# Game theory - lab 2

David Milec

Czech Technical University in Prague

*milecdav@fel.cvut.cz*

December 14, 2020

# Overview

# Getting Close to Optimal Strategy

Monte Carlo tree search

- Explores the possible action tree in a way that tries to balance exploration and exploitation
- When the node is visited for the first time, evaluate using heuristic/rollout
- Save the value received from rollout, update all nodes up to the tree and go again from the root
- Selects the nodes to visit based on the values from previous rollouts

# Monte Carlo in an Image

Graphical example of the steps performed in one Monte Carlo tree search update

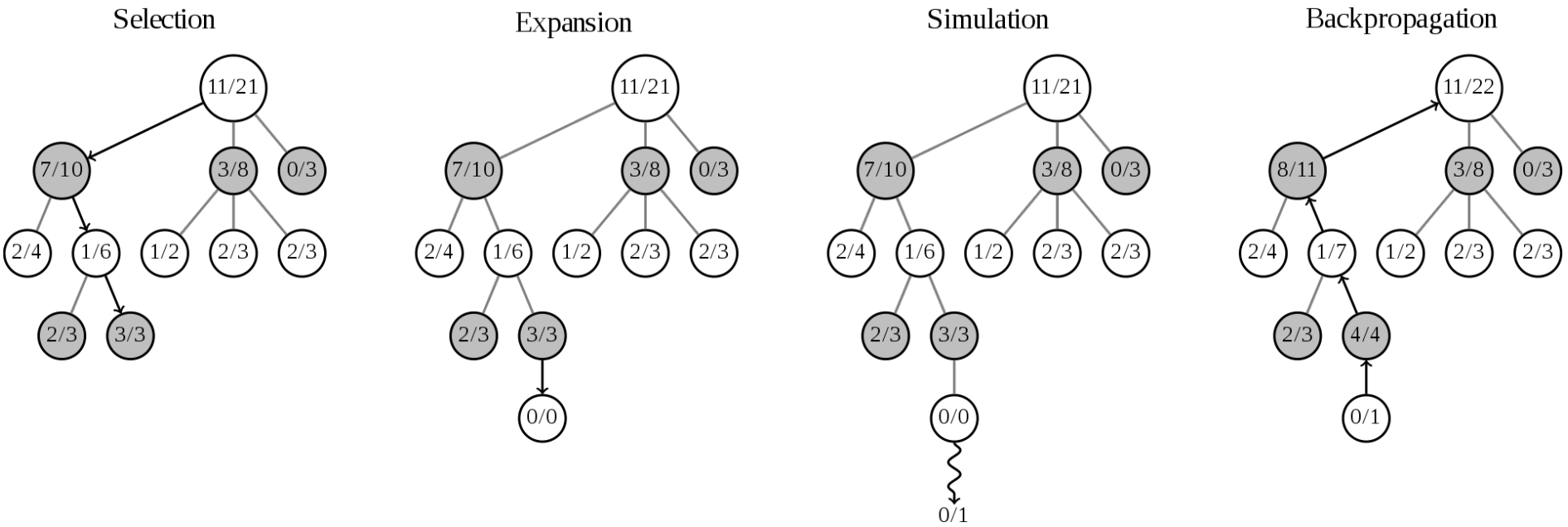
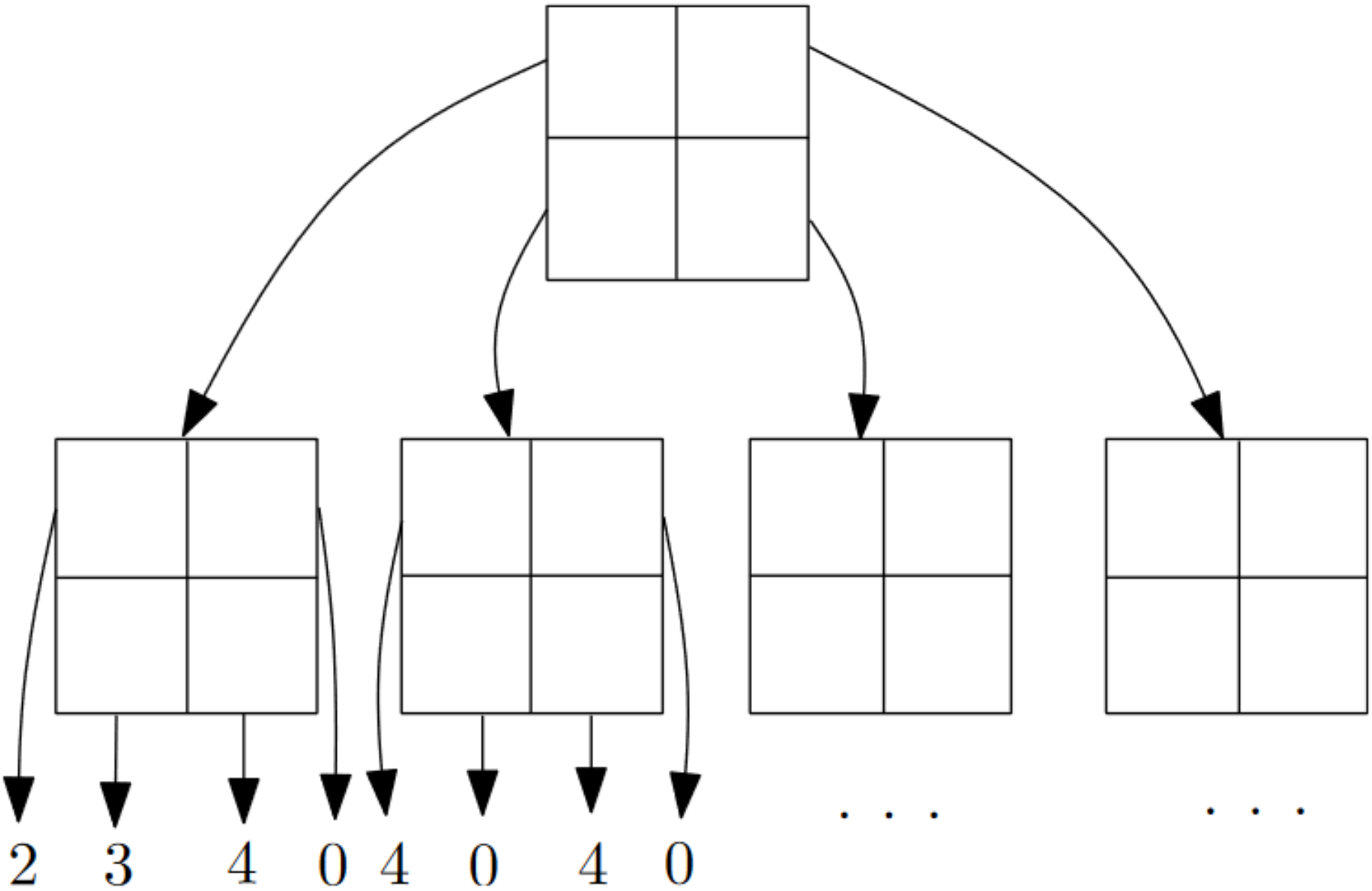



Image from
`https://en.wikipedia.org/wiki/Monte_Carlo_tree_search`

# Simultaneous move game example



# Simultaneous moves in MCTS

Decoupled

- In each state use UCT computations and action picking independently for each player
- Combine the best actions for both players in the selection phase
- Should be easier to implement and simulate

Sequential

- Split the decision nodes to separate both players creating the game where one player knows where the other moved
- If the player that knows the move is the opponent the strategy will be more defensive as it will compute with the worst case action

# Asymptotically Optimal Strategy

Value iteration

- Adaptation of algorithm used to solve MDPs
- Stores values for all possible states of the game
- Iteratively updates those values based on possible actions in each state, solving matrix game created from next state values
- In the end uses the computed values to computed best strategy

# Value Iteration

$S$ is the state space, $v : S \to \mathbb{R}$ is value in each state, $\mathcal{A}$ is set of all combinations of actions and $A : S \to \mathcal{A}$ is a function returning all possible action tuples available in a given state. $Q$ is a matrix game created for each state in each iteration, $r : S \times \mathcal{A} \to \mathbb{R}$ is immediate payoff and $T : S \times \mathcal{A} \to S$ is a transition function. $\gamma$ is discounting constant.

$$\forall s \in S \quad \text{initialize} \quad v(s) = 0 \quad \text{and until v converges}$$
$$\forall s \in S$$
$$\forall (a_1, a_2) \in A(s)$$
$$Q(a_1, a_2) = r(s, a_1, a_2) + \gamma v(T(s, a_1, a_2))$$
$$v(s) = \max_x \min_y xQy$$

# Computing Nash Equilibirum

$$
\begin{aligned}
\text{maximize} \quad & U \\
\text{subject to} \quad & \sum_{a_1 \in A_1} x(a_1) u_1(a_1, a_2) \geq U && \forall a_2 \in A_2 \\
& \sum_{a_1 \in A_1} x(a_1) = 1 \\
& x(a_1) \geq 0 && \forall a_1 \in A_1
\end{aligned}
$$

# The End