show that for generalized mean-payoff objectives, for the adversary, pure and memoryless strategies are sufficient. Under finite-memory strategies for player Max, this result is established using the following ideas:

- In general for prefix-independent objectives (objectives that do not change if finite prefixes are added or removed from a path), we show that sub-game perfect strategies exist, where a strategy is sub-game perfect if it is optimal after every finite history. Such a result is known for infinite-memory strategies using results from martingale theory [35]. Our proof for finite-memory strategies is conceptually simpler, and uses combinatorial arguments and well-known discrete properties of MDPs (see Lemma 2, Section 3).
- Then using the above result we show that for a sub-class of prefix-independent objectives (that subsume generalized mean-payoff objectives) for the adversary pure memoryless strategies suffice (see Theorem 1, Section 3). Moreover, for this class of objectives we establish determinacy when each player is restricted to finite-memory strategies, which is of independent interest (see also Theorem 1); and also show that such determinacy result does not hold for all prefix-independent objectives (see Remark 3).
- For MDPs, we generalize a result of [39] from graphs to MDPs, to obtain a linear-programming solution for the generalized mean-payoff objectives under finite-memory strategies (see Theorem 3, Section 4).

Combining these results we obtain the coNP upper bound for the basic decision problem for $2\frac{1}{2}$-player graph games and the coNP lower bound follows from existing results on 2-player graph games (see Theorem 5, Section 4).

Related works. We have described the most relevant related works in the paragraph *Previous results.* We discuss other relevant related works. Markov decision processes with multiple objectives have been studied in numerous works, for various quantitative objectives, such as mean-payoff [14, 8], discounted sum [20, 17], total reward [32] as well as qualitative objectives [30], and their combinations [19, 2, 3, 25]. The problem of 2-player graph games with multiple quantitative objectives has also been widely studied both for finite-memory strategies [48, 22, 37, 9, 47] as well as infinite-memory strategies [48, 23]. In contrast, for $2\frac{1}{2}$-player games with multiple quantitative objectives only few results are known [4, 24], because of the inherent difficulty to handle two-players, probabilistic transitions, as well as multiple objectives all at the same time. A semi-decision procedure for approximation of the almost-sure problem for $2\frac{1}{2}$-player games with generalized mean-payoff objectives was presented in [4], which we significantly improve. The class of $2\frac{1}{2}$-player graph games with positive Boolean combinations of total-reward objectives was considered in [24], and the problem was established to be PSPACE-hard and undecidable for pure strategies.

2 Definitions

Probability distributions. For a finite set S, we denote by $\Delta(S)$ the set of all probability distributions over S, i.e., the set of functions $p : S \to [0,1]$ such that $\sum_{s\in S} p(s) = 1$. The *support* of p is the set $\mathsf{Supp}(p) = \{s \in S \mid p(s) > 0\}$. For a set $U \subseteq S$ let $p(U) = \sum_{s\in U} p(s)$.

Perfect-information stochastic games. A *perfect-information stochastic game* (for brevity, stochastic games in the sequel) is a tuple $\mathcal{G} = \langle S, (S_{\mathsf{Max}}, S_{\mathsf{Min}}), A, \delta\rangle$, consisting of a finite set $S = S_{\mathsf{Max}} \uplus S_{\mathsf{Min}}$ of states partitioned into the set S_{Max} of states controlled by player Max (depicted as round states in figures) and the set S_{Min} of states controlled by player Min (depicted as square states in figures), a finite set A of actions, and a probabilistic transition function $\delta : S \times A \to \Delta(S)$. If $\delta(s,a)(s') > 0$, we say that s' is an *a-successor* of s. A transition $\delta(s,a)$ is *deterministic* if $\delta(s,a)(s') = 1$ for some state s'. The underlying graph of $\mathcal{G}$ is (S,E) where $E = \{(s,s') \mid \delta(s,a)(s') > 0 \text{ for some } a \in A\}$.

For complexity results, we consider that the probabilities in stochastic games are rational numbers with numerator and denominator encoded in binary.

Markov decision processes and end-components. A *Markov decision process* (MDP) is the special case of a stochastic game where either $S_{\mathsf{Max}} = \varnothing$, or $S_{\mathsf{Min}} = \varnothing$. Given a state $s \in S$ and a set $U \subseteq S$, let $A_U(s)$ be the set of all actions $a \in A$ such that $\mathsf{Supp}(\delta(s,a)) \subseteq U$. A *closed* set in an MDP is a set $U \subseteq S$ such that $A_U(s) \neq \varnothing$ for all $s \in U$. A set $U \subseteq S$ is an *end-component* [27] if (i) U is closed, and (ii) the graph (U, E_U) is strongly connected where $E_U = \{(s,t) \in U \times U \mid \delta(s,a)(t) > 0 \text{ for some } a \in A_U(s)\}$ denote the set of edges given the actions. We denote by $\mathcal{E}(M)$ the set of all end-components of an MDP M.

Markov chains and recurrent sets. A *Markov chain* is the special case of an MDP where the action set A is a singleton. In Markov chains, end-components are called *closed recurrent sets.*

Plays and strategies. A *play* is an infinite sequence $s_0 s_1 \ldots \in S^\omega$ of states. A *randomized strategy* for Max is a recipe to describe what is the next action to play after a prefix of a play ending in a state controlled by player Max; formally, it is a function $\sigma : S^* S_{\mathsf{Max}} \to \Delta(A)$ that provides probability distributions over the action set. A *pure strategy* is a function $\sigma : S^* S_{\mathsf{Max}} \to A$ that provides a single action, which can be seen as a special case of randomized strategy where for every play prefix $\rho \in S^* S_{\mathsf{Max}}$ there exists an action $a \in A$ such that $\sigma(\rho)(a) = 1$.

We consider the following memory restrictions on strategies. A strategy σ is *memoryless* if it is independent of the past and depends only on the current state, that is $\sigma(\rho) = \sigma(\mathsf{Last}(\rho))$ for all play prefixes $\rho \in S^* S_{\mathsf{Max}}$, where $\mathsf{Last}(s_0 \ldots s_k) = s_k$. In the sequel, we call memoryless strategies the pure memoryless strategies, and we emphasize that strategies $\sigma : S_{\mathsf{Max}} \to \Delta(A)$ are not necessarily pure by calling them randomized memoryless.

A strategy σ uses *finite memory* if it can be described by a transducer

$\langle M, m_0, \sigma_u, \sigma_n\rangle$ consisting of a finite set M (the memory set), an initial memory value $m_0 \in M$, an update function $\sigma_u : M \times S \to M$ for the memory, and a next-action function $\sigma_n : M \to \Delta(A)$; the transducer $\langle M, m_0, \sigma_u, \sigma_n\rangle$ defines the strategy σ such that $\sigma(\rho) = \sigma_n(\hat{\sigma}_u(m_0, \rho))$ for all play prefixes $\rho \in S^* S_{\mathsf{Max}}$ where $\hat{\sigma}_u$ extends σ_u to sequences of states as usual (i.e., $\hat{\sigma}_u(m, \rho{\cdot}s) = \sigma_u(\hat{\sigma}_u(m, \rho), s)$). Given a finite-memory strategy σ for player Max, let $\mathcal{G}_\sigma = \langle S', (\varnothing, S'_{\mathsf{Min}}), A, \delta'\rangle$ be the MDP obtained by playing σ in $\mathcal{G}$, where $S' = S'_{\mathsf{Min}} = S \times M$ and the transition function δ' is defined for all $\langle s, m\rangle \in S'$ and action $a \in A$ of player Min as follows, for all $s' \in S$, where $m' = \sigma_u(m, s)$:

- if $s \in S_{\mathsf{Max}}$, then $\delta'(\langle s, m\rangle, a)(\langle s', m'\rangle) = \sum_{b \in A} \sigma_n(m')(b) \cdot \delta(s, b)(s')$;
- if $s \in S_{\mathsf{Min}}$, then $\delta'(\langle s, m\rangle, a)(\langle s', m'\rangle) = \delta(s, a)(s')$.

Strategies π for player Min are defined analogously, as well as the memory restrictions. A strategy that is not finite-memory is referred to as an infinite-memory strategy. We denote by Σ the set of all strategies for player Max, and by Σ^{PM}, and Σ^{FM} respectively the set of all pure memoryless, and all finite-memory strategies for player Max. We use analogous notation Π, Π^{PM}, and Π^{FM} for player Min.

Objectives. An *objective* is a Borel-measurable set of plays [5]. In this work we consider conjunctions of mean-payoff objectives. Some of our results are related to more general classes of prefix-independent and shuffle-closed objectives. We define the relevant objectives below:

1. *Prefix-independent objectives.* An objective $\Omega \subseteq S^\omega$ is *prefix-independent* if for all plays $\rho \in S^\omega$, and all states $s \in S$, we have $\rho \in \Omega$ if and only if $s \cdot \rho \in \Omega$, that is the objective is independent of the finite prefixes (of arbitrary length) of the plays.
2. *Shuffle-closed objectives.* A *shuffle* of two plays ρ_1, ρ_2 is a play $\rho = u_1u_2u_3\dots$ such that $u_i \in S^*$ for all $i \geq 1$, and $\rho_1 = u_1u_3u_5\dots$ and $\rho_2 = u_2u_4u_6\dots$. An objective $\Omega \in S^\omega$ is closed under shuffling, if all shuffles of all plays $\rho_1, \rho_2 \in \Omega$ belong to Ω.
3. *Multi-mean-payoff objectives.* Let $\mathsf{rwd} : S \to \mathbb{Q}^k$ be a *reward function*[3] that assigns a k-dimensional vector of weights to each state. For $1 \leq j \leq k$, we denote by $\mathsf{rwd}_j : S \to \mathbb{Q}$ the projection of the function rwd on the j-th dimension. The conjunction of *mean-payoff-inf* objectives (which we refer as generalized mean-payoff objectives) is the set

$$\mathsf{MeanInf} = \left\{ s_0 s_1 \dots \in S^\omega \mid \bigwedge_{j=1}^{k} \liminf_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{rwd}_j(s_i) \geq 0 \right\}$$

that contains all plays for which the long-run average of weights (computed as $\liminf$) is non-negative[4] in all dimensions. The objectives inside the

[3] We use rational rewards to be able to state complexity results. All other results in this paper hold if the rewards are real numbers.

[4] Note that it is not restrictive to define mean-payoff objectives with a threshold 0 since we can obtain mean-payoff objectives defined as the long-run average of weights above any threshold ν by subtracting the constant ν to the reward function.

above conjunction (indexed by j) are called one-dimensional mean-payoff-inf objectives (in dimension j), and denoted $\mathsf{MeanInf}_j$. The conjunction of *mean-payoff-sup* objectives is the set $\mathsf{MeanSup}$ defined analogously, replacing $\liminf$ by $\limsup$ in the definition of $\mathsf{MeanInf}$.

Remark 1. *It is easy to show that mean-payoff-inf objectives are closed under shuffling, and that the conjunction of objectives that are closed under shuffling is closed under shuffling [38]. However, the conjunctions of mean-payoff-sup objectives are in general not closed under shuffling [48, Example 1].*

Probability measures. Given an initial state s, and a pair of strategies (σ, π) for Max and Min, a finite prefix $\rho = s_0 \cdots s_n$ of a play is *compatible* with σ and π if $s_0 = s$ and for all $0 \leq i \leq n-1$, there exists an action $a_i \in A$ such that $\delta(s_i, a_i)(s_{i+1}) > 0$, and either $s_i \in S_{\mathsf{Max}}$ and $\sigma(s_0 \cdots s_i)(a_i) > 0$, or $s_i \in S_{\mathsf{Min}}$ and $\pi(s_0 \cdots s_i)(a_i) > 0$. A probability can be assigned in a standard way to every finite play prefix ρ, and by Caratheodary's extension theorem a probability measure $\mathbb{P}_s^{\sigma,\pi}(\cdot)$ of objectives can be uniquely defined. For MDPs, we omit the strategy of the player with empty set of states, and for instance if $S_{\mathsf{Min}} = \varnothing$ we denote by $\mathbb{P}_s^{\sigma}(\cdot)$ the probability measure under strategy σ of player Max.

Value and almost-sure winning. The optimal *value* from an initial state s of a game with objective Ω is defined by

$$\langle\langle \mathsf{Max} \rangle\rangle_{val}(\Omega, s) = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \mathbb{P}_s^{\sigma,\pi}(\Omega).$$

By Martin's determinacy result [42], the optimal value is also $\langle\langle \mathsf{Min} \rangle\rangle_{val}(\Omega, s) = \inf_{\pi \in \Pi} \sup_{\sigma \in \Sigma} \mathbb{P}_s^{\sigma,\pi}(\Omega)$, the infimum probability of satisfying Ω that player Min can ensure against all strategies of player Max. In other words the determinacy shows that $\langle\langle \mathsf{Max} \rangle\rangle_{val}(\Omega, s) = \langle\langle \mathsf{Min} \rangle\rangle_{val}(\Omega, s)$, and the order of sup and inf in the quantification of the strategies can be exchanged.

A strategy σ for player Max is *optimal* from a state s if for all strategies π for player Min it ensures that $\mathbb{P}_s^{\sigma,\pi}(\Omega) \geq \langle\langle \mathsf{Max} \rangle\rangle_{val}(\Omega, s)$. The value (or winning probability) of a strategy σ in state s is $\langle\langle \sigma \rangle\rangle_{val}(\Omega, s) = \inf_{\pi \in \Pi} \mathbb{P}_s^{\sigma,\pi}(\Omega)$. We omit analogous definitions for player Min.

We say that player Max wins almost-surely from an initial state s if there exists a strategy σ for Max such that for every strategy π of player Min we have $\mathbb{P}_s^{\sigma,\pi}(\Omega) = 1$. The state s and the strategy σ are called *almost-sure* winning for player Max.

Finite-memory values and almost-sure winning. The optimal *finite-memory value* (for player Max) is defined analogously, when the players are restricted to finite-memory strategies:

$$\langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\Omega, s) = \sup_{\sigma \in \Sigma^{FM}} \inf_{\pi \in \Pi^{FM}} \mathbb{P}_s^{\sigma,\pi}(\Omega).$$

A strategy σ is *optimal for finite memory* from a state s if it uses finite memory and for all finite-memory strategies π for player Min it ensures that $\mathbb{P}_s^{\sigma,\pi}(\Omega) \geq \langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\Omega, s)$. We define analogously almost-sure winning with finite-memory strategies, and the finite-memory value $\langle\!\langle \sigma \rangle\!\rangle_{val}^{FM}(\Omega, s)$ of σ in state s (against finite-memory strategies of player Min). We define the finite-memory value for player Min by $\langle\!\langle \mathsf{Min} \rangle\!\rangle_{val}^{FM}(\Omega, s) = \inf_{\pi \in \Pi^{FM}} \sup_{\sigma \in \Sigma^{FM}} \mathbb{P}_s^{\sigma,\pi}(\Omega)$ and the finite-memory value of strategy π for player Min by $\langle\!\langle \pi \rangle\!\rangle_{val}^{FM}(\Omega, s) = \sup_{\sigma \in \Sigma^{FM}} \mathbb{P}_s^{\sigma,\pi}(\Omega)$. We show in Theorem 1 for a large class of objectives (namely, prefix-independent shuffle-closed objectives) that the finite-memory value for player Max and for player Min coincide, and allowing arbitrary strategies for player Min (against finite-memory strategies for player Max) does not change the finite-memory value.

Subgame-perfect strategies. Given a strategy σ for Max, and a finite prefix $\rho = s_0 \cdots s_k$ of a play, we denote by σ_ρ the strategy that plays from the initial state s_k what σ would play after the prefix ρ, i.e. such that $\sigma_\rho(s_k \cdot \rho') = \sigma(\rho \cdot \rho')$ for all play prefixes ρ', and $\sigma_\rho(s \cdot \rho')$ is arbitrarily defined for all $s \neq s_k$.

A strategy σ for Max is *subgame-perfect* if for all nonempty play prefixes $\rho \in S^+$, the strategy σ_ρ is optimal from the initial state $\mathsf{Last}(\rho)$. Analogously, the strategy σ is *subgame-perfect-for-finite-memory* if all strategies σ_ρ are optimal-for-finite-memory strategies from $\mathsf{Last}(\rho)$.

Value problems. Given an objective Ω, a threshold $\lambda \in \mathbb{Q}$, and an initial state s, the *value-strategy problem* asks whether there exists a strategy σ for player Max such that $\langle\!\langle \sigma \rangle\!\rangle_{val}(\Omega, s) \geq \lambda$ (or whether there exists a finite-memory strategy σ for player Max such that $\langle\!\langle \sigma \rangle\!\rangle_{val}^{FM}(\Omega, s) \geq \lambda$). The *value problem* asks whether $\langle\!\langle \mathsf{Max} \rangle\!\rangle_{val}(\Omega, s) \geq \lambda$ (resp., whether $\langle\!\langle \mathsf{Max} \rangle\!\rangle_{val}^{FM}(\Omega, s) \geq \lambda$).

End-component lemma. An important property of the end-components in MDPs is that for all strategies (with finite memory or not) with probability 1 the set of states that are visited infinitely often along a play is an end-component [27, 28]. Given a play $\rho \in S^\omega$, let $\mathsf{Inf}(\rho)$ be the set of states that occur infinitely often in ρ.

Lemma 1. *[27, 28] Given an MDP M, for all states $s \in S$ and all strategies $\sigma \in \Sigma$, we have $\mathbb{P}_s^\sigma(\{\rho \mid \mathsf{Inf}(\rho) \in \mathcal{E}(M)\}) = 1$.*

Remark 2 (Key properties for MDPs)**.** *The end-component lemma is useful in the analysis of MDPs with prefix-independent objectives, which can be decomposed into the analysis of the end-components (which have useful connectedness properties), and a reachability analysis to the end-components. Moreover, suppose we consider prefix-independent objectives, and the MDP restricted to an end-component U. Then it follows from the results of [13] that either all states of U have value 1 or all states of U have value 0. Hence for prefix-independent objectives in MDPs, the optimal value is the optimal reachability probability to the* winning *end-components, where a winning end-component is an end-component with value 1.*

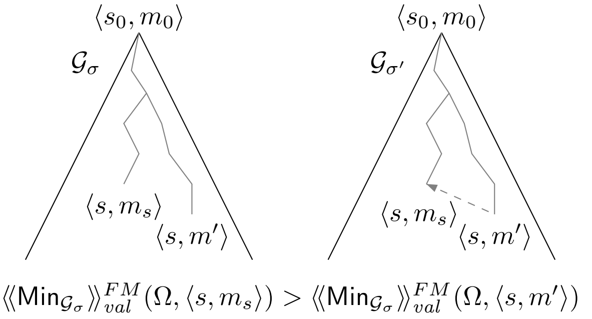

$\langle s_0, m_0 \rangle$
$\mathcal{G}_\sigma$
$\langle s, m_s \rangle$
$\langle s, m' \rangle$
$\langle s_0, m_0 \rangle$
$\mathcal{G}_{\sigma'}$
$\langle s, m_s \rangle$
$\langle s, m' \rangle$
$\langle\!\langle \mathsf{Min}_{\mathcal{G}_\sigma} \rangle\!\rangle^{FM}_{val}(\Omega, \langle s, m_s \rangle) > \langle\!\langle \mathsf{Min}_{\mathcal{G}_\sigma} \rangle\!\rangle^{FM}_{val}(\Omega, \langle s, m' \rangle)$

Figure 1: Lemma 2: construction of a strategy σ' with higher value in subgames than the optimal-for-finite-memory strategy σ.

3 Half-Memoryless Result under Finite-Memory Strategies

We show a general result that gives a sufficient condition for existence of memoryless strategies (for one of the players) in games played with finite-memory strategies.

Comment on finite- vs. infinite-memory proof. The statement and proof structure of the result are similar to [35, Theorem 5.2] that established a sufficient condition for existence of memoryless optimal strategies in games played with arbitrary (infinite-memory) strategies. However, the proof uses different techniques. The key to establish the existence of memoryless strategies for one of the players is to first establish the existence of subgame-perfect strategies for the other player. We establish such a result in Lemma 2 for finite-memory strategies. Without the restriction of finite memory, only the existence of ε-subgame-perfect strategies is known, and the proof requires intricate arguments and involved mathematical machinery such as Doob's convergence theorem for martingales [35, Theorem 4.1]. Our proof is combinatorial and uses basic results on MDPs (e.g., discrete properties of end-components).

Key ideas of the proof. The proof of Lemma 2 consists in constructing from a finite-memory strategy σ a strategy that is subgame-perfect-for-finite-memory by successively "improving" the value of the strategy σ_ρ for each finite prefix ρ. Improvements are obtained by modifying some transitions in the transducer defining σ, from the state reached after following the finite prefix ρ. The modification of transitions does not change the memory space of the strategy, and since we consider finite-memory strategies, although there may be infinitely many finite prefixes ρ where the strategy needs to be "improved", there is only a finite number of memory states to consider for improvement, which guarantees the improvement process to terminate and yields a subgame-perfect-for-finite-memory strategy.

Lemma 2. *In every stochastic game with a prefix-independent objective, there exists a subgame-perfect-for-finite-memory strategy for player* Max.

Proof. Our proof is established using the following key steps:

1. Existence of an optimal-for-finite-memory strategy for player Max.
2. Modification of the strategy for improvement of values after finite prefixes.
3. The proof that the modification provides an improvement in two parts: once the strategy for player Max is fixed, we have an MDP. In the MDP, we first show properties of the end-components, and second we provide bounds on the optimal reachability probability to the end-components to establish the improvement.

Optimal-for-finite-memory strategy. We show the existence of a finite-memory strategy σ for player Max in the game $\mathcal{G}$ such that σ is optimal-for-finite-memory from every state for the prefix-independent objective Ω. The fact that such a strategy always exists is as follows: it follows from [34, Theorem 4.3] that it suffices to prove the result for almost-sure winning strategies. Consider the set Z of states with value 1 for finite-memory strategies. We need to show that there exists a finite-memory almost-sure winning strategy in Z. Let $0 < \varepsilon < 1$, and consider a finite-memory strategy that ensures value at least $1 - \varepsilon$ from all states in Z. If a strategy can ensure positive winning from every state of a game, then it is almost-sure winning by the result of [13]. The existence of an optimal-for-finite-memory strategy follows.

Notation. Consider an optimal-for-finite-memory strategy σ. Thus for all states s of the game $\mathcal{G}$ there exists a memory value m_s in the transducer of σ such that the value of the objective Ω in the MDP $\mathcal{G}_\sigma$ is the optimal finite-memory value, that is $\langle\!\langle \mathsf{Min}_{\mathcal{G}_\sigma} \rangle\!\rangle^{FM}_{val}(\Omega, \langle s, m_s \rangle) = \langle\!\langle \mathsf{Max}_{\mathcal{G}} \rangle\!\rangle^{FM}_{val}(\Omega, s)$ where the subscript in $\mathsf{Min}_{\mathcal{G}_\sigma}$ indicates that the value is computed in the MDP $\mathcal{G}_\sigma$ (which is a MDP for player Min) while $\mathsf{Max}_{\mathcal{G}}$ gives the optimal value for player Max in the game $\mathcal{G}$.

Modification of the strategy. If the strategy σ is subgame-perfect-for-finite-memory, then the proof is done. Otherwise, there exists a state $\langle s, m' \rangle$ in $\mathcal{G}_\sigma$ with value below the optimal finite-memory value of s, namely such that $\langle\!\langle \mathsf{Min}_{\mathcal{G}_\sigma} \rangle\!\rangle^{FM}_{val}(\Omega, \langle s, m_s \rangle) > \langle\!\langle \mathsf{Min}_{\mathcal{G}_\sigma} \rangle\!\rangle^{FM}_{val}(\Omega, \langle s, m' \rangle)$. We construct an *improved* strategy σ' as follows: the strategy σ' plays like σ except that when the state $\langle s, m' \rangle$ is reached, the strategy σ' plays like σ is playing from state $\langle s, m_s \rangle$ (equivalently, we remove the outgoing transitions from state $\langle s, m' \rangle$ in $\mathcal{G}_\sigma$, and replace them by a deterministic transition to state $\langle s, m_s \rangle$ on all actions to obtain $\mathcal{G}_{\sigma'}$, as illustrated in Figure 1). Note that the new strategy σ' has the same memory set as σ. We show below that the value of every state in $\mathcal{G}_{\sigma'}$ is at least as large as the value of the same state in $\mathcal{G}_\sigma$ $(\star)$. It follows that the value of state $\langle s, m' \rangle$ in $\mathcal{G}_{\sigma'}$ is the optimal finite-memory value from s, and by repeating the same construction in every state where the value is below the optimal finite-memory value, we obtain (in finitely many steps) a subgame-perfect-for-finite-memory strategy for player Max.

Proof of $(\star)$. We proceed with the proof of $(\star)$, which has two steps as mentioned above. We first define the notion of value class.

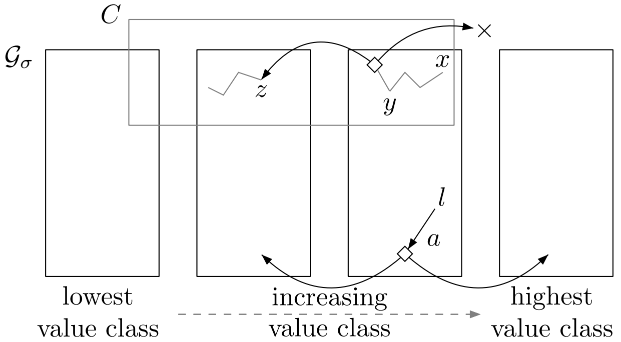


Figure 2: Lemma 2: value-class analysis. No end-component C can lie across several value classes.

Value class and properties. In the MDP $\mathcal{G}_\sigma$, a *value class* is a maximal subset of states that have the same value (defined as the infimum over the strategies of player Min). The following property holds in $\mathcal{G}_\sigma$, for every state $l = \langle \cdot, \cdot \rangle$, and action $a \in A$: consider the value class of l, if there is an a-successor of l in a lower value class, then there is also an a-successor of l in a higher value class (Figure 2). If we consider the partition defined by the value classes in $\mathcal{G}_\sigma$, this property also holds in the modified MDP $\mathcal{G}_{\sigma'}$ corresponding to strategy σ', because the new deterministic transition (dashed edge of Figure 1) goes to a higher value class.

Properties of end-components. Now, we claim that in the modified MDP $\mathcal{G}_{\sigma'}$ every end-component is included in some value class (of the original MDP $\mathcal{G}_\sigma$). We show this by contradiction (see also Figure 2). Assume that there is an end-component C in $\mathcal{G}_{\sigma'}$ with non-empty intersection with different value classes (of the original MDP $\mathcal{G}_\sigma$). Let $x \in C$ be a state of C with largest value. Since C is strongly connected, there is a path from x to a lower value class, and on this path there is a state $y \in C$ with largest value that has an a-successor z with lower value (for some $a \in A_C(y)$). It follows that y has also an a-successor with higher value, according to the above property. This successor is outside C since there is no larger value class in C than the value class of y. This is in contradiction with the fact that end-components are closed sets (and that $a \in A_C(y)$). We conclude that in $\mathcal{G}_{\sigma'}$ every end-component is included in some value class (of the original MDP $\mathcal{G}_\sigma$). Therefore, the value of each end-component in $\mathcal{G}_{\sigma'}$ is at least as large as the value of the value class containing it (in $\mathcal{G}_\sigma$). It also follows that the new deterministic transitions from $\langle s, m' \rangle$ to $\langle s, m_s \rangle$ do not belong to any end-component in $\mathcal{G}_{\sigma'}$.

Optimal reachability probability. The key steps to obtain the bound on optimal reachability probability is as follows: we observe that the optimal reachability probability in MDPs is characterized by a minimizing linear-programming solution, and we show that the solution before the modification is a feasible solution after the modification. We now present the details.

Optimal value via optimal reachability. We show that the value of the state $\langle s, m' \rangle$ in $\mathcal{G}_{\sigma'}$ is strictly greater than the value of $\langle s, m \rangle$ in $\mathcal{G}_\sigma$ (for player Max). Let S_{losing} be the union of all end-components in $\mathcal{G}_\sigma$ with value 0 for the prefix-independent objective Ω (thus losing for player Max, and winning for player Min). By Remark 2, the optimal value for player Min in the MDP is the optimal reachability probability to S_{losing}.

Optimal reachability probability to S_{losing}. Consider the following linear program in $\mathcal{G}_\sigma = \langle S', (\varnothing, S'_{\mathsf{Min}}), A, \delta' \rangle$ that computes the value (for player Min) of each state $l \in S'$ of $\mathcal{G}_\sigma$ in variable x_l, by solving a reachability problem to the states in S_{losing}:

$$\text{minimize } \textstyle\sum_{l \in S'} x_l$$

$$x_l \geq \textstyle\sum_{k \in S'} \delta'(l,a)(k) \cdot x_k \text{ for all } l \in S', a \in A$$

$$x_l = 1 \text{ for all } l \in S_{losing}$$

The correctness of the linear program to compute optimal reachability probability is standard [31]. Let x^* be an optimal solution of this linear program. Note that the values are computed for player Min, and thus $x^*_l = 1 - \langle\!\langle \mathsf{Max} \rangle\!\rangle^{FM}_{val}(\Omega, l)$. It follows that $x^*_{\langle s, m_s \rangle} < x^*_{\langle s, m' \rangle}$.

Feasible solution. Consider the modified MDP $\mathcal{G}_{\sigma'}$ (with same state space as $\mathcal{G}_\sigma$), in which the union of end-components with value 0 is contained in S_{losing}. Therefore, considering the same linear program for $\mathcal{G}_{\sigma'}$ provides an upper bound on the new value (for player Min). For each $l \in S'$, define $y_l = \begin{cases} x^*_l & \text{if } l \neq \langle s, m' \rangle \\ x^*_{\langle s, m_s \rangle} & \text{if } l = \langle s, m' \rangle \end{cases}$
Then $(y_l)_{l \in S'}$ is a feasible solution to the linear program for $\mathcal{G}_{\sigma'}$, and for the optimal solution y^*, we have $y^*_l \leq y_l \leq x^*_l$ (and for $l' = \langle s, m' \rangle$ we have $y^*_{l'} \leq y_{l'} < x^*_{l'}$). Since $y^*_{l'}$ is only an upper bound of the new value of s for player Min in $\mathcal{G}_{\sigma'}$, it shows that the value improved for player Max in every state. Since the value of $\langle s, m_s \rangle$ in $\mathcal{G}_\sigma$ was the optimal finite-memory value, it follows that in $\mathcal{G}_{\sigma'}$ the value of $\langle s, m_s \rangle$ is also the optimal finite-memory value. Since all transitions of $\langle s, m' \rangle$ lead to $\langle s, m_s \rangle$, the value of $\langle s, m' \rangle$ in $\mathcal{G}_{\sigma'}$ is the optimal finite-memory value from s, which concludes the proof of ($\star$). □

The result of [35, Theorem 5.2] shows that in games where the players are allowed to use arbitrary strategies (thus not restricted to finite-memory strategies), memoryless optimal strategies exist for player Min if the objective of player Max is prefix-independent and closed under shuffling. The proof of this result uses an analogue of Lemma 2 for arbitrary strategies, and relies on edge induction, a technique that became standard [15, 35, 36, 38]. The shape of the argument is not specific to games with arbitrary strategies: in games where the players are restricted to finite-memory strategies, we can follow the same line of proof (using Lemma 2) to show that if the objective of a player is prefix-independent and closed under shuffling, then memoryless optimal strategies exist for the other player.

Theorem 1. *In stochastic games, if the objective Ω of player* Max *is prefix-independent and closed under shuffling, and player* Max *is restricted to finite-memory strategies, then player* Min *has a memoryless optimal-for-finite-memory strategy (as well as a memoryless optimal strategy), and determinacy holds under finite-memory strategies. More precisely, for all states s we have:*

$$\langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\Omega, s) = \langle\langle \mathsf{Min} \rangle\rangle_{val}^{FM}(\Omega, s) =: v(s), \textit{ and}$$

$$\sup_{\sigma\in\Sigma^{FM}} \inf_{\pi\in\Pi} \mathbb{P}_s^{\sigma,\pi}(\Omega, s) = v(s) = \inf_{\pi\in\Pi^{PM}} \sup_{\sigma\in\Sigma^{FM}} \mathbb{P}_s^{\sigma,\pi}(\Omega, s).$$

Significance of Theorem 1. We first remark on the significance of the result, and then present the main steps of the proof. First, the result establishes determinacy for finite-memory strategies i.e., $\langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\Omega, s) = \langle\langle \mathsf{Min} \rangle\rangle_{val}^{FM}(\Omega, s) = v(s)$, which implies that even for finite-memory strategies the order of sup and inf can be exchanged. However, note that the finite-memory value is different from the value under infinite-memory strategies, and the determinacy for finite-memory does not follow from the determinacy for infinite-memory strategies. Second, $\sup_{\sigma\in\Sigma^{FM}} \inf_{\pi\in\Pi} \mathbb{P}_s^{\sigma,\pi}(\Omega, s) = v(s)$ implies that as long as player Max is restricted to finite-memory strategies, whether player Min uses finite-memory or infinite-memory strategies does not matter. Finally, $v(s) = \inf_{\pi\in\Pi^{PM}} \sup_{\sigma\in\Sigma^{FM}} \mathbb{P}_s^{\sigma,\pi}(\Omega, s)$ implies that against finite-memory strategies of player Max there exists a pure memoryless strategy for player Min that is optimal (even considering all infinite-memory strategies for player Min).

Main steps of the proof. We present the key steps of the proof of Theorem 1, and we show that the argument in the proof of [35, Theorem 5.2] (which we refer to for the precise technical steps) can be adapted for finite-memory strategies. The key steps are: (i) induction on the number of player-Min states; (ii) creating different games for different choices at a player-Min state, in which player Min has memoryless optimal strategies by induction hypothesis; and (iii) showing the value of the original game is at least the minimum of the value of the different games, thus memoryless strategies suffice.

Induction on player-Min *states.* The proof is by induction on the number of states of player Min. The base case $|S_{\mathsf{Min}}| = 0$ corresponds to games with only states of player Max. The result holds trivially in that case (the empty strategy of player Min is memoryless). For the induction step, assume that the result holds for all games with $|S_{\mathsf{Min}}| < k$, and consider a game $\mathcal{G}$ with $|S_{\mathsf{Min}}| = k$.

Different games for different choices. We explain the rest of the proof assuming the action set contains only two actions, that is $A = \{a, b\}$. The proof is the same for an arbitrary finite set of actions, with more complication in the notation. In $\mathcal{G}$, consider a state $\hat{s} \in S_{\mathsf{Min}}$ of player Min and construct two games $\mathcal{G}_a$ and $\mathcal{G}_b$ obtained from $\mathcal{G}$ by removing $\hat{s}$ and by replacing the incoming transitions to $\hat{s}$

by transitions to its a-successors and b-successors respectively. The transition function of $\mathcal{G}_x$ (for $x \in \{a, b\}$) is defined by $\delta_x(s, c)(s') = \delta(s, c)(s') + \delta(s, c)(\hat{s}) \cdot \delta(\hat{s}, x)(s')$ for all $s, s' \in S \setminus \{\hat{s}\}$, and all actions $c \in A$.

Value of original game at least the minimum of the value of the two games. In $\mathcal{G}_a$ and $\mathcal{G}_b$ the number of states of player Min is $k - 1$. Hence by the induction hypothesis there exist memoryless strategies $\pi^{\mathcal{G}_a}$ and $\pi^{\mathcal{G}_b}$ for player Min that are optimal-for-finite-memory (as well as optimal among the infinite-memory strategies) in $\mathcal{G}_a$ and $\mathcal{G}_b$ respectively. The proof proceeds by showing that in the game $\mathcal{G}$, player Min cannot obtain a lower (i.e., better) value than in one of the games $\mathcal{G}_a$ or $\mathcal{G}_b$, that is for all strategies π of player Min, for all states $s \neq \hat{s}$ we have[5]:

$$\langle\langle \pi^{\mathcal{G}} \rangle\rangle_{val}^{FM}(\Omega, s) \geq \min \left\{ \langle\langle \pi^{\mathcal{G}_a} \rangle\rangle_{val}^{FM}(\Omega, s), \langle\langle \pi^{\mathcal{G}_b} \rangle\rangle_{val}^{FM}(\Omega, s) \right\}. \tag{1}$$

To show this, we consider subgame-perfect-for-finite-memory strategies σ_a and σ_b for player Max in games $\mathcal{G}_a$ and $\mathcal{G}_b$ respectively (which exist by Lemma 2), and we construct a finite-memory strategy σ in $\mathcal{G}$ that achieves, against all strategies π, a value at least as large as either σ_a in $\mathcal{G}_a$ or σ_b in $\mathcal{G}_b$. Intuitively, σ switches between σ_a and σ_b, playing according to σ_a when in the last visit to $\hat{s}$ player Min played action a (thus as in $\mathcal{G}_a$), and playing according to σ_b when in the last visit to $\hat{s}$ player Min played action b (thus as in $\mathcal{G}_b$). To formally define σ, given a play prefix in $\mathcal{G}$ we use projections onto plays in $\mathcal{G}_a$ (resp., $\mathcal{G}_b$) that erase all sub-plays between successive visits to $\hat{s}$ where action b (resp., action a) was played in $\hat{s}$. Note that σ uses finite memory. The plays compatible with σ and π are shuffles of plays compatible with σ_a in $\mathcal{G}_a$ and plays compatible with σ_b in $\mathcal{G}_b$, and since the objective Ω is closed under shuffling, the probability measure of the plays satisfying the objective in $\mathcal{G}$ is no lower than the value of either games $\mathcal{G}_a$ or $\mathcal{G}_b$:

$$\mathbb{P}_s^{\sigma,\pi}(\Omega) \geq \min \left\{ \langle\langle \pi^{\mathcal{G}_a} \rangle\rangle_{val}^{FM}(\Omega, s), \langle\langle \pi^{\mathcal{G}_b} \rangle\rangle_{val}^{FM}(\Omega, s) \right\}.$$

It follows that (1) holds, and thus the optimal-for-finite-memory (as well as optimal among infinite-memory strategies) strategies in the games $\mathcal{G}_a$ and $\mathcal{G}_b$ (extended to play a and b respectively in $\hat{s}$) are sufficient for player Min in $\mathcal{G}$. Therefore by the induction hypothesis, memoryless strategies are sufficient for player Min to achieve the optimal finite-memory value, let π be such a strategy. By the same argument and using the induction hypothesis, for the finite-memory strategy σ for player Max in $\mathcal{G}$ we have $\langle\langle \sigma \rangle\rangle_{val}(\Omega, s) = \langle\langle \sigma \rangle\rangle_{val}^{FM}(\Omega, s) = \langle\langle \pi \rangle\rangle_{val}^{FM}(\Omega, s)$, which gives $\langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\Omega, s) = \langle\langle \mathsf{Min} \rangle\rangle_{val}^{FM}(\Omega, s)$. Note that our proof handled that the strategies for player Min are allowed to be infinite-memory, and the result still holds.

Remark 3. *The determinacy result of Theorem 1, which allows to switch the* sup *and* inf *operators ranging over finite-memory strategies, is true for prefix-independent shuffle-closed objectives. We present an example to show that such a result does not hold for general prefix-independent objectives that are*

[5] We assume that the value $\langle\langle \pi^{\mathcal{G}} \rangle\rangle_{val}^{FM}(\Omega, s)$ of a strategy $\pi^{\mathcal{G}}$ is computed in the game $\mathcal{G}$ in superscript.

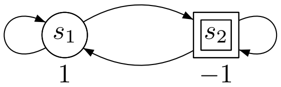


Figure 3: A game with prefix-independent objective $\mathsf{Büchi}(s_2) \wedge (\mathsf{coBüchi}(s_2) \vee \mathsf{MeanSup})$ that is not determined under finite-memory strategies.

not closed under shuffling. Consider the game of Figure 3, with the objective $\Omega = \mathsf{Büchi}(s_2) \wedge (\mathsf{coBüchi}(s_2) \vee \mathsf{MeanSup})$ where $\mathsf{Büchi}(s_2)$ is the set of plays that visit s_2 infinitely often, and $\mathsf{coBüchi}(s_2)$ is the set of plays that eventually stay in s_2 forever. Note that the game is even non-stochastic. We show that $\langle\!\langle \mathsf{Max} \rangle\!\rangle_{val}^{FM}(\Omega, s_1) = 0$ and $\langle\!\langle \mathsf{Min} \rangle\!\rangle_{val}^{FM}(\Omega, s_1) = 1$. Intuitively, after either player fixed a finite-memory strategy, the other player can win using slightly more memory than the first player (but still finite memory). For all finite-memory strategies σ of player Max, either (i) there exists a compatible play that eventually stays forever in s_1, and then the objective $\mathsf{Büchi}(s_2)$ is violated, or (ii) s_2 is visited infinitely often in all compatible plays and player Min can ensure with a finite-memory strategy that both objectives $\mathsf{MeanSup}$ and $\mathsf{coBüchi}(s_2)$ are violated by staying in s_2 one more time than player Max stayed in s_1, and then going back to s_1. It follows that $\langle\!\langle \sigma \rangle\!\rangle_{val}^{FM}(\Omega, s_1) = 0$. Analogously, against all finite-memory strategies π of player Min, player Max can ensure that the objective Ω is satisfied (by staying in s_1 one more time than player Min stayed in s_2, and then going to s_2), thus $\langle\!\langle \pi \rangle\!\rangle_{val}^{FM}(\Omega, s_1) = 1$. Hence $\langle\!\langle \mathsf{Max} \rangle\!\rangle_{val}^{FM}(\Omega, s_1) \neq \langle\!\langle \mathsf{Min} \rangle\!\rangle_{val}^{FM}(\Omega, s_1)$ and the game of Figure 3 is not determined under finite-memory strategies.

Upper bound on memory. We now show that for prefix-independent shuffle-closed objectives, the memory required for player Max is exponential as compared to the memory required for the same objective in MDPs. If there are k states for player Min, then the optimal-for-finite-memory strategy σ constructed for player Max in the proof of Theorem 1 is as follows: it considers strategies in the choice-fixed games ($\mathcal{G}_a$ and $\mathcal{G}_b$) with $k-1$ states for player Min, and the strategy in the original game considers projections of plays and then copies the strategies of the choice-fixed games. Thus the memory required for player Max in games with k states for player Min is the union of the memory required for the choice-fixed games with $k-1$ states, and there are at most $|A|$ such choice-fixed games. If we denote by $M(k)$ the memory required for player Max in games with k player-Min states, then the following recurrence is satisfied:

$$M(k) = |A| \cdot M(k-1).$$

Note that $M(0)$ represents the memory bound for MDPs, and thus we get a bound on $M(k) = |A|^k \cdot M(0)$ in games that is greater than the memory bound for MDPs by an exponential factor.

Theorem 2. *In stochastic games with a prefix-independent shuffle-closed objective Ω, an upper bound on the memory required for optimal-for-finite-memory* *strategies is* $|A|^{|S_{\mathsf{Min}}|} \cdot M(0)$*, where* $M(0)$ *is an upper bound on memory required for objective* Ω *in MDPs.*

4 Generalized Mean-Payoff Objectives under Finite-Memory Strategies

In generalized-mean-payoff games, infinite-memory strategies are more powerful than finite-memory strategies, even in 1-player games with only deterministic transitions, i.e., graphs [48, Lemma 7].[6] It follows that in general $\langle\langle \mathsf{Max} \rangle\rangle_{val}(\Omega, s) \neq \langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\Omega, s)$ in generalized-mean-payoff games (for both $\Omega = \mathsf{MeanSup}$ and $\Omega = \mathsf{MeanInf}$). In this section, we consider the value problem for finite-memory strategies, and present complexity results showing that the problem is in PTIME for MDPs, and is coNP-complete for games. Finally we present optimal bounds for memory required in $2\frac{1}{2}$-player games.

4.1 Generalized mean-payoff objectives under finite-memory in MDPs

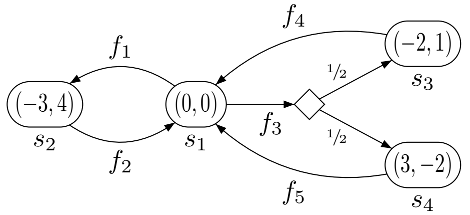
We consider the value problem for finite-memory strategies in MDPs with generalized mean-payoff objectives. First we show that randomized memoryless strategies are as powerful as finite-memory strategies, and then using this result we show that the value problem can be solved in polynomial time.

Note that in finite-state Markov chains with a fixed reward function, from all states s, the probability that the conjunction $\mathsf{MeanSup}$ of mean-payoff-sup objectives holds from s is the same as the probability that the conjunction $\mathsf{MeanInf}$ of mean-payoff-inf objectives holds from s [31]. It follows that in MDPs with finite-memory strategies, the value for mean-payoff-sup and mean-payoff-inf objectives coincides, thus $\langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\mathsf{MeanSup}, s) = \langle\langle \mathsf{Max} \rangle\rangle_{val}^{FM}(\mathsf{MeanInf}, s)$ for all states s.

Key ideas. Let $M = \langle S, A, \delta \rangle$ be an MDP and $\mathsf{rwd} : S \to \mathbb{R}^k$ be a reward function. The key ideas to show that randomized memoryless strategies are sufficient for generalized mean-payoff objectives are: (i) first observe that the mean-payoff value of a play depends only on the frequency of occurrence of each state, (ii) under finite-memory strategies the frequencies are well defined (with probability 1) for each state and action, and (iii) given the frequencies of a finite-memory strategy, a randomized memoryless strategy that plays at every state an action with probability proportional to the given frequencies achieves the same frequencies as the finite-memory strategy.

Definition of frequencies. In this paragraph, we assume that plays are sequences of alternating states and actions, and the probability measure $\mathbb{P}_s^{\sigma}(\cdot)$ is (uniquely) defined over Borel sets of such sequences. Given a play $\rho = s_0 a_0 s_1 a_1 \cdots \in (SA)^{\omega}$, let $N_i^{\rho}(s)$ be the number of occurrences of state s in ρ up to position

[6] In the example of [48, Lemma 7] all finite-memory strategies have winning probability 0 while there exists an almost-sure winning strategy (with infinite memory).



$$(E1)\begin{cases} f_1 = f_2 \\ f_1 + f_3 = f_2 + f_4 + f_5 \\ f_4 = \frac{f_3}{2} \\ f_5 = \frac{f_3}{2} \end{cases}$$

$$(E2)\begin{cases} -3f_2 - 2f_4 + 3f_5 \geq 0 \\ 4f_2 + f_4 - 2f_5 \geq 0 \end{cases}$$

$$(E3)\ \ f_1 + f_2 + f_3 + f_4 + f_5 = 1$$

Figure 4: Linear program for an MDP with two-dimensional mean-payoff objective (the constraints $f_i \geq 0$ for $i = 1, \dots, 5$ are omitted in the figure).

i, and let $N_i^\rho(s,a)$ be the number of occurrences of the pair (s,a) of state s followed by action a in ρ up to position i. The frequency f_s^ρ of a state s in ρ and the frequency $f_{s,a}^\rho$ of a pair (s,a) are:

$$f_s^\rho = \lim_{i\to\infty} \frac{N_i^\rho(s)}{i} \text{ (if the limit exists),}$$

$$f_{s,a}^\rho = \lim_{i\to\infty} \frac{N_i^\rho(s,a)}{i} \text{ (if the limit exists).}$$

Mean-payoff values from frequencies. It is easy to see that if the frequency f_s^ρ is defined, then the mean-payoff values $v_j = \limsup_{n\to\infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \mathsf{rwd}_j(s_i)$ of ρ (for $j = 1, \dots, k$) can be computed from the values of f_s^ρ as follows: $v_j = \sum_{s\in S} f_s^\rho \cdot \mathsf{rwd}_j(s)$.

Almost-sure existence of frequencies. We note that in finite-state Markov chains (and in finite-state MDPs with finite-memory strategies) the frequencies are defined with probability 1 [31]:

$$\mathbb{P}_s^\sigma(\{\rho \mid f_s^\rho \text{ and } f_{s,a}^\rho \text{ are defined for all } s \in S \text{ and } a \in A\}) = 1.$$

Given a finite-memory strategy σ in an MDP M, consider a closed recurrent set B of the Markov chain M_σ obtained by playing σ in M[7]. Then, given $s_0 \in B$ the frequencies are fixed in B, that is there exist (unique) frequency vectors $\lambda : B \to [0,1]$ and $\mu : B \times A \to [0,1]$ such that:

$$\begin{aligned} &\mathbb{P}_{s_0}^\sigma(\{\rho \mid \tfrac{f_{s,a}^\rho}{f_s^\rho} = \mu(s,a) \text{ for all } s \in B \text{ and } a \in A\}) = 1, \text{ and} \\ &\mathbb{P}_{s_0}^\sigma(\{\rho \mid f_s^\rho = \lambda(s) \text{ for all } s \in B\}) = 1. \end{aligned}$$

Strategies given frequencies. In the closed recurrent set B, consider a finite-memory strategy σ' that plays actions with the same frequencies as σ, that is such that:

$$\mathbb{P}_{s_0}^{\sigma'}(\{\rho \mid \frac{f_{s,a}^\rho}{f_s^\rho} = \mu(s,a) \text{ for all } s \in B \text{ and } a \in A\}) = 1$$

[7] The state space of M_σ is the Cartesian product of the set of states of M and of the memory set of σ. However, we say that $s \in B$ if (s,m) is a state of B for some memory value m of σ.

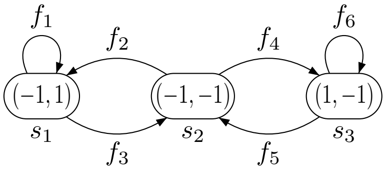


Figure 5: The (disjoint) union of two end-components corresponds to a solution of LP ($f_1 = f_6 = \frac{1}{2}$ and $f_2 = f_3 = f_4 = f_5 = 0$). However, no single end-component is a solution.

Then the frequencies of the states in B are given by λ:

$$\mathbb{P}_{s_0}^{\sigma'}(\{\rho \mid f_s^\rho = \lambda(s) \text{ for all } s \in B\}) = 1.$$

Sufficiency of randomized memoryless strategies. Now, consider a randomized memoryless strategy $\bar{\sigma}$ such that $\bar{\sigma}(s) = d_s$ for all $s \in B$ where $d_s \in \Delta(A)$ is such that $d_s(a) = \mu(s, a)$ for all $a \in A$. Thus $\bar{\sigma}$ plays each action with the frequencies given by μ, as in σ. It follows that

$$\mathbb{P}_{s_0}^{\bar{\sigma}}(\{\rho \mid \frac{f_{s,a}^\rho}{f_s^\rho} = \mu(s,a) \text{ for all } s \in B \text{ and } a \in A\}) = 1,$$

and thus

$$\mathbb{P}_{s_0}^{\bar{\sigma}}(\{\rho \mid f_s^\rho = \lambda(s) \text{ for all } s \in B\}) = 1.$$

Since the mean-payoff value depends only on the frequency of each state, it follows that $\mathbb{P}_{s_0}^{\sigma}(\mathsf{MeanSup}) = \mathbb{P}_{s_0}^{\bar{\sigma}}(\mathsf{MeanSup})$ for all $s_0 \in B$, and $\bar{\sigma}$ is a randomized memoryless strategy. Thus randomized memoryless strategies can achieve the same values as arbitrary finite-memory strategies. Note that the result holds for conjunctions of mean-payoff-inf objectives as well (by the same proof). By Remark 2 the winning probability from an initial state is the maximum probability to reach end-components with value 1, which is obtained by a pure memoryless strategy. It follows that randomized memoryless strategies are sufficient in MDPs with mean-payoff objectives to realize the finite-memory value.

Lemma 3. *In all MDPs with a generalized mean-payoff objective, there exists an optimal-for-finite-memory strategy that is randomized memoryless.*

Polynomial-time algorithm We present a polynomial-time algorithm to compute the value in generalized mean-payoff MDPs with finite-memory strategies. The key steps of the algorithm are:

- The algorithm determines all end-components with value 1 (the winning end-components), and then computes the maximum probability to reach the union of the winning end-components (see Remark 2).
- The first step to obtain the winning end-components is to define a linear program based on the frequencies that gives a union of end-components

with frequencies that satisfy the generalized mean-payoff objective. However, this union of end-components itself may not be connected, even though it is part of a larger end-component. In the infinite-memory strategy case, the paths between the union of end-components can be used with vanishing frequency to ensure the generalized mean-payoff objectives. However, for finite-memory strategies connectedness of the union of the end-components must be ensured. We show how to combine the linear program with a graph-based algorithm to ensure connectedness and get a polynomial-time algorithm.

Frequency-based linear program. It is known that the winning probability for reachability objectives can be computed in polynomial time using a reduction to linear programming [31]. To complete the proof, we present a solution to compute the winning end-components in polynomial time. Our approach extends a technique for finding in a graph a cycle with sum of rewards equal to zero in all dimensions [39]. First, we present a linear program LP to find a union of end-components with nonnegative sum of rewards (the end-components may be disjoint). The variables $f_{s,a}$ represent the frequency of playing action a in state s. The linear program LP consists of the following constraints (see also Figure 4):

(E1) for each $s \in S$: $\sum_{a \in A} f_{s,a} = \sum_{t \in S} \sum_{a \in A} f_{t,a} \cdot \delta(t,a)(s)$

(E2) $\sum_{s \in S} \sum_{a \in A} f_{s,a} \cdot \mathsf{rwd}(s) \geq 0$ (component-wise)

(E3) $\sum_{s \in S} \sum_{a \in A} f_{s,a} = 1$

(E4) for each $s \in S$ and $a \in A$: $f_{s,a} \geq 0$

The equations (E1) above express that in every state, the incoming frequency is equal to the outgoing frequency. Equation (E2) ensures that the mean-payoff value is nonnegative (in all dimensions). Equations (E3) and (E4) require that the frequencies are nonnegative and sum up to 1.

Illustration. In the example of Figure 4, a solution to the linear program gives for instance $f_1 = \frac{1}{16}$ and $f_3 = \frac{7}{16}$, which corresponds to a randomized memoryless strategy that chooses from s_1 to go to s_2 with probability $\frac{1}{1+7} = \frac{1}{8}$ and to go to $\{s_3, s_4\}$ with probability $\frac{7}{1+7} = \frac{7}{8}$. This strategy satisfies the conjunction of mean-payoff objectives with probability 1 (it ensures that the long-run average of the rewards is $\frac{1}{32} \geq 0$ in both dimensions).

Issues regarding connectedness. Arguments similar to the proof of [39, Theorem 2.2] show that the linear program LP has a solution if and only if there exists a union of end-components in M and associated frequencies with nonnegative sum of rewards. However, this union of end-components need not to be connected and thus may not be an end-component (see Figure 5 where the union of the end-components $\{s_1\}$ and $\{s_3\}$ corresponds to a solution of LP). Note that connectedness is not an issue for infinite-memory strategies: in the example

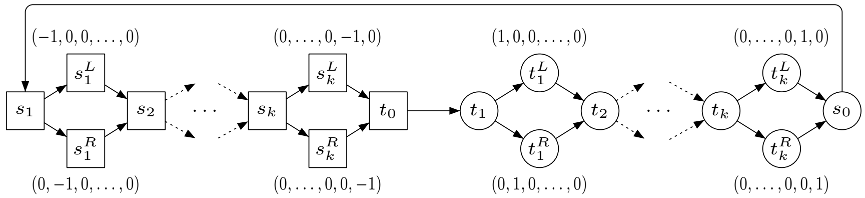


Figure 6: A family of generalized mean-payoff games where player Max (round states) needs exponential memory to win almost-surely (and finite memory is sufficient).

of Figure 5 there exists an infinite-memory strategy to ensure the mean-payoff objectives with probability 1 (see [48, Lemma 7]).

Ensuring connectedness and frequencies. To find single end-components with nonnegative sum of rewards, we adapt a technique presented in [39, Section 3]. Construct a graph G_M with set S of vertices, and for each pair $(s,a) \in S \times A$, if the linear program $\mathsf{LP} \wedge f_{s,a} > 0$ has a solution, add edges (s,t) in G_M for all a-successors t of s. If the graph G_M is strongly connected, then it defines an end-component with nonnegative sum of rewards in M. Otherwise, consider the maximum-scc decomposition of G_M, and iterate the algorithm in each scc, until the state space reduces to one element. The algorithm identifies in this way all (maximal) winning end-components and arguments similar to [39, Theorem 3.3] show that this algorithm runs in polynomial time, as the recursion depth is bounded by the number of states, and the scc decomposition ensures that the graphs in each recursive call of a given depth are disjoint.

Theorem 3. *The following assertions hold for MDPs with generalized mean-payoff objectives $\Omega \in \{\mathsf{MeanSup}, \mathsf{MeanInf}\}$:*

1. *There exists a randomized memoryless strategy σ such that $\langle\!\langle \mathsf{Max} \rangle\!\rangle_{val}^{FM}(\Omega, s) = \mathbb{P}_s^{\sigma}(\Omega, s)$ for all states s (i.e., randomized memoryless optimal strategies wrt. to finite-memory strategies).*

2. *The value and value-strategy problems for generalized mean-payoff MDPs under finite-memory strategies (i.e., whether $\langle\!\langle \mathsf{Max} \rangle\!\rangle_{val}^{FM}(\Omega, s) \geq \lambda$) can be solved in polynomial time.*

Insufficiency of pure memoryless strategies. While we show that randomized memoryless strategies are sufficient, the example of Figure 4 shows that pure memoryless strategies are not sufficient to achieve the optimal finite-memory value: from s_1, a pure memoryless strategy can either choose s_2 and then the mean-payoff value in the first dimension is $-\frac{3}{2} < 0$, or choose $\{s_3, s_4\}$ and then the mean-payoff value in the second dimension is $-\frac{1}{2} < 0$. Thus for all pure memoryless strategies, the generalized mean-payoff objective is violated with probability 1 although there exists an almost-sure winning *randomized* memoryless strategy (see the paragraph *Illustration* after Lemma 3).

4.2 Generalized mean-payoff objectives under finite-memory in $2\frac{1}{2}$-player games

We present a result analogous to Theorem 1 for generalized mean-payoff stochastic games showing that memoryless strategies are sufficient for player Min against finite-memory strategies. Note that the result extends Theorem 1 as mean-payoff-sup objectives are not closed under shuffling (Remark 1).

Theorem 4. *In stochastic games with objective $\Omega \in \{\mathsf{MeanSup}, \mathsf{MeanInf}\}$, there exists an optimal-for-finite-memory strategy for player* Max*, there exists a memoryless optimal-for-finite-memory strategy for player* Min*, and determinacy holds under finite-memory strategies, that is for all states s:*

$$\langle\!\langle \mathsf{Max} \rangle\!\rangle_{val}^{FM}(\Omega, s) = \langle\!\langle \mathsf{Min} \rangle\!\rangle_{val}^{FM}(\Omega, s) =: v(s), \textit{and}$$

$$\sup_{\sigma \in \Sigma^{FM}} \inf_{\pi \in \Pi} \mathbb{P}_s^{\sigma,\pi}(\Omega, s) = v(s) = \inf_{\pi \in \Pi^{PM}} \sup_{\sigma \in \Sigma^{FM}} \mathbb{P}_s^{\sigma,\pi}(\Omega, s).$$

Proof. For mean-payoff-inf objectives ($\Omega = \mathsf{MeanInf}\}$) the result follows from Theorem 1. We consider mean-payoff-sup objectives ($\Omega = \mathsf{MeanSup}\}$). Once a finite-memory strategy for player Max is fixed we have an MDP for player Min, with a disjunction of mean-payoff objectives. We now analyze the MDP problem. Since the objective is prefix-independent, by Remark 2, every end-component has value either 1 or 0. It follows that in every end-component with value 1, one of the mean-payoff objectives is satisfied with value 1, for which positional strategies are sufficient [31, 44]. The optimal reachability to winning end-components is also achieved by positional strategies. Given the existence of positional strategies for player Min in MDPs, we consider the game problem. It follows that once a finite-memory strategy for player Max is fixed, the counter-strategy for player Min is also finite-memory, and for finite-memory strategies mean-payoff-sup and mean-payoff-inf objectives coincide. The desired result follows. □

It follows that the value problem for generalized mean-payoff games with finite-memory strategies can be solved in coNP by guessing a memoryless strategy for player Min and checking whether the value of the resulting MDP under finite-memory strategies for player Max is above the given threshold, which can be done in polynomial time (Theorem 3). By the result of [48, Lemma 5, Lemma 6], the problem of deciding the existence of a finite-memory almost-sure winning strategy for player Max in a game (even with only deterministic transitions) with a conjunction of mean-payoff-sup or mean-payoff-inf objectives is coNP-hard. Theorem 5 summarizes the results of this section.

Theorem 5. *The value and value-strategy problems for stochastic games with generalized mean-payoff-(inf or sup) objectives played with finite-memory strategies for player* Max *(and finite- or infinite-memory strategies for player* Min*) are coNP-complete.*

4.3 Memory bounds for strategies in $2\frac{1}{2}$-player games

We present both exponential lower bound and upper bound on memory of strategies.

Lower bound. We show that in games where finite memory is sufficient to win almost-surely a conjunction of mean-payoff objectives, exponential memory is necessary in general, even with randomized strategies. There exists a family of generalized mean-payoff games in which player Max has a finite-memory almost-sure winning strategy, and every almost-sure winning strategy needs exponential memory. The family of games is illustrated in Figure 6, and is essentially the same family used in the proof of [22, Lemma 8] (for 2-player games with pure finite-memory strategies).

Lower bound family. For $k = 1, 2, \ldots$, the k-th game in the family consists of $2k$ gadgets where the first k gadgets belong to player Min, and the last k gadgets belong to player Max, for a total of $6k + 2$ states (Figure 6). The action set is $\{L, R\}$ and the unique L-successor of s_i is s_i^L, the unique L-successor of t_i is t_i^L, and similarly for R-successors. Thus all transitions in the games are deterministic. The reward function has dimension $2k$. All states have reward 0 in all dimensions except the following states and dimensions:

- $\mathsf{rwd}_{2i-1}(s_i^L) = \mathsf{rwd}_{2i}(s_i^R) = -1$
- $\mathsf{rwd}_{2i-1}(t_i^L) = \mathsf{rwd}_{2i}(t_i^R) = 1$

An almost-sure winning strategy for player Max in this game is to copy in every state t_i the choice of player Min in state s_i, namely choosing t_i^L if player Min has chosen s_i^L, and choosing t_i^R if player Min has chosen s_i^R. This strategy ensures that the sum of rewards in all dimensions is bounded (in the interval $[-1, 1]$) and thus the mean-payoff value is 0 in all dimensions. The memory needed to describe this strategy requires k bits to remember the last k choices of player Min, thus a memory set of size 2^k.

Lemma 4. *There exists a family of games G_k with $O(k)$ states and generalized mean-payoff objective of dimension $2k$ such that a finite-memory almost-sure winning strategy exists, and all almost-sure winning randomized strategies require memory of size at least 2^k.*

Proof. We show that no randomized strategy with memory of size less than 2^k is almost-sure winning in the family of games G_k presented above (see Figure 6). The proof is by contradiction. Consider a strategy σ with memory of size less than 2^k, and show that σ is not almost-sure winning. Since σ has memory of size less than 2^k, after every visit to s_1, it is possible for player Min to make two different sequences of choices until reaching t_1, such that from t_1 the strategy σ of player Max behaves identically because it is in the same memory state. Formally, for all play prefixes $\rho \in S(S^{4k+2})^*$ with $\mathsf{Last}(\rho) = s_0$, there exist two play prefixes $\rho_1, \rho_2 \in S^{2k+1}$ with $\rho_1 \neq \rho_2$ such that $\hat{\sigma}_u(m_0, \rho\rho_1) = \hat{\sigma}_u(m_0, \rho\rho_2)$, where m_0 is the initial memory and $\hat{\sigma}_u$ is the update function of the strategy σ. We can view ρ_1 and ρ_2 as sequences of k bits taking value L or R. Since $\rho_1 \neq \rho_2$, there is an index $1 \leq i \leq k$ where the i-th bit differs in ρ_1 and ρ_2; we say that there is a star at position i after prefix ρ. Intuitively, the star represents the possibility for player Min to pick either L or R at position i (by playing according to ρ_1 or ρ_2) without affecting the future choices of player Max. Note that the choice of L or R at position i changes the reward in dimensions $2i-1$ and $2i$ (respective rewards 0 and -1 or -1 and 0) and possibly in other dimensions. Consider the (finite-memory) strategy for player Min that for each such prefix ρ plays as prescribed by ρ_1, and consider the Markov chain obtained by fixing the strategies σ and π in the game G_k. Consider a closed recurrent set in this Markov chain, and compute for each index $i = 1, \ldots, k$ the frequency of occurrence of a star (the frequencies are well-defined because the Markov chain has finitely many states). It is easy to see that for some index i this frequency is greater than 0 (in fact, at least $\frac{1}{k}$). Let f_* be the frequency of occurrence of a star at position i, relative to the frequency of occurrence of state s_1. Given the index i, we consider the dimensions $2i-1$ and $2i$. In the closed recurrent set, we can analogously compute the relative frequency of the left and right choices at s_i and t_i (without counting the choices when there is a star at position i), thus computing the frequency f_1 of rewards $(-1, 0)$ in dimensions $2i-1$ and $2i$, the frequency f_2 of rewards $(0, -1)$, the frequency g_1 of $(1, 0)$, and g_2 of $(0, 1)$. Thus we have $f_1 + f_2 + f_* = 1$ and $g_1 + g_2 = 1$. Since $f_* > 0$, it follows that either $f_1 + f_* > g_1$ or $f_2 + f_* > g_2$, that is either in dimension $2i-1$ or $2i$, the expected reward can be made negative by choosing either always the choice L or always the choice R at all states that have a star at position i (remember that this modifies the strategy π of player Min, but does not affect the choices made by the strategy σ of player Max along the play). It follows that almost-surely the mean-payoff value is negative in that dimension, showing that σ is not almost-surely winning.

It follows that exponential memory is necessary in general to win almost-surely a generalized mean-payoff game (even when finite memory is sufficient). □

Upper bound. Theorem 2 and Theorem 3 establish an $|A|^{|S_{\mathsf{Min}}|}$ upper bound on memory required for optimal-for-finite-memory strategies. Thus we obtain the following result.

Theorem 6. *The optimal bound for memory required for optimal-for-finite-memory strategies for player* Max *in generalized mean-payoff stochastic games is exponential.*

5 Generalized Mean-Payoff Objectives under Infinite-Memory Strategies

In this section, we consider games with a conjunction of mean-payoff objectives and infinite-memory strategies for player Max (which are more powerful than finite-memory strategies [48, Lemma 7]).

5.1 MeanInf objectives

Since MeanInf objectives are prefix-independent and closed under shuffling, it follows from the results of [35, Theorem 5.2] that for player Min memoryless optimal strategies exist. Therefore the value and value-strategy problems can be solved in coNP by guessing a (optimal) memoryless strategy for player Min, and then solving an MDP with conjunction of mean-payoff objectives under infinite-memory strategies, which can be done in polynomial time by the result of [8, Section 3.2]. A matching coNP-hardness bound is known for 2-player games [48, Theorem 7].

Theorem 7. *The value and the value-strategy problems for stochastic games with generalized mean-payoff-inf objectives under infinite-memory strategies are coNP-complete.*

5.2 MeanSup objectives

We focus on the *almost-sure winning problem* for generalized mean-payoff objectives, which is to decide whether there exists an almost-sure winning strategy for player Max from a given state. We show that the almost-sure winning problem is in NP $\cap$ coNP for a conjunction of mean-payoff-sup objectives.

Remark 4. *As mentioned in [16, Remark 1], it follows from the results of [18, Lemma 7] and [34, Theorem 4.1] that since mean-payoff objectives are prefix-independent objectives, the memory requirement for optimal strategies of both players is the same as for almost-sure winning strategies, and if the almost-sure winning problem is in NP $\cap$ coNP, then the value-strategy problem is also in NP $\cap$ coNP. Thus it will follow from our results for the almost-sure problem that the value and value-strategy problems are in NP $\cap$ coNP for* MeanSup *objectives.*

For mean-payoff-sup objectives, we show that the almost-sure winning problem is in NP $\cap$ coNP. For player Max to be almost-sure winning for a conjunction of mean-payoff-sup objectives, it is necessary to be almost-sure winning for each one-dimensional mean-payoff-sup objective, and we show that it is sufficient. An almost-sure winning strategy is to play in rounds according to the almost-sure winning strategy of each one-dimensional objective successively, for a duration that is always finite but long enough to ensure the corresponding one-dimensional average of rewards (thus over finite plays) tends to the objective mean-payoff value with high probability (that tends to 1 as the number of rounds increases).

Lemma 5. *If in a game, for every one-dimensional mean-payoff-sup objective* $\mathsf{MeanSup}_j$ *($j = 1, \dots, k$) all states are almost-sure winning for player* Max*, then for the objective* $\mathsf{MeanSup} = \bigwedge_{1 \le j \le k} \mathsf{MeanSup}_j$ *all states are almost-sure winning for player* Max*.*

Proof. To show this result, consider for each objective $\mathsf{MeanSup}_j$ an almost-sure winning strategy σ_j. Then for all $\varepsilon > 0$, there exists a number of steps N_ε such that for all $N \geq N_\varepsilon$, for all states s, all dimensions $1 \leq j \leq k$, and all strategies π of player Min, we have [16, Lemma 1]:

$$\mathbb{P}_s^{\sigma_j,\pi}\left(\left\{s_0 s_1 \cdots \in S^\omega \mid \frac{1}{N} \cdot \sum_{i=0}^{N-1} \mathsf{rwd}_j(s_i) \geq -\varepsilon\right\}\right) \geq 1-\varepsilon.$$

Thus by playing according to strategy σ_j for a large enough number of steps, the average reward in dimension j can be made arbitrarily close to 0, with probability arbitrarily close to 1.

Let $W = \max_{s \in S, 1 \leq j \leq k} |\mathsf{rwd}_j(s)|$ be the largest reward in absolute value. We construct an almost-sure winning strategy σ for the objective $\mathsf{MeanSup}$ as follows. The strategy σ plays in rounds numbered $1, 2, \dots,$ where at round ℓ the strategy σ plays for each dimension $j = 1, \dots, k$ successively as follows: let Z be the length of the current play prefix, and let $\varepsilon_Z = \frac{1}{Z}$; play according to strategy σ_j from the current play prefix for $N_Z = \max\{N_{\varepsilon_Z}, Z^2 \cdot W\}$ steps.

Now we show that the strategy σ is almost-sure winning for the conjunction of mean-payoff-sup objectives. Consider an arbitrary dimension j. For all play prefixes of length Z, the total reward is at least $-Z \cdot W$. It follows that after playing according to σ_j for N_Z steps (at round ℓ), since $N_Z \geq N_{\varepsilon_Z}$, against all strategies π of player Min, we have:

$$\mathbb{P}_s^{\sigma,\pi}\left(\left\{s_0 s_1 \cdots \in S^\omega \mid \frac{1}{N_Z} \cdot \sum_{i=0}^{N_Z-1} \mathsf{rwd}_j(s_i) \geq \frac{-Z \cdot W - \varepsilon_Z \cdot N_Z}{Z + N_Z}\right\}\right) \geq 1-\varepsilon_Z.$$

Note that $\frac{-Z \cdot W - \varepsilon_Z \cdot N_Z}{Z+N_Z} \geq \frac{-Z \cdot W - \varepsilon_Z \cdot N_Z}{N_Z} \geq \frac{-Z \cdot W}{Z^2 \cdot W} - \varepsilon_Z \geq -\frac{2}{Z}$. Therefore,

$$\limsup_{Z \to \infty} \mathbb{P}_s^{\sigma,\pi}\left(\left\{s_0 s_1 \cdots \in S^\omega \mid \frac{1}{N_Z} \cdot \sum_{i=0}^{N_Z-1} \mathsf{rwd}_j(s_i) \geq \frac{-2}{Z}\right\}\right) \geq \limsup_{Z \to \infty} 1-\varepsilon_Z = 1.$$

By Fatou's lemma [5], for a sequence $\mathcal{E}_Z$ of measurable sets we have that $\limsup_{Z \to \infty} \mathbb{P}(\mathcal{E}_Z) \leq \mathbb{P}(\limsup_{Z \to \infty} \mathcal{E}_Z)$. Hence we have

$$\mathbb{P}_s^{\sigma,\pi}\underbrace{\left(\limsup_{Z \to \infty}\left\{s_0 s_1 \cdots \in S^\omega \mid \frac{1}{N_Z} \cdot \sum_{i=0}^{N_Z-1} \mathsf{rwd}_j(s_i) \geq \frac{-2}{Z}\right\}\right)}_{\varphi} = 1.$$

We show that $\varphi \subseteq \mathsf{MeanSup}_j$. Consider a play $\rho = s_0 s_1 \cdots \in \varphi$. Since for $Z \to \infty$, we have $N_Z \to \infty$ and $\frac{-2}{Z} \to 0$, for *all* $\varepsilon > 0$ there exist infinitely many integers N such that $\frac{1}{N} \cdot \sum_{i=0}^{N-1} \mathsf{rwd}_j(s_i) \geq -\varepsilon$. Therefore $\limsup_{N \to \infty} \frac{1}{N} \cdot \sum_{i=0}^{N-1} \mathsf{rwd}_j(s_i) \geq 0$ and thus $\rho \in \mathsf{MeanSup}_j$. It follows that $\mathbb{P}_s^{\sigma,\pi}\left(\mathsf{MeanSup}_j\right) = 1$ and since this holds for all dimensions j, we have $\mathbb{P}_s^{\sigma,\pi}(\mathsf{MeanSup}) = 1$. Hence σ is almost-sure winning for the conjunction of mean-payoff-sup objectives, from all states s. □

Theorem 8. *The value and the value-strategy problems for stochastic games with generalized mean-payoff-sup objectives under infinite-memory strategies are in NP $\cap$ coNP.*

The NP algorithm. By Lemma 5, an NP algorithm for the almost-sure winning problem is to guess the set Win of almost-sure winning states, and check that (i) Win *induces a subgame* for player Max, that is for every state $s \in \mathsf{Win} \cap S_{\mathsf{Max}}$ there exists an action $a \in A$ such that $\mathsf{Supp}(\delta(s,a)) \subseteq \mathsf{Win}$, and for every state $s \in \mathsf{Win} \cap S_{\mathsf{Min}}$ for all actions $a \in A$, we have $\mathsf{Supp}(\delta(s,a)) \subseteq \mathsf{Win}$; and (ii) in the subgame induced by Win every state is almost-sure winning for every one-dimensional mean-payoff-sup objective, which can be checked in NP [41]. This establishes the first part of Theorem 8.

The coNP algorithm. We now show that the almost-sure winning problem is also in coNP. Given a set $T \subseteq S$ of states, let

$$\begin{aligned}\mathsf{Cpre}_{\mathsf{Min}}(T) = &\{s \in S_{\mathsf{Max}} \mid \forall a \in A : \mathsf{Supp}(\delta(s,a)) \cap T \neq \varnothing\} \cup \\ &\{s \in S_{\mathsf{Min}} \mid \exists a \in A : \mathsf{Supp}(\delta(s,a)) \cap T \neq \varnothing\}.\end{aligned}$$

be the set of *controllable predecessors* of set T for player Min. The *positive attractor* $\mathsf{Attr}_{\mathsf{Min}}(T)$ of T for player Min is the least fixed point of the operator $f : 2^S \to 2^S : X \mapsto T \cup \mathsf{Cpre}_{\mathsf{Min}}(X)$, that is the set of all states from which player Min has a memoryless strategy to ensure that T is reached with positive probability, against all strategies of player Max. Note that the set $S \setminus \mathsf{Attr}_{\mathsf{Min}}(\{s\})$ induces a subgame for player Max.

Let $\overline{\mathsf{Win}} = S \setminus \mathsf{Win}$ be the set of all states that are not almost-sure winning for player Max (for the conjunction $\mathsf{MeanSup}$ of mean-payoff-sup objectives). By Lemma 5 it follows that (i) there exist a dimension j and a state $s \in \overline{\mathsf{Win}}$ that is not almost-sure winning for player Max (for the one-dimensional objective $\mathsf{MeanSup}_j$), and (ii) in the subgame induced by the set $S \setminus \mathsf{Attr}_{\mathsf{Min}}(\{s\})$, the set of states that are not almost-sure winning for player Max (for $\mathsf{MeanSup}$) is $\overline{\mathsf{Win}} \setminus \mathsf{Attr}_{\mathsf{Min}}(\{s\})$ (because if player Max has an almost-sure winning strategy in the subgame, then this strategy is also almost-sure winning in the original game).

It follows that the set $\overline{\mathsf{Win}}$ can be partitioned into sets $U_1, \dots, U_n$ such that for all $1 \leq i \leq n$ there exists $\varnothing \neq R_i \subseteq U_i$ such that $U_i = \mathsf{Attr}_{\mathsf{Min}}(R_i)$ and for some dimension j the states of R_i are not almost-sure winning for player Max for the one-dimensional objective $\mathsf{MeanSup}_j$, in the subgame induced by $S \setminus (U_1 \cup \dots \cup U_{i-1})$. In each set U_i we can fix a memoryless strategy π for player Min as follows: in R_i where the game objective is $\mathsf{MeanSup}_j$, fix a memoryless optimal strategy for player Min to violate the objective $\mathsf{MeanSup}_j$ with positive probability (which exists for one-dimensional mean-payoff games [41]), and in $U_i \setminus R_i$ fix a memoryless strategy to ensure R_i is reached with positive probability (which exists by the definition of positive attractor). The strategy π ensures that against any strategy of player Max, the objective $\mathsf{MeanSup}$ is violated with positive probability.

Hence a coNP algorithm for the almost-sure winning problem is to guess the set $\overline{\mathsf{Win}}$ of states that are not almost-sure winning for player Max, and a memoryless strategy π for player Min. The verification can be done by checking that the MDP obtained by playing π in the game is not almost-sure winning for the $\mathsf{MeanSup}$ objective, which can be done in polynomial time [8, Section 3.2]. This establishes the second part of Theorem 8.

Note that improving the NP $\cap$ coNP bound to PTIME for even single dimensional objectives would be a major breakthrough, as it would imply a polynomial solution for simple stochastic games [26].

6 Conclusion

In this work we consider $2\frac{1}{2}$-player games with generalized mean-payoff objectives. We establish an optimal complexity result of coNP-completeness under finite-memory strategies, which significantly improves the previously known semi-decision procedure, even for the special case of the almost-sure problem. We also establish optimal bounds for the memory required for finite-memory strategies. Given several quantitative objectives, a more general problem is to consider a different probability threshold for each objective (in contrast we consider the probability of the conjunction of the objectives). For the almost-sure problem the more general problem coincides with the problem we consider. The more general problem is open, even for the special case of multiple reachability objectives in $2\frac{1}{2}$-player games.

References

[1] E. Altman. *Constrained Markov Decision Processes (Stochastic Modeling).* Chapman & Hall/CRC, 1999.

[2] C. Baier, C. Dubslaff, and S. Klüppelholz. Trade-off analysis meets probabilistic model checking. In *CSL-LICS 2014*, pages 1:1–1:10, 2014.

[3] C. Baier, J. Klein, S. Klüppelholz, and S. Wunderlich. Weight monitoring with linear temporal logic: complexity and decidability. In *CSL-LICS 2014*, pages 11:1–11:10, 2014.

[4] N. Basset, M. Z. Kwiatkowska, U. Topcu, and C. Wiltsche. Strategy synthesis for stochastic games with multiple long-run objectives. In *TACAS*, LNCS 9035, pages 256–271. Springer, 2015.

[5] P. Billingsley. *Probability and Measure.* Wiley-Interscience, 1995.

[6] R. Bloem, K. Chatterjee, T. A. Henzinger, and B. Jobstmann. Better quality in synthesis through quantitative objectives. In *Proc. of CAV*, LNCS 5643, pages 140–156. Springer, 2009.

[7] A. Bohy, V. Bruyère, E. Filiot, and J.-F. Raskin. Synthesis from LTL specifications with mean-payoff objectives. In *Proc. of TACAS*, LNCS 7795, pages 169–184. Springer, 2013.

[8] T. Brázdil, V. Brozek, K. Chatterjee, V. Forejt, and A. Kucera. Markov decision processes with multiple long-run average objectives. *Logical Methods in Computer Science*, 10(1:13), 2014.

[9] R. Brenguier and J. Raskin. Pareto curves of multidimensional mean-payoff games. In *CAV 2015*, pages 251–267, 2015.

[10] J. R. Büchi and L. H. Landweber. Solving sequential conditions by finite-state strategies. *SIAM J. on Control and Opt.*, 25(1):206–230, 1987.

[11] P. Cerný, K. Chatterjee, T. A. Henzinger, A. Radhakrishna, and R. Singh. Quantitative synthesis for concurrent programs. In *Proc. of CAV*, LNCS 6806, pages 243–259. Springer, 2011.

[12] P. Cerný, S. Gopi, T. A. Henzinger, A. Radhakrishna, and N. Totla. Synthesis from incompatible specifications. In *Proc. of EMSOFT*, pages 53–62. ACM-Press, 2012.

[13] K. Chatterjee. Concurrent games with tail objectives. *Theor. Comput. Sci.*, 388:181–198, December 2007.

[14] K. Chatterjee. Markov decision processes with multiple long-run average objectives. In *FSTTCS*, pages 473–484, 2007.

[15] K. Chatterjee and L. Doyen. Energy parity games. *Theoretical Computer Science*, 458(2):49–60, 2012.

[16] K. Chatterjee, L. Doyen, H. Gimbert, and Y. Oualhadj. Perfect-information stochastic mean-payoff parity games. In *FoSSaCS*, LNCS 8412, pages 210–225. Springer-Verlag, 2014.

[17] K. Chatterjee, V. Forejt, and D. Wojtczak. Multi-objective discounted reward verification in graphs and MDPs. In *LPAR*, pages 228–242, 2013.

[18] K. Chatterjee, T. A. Henzinger, and F. Horn. Stochastic games with finitary objectives. In *MFCS*, LNCS 5734, pages 34–54. Springer, 2009.

[19] K. Chatterjee, Z. Komárková, and J. Kretínský. Unifying two views on multiple mean-payoff objectives in Markov Decision Processes. In *LICS 2015*, pages 244–256, 2015.

[20] K. Chatterjee, R. Majumdar, and T. A. Henzinger. Markov Decision Processes with multiple objectives. In *STACS*, pages 325–336, 2006.

[21] K. Chatterjee, A. Pavlogiannis, A. Kößler, and U. Schmid. A framework for automated competitive analysis of on-line scheduling of firm-deadline tasks. In *RTSS*, pages 118–127. IEEE, 2014.

[22] K. Chatterjee, M. Randour, and J.-F. Raskin. Strategy synthesis for multi-dimensional quantitative objectives. *Acta Inf.*, 51:129–163, 2014.

[23] K. Chatterjee and Y. Velner. Hyperplane separation technique for multi-dimensional mean-payoff games. In *CONCUR*, pages 500–515, 2013.

[24] T. Chen, V. Forejt, M. Z. Kwiatkowska, A. Simaitis, and C. Wiltsche. On stochastic games with multiple objectives. In *MFCS*, pages 266–277, 2013.

[25] L. Clemente and J.-F. Raskin. Multidimensional beyond worst-case and almost-sure problems for mean-payoff objectives. In *Proc. of LICS: Logic in Computer Science*, pages 257–268. IEEE, 2015.

[26] A. Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992.

[27] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, 1995.

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

[29] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Int. Journal of Game Theory*, 8(2):109–113, 1979.

[30] K. Etessami, M. Z. Kwiatkowska, M. Y. Vardi, and M. Yannakakis. Multi-objective model checking of Markov decision processes. *Logical Methods in Computer Science*, 4(4), 2008.

[31] J. Filar and K. Vrieze. *Competitive Markov Decision Processes.* Springer, 1997.

[32] V. Forejt, M. Z. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Quantitative multi-objective verification for probabilistic systems. In *TACAS*, pages 112–127, 2011.

[33] D. Gillette. Stochastic games with zero stop probability. *Contributions to the Theory of Games*, 3:179–187, 1957.

[34] H. Gimbert and F. Horn. Solving simple stochastic tail games. In *Proc. of SODA*, pages 847–862. SIAM, 2010.

[35] H. Gimbert and E. Kelmendi. Two-player perfect-information shift-invariant submixing stochastic games are half-positional. *CoRR*, abs/1401.6575, 2014.

[36] H. Gimbert and W. Zielonka. Games where you can play optimally without any memory. In *CONCUR*, LNCS 3653, pages 428–442. Springer, 2005.

[37] M. Jurdzinski, R. Lazic, and S. Schmitz. Fixed-dimensional energy games are in pseudo-polynomial time. In *ICALP*, pages 260–272, 2015.

[38] E. Kopczynski. Half-positional determinacy of infinite games. In *ICALP (2)*, LNCS 4052, pages 336–347. Springer, 2006.

[39] S. R. Kosaraju and G. F. Sullivan. Detecting cycles in dynamic graphs in polynomial time (preliminary version). In *STOC*, pages 398–406. ACM, 1988.

[40] O. Kupferman and M. Y. Vardi. Safraless decision procedures. In *FOCS*, pages 531–542. IEEE Computer Society Press, 2005.

[41] T. A. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoff. *Siam Review*, 11:604–607, 1969.

[42] D. A. Martin. The determinacy of Blackwell games. *J. Symb. Log.*, 63(4):1565–1581, 1998.

[43] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. of POPL*, pages 179–190. ACM Press, 1989.

[44] M. L. Puterman. *Markov Decision Processes.* J. Wiley & Sons, 1994.

[45] T. Raghavan and J. Filar. Algorithms for stochastic games — a survey. *ZOR — Methods and Models of Oper. Research*, 35:437–472, 1991.

[46] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.

[47] Y. Velner. Finite-memory strategy synthesis for robust multidimensional mean-payoff objectives. In *CSL-LICS 2014*, pages 79:1–79:10, 2014.

[48] Y. Velner, K. Chatterjee, L. Doyen, T. A. Henzinger, A. Rabinovich, and J.-F. Raskin. The complexity of multi-mean-payoff and multi-energy games. *Information and Computation*, 241:177–196, 2015.