

# Optimal Strategy Synthesis for Sure–Almost–Sure Stochastic Parity Games

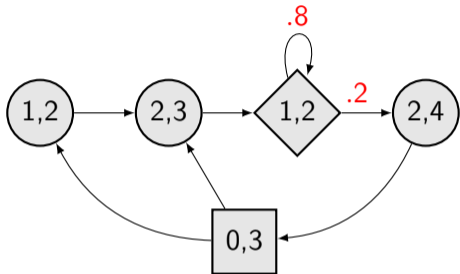
Laurent Doyen<sup>1</sup> and Shibashis Guha<sup>2</sup>

<sup>1</sup>Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF

<sup>2</sup>Tata Institute of Fundamental Research, Mumbai

STACS 2026

## Sure-almost-sure stochastic parity game



○ Player 1 vertices    □ Player 2 vertices    ◇ Probabilistic vertices

- Turn-based stochastic parity games.
- Two parity conditions (maximum priority visited infinitely often is even):
  - **Sure**: satisfied on all plays.
  - **Almost-sure**: satisfied with probability 1.
- Goal: synthesize optimal strategies satisfying both.

## Related Work (Sure-almost-sure parity game)

### MDPs

- Has been studied for the special case of Markov decision processes (MDPs) ([Berthon, Randour, Raskin '19](#))
- Markov decision processes are special cases of stochastic games where there are only  $\bigcirc$  (Player 1) vertices and  $\diamond$  probabilistic vertices, and there is *no*  $\square$  (Player 2) vertex.

### Stochastic games

- The problem is coNP-complete and these games are determined ([Chatterjee, Piterman '19](#))
- Player 2 guesses a *memoryless* winning strategy.
- Does not provide any idea about how Player 1 needs to play to satisfy the specification.

## Our contributions

1. Provide an algorithm to compute the **winning set** of the system player (Player 1).
2. Designing a **winning strategy** for Player 1.  
This also leads to considering the **special cases**: computational complexity and memory requirements (fixed number of priorities, Büchi, coBüchi)
3. A **new construction of a parity game** equivalent to the intersection of two parity games with a blow up that is exponential only in either of the parity indices.  
Crucial to prove that the synthesis problem is in  $NP \cap coNP$  when one of the parity conditions has a fixed index.

## Strategy construction for sure-almost sure parity in MDPs

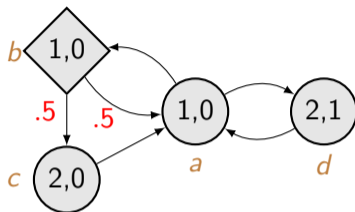


Figure: MDP Example (Berthon, Randour, Raskin' 17)

- Player 1 has a strategy  $\sigma$  to satisfy parity 1 surely and parity 2 almost-surely (with probability 1).
- Strategy plays in infinite sequence of rounds.
- In round  $i$ , in state  $a$ , the strategy chooses  $b$  for  $n_i$  steps, such that the probability to reach  $c$  during round  $i$  is larger than  $1 - 2^{-i}$ .
- If during round  $i$ ,  $c$  is not reached, then  $\sigma$  goes to  $d$  once.

## Strategy construction for sure-almost sure parity in MDPs

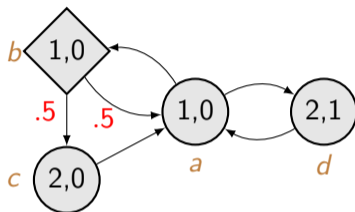


Figure: MDP Example (Berthon, Randour, Raskin' 17)

The strategy  $\sigma$  of Player 1 is an **infinite memory strategy**.

It satisfies surely Büchi( $\{c, d\}$ ) and almost-surely coBüchi( $\{a, b, c\}$ ).

Further, no finite memory strategy of Player 1 exists here.

# Strategy construction for sure-almost sure parity in stochastic games

In MDPs, the construction of a strategy heavily depends on the existence of so-called *end-components*.

## Strategy for Player 1 in stochastic games

- Start by playing an **almost-sure winning strategy** for Player 1 for the objective  $\text{Parity}(\Omega_1) \cap \text{Parity}(\Omega_2)$ .
- $\text{Parity}(\Omega_1)$  can be violated with probability 0.
- Switch from time to time to a strategy to visit a large priority for  $\Omega_1$  which may though ignore the second priority.
- Switching **should be rare** thereby probability of violating second priority should be 0.
- Switching guarantees that all outcomes satisfy  $\text{Parity}(\Omega_1)$  surely.

## Switching condition in stochastic games

A **long sequence of odd priorities** is seen without seeing any larger priority.

Given odd priority  $p$  and integer  $N$ , let

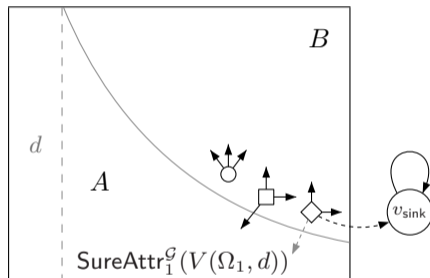
$$\text{Unlucky}(p, N) = \{\rho \in V^\omega \mid \text{the first } N \text{ elements of } \pi_{\Omega_1 \geq p}(\rho) \text{ are } p\}$$

be the event that at least  $N$  priorities  $\geq p$  occur and the first  $N$  such priorities are  $p$ .

3	3	3	3	3	3	3	
---	---	---	---	---	---	---	--

Unlucky(3, 7)

## Winning region of Player 1 with an even largest $\Omega_1$ -priority



Decomposition of a winning region with an even largest  $\Omega_1$ -priority  $d$ , into the sure attractor  $A$  (for Player 1) to  $d$  and the subgame closure of  $B = V \setminus A$ .

## Winning strategy of Player 1 with an even largest $\Omega_1$ -priority

When the largest  $\Omega_1$ -priority is even, **all vertices in  $\mathcal{G}$  are sure-almost-sure winning** for Player 1 for the parity objectives  $\text{Parity}(\Omega_1)$  and  $\text{Parity}(\Omega_2)$  if and only if Player 1 has an

- an **almost-sure winning strategy**  $\sigma_{AS}$  for Player 1 from every vertex of  $\mathcal{G}$ , for objective  $\text{Parity}(\Omega_1) \cap \text{Parity}(\Omega_2)$ ;
- a **sure-almost-sure winning strategy**  $\sigma_{sub}$  for Player 1 from every vertex in  $[\mathcal{G}]_B$  for the parity objectives  $\text{Parity}(\Omega_1)$  and  $\text{Parity}(\Omega_2)$  in the subgame closure  $[\mathcal{G}]_B$ , where  $B = V \setminus A$  and  $A = \text{SureAttr}_1(V(\Omega_1, d))$ .

## Winning strategy of Player 1

When its largest  $\Omega_1$ -priority is even.

$\tau \in V^* V_1$ : a history,  $k_p$ : a pointer to a position in the history,  $i_p$ : phase number associated with priority  $p$ ,  $N_{i_p}$ : Probability of Unlucky( $p, N_{i_p}$ ) is less than  $2^{-i_p}$ .  
Initially unlucky =  $\perp$  (false) and  $k_p = i_p = 0$  for all odd  $p$ .

### Strategy

1. if  $\Omega_1(\text{Last}(\tau)) = d$ , then set unlucky =  $\perp$  and play  $\sigma_{AS}(\tau)$ ;
2. if unlucky =  $\perp$ ,
  - a. if  $|\pi_{\Omega_1 \geq p}(\tau(k_p \triangleright))| < N_{i_p}$  for all odd  $p$ , then play  $\sigma_{AS}(\tau)$ ;
  - b. else, for each odd  $p$  such that  $|\pi_{\Omega_1 \geq p}(\tau(k_p \triangleright))| \geq N_{i_p}$ ,
    - i. if the first  $N_{i_p}$  elements of  $\pi_{\Omega_1 \geq p}(\tau(k_p \triangleright))$  are all  $p$ 's, then set unlucky =  $\top$ ;
    - ii. increment  $i_p$ , and set  $k_p = |\tau|$ ;

5	3	5	5	3	5	3	3
---	---	---	---	---	---	---	---

Not Unlucky(3, 4) but Unlucky(5, 4)

## Winning strategy of Player 1 with an even largest $\Omega_1$ -priority

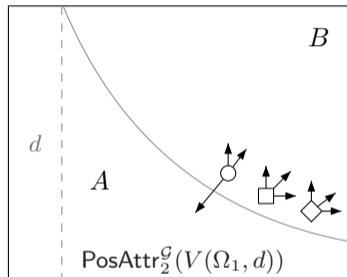
The sure-almost-sure winning region  $W$  (for Player 1) when its largest  $\Omega_1$ -priority is even.

The variable  $k_p$  is a pointer to a position in the history, and  $i_p$  is the phase number associated with priority  $p$ . Initially  $\text{unlucky} = \perp$  (false) and  $k_p = i_p = 0$  for all odd  $p$ .

History  $\tau \in V^* V_1$ :

3. if  $\text{unlucky} = \perp$ , then play  $\sigma_{AS}(\tau)$ ;
4. if  $\text{unlucky} = \top$ ,
  - a. if  $\text{Last}(\tau) \in B$ , then play  $\sigma_{sub}(\tau')$  where  $\tau'$  is the longest suffix of  $\tau$  that is entirely in  $B$ ;
  - b. else  $\text{Last}(\tau) \in A$ , and play  $\sigma_{Attr}(\tau)$ .

## Winning region of Player 2 with an odd largest $\Omega_1$ -priority



Decomposition of a winning region with an odd largest  $\Omega_1$ -priority  $d$ , into the positive attractor  $A$  (for Player 2) to  $d$  and the subgame induced by  $B = V \setminus A$ .

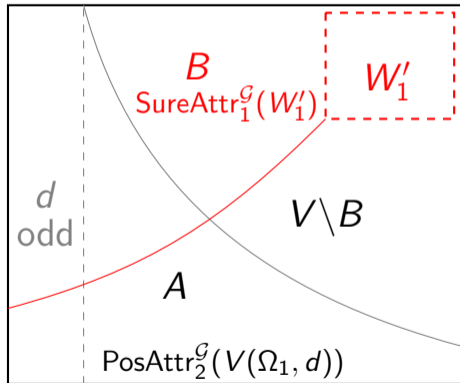
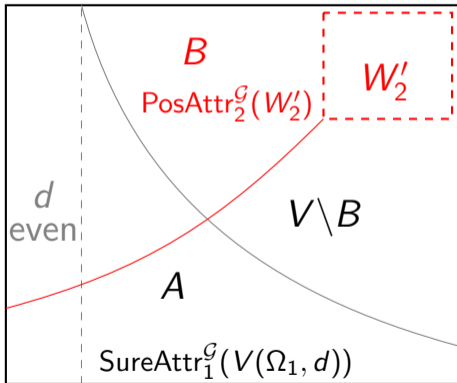
## Other cases

- Largest  $\Omega_1$ -priority is odd in the winning region
- Largest  $\Omega_1$ -priority is even in the spoiling region

Solved by recursive calls on subgames with a smaller number of  $\Omega_1$ -priorities.

## Zielonka-style algorithm for sure-almost-sure synthesis problem

For even  $d$ , restrict  $\mathcal{G}$  to  $\mathcal{G}|_{W_{AS}}$  where  $W_{AS}$  is the almost-sure winning region for  $\text{Parity}(\Omega_1) \cap \text{Parity}(\Omega_2)$ .



# Zielonka-style algorithm for sure-almost-sure synthesis problem

## Time complexity

$\text{ASPG}(N, d)$ : time complexity of solving almost-sure parity games with  $N$  vertices and largest priority  $d$ .

Here  $N = n \cdot \sqrt{\min(d_1^{d_2}, d_2^{d_1})}$  and  $n$  is the number of vertices in the game and  $d = d_1 \cdot d_2$ .

Let  $T_n^{d_1}$  be the running time of the algorithm on a game with  $n$  vertices and largest  $\Omega_1$ -priority  $d_1$ .

$$T_n^{d_1} \in O(n^{d_1} \cdot \text{ASPG}(N, d_1 \cdot d_2)).$$

## Special Cases

**Table:** Memory requirement (0 stands for memoryless) and complexity bound for the special cases. New significant results appear in blue.

AS, $\longrightarrow$ S, $\downarrow$	<b>Büchi</b>	<b>coBüchi</b>	<b>Parity<sup>f</sup></b>	<b>Parity</b>
<b>Büchi</b>	finite, P	$\infty$ , P	$\infty$ , P	$\infty$ , NP $\cap$ coNP
<b>coBüchi</b>	0, P	0, P	0, P	0, NP $\cap$ coNP
<b>Parity<sup>f</sup></b>	finite, P	$\infty$ , P	$\infty$ , P	$\infty$ , NP $\cap$ coNP
<b>Parity</b>	finite, NP $\cap$ coNP	$\infty$ , NP $\cap$ coNP	$\infty$ , NP $\cap$ coNP	$\infty$ , coNP-C

## Special Cases

**Table:** Memory requirement (0 stands for memoryless) and complexity bound for the special cases. New significant results appear in blue.

AS, $\longrightarrow$ S, $\downarrow$	<b>Büchi</b>	<b>coBüchi</b>	<b>Parity<sup>f</sup></b>	<b>Parity</b>
<b>Büchi</b>	finite, P	$\infty$ , P	$\infty$ , P	$\infty$ , NP $\cap$ coNP
<b>coBüchi</b>	0, P	0, P	0, P	0, NP $\cap$ coNP
<b>Parity<sup>f</sup></b>	finite, P	$\infty$ , P	$\infty$ , P	$\infty$ , NP $\cap$ coNP
<b>Parity</b>	finite, NP $\cap$ coNP	$\infty$ , NP $\cap$ coNP	$\infty$ , NP $\cap$ coNP	$\infty$ , coNP-C

Thank you for your attention