Model-based learning

the other RL technic

Guillaume Lozenguez

@imt-lille-douai.fr

IMT Lille Douai
École Mines-Télécom
IMT-Université de Lille

Model-based learning

Main Idea:

- Random trajectories (a lot)
- Until each transition is visited several times.
- Compute an optimal policy.

Potentially:

- Require driving exploration
- Only incomplete exploration can be performed

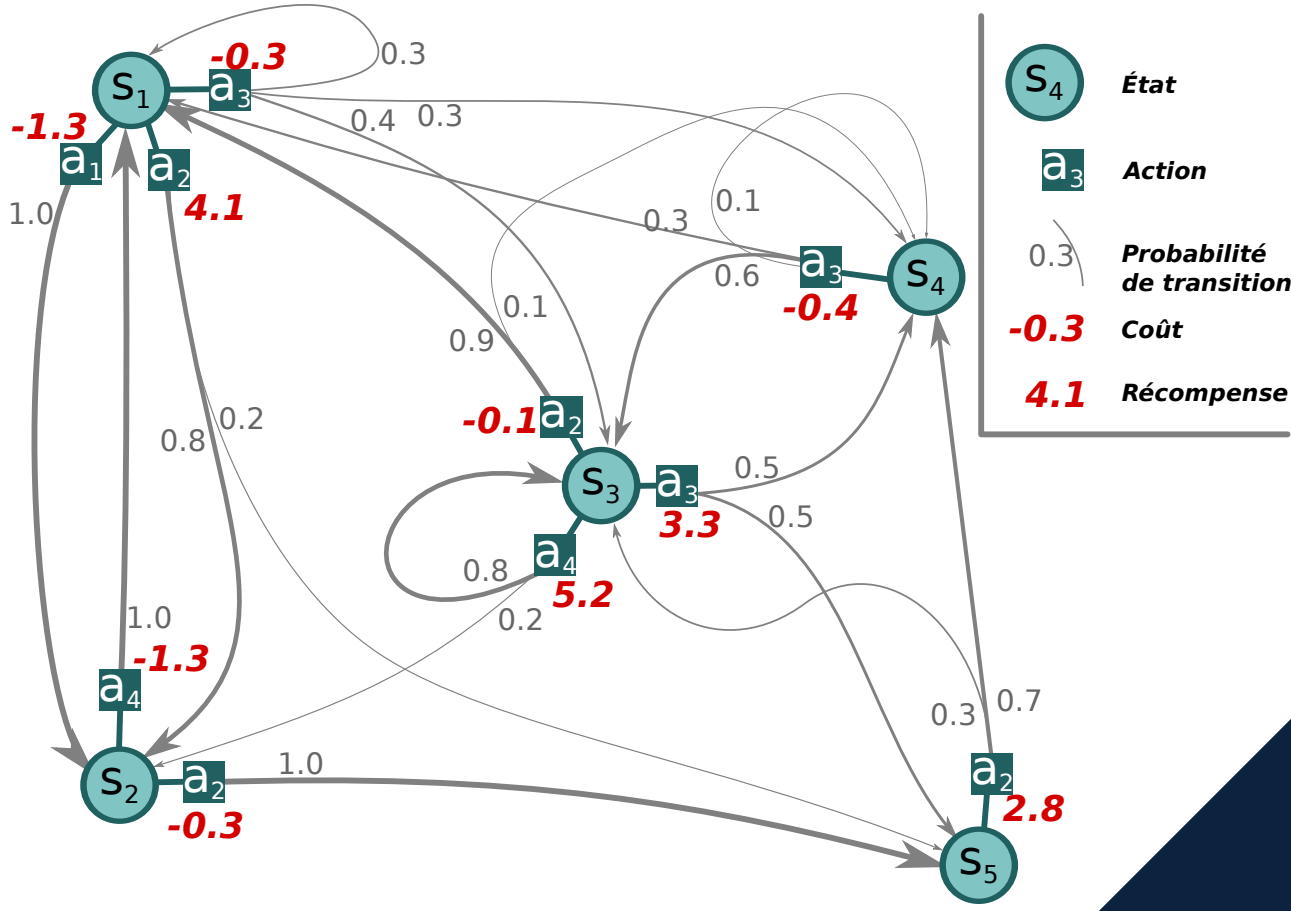
Markov Decision Process

MDP: $\langle S, A, T, R \rangle$:

S : set of system's states
A : set of possible actions
T : S × A × S → [0, 1] : transitions
R : S × A → R : cost/rewards

Optimal policy:

The policy π^* maximizing Bellman



Solving MDP: Value Iteration

Input: an **MDP:** $\langle S, A, T, R \rangle$; precision error: ϵ ; discount factor γ ; initial $V(s)$

1. Repeat until the **maximal delta** < ϵ

   For each state $s \in S$

   - Search the action a^* maximizing the Bellman Equation
   - Update $\pi(s)$ and $V(s)$ by considering action a^*
   - Compute the delta value between the previous and the new $V(s)$

Output: an optimal π^* and associated **V-values**.

$$V^{\pi}(s) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \times V^{\pi}(s')$$

Application to 421

- Python implementation - playerMDP.py :

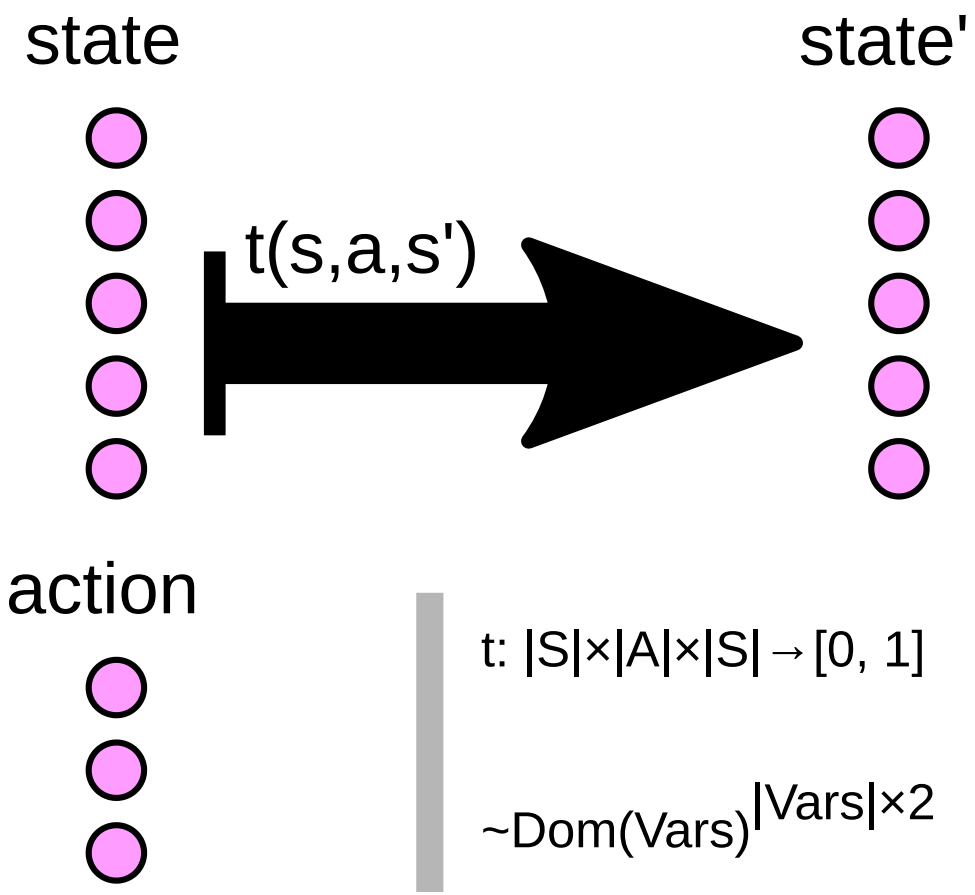
```
solver= MDP()
solver.learnModel( Engine() )
solver.valueIteration()

player= PiPlayer( solver.policy() )
```

- Learning phase: Estimate **t** and **r**:
  - *10 000* simulations for each couple (s, a)
- Value iteration:
  - *3* iterations (directed and finit game)
- Average score (100 000 games): **~338**

(To notice: decreasing the learning phase impact the average score)

The Curse of Dimensionality

Fonction de Transition:

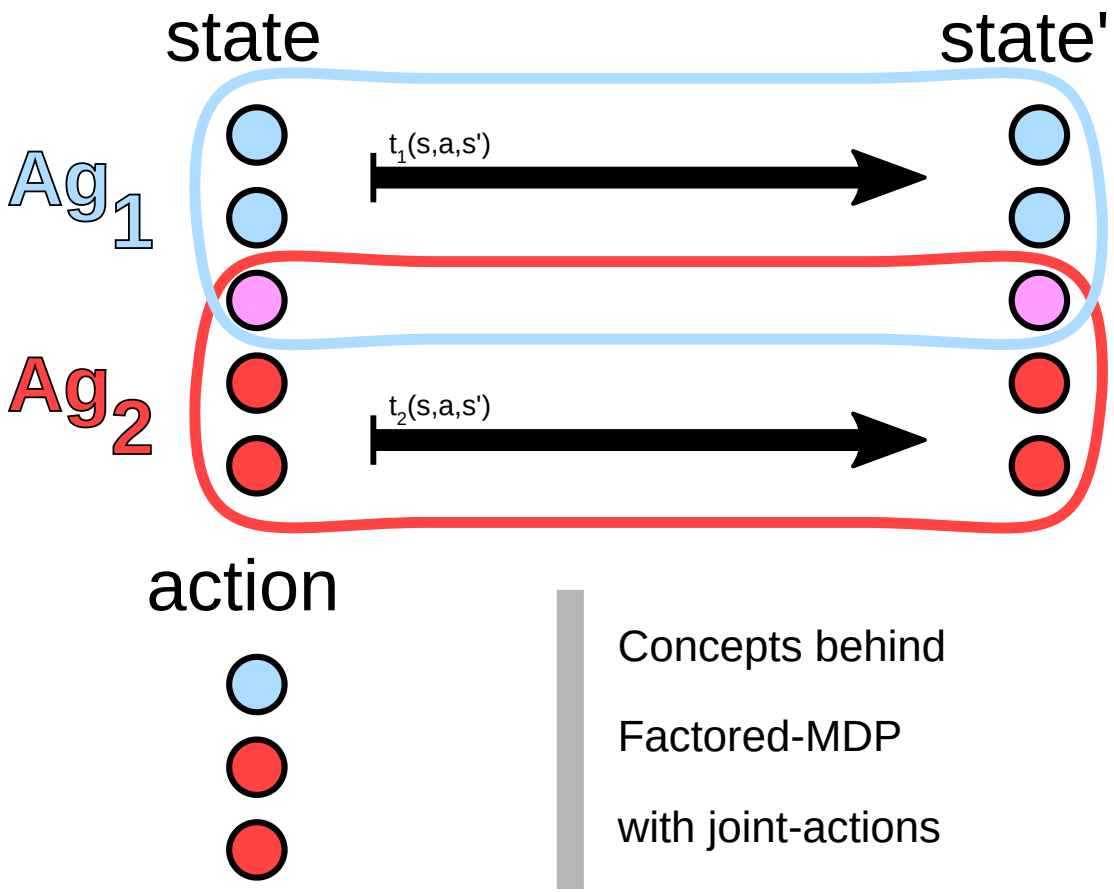


$t: |S| \times |A| \times |S| \rightarrow [0, 1]$

$\sim Dom(Vars)^{|Vars| \times 2}$

Le fléau de la dimension

Factored Transition function:



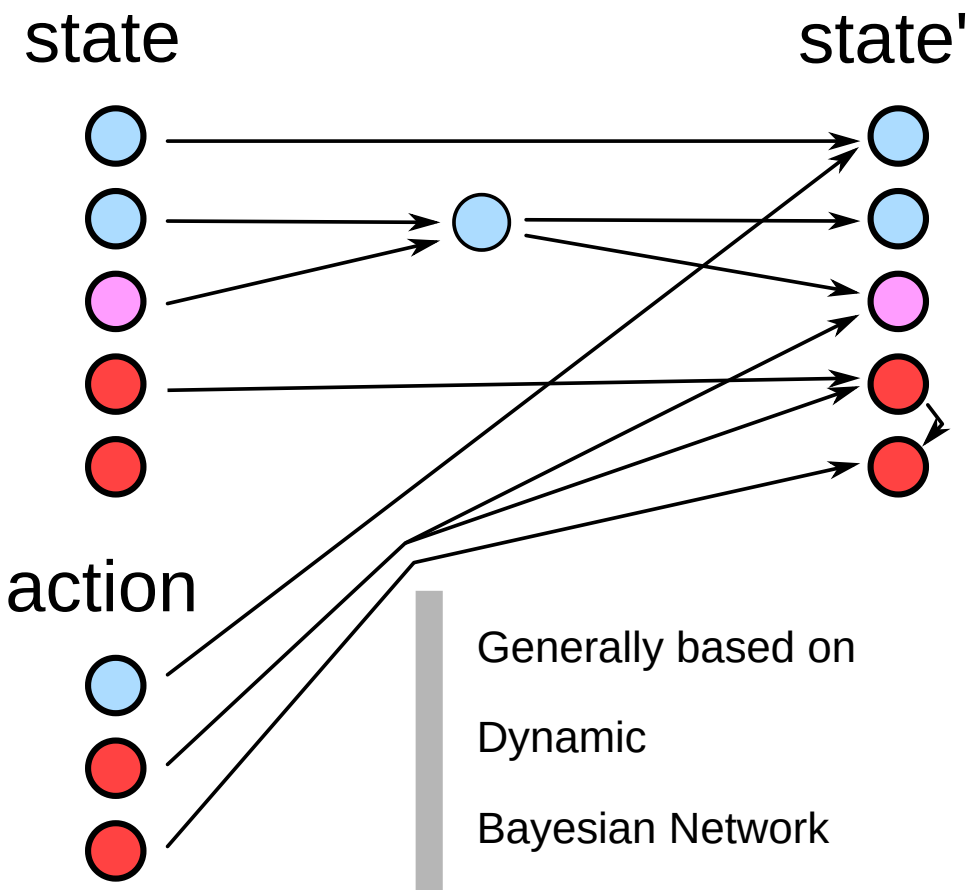
Concepts behind

Factored-MDP

with joint-actions

Le fléau de la dimension

Factored Transition function:



Example: Zombie Dice

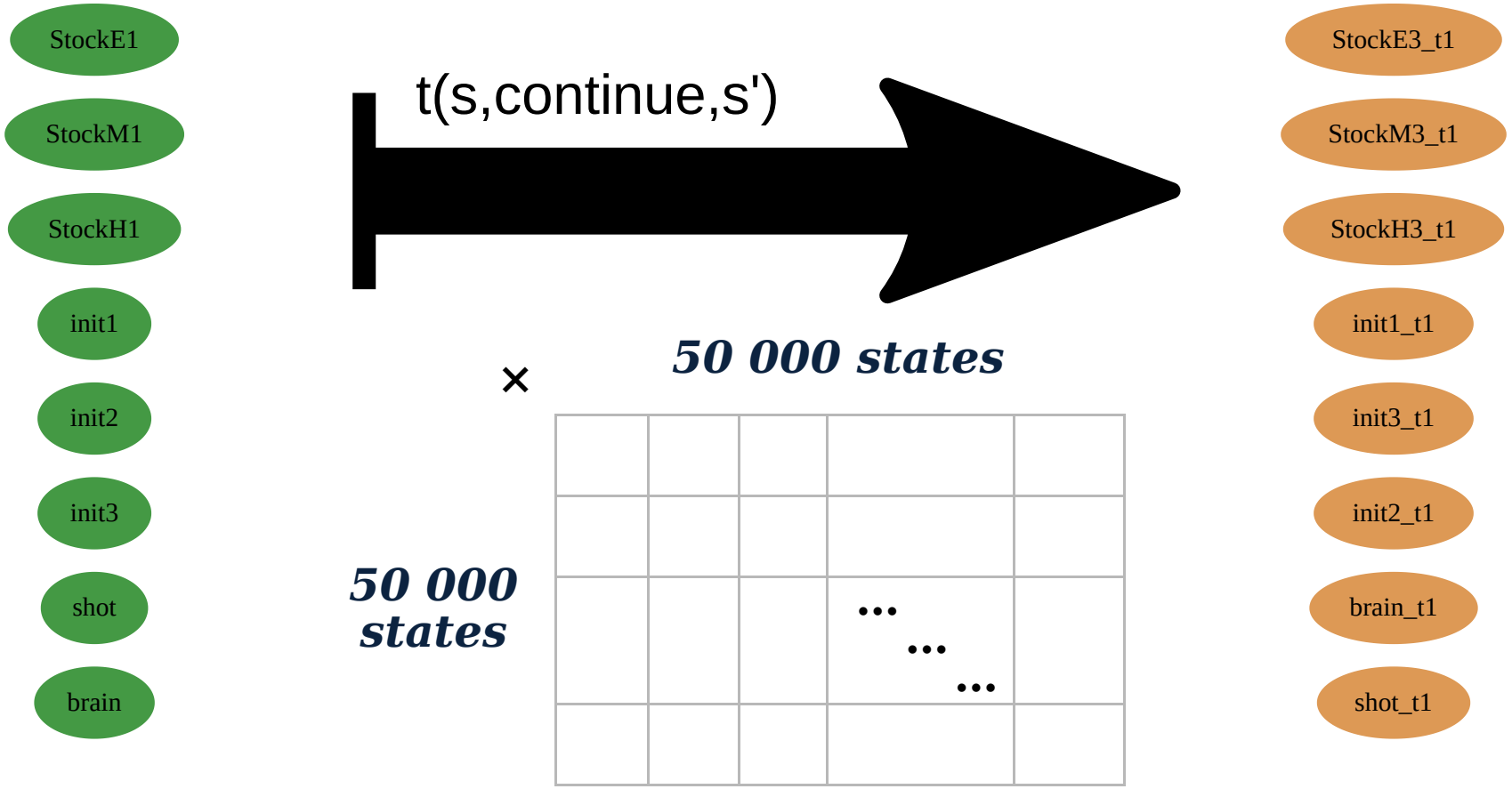
Eat maximum brains

without dying (3 damages)

- Players are zombies.
- They try to catch humans three at a time.
- Humans are dice with probability to fight back.

Example: Zombie Dice

Matrice complète



Example: Zombie Dice

Dynamic Bayesian Network (Continue)