Life is Random, Time is not

Markov Decision Processes with Window Objectives

Thomas Brihaye, ***Florent Delgrange***, Youssouf Oualhadj and Mickael Randour

1. Strategy synthesis

Finding good controllers for systems interacting with an environment

- **Game setting:** ensure a specified behavior against all possible strategies of the environment
- **Markov Decision Process (MDP) setting:**
  - **stochastic** environment
  - ensure a specified behavior with a sufficient probability
- **Classical objectives** reason about infinite runs *in their limit*
- **Window objectives in games:** ensure a good behavior in a parametrized time frame all along the run

$\longrightarrow$ *conservative approximations of classical objectives*

Aim of the work: *introducing **window objectives** in the **stochastic** context*

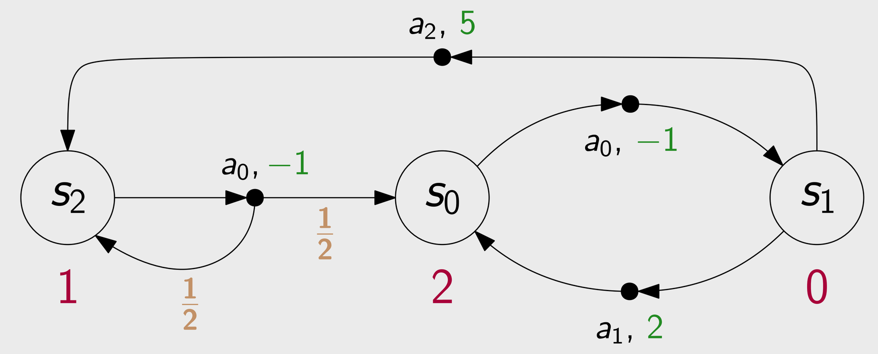
2. Markov Decision Processes

An MDP $\mathcal{M} = (S, A, \Delta)$ is a tuple such that

- S is the **set of states** of the system
- A is the **set of actions** of the system
- $\Delta : S \times A \to \mathcal{D}(S)$ is the **probability transition function**

It can be equipped with

- $w : A \to \mathbb{Z}$ a **weight function** or
- $p : S \to \{0, 1, \ldots, d\}$ a **priority function** ($d \leq |S| + 1$ w.l.o.g.)



3. Runs and strategies

- **Runs:** $\rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \cdots \in \text{Runs}(\mathcal{M})$

$\rightsquigarrow$ infinite sequences such that $\Delta(s_i, a_i)(s_{i+1}) > 0$

- **Strategy:** σ chooses at each step an action
  - **pure finite-memory strategies:** choose actions according to a finite amount of information gathered in the past
  - **pure memoryless strategies:** $\sigma : S \to A$

By fixing a strategy σ, a *Markov chain* is induced (fully stochastic process)

$\rightsquigarrow$ Let $E \subseteq \text{Runs}(\mathcal{M})$, $\mathbb{P}^{\sigma}_{\mathcal{M}}[E]$: *probability measure of the event E*

4. Classical objectives

- $\text{MeanPayoff} = \left\{ \rho = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots \in \text{Runs}(\mathcal{M}) \;\middle|\; \liminf_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n} a_i \geq 0 \right\}$

$\longrightarrow$ asks for the average weight per action to be positive in the limit

- $\text{Parity} = \{\rho \in \text{Runs}(\mathcal{M}) \mid \min_{s \in \inf(\rho)} p(s) = 0 \pmod 2\}$

$\longrightarrow$ asks for the minimum priority seen infinitely often to be even

$\rightsquigarrow$ canonical way of encoding ω-regular properties

$\sigma = \{s_2 \mapsto a_0,\ s_0 \mapsto a_0,\ s_1 \mapsto a_1\} \implies \mathbb{P}^{\sigma}_{\mathcal{M}}[\text{MeanPayoff}] = \mathbb{P}^{\sigma}_{\mathcal{M}}[\text{Parity}] = 1$

5. Window objectives

- Runs that exhibit **good behaviors** within a configurable **time frame**
- **Strengthen traditional objectives** (that only require correct behaviors at the limit)

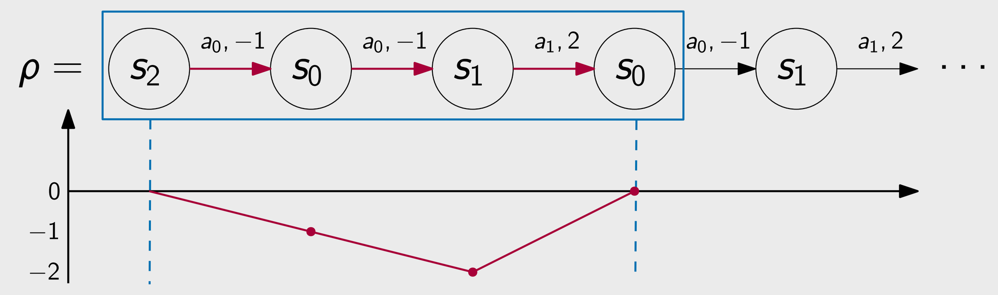
$\rightsquigarrow$ Make use of the **window formalism** to reason about **behaviors** in a given time bound

Good Window objective

$\text{GW}(\lambda) = \{\rho \in \text{Runs}(\mathcal{M}) \mid \textit{good behavior in at most } \lambda \textit{ steps from } s_0\}$

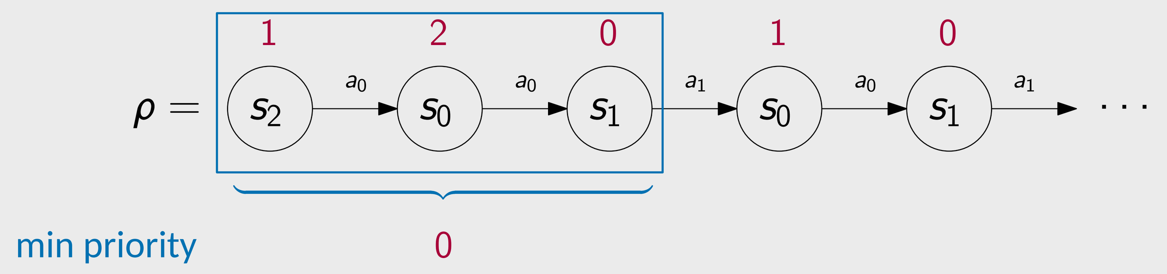
- **Window Mean-Payoff**

Positive sum in at most $\lambda = 3$ steps?



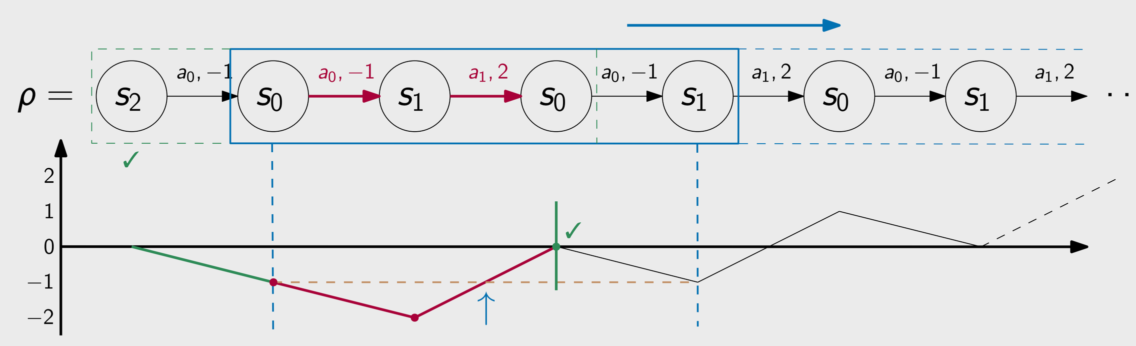
- **Window Parity**

Minimum priority even in at most $\lambda = 3$ steps?



Direct Fixed Window objective

- $\text{DFW}(\lambda) \equiv \Box\text{GW}(\lambda)$ $\rightsquigarrow$ good window from every position of the run
- Good window of maximal size λ **sliding along the run**

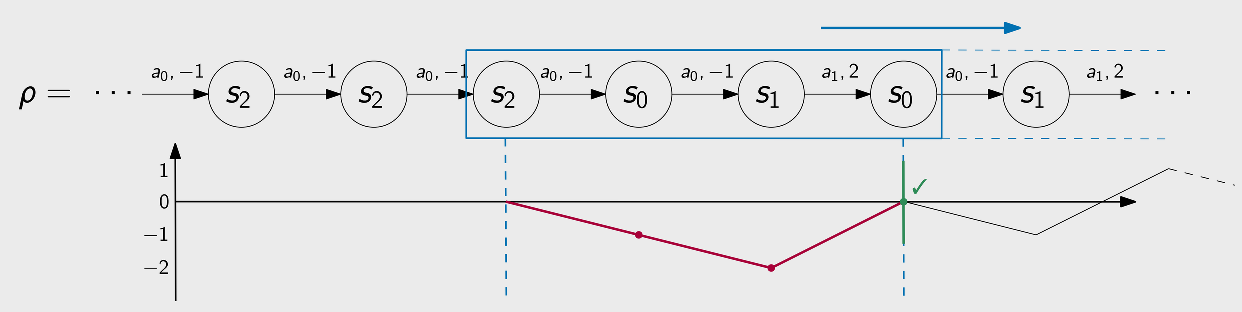


***Prefix independence*: Fixed and Bounded Window objectives**

- Window objectives from some point on
- **Fixed Window objective:** $\text{FW}(\lambda) \equiv \Diamond\text{DFW}(\lambda) \equiv \Diamond\Box\text{GW}(\lambda)$

$\rightsquigarrow$ reach a position from which all windows are good

- **Bounded Window objective:** $\text{BW} \equiv \exists \lambda > 0,\ \text{FW}(\lambda)$



6. Decision problem

Let $\mathcal{M} = (S, A, \Delta)$, $s \in S$, $\lambda > 0$, $\mathbb{O} \in \{\text{DFW}(\lambda), \text{FW}(\lambda), \text{BW}\}$ and $\alpha \in [0,1] \cap \mathbb{Q}$,

- **MDPs:** $\exists?\sigma,\ \mathbb{P}^{\sigma}_{\mathcal{M},s}[\mathbb{O}] \geq \alpha$
- **Games:** $\exists?\sigma,\ \forall \rho \in \text{Runs}^{\sigma}(\mathcal{M}), \rho \in \mathbb{O}$

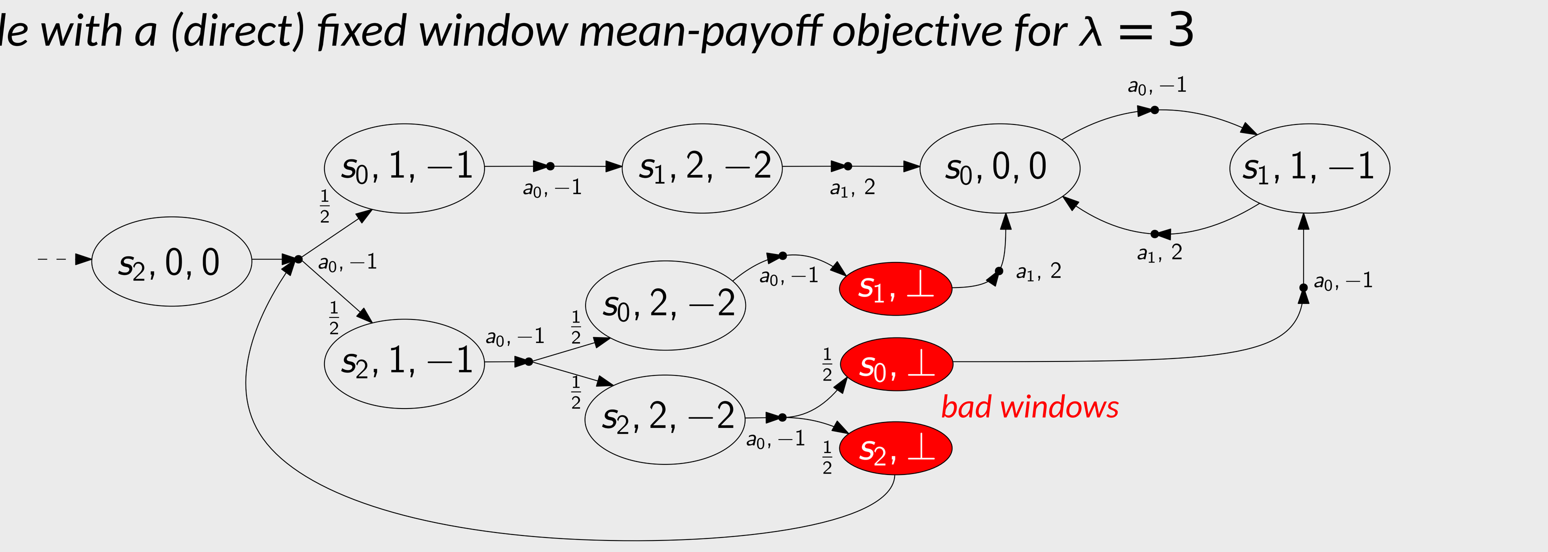
7. Fixed case: strategy requirements

- ***Pure finite-memory strategies*** are sufficient
- **Main tools:** natural reductions from DFW to **safety** and FW to **co-Büchi**

$\rightsquigarrow$ **unfolding** based on the maximal window size λ

- **Idea: incorporate weights (resp. priorities)** as well as the **current number of steps in the state space** of the MDP

Example with a (direct) fixed window mean-payoff objective for $\lambda = 3$



8. Prefix independent objectives: looking at the limit

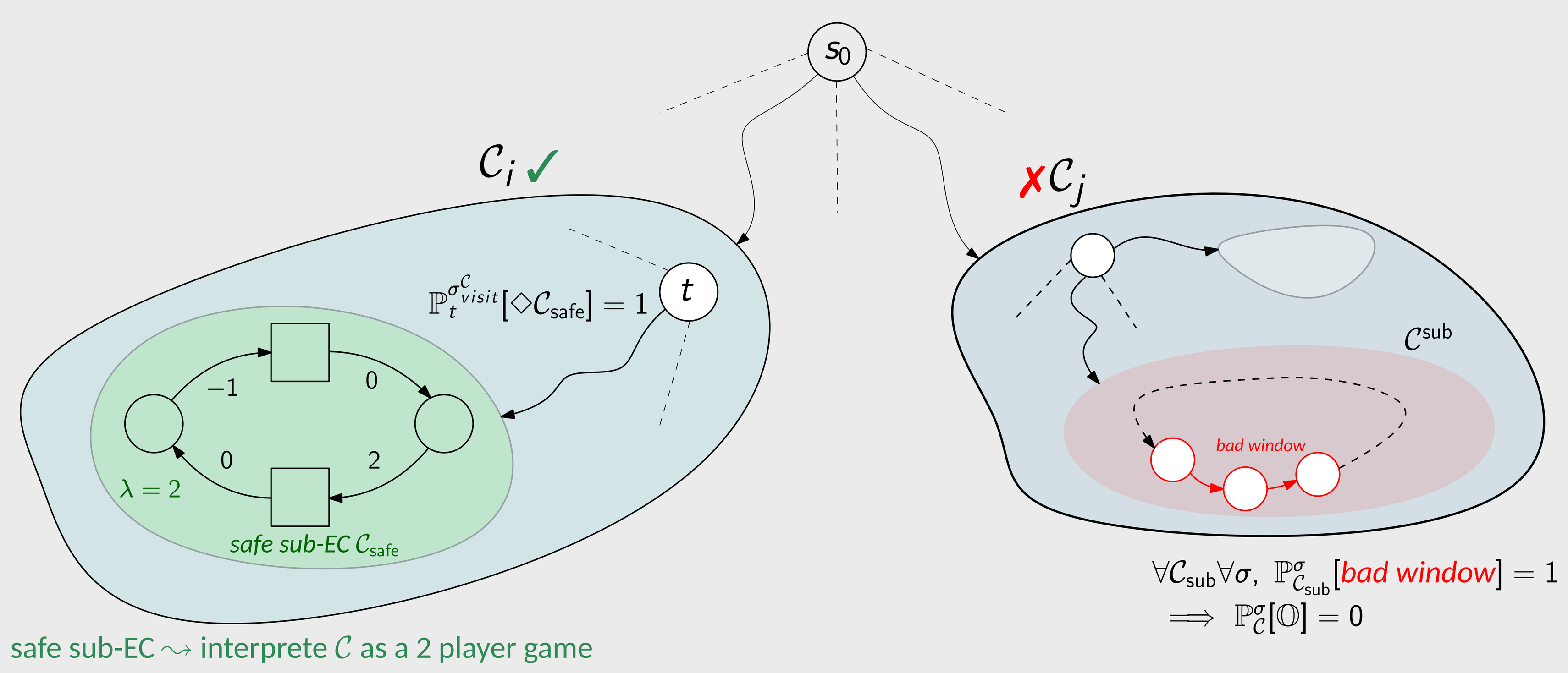
- **End-component (EC): strongly connected** sub-MDP $\mathcal{C}$ of $\mathcal{M}$ formed by **states and actions allowing to never leave** $\mathcal{C}$ $\rightsquigarrow \forall \sigma, \mathbb{P}^{\sigma}_{\mathcal{M}}[\{\rho \mid \lim(\rho) \in \text{EC}(\mathcal{M})\}] = 1$
- **Maximal end-component (MEC):** EC formed by a maximal union of non-disjoint ECs

$\rightsquigarrow$ $\text{MEC}(\mathcal{M})$ computable in polynomial time

MEC classification $\longrightarrow$ **for $\mathbb{O} \in \{\text{FW}(\lambda), \text{BW}\}$, 2 types of MECs: ✓ and ✗**

✓ $\forall s$ of $\mathcal{C}$, $\exists \sigma$, $\mathbb{P}^{\sigma}_{\mathcal{C},s}[\mathbb{O}] = 1$

✗ $\forall s$ of $\mathcal{C}$, $\forall \sigma$, $\mathbb{P}^{\sigma}_{\mathcal{C},s}[\mathbb{O}] = 0$



9. Complexity results and perspectives

	parity		mean-payoff	
	complexity	memory	complexity	memory
DFW	P-c.	polynomial	EXPTIME/PSPACE-h.	pseudo-polynomial
FW			P-c.	polynomial
BW		memoryless	NP ∩ coNP	memoryless

- Multi-objectives, expected window size
- *arXiv*: CoRR abs/1901.03571
- Tool support: extension of STORM