

Model-Free Reinforcement Learning for Stochastic Parity Games

Ernst Moritz Hahn 

University of Twente, Enschede, The Netherlands
E.M.Hahn@utwente.nl

Mateo Perez 


University of Colorado Boulder, CO, USA
Mateo.Perez@Colorado.EDU

Sven Schewe 

University of Liverpool, UK
Sven.Schewe@liverpool.ac.uk

Fabio Somenzi 

University of Colorado Boulder, CO, USA
Fabio@Colorado.EDU

Ashutosh Trivedi 

University of Colorado Boulder, CO, USA
Asutosh.Trivedi@Colorado.EDU

Dominik Wojtczak 

University of Liverpool, UK
D.Wojtczak@liverpool.ac.uk

Abstract

This paper investigates the use of model-free reinforcement learning to compute the optimal value in two-player stochastic games with parity objectives. In this setting, two decision makers, player Min and player Max, compete on a finite game arena – a stochastic game graph with unknown but fixed probability distributions – to minimize and maximize, respectively, the probability of satisfying a parity objective. We give a reduction from stochastic parity games to a family of stochastic reachability games with a parameter ε , such that the value of a stochastic parity game equals the limit of the values of the corresponding simple stochastic games as the parameter ε tends to 0. Since this reduction does not require the knowledge of the probabilistic transition structure of the underlying game arena, model-free reinforcement learning algorithms, such as minimax Q-learning, can be used to approximate the value and mutual best-response strategies for both players in the underlying stochastic parity game. We also present a streamlined reduction from $1\frac{1}{2}$ -player parity games to reachability games that avoids recourse to nondeterminism. Finally, we report on the experimental evaluations of both reductions.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects; Computing methodologies → Machine learning algorithms; Mathematics of computing → Markov processes; Theory of computation → Convergence and learning in games

Keywords and phrases Reinforcement learning, Stochastic games, Omega-regular objectives

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2020.21

Funding This work has been supported by the National Natural Science Foundation of China (Grant Nr. 61532019), by the Engineering and Physical Sciences Research Council grants EP/M027287/1 and EP/P020909/1, by a CU Boulder Research and Innovation Office grant, and by the National Science Foundation grant 2009022.



© Ernst Moritz Hahn, Mateo Perez, Sven Schewe, Fabio Somenzi, Ashutosh Trivedi, and Dominik Wojtczak;

licensed under Creative Commons License CC-BY

31st International Conference on Concurrency Theory (CONCUR 2020).

Editors: Igor Konnov and Laura Kovács; Article No. 21; pp. 21:1–21:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Reinforcement Learning (RL, [34]) is an optimization approach applicable to stochastic games with unknown but fixed probability distributions (unknown stochastic games). Users of RL must specify a scalar reward signal whose expected return should be maximized. Therefore, an objective to be satisfied has to be expressed in terms of scalar rewards, whose expected value is then maximized. Turning a specification manually into such an optimization problem is laborious and error prone [38, 22]. We therefore need an automatic conversion from specifications to rewards in order to leverage the power of a logical specification in RL.

Reinforcement learning approaches can broadly be classified as model-free [33] and model-based [12, 37]. Model-based approaches sample the environment and rewards to learn a model of the stochastic game, while model-free approaches aim to compute optimal strategies without explicitly estimating the transition probabilities and rewards. Model-free approaches to RL, such as Q-learning [34], are asymptotically space-efficient [33] and enable the use of universal approximation architectures, such as deep neural networks [13, 22], to store optimal strategies; they have been demonstrated to scale well [33, 32, 15]. The focus of this paper is to enable model-free RL to learn optimal strategies in unknown stochastic games with ω -regular objectives [29] given as parity objectives.

We study stochastic parity games with unknown transition structure, where two players – Player Min and Player Max – take turns to choose actions on a stochastic game arena (SGA) to form an infinite play. Each position-action pair of the SGA is colored with a priority from a finite set of natural numbers. The goal of Player Min is to choose her actions so as to minimize the probability that the highest infinitely-occurring priority is an odd number, while the goal of Player Max is the opposite. It is known that stochastic parity games are positionally determined [8, 7] and, when the transition structure is known, the optimal strategies can be computed in $\text{NP} \cap \text{co-NP}$. This paper investigates the use of model-free reinforcement learning in approximating the value of the stochastic parity game when the transition structure is not known.

Littman [25] proposed the *Markov Games* framework to study the optimal strategies of players in stochastic games with unknown but fixed probability distributions. For payoffs of the players defined using stochastic reachability payoffs (under the stopping game assumption) or, analogously, stochastic discounted payoffs, Littman generalized the classical Q -learning [36, 4] algorithm to compute the optimal value of a game. This algorithm, known as the minimax- Q algorithm, was shown to converge [26] to the optimal value of the game. To enable the application of off-the-shelf convergent RL algorithms, such as the minimax- Q algorithm, for stochastic parity objectives, one needs to reduce the stochastic parity objective to a stochastic reachability objective. Moreover, to enable model-free RL, such a reduction cannot use any information about the transition structure (such as end-component decomposition) of the underlying stochastic game arena. This paper provides such model-free reduction from stochastic parity games to stochastic reachability games.

There are translations of stochastic parity games to stochastic games with scalar payoffs that – unlike model-free RL – require that the model is known [8, 6, 2]. Other approaches are applicable to non-stochastic parity games [5], or to qualitative games [7], whose goal is to compute strategies that guarantee almost-sure satisfaction of the parity condition.

The problem of translating ω -regular objectives into scalar rewards was recently solved in the case of one strategic agent (also known as the case of $1\frac{1}{2}$ players) when the environment is modeled as a Markov Decision Process (MDP) [17]. The MDP is equipped with a Büchi acceptance condition, and the resulting $1\frac{1}{2}$ -player Büchi game is reduced to a reachability

game by adding a sink state, which is reachable via all accepting transitions of the Büchi game with probability ε . As ε tends to 0, the probability of reaching the sink approaches the probability of satisfaction of the objective.

The translation of ω -regular objectives to scalar rewards for $1\frac{1}{2}$ -player games makes use of Büchi automata with restricted nondeterminism (limit-deterministic automata [35, 10, 16, 31] and, more generally, good-for-MDPs (GFM) automata [18]).

However, for $2\frac{1}{2}$ players, even the restricted nondeterminism that is acceptable for $1\frac{1}{2}$ players may lead to incorrect results, because a strategic player may force the objective automaton to reject a valid computation. Therefore, to solve the case of two strategic agents, one can resort to deterministic parity (or similarly powerful [29, 21]) automata. Since all ω -regular objectives are expressible as deterministic parity conditions, we present in Section 3 a reduction of stochastic *parity* games [7, 8] to stochastic reachability games [30, 9]. The latter can be solved by known model-free RL algorithms [25, 26].

For $1\frac{1}{2}$ -player games, a more efficient reduction from stochastic parity games to stochastic reachability games is possible, and we discuss such a reduction in Section 4. Such a direct translation relieves the RL agent from the task of controlling the nondeterministic choices made by the Büchi automaton that encodes the objective.

2 Preliminaries

A *probability distribution* over a finite set S is a function $d: S \rightarrow [0, 1]$ such that $\sum_{s \in S} d(s) = 1$. Let $\mathcal{D}(S)$ denote the set of all discrete distributions over S . We say a distribution $d \in \mathcal{D}(S)$ is a *point distribution* if $d(s) = 1$ for some $s \in S$. For $d \in \mathcal{D}(S)$ we write $\text{supp}(d)$ for $\{s \in S: d(s) > 0\}$.

2.1 Stochastic Parity Games and Simple Stochastic Games

A *stochastic game arena* (SGA) \mathcal{G} is a tuple $(S, A, T, S_{\text{Min}}, S_{\text{Max}})$, where S is a finite set of states, A is a finite set of *actions*, $T: S \times A \rightarrow \mathcal{D}(S)$ is the *probabilistic transition (partial) function*, and $\{S_{\text{Min}}, S_{\text{Max}}\}$ is a partition of the set of states S .

For $s \in S$, $A(s)$ denotes the set of actions that can be selected in state s . For states $s, s' \in S$ and $a \in A(s)$ we write $p(s'|s, a)$ for $T(s, a)(s')$. A *run* of \mathcal{G} is an ω -word $\langle s_0, a_1, s_1, \dots \rangle \in S \times (A \times S)^\omega$ such that $p(s_{i+1}|s_i, a_{i+1}) > 0$ for all $i \geq 0$. A finite run is a finite such sequence, that is, a word in $S \times (A \times S)^*$. For an infinite run r , we write $\text{inf}(r)$ for the set of state-action pairs that appear infinitely often in r . We write $\text{Runs}^{\mathcal{G}}(\text{FRuns}^{\mathcal{G}})$ for the set of runs (finite runs) of the SGA \mathcal{G} and $\text{Runs}^{\mathcal{G}}(s)(\text{FRuns}^{\mathcal{G}}(s))$ for the set of runs (finite runs) of the SGA \mathcal{G} starting from state s . We write $\text{last}(r)$ for the last state of a finite run r .

A game on an SGA \mathcal{G} starts with a token in an *initial state* $s \in S$; players Min and Max construct an infinite run by taking turns to choose enabled actions, and then moving the token to a successor state sampled from the selected distribution. A strategy of player Min in \mathcal{G} is a partial function $\mu: \text{FRuns} \rightarrow \mathcal{D}(A)$, defined for $r \in \text{FRuns}$ if and only if $\text{last}(r) \in S_{\text{Min}}$, such that $\text{supp}(\sigma(r)) \subseteq A(\text{last}(r))$. A strategy ν of player Max is defined analogously.

A strategy σ is *pure* if $\sigma(r)$ is a point distribution wherever it is defined; otherwise, σ is *mixed*. We say that σ is *stationary* if $\text{last}(r) = \text{last}(r')$ implies $\sigma(r) = \sigma(r')$ wherever σ is defined. A strategy is *positional* if it is both pure and stationary. Let Σ_{Min} and Σ_{Max} be the sets of all strategies of player Min and player Max, respectively. Similarly, Π_{Min} and Π_{Max} denote the sets of all *positional* strategies of player Min and player Max, respectively.

Let $\text{Runs}_{\mu, \nu}^{\mathcal{G}}(s)$ denote the subset of runs $\text{Runs}^{\mathcal{G}}(s)$ starting from state s that are consistent with player Min and player Max following strategies μ and ν , respectively. The behavior of an SGA \mathcal{G} under a strategy pair $(\mu, \nu) \in \Sigma_{\text{Min}} \times \Sigma_{\text{Max}}$ is defined on a probability space

$(Runs_{\mu,\nu}^{\mathcal{G}}(s), \mathcal{F}_{Runs_{\mu,\nu}^{\mathcal{G}}(s)}, \Pr_{\mu,\nu}^{\mathcal{G}}(s))$ over the set of infinite runs $Runs_{\mu,\nu}^{\mathcal{G}}(s)$. Given a random variable $f: Runs^{\mathcal{G}} \rightarrow \mathbb{R}$ over the infinite runs of \mathcal{G} , we denote by $\mathbb{E}_{\mu,\nu}^{\mathcal{G}}(s) \{f\}$ the expectation of f over the runs in the probability space $(Runs_{\mu,\nu}^{\mathcal{G}}(s), \mathcal{F}_{Runs_{\mu,\nu}^{\mathcal{G}}(s)}, \Pr_{\mu,\nu}^{\mathcal{G}}(s))$.

Let $[k]$ denote the set of natural numbers $\{0, 1, \dots, k-1\}$. We consider the following payoffs of player Min to player Max.

- **Stochastic Parity Payoff.** A stochastic parity payoff in an SGA \mathcal{G} is defined by a priority function $\text{pri}: S \times A \rightarrow [k]$ that assigns to each state-action pair a natural number called the priority (or color) of that pair. The stochastic parity payoff $\mathcal{P}_{\mu,\nu}^{\mathcal{G}}(s)$ for a strategy pair $(\mu, \nu) \in \Sigma_{\text{Min}} \times \Sigma_{\text{Max}}$ from an initial state $s \in S$ is the probability that the highest recurring priority is odd, i.e.,

$$\mathcal{P}_{\mu,\nu}^{\mathcal{G}}(s) = \Pr_{\mu,\nu}^{\mathcal{G}}(s) \left\{ r \in Runs_{\mu,\nu}^{\mathcal{G}}(s) : \max \{ \text{pri}(s, a) : (s, a) \in \text{inf}(r) \} \text{ is odd} \right\} .$$

A *stochastic Büchi payoff* is a stochastic parity payoff with $k = 2$. It is customarily specified as a set of *accepting* transitions: those with priority 1.

- **Stochastic Reachability Payoff.** A stochastic reachability payoff in an SGA \mathcal{G} is defined by two distinguished sink states: the *accepting sink* state $s_a \in S$ and the *rejecting sink* state $s_r \in S$. Recall that a sink state s satisfies $p(s|s, a) = 1$ for all $a \in A$. The stochastic reachability payoff $\mathcal{R}_{\mu,\nu}^{\mathcal{G}}(s)$ for a strategy pair $(\mu, \nu) \in \Sigma_{\text{Min}} \times \Sigma_{\text{Max}}$ from an initial state $s \in S$ is the probability of reaching the accepting sink s_a , i.e.,

$$\mathcal{R}_{\mu,\nu}^{\mathcal{G}}(s) = \Pr_{\mu,\nu}^{\mathcal{G}}(s) \left\{ r \in Runs_{\mu,\nu}^{\mathcal{G}}(s) : r \text{ visits } s_a \right\} .$$

We refer to a stochastic game arena with a parity payoff as a stochastic parity game (SPG) $\mathbb{G} = (\mathcal{G}, \text{pri})$, and to a stochastic game arena with a reachability payoff as a stochastic reachability game (SRG) $\mathbb{G}' = (\mathcal{G}, s_a, s_r)$.

We assume that an SRG is a *stopping game*, i.e., for every pair of strategies $(\mu, \nu) \in \Sigma_{\text{Min}} \times \Sigma_{\text{Max}}$ and every initial state $s \in S$, the set $\{s_a, s_r\}$ is visited with probability 1. This implies that there are no sinks besides s_a and s_r and, in addition, that at least one sink is reachable with positive probability from every state in S .

Given a payoff function $\mathcal{C} \in \{\mathcal{P}, \mathcal{R}\}$, the objective of player Max in the corresponding game \mathbb{G} is to maximize the payoff, while the objective of player Min is the opposite. For every state $s \in S$, we define its *upper value* $\overline{\text{Val}}(\mathbb{G}, s)$ as the minimum payoff player Min can ensure irrespective of player Max's strategy. Similarly, the *lower value* $\underline{\text{Val}}(\mathbb{G}, s)$ of a state $s \in S$ is the maximum payoff player Max can ensure irrespective of player Min's strategy, i.e.,

$$\overline{\text{Val}}(\mathbb{G}, s) = \inf_{\mu \in \Sigma_{\text{Min}}} \sup_{\nu \in \Sigma_{\text{Max}}} \mathcal{C}_{\mu,\nu}^{\mathcal{G}}(s) \quad \text{and} \quad \underline{\text{Val}}(\mathbb{G}, s) = \sup_{\nu \in \Sigma_{\text{Max}}} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{C}_{\mu,\nu}^{\mathcal{G}}(s) .$$

The inequality $\underline{\text{Val}}(\mathbb{G}, s) \leq \overline{\text{Val}}(\mathbb{G}, s)$ holds of all two-player zero-sum games. A game is *determined* when, for every state $s \in S$, its lower value and its upper value are equal; we then say that the value of the game Val exists and $\text{Val}(\mathbb{G}, s) = \underline{\text{Val}}(\mathbb{G}, s) = \overline{\text{Val}}(\mathbb{G}, s)$ for every $s \in S$. For a strategy $\mu \in \Sigma_{\text{Min}}$ of player Min and similarly, for a strategy $\nu \in \Sigma_{\text{Max}}$ of player Max, we define their values Val^{μ} and Val^{ν} as

$$\text{Val}^{\mu} : s \mapsto \sup_{\nu \in \Sigma_{\text{Max}}} \mathcal{C}_{\mu,\nu}^{\mathcal{G}}(s) \quad \text{and} \quad \text{Val}^{\nu} : s \mapsto \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{C}_{\mu,\nu}^{\mathcal{G}}(s) .$$

We say that a positional strategy $\mu_* \in \Pi_{\text{Min}}$ of player Min is optimal if $\text{Val}^{\mu_*} = \text{Val}$. Similarly, a positional strategy $\nu_* \in \Pi_{\text{Max}}$ of player Max is optimal if $\text{Val}^{\nu_*} = \text{Val}$. We say that a game is *positionally determined* if both players have positional optimal strategies.

- **Theorem 1** ([8, 7, 9]). *Stochastic parity games and stochastic reachability games are positionally determined and are in $\text{NP} \cap \text{co-NP}$.*

2.2 Markov Decision Processes and Markov Chains

If, for every state $s \in S_{\text{Min}}$, or for every state $s \in S_{\text{Max}}$, $A(s)$ is a singleton, then \mathcal{G} is a *Markov Decision Process* (MDP). If we want to emphasize that only Player Max (Min) has choices, we refer to Max-MDPs (Min-MDPs). If, for every state $s \in S$, $A(s)$ is a singleton, then \mathcal{G} is a *Markov chain*. We denote by \mathcal{G}_μ (\mathcal{G}_ν) the MDP obtained by fixing the strategy of player Min (Max) to positional strategy $\mu \in \Pi_{\text{Min}}$ ($\nu \in \Pi_{\text{Max}}$). If the strategies of both players are fixed, we denote the resulting Markov chain by $\mathcal{G}_{\mu,\nu}$.

Since an MDP has one strategic player, we can define an MDP by the tuple (S, A, T) . For an MDP $\mathcal{M} = (S, A, T)$, we define its directed underlying graph $GR(\mathcal{M}) = (V, E)$ where $V = S$ and $E = \{(s, s') : T(s, a)(s') > 0 \text{ for some } a \in A(s)\}$. A sub-MDP of \mathcal{M} is an MDP $\mathcal{M}' = (S', A', T')$, where $S' \subset S$, $A' \subseteq A$, is such that $A'(s) \subseteq A(s)$ for every $s \in S'$, and T' is T restricted to S' and A' . \mathcal{M}' is closed under probabilistic transitions, i.e., for all $s \in S'$ and $a \in A'$, we have that $T(s, a)(s') > 0$ implies that $s' \in S'$. An *end-component* [11] of an MDP is a sub-MDP such that its underlying graph is strongly connected. Once an end-component C of an MDP is entered, there is a strategy that visits every state-action combination in C with probability 1 and stays in C forever. Moreover, for every strategy the union of the end-components is visited with probability 1.

A *bottom strongly connected component* (BSCC) of a Markov chain is a recurrent class. A BSCC is *even* if the highest priority of its transitions is even; otherwise the BSCC is *odd*.

3 From Stochastic Parity Games to Stochastic Reachability Games

In this section, we now show how to construct, for a Stochastic Parity Game \mathbb{G} , a family of Stochastic Reachability Games (SRGs) \mathbb{G}^ε , parametrized by a parameter $\varepsilon \in (0, 1)$, that have strong convergence properties to \mathbb{G} : for sufficiently small ε , optimal positional strategies (of either player) for \mathbb{G}^ε are also optimal positional strategies for \mathbb{G} , and the limit value (when ε goes to 0) of \mathbb{G}^ε goes to the value of \mathbb{G} for every state. Thus, learning optimal strategies for this family of SRGs can be used to obtain optimal strategies for \mathbb{G} .

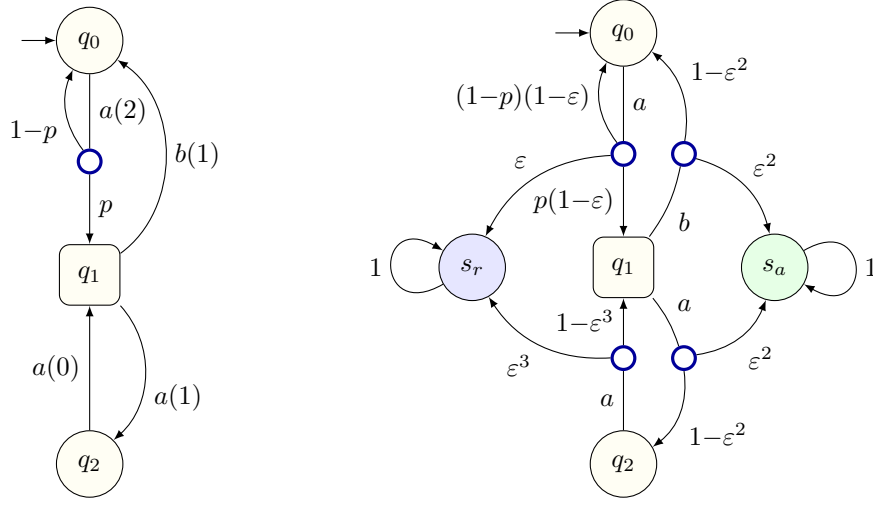
3.1 Limit Reachability Theorem

Given a stochastic parity game $\mathbb{G} = (\mathcal{G} = (S, A, T, S_{\text{Min}}, S_{\text{Max}}), \text{pri})$, with $\text{pri} : S \times A \rightarrow [k]$, and $\varepsilon \in (0, 1)$, we define a related stochastic reachability game $\mathbb{G}^\varepsilon = (\mathcal{G}^\varepsilon = (S \cup \{s_a, s_r\}, A, T^\varepsilon, S_{\text{Min}} \cup \{s_r\}, S_{\text{Max}} \cup \{s_a\}), s_a, s_r)$ such that:

$$T^\varepsilon(s, a)(s') = \begin{cases} 1 & \text{if } s = s' = s_a \text{ or } s = s' = s_r \\ \varepsilon^{k-i} & \text{if } s \in S, s' = s_a, i = \text{pri}(s, a), \text{ and } i \text{ is odd} \\ \varepsilon^{k-i} & \text{if } s \in S, s' = s_r, i = \text{pri}(s, a), \text{ and } i \text{ is even} \\ (1 - \varepsilon^{k-i}) \cdot T(s, a)(s') & \text{if } s, s' \in S \text{ and } i = \text{pri}(s, a) \\ 0 & \text{otherwise.} \end{cases}$$

An example is shown in Figure 1, where boxes denote states in S_{Max} , large circles denote states in S_{Min} and small circles denote probabilistic branches. Intuitively, for small enough ε , the chance of prematurely moving to the wrong sink is negligible. Specifically, the probability of reaching a sink from a transient transition is negligible, while in a recurrent set of states, the probability of reaching a sink from a lower priority transition is negligible compared to the probability of doing so from a higher priority transition. The definition of T^ε also applies to $1\frac{1}{2}$ -player games and to Markov chains; it is illustrated in Figure 2.

21:6 Model-Free Reinforcement Learning for Stochastic Parity Games



■ **Figure 1** An SPG \mathbb{G} (left) and the corresponding SRG \mathbb{G}^ε (right).

► **Proposition 2.** For every SPG \mathbb{G} , the SRG \mathbb{G}^ε is a stopping game.

In the next section, we prove the following lemma.

► **Lemma 3.** For every positional strategy pair $(\mu, \nu) \in \Pi_{\text{Min}} \times \Pi_{\text{Max}}$ and every state $s \in S$ we have that $\mathcal{P}_{\mu, \nu}^{\mathbb{G}}(s) = \lim_{\varepsilon \downarrow 0} \mathcal{R}_{\mu, \nu}^{\mathbb{G}^\varepsilon}(s)$. Moreover, for sufficiently small ε , optimal strategies for \mathbb{G}^ε are also optimal for \mathbb{G} .

► **Theorem 4.** For every stochastic parity game $\mathbb{G} = ((S, A, T, S_{\text{Min}}, S_{\text{Max}}), \text{pri})$ and the set of stochastic reachability games \mathbb{G}^ε , we have that $\text{Val}(\mathbb{G}, s) = \lim_{\varepsilon \downarrow 0} \text{Val}(\mathbb{G}^\varepsilon, s)$, for all $s \in S$.

Proof. Notice that, for every positional strategy $\nu \in \Pi_{\text{Max}}$, we have that:

$$\begin{aligned} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{P}_{\mu, \nu}^{\mathbb{G}}(s) &= \min_{\mu \in \Pi_{\text{Min}}} \mathcal{P}_{\mu, \nu}^{\mathbb{G}}(s) = \min_{\mu \in \Pi_{\text{Min}}} \lim_{\varepsilon \downarrow 0} \mathcal{R}_{\mu, \nu}^{\mathbb{G}^\varepsilon}(s) \\ &= \lim_{\varepsilon \downarrow 0} \min_{\mu \in \Pi_{\text{Min}}} \mathcal{R}_{\mu, \nu}^{\mathbb{G}^\varepsilon}(s) = \lim_{\varepsilon \downarrow 0} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{R}_{\mu, \nu}^{\mathbb{G}^\varepsilon}(s). \end{aligned} \quad (1)$$

The first and the last equalities follow due to the optimality of positional strategies in stochastic parity games (Theorem 1), while the second and the third equalities follows from Lemma 3. Now, observe that for all $s \in S$ we have that:

$$\begin{aligned} \text{Val}_{\mathcal{P}}(\mathbb{G}, s) &= \sup_{\nu \in \Sigma_{\text{Max}}} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{P}_{\mu, \nu}^{\mathbb{G}}(s) && \text{(by definition)} \\ &= \max_{\nu \in \Pi_{\text{Max}}} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{P}_{\mu, \nu}^{\mathbb{G}}(s) && \text{(from Theorem 1)} \\ &= \max_{\nu \in \Pi_{\text{Max}}} \lim_{\varepsilon \downarrow 0} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{R}_{\mu, \nu}^{\mathbb{G}^\varepsilon}(s) && \text{(from (1))} \\ &= \lim_{\varepsilon \downarrow 0} \max_{\nu \in \Pi_{\text{Max}}} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{R}_{\mu, \nu}^{\mathbb{G}^\varepsilon}(s) && \text{(from Lemma 3)} \\ &= \lim_{\varepsilon \downarrow 0} \sup_{\nu \in \Sigma_{\text{Max}}} \inf_{\mu \in \Sigma_{\text{Min}}} \mathcal{R}_{\mu, \nu}^{\mathbb{G}^\varepsilon}(s) && \text{(from Theorem 1)} \\ &= \lim_{\varepsilon \downarrow 0} \text{Val}_{\mathcal{R}}(\mathbb{G}^\varepsilon, s) && \text{(by definition).} \quad \blacktriangleleft \end{aligned}$$

3.2 Absorption Probabilities

For a pair of positional strategies (μ, ν) , a stochastic game arena \mathcal{G} (\mathcal{G}^ε) reduces to a Markov chain $\mathcal{G}_{\mu, \nu}$ ($\mathcal{G}_{\mu, \nu}^\varepsilon$), whose states are partitioned into a set of transient states and one or more recurrent (communicating) classes, where a *communicating class* is a class that becomes recurrent when removing the sinks s_a and s_r . Comparing the Markov chains $\mathcal{G}_{\mu, \nu}$ and $\mathcal{G}_{\mu, \nu}^\varepsilon$, one observes that:

- Every transient state of the Markov chain $\mathcal{G}_{\mu, \nu}$ remains transient in $\mathcal{G}_{\mu, \nu}^\varepsilon$.
- All recurrent classes of $\mathcal{G}_{\mu, \nu}$ become communicating classes of $\mathcal{G}_{\mu, \nu}^\varepsilon$.
- The chain $\mathcal{G}_{\mu, \nu}^\varepsilon$ is absorbing; the runs that do not eventually reach either s_a or s_r form a set of measure 0.
- Since a positional strategy selects one action for each state, exactly one priority in $[k]$, denoted by $\text{pri}(s)$ is associated to each state of $\mathcal{G}_{\mu, \nu}$.

Note that the runs of $\mathcal{G}_{\mu, \nu}$ that do not reach some recurrent class are a set of measure 0. Moreover, the runs that reach the absorbing states of \mathbb{G}^ε without going through a recurrent class of \mathbb{G} are a set, whose measure converges to 0 when ε goes to 0. Hence, we can analyze the Markov chains induced by positional strategies (μ, ν) one recurrent class at a time.

► **Lemma 5.** *Suppose the Markov chain \mathcal{M} is recurrent.*

1. *The sum of the absorption probabilities of the two sinks of \mathcal{M}^ε is always 1.*
2. *The limit, for ε that goes to 0, of the absorption probabilities of the odd sink of \mathcal{M}^ε is 1 if, and only if, the highest priority of the states in \mathcal{M} is odd.*
3. *The limit, for ε that goes to 0, of the absorption probabilities of the even sink of \mathcal{M}^ε is 1 if, and only if, the highest priority of the states in \mathcal{M} is even.*

Proof. The first claim follows from \mathcal{M}^ε being a stopping game (cf. Proposition 2). For the second claim, let M be the $n \times n$ transition matrix of \mathcal{M} . Let $\text{pri}(i)$ be the priority of state i . The Markov chain \mathcal{M}^ε is absorbing and its transition matrix M^ε can be written in the following form:

$$M^\varepsilon = \begin{pmatrix} I_2 & 0 \\ R & Q \end{pmatrix},$$

where I_u is the $u \times u$ identity matrix, R is $n \times 2$, and Q is $n \times n$. The first two rows and columns of M^ε are named o and e (odd and even, respectively). The other rows and columns are numbered from 0 to $n - 1$. Let E be the $n \times n$ diagonal matrix such that

$$e_{ii} = \varepsilon^{k - \text{pri}(i)}.$$

The matrix R is defined by $r_{io} = e_{ii}$ if $\text{pri}(i)$ is odd and 0 otherwise; likewise $r_{ie} = e_{ii}$ if $\text{pri}(i)$ is even and 0 otherwise. The matrix Q is defined by

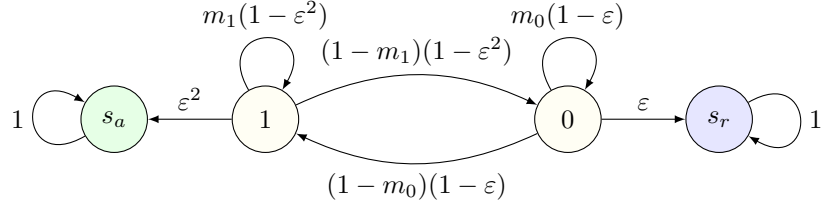
$$Q = (I_n - E) \cdot M.$$

The probabilities of reaching the sinks from the remaining states are computed as

$$P = (I_n - Q)^{-1} \cdot R,$$

where $N = (I_n - Q)^{-1}$ is called the *fundamental matrix* for the absorbing chain M^ε and n_{ij} is the expected number of times the absorbing chain is in state j if it starts in state i [23, Theorem 3.2.1] [14, Theorem 11.4]. Since M is recurrent, $n_{ij} > 0$. Let $N_i^+(\varepsilon) = \max_j \{n_{ij}\}$ and $N_i^-(\varepsilon) = \min_j \{n_{ij}\}$. That gives lower and upper bounds for the rows of N that are strictly positive row vectors with uniform entries. Then,

$$\lim_{\varepsilon \downarrow 0} \frac{N_i^-(\varepsilon)}{N_i^+(\varepsilon)} \cdot \frac{\sum_{0 \leq \ell < n} r_{\ell o}}{\sum_{0 \leq \ell < n} r_{\ell e}} \leq \lim_{\varepsilon \downarrow 0} \frac{p_{io}}{p_{ie}} \leq \lim_{\varepsilon \downarrow 0} \frac{N_i^+(\varepsilon)}{N_i^-(\varepsilon)} \cdot \frac{\sum_{0 \leq \ell < n} r_{\ell o}}{\sum_{0 \leq \ell < n} r_{\ell e}}.$$



■ **Figure 2** An augmented Markov chain.

Since $\frac{N_i^+(\epsilon)}{N_i^-(\epsilon)}$ converges to the ratio of the largest stationary probability in \mathcal{M} to the smallest such probability, regardless of i , its limit is positive and finite. The limit of p_{io}/p_{ie} is therefore determined by the ratio of the sums of the columns of R . Both columns are polynomials in ϵ with no constant term, and the lowest degree of the two polynomials is different – one is even and the other is odd. Therefore, *either* p_{io}/p_{ie} goes to infinity (or the denominator stays 0), *or* p_{io}/p_{ie} goes to 0. Since $p_{io} + p_{ie} = 1$, this entails that one probability goes to 1 (the one for the column of R with the lowest degree term), while the other goes to 0.

The proof for the third claim is similar. ◀

► **Example 6.** Figure 2 shows an augmented Markov chain whose original Markov chain has two states, State 0 with priority 2 and State 1 with priority 1. The probabilities m_0 and m_1 are from $[0, 1]$. The transition matrix, M^ϵ , of the augmented chain is given by:

$$M^\epsilon = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \epsilon & m_0(1 - \epsilon) & (1 - m_0)(1 - \epsilon) \\ \epsilon^2 & 0 & (1 - m_1)(1 - \epsilon^2) & m_1(1 - \epsilon^2) \end{pmatrix}.$$

The fundamental matrix N is:

$$N = \frac{1}{|I - Q|} \cdot \begin{pmatrix} 1 - m_1 + m_1\epsilon^2 & (1 - m_0)(1 - \epsilon) \\ (1 - m_1)(1 - \epsilon^2) & 1 - m_0 + m_0\epsilon \end{pmatrix},$$

with $|I - Q| = \epsilon(1 - m_1 + \epsilon(1 - m_0) - \epsilon^2(1 - m_0 - m_1))$. The probabilities of eventually reaching the sinks are given by:

$$N \cdot R = \frac{\epsilon}{|I - Q|} \begin{pmatrix} \epsilon(1 - \epsilon)(1 - m_0) & 1 - m_1 + m_1\epsilon^2 \\ \epsilon(1 - m_0 + m_0\epsilon) & (1 - \epsilon^2)(1 - m_1) \end{pmatrix}.$$

Both rows of $N \cdot R$ sum to 1 and

$$\lim_{\epsilon \downarrow 0} N \cdot R = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix},$$

as expected.

Proof of Lemma 3. For a given pair of positional strategies $(\mu, \nu) \in \Pi_{\text{Min}} \times \Pi_{\text{Max}}$ on \mathcal{G} , and for every state s of the Markov chain $\mathcal{G}_{(\mu, \nu)}$, Lemma 5 shows that the following holds for every state $s \in S$:

1. if the state s is in an even BSCC of $\mathcal{G}_{(\mu, \nu)}$ then $\lim_{\epsilon \downarrow 0} \mathcal{R}_{\mu, \nu}^\epsilon(s) = 0 = \mathcal{P}_{\mu, \nu}^{\mathcal{G}}(s)$,
2. if the state s is in an odd BSCC of $\mathcal{G}_{(\mu, \nu)}$ then $\lim_{\epsilon \downarrow 0} \mathcal{R}_{\mu, \nu}^\epsilon(s) = 1 = \mathcal{P}_{\mu, \nu}^{\mathcal{G}}(s)$,

Let $s \in S$ be a transient state of $\mathcal{G}_{(\mu,\nu)}$. Let $f_s^{\mu,\nu}$ be the expected number of transitions taken before reaching a BSCC when starting at s in $\mathcal{G}_{(\mu,\nu)}$. Note that, with argument similar to the one used in [17, Lemma 2], one shows that

$$\mathcal{R}_{\mu,\nu}^{\mathcal{G}^\varepsilon}(s) - \varepsilon f_s^{\mu,\nu} \leq \mathcal{P}_{\mu,\nu}^{\mathcal{G}}(s) \leq \mathcal{R}_{\mu,\nu}^{\mathcal{G}^\varepsilon}(s) + \varepsilon f_s^{\mu,\nu} .$$

That is, the effect of transient states vanishes with ε . Therefore, as ε goes to 0, $\mathcal{R}_{\mu,\nu}^{\mathcal{G}^\varepsilon}(s)$ tends to $\mathcal{P}_{\mu,\nu}^{\mathcal{G}}(s)$.

Note that this means that, for either player, for every strategy μ or ν that is superior over a strategy μ' or ν' , respectively, in $\mathcal{P}^{\mathcal{G}}$, there is an $\varepsilon' > 0$ such that, for $\varepsilon \in (0, \varepsilon')$, μ or ν is superior over μ' or ν' , respectively, in $\mathcal{R}^{\mathcal{G}^\varepsilon}$.

Given that optimal strategies are positional, and that there are only finitely many positional strategies, this implies that there is an $\varepsilon' > 0$ such that, for $\varepsilon \in (0, \varepsilon')$, optimal strategies for either player in $\mathcal{R}^{\mathcal{G}^\varepsilon}$ are also optimal in $\mathcal{P}^{\mathcal{G}}$. ◀

4 Markov Decision Processes with Parity Objectives

The reduction of Section 3 works for all stochastic parity games, but, for a parity condition with k priorities, it employs up to k distinct powers of ε . In practice, this may lead to slow convergence as a reinforcement learner will require long episodes. We introduce another reduction, which is only valid for $1\frac{1}{2}$ -player games (Markov decision processes), but only uses the first power of ε . We consider Max-MDPs. (Min-MDPs can be treated by dualizing the MDP first.)

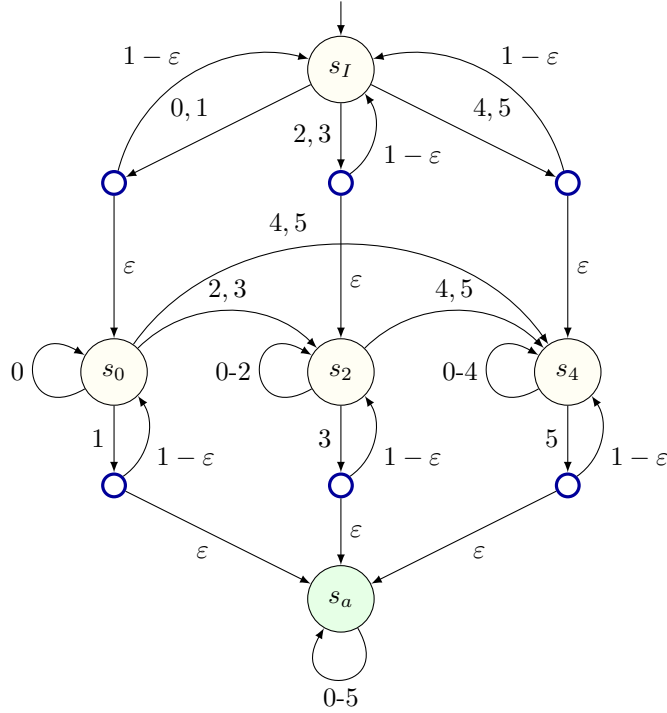
Our reduction is obtained by composing the MDP with the following *priority tracker* gadget.

► **Definition 7** (Priority Tracker). *Given a set of priorities $[k]$ and a parameter $0 \leq \varepsilon \leq 1$, the priority tracker \mathcal{T}^ε is an MDP (S, A, T) where:*

- $S = \{s_I, s_a\} \cup \{s_{2c} : 2c \in [k]\}$ is the set of states with $2 + \lceil k/2 \rceil$ states including a distinguished initial state s_I , an accepting sink state s_a , and a state s_{2c} for each pair $(2c, 2c + 1)$ of priorities (w.l.o.g., we assume that k is even);
- $A = [k]$ is the set of actions that are labeled by priorities from the set $[k]$; and
- $T : S \times A \rightarrow \mathcal{D}(S)$ is the transition function defined in the following way.

$$T(s, a)(s') = \begin{cases} 1 - \varepsilon & \text{if } s = s' = s_I \\ \varepsilon & \text{if } s = s_I \text{ and } s' = s_{2c} \text{ and } a \in \{2c, 2c + 1\} \\ 1 & \text{if } s = s' = s_{2c} \text{ and } a < 2c \\ \varepsilon & \text{if } s = 2c, a = 2c + 1, \text{ and } s' = s_a \\ 1 - \varepsilon & \text{if } s = 2c, a = 2c + 1, \text{ and } s' = 2c \\ 1 & \text{if } s = 2c, a > 2c + 1, \text{ and } s' = 2\lfloor a/2 \rfloor \\ 1 & \text{if } s = s' = s_a \\ 0 & \text{otherwise.} \end{cases}$$

An example of the priority tracker for priority set $\{0, 1, \dots, 5\}$ is shown in Figure 3. Intuitively, for small enough ε , the gadget is, with high probability, still in state s_I when the MDP enters an end component. Moreover, for small enough ε , the gadget is very likely to see the dominant priority of the end component before it reaches s_a , in which case it reaches s_a with probability 1 if and only if the dominating priority of the end component is odd.



■ **Figure 3** Priority tracker gadget for priorities 0-5.

To prove that the gadget of Figure 3 may be used for $1\frac{1}{2}$ -player stochastic games (Max-MDPs), we make use of a Büchi automaton (i.e., with a parity condition with priorities 0 and 1), which is derived from the priority tracker gadget, as follows.

The *Parity to Büchi* (PtB) gadget for $2k$ priorities, \mathcal{P}_k , is a Büchi automaton over the alphabet $[2k]$ that accepts the language of the following LTL property:

$$\text{GF}\{2k - 1\} \vee (\text{FG}[2k - 2] \wedge (\text{GF}\{2k - 3\} \vee \dots)) . \quad (2)$$

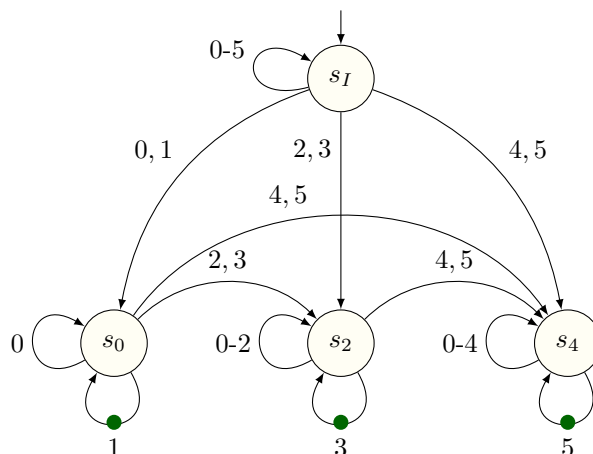
The PtB \mathcal{P}_6 is shown in Figure 4. In general, the PtB is related to the priority tracker gadget for the same number of priorities by the following transformation. One replaces the transitions from the initial state by nondeterministic transitions (all non-accepting), and uses non-accepting transitions:

- from state $2c$, one self-loops with every priority $d < 2c + 1$;
 - from state $2c$, one moves to state $2 \cdot \lfloor d/2 \rfloor$ for every priority $d > 2c + 1$;
- and accepting transitions
- from state $2c$, one self-loops with every priority $d = 2c + 1$.

► **Lemma 8.** *Let \mathbb{B} be the synchronous composition of the Max-MDP $\mathbb{M} = ((S, A, T, \emptyset, S), \text{pri})$ and \mathcal{P}_{2k} , with $\text{pri} : S \times A \rightarrow [2k]$. Then, for every $s \in S$,*

$$\text{Val}(\mathbb{M}, s) = \text{Val}(\mathbb{B}, (s, s_I)) .$$

Proof. The automaton \mathcal{P}_{2k} is good for MDPs [18], because it is a suitable limit-deterministic Büchi automaton [16, 31]. This means that it can be composed with any Max-MDP equipped with a parity condition to compute the probability of satisfaction of (2). ◀



■ **Figure 4** PtB gadget for priorities 0-5. The transitions marked with a dot are accepting.

Let $\mathbb{M} = (\mathcal{M}, \text{pri})$ be a stochastic game arena, with no choices for Player Min and with parity objective, and \mathcal{T}^ε be the priority tracker. We define $\mathbb{M}_{\mathcal{T}}^\varepsilon$ to be the synchronous composition of \mathbb{M} with \mathcal{T}^ε , synchronized with the priorities of the transitions. We assume that $\mathbb{M}_{\mathcal{T}}^\varepsilon$ is equipped with a reachability objective with s_a as the accept state.

► **Theorem 9.** *For every Max-MDP $\mathbb{M} = ((S, A, T, S_{\text{Min}}, S_{\text{Max}}), \text{pri})$ and its induced set of stochastic reachability games $\mathbb{M}_{\mathcal{T}}^\varepsilon$, we have that, for all $s \in S$,*

$$\text{Val}(\mathbb{M}, s) = \lim_{\varepsilon \downarrow 0} \text{Val}(\mathbb{M}_{\mathcal{T}}^\varepsilon, (s, s_I)) .$$

Proof. The proof is in two parts.

■ **Bounding the limit from below.** In the first part of the proof we show that, for every $\delta > 0$, there exists an $\varepsilon_\delta > 0$ such that for every $\varepsilon < \varepsilon_\delta$ we have that

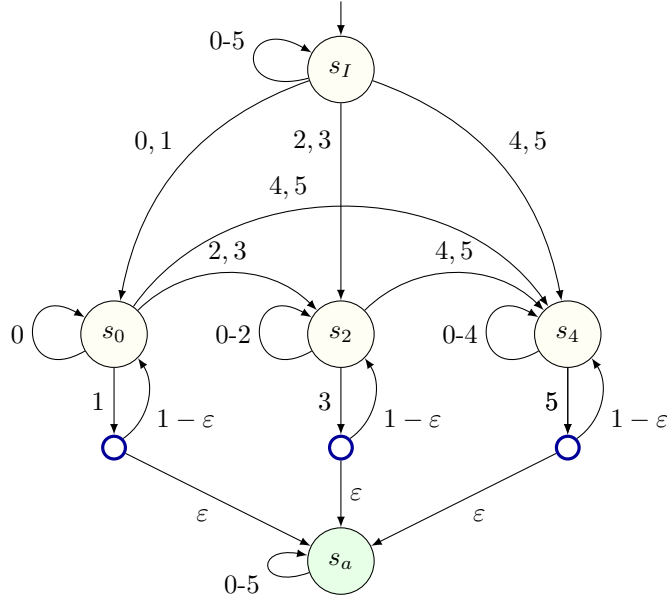
$$\text{Val}(\mathbb{M}_{\mathcal{T}}^\varepsilon, (s, s_I)) \geq \text{Val}(\mathbb{M}, s) - \delta.$$

Assume that player Max plays a positional strategy, which is optimal for \mathbb{M} . We first consider the special cases that s is in a winning or losing BSCC in the Markov chain induced by this strategy.

- The state s is in an accepting BSCC with dominating odd priority o . In this case, the probability to satisfy the parity objective is 1. At the same time, in composition with the priority tracker gadget, no state s_e with $e > o$ can be reached in the priority tracker, and there is a path from every state to s_a in the finite product Markov chain. The reachability probability is therefore also 1. In this case $\text{Val}(\mathbb{M}_{\mathcal{T}}^\varepsilon, (s, s_I)) = \text{Val}(\mathbb{M}, s)$ holds.
- The state s is in a rejecting BSCC. In this case, the probability to satisfy the parity objective is 0, and $\text{Val}(\mathbb{M}_{\mathcal{T}}^\varepsilon, (s, s_I)) \geq \text{Val}(\mathbb{M}, s)$ holds trivially.

It is therefore enough to select ε_δ small enough that the chance of progressing away from the initial state s_I of the priority tracker before reaching a BSCC in the induced Markov chain happens with a probability below δ .

Note that the chance of reaching any BSCC *with* the priority tracker still in state s_I is naturally no bigger than the chance of reaching the BSCC *with or without* the



■ **Figure 5** The reduction of the PtB gadget for priorities 0-5 to the reachability problem

priority tracker being in state s_I . Therefore, with the two special cases from above, $\text{Val}(\mathbb{M}_{\mathcal{T}}^{\varepsilon}, (s, s_I)) \geq \text{Val}(\mathbb{M}, s) - \delta$ follows. (ε_{δ} can, for example, be chosen as δ divided by the expected number of transitions taken before reaching a BSCC.)

- **Bounding the limit from above.** In the second part of the proof we show

$$\text{Val}(\mathbb{M}_{\mathcal{T}}^{\varepsilon}, (s, s_I)) \leq \text{Val}(\mathbb{B}^{\varepsilon}, (s, s_I)) \leq \text{Val}(\mathbb{B}, (s, s_I)) + \delta = \text{Val}(\mathbb{M}, s) + \delta.$$

For the first inequality, note that \mathbb{B}^{ε} is similar to $\mathbb{M}_{\mathcal{T}}^{\varepsilon}$, except in the nondeterministic vs. probabilistic transitions from state s_I . Therefore, the priority tracker gadget can be interpreted as what one gets when the player uses a particular positional randomized strategy to resolve the nondeterminism in \mathbb{B}^{ε} . As this is one possible strategy to resolve the nondeterminism, the inequality follows.

The equation follows from the fact that the PtB is good for MDPs [18], because it is a suitable limit-deterministic Büchi automaton [16, 31] that recognizes the words of priorities, where the highest priority that occurs infinitely often is odd.

To establish the middle inequality, consider the gadget \mathbb{B}^{ε} that is obtained by composing the gadget in Figure 5 with the MDP. Hahn et al. [17, Lemma 2] showed for such SLDBA that, for every δ' , there exists ε_0 such that, for all $\varepsilon < \varepsilon_0$, we have that

$$\text{Val}(\mathbb{B}, (s, s_I)) - \delta' \leq \text{Val}(\mathbb{B}^{\varepsilon}, (s, s_I)) \leq \text{Val}(\mathbb{B}, (s, s_I)) + \delta' \quad (3)$$

holds; this provides the second inequality. ◀

While the previous theorem proves that the priority-tracker works in the case of Max-MDPs, unfortunately this reduction cannot be used for general stochastic games, or even Min-MDPs, where (just as the nondeterminism in the MDP and the PtB gadget are resolved by different players) Min can gain an unfair advantage from knowing the state of the priority tracker gadget, as demonstrated by the following lemma.

► **Lemma 10.** *The priority tracker construction is incorrect for stochastic parity games, even when restricted to Min-MDPs.*

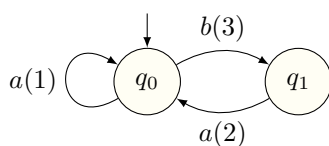
Proof. The game of Figure 6 shows that the priority-tracker should not be used for Min-MDPs (or generally for $2\frac{1}{2}$ -player games). The example game is a Min-MDP: Min is the only player who makes moves. Yet, Max wins because the highest recurring priority is either 3 or 1. If, however, Min chooses b from q_0 until the priority-tracker enters state s_2 , and then switches to action a , the highest recurring priority is intuitively deemed to be even (as it is not 3, while the priority tracker is in the component that intuitively checks if it is 2 or 3) and Min is adjudicated the game, because Max cannot reach the reachability target s_a . ◀

5 Experimental Results

Our reduction from stochastic parity games to stochastic reachability games can be done on the fly because the augmentation of the game graph only requires knowledge of the current priority. This enables us to use model-free RL to solve the stochastic reachability game that results from our reduction. We use minimax Q-learning for alternating Markov Games [26] to compute strategies. By assigning an undiscounted reward of +1 for reaching s_a (and 0 reward otherwise), the values of state-action pairs computed by Q-learning are direct estimates of the probability of reaching s_a , and hence of the probability to satisfy the property.

The models we use to test our reduction [27] are in Table 1. This table presents the name of the game, the number of states of the game, the priorities in the game, the probability to satisfy the property by player Max when both players play optimally, and the probability as estimated by Q-learning. As a verifying step, we fixed Max player’s strategy after learning and solved the resulting stochastic parity game with MUNGOJERRIE [17]. The resulting probability to satisfy the property is presented in the table. We also show the time (in seconds) that it took for Q-learning to compute the strategies, and the value of ϵ used in learning. Each episode was run until it terminated by a transition to a sink.

In `coprobActive` [1], a cop and a robber take turns, deterministically moving to adjacent vertices on a house-shaped graph. Cop is player Max and robber is player Min. The ω -regular property for our game is to eventually reach a state where the cop and robber are at the same vertex. Both players select their starting vertex in the first move of the game, starting with the cop’s selection. In this game, the cop has a winning strategy. `coprobPassive` is identical except the robber has an additional action where she does not move. In this variant, the robber has a winning strategy. `coprobActiveP` and `coprobPassiveP` are identical to the prior two models except that moves are only successful with probability 1/2. The state will remain unchanged if the move is unsuccessful. The cop has a winning strategy in both of these models. The `randomME` model is a randomized mutual exclusion protocol [3, p. 836], modified to allow simultaneous requests by the two clients. Player Max controls the arbiter, while player Min controls both clients. The objective of the game is to guarantee absence of



■ **Figure 6** A parity game. Both states are controlled by player Min.

■ **Table 1** Q-learning results for games. Estimated probabilities, verifying probabilities, and times are the average of three runs. We tuned hyperparameters individually for each experiment.

Name	states	priorities	prob.	estim. prob.	verify prob.	time (s)	ϵ
<code>coprobActive</code>	104	0,1	1	0.99	1	1.13	0.05
<code>coprobPassive</code>	105	0,1	0	0	0	0.46	0.05
<code>coprobActiveP</code>	105	0,1	1	0.99	1	2.97	0.03
<code>coprobPassiveP</code>	105	0,1	1	0.99	1	4.01	0.03
<code>coprobSafe</code>	148	0,1	1	0.99	1	3.50	0.03
<code>coprobSafeP</code>	150	0,1	13/15	0.85	0.86	13.57	0.03
<code>randomME</code>	30	1,2,3	1	0.95	1	4.43	0.04
<code>harding</code>	6	0,1,2	1	0.96	1	2.88	0.04
<code>smg1</code>	8	0,1	1	0.97	1	2.91	0.02
<code>difference</code>	99241	0,1	1	0.92	1	12.23	0.1
<code>ttt</code>	6321	0,1	1	1	1	1.57	0.07
<code>coins</code>	38200	0,1	1	0.97	1	2.92	0.05
<code>penney</code>	1745	0,1	1/3	0.33	0.33	0.28	0.1
<code>robots</code>	45784	0,1	1	1	0.94	1031.29	0.003

starvation for one client. The `harding` example [20] shows that the use of nondeterministic automata to express ω -regular objectives for games with two strategic players may lead to incorrect results. In `smg1` messages are exchanged between a server and a client [24]. In `difference` [39], the Max player chooses digits and the Min players assigns them to places in two two-digit numbers, x and y . The goal of Player Max is to guarantee $x - y \geq 40$. Example `ttt` is a model of the tic-tac-toe game. In `coins` [39], the two players remove in turn a coin from one end of a row of 4 coins. Player Max tries to collect coins worth at least as much as the coins collected by Player Min. In `penney` [28], each player chooses a sequence of three heads and tails. A fair coin is then tossed repeatedly, and the first player whose sequence turn up wins. In `robots` [19], two robots, each controlled by one player, navigate a grid world. Table 1 shows a strong correlation between the value of ϵ needed to reliably learn an optimal strategy and the learning time.

The reduction of Section 4 from parity objectives to reachability objectives for Markov decision processes can be done on the fly because it only requires knowledge of the current priority. As before, we can use model-free RL to solve the resulting reachability objective.

In Table 2 we compare 3 methods that use Q-learning to learn a strategy that maximizes the probability of satisfying a parity objective in a MDP. In Method 1, we translate the parity objective into a SLDBA objective and use the reduction from [17]. In Method 2, we treat our MDP as a stochastic game (with only one player) and utilize the reduction from Section 3. In Method 3, we use the reduction introduced in Section 4. We tuned the hyperparameters to minimize time subject to the following constraints. First, the strategies produced, as verified by the model checker, satisfied the property with the maximum probability. Second, since each method produces an estimate of the probability of the satisfaction of the property, the estimated probability of satisfaction was within 10% of the true value.

The examples `deferred` and `chocolates` have properties where the learner has many opportunities to transition the SLDBA to its final accepting region. The difficulty here is that the learner must learn to wait to make this transition, which happens with low probability during the initial phase of Q-learning where the learner explores randomly. Methods 2 and 3 perform better in these examples because there is no additional choice for the learner to

■ **Table 2** Q-learning results for MDPs. The Q-table for each experiment was initialized to zero. The number of states with the SLDBA objective is listed first, followed by the parity objective.

Name	states	priorities	Meth. 1 time (s)	Meth. 2 time (s)	Meth. 3 time (s)
deferred	74,25	1,2	3.63	2.73	0.52
trafficNtk	392,773	0,1,2	0.45	1.87	1.88
chocolates	7168,1034	1,2	1523.89	13.21	8.14
shoot1	1175,595	0,1,2	0.36	> 2000	33.26
agridGR2	252,216	0-5	25.59	58.26	13.97

learn. In `shoot1` and `trafficNtk`, all methods are able to produce strategies that satisfy the property with the maximum probability relatively easily. However, Methods 2 and 3 require small values of ϵ in order for the estimated probabilities to be close to their true values, increasing the learning time. In `agridGR2`, the large number of priorities is harmful to Method 2's performance due to the increasing powers of ϵ . Throughout each of these experiments, Method 3 outperforms Method 2 and is competitive with Method 1.

6 Conclusion

We have presented a reduction from stochastic parity games to stochastic reachability games that allows one to apply model-free reinforcement learning to the computation of the game values and optimal strategies. We have also described a translation that, while only suitable for $1\frac{1}{2}$ -player games – more precisely, for Max-MDPs – requires shorter training episodes than the more general reduction. Initial experiments show that the proposed approach allows an off-the-shelf reinforcement learning algorithm like minimax Q-learning to compute optimal strategies for games of moderate size.

References

- 1 M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8:1–12, 1984.
- 2 D. Andersson and Miltersen P. B. The complexity of solving stochastic games on graphs. In *Algorithms and Computation*, pages 112–121, 2009.
- 3 C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
- 4 V. S. Borkar and S. P. Meyn. The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.
- 5 K. Chatterjee and N. Fijalkow. A reduction from parity games to simple stochastic games. In *Games, Automata, Logics and Formal Verification, GandALF*, pages 74–86, June 2011.
- 6 K. Chatterjee and T. A. Henzinger. Reduction of stochastic parity to stochastic mean-payoff games. *Inf. Process. Lett.*, 106(1):1–7, 2008.
- 7 K. Chatterjee, M. Jurdziński, and T. A. Henzinger. Simple stochastic parity games. In *Computer Science Logic (CSL)*, pages 100–113, 2003.
- 8 K. Chatterjee, M. Jurdziński, and T. A. Henzinger. Quantitative stochastic parity games. In *Symposium on Discrete Algorithms, SODA*, pages 121–130, 2004.
- 9 A. Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992.
- 10 C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *J. ACM*, 42(4):857–907, July 1995.
- 11 L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1998.
- 12 J. Fu and U. Topcu. Probably approximately correct MDP learning and control with temporal logic constraints. In *Robotics: Science and Systems*, July 2014.
- 13 I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

- 14 C. M. Grinstead and J. L. Snell. *Introduction to Probability*. Amer. Math. Soc., 1997.
- 15 A. Guez et al. An investigation of model-free planning. *CoRR*, abs/1901.03559, 2019. [arXiv:1901.03559](https://arxiv.org/abs/1901.03559).
- 16 E. M. Hahn, G. Li, S. Schewe, A. Turrini, and L. Zhang. Lazy probabilistic model checking without determinisation. In *Concurrency Theory, (CONCUR)*, pages 354–367, 2015.
- 17 E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak. Omega-regular objectives in model-free reinforcement learning. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 395–412, 2019. LNCS 11427.
- 18 E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak. Good-for-MDPs automata for probabilistic analysis and reinforcement learning. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 306–323, 2020. LNCS 12078.
- 19 E. M. Hahn, S. Schewe, A. Turrini, and L. Zhang. A simple algorithm for solving qualitative probabilistic parity games. In *Computer Aided Verification, Part II*, pages 291–311, 2016. LNCS 9780.
- 20 A. Harding, M. Ryan, and P.-Y. Schobbens. A new algorithm for strategy synthesis in LTL games. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2005)*, pages 477–492, Edinburgh, UK, 2005. LNCS 3440.
- 21 T. A. Henzinger and N. Piterman. Solving games without determinization. In *15th Conference on Computer Science Logic*, pages 394–409, Szeged, Hungary, September 2006. LNCS 4207.
- 22 Alex Irpan. Deep reinforcement learning doesn’t work yet. <https://www.alexirpan.com/2018/02/14/r1-hard.html>, 2018.
- 23 J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Van Nostrand, 1960.
- 24 M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Computer Aided Verification (CAV)*, pages 585–591, July 2011. LNCS 6806.
- 25 M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 157–163, 1994.
- 26 M. L. Littman and C. Szepesvári. A generalized reinforcement-learning model: Convergence and applications. In *International Conference on Machine Learning*, pages 310–318, 1996.
- 27 Stochastic parity game reinforcement learning benchmarks. <https://github.com/cuplv/parityRLBenchmarks>, 2020.
- 28 W. Penney. Problem 95. Penney-ante. *Journal of Recreational Mathematics*, 2(4):241, 1969.
- 29 D. Perrin and J.-É. Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Elsevier, 2004.
- 30 L. S. Shapley. Stochastic games. *Proc. Nat. Acad. Sci. U.S.A.*, 39:1095–1100, 1953.
- 31 S. Sickert, J. Esparza, S. Jaax, and J. Křetínský. Limit-deterministic Büchi automata for linear temporal logic. In *Computer Aided Verification (CAV)*, pages 312–332, 2016. LNCS 9780.
- 32 D. Silver et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, January 2016.
- 33 A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman. PAC model-free reinforcement learning. In *International Conference on Machine Learning, ICML*, pages 881–888, 2006.
- 34 R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, second edition, 2018.
- 35 M. Y. Vardi. Automatic verification of probabilistic concurrent finite state programs. In *Foundations of Computer Science*, pages 327–338, 1985.
- 36 Christopher J. C. H. Watkins and Peter Dayan. Q-learning. In *Machine Learning*, pages 279–292, 1992.
- 37 M. Wen and U. Topcu. Probably approximately correct learning in stochastic games with temporal logic specifications. In *IJCAI*, pages 3630–3636, 2016.
- 38 E. Wiewiora. Reward shaping. In *Encyclopedia of Machine Learning*, pages 863–865. Springer, 2010.
- 39 P. Winkler. *Mathematical Puzzles: A Connoisseur’s Collection*. A K Peters, 2004.