



**TECHNISCHE
UNIVERSITÄT
DRESDEN**

Fakultätsname Informatik Fachrichtung Informatik Institutsname Intelligente Systeme

The Gizmo Player

**Simon Dollé
Jan Kopcsek
Alper Tunga**

Dresden, 13.02.2008

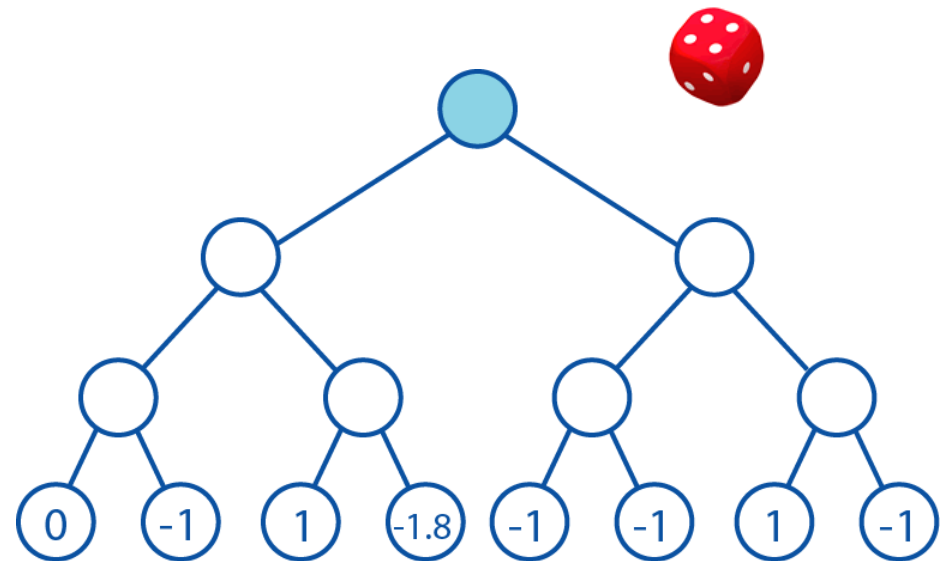
Finding a heuristic function

Two ways for learning a heuristic function:

- Deductive
 - Analyzing the rules
 - Identify common elements like game boards or pieces
 - Finding patterns
- Inductive
 - Playing and learning from experience
 - Monte Carlo strategy

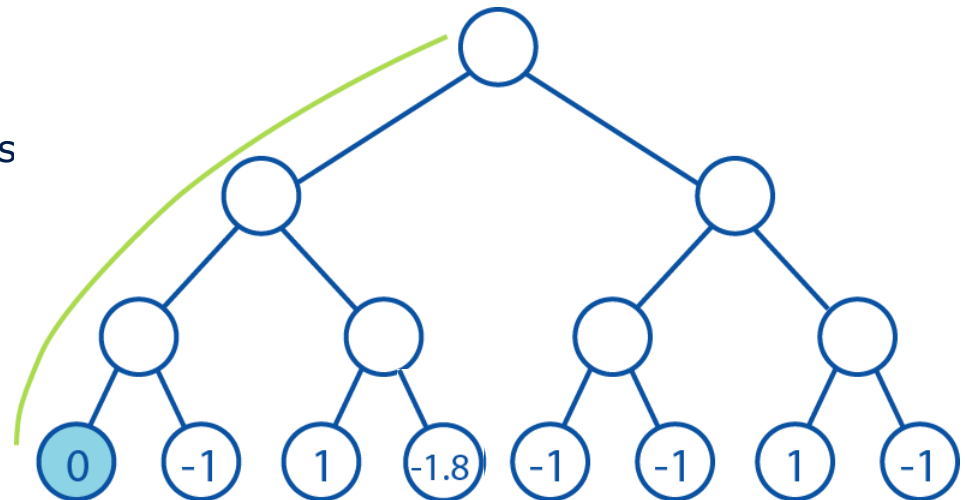
Monte Carlo Strategy

- Play random games
- Compute the means of scores for each move
 - Use them as a heuristic function



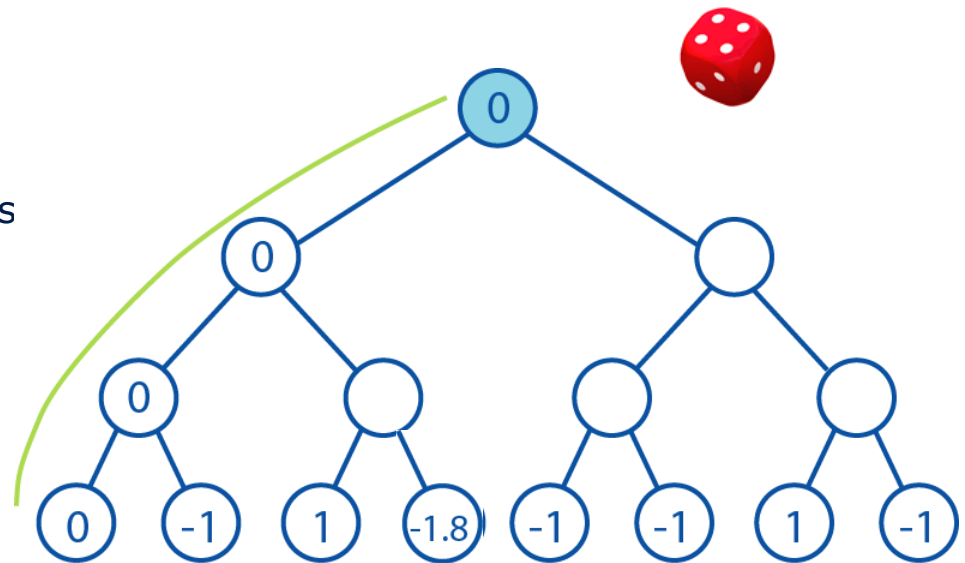
Monte Carlo Strategy

- Play random games
- Compute the means of scores for each move
 - Use them as a heuristic function



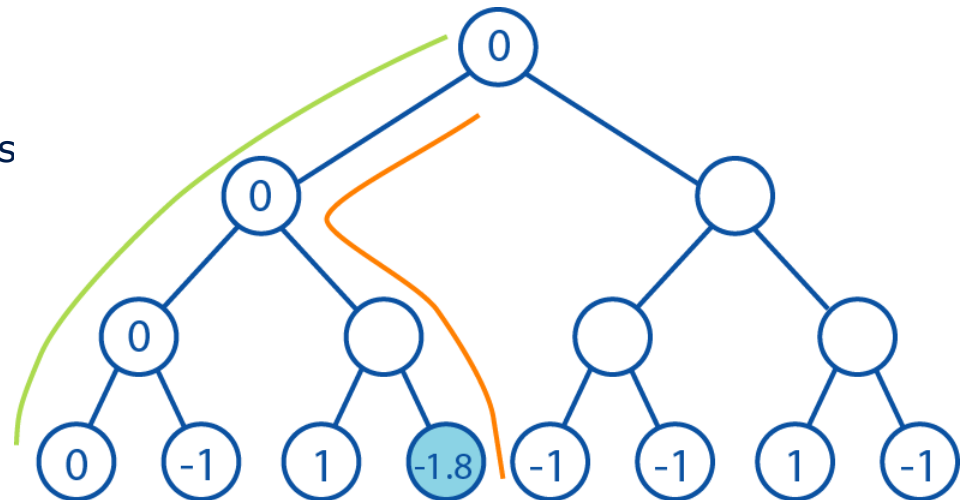
Monte Carlo Strategy

- Play random games
- Compute the means of scores for each move
 - Use them as a heuristic function



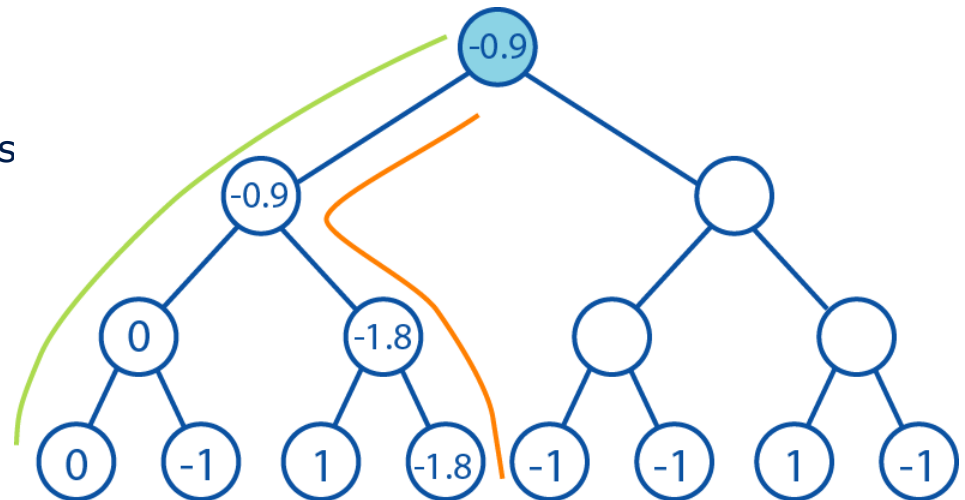
Monte Carlo Strategy

- Play random games
- Compute the means of scores for each move
 - Use them as a heuristic function



Monte Carlo Strategy

- Play random games
- Compute the means of scores for each move
 - Use them as a heuristic function

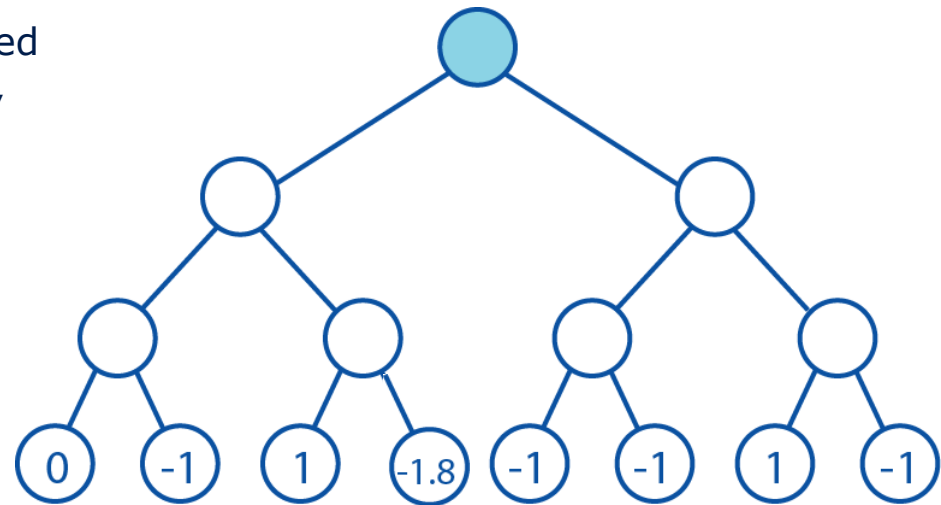


Monte Carlo Strategy

- Problem:
 - Same effort spend on interesting moves and uninteresting moves
 - Equivalent to play against a dummy player
- UCT Algorithm (Upper Confidence Bound for Trees):
 - An algorithm to balance:
 - Exploration of interesting parts of the graph
 - Exploration of new parts
 - Make random games more realistic

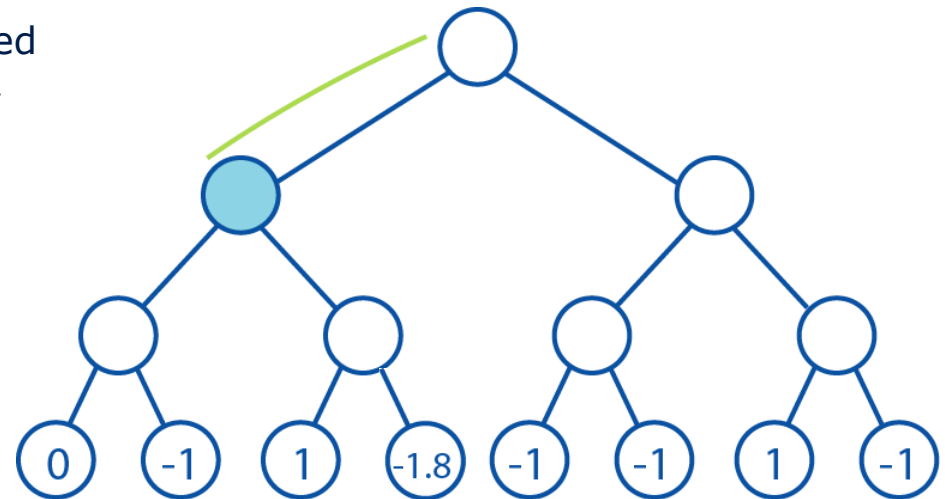
UCT Algorithm

- As long as there are unexplored moves from our current state, explore them



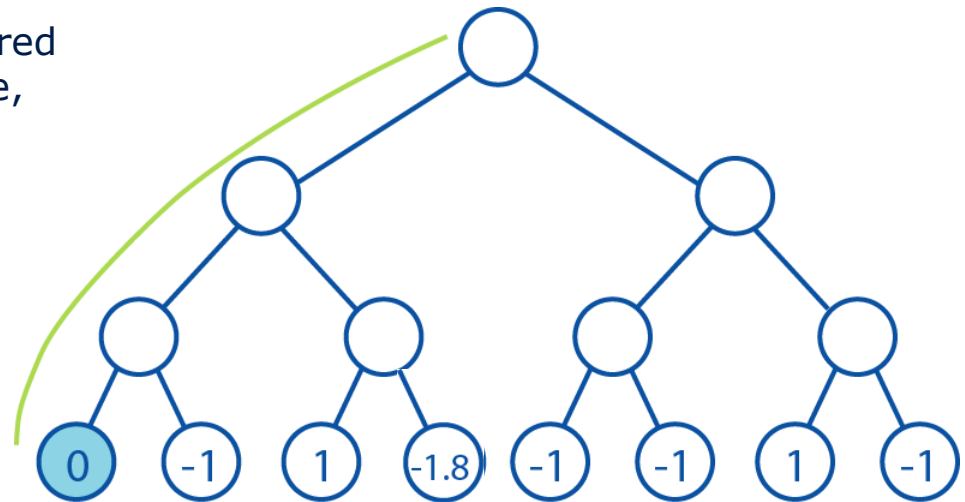
UCT Algorithm

- As long as there are unexplored moves from our current state, explore them



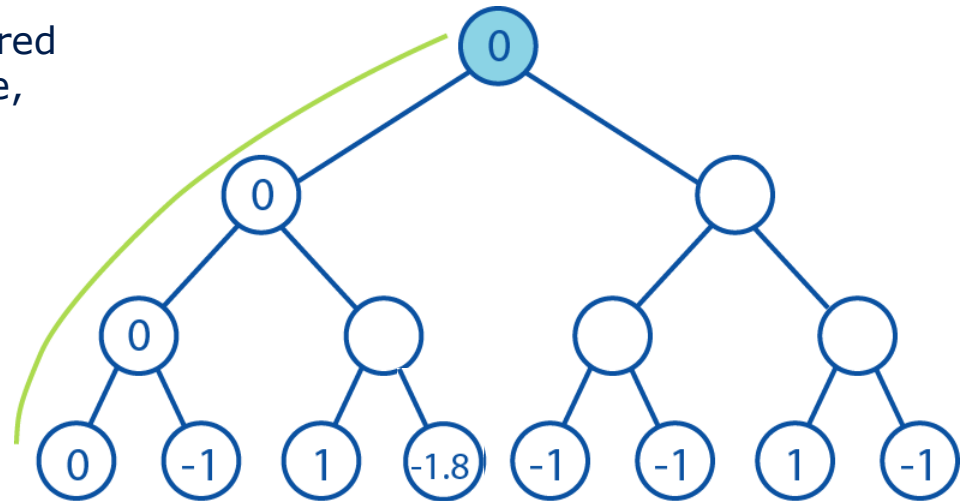
UCT Algorithm

- As long as there are unexplored moves from our current state, explore them



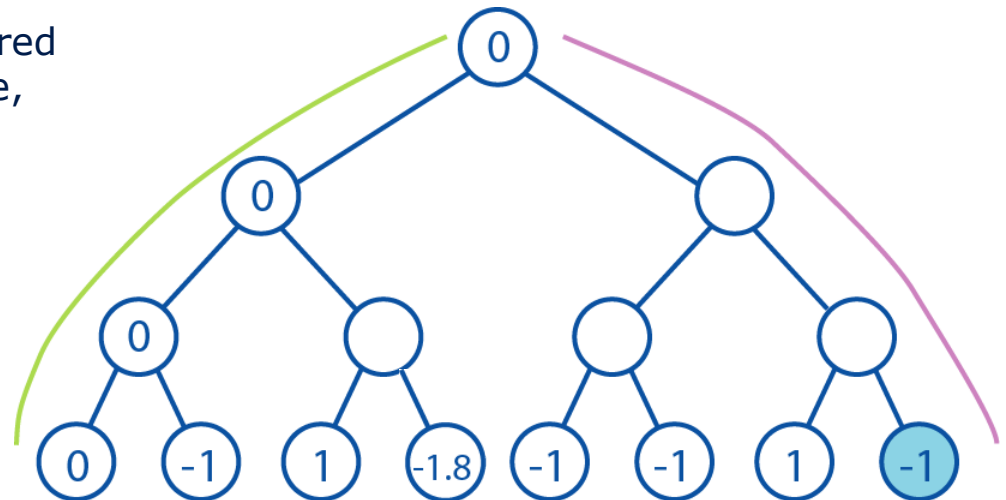
UCT Algorithm

- As long as there are unexplored moves from our current state, explore them



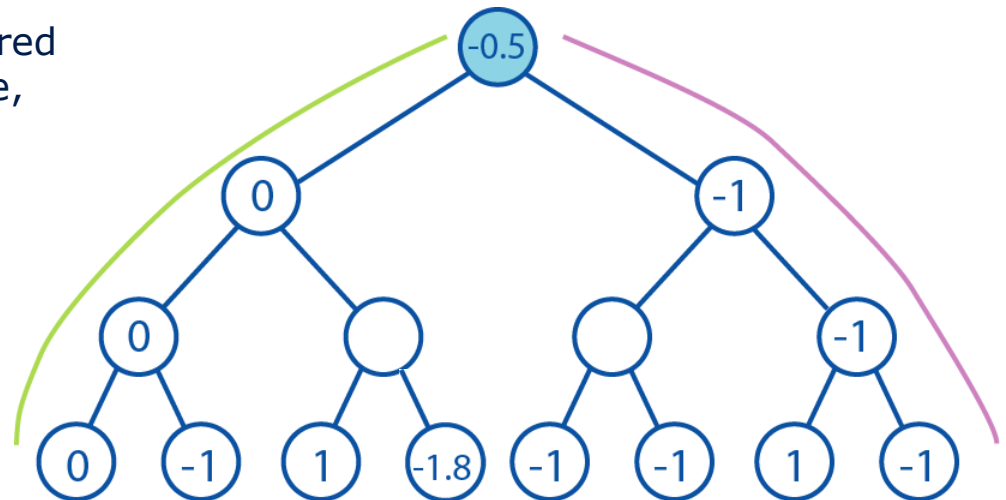
UCT Algorithm

- As long as there are unexplored moves from our current state, explore them



UCT Algorithm

- As long as there are unexplored moves from our current state, explore them

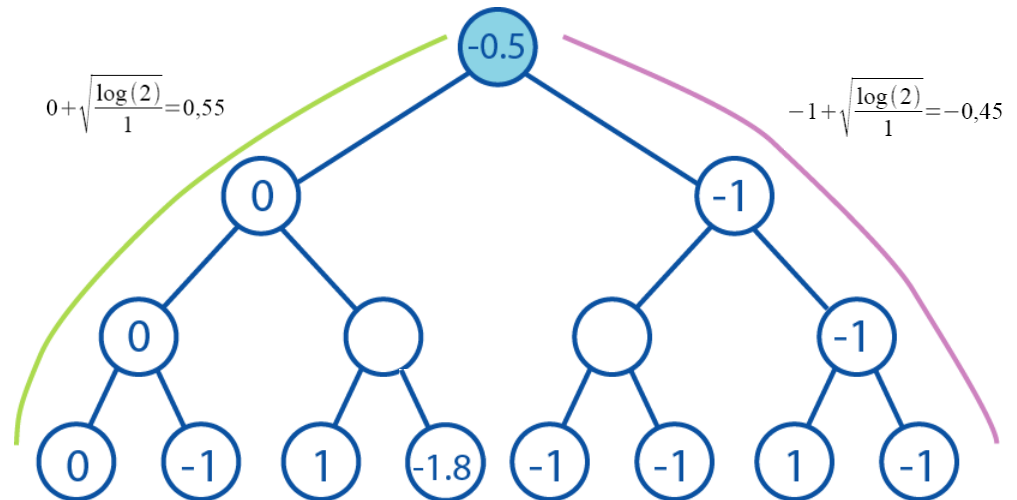


UCT Algorithm

- As long as there are unexplored moves from our current state, explore them
- Otherwise, choose the one with the highest score using

$$h + \sqrt{\frac{\log(n)}{n_i}}$$

h : the heuristic value
 n : the number of games through the parent node
 n_i : the number of games through the node



UCT Algorithm

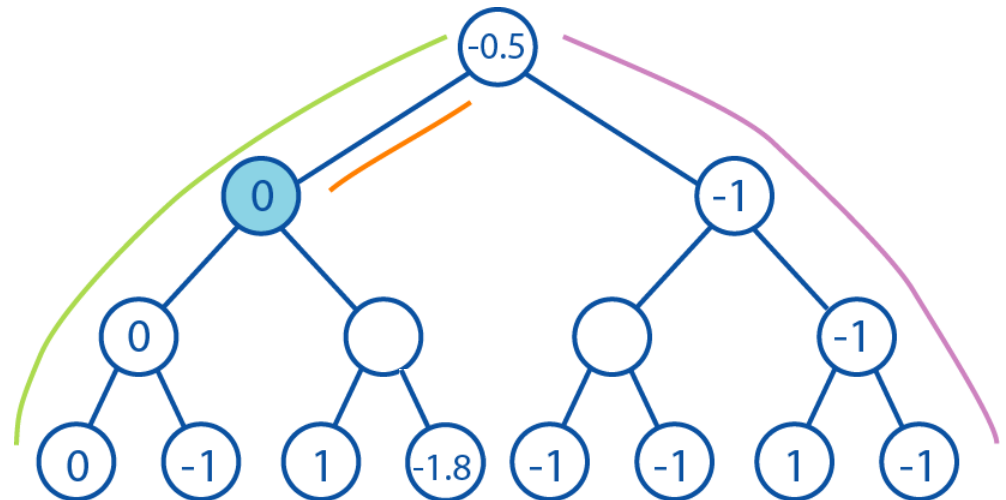
- As long as there are unexplored moves from our current state, explore them
- Otherwise, choose the one with the highest score using

$$h + \sqrt{\frac{\log(n)}{n_i}}$$

h : the heuristic value

n : the number of games through the parent node

n_i : the number of games through the node

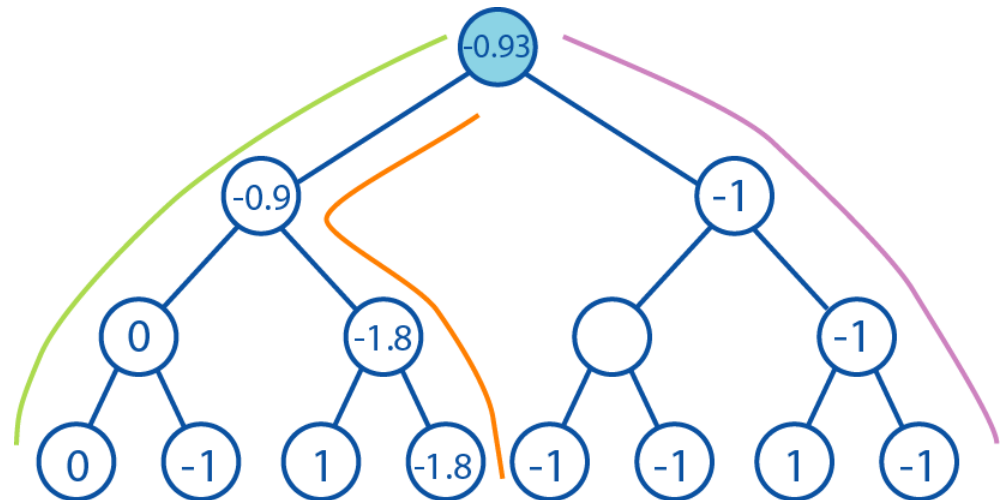


UCT Algorithm

- As long as there are unexplored moves from our current state, explore them
- Otherwise, choose the one with the highest score using

$$h + \sqrt{\frac{\log(n)}{n_i}}$$

h : the heuristic value
 n : the number of games through the parent node
 n_i : the number of games through the node

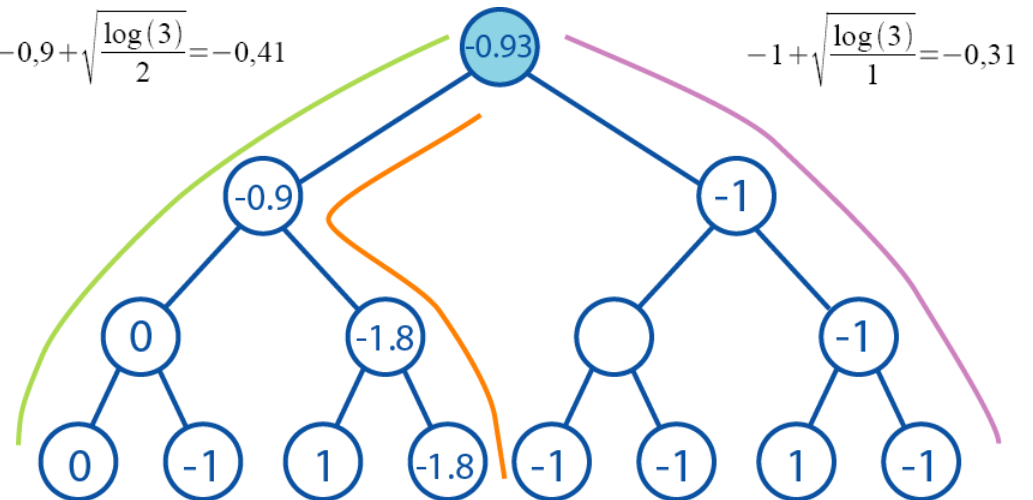


UCT Algorithm

- As long as there are unexplored moves from our current state, explore them
- Otherwise, choose the one with the highest score using

$$h + \sqrt{\frac{\log(n)}{n_i}}$$

h : the heuristic value
 n : the number of games through the parent node
 n_i : the number of games through the node

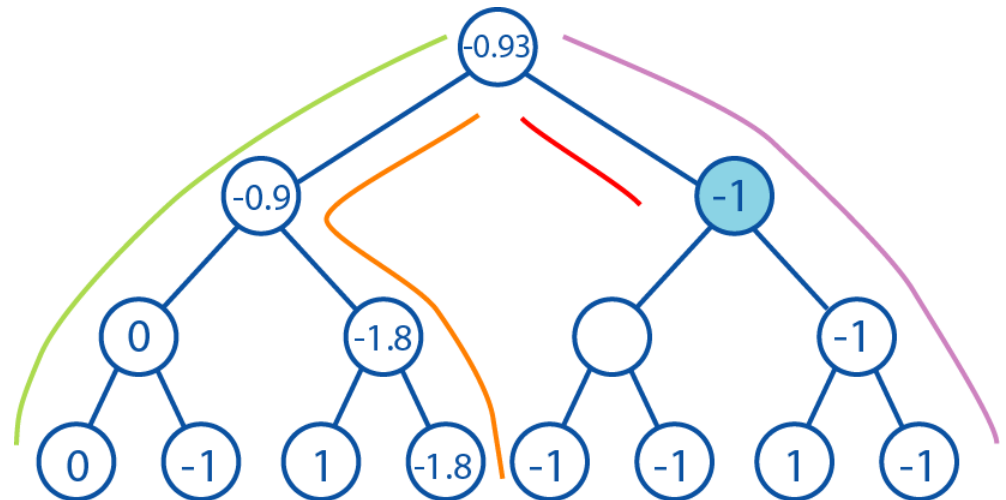


UCT Algorithm

- As long as there are unexplored moves from our current state, explore them
- Otherwise, choose the one with the highest score using

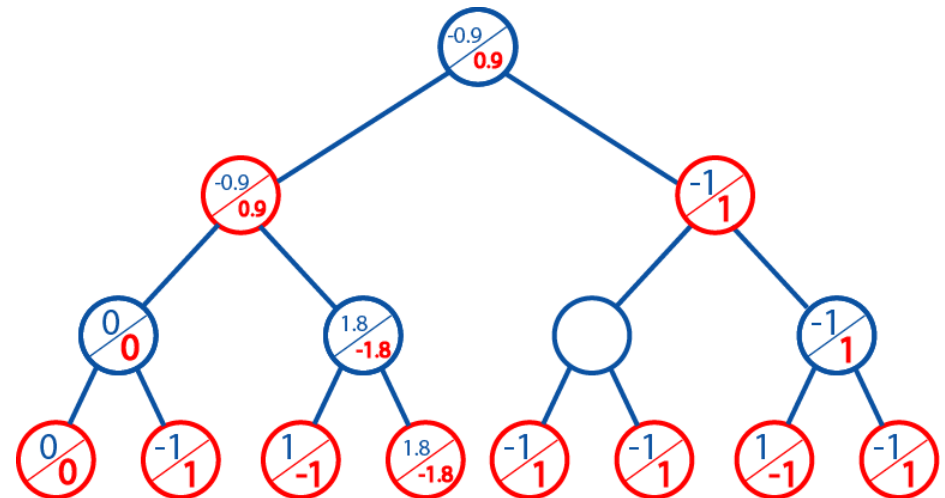
$$h + \sqrt{\frac{\log(n)}{n_i}}$$

h : the heuristic value
n : the number of games through the parent node
n_i : the number of games through the node



UCT Algorithm

- Which move to play?
 - The one with the highest heuristic value
- In multiplayer games:
 - Store the heuristic value for each player



Good points

- Heuristic directly linked to the final score
- Heuristic converges to min-max values
- Time scalable
- Easily parallelisable

Problems

- Simultaneous moves:
 - What rule to choose to explore the nodes?
 - Which move to play?
- Long games and loops:
 - Depth first search problem

Thank you for your attention

And good luck to your players