University of Basel

Online Knowledge Enhancements for Monte Carlo Tree Search in Probabilistic Planning

Bachelor presentation

Marcel Neidinger <m.neidinger@unibas.ch>

Department of Mathematics and Computer Science,

University of Basel

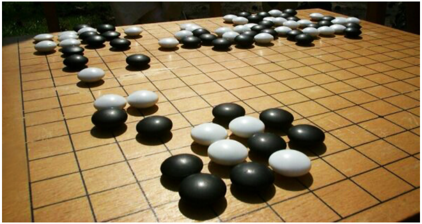
13. February 2017

What is Probabilistic Planning?

- Solve planning tasks with probabilistic transitions
- Models a **Markov Decision Problem** given by $M = \langle V, s_0, A, T, R \rangle$
  - **A set of binary variables** V **inducing States** $S = 2^V$
  - An initial state $s_0 \in S$
  - A set of applicable actions A
  - A transition model $T : S \times A \times S \rightarrow [0; 1]$
  - **A Reward** $R(s, a)$
- **Monte Carlo Tree Search** algorithms solve MDPs

Monte Carlo Tree Search Algorithms

- Algorithmic framework to solve MDPs
- Used especially in computer Go

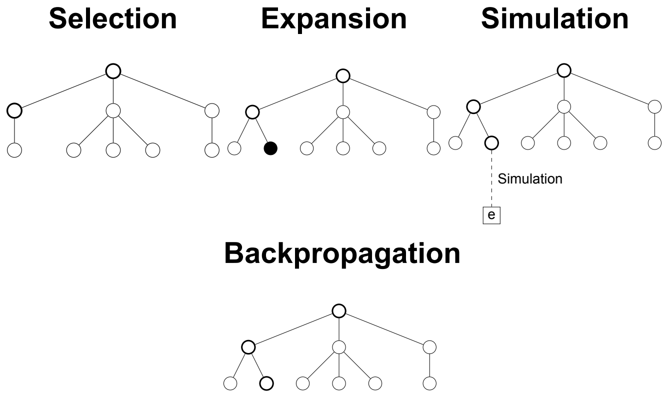


Go Board[1]

Lee Sedol[2]

[1] Source: https://commons.wikimedia.org/wiki/File:Go_board.jpg
[2] Source: https://qz.com/639952/googles-ai-won-the-game-go-by-defying-millennia-of-basic-human-instinct/

Four phases - Two components



Monte Carlo Tree node

- MCTS tree for a MDP M
- Important information in a tree node
  - A state $s \in S$
  - A counter $N^{(i)}$ for the number of visits
  - A counter $N^{(i)}(s, a)\, \forall a \in A$ for the number of times a was selected in s
  - A reward estimate $Q^{(i)}(s, a)$ for action a in state s

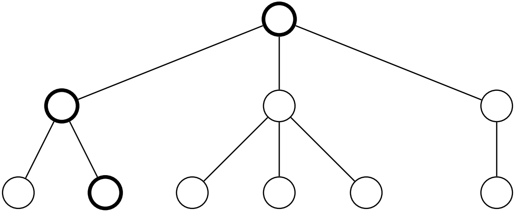
Online Knowledge

- AlphaGo used Neural Networks for the two policis $\rightarrow$ Domain-specific knowledge
- We want **domain independent** enhancements

Overview

- Tree-Policy Enhancements
  - All Moves as First
    - α-AMAF
    - Cutoff-AMAF
  - Rapid Action Value Estimation
- Default-Policy Enhancements
  - Move-Average Sampling Technique
- Conclusion

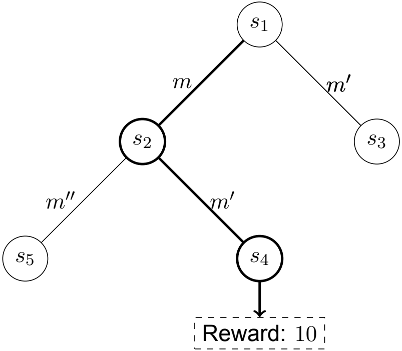
What is a Tree Policy?



- Iterate through the **known part** of the tree and select an action given a node
- Use a Q value for a state-action pair to estimate an actions reward

UCT

- MCTS implementation first proposed in 2006



UCT

- Reward approximation, parent node v_l, child node v_j

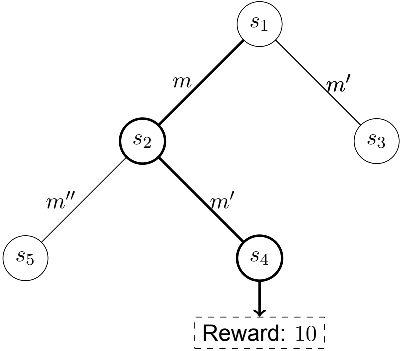
$$UCT(v_l, v_j) = Q^{(i)}(s_l, a_j) + 2\, C_p \sqrt{\frac{2 \ln N^{(i)}(s_l)}{N^{(i+1)}(s_j)}} \tag{1}$$

- From parent v_l select child node v^* that maximises

$$v^* = \max_{v_j}\{UCT(n_l, n_j)\} \tag{2}$$

All Moves as First - Idea

- UCT score needs several trials to become reliable
- **Idea:** Generalize informations extracted from trials
- **Implementation:** Use additional (node-independant) score that updates unselected actions as well



State	Action	Reward
s_1	m	…

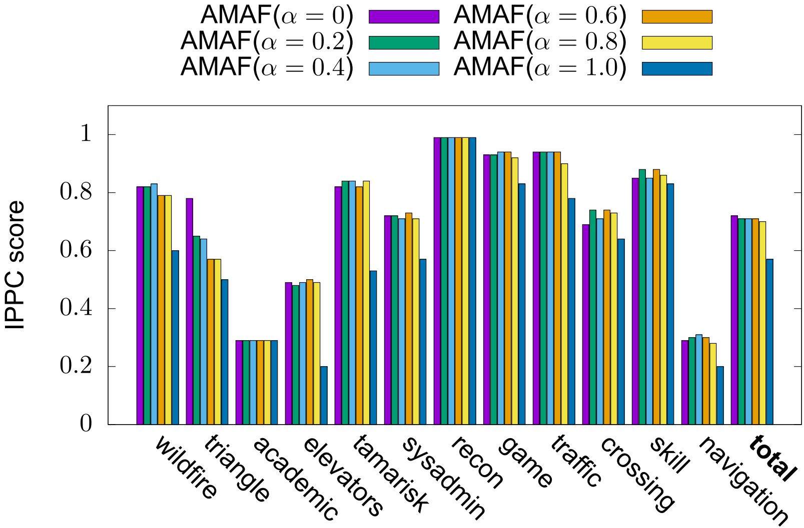
All Moves as First - α-AMAF

- **Idea:** Combine UCT and AMAF score

$$SCR = \alpha AMAF + (1 - \alpha) UCT \tag{3}$$

- Choose action with highest SCR

All Moves as First - α-AMAF - Results



All Moves as First - α-AMAF - Problems

- With more trials UCT becomes more reliable
- AMAF score has higher variance

We want to discontinue using AMAF score after some time

All Moves as First - α-AMAF - Problems

- With more trials UCT becomes more reliable
- AMAF score has higher variance

We want to discontinue using AMAF score after some time

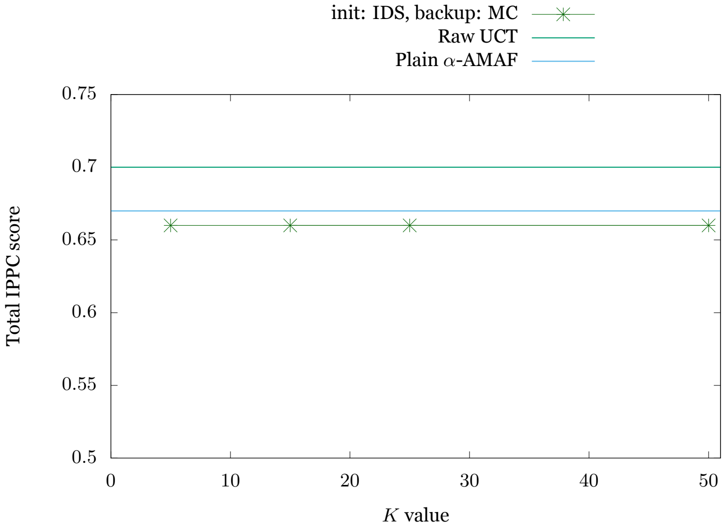
All Moves as First - Cutoff-AMAF

- Introduce cutoff parameter K

$$SCR = \begin{cases} \alpha AMAF + (1-\alpha)UCT & \text{, for } i \leq k \\ UCT & \text{, else} \end{cases} \tag{4}$$

- Use AMAF score only in the first k trials

All Moves as First - Cutoff-AMAF - Results



All Moves as First - Cutoff-AMAF - Problems

- How to choose the parameter K?
- When is the UCT score reliable enough?

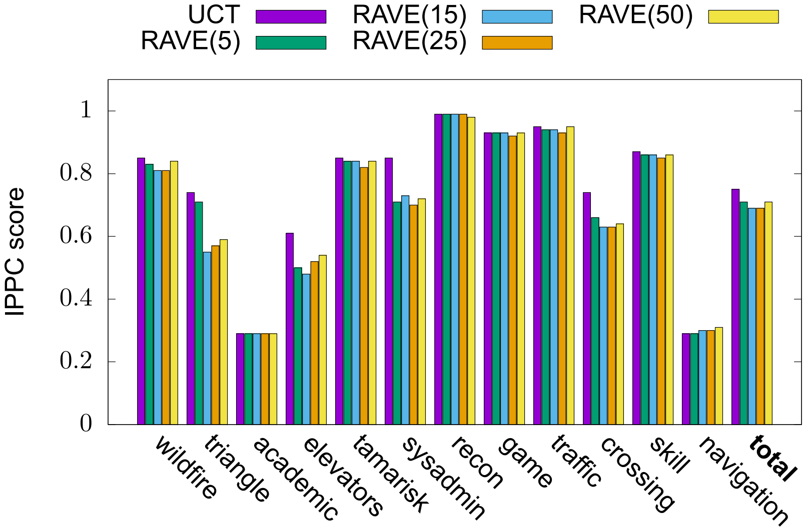
Rapid Actio Value Estimation - Idea

- First introduced in 2007 for computer go
- Use soft cutoff

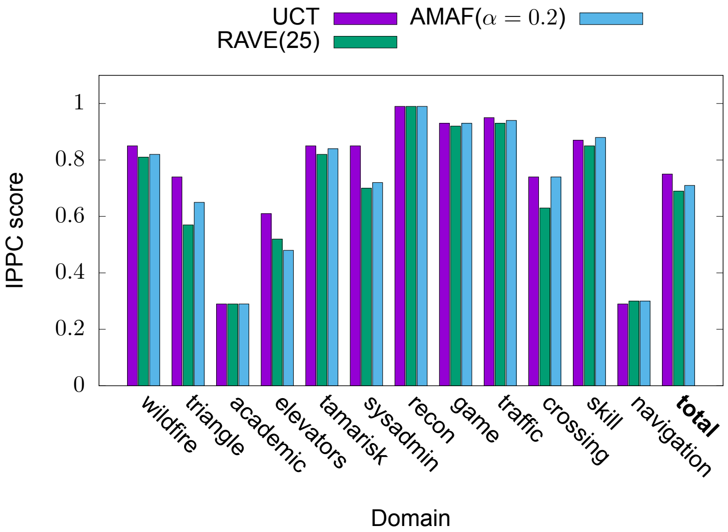
$$\alpha = max\left\{0, \frac{V - v(n)}{V}\right\} \tag{5}$$

- Use UCT for often visited nodes and AMAF score for less-visited

Rapid Action Value Estimation - Results

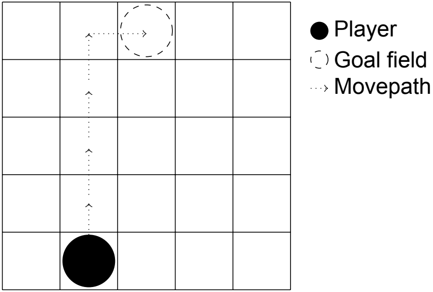


All Moves as First - Conclusion



Rapid Action Value Estimation - Problems

- PROST uses problem description **with** conditional effects
- Also no preconditions given
- PROST description is **more general**



In PROST:

- Action: $move\_up$

In e.g. computer chess

- Action: $move\_a2\_to\_a3$

Predicate Rapid Action Value Estimation

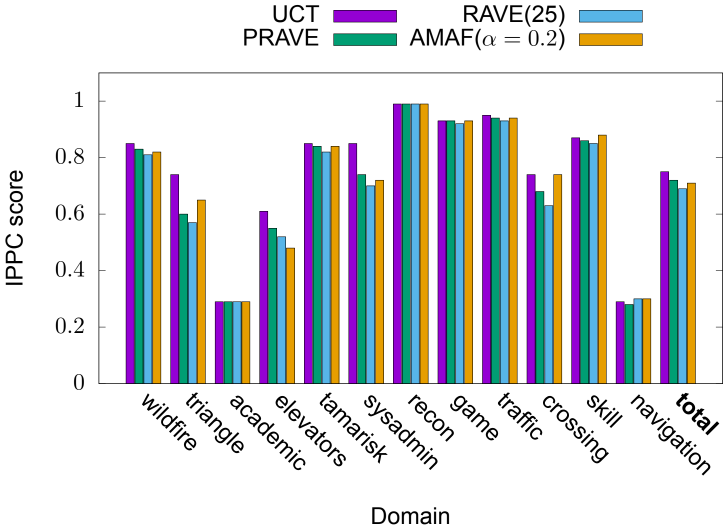
- A state has **predicates** that give some context
- **Idea** Use predicates to find similar states and use their score

$$Q_{PRAVE}(s, a) = \frac{1}{N} \sum_{p \in P} Q_{RAVE}(p, a) \tag{6}$$

- and weight with

$$\alpha = \left\{ 0, \frac{V - v(n)}{V} \right\} \tag{7}$$

All Moves as First - Conclusion - Revisited



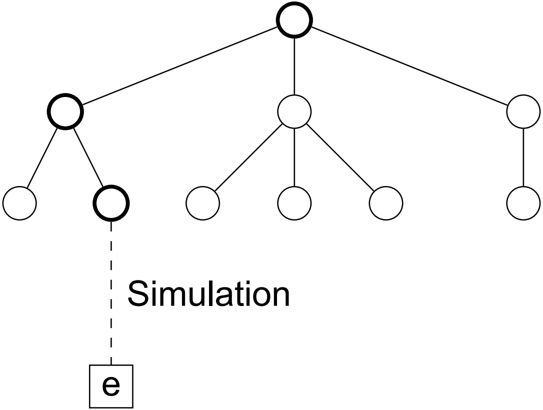
Overview

Tree-Policy Enhancements
- All Moves as First
  - α-AMAF
  - Cutoff-AMAF
- Rapid Action Value Estimation

Default-Policy Enhancements
- Move-Average Sampling Technique

Conclusion

What is a Default Policy?

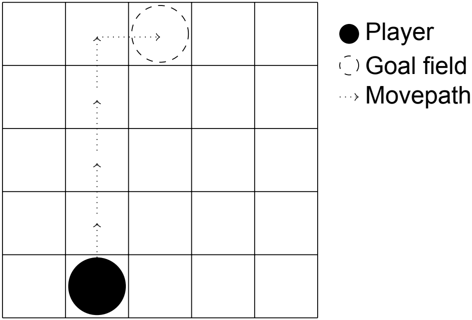


- Simulate the outcome of a trial
- Basic default policy: **random walk**

X-Average Sampling Technique

- Use **tree knowledge** to bias default policy towards moves that are more goal-oriented

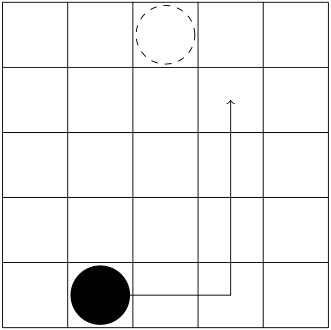
Move-Average Sampling Technique - Idea - Sample Game



- Introduce $Q(a)$
- Use moves that are good **on average**
- Choose action according to:

$$P(a) = \frac{e^{\frac{Q(a)}{\tau}}}{\sum_{b \in A} e^{\frac{Q(b)}{\tau}}} \quad (8)$$

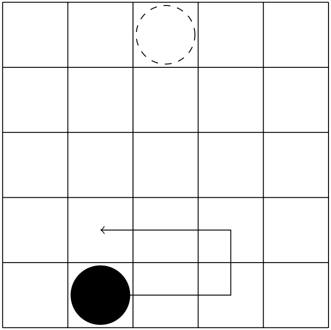
Move-Average Sampling Technique - Idea - Example



Actions: *r,r,u,u,u*

$Q(r) = 1; N(r) = 2$

$Q(u) = 6; N(u) = 3$



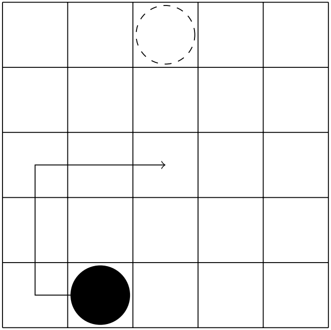
Actions: *r,r,u,l,l*

$Q(r) = 2; N(r) = 4$

$Q(u) = 7; N(u) = 4$

$Q(l) = 3; N(l) = 2$

Move-Average Sampling Technique - Idea - Example (2)

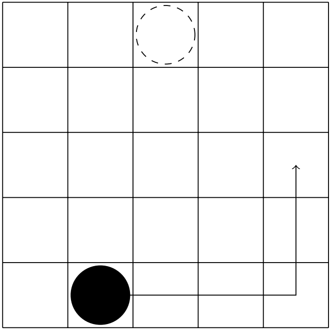


Actions: *l,u,u,r,r*

$Q(r) = 7; N(r) = 6$

$Q(u) = 8; N(u) = 6$

$Q(l) = 2; N(l) = 3$



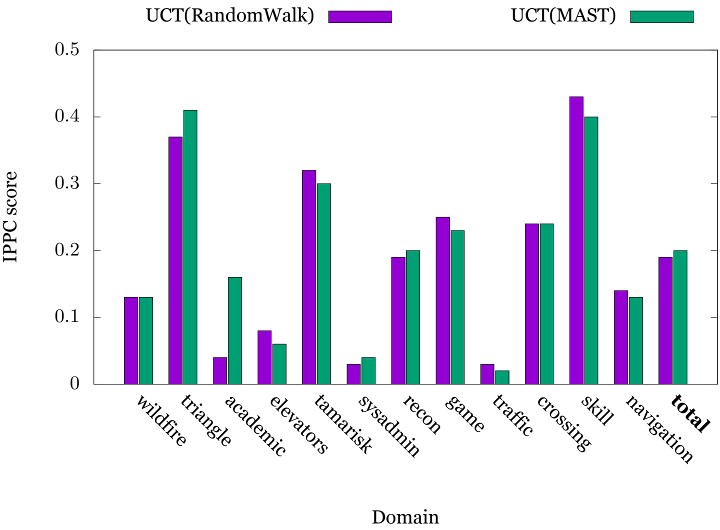
Actions: *r,r,r,u,u*

$Q(r) = 7; N(r) = 9$

$Q(u) = 9; N(u) = 8$

$Q(l) = 2; N(l) = 3$

Move-Average Sampling Technique - Results



Overview

Tree-Policy Enhancements
- All Moves as First
  - α-AMAF
  - Cutoff-AMAF
- Rapid Action Value Estimation

Default-Policy Enhancements
- Move-Average Sampling Technique

Conclusion

Conclusion

- Tree-policy enhancements
  - α-AMAF and RAVE performe worse than standard UCT
  - PRAVE performs slightly better but still worse than standard UCT
- Default-policy enhancements
  - MAST outperforms RandomWalk

Questions?

m.neidinger@unibas.ch