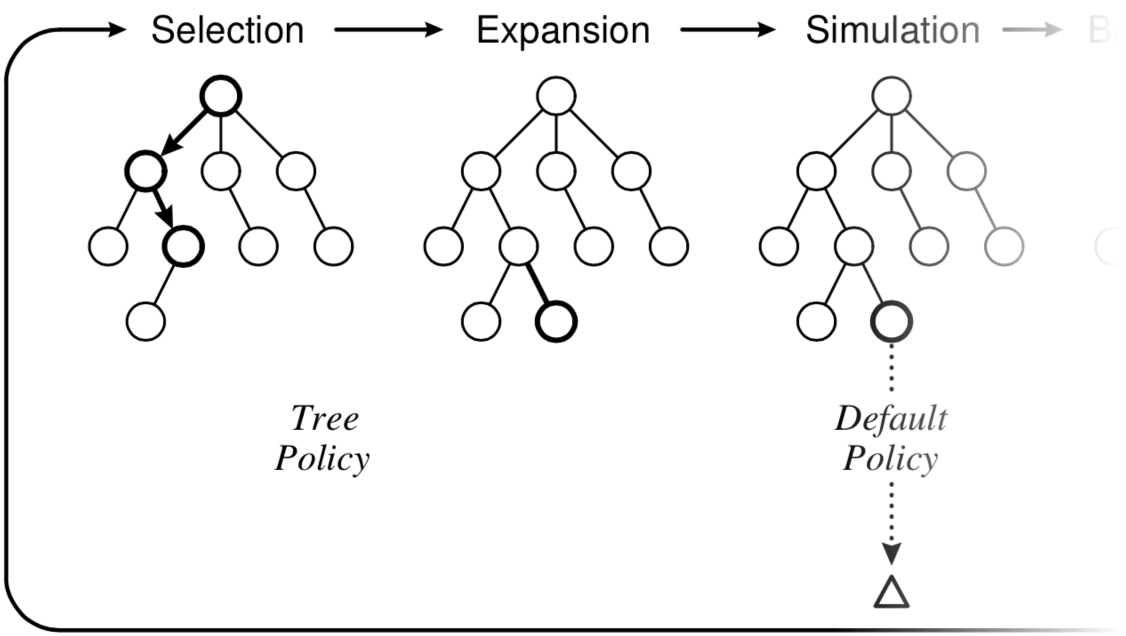
Selection Enhancements

Florian Weber

Motivation

- Selection assigns some numeric score to each action in order to balance exploration with exploitation
  - We heard already of the use of UCB1 for node selection in UCT
  - In the following this approach is called MCTS
- Enhancement:
  - Bias the search in
    - a specific direction,
    - certain actions,
    - reward estimates



Agenda

First Play Urgency (FPU)

- How MCTS does it: Visit unexplored nodes in random order
  - Trees with large branching factors
    => exploitation will rarely occur

- Idea of FPU:
  - Score unvisited nodes with a fixed value
  - Score visited ones with UCB1
    => early exploitations are encouraged

Transpositions

- How MCTS does it: Moves are represented as a search tree.
  - We have a transposition when several paths are leading to the same position
  - The statistics are dependent on the path to the position

- Idea of Transpositions: A Directed Acyclic Graph (DAG) is used to store the information about transpositions
  - Identifying transpositions is easy. Mostly transposition tables are used
  - Statistics are cumulated for the position – independently of where it occurs in the tree

Move Groups

- How MCTS does it: A lot of simulation is needed to find groups with a highly correlated expected reward (high branching factor)

- Idea of Move Groups: Similar moves (hence correlated actions) are grouped to reduce the branching factor
  - All actions are collected into Move Groups
  - UCB1 decides which of them is used to select a move from

Monte Carlo Paraphrase Generation (MCPG)

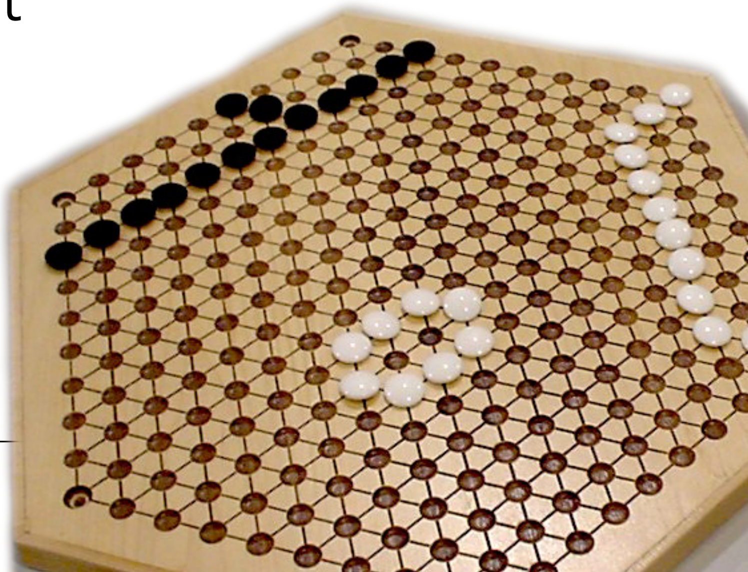
- How MCTS does it: The (average) score expectation is used for selection
- Idea of MCPG: The maximum reachable score for each state is used for selection
- Benefits/where is it used?
  - It is named after it‵s usage in generating paraphrases of natural language statements

Agenda

Decisive and Anti-Decisive Moves

- In many games you can find decisive and anti-decisive moves
  - Decisive move
    - One that leads immediately to a win
  - Anti-decisive move
    - Prevents the opponent from doing a decisive move in the next turn

- Idea: Selection and simulation policy:
  - If either player has a decisive move, play it
  - Otherwise play standard policy



Progressive Bias

- How MCTS does it: The information of nodes visited only a few times is not reliable

- Idea of Progressive Bias: Provide in this case more accurate information via a domain specific heuristic value H_i
  - Term added to the selection formula:

$$f(n_1) = \frac{H_i}{n_i+1}$$

node with index i, visited n_i times

  - The influence of $f(n_1)$ is high when a few games have been played
  - It decreases when more games have been played

Search Seeding

- How MCTS does it: Every node is initialized with zero wins & visits

- Idea of Search Seeding: The statistics at each node is initialized (=seeded) according to some heuristic knowledge. It’s like a warm up for the nodes.
  - This extra knowledge can safe time since the need for simulation may be reduced

- Different approaches
  - Adding virtual wins & visits, prior estimates would remain permanently
  - Some transient estimate blended into the regular value → see Progressive Bias

Opening Books

1) Using Opening Books for MCTS:

- Employing the book till an unlisted position is reached

2) Build your own ones:

- Generating an opening book with MCTS
  - Meta Monte-Carlo Search
    - Generate book online/offline

History Heuristic

- Idea: Using information about moves previously played
- Grandfather heuristic: history information is used to initialize action value estimates for new nodes
  - Expected return $Q_{grand}(s_t, a) = Q_{UCT}(s_{t-2}, a)$ with state s, action a
- General game player CadiaPlayer: history heuristic used for seeding node values

Agenda

1 Domain Independent Enhancements

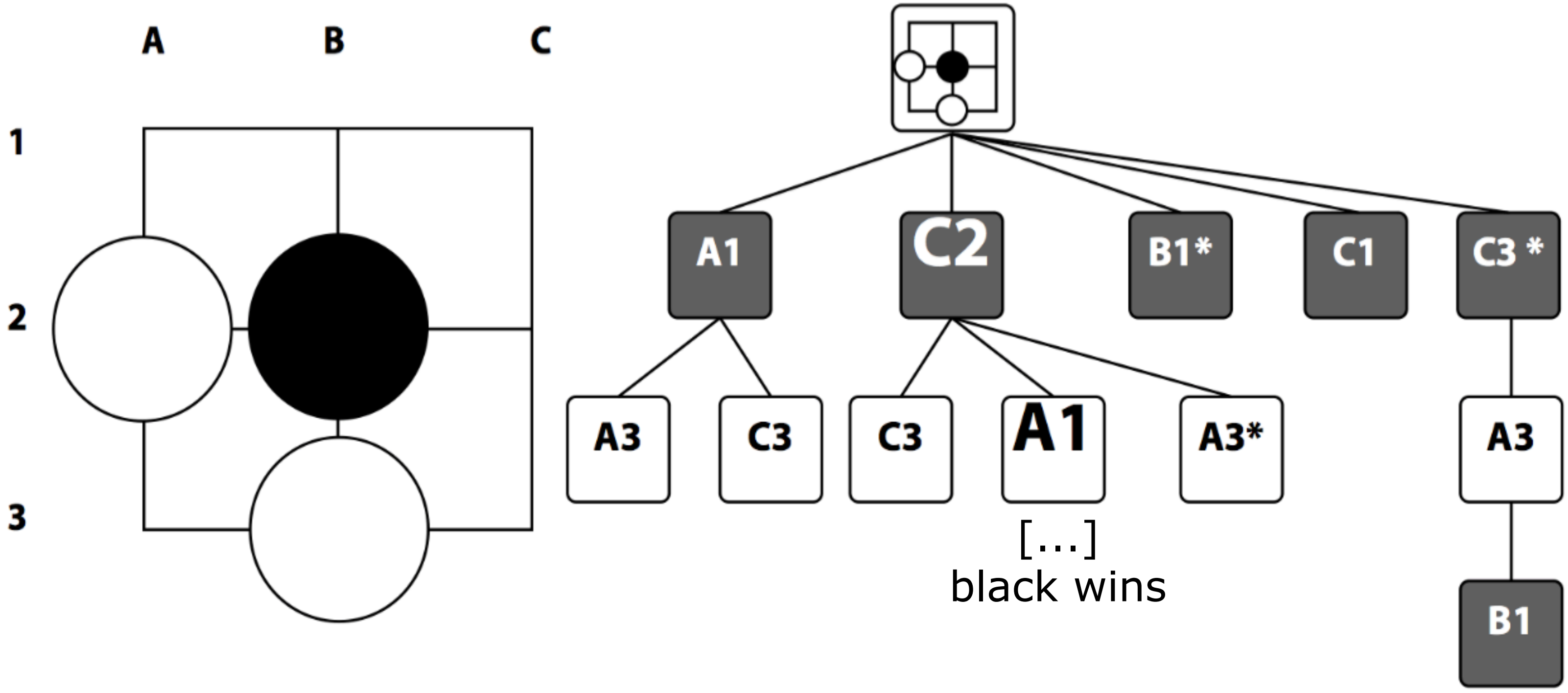
2 Domain Dependent Enhancements

3 All Moves As First Heuristic (AMAF)

4 Rapid Action Value Estimation (RAVE)

All Moves As First Heuristic (AMAF)

- How MCTS does it: after the play-out the estimated rewards for selected nodes are updated (here: C2, A1)



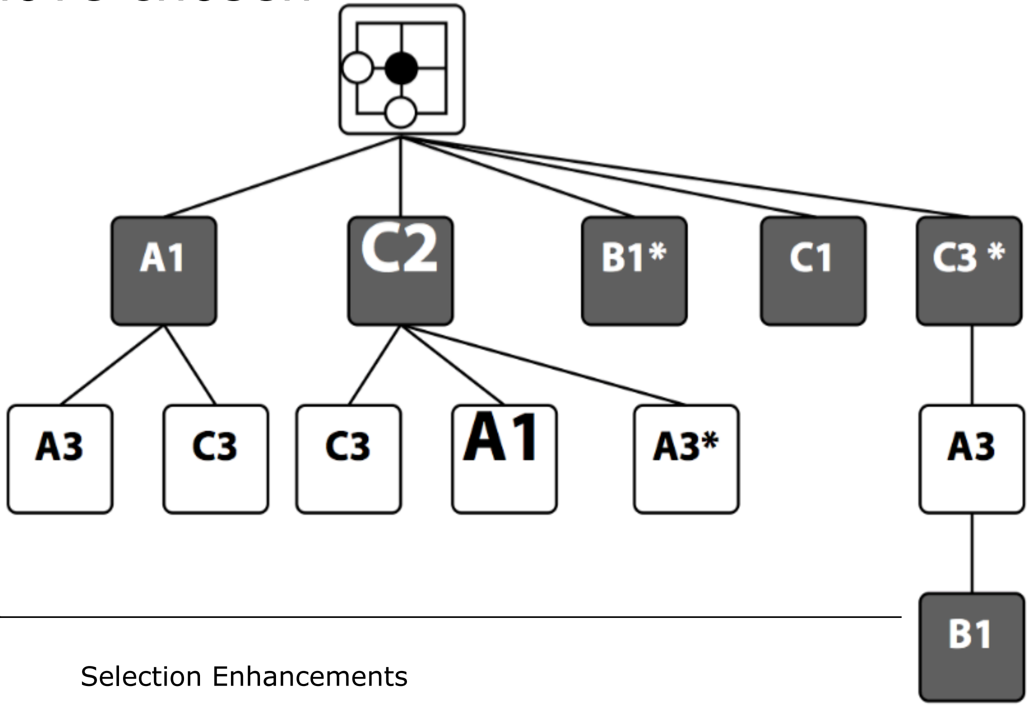
All Moves As First Heuristic (AMAF)

- How MCTS does it: after the play-out the estimated rewards for selected nodes are updated (here: C2, A1)
- Idea of AMAF: the reward estimate for an action a from a state s is updated whenever a is encountered during a play-out, even if a was not the actual move chosen

Play-out Path:

Black **C2** (UCT)
White **A1** (UCT)
Black B1 (*)
White A3 (*)
Black C3 (*)

Win for black



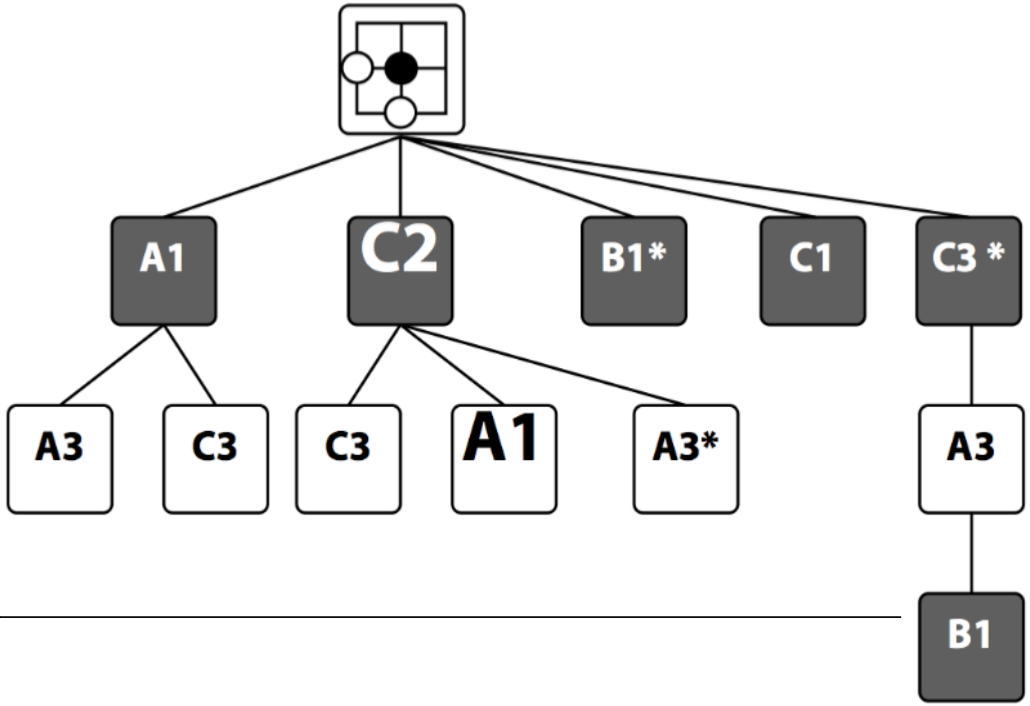
Some-First AMAF

- Idea: The history of the play-out moves is truncated after the first m random moves
  - For $m=0$ => standard UCT
  - For $m>play\text{-}out\ length$ => standard AMAF

Play-out Path:

Black **C2** (UCT)
White **A1** (UCT)
Black B1 (*)
White A3 (*)
Black C3 (*)

Win for black



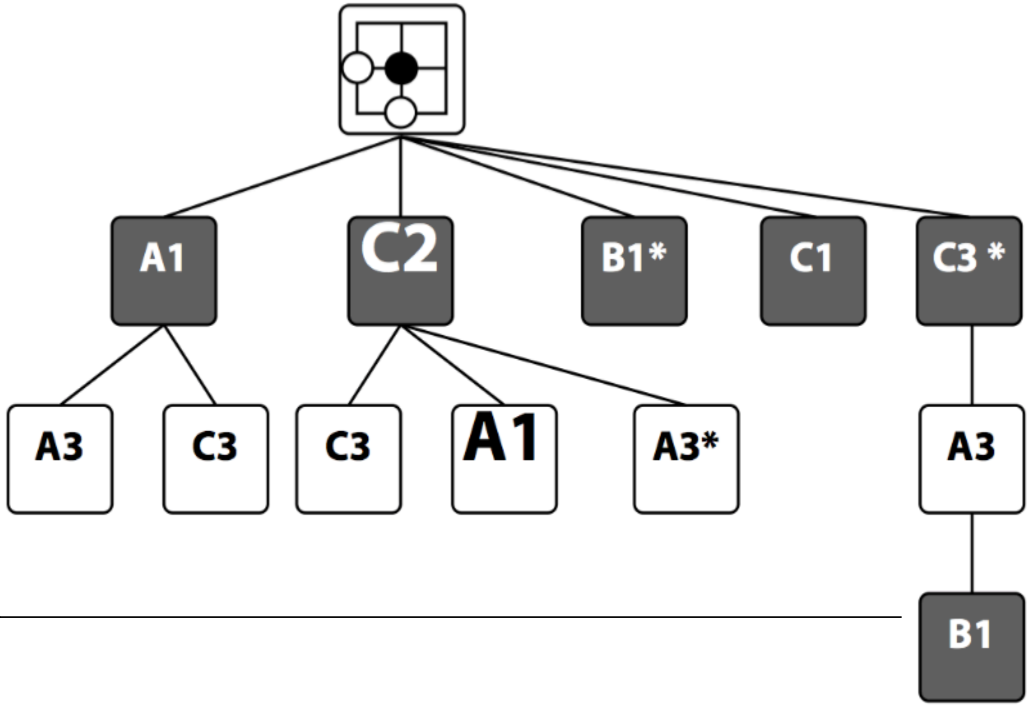
Some-First AMAF

- Idea: The history of the play-out moves is truncated after the first m random moves
  - For $m=2$ …

Play-out Path:

Black **C2** (UCT)
White **A1** (UCT)
Black B1 (*)
White A3 (*)
~~Black C3 (*)~~

Win for black



Cutoff AMAF

- Idea: Warm up with AMAF data, then use more accurate UCT
  - Update statistics for the first k simulations
  - Afterwards standard UCT is used
  - For $k=0$ => standard UCT
  - For $sufficiently\ high\ k$ => standard AMAF

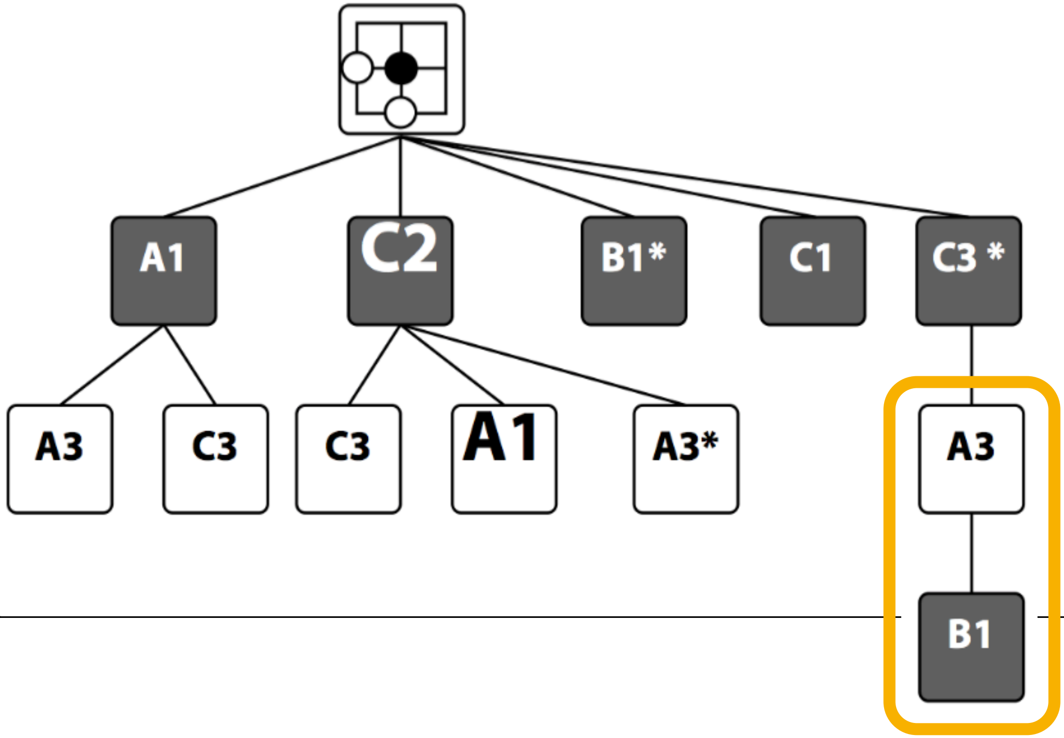
Permutation AMAF

- Idea: Also updates nodes that can be reached by permutations of moves that preserve the state reached
  - Stone colors and player alternation are preserved
  - In our example additional to the AMAF updates, the nodes A3 and B1 are updated

Play-out Path:

Black **C2** (UCT)
White **A1** (UCT)
Black B1 (*)
White A3 (*)
Black C3 (*)

Win for black



α-AMAF

- Idea: Blends UCT score U with AMAF score A
  - The score is calculated like this:

    $$\alpha A + (1 - \alpha) U$$
  - For $\alpha = 0$ => standard UCT
  - For $\alpha = 1$ => standard AMAF

  - So which α should we choose?

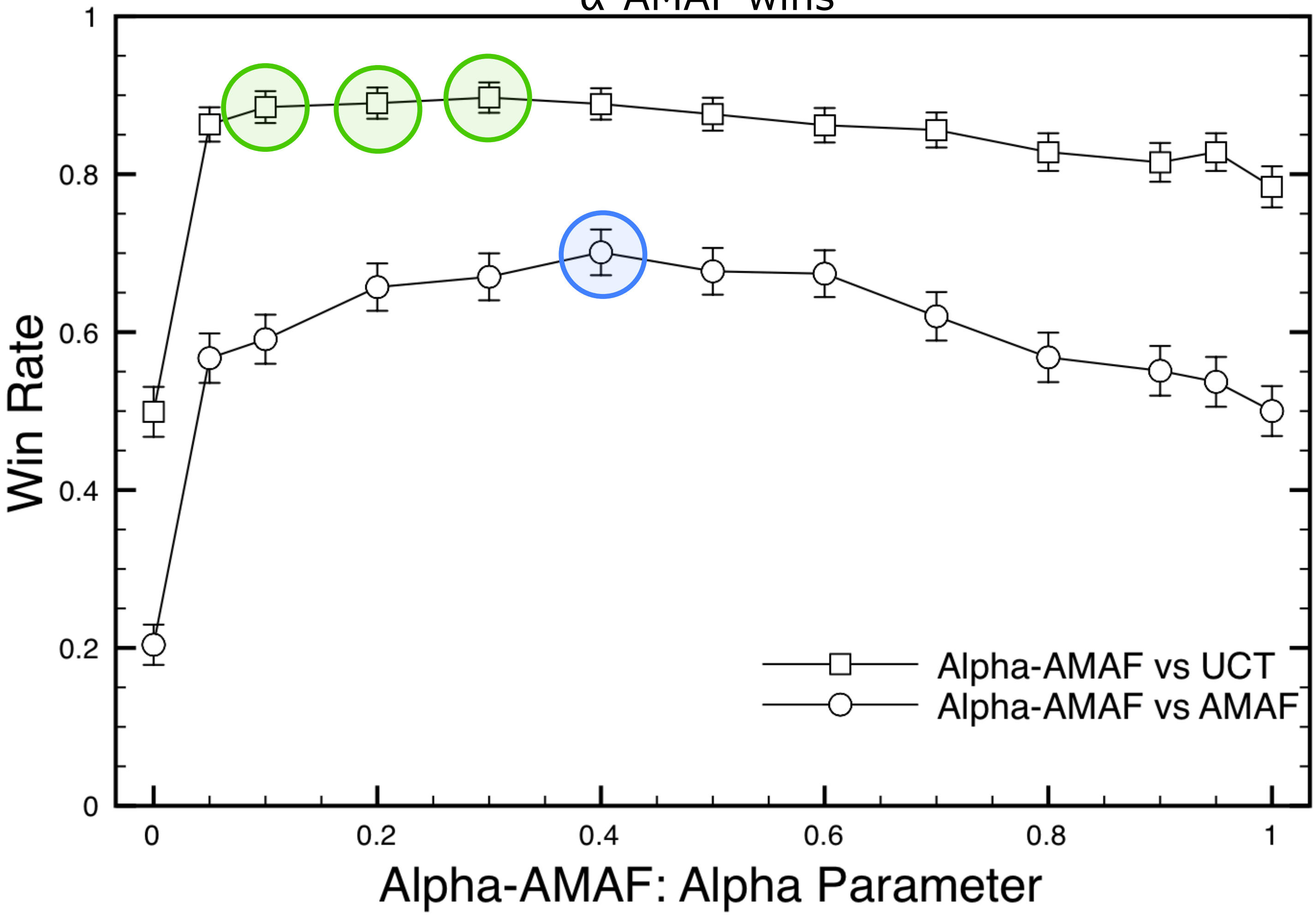
All Moves As First Heuristic (AMAF)

α–AMAF

- Alpha parameter:

$$\alpha A + (1 - \alpha) U$$

α-AMAF wins



α-AMAF vs. standard UCT: α=0.1, 0.2, or 0.3 slightly better

α-AMAF vs. standard AMAF: α=0.4 is best

performance decreases at the extremes 0 and 1

Agenda

1 Domain Independent Enhancements

2 Domain Dependent Enhancements

3 All Moves As First Heuristic (AMAF)

4 Rapid Action Value Estimation (RAVE)

Rapid Action Value Estimation (RAVE)

- Idea: similar to α-AMAF but
  - Instead of using a fixed α, the α value used at each node decreases with each visit:

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

$v(n)$ play-outs through node,
fixed integer $V > 0$

  - Exploited areas of the tree will use the accurate statistics (UCT) more than unexploited areas of the tree

  - As a reminder:
    - α-AMAF: $\alpha A + (1 - \alpha)U$
    - For $\alpha = 0$ => standard UCT
    - For $\alpha = 1$ => standard AMAF

Variants

(1) Killer RAVE: Only the most important moves are used for the RAVE updates

- Important moves:
  - Those which have been flagged as strong in other parts of the tree
- Benefits/where is it used?
  - More beneficial than standard RAVE for game Havannah

(2) RAVE-max: More robust variant of RAVE

- Replacing the RAVE value of move j, Y_j with
  - $max(X_j,\ Y_j)$, $X_j{=}mean(j)$
- Able to correct extreme cases of RAVE underestimation
- Benefits/where is it used?
  - Good for degenerated cases for game Sum of Switches
  - Less successful for Go

PoolRAVE

- Idea: Modify MCTS as follows:
  - Build a pool of the k best moves according to RAVE
  - Choose one move m from the pool
  - Play m with a probability p, else the default policy

- Benefits/where is it used?
  - The probability p is used to prevent being too deterministic
    - It helps to keep the diversity of the simulations when biasing Monte-Carlo

	Go	Havannah	Arimaa	Sum of Switches	Cadia Player
UCT	X	X	X	X	X
First Play Urgency	X				
Move Groups	X				
Transpositions	X		X		
MCPG					
(Anti)Decisive Moves	X	X			
Progesssive Bias	X				
Opening Books	X				
Search Seeding	X				
History Heuristic	X		X		X
AMAF	X	X			
RAVE	X	X	X	X	X
Killer RAVE		X			
RAVE-max				X	
PoolRAVE	X	X			

Thank you very much for you attention!

QUESTIONS?