Monte Carlo Tree Search

Olivier Sigaud

Sorbonne Université
http://www.isir.upmc.fr/personnel/sigaud

Motivation

- A domain independent algorithm to plan in one's head to determine the best next action
- Example: two-player games (Chess, Go...).
- Minimax defeated Kasparov in 1998, was considering the whole tree, too expensive at Go
- MCTS was the leading technique at Go before AlphaZero
- Requires an internal simulator
- Requires a capability to reset anywhere
- Very efficient tree search method

Gelly, S., Wang, Y., Munos, R., and Teytaud, O. Modification of UCT with patterns in Monte-Carlo go. Technical Report 32, RR-6062, INRIA, 2006.

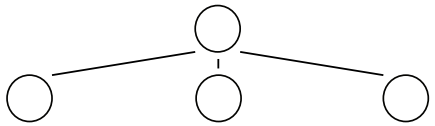
Overview

- Somewhere between breadth-first and depth-first search
- Similar to A^* without the admissible heuristic
- The cost is in the numerous simulations $\rightarrow$ AlphaZero improves this
- Four processes: Selection, Expansion, Simulation, Update

Initial step

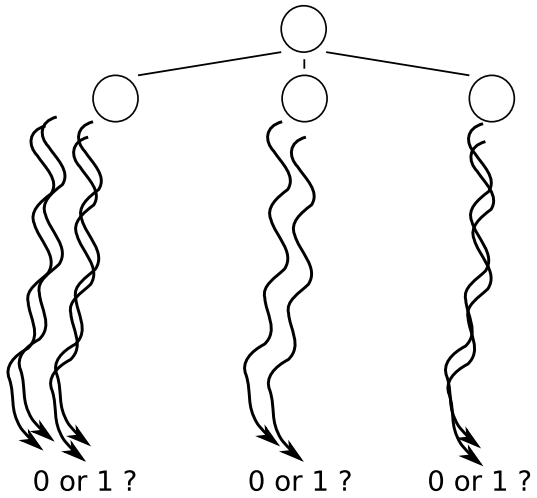
- A node represents a discrete state, an edge represents a discrete action
- The process starts with an empty node
- This node corresponds to the current state where the next action has to be chosen

Initial step: Expansion



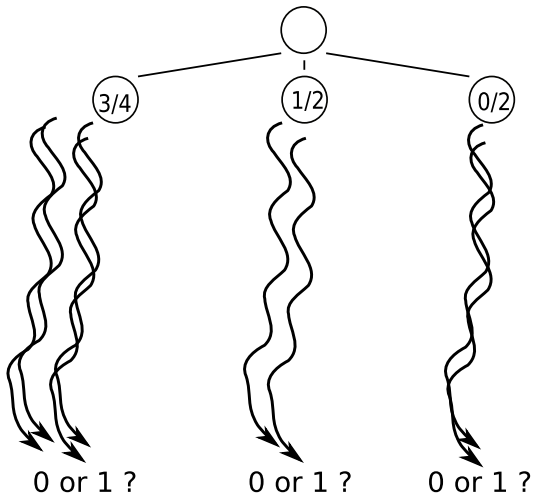
- The MCTS agent tries actions (in its head), resulting in adding child nodes

Initial step: Simulation



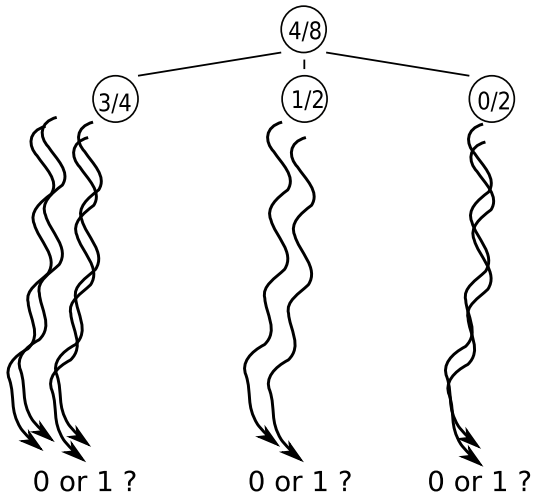
- From each selected node, it performs random simulations (Monte Carlo) to evaluate the node (without adding nodes yet)
- Initial child node selection is random

Initial step: Update



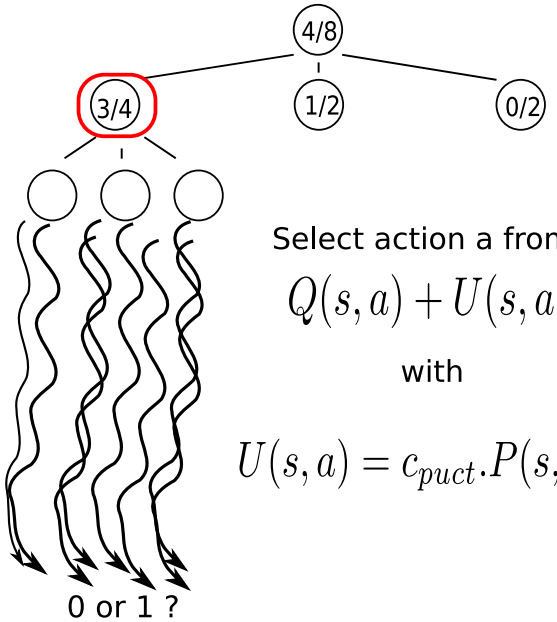
- It updates the values of children based on the statistics of the simulations
- The value is a state value $V(s)$

Initial step: Update parents



- It also updates the values of parents
- Note that state values $V(s)$ could be changed into state-action values $Q(s,a)$ using $Q(s,a) = r(s,a) + \gamma V(s')$
- In Go, $Q(s,a) = V(s')$ (no intermediate reward)

Selection



Select action a from

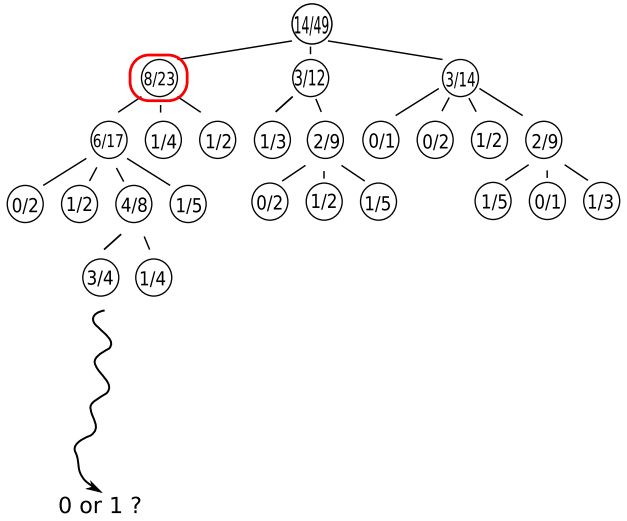
$$Q(s,a) + U(s,a)$$

with

$$U(s,a) = c_{puct}.P(s,a).\sqrt{\frac{\sum_b N(s,b)}{1+N(s,a)}}$$

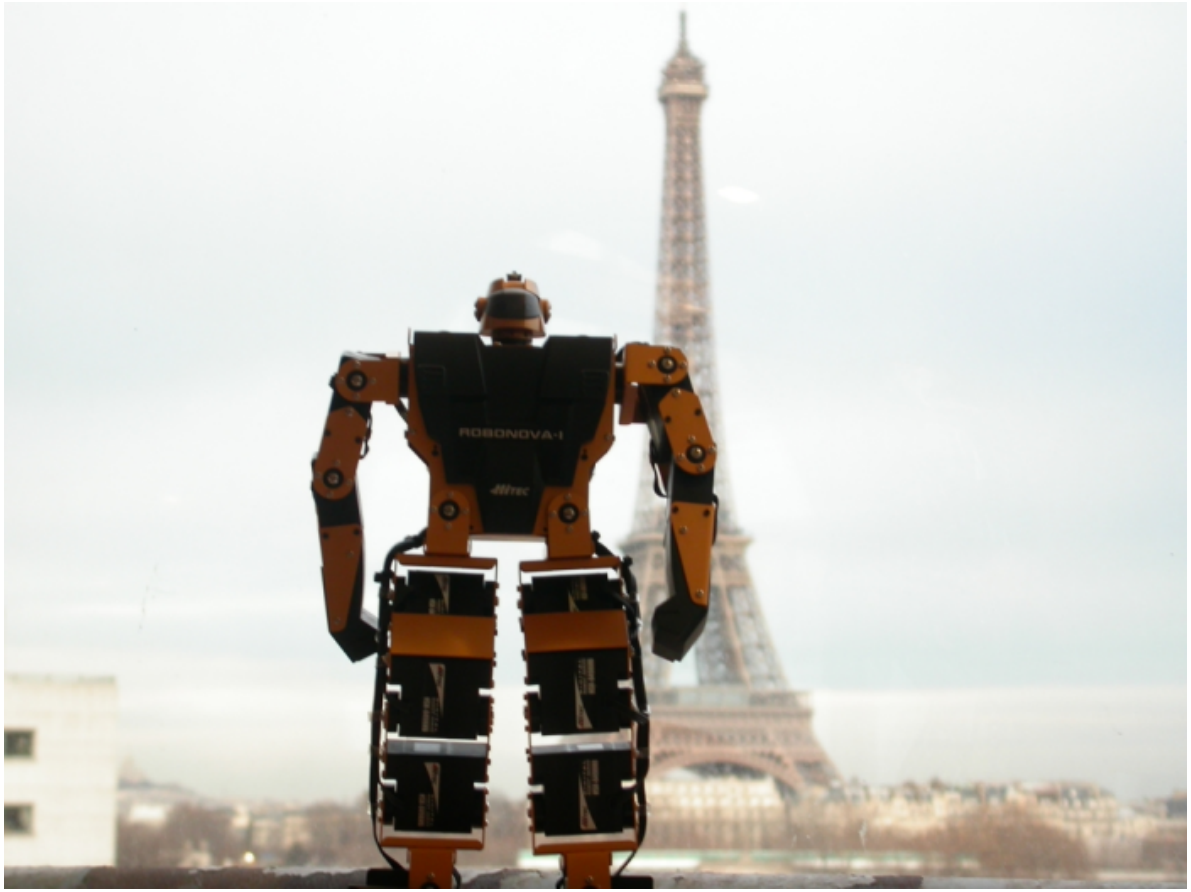
- Selection operates over all expanded nodes
- That's where the reset-anywhere property is necessary
- It favors leaf nodes with a higher chance of success
- But it avoids ignoring too much lower success nodes
- A lower $N(s,a)$ results in a higher $U(s,a)$
- The selected node is expanded, and the process is repeated

Action Selection



- After some budget, the search process stops
- The agent performs the action leading to the most visited first level child
- In exploration mode, some noise is added
- The current agent state is updated, and the process starts again
- MPC-like process
- A lot of computations are forgotten...

Any question?



Send mail to: `Olivier.Sigaud@upmc.fr`

Gelly, S., Wang, Y., Munos, R., and Teytaud, O.

Modification of UCT with patterns in monte-carlo go.
Technical Report 32, RR-6062, INRIA, 2006.