

Monte Carlo Tree Search

Jomi Fred Hübner

<https://jomifred.github.io/ia/>

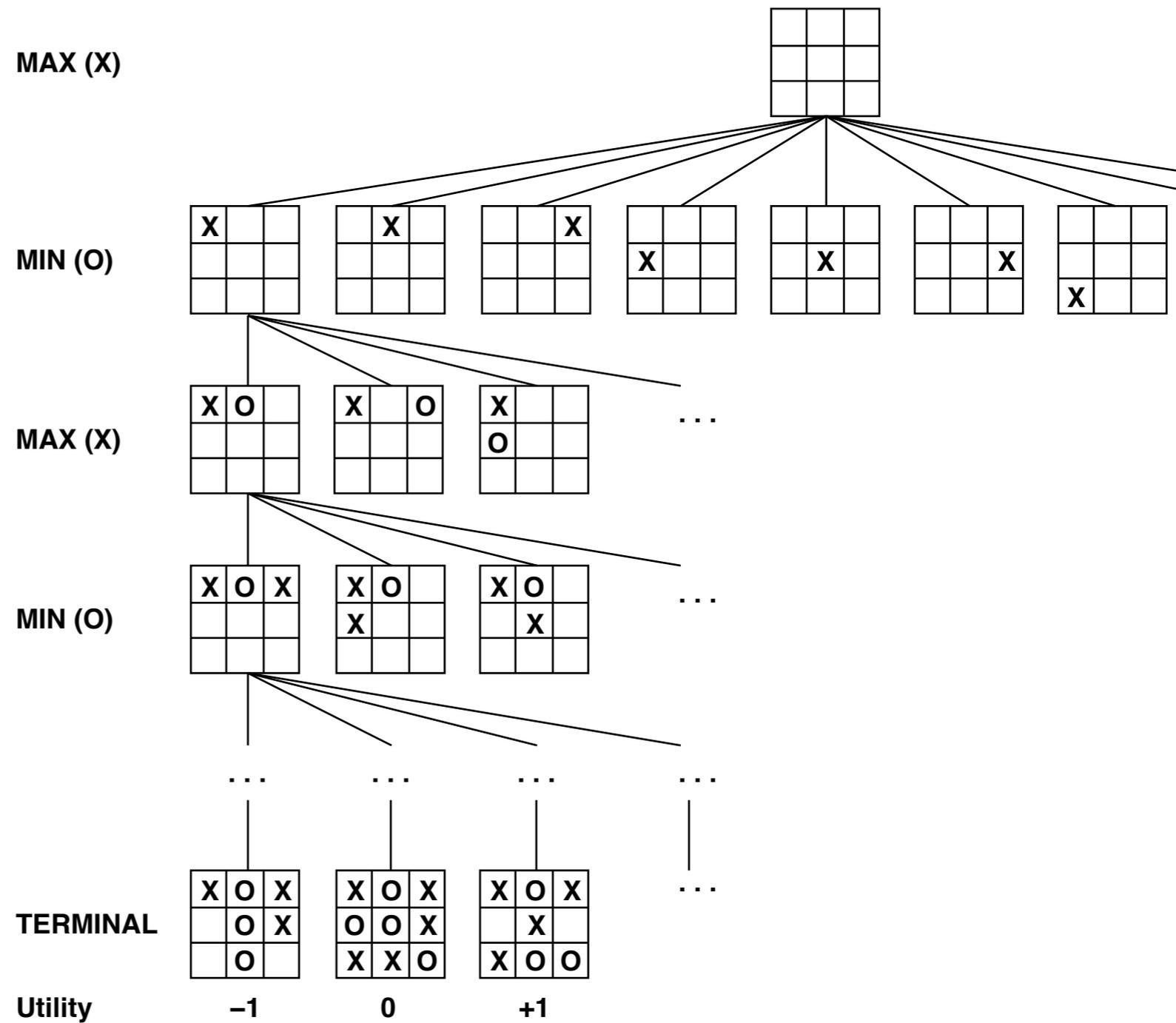


Contexto e Motivação

Jogos

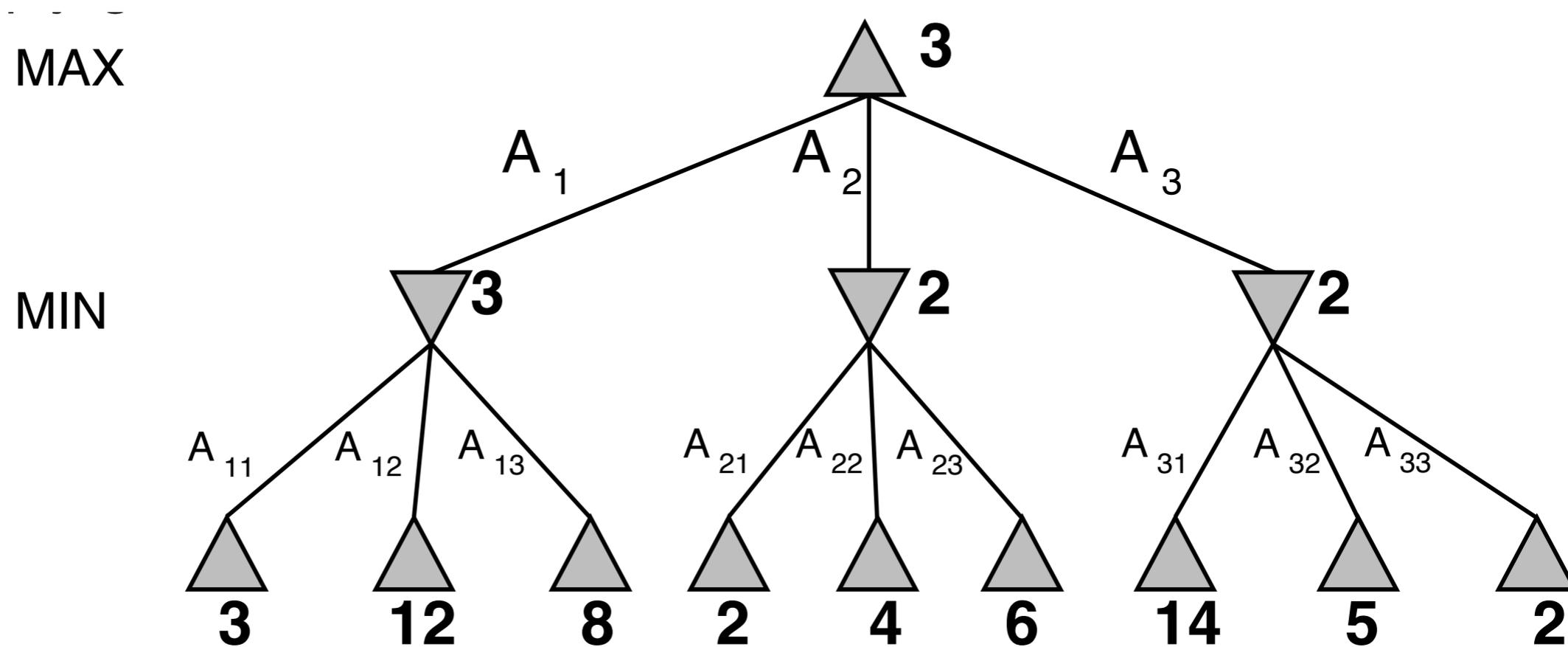
(árvore de busca)

- Figura do adversário na busca
- Próximo estado não é controlado pelo agente
- **Utilidade** dos estados



Jogos

(minimax)



MiniMax

- Completo: sim, se a árvore do jogo for finita
- Ótimo: sim, se o oponente for ótimo
- Complexidade exponencial

- Pode explorar a árvore só até um nível!
- Conhecimento do problema é colocado na função de utilidade
- Precisa explorar **todos** os caminhos?

Monte Carlo

estimar o **valor**
de um estado
observando partidas
aleatórias

Exploration x Exploitation

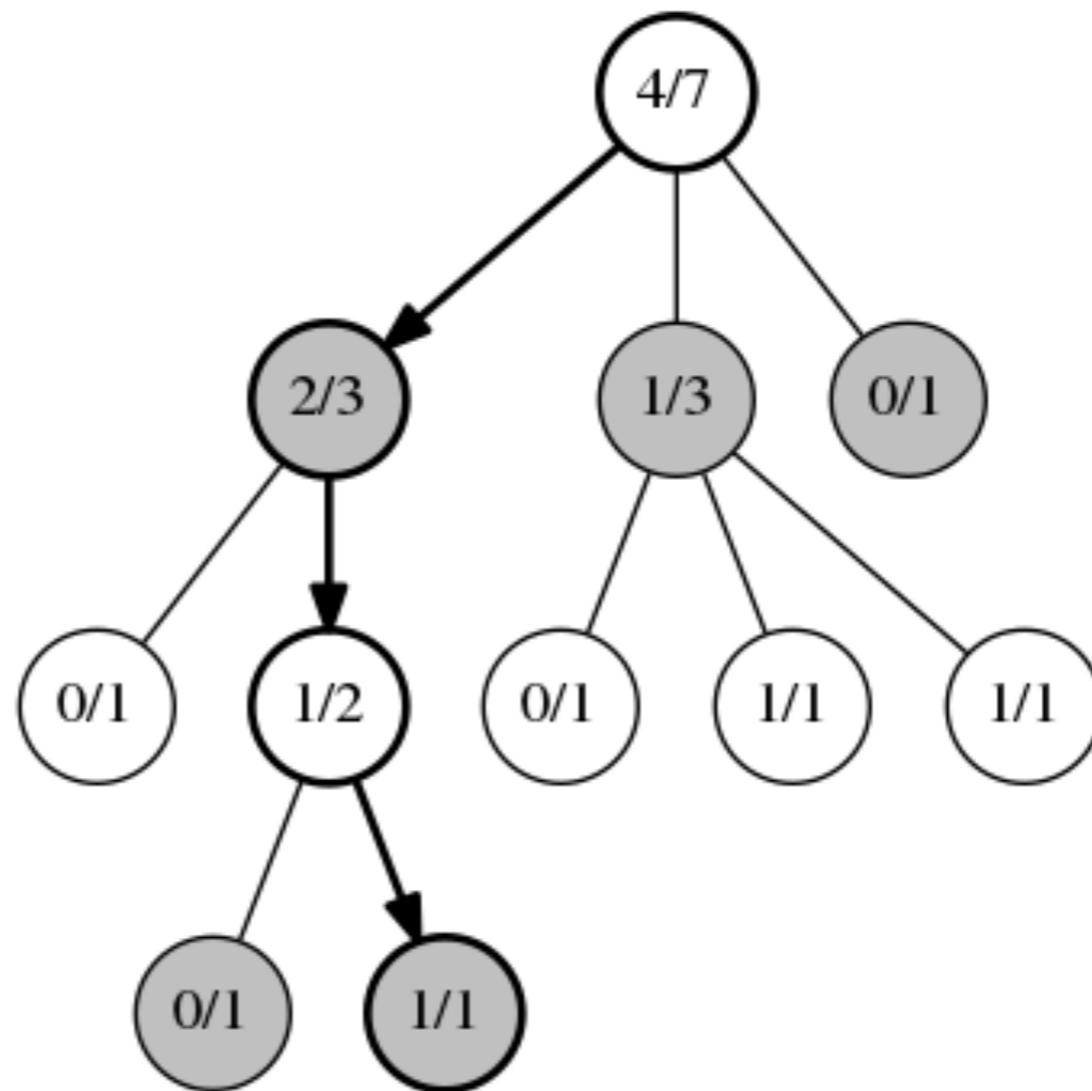
- Aprender o valor de ações que não conhecemos (Monte Carlo)
- Focar a busca em estados mais promissores (Tree Search)

Etapas

- Seleccionar
- Expandir
- Simular
- Atualizar

Etapas

- **Selecionar**
- Expandir
- Simular
- Atualizar

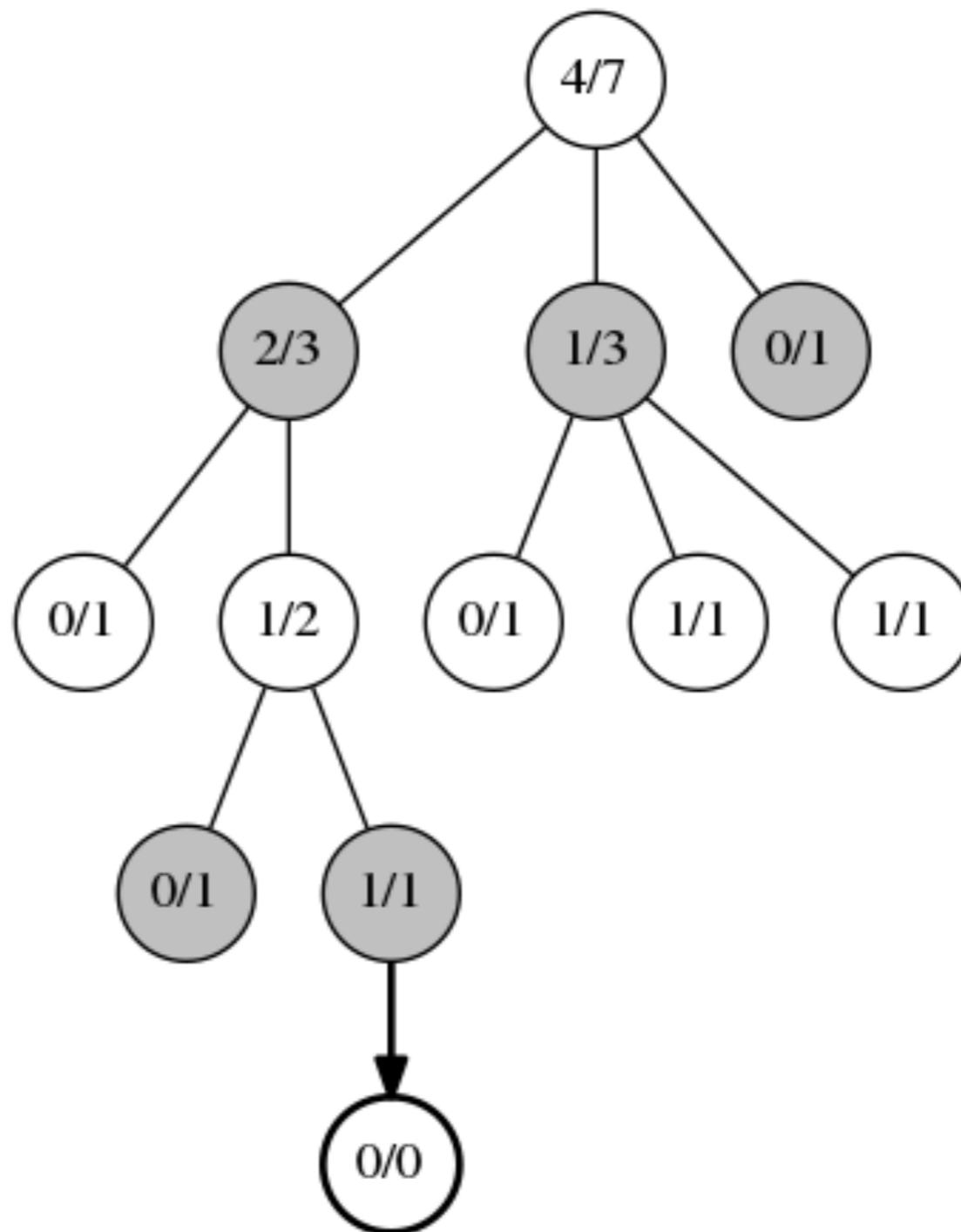


UCB =
$$\frac{w_i}{n_i} + c \sqrt{\frac{\ln t}{n_i}}$$
Upper Confidence Bound

w_i número vitórias pelo nodo i
 n_i número de jogos
 c fator de exploração $[0,1]$
 t número de jogos do modo pai

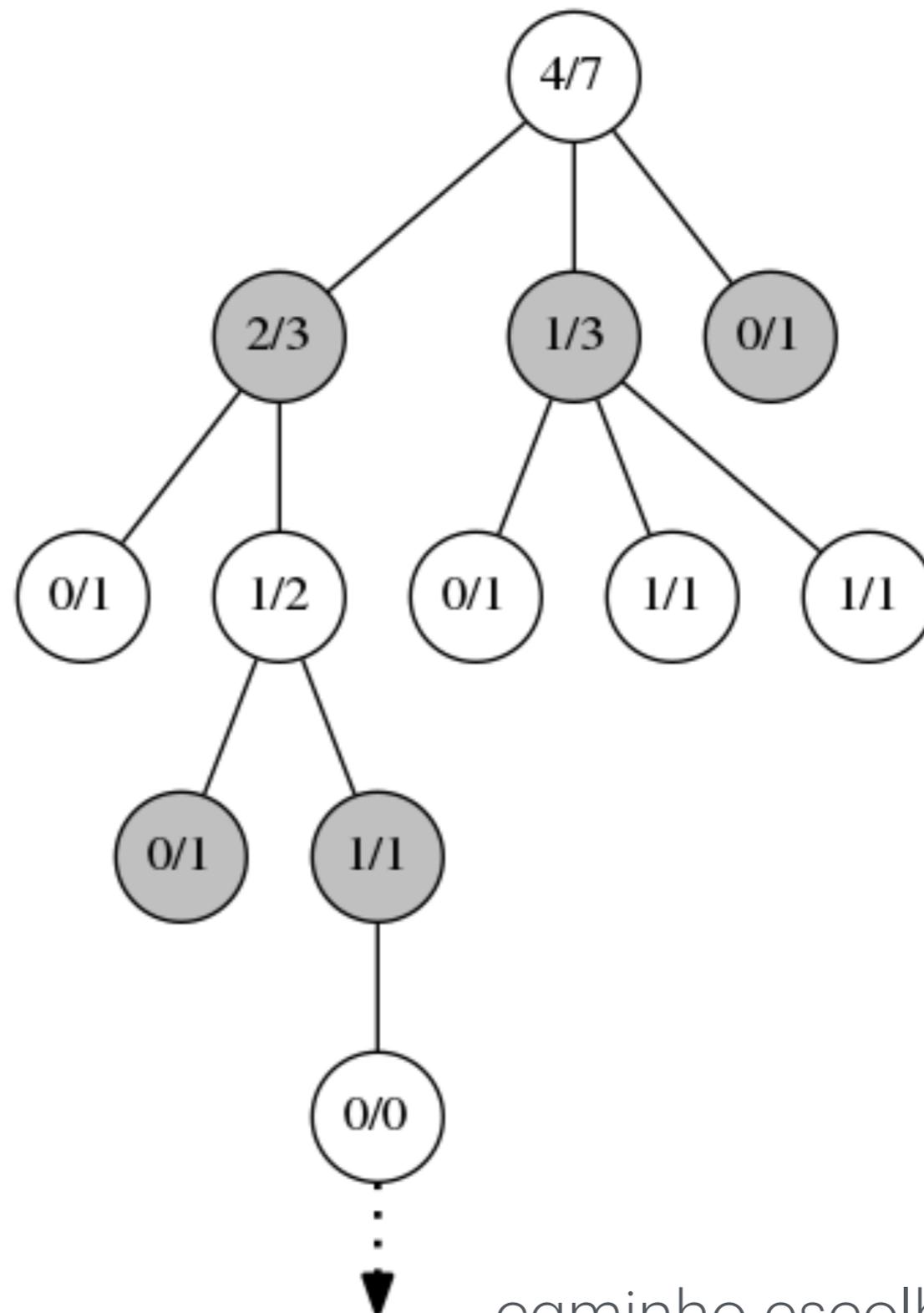
Etapas

- Seleccionar
- **Expandir**
- Simular
- Atualizar



Etapas

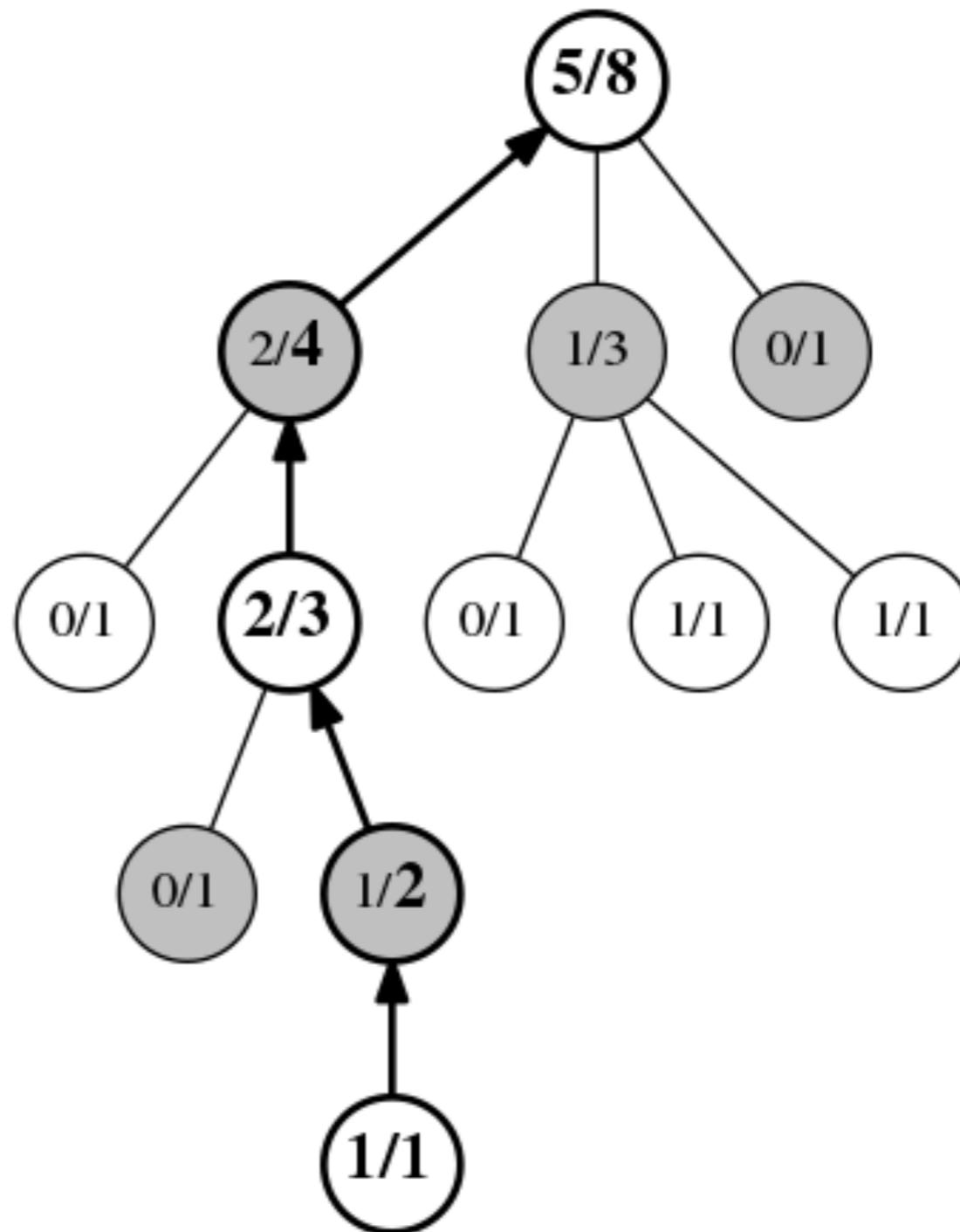
- Selecionar
- Expandir
- **Simular**
- Atualizar



caminho escolhido
aleatoriamente!

Etapas

- Seleccionar
- Expandir
- Simular
- **Atualizar**



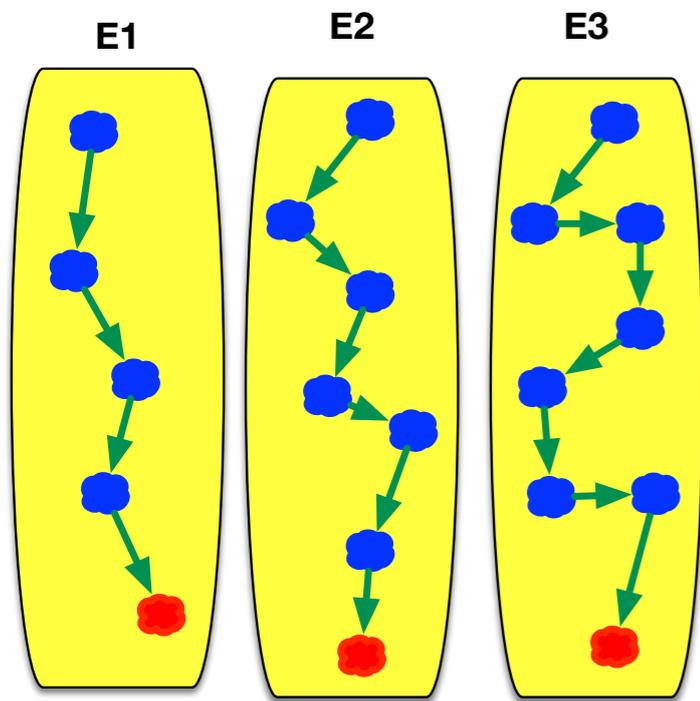
MCTS

- Simples, sucesso em Go (AlphaGo)
- Balanço exploração / exploração
- Dispensa conhecimento do problema
- Aproxima o valor dos estados/ação, $Q(s, a)$ como RL
- Pode demorar na fase de exploração

Material

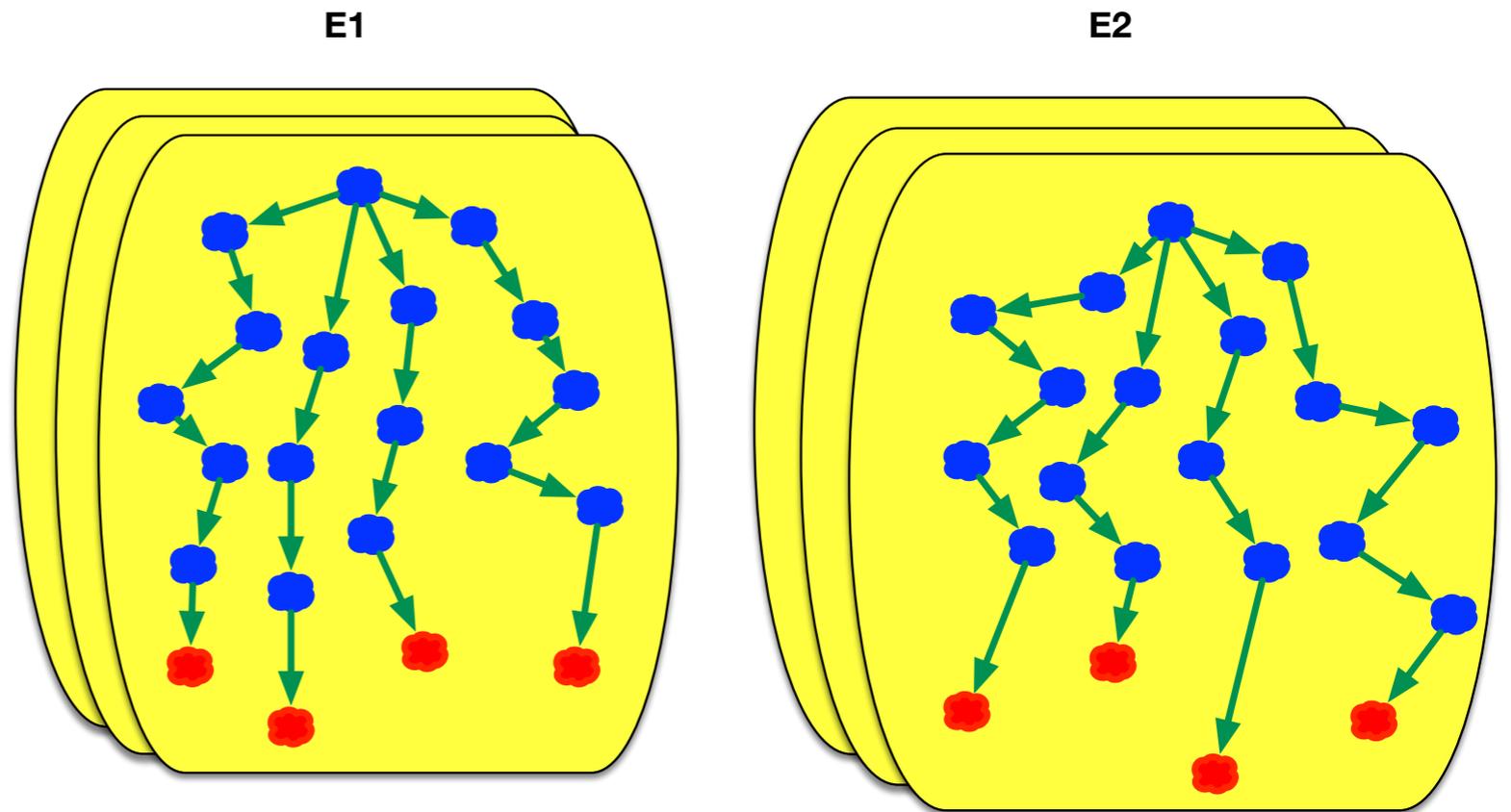
- A Survey of Monte Carlo Tree Search Methods, Cameron B. Browne et al, 2012
- Monte Carlo Tree Search: Implementing Reinforcement Learning in Real-Time Game Player, Masoud Masoumi Moghadam, 2020

Monte Carlo RL



One Episode = One Simulation

Monte Carlo Tree Search



One Episode = Many Simulations, a Series of Trees