Stochastic Games with Synchronizing Objectives

Laurent Doyen
CNRS & LMF, ENS Paris-Saclay
Gif-sur-Yvette, France
doyen@lsv.fr

Abstract

We consider two-player stochastic games played on a finite graph for infinitely many rounds. Stochastic games generalize both Markov decision processes (MDP) by adding an adversary player, and two-player deterministic games by adding stochasticity. The outcome of the game is a sequence of distributions over the states of the game graph. We consider synchronizing objectives, which require the probability mass to accumulate in a set of target states, either always, once, infinitely often, or always after some point in the outcome sequence; and the winning modes of sure winning (if the accumulated probability is equal to 1) and almost-sure winning (if the accumulated probability is arbitrarily close to 1).

We present algorithms to compute the set of winning distributions for each of these synchronizing modes, showing that the corresponding decision problem is PSPACE-complete for synchronizing once and infinitely often, and PTIME-complete for synchronizing always and always after some point. These bounds are remarkably in line with the special case of MDPs, while the algorithmic solution and proof technique are considerably more involved, even for deterministic games. This is because those games have a flavour of imperfect information, in particular they are not determined and randomized strategies need to be considered, even if there is no stochastic choice in the game graph. Moreover, in combination with stochasticity in the game graph, finite-memory strategies are not sufficient in general (for synchronizing infinitely often).

CCS Concepts: • **Mathematics of computing** → **Stochastic processes**; • **Theory of computation** → *Algorithmic game theory*; *Logic and verification.*

Keywords: Stochastic games, Distributions, Synchronization, Complexity

LICS '22, August 2–5, 2022, Be'er Sheva, Israel

ACM ISBN 978-1-4503-9351-5/22/08…$15.00
https://doi.org/10.1145/3531130.3532439

ACM Reference Format:
Laurent Doyen. 2022. **Stochastic Games with Synchronizing Objectives**. In *37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS '22), August 2–5, 2022, Be'er Sheva, Israel.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3531130.3532439

1 Introduction

Stochastic games are a central model to solve synthesis problems for reactive systems [7, 12], which consist of a nonterminating finite-state program receiving input from an arbitrary, possibly stochastic, environment. The goal of synthesis is to construct a program that satisfies with the largest possible probability a given logical specification regardless of the behaviour of the environment.

Synthesis naturally reduces to solving a two-player stochastic game on a graph, where the logical specification defines the objective of the game, as a language of infinite words, representing the set of infinite paths through the graph that are winning for one player. A wealth of results are known for stochastic games with *perfect information*, where the players are fully informed about the state of the game graph [11], such as Martin's determinacy result and the existence of pure (non-randomized) ε-optimal strategies for Borel objectives [26], as well as decidability for ω-regular objectives, see Chatterjee and Henzinger's survey [11] for details and references.

The assumption of perfect information is not realistic in systems consisting of several components where each component has no access to the internal state of the other components. Models of games with *imperfect information* are notoriously more complicated to solve [31], and, combined with the probabilistic and adversarial aspects of stochastic games in general lead to undecidability, even for the simple class of reachability objectives. For instance, distributed games are undecidable, even without stochasticity [30, 32], and partial-observation games are undecidable, even without adversary, for quantitative analysis of finitary objectives [25, 29] and for qualitative analysis of infinitary objectives [3]; randomized strategies are more powerful than pure strategies [10], and determinacy no longer holds [6].

Recent works proposed new decidable models with a flavour of imperfect information, for the control of a large population of identical processes, modeled as a finite-state machine. The global state of the game is a distribution over the local states of the processes, and the specification describes which sequences of distributions are winning. The

distributions can be discrete [1, 13] or continuous [2, 24]. The control may be applied uniformly, independently of the local state of each process, as in non-deterministic [5], and probabilistic automata [13], or it may depend on the local history of states, as in Markov decision processes (MDPs) [2, 18]. In both cases imperfect information arises: either because the control is global, thus not aware of the local state of individual processes, or because the control is local, thus not aware of the global states on which the specification is defined.

In this paper, we consider the control problem for a continuous population of processes modeled as a stochastic game with local control, and objective defined by finitary and infinitary synchronization properties [5, 13, 18]. Informally, synchronization happens in a sequence of distributions when (almost) all processes are synchronously in a set of designated target states, that is when, either for $\varepsilon = 0$, or for all $\varepsilon > 0$, there is a distribution in the sequence where the probability mass in the target states is at least $1 - \varepsilon$. We consider finitary synchronization objectives where synchronization should happen once or forever along the sequence of distributions, called respectively *eventually* and *always* synchronizing; and infinitary objectives where synchronization should happen infinitely often or eventually forever, called respectively *weakly* and *strongly* synchronizing [18]. We distinguish the *sure* winning mode for $\varepsilon = 0$, and the *almost-sure* winning mode for $\varepsilon \to 0$ (where the synchronization objective must be satisfied for all $\varepsilon > 0$).

The most interesting and challenging objectives are eventually and weakly synchronizing, analogous to reachability and Büchi objectives. For those objectives, it is known that finite memory is not sufficient for almost-sure winning, already in MDPs [18], and determinacy does not hold. Therefore, both the construction of a winning strategy (to show that player 1 is almost-sure winning), and the construction of a spoiling strategy for the adversary (to show that player 1 is not almost-sure winning) are non trivial. In particular, the traditional approach of constructing a winning strategy for player 2 for the complement of the objective to obtain a spoiling strategy cannot work. The construction of a spoiling strategy must be carried out after fixing an arbitrary infinite-memory strategy for player 1, which is a substantial complication. This is the main technical challenge to prove the correctness of our algorithm. We show that the control problem for eventually and weakly synchronizing is PSPACE-complete. For always and strongly synchronizing, a simple reduction to traditional safety and coBüchi stochastic games induces a polynomial-time solution. Omitted proofs and additional material can be found in an extended version of this paper [17].

Applications and Related Works. The main interest of this contribution lies in the combination of adversarial, stochastic, and infinitary aspects with a flavour of imperfect information in a decidable model. The works on (discrete) parameterized control considered finitary synchronization objectives (reachability of a synchronized distribution), either with an adversary [5], or with stochasticity [13], but not with both. With continuous distributions, the central model that has been studied is MDPs, thus with stochasticity but no adversary, either for finitary [2] or infinitary objectives [1, 18]. The solution of the control problem studied in this paper is known for MDPs [18].

Like in all the above previous works, the main limitation of this population model is the absence of communication between the processes. While communication plays a central role in distributed programming applications [20], self-organization and coordinated behaviour can emerge from large crowds of individuals with limited sensing ability, without signaling, and without centralized control [14, 15]. The highly developed local control necessary to achieve a complex collective behaviour may emerge naturally [21] or be engineered [27].

The line of work followed in this paper can also be viewed as an attempt to propose decidable models that are still rich enough to describe interesting natural phenomena. Many systems in natural computing exhibit several instances of the same anonymous process (without pre-defined identity or hierarchy), from particle physics to flock of birds. Examples of biological systems such as yeast [2, 5], and simple chemical systems [24] illustrate the synthesis applications of this model. The same principle underlies synthetic biology where a local control program is executed in every instance of the process [19, 28]. In more complex systems, the computational mechanisms behind local decision-making towards global behaviours have multiple origins that require more sophisticated computational models [15].

2 Definitions

A *probability distribution* on a finite set S is a function $d : S \to [0, 1]$ such that $\sum_{s \in S} d(s) = 1$. The *support* of d is the set $\mathrm{Supp}(d) = \{s \in S \mid d(s) > 0\}$. We denote by $\mathcal{D}(S)$ the set of all probability distributions on S.

Given a set $T \subseteq S$, let $d(T) = \sum_{s \in T} d(s)$. For $T \neq \emptyset$, the *uniform distribution* on T assigns probability $\frac{1}{|T|}$ to every state in T. Given $s \in S$, we denote by 1_s the *Dirac distribution* on s that assigns probability 1 to s (which we often identify with s).

Stochastic games. A *two-player stochastic game* (or simply, a game) $\mathcal{G} = \langle Q, \mathsf{A}, \delta \rangle$ consists of a finite set Q of states, a finite nonempty set A of actions, and a probabilistic transition function $\delta : Q \times \mathsf{A} \times \mathsf{A} \to \mathcal{D}(Q)$. We typically denote by $n = |Q|$ the size of the state space, and by η the smallest positive probability in the transitions of $\mathcal{G}$.

From an initial state $q_0 \in Q$, the game is played in (infinitely many) rounds as follows. Each round starts in a state $q_i \in Q$, the first round starts in the initial state q_0. In each round, player 1 chooses an action $a \in \mathsf{A}$, and then given a, player 2 chooses an action $b \in \mathsf{A}$. Given the state q_i in which the round started, the next round starts in q_{i+1} with probability $\delta(q_i, a, b)(q_{i+1})$. Note that the game is turn-based as player 2 sees the action chosen by player 1 before playing.

A state q is a *player*-1 *state* if $\delta(q, a, b) = \delta(q, a, b')$ for all $a, b, b' \in \mathsf{A}$, and it is a *player*-2 *state* if $\delta(q, a, b) = \delta(q, a', b)$ for all $a, a', b \in \mathsf{A}$. We write $\delta(q, a, -)$ or $\delta(q, -, b)$ to emphasize and recall that q is a player-1 or player-2 state. In figures, player-1 states are shown as circles, player-2 states as boxes. The value of the transition probabilities are not shown on figures, but diamonds represent the probabilistic choices (the main results of this paper are independent of the exact value of transition probabilities).

Classical special cases of stochastic games include Markov decision processes (MDPs), also called one-player stochastic games, where all states are player-1 states; adversarial MDPs where all states are player-2 states; and deterministic games where $\delta(q, a, b)$ is a Dirac distribution for all $q \in Q$ and all $a, b \in \mathsf{A}$.

A *play* in $\mathcal{G}$ is an infinite sequence $\pi = q_0\, a_0 b_0\, q_1 a_1 b_1\, q_2 \ldots \in (QAA)^\omega$ such that $\delta(q_i, a_i, b_i)(q_{i+1}) > 0$ for all $i \geq 0$. The prefix $q_0\, a_0 b_0\, q_1 \ldots q_k$ of the play π is denoted by $\pi(k)$, its length is $|\pi(k)| = k$ and its last element is $\mathsf{Last}(\pi(k)) = q_k$. The set of all plays in $\mathcal{G}$ is denoted by $\mathsf{Play}(\mathcal{G})$, and the set of corresponding finite prefixes (or histories) is denoted by $\mathsf{Pref}(\mathcal{G})$.

Strategies. A *strategy* for player 1 in $\mathcal{G}$ is a function $\sigma : \mathsf{Pref}(\mathcal{G}) \to \mathcal{D}(\mathsf{A})$, and for player 2 it is a function $\tau : \mathsf{Pref}(\mathcal{G}) \times \mathsf{A} \to \mathcal{D}(\mathsf{A})$. We denote by Σ, and Θ, the sets of all player-1 strategies, and all player-2 strategies, respectively. A strategy σ for player 1 is *pure* if $\sigma(\rho)$ is a Dirac distribution for all $\rho \in \mathsf{Pref}(\mathcal{G})$; it is *counting* if $|\rho| = |\rho'|$ and $\mathsf{Last}(\rho) = \mathsf{Last}(\rho')$ implies $\sigma(\rho) = \sigma(\rho')$ for all $\rho, \rho' \in \mathsf{Pref}(\mathcal{G})$; and it is *memoryless* if $\mathsf{Last}(\rho) = \mathsf{Last}(\rho')$ implies $\sigma(\rho) = \sigma(\rho')$ for all $\rho, \rho' \in \mathsf{Pref}(\mathcal{G})$. We view deterministic strategies for player 1 as functions $\sigma : \mathsf{Pref}(\mathcal{G}) \to \mathsf{A}$, and counting strategies as functions $\sigma : \mathbb{N} \times Q \to \mathcal{D}(\mathsf{A})$.

A strategy σ (for player 1) uses *finite memory* if there exists a right congruence $\approx$ of finite index (i.e., that can be generated by a finite-state transducer) over $\mathsf{Pref}(\mathcal{G})$ such that $\rho \approx \rho'$ implies $\sigma(\rho) = \sigma(\rho')$. We omit analogous definitions of pure, counting, memoryless, and finite-memory strategies for player 2.

State-based objectives. The traditional view is to consider the semantics of probabilistic systems as a probability distribution over sequences (of interleaved states and actions), i.e., over plays.

We denote by $\Pr_{d_0}^{\sigma,\tau}$ the standard probability measure on the sigma-algebra over the set of (infinite) plays, generated by the cylinder sets spanned by the (finite) prefixes of plays [4]. Given a prefix $\rho = q_0\, a_0 b_0\, q_1 \ldots q_k$, the cylinder set $\mathrm{Cyl}(\rho) = \{\pi \in \mathsf{Play}(\mathcal{G}) \mid \pi(k) = \rho\}$ has probability:

$$\Pr_{d_0}^{\sigma,\tau}(\mathrm{Cyl}(\rho)) = d_0(q_0) \cdot \prod_{i=0}^{k-1} \sigma(\rho(i))(a_i) \cdot \tau(\rho(i), a_i)(b_i) \cdot \delta(q_i, a_i, b_i)(q_{i+1}).$$

We say that ρ is *compatible* with σ (from d_0) if $\Pr_{d_0}^{\sigma,\tau}(\mathrm{Cyl}(\rho)) > 0$ for some player-2 strategy τ.

State-based objectives, in this traditional semantics, are sets of plays. We consider the following state-based objectives, expressed by LTL formulas [4] where $T \subseteq Q$ is a set of target states: the reachability and safety objectives $\Diamond T$ and $\Box T$, their bounded variants $\Diamond^{=k} T$, $\Diamond^{\leq k} T$, and $\Box^{\leq k} T$ (where $k \in \mathbb{N}$), and the coBüchi objective $\Diamond\Box T$. As each of the above objectives φ is a measurable set, the probability $\Pr_{d_0}^{\sigma,\tau}(\varphi)$ that φ is satisfied along a play with initial distribution d_0 and strategies σ for player 1 and τ for player 2 is well defined [34]. In particular, we say that player 1 is *almost-sure* winning from an initial distribution d_0 for a state-based objective φ if he has a strategy to win with probability 1, that is $\exists \sigma \in \Sigma \cdot \forall \tau \in \Theta : \Pr_{d_0}^{\sigma,\tau}(\varphi) = 1$.

Distribution-based objectives. An alternative view is to consider probabilistic systems as generators of sequences of probability distributions (over states) [24]. We denote by $\mathcal{G}_{d_0}^{\sigma,\tau}$ the *outcome sequence* $d_0, d_1, \ldots$ where $d_i \in \mathcal{D}(Q)$ is, intuitively, the probability distribution over states after i rounds defined, for all $q \in Q$, by:

$$d_i(q) = \Pr_{d_0}^{\sigma,\tau}(\Diamond^{=i}\{q\}) = \sum_{\substack{\rho \in \mathsf{Pref}(\mathcal{G}) \\ |\rho| = i \\ \mathsf{Last}(\rho) = q}} \Pr_{d_0}^{\sigma,\tau}(\mathrm{Cyl}(\rho)).$$

For a Dirac distribution $d_0 = 1_q$, we often write $\mathcal{G}_q^{\sigma,\tau}$ instead of $\mathcal{G}_{1_q}^{\sigma,\tau}$. We also sometimes omit the subscript d_0 when the initial distribution is clear from the context. We denote by $\mathcal{G}_{d_0}^{\sigma,\tau}(T)$ the sequence of numbers $d_0(T), d_1(T), \ldots$.

Distribution-based objectives, in this alternative semantics, are sets of infinite sequences of distributions over states. In particular, given a set $T \subseteq Q$ of target states, *synchronizing objectives* informally require that the probability mass in T tends to 1 (or is equal to 1) in a sequence $(d_k)_{k \in \mathbb{N}}$, in either all, some, infinitely many, or all but finitely many positions [18]. For $0 \leq \varepsilon \leq 1$, we say that a sequence $\bar{d} = d_0 d_1 \ldots$ of probability distributions is, always, eventually, weakly, or strongly $(1-\varepsilon)$-synchronizing in T if $d_i(T) \geq 1 - \varepsilon$, respectively, for all $i \geq 0$, for some $i \geq 0$, for infinitely many i's, or for all but finitely many i's.

For each synchronizing mode $\lambda \in \{always, event, weakly, strongly\}$, we consider *winning modes* that require

either that ε equals 0 (sure winning mode), or that ε tends to 0 (almost-sure winning mode).

We say that player 1 is:

- *sure* winning for a synchronizing mode λ in T from an initial distribution d_0 if he has a strategy to ensure 1-synchronizing in T, or $\exists\sigma \in \Sigma \cdot \forall\tau \in \Theta : \mathcal{G}_{d_0}^{\sigma,\tau}$ is 1-synchronizing in T in mode λ.
- *almost-sure* winning for a synchronizing mode λ in T from an initial distribution d_0 if he has a strategy to ensure $(1-\varepsilon)$-synchronizing in T for all $\varepsilon > 0$, or $\exists\sigma \in \Sigma \cdot \forall\tau \in \Theta \cdot \forall\varepsilon > 0 : \mathcal{G}_{d_0}^{\sigma,\tau}$ is $(1-\varepsilon)$-synchronizing in T in mode λ.

We denote by $\langle\langle 1\rangle\rangle^{\lambda}_{sure}(\mathcal{G}, T)$ (or simply $\langle\langle 1\rangle\rangle^{\lambda}_{sure}(T)$ when the game $\mathcal{G}$ is clear from the context) the set of distributions d from which player 1 is sure winning for synchronizing mode λ in T; we define analogously the set $\langle\langle 1\rangle\rangle^{\lambda}_{almost}(\mathcal{G}, T)$, and we say that player 1 is (sure or almost-sure) winning from d, or that d is (sure or almost-sure) winning. If $d \notin \langle\langle 1\rangle\rangle^{\lambda}_{almost}(\mathcal{G}, T)$, we say that player 2 can *spoil* player 1 from d for almost-sure synchronizing in mode λ.

It immediately follows from the definitions that for all $\lambda \in \{always, event, weakly, strongly\}$, and for all $\mu \in \{sure, almost\}$:

- $\langle\langle 1\rangle\rangle^{always}_{\mu}(T) \subseteq \langle\langle 1\rangle\rangle^{strongly}_{\mu}(T) \subseteq \langle\langle 1\rangle\rangle^{weakly}_{\mu}(T) \subseteq \langle\langle 1\rangle\rangle^{event}_{\mu}(T)$, and
- $\langle\langle 1\rangle\rangle^{\lambda}_{sure}(T) \subseteq \langle\langle 1\rangle\rangle^{\lambda}_{almost}(T)$.

In general, these inclusions cannot be strengthened to equality even for MDPs [18], except for always synchronizing where we show that $\langle\langle 1\rangle\rangle^{always}_{sure}(T) = \langle\langle 1\rangle\rangle^{always}_{almost}(T)$ holds in stochastic games.

We are interested in computing the sets $\langle\langle 1\rangle\rangle^{\lambda}_{sure}(\mathcal{G}, T)$ and $\langle\langle 1\rangle\rangle^{\lambda}_{almost}(\mathcal{G}, T)$ for the four synchronizing modes $\lambda \in \{always, event, weakly, strongly\}$, which we generically call *winning regions*. It is sufficient to have an algorithm that computes the set of states q such that 1_q is winning: to know if a distribution d is winning, consider a new state q_d with stochastic transitions $\delta(q_d, a, b)(q) = d(q)$ for all $q \in Q$, and all $a, b \in \mathsf{A}$. We consider the *membership problem*, which is to decide, given a game $\mathcal{G}$, a set T, and a state q, whether $1_q \in \langle\langle 1\rangle\rangle^{\lambda}_{sure}(\mathcal{G}, T)$ (resp., whether $1_q \in \langle\langle 1\rangle\rangle^{\lambda}_{almost}(\mathcal{G}, T)$).

Attractors and subgames. Let $\mathsf{CPre} : 2^Q \to 2^Q$ be the *controllable predecessor operator* defined for all $s \subseteq Q$ by $\mathsf{CPre}(s) = \{q \in Q \mid \exists a \in \mathsf{A} \cdot \forall b \in \mathsf{A} : \mathsf{Supp}(\delta(q, a, b)) \subseteq s\}$. Intuitively, $\mathsf{CPre}(s)$ is the set of states from which player 1 can ensure to be in s after one round, regardless of the action chosen by player 2 and of the outcome of the probabilistic transition.

For a set $T \subseteq Q$, the *attractor* $\mathsf{Attr}(T, \mathcal{G})$ is the least fixed point of the operator $x \mapsto \mathsf{CPre}(x) \cup T$, that is $\mathsf{Attr}(T, \mathcal{G}) = \bigcup_{i \geq 0} \mathsf{CPre}^i(T)$ (where $\mathsf{CPre}^0(T) = T$). It is the set of states in $\mathcal{G}$ from which player 1 has a (pure memoryless) strategy to ensure eventually reaching T [33]. We refer to such a memoryless strategy as an attractor strategy.

Let $\mathsf{PosPre}_1 : 2^Q \to 2^Q$ be the *positive predecessor operator* for player 1 defined for all $s \subseteq Q$ by $\mathsf{PosPre}_1(s) = \{q \in Q \mid \exists a \in \mathsf{A} \cdot \forall b \in \mathsf{A} : \mathsf{Supp}(\delta(q, a, b)) \cap s \neq \emptyset\}$, and let $\mathsf{PosPre}_2 : 2^Q \to 2^Q$ be defined symmetrically by $\mathsf{PosPre}_2(s) = \{q \in Q \mid \forall a \in \mathsf{A} \cdot \exists b \in \mathsf{A} : \mathsf{Supp}(\delta(q, a, b)) \cap s \neq \emptyset\}$.

For a set $T \subseteq Q$, the *positive attractor* $\mathsf{PosAttr}_i(T, \mathcal{G})$ for player i ($i = 1, 2$) is the least fixed point of the operator $x \mapsto \mathsf{PosPre}_i(x) \cup T$. There exists a pure memoryless strategy for player i (referred to as *positive-attractor* strategy) to ensure, regardless of the strategy for player $3-i$ that from all states in $\mathsf{PosAttr}_i(T, \mathcal{G})$, the set T is reached within $n = |Q|$ steps with positive probability (in fact, bounded probability, at least η^n where η is the smallest positive probability in the transitions of $\mathcal{G}$).

A set $S \subseteq Q$ induces a *subgame* of $\mathcal{G}$ if for all $q \in S$, there exist $a_q, b_q \in \mathsf{A}$ such that $\delta(q, a_q, b_q)(S) = 1$. We denote by $\mathcal{G} \restriction [S] = \langle S, \mathsf{A}, \delta_S \rangle$ the subgame induced by S, where for all $q \in S$ and $a \in \mathsf{A}$, if $\delta(q, a, b_a)(S) = 1$ for some $b_a \in \mathsf{A}$ (note that this condition holds for $a = a_q$), then for all $b \in \mathsf{A}$ we define $\delta_S(q, a, b) = \delta(q, a, b)$ if $\delta(q, a, b)(S) = 1$, and $\delta_S(q, a, b) = \delta(q, a, b_a)$ if $\delta(q, a, b)(S) < 1$; otherwise $\delta(q, a, b)(S) < 1$ for all $b \in \mathsf{A}$, and then we define $\delta_S(q, a, b) = \delta_S(q, a_q, b)$ for all $b \in \mathsf{A}$. We use this definition of subgame to keep the same alphabet of actions in every state. For instance, the set $S = Q \setminus \mathsf{PosAttr}_i(T, \mathcal{G})$ (for $i = 1, 2$) induces a subgame of $\mathcal{G}$.

A set $S \subseteq Q$ is a *trap* for player 1 in $\mathcal{G}$ if for all states $q \in S$ and all actions $a \in \mathsf{A}$, there exists $b \in \mathsf{A}$ such that $\delta(q, a, b)(S) = 1$. Intuitively, player 2 has a strategy to keep player 1 trapped in S. Dually, the set S is a *trap* for player 2 in $\mathcal{G}$ if for all states $q \in S$, there exists $a \in \mathsf{A}$ such that for all $b \in \mathsf{A}$ we have $\delta(q, a, b)(S) = 1$. Note that S is a trap for player 2 if and only if $S \subseteq \mathsf{CPre}(S)$. A key property of traps is that if player i has no winning strategy for some state-based objective from a state q in a trap S for player i, in the subgame $\mathcal{G} \restriction [S]$, then for the same objective in $\mathcal{G}$ player i has no winning strategy from q.

For deterministic games, the operators CPre and PosPre_1 coincide, as well as the attractor $\mathsf{Attr}(T, \mathcal{G})$ and positive attractor $\mathsf{PosAttr}_1(T, \mathcal{G})$, and therefore the set $U = Q \setminus \mathsf{Attr}(T, \mathcal{G})$ induces a subgame of $\mathcal{G}$.

We recall basic properties derived from the definitions of the positive attractor, and the analysis of stochastic games with almost-sure reachability objective [9, 16].

Lemma 1. *If a distribution d_0 is almost-sure winning for a reachability objective $\Diamond T$ in a game $\mathcal{G}$, then there exists a memoryless player-1 strategy σ such that for all $\varepsilon > 0$, there exists an integer h_ε such that for all player-2 strategies σ, $\Pr_{d_0}^{\sigma,\tau}(\Diamond^{\leq h_\varepsilon} T) \geq 1 - \varepsilon$.*

Lemma 2. *If a distribution d_0 is not almost-sure winning for a reachability objective $\Diamond T$ in a game $\mathcal{G}$, then there exists a memoryless player-2 strategy τ such that for all player-1 strategies σ, for all $i \geq 0$ we have $\Pr_{d_0}^{\sigma,\tau}(\Diamond^{=i} T) \leq 1 - \eta_0 \cdot \eta^n$ where $\eta_0 = \min\{d_0(q) \mid q \in \mathrm{Supp}(d_0)\}$ is the smallest positive probability in the initial distribution d_0.*

In Lemma 2 it is crucial to notice that the bound $\eta \cdot \eta^n$ is independent of the number i of steps.

Strongly connected component. In a directed graph $\langle V, E\rangle$, a *strongly connected component (SCC)* is a nonempty set $s \subseteq V$ such that for all $v, v' \in s$, there exists a nonempty path from v to v'. Our definition excludes singletons $\{v\}$ to be an SCC if there is no self-loop (v, v) in E. The *period* of an SCC is the greatest common divisor of the lengths of all its cycles.

3 Sure Synchronizing

In the rest of this paper, when the initial distribution d_0 is irrelevant or clear from the context, we denote the i-th element d_i in the sequence $\mathcal{G}_{d_0}^{\sigma,\tau} = d_0, d_1, \ldots$ by $\mathcal{G}_i^{\sigma,\tau}$.

For sure winning, only the support of distributions is important (not the exact value of probabilities). Intuitively for player 1, the worst case that can happen is that player 2 uses the *uniform strategy* τ_{u} that plays all actions uniformly at random, in order to scatter the probability mass in as many states as possible, where $\tau_{\mathrm{u}}(\rho)(a) = \frac{1}{|\mathsf{A}|}$ for all $\rho \in \mathrm{Pref}(\mathcal{G})$ and $a \in \mathsf{A}$. Formally, given a set $T \subseteq Q$ and an arbitrary player-1 strategy $\sigma \in \Sigma$ it is easy to show that $\mathrm{Supp}(\mathcal{G}_i^{\sigma,\tau}(T)) \subseteq \mathrm{Supp}(\mathcal{G}_i^{\sigma,\tau_{\mathrm{u}}}(T))$, for all player-2 strategies $\tau \in \Theta$ and all $i \geq 0$.

Therefore, in all synchronizing modes, player 1 is sure winning if and only if player 1 is sure winning against the uniform strategy τ_{u} for player 2, and computing the sure winning distributions in stochastic games reduces to the same problem in MDPs (obtained by fixing τ_{u} in $\mathcal{G}$), which is known [18]. We immediately derive the following results.

Theorem 1. *The membership problem for sure always and strongly synchronizing can be solved in polynomial time, and pure memoryless strategies are sufficient for player* 1.

The membership problem for sure eventually and weakly synchronizing is PSPACE-complete, and pure strategies with exponential memory are sufficient (and may be necessary) for player 1.

Note that when the uniform strategy τ_{u} is fixed, the controllable predecessor operator CPre coincides with the predecessor operator used to solve the membership problem for MDPs [18].

Also note that even in the case of deterministic games, the winning regions for sure and almost-sure winning do not coincide, for eventually and weakly synchronizing, as illustrated by the game $\mathcal{G}_{\mathsf{win}}$ (Figure 1a). The state q_1 is almost-sure winning (as we show in the beginning of Section 4.1), but not sure winning for weakly synchronizing in $T = \{q_1, q_3\}$ (e.g., against the uniform strategy for player 2). We show in Section 4.3 that in deterministic games, the winning regions for sure and almost-sure winning do coincide for always and strongly synchronizing (even for all stochastic games in the case of always synchronizing).

4 Almost-Sure Synchronizing

We first consider almost-sure weakly synchronizing, which is the most interesting and challenging case. We present an algorithm to compute the set $\langle\langle 1\rangle\rangle_{almost}^{weakly}(T)$ and we show that pure counting strategies are sufficient for player 1.

4.1 Weakly synchronizing in deterministic games

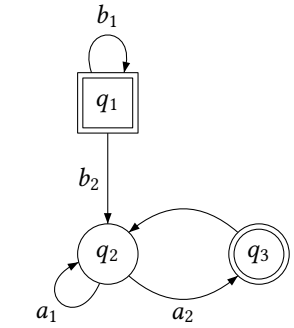
The key ideas of the algorithm are easier to present in the special case of deterministic games, and with the assumption that pure counting strategies are sufficient for player 1 (but player 2 is allowed to use an arbitrary strategy). We show in Section 4.2 how to compute the winning region for almost-sure weakly synchronizing in general stochastic games, and without any assumption on the strategies of player 1. It will follow from our results that pure counting strategies are in fact sufficient for player 1.

Given a deterministic game $\mathcal{G} = \langle Q, \mathsf{A}, \delta\rangle$, a *selector* is a function $\alpha : Q \to \mathsf{A}$, and for a set $s \subseteq Q$, let $\delta_\alpha(s) = \{\delta(q, \alpha(q), b) \mid q \in s \wedge b \in \mathsf{A}\} \subseteq Q$. A pure counting strategy can be viewed as an infinite sequence of selectors. The *subset construction* for $\mathcal{G}$ is the graph $\mathcal{P}(\mathcal{G}) = \langle V, E\rangle$ where $V = 2^Q \setminus \{\varnothing\}$ and $E = \{(s, \delta_\alpha(s)) \mid s \in V \wedge \alpha \text{ is a selector}\}$. Given a set $T \subseteq Q$ of target states, and a set $s \in V$, we say that s is *accepting* if $s \subseteq T$ (for singletons $\{q\}$ we simply say that q is accepting).

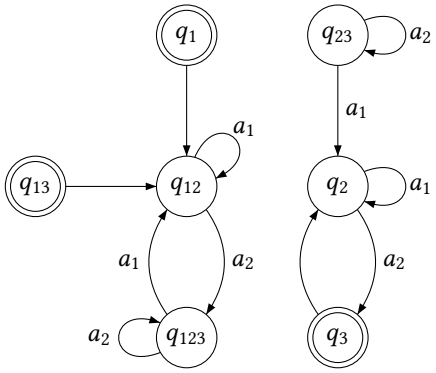
The *central property* of the subset construction $\mathcal{P}(\mathcal{G})$ is that for every sequence of selectors $\alpha_1, \alpha_2, \ldots, \alpha_k$, the sequence $s_1, s_2, \ldots, s_{k+1}$ such that $s_{i+1} = \delta_{\alpha_i}(s_i)$ for all $1 \leq i \leq k$, is a path in $\mathcal{P}(\mathcal{G})$ and that for every state $q \in s_{k+1}$, there exists a play prefix $q_1\, a_1 b_1\, q_2 \ldots q_{k+1}$ in $\mathcal{G}$ such that $q_{k+1} = q$ and $q_i \in s_i$ for all $1 \leq i \leq k$, that is compatible with the given sequence of selectors, $a_i = \alpha_i(q_i)$. The central property is easily proved by induction on k [10]. Using König's Lemma [23], the central property holds for infinite sequences, namely for every infinite path $s_1, s_2, \ldots$ in $\mathcal{P}(\mathcal{G})$, there exists an infinite play $q_1\, a_1 b_1\, q_2 \ldots$ in $\mathcal{G}$ such that $q_i \in s_i$ and $a_i = \alpha_i(q_i)$ for all $i \geq 1$. We also mention a simple monotonicity property: if $s \subseteq s'$, then $\delta_\alpha(s) \subseteq \delta_\alpha(s')$ for all selectors α.

We illustrate the key technical insights with two examples. First consider the deterministic game $\mathcal{G}_{\mathsf{win}}$ in Figure 1a, where the target set is $T = \{q_1, q_3\}$. Only state q_2 has a relevant choice for player 1, and only q_1 has a relevant choice for player 2 (for the sake of clarity, we denote by a_1, a_2 the actions of player 1, and by b_1, b_2 the actions of player 2).

Player 1 is almost-sure weakly synchronizing in T from every state. A winning strategy plays a_1 at even rounds,

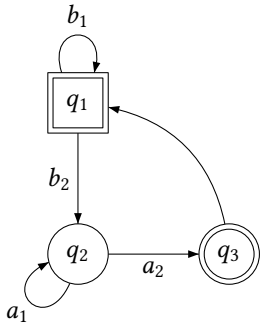


(a) $\mathcal{G}_{\mathsf{win}}$

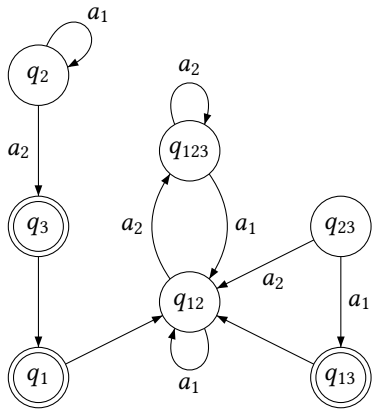


(b) Subset construction

Figure 1. The deterministic game $\mathcal{G}_{\mathsf{win}}$ where player 1 is almost-sure weakly synchronizing (from all states), and its subset construction.



(a) $\mathcal{G}_{\mathsf{lose}}$



(b) Subset construction

Figure 2. The deterministic game $\mathcal{G}_{\mathsf{lose}}$ where player 1 is not almost-sure weakly synchronizing (no matter the initial state), and its subset construction.

and a_2 at odd rounds. Note that without the self-loop on q_2, player 1 is no longer almost-sure weakly synchronizing in T from q_1 (but still from q_2 and from q_3).

Consider the subset construction in Figure 1b, obtained by considering all subsets $q_{12} = \{q_1, q_2\}$, $q_{13} = \{q_1, q_3\}$, etc. of Q, and with an edge from s to s' if there exists a selector $\alpha : Q \to \mathsf{A}$ such that $s' = \delta_\alpha(s)$. Intuitively, the selector describes the actions played by a pure counting strategy of player 1 at a given round. Figure 1b labels the edges (s, s') with the action played in q_2 by the corresponding selector in s (if relevant, i.e., if $q_2 \in s$). Accepting sets $s \subseteq T$ are marked by a double line.

An accepting strongly connected component is an SCC containing an accepting set, such as $C = \{\{q_2\}, \{q_3\}\}$ in our example. This is a witness that player 1 is almost-sure weakly synchronizing in T from all states q such that C is reachable from $\{q\}$ in $\mathcal{P}(\mathcal{G})$. This sufficient condition for almost-sure winning is not necessary, as player 1 is almost-sure winning from q_1 as well, but the set $\{q_1\}$ cannot reach an accepting SCC in $\mathcal{P}(\mathcal{G})$. However, we will show that if there is no accepting SCC in the subset construction, then there is no state from which player 1 is almost-sure weakly synchronizing in T.

This situation is illustrated in Figure 2 where the subset construction for the game $\mathcal{G}_{\mathsf{lose}}$ contains no accepting SCC, and player 1 is not almost-sure weakly synchronizing in T (no matter from which initial state). This is not trivial to see, and we present the crux of the argument below. Although player 1 is not almost-sure weakly synchronizing in T, it is not true either that player 2 can fix a strategy τ in $\mathcal{G}_{\mathsf{lose}}$ to prevent player 1 from almost-sure winning for weakly synchronizing in T. This means that in general spoiling strategies can be constructed only after a strategy for the other player has been fixed, which brings technical difficulty in the proofs.

Why player 1 *can spoil player* 2 *in* $\mathcal{G}_{\mathsf{lose}}$. Given an arbitrary strategy τ for player 2, we can construct a strategy σ for player 1 such that the outcome sequence $\mathcal{G}_{\mathsf{lose}}^{\sigma,\tau}$ (from any initial distribution) is almost-sure weakly synchronizing in T.

Consider the strategy σ_{loop} that always plays a_1 (to loop through q_2), and note that in the outcome $\mathcal{G}_{\mathsf{lose}}^{\sigma_{loop},\tau} = d_0, d_1, \ldots$ (from any initial distribution d_0), the probability

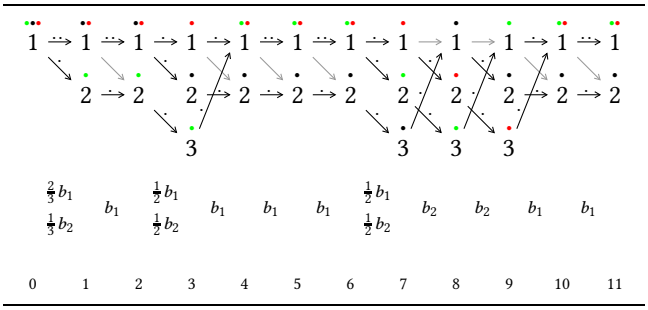


Figure 3. Construction of a spoiling strategy for player 2 (in $\mathcal{G}_{\text{lose}}$).

mass in q_2 is non-decreasing, hence $\lim_{k\to\infty} d_k(q_2)$ exists, which we denote by $\alpha(d_0)$. We construct σ to play as follows, starting with $\varepsilon = \frac{1}{2}$: (1) Given $\varepsilon > 0$ and the current distribution d, play σ_{loop} for n_ε rounds, where n_ε is such that $d_{n_\varepsilon}(q_2) \geq \alpha(d) - \varepsilon$, then (2) play a_2 in the next round, and (3) repeat from (1) with $\varepsilon := \frac{\varepsilon}{2}$. In the outcome $\mathcal{G}^{\sigma,\tau}_{\text{lose}}$, after playing a_2, the probability mass in q_2 is the probability mass transferred from q_1 in the previous step, which is at most ε. It follows that the probability mass in $T = \{q_1, q_3\}$ is at least $1 - \varepsilon$. The repetition of this pattern for $\varepsilon \to 0$ entails that $\mathcal{G}^{\sigma,\tau}_{\text{lose}}$ is almost-sure weakly synchronizing in T.

Why player 2 can spoil player 1 in $\mathcal{G}_{\text{lose}}$. We sketch the crux of the argument for initial state q_1, showing that player 2 can spoil an arbitrary strategy $\sigma : \mathbb{N} \times Q \to \mathsf{A}$ for player 1 (that is pure and counting), which is an infinite sequence of selectors and thus corresponds to an infinite path from $\{q_1\}$ in the subset construction (shown in Figure 2b).

Such an infinite path is shown in Figure 3 where an edge (s, s') labeled by a selector α is drawn as the set of edges (q, q') such that $q \in s$ and $q' = \delta(q, \alpha(q), b)$ for some $b \in \mathsf{A}$. Note that, from some point on, all sets in such a path contain a non-target state $q \in Q \setminus T$, and a spoiling strategy for player 2 must ensure a bounded probability mass in $Q \setminus T$, at every round from some point on.

In the example, player 2 can ensure a probability mass of $\frac{1}{3}$ in state $q_2 \in Q \setminus T$ from the second round on. In Figure 3, we put three tokens, each carrying a probability mass of $\frac{1}{3}$, in the initial state q_1 and we show how the three tokens can move along the edges to always maintain one token in q_2 (after the first round). The choice of which edge from q_1 is taken by a token is made by player 2 with the corresponding action b_1 or b_2 (the corresponding randomized selector is shown below the figure for each round – edges are drawn in gray if no token flows through it). It is easy to show that the pattern suggested in Figure 3 can be prolonged ad infinitum. A key insight is that player 2 should not move a token from q_1 to q_2 at every round (which may cause a depletion of tokens), but only in the rounds where player 1 sends the probability mass from q_2 to q_3.

Each token follows a play that is compatible with the strategy of player 1. The set of three plays that are followed by the three tokens has the property that in every round after round 1 at least one of the plays is in q_2. We say that the plays (or the tokens) *cover* the state q_2 from round 2 on. Note that it would be easy to cover q_2 from some round n_0 on by using plays of the form $(q_1)^n q_2 Q^\omega$ $(n = n_0, n_0 + 1, \dots)$, but this is an infinite set of plays, which would require infinitely many tokens and would not allow a positive lower bound on the probability mass of each token. The key to cover q_2 with a finite number of plays is to reuse the tokens when possible. It is however not obvious in general how to construct finitely many such plays, given an arbitrary strategy of player 1.

We show in Lemma 3 that in deterministic games, a fixed number K of tokens (each representing a probability mass $\frac{1}{K}$) is sufficient for player 2 to spoil any pure counting strategy of player 1, where $K = 2^{|Q|}$.

Lemma 3. *The following equivalence holds in deterministic games: there exists a state q from which player 1 has a pure counting strategy that is almost-sure winning for weakly synchronizing in T if and only if there exist a strongly connected component C in $\mathcal{P}(\mathcal{G})$ and a set $s \in C$ that is accepting.*

Proof. First, if there exists a strongly connected component C in $\mathcal{P}(\mathcal{G})$ and an accepting set $s \in C$, then there exists an infinite path $s_0 s_1 \dots$ in $\mathcal{P}(\mathcal{G})$ from $s_0 = s$ that visits s infinitely often. Consider the corresponding sequence of selectors $\alpha_0 \alpha_1 \dots$ (such that $s_{i+1} = \delta_{\alpha_i}(s_i)$). It is easy to see that from all states $q \in s$, the pure counting strategy σ defined by $\sigma(i, q) = \alpha_i(q)$ is sure (thus also almost-sure) winning for weakly synchronizing in $s \subseteq T$.

For the second direction, we prove the contrapositive. Assume that no SCC in $\mathcal{P}(\mathcal{G})$ contains an accepting set, and show that from all states q_0 no pure counting strategy for player 1 is almost-sure winning for weakly synchronizing in T.

Let $\sigma : \mathbb{N} \to (Q \to \mathsf{A})$ be a pure counting strategy for player 1, and we construct a spoiling strategy for player 2 from q_0. First, consider the sequence $s_0, s_1, \dots$ defined by $s_0 = \{q_0\}$ and $s_{i+1} = \delta_{\sigma(i)}(s_i)$, and as this sequence is a path in the subset construction $\mathcal{P}(\mathcal{G})$, by our assumption there exists an index i_0 such that s_i is non-accepting for all $i \geq i_0$. Let Reject $= Q \setminus T$. We have:

$$s_i \cap \text{Reject} \neq \varnothing \text{ for all } i \geq i_0.$$

By the central property of the subset construction, for every set s_i and every state $q_i \in s_i$ (in particular for $q_i \in s_i \cap$ Reject if $i \geq i_0$), there exists a play of length i from q_0 to q_i that is compatible with the strategy σ. Player 2 can use such plays to inject positive probability into q_i at every position i. However, if all those plays form an infinite set (as illustrated by the plays $(q_1)^n q_2 Q^\omega$ in the example of Figure 2a), then

player 2 may not be able to guarantee a bounded probability mass in q_i at every round i from some point on.

Now we construct a data structure that will help player 2 to determine their strategy and to construct a *finite* set Π of plays compatible with σ in $\mathcal{G}$ such that, for all $i \geq i_0$, there exists a play $\pi \in \Pi$ with $\mathsf{Last}(\pi(i)) \in \mathrm{Reject}$.

The data structure consists, for each position $i \geq i_0$, of a tuple $u_i = \langle r_1, \ldots, r_k \rangle$ of registers, each storing a nonempty subset of Q. The number of registers is not fixed (but will never decrease along the sequence $u_{i_0}, u_{i_0+1}, \ldots$), and for $i = i_0$ let $u_{i_0} = \langle r_1 \rangle$, thus u_{i_0} consists of one register, and let $r_1 = \{q_{i_0}\}$ where $q_{i_0} \in s_{i_0} \cap \mathrm{Reject}$. Each register r in u_i corresponds to one token, and stores the possible states in which the token can be after i steps by following a play compatible with σ.

Given $u_i = \langle r_1, \ldots, r_k \rangle$, define u_{i+1} as follows: first, let $r'_j = \sigma(i)(r_j)$ be the set obtained from r_j by following the selector $\sigma(i)$ at position i (which is the same selector used to define s_{i+1} from s_i). We consider two cases:

1. if all registers are accepting, that is $r'_j \cap \mathrm{Reject} = \varnothing$ for all $1 \leq j \leq k$: let $q_{i+1} \in s_{i+1} \cap \mathrm{Reject}$ and create a new register $r'_{k+1} = \{q_{i+1}\}$, define $u_{i+1} = \langle \underbrace{r'_1, \ldots, r'_k}_{\text{accepting}}, r'_{k+1} \rangle$;
2. otherwise, some register is non-accepting, and let j be the largest index such that $r'_j \cap \mathrm{Reject} \neq \varnothing$. Let $q_{i+1} \in r'_j \cap \mathrm{Reject}$ and we remove the register at position j, and replace it by a new register $r'_{k+1} = \{q_{i+1}\}$ at the end of the tuple. Define $u_{i+1} = \langle r'_1, \ldots, r'_{j-1}, \underbrace{r'_{j+1}, \ldots, r'_k}_{\text{accepting}}, r'_{k+1} \rangle$.

In both cases, we say that the parent of a register r'_l (for $l \leq k$) that occurs in u_{i+1} is the register r_l in u_i, and that r'_{k+1} has no parent (in the second case above, we say that r'_{k+1} is a clone child of r_j). An ancestor of a register r in u_i is either r or an ancestor of the parent of r (but not of the clone parent of r). In the sequel, we assume that the registers in a tuple u_i are called $r_1, r_2, \ldots$ with consecutive indices in the order they appear in u_i.

We now state key invariant properties of the sequence $u_{i_0}, u_{i_0+1}, \ldots$, for all $i \geq i_0$:

- (*consistency property*) in every u_i, for every register r in u_i we have $r \subseteq s_i$;
- (*singleton property*) in every u_i, the rightmost register is a singleton containing a non-accepting state;
- (*chain property*) for all $j \geq i$, every chain of registers (with order defined by the parent relation) from an ancestor register r in u_i to a register r' in u_j is a path in the subset construction labeled by the selectors $\sigma(i), \ldots, \sigma(j)$;
- (*key property*) in every u_i, if there are k registers at the right of a register r, then r has at least k ancestors that are accepting.

It is easy to verify that these properties hold by construction of u_{i_0}, and of u_{i+1} from u_i (by induction). In particular the key property holds because whenever a register is appended (or moved) to the right of a register r, then r is accepting.

By our initial assumption, a register r cannot have more than $2^{|Q|}$ ancestors that are accepting (using the chain property). Then it follows from the key property that a tuple u_i cannot have more than $2^{|Q|}$ registers, and since the number of registers is not decreasing, there is an index i_0^* such that all u_i, for $i \geq i_0^*$, contain the same number $K \leq 2^{|Q|}$ of registers. We are now ready to construct the set Π, containing K plays.

For each $i > i_0^*$, consider the permutation f_i on $\{1, \ldots, K\}$ that maps index j to $k = f_i(j)$ such that r_k in u_{i-1} is the parent of r_j in u_i (or clone parent if r_j has no parent).

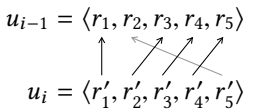


Figure 4. A permutation on registers.

Following the permutation f_i at position i (for $i > i_0^*$), we can define K equivalence classes $C_1, \ldots, C_k$ of registers that contain exactly one register from each u_i: the register r_j in the tuple u_i belongs to the class C_k with $k = (f_{i_0^*+1} \circ f_{i_0^*+2} \circ \cdots \circ f_i)(j)$, see the illustration in Figure 5. Note in particular that every rightmost register r_K (highlighted in Figure 5), which is a singleton, belongs to some class.

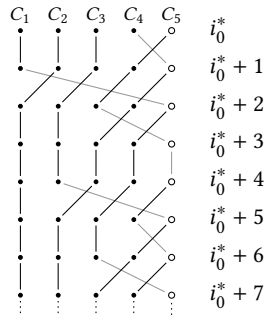


Figure 5. A sequence of permutations on registers.

Using the central property of the subset construction, from those K classes we construct K infinite plays in $\mathcal{G}$, all compatible with $\sigma(i_0^*), \sigma(i_0^* + 1), \ldots$, and such that for all $i \geq i_0^*$, for all registers r_j in u_i, one of the plays is in a

state of r_j at position i. In particular, since r_K is always a non-accepting singleton, the constructed plays cover a non-accepting state at every position $i \geq i_0^*$.

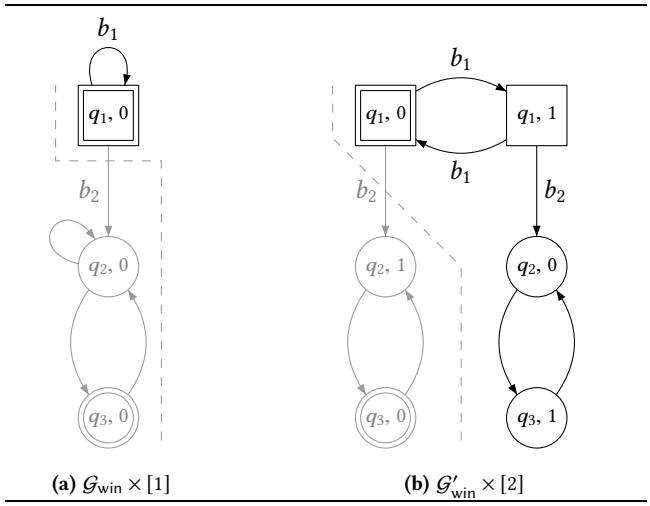
Given an equivalence class C, consider the register r of u_i in C, and the register r' of u_{i+1} in C. Then, by the central property of the subset construction, for all states $q' \in r'$, there exists a state $q \in r$ such that $q' = \delta(q, \alpha_i(q), b)$ for some action $b \in \mathsf{A}$ of player 2 (no matter whether r is the parent or clone parent of r'). Now consider all *singular* positions i where r_K (the rightmost, singleton, register) is the register of u_i in C. Between any such two positions $i_1 < i_2$, there exists a (segment of) play in $\mathcal{G}$ from the state in the register r_K at position i_1 to the state in the register r_K at position i_2, that is compatible with the strategy σ (at the corresponding positions $i_1, i_1+1, \ldots, i_2$) using the chain property. Analogously, from some state in $s_{i_0^*}$ (using the consistency property) there is a segment of play to the state in register r_K at the first singular position i_0. The constructed segments can be concatenated to form a single play, and if this play is finite, we can prolong it to an infinite play compatible with σ.

Given the K plays in $\mathcal{G}$ constructed in this way from $C_1, \ldots, C_k$, it is easy to construct a strategy for player 2 that ensures a probability mass of $\frac{1}{K}$ (a token) will move along each of the K plays (possibly using randomization), and therefore a probability mass of at least $\frac{1}{K}$ in a non-accepting state at every round $i \geq i_0^*$, showing that the strategy σ of player 1 is not almost-sure winning for weakly synchronizing in T, which concludes the proof. □

In the deterministic game $\mathcal{G}_{\mathsf{win}}$ of Figure 1a, the strongly connected component $C = \{\{q_2\}, \{q_3\}\}$ in the subset construction $\mathcal{P}(\mathcal{G}_{\mathsf{win}})$, which contains the accepting set $U = \{q_2\}$, shows that player 1 is almost-sure winning from some state (Lemma 3), namely from q_2 and from q_3. This holds even if there is no self-loop on q_2. However, whether player 1 is almost-sure winning from q_1 depends on the presence of that self-loop: with the self-loop on q_2, the period of the SCC C is $p = 1$ (see Figure 1b) and player 1 is almost-sure winning from q_1, whereas without the self-loop, the period of C is $p = 2$ and player 1 is not almost-sure winning from q_1 (player 2 can inject an equal mass of probability from q_1 to q_2 in two successive rounds, and as those masses can never merge, the probability mass in q_2 is always bounded away from 1). In fact, player 1 needs to ensure that any probability mass injected in C is always injected at the same round (modulo p), where p is the period of C.

To track the number of rounds modulo p, define the game $\mathcal{G} \times [p]$ that follows the transitions of $\mathcal{G}$ and decrements a tracking counter (modulo p) along each transition (Figure 6). Formally, let $\mathcal{G} \times [p] = \langle Q', \mathsf{A}, \delta' \rangle$ where $Q' = Q \times \{p-1, \ldots, 1, 0\}$ and δ' is defined as follows, for all $\langle q, i \rangle, \langle q', j \rangle \in Q'$ and $a \in \mathsf{A}$:

$$\delta'(\langle q, i \rangle, a)(\langle q', j \rangle) = \begin{cases} \delta(q, a)(q') & \text{if } j = i - 1 \mod p, \\ 0 & \text{otherwise.} \end{cases}$$



(a) $\mathcal{G}_{\mathsf{win}} \times [1]$ **(b)** $\mathcal{G}'_{\mathsf{win}} \times [2]$

Figure 6. Removal of positive attractor in $\mathcal{G}_{\mathsf{win}} \times [1]$ and $\mathcal{G}'_{\mathsf{win}} \times [2]$, where $\mathcal{G}'_{\mathsf{win}}$ is the variant of $\mathcal{G}_{\mathsf{win}}$ without a self-loop on q_2.

A simple property relating the game $\mathcal{G}$ with the games $\mathcal{G} \times [p]$ is that player 1 can fix (in advance, regardless of the strategy of player 2) the value of the counter when synchronization occurs in T.

Lemma 4. *Player* 1 *is almost-sure weakly synchronizing in* T *from* q *in the game* $\mathcal{G}$ *if and only if for all* $p \geq 0$*, there exists* $0 \leq i \leq p-1$ *such that player* 1 *is almost-sure weakly synchronizing in* $T \times \{0\}$ *from* $\langle q, i \rangle$ *in* $\mathcal{G} \times [p]$*.*

Given an accepting set $U \subseteq T$ that belongs to an SCC with period p in $\mathcal{P}(\mathcal{G})$, we solve the game $\mathcal{G}$ by removing from $\mathcal{G} \times [p]$ the attractor W of $U \times \{0\}$, and by recursively solving the subgame of $\mathcal{G} \times [p]$ induced by $Q \setminus W$, with target set $T \times \{0\}$.

In $\mathcal{G}_{\mathsf{win}}$, the set $U = \{q_3\}$ belongs to an SCC of period 1, and its attractor is $W = \{q_2, q_3\}$. After removal of the attractor, the subgame is winning for player 1 (Figure 6a). In the variant $\mathcal{G}'_{\mathsf{win}}$ of $\mathcal{G}_{\mathsf{win}}$ without a self-loop on q_2, the set $U = \{q_3\}$ is also in an SCC, but with period 2. After removal of the attractor to $U \times [0]$ in $\mathcal{G}'_{\mathsf{win}} \times [2]$, the subgame is not winning for player 1 (Figure 6b).

By Lemma 4, solving $\mathcal{G}$ with target set T is equivalent to solving $\mathcal{G} \times [p]$ with target set $T \times \{0\}$. However, in general combining two almost-sure winning strategies constructed in two different subgames may not give an almost-sure winning strategy (if, for instance, one strategy ensures the probability mass in $T \times \{0\}$ tends to 1 at even rounds, and the other strategy at odd rounds). To establish the correctness

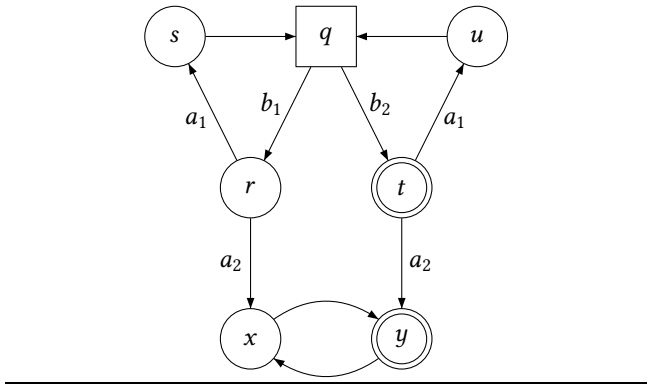


Figure 7. A deterministic game.

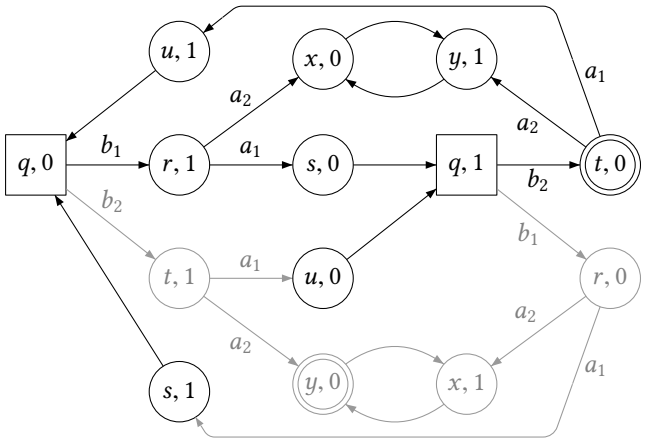


Figure 8. The game $\mathcal{H}_{sub}$ computed by Algorithm 1 (line 5) for the game of Figure 7, where the shaded region is the set W (line 4), and the set $U = \{\langle y, 0\rangle\}$ is self-recurrent.

of our solution, note that all states in $U \times \{0\}$ belong to the (controllable predecessor of the) attractor of $U \times \{0\}$ in $\mathcal{G} \times [p]$, and since the period of the SCC containing U is p, in $\mathcal{P}(\mathcal{G} \times [p])$ there is a path from $U \times \{0\}$ to itself of length $\ell = k \cdot p$ for all sufficiently large k. See the Frobenius problem [22] for questions related to computing the largest k_0 such that there is no such path of length $\ell = k_0 \cdot p$. It follows that, for $W = \mathsf{Attr}(U \times \{0\}, \mathcal{G} \times [p])$, given a state $\langle q, i\rangle \in W$ and an arbitrary length $\ell_0 = i + k \cdot p$ for $k \geq k_0$, player 1 has a strategy to get eventually synchronized in $U \times \{0\} \subseteq T \times \{0\}$ at round ℓ_0 (where the tracking counter is 0), and by the same argument at any round $\ell_1 = \ell_0 + k' \cdot p$ for $k' \geq k_0$, and so on. That is, for any sequence $i_0, i_1, \ldots$ such that $i_j - i_{j-1} \geq k_0 \cdot p$ (where $i_{-1} = 0$) and $i_j = i \mod p$ for all $j \geq 0$, player 1 has a strategy from $\langle q, i\rangle$ such that for all outcomes $d_0, d_1, \ldots$ of a strategy of player 2 in $\mathcal{G} \times [p]$, we have $\liminf_{k\to\infty} d_{i_k}(T \times \{0\}) = 1$ (thus also $\limsup_{k\to\infty} d_{i_k}(T \times \{0\}) = 1$).

Intuitively, we can choose the sequence $i_0, i_1, \ldots$ in order to synchronize the probability mass in W with the outcome of the strategy constructed in the subgame of $\mathcal{G}\times[p]$ induced by the complement of W.

Lemma 5. *Given a set U in a strongly connected component of period p in the subset construction $\mathcal{P}(\mathcal{G})$ that is accepting ($U \subseteq T$), let $W = \mathsf{Attr}(U \times \{0\}, \mathcal{G} \times [p])$ and $\mathcal{H} = \mathcal{G} \times [p] \restriction [Q \times [p] \setminus W]$. We have:*

$$\langle\!\langle 1\rangle\!\rangle^{weakly}_{almost}(\mathcal{G}, T) = \{q \in Q \mid \exists i : \langle q, i\rangle \in W \cup \langle\!\langle 1\rangle\!\rangle^{weakly}_{almost}(\mathcal{H}, T \times \{0\})\}.$$

Lemma 5 suggests a recursive procedure to compute the almost-sure winning set for weakly synchronizing objective, shown as Algorithm 1. We illustrate the execution on the example of Figure 7. First the set $U = \{y\}$ is accepting and belongs to an SCC of period $p = 2$ (line 2) in the subset construction (line 1). The game $H = \mathcal{G} \times [p]$ with tracking counter modulo $p = 2$ is shown in Figure 8, with the attractor W to $U \times \{0\} = \{\langle y, 0\rangle\}$ shaded (lines 3-4). In the induced subgame (line 5), all states except $\langle x, 0\rangle$ and $\langle y, 1\rangle$ are winning, which is found in the recursive call (line 6). It follows that all states (all Dirac distributions) are winning for player 1, and in fact all distributions that do not contain both x and y in their support are winning.

We establish the correctness and termination of Algorithm 1 as follows. The correctness straightforwardly follows from Lemma 5, and we show that the depth of the recursive calls in $Solve(\mathcal{G}_0, T)$ is bounded by the size $|Q_{\mathcal{G}_0}|$ of the state space of $\mathcal{G}_0$. This is not immediately obvious, since the size of the first argument $\mathcal{G}$ in a recursive call may increase (the game $\mathcal{H}_{sub}$ is a subgame of $\mathcal{H}$, which is p times bigger than $\mathcal{G}$). However, we claim that an invariant of the execution of $Solve(\mathcal{G}_0, T_0)$ is that, in all recursive calls $Solve(\mathcal{G}, T)$, the algorithm only needs to consider states of the first argument $\mathcal{G}$ that form a subgame (isomorphic to a subgame) of $\mathcal{G}_0 \times [k]$ for some k. This holds in the initial call (take $k = 1$), and if $\mathcal{G}$ is a subgame of $\mathcal{G}_0 \times [k]$, then the period p computed at line 2 is a multiple of k, and therefore in all states $\langle\langle q, i\rangle, j\rangle$ in $(\mathcal{G}_0 \times [k]) \times [p]$ the value $j - i \mod k$ is constant along the transitions. Given the target states $(T \times \{0\}) \times \{0\}$ we only need to consider states $\langle\langle q, i\rangle, j\rangle$ with $j - i = 0 \mod k$, and we can project $\langle\langle q, i\rangle, j\rangle$ to $\langle q, j\rangle$ without loss. It follows that $\mathcal{H}$ (and also $\mathcal{H}_{sub}$ used in the recursive call) can be viewed as a subgame of $\mathcal{G}_0 \times [p]$. Moreover, the attractor W contains at least one state for every value of the tracking counter, and therefore the size of the game $\mathcal{G}$ measured as $\max_i |\{q \in Q_{\mathcal{G}_0} \mid \langle q, i\rangle \in Q_{\mathcal{G}}\}|$ is strictly decreasing. It follows that there are at most $|Q_{\mathcal{G}_0}|$ recursive calls in $Solve(\mathcal{G}_0, T)$.

Given $0 \leq i \leq p - 1$, the *slice* at i of a set $W \subseteq Q \times [p]$ is the set $\{q \in Q \mid \langle q, i\rangle \in W\}$.

Algorithm 1: $Solve(\mathcal{G}, T)$

Input : $\mathcal{G} = \langle Q, \mathsf{A}, \delta \rangle$ is a deterministic game, $T \subseteq Q$ is a target set.

Output : The set $\{q \in Q \mid 1_q \in \langle\!\langle 1 \rangle\!\rangle^{weakly}_{almost}(\mathcal{G}, T)\}$.

begin

1 **if** *there is an SCC* C *of* $\mathcal{P}(\mathcal{G})$ *containing a set* $U \in C$ *with* $U \subseteq T$ **then**

2 $p \leftarrow$ period of C ;

3 $\mathcal{H} \leftarrow \mathcal{G} \times [p]$;

4 $W \leftarrow \mathsf{Attr}(U \times \{0\}, \mathcal{H})$;

5 $\mathcal{H}_{sub} \leftarrow \mathcal{H} \restriction [Q \times [p] \setminus W]$;

6 **return** $\{q \in Q \mid \exists i : \langle q, i \rangle \in W \cup Solve(\mathcal{H}_{sub}, T \times \{0\} \setminus W)\}$;

else

7 **return** $\varnothing$;

Lemma 6. *Algorithm 1 computes the almost-sure winning Dirac distributions for weakly synchronizing in deterministic games. It can be implemented in PSPACE.*

Theorem 2. *The membership problem for almost-sure weakly synchronizing in deterministic games is PSPACE-complete.*

4.2 Weakly synchronizing in stochastic games

We present an algorithm to compute the almost-sure winning region for weakly synchronizing objectives in stochastic games, which generalizes the result of Section 4.1. This algorithm has the flavor of the algorithm for deterministic games, with additional complications due to the probabilistic transitions in the game. The proof is also more technical because we no longer assume that pure counting strategies are sufficient for player 1 (but we show that such strategies are indeed always sufficient for almost-sure winning).

Recall that throughout this section we consider a stochastic game $\mathcal{G} = \langle Q, \mathsf{A}, \delta \rangle$ and we denote by $n = |Q|$ the size of the state space, and by η the smallest positive probability in the transitions of $\mathcal{G}$. We consider the almost-sure weakly synchronizing objective defined by a set $T \subseteq Q$ of accepting states.

Given a set $U \subseteq Q$, consider the sequence $U_i = \mathsf{CPre}^i(U)$ for $i \geq 1$ (and $U_0 = U$). Since $U_i \subseteq Q$, this sequence is ultimately periodic. Consider the least $k \geq 0$ for which there exists $r > 1$ such that $U_k = U_{k+r}$, and consider the least such r, called the *period*. It is easy to see that $k, r \leq 2^n$. For $R = U_k$ we call $\langle R, r, k \rangle$ the *periodic scheme* of U and we refer to its elements as $\mathsf{R}(U) = U_k$, $\mathsf{r}(U) = r$, and $\mathsf{k}(U) = k$. The set U is *self-recurrent* if $U \neq \varnothing$ and there exists an index $0 \leq t < r$ such that all states in $U \times \{t\}$ are almost-sure winning for the (state-based) reachability objective $\Diamond(R \times \{0\})$ in $\mathcal{G} \times [r]$. The intuitive meaning of being self-recurrent appears in Lemma 7 below. Note that in deterministic games $\mathcal{G}$, an accepting set $U \subseteq T$ contained in a strongly connected component C of $\mathcal{P}(\mathcal{G})$ is self-recurrent. Self-recurrent sets are the key to generalize the result of Lemma 3 to stochastic games. The argument of the proof is more involved, and presented in the following three lemmas.

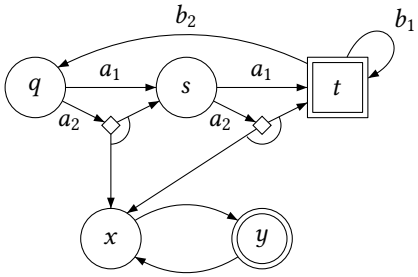


Figure 9. A stochastic game $\mathcal{G}$.

Lemma 7. *If there exists a self-recurrent set* $U \subseteq T$*, then there exists a state from which player* 1 *is almost-sure winning for weakly synchronizing in* T*.*

The almost-sure winning strategy for player 1 constructed in the proof of Lemma 7 is pure and counting.

For the converse of Lemma 7, the structure of the argument is similar to the proof of Lemma 3 for deterministic games and pure strategies. However, the technical details are more involved due to stochasticity (in the game graph, and in the strategy of player 1).

Lemma 8. *Let* $\mathcal{G}$ *be a stochastic game. There exists* $\varepsilon_w > 0$ *and* $N_w \in \mathbb{N}$ *such that the following holds: if there exists no set* $U \subseteq T$ *that is self-recurrent, then in* $\mathcal{G}$ *for all player-*1 *strategies* σ*, for all but at most* N_w *rounds* i*, there exists a player-*2 *strategy* τ *such that* $\mathcal{G}_i^{\sigma,\tau}(T) \leq 1 - \varepsilon_w$*.*

The proof of Lemma 8 constructs a set of cardinality N_w containing the positions that player 2 does not cover from initial distribution d_0. Note the order of the quantifiers in the statement of Lemma 8: player 2 may use different strategies to cover different positions. We use the structure of argument of the proof of Lemma 3 to show that a single strategy of player 2 can cover all but finitely many positions, and we obtain the generalization of Lemma 3 to stochastic games.

Lemma 9. *Let* $\mathcal{G}$ *be a stochastic game. The following equivalence holds: there exists a self-recurrent set* $U \subseteq T$*, if and only if there exists a state from which player* 1 *is almost-sure winning for weakly synchronizing in* T*.*

We present Algorithm 2 to compute the almost-sure winning set for weakly synchronizing objectives. We use the

Algorithm 2: $Solve(\mathcal{H}, p, T)$

Input : $\mathcal{G} = \langle Q, \mathsf{A}, \delta \rangle$ is a stochastic game, $T \subseteq Q$ is a target set.

Output : The set $\{\mathrm{Supp}(d) \mid d \in \langle\!\langle 1 \rangle\!\rangle^{weakly}_{almost}(\mathcal{G}, T)\}$.

begin

1 $r \leftarrow 1$

2 $\mathcal{H} \leftarrow G \times [r]$

3 $S \leftarrow Q \times [r]$

4 $K \leftarrow S$

5 **repeat**

6 **if** *there is a self-recurrent set* $U \subseteq T$ *in* $\mathcal{H} \restriction [K]$ **then**

7 Let $\langle R, r, k \rangle$ be the periodic scheme of U

8 $\mathcal{H} \leftarrow \mathcal{G} \times [r]$

9 $K, S \leftarrow \mathrm{expand}(r, K, S)$

10 Let W be the almost-sure winning region for the (state-based) reachability objective $\Diamond(R \times \{k \mod r\})$ in $\mathcal{H} \restriction [K]$

11 $X \leftarrow \mathsf{PosAttr}_1(W, \mathcal{H} \restriction [K])$

12 $K \leftarrow K \setminus X$

else

13 $L \leftarrow \mathsf{PosAttr}_2(K, \mathcal{H} \restriction [S])$

14 $S \leftarrow S \setminus L$

15 $K \leftarrow S$

until $K = \varnothing$

16 **return** $\{s \subseteq Q \mid \exists i : s \times \{i\} \subseteq S\}$

game of Figure 9 for illustration. The game contains a self-recurrent set $U = \{y\}$ (with period 2), thus player 1 is winning from x and from y. Player 1 is also winning from the other states: the mass of probability that eventually stays in t is winning, and the remaining mass of probability can be injected in U by player 1 at the correct times to be synchronized modulo the period 2, thanks to the consecutive transitions on a_2 from q and from s.

Given the game $\mathcal{G}$ as input, the algorithm considers subgames of $\mathcal{H} = \mathcal{G} \times [r]$ where $r = 1$ initially (lines 1-2), with state space S, initially $S = Q \times [1]$ (line 3). The working variable K (line 4) is used to compute losing states for player 1.

The algorithm proceeds iteratively to construct K, by removing states from S. In the loop of line 5, as long as there is a self-recurrent set U in the subgame $\mathcal{H} \restriction [K]$, we expand the game H to track the number of rounds modulo the period of U (which must be a multiple of r). Given a set $S \subseteq Q \times [p]$, and a period r that is a multiple of p, the *r-expansion* of S is the set $\{\langle q, i \rangle \mid 0 \leq i \leq r-1 \wedge \langle q, i \mod p \rangle \in S\}$. The expand function computes the r-expansion of S and K at line 9 (where $p = |\{i \mid \langle q, i \rangle \in S\}|$ can be derived from the set S). In the expanded game $\mathcal{H}$ (line 8), we compute the

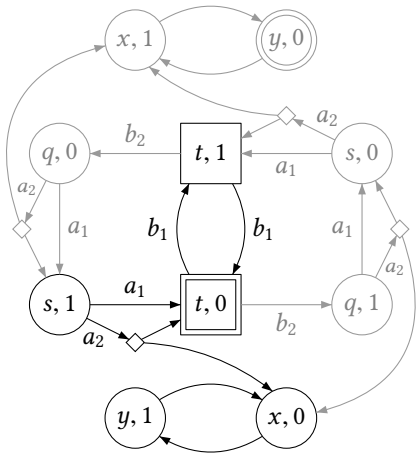


Figure 10. The game $\mathcal{H} = \mathcal{G} \times [2]$ for the game $\mathcal{G}$ of Figure 9 with the shaded region X computed by Algorithm 2 (line 11).

almost-sure winning region W for the (state-based) reachability objective $\Diamond(R \times \{k\})$, which we call a *core winning region*, where $k = \mathsf{k}(U) \bmod r$. From the states in W player 1 is almost-sure weakly synchronizing in $T \times \{0\}$ (see the proof of Lemma 7). Figure 10 shows the 2-expansion of the game of Figure 9. The value k is such that from $R \times \{k\}$ player 1 can inject all the probability mass into $T \times \{0\}$ (in fact, in $U \times \{0\}$).

The next iteration starts after removing from the state space K the positive attractor for player 1 to W (lines 11-12), thus ensuring $\mathcal{H} \restriction [K]$ is again a subgame (the dark part of Figure 10). Whenever there is no set U in $\mathcal{H} \restriction [K]$ satisfying the conditions of Lemma 9, the whole state space K is losing for player 1 and we remove its positive attractor for player 2 (lines 13-15). This part of the algorithm is illustrated in the game of Figure 11, where the self-recurrent set $U = \{y\}$ (with period 1) induces a core winning region $W = \{x, y\}$, and in the subgame obtained by removing the positive attractor for player 1 to W, the set $U = \{q\}$ is self-recurrent. The remaining subgame with state space $\{s\}$ has

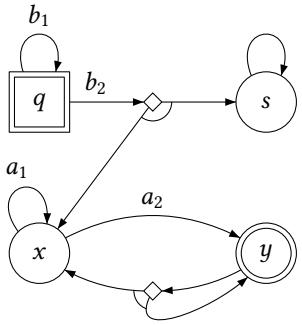


Figure 11. A stochastic game.

no self-recurrent set, thus we remove s and its positive attractor $\{q, s\}$ for player 2, showing that q and s are losing for player 1.

The loop (line 5) terminates when $K = \emptyset$, which can happen if either the state space S can be partitioned by positive attractors to core winning regions, and then the whole state space S is winning for player 1, or if all states in S are losing ($L = S$), and then the winning region for player 1 is empty. The algorithm then returns the slices of the winning region, which correspond to the support of the winning distributions.

The correctness of Algorithm 2 is established in Lemma 10, which also shows PSPACE upper bound for the membership problem.

Lemma 10. *Given a stochastic game and a set T of target states, Algorithm 2 computes the supports of the distributions from which player* 1 *is almost-sure winning for weakly synchronizing in T. This algorithm can be implemented in PSPACE.*

We obtain the following theorem, where the PSPACE upper bound is given by Lemma 10, and the lower bound and memory requirement hold in the special case of MDPs [18, Theorem 6].

Theorem 3. *The membership problem for almost-sure weakly synchronizing in stochastic games is PSPACE-complete, and pure counting strategies are sufficient for player* 1*. Infinite memory is necessary in general.*

4.3 Other synchronizing objectives

The almost-sure winning region for the other synchronizing objectives can be computed relatively easily.

Theorem 4. *The membership problem for almost-sure always and strongly synchronizing can be solved in polynomial time, and pure memoryless strategies are sufficient for player* 1*.*

The membership problem for almost-sure eventually and weakly synchronizing is PSPACE-complete, and pure counting strategies are sufficient for player 1*. Infinite memory is necessary in general.*

5 Conclusion

Stochastic games with synchronizing objectives combine stochasticity with the presence of an adversary and a flavour of imperfect information, which together tend to bring undecidability in a continuous setting [25, 29]. The form of imperfect information in these games differs from the traditional setting where the strategy of player 1 is uniform (the same action is played in all states) [3, 13]. Here, player 1 can see the local state of the game, but needs to enforce a global objective defined on state distributions, which are not visible to player 1. Beyond decidability, it is perhaps surprising that the membership problem for games is no harder than for MDPs (PSPACE-complete), although the proof techniques are significantly more involved, mainly due to the presence of an adversary, and the lack of determinacy.

The main question raised by this model is whether it is possible to extend it with a form of communication, while remaining decidable. This would bring us closer to a wide range of applications in synthetic biology [28, 35] and chemical reaction networks [8].

Acknowledgment. The author is grateful to Matthias Függer for pointing out relevant references in the literature on synthetic biology, and to Mahsa Shirmohammadi and Marie van den Bogaard for preliminary discussions about games with synchronizing objectives and for inspiring the example of Figure 2.

References

[1] M. Agrawal, S. Akshay, B. Genest, and P. S. Thiagarajan. 2012. Approximate Verification of the Symbolic Dynamics of Markov Chains. In *Proc. of LICS: Logic in Computer Science*. IEEE, 55–64.

[2] S. Akshay, B. Genest, and N. Vyas. 2018. Distribution-based objectives for Markov Decision Processes. In *Proc. of LICS: Logic in Computer Science*. ACM, 36–45.

[3] C. Baier, M. Größer, and N. Bertrand. 2012. Probabilistic ω-automata. *Journal of the ACM* 59, 1 (2012), 1:1–1:52.

[4] C. Baier and J.-P. Katoen. 2008. *Principles of Model Checking*. MIT.

[5] N. Bertrand, M. Dewaskar, B. Genest, and H. Gimbert. 2017. Controlling a Population. In *Proc. of CONCUR: Concurrency Theory (LIPIcs, Vol. 85)*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 12:1–12:16.

[6] N. Bertrand, B. Genest, and H. Gimbert. 2017. Qualitative Determinacy and Decidability of Stochastic Games with Signals. *Journal of the ACM* 64, 5 (2017), 33:1–33:48.

[7] J. R. Büchi. 1962. On a decision method in restricted second order arithmetic. In *Proc. of International Congress of Logic, Methodology and Philisophical Science 1960*. Stanford University Press, 1–11.

[8] L. Cardelli, M. Kwiatkowska, and L. Laurenti. 2018. Programming discrete distributions with chemical reaction networks. *Nat. Comput.* 17, 1 (2018), 131–145.

[9] K. Chatterjee. 2007. *Stochastic ω-regular Games*. Ph. D. Dissertation. University of California, Berkeley.

[10] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. 2007. Algorithms for Omega-regular Games of Incomplete Information. *Logical Methods in Computer Science* 3, 3:4 (2007).

[11] K. Chatterjee and T. A. Henzinger. 2012. A survey of stochastic ω-regular games. *J. Comput. System Sci.* 78, 2 (2012), 394–413.

[12] A. Church. 1963. Logic, arithmetics, and automata. In *Proc. of International Congress of Mathematicians, 1962*. Institut Mittag-Leffler, 23–35.

[13] T. Colcombet, N. Fijalkow, and P. Ohlmann. 2020. Controlling a Random Population. In *Proc. of FoSSaCS: Foundations of Software Science and Computation Structures (LNCS 12077)*. Springer, 119–135.

[14] I. D. Couzin. 2009. Collective cognition in animal groups. *Trends in cognitive sciences* 13, 1 (2009), 36–43.

[15] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. 2005. Effective leadership and decision-making in animal groups on the move. *Nature* 433 (2005), 513–516.

[16] L. de Alfaro, T. A. Henzinger, and O. Kupferman. 2007. Concurrent reachability games. *Theoretical Computer Science* 386, 3 (2007), 188–217.

[17] L. Doyen. 2022. Stochastic Games with Synchronizing Objectives. *CoRR* abs/2202.12767 (2022).

[18] L. Doyen, T. Massart, and M. Shirmohammadi. 2019. The Complexity of Synchronizing Markov Decision Processes. *J. Comput. System Sci.* 100 (2019), 96–129.
[19] M. Elowitz and S. Leibler. 2000. A synthetic oscillatory network of transcriptional regulators. *Nature* 403, 335-338 (2000).
[20] J. Esparza. 2014. Keeping a Crowd Safe: On the Complexity of Parameterized Verification (Invited Talk). In *Proc. of STACS: Symposium on Theoretical Aspects of Computer Science (LIPIcs, Vol. 25)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 1–10.
[21] V. V. Isaeva. 2012. Self-organization in biological systems. *Biology Bulletin of the Russian Academy of Sciences* 39 (2012), 110–118.
[22] R. Kannan. 1992. Lattice translates of a polytope and the Frobenius problem. *Combinatorica* 12 (1992), 161–177.
[23] D. König. 1936. *Theorie der endlichen und unendlichen Graphen*. Akademische Verlagsgesellschaft, Leipzig.
[24] V. A. Korthikanti, M. Viswanathan, G. Agha, and Y. Kwon. 2010. Reasoning about MDPs as Transformers of Probability Distributions. In *Proc. of QEST: Quantitative Evaluation of Systems*. IEEE Computer Society, 199–208.
[25] O. Madani, S. Hanks, and A. Condon. 2003. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artif. Intell.* 147, 1-2 (2003), 5–34.
[26] D. A. Martin. 1998. The determinacy of Blackwell games. *The Journal of Symbolic Logic* 63, 4 (1998), 1565–1581.
[27] C. J. Myers. 2016. *Engineering genetic circuits*. CRC Press.
[28] A. A. K. Nielsen, B. S. Der, J. Singh, P. Vaidyanathan, V. Paralanov, E. A. Strychalski, D. Ross, D. Densmore, and C. A. Voigt. 2016. Genetic circuit design automation. *Science* 352(6281), aac7341 (2016).
[29] A. Paz. 1971. *Introduction to probabilistic automata*. Academic Press.
[30] A. Pnueli and R. Rosner. 1990. Distributed Reactive Systems are hard to Synthesize. In *Proc. of FOCS: Foundation of Computer Science*. 746–757.
[31] John H. Reif. 1984. The complexity of two-player games of incomplete information. *J. Comput. System Sci.* 29, 2 (1984), 274–301.
[32] S. Schewe. 2014. Distributed synthesis is simply undecidable. *Inf. Process. Lett.* 114, 4 (2014), 203–207.
[33] W. Thomas. 1997. Languages, Automata, and Logic. In *Handbook of Formal Languages*. Vol. 3, Beyond Words. Springer, Chapter 7, 389–455.
[34] M. Y. Vardi. 1985. Automatic Verification of Probabilistic Concurrent Finite-State Programs. In *Proc. of FOCS: Foundations of Computer Science*. IEEE Computer Society, 327–338.
[35] O. Vo, H.-M. Lee, and D. Na. 2019. Synthetic Bacteria for Therapeutics. *Journal of Microbiology and Biotechnology* 29, 6 (2019), 845–855.