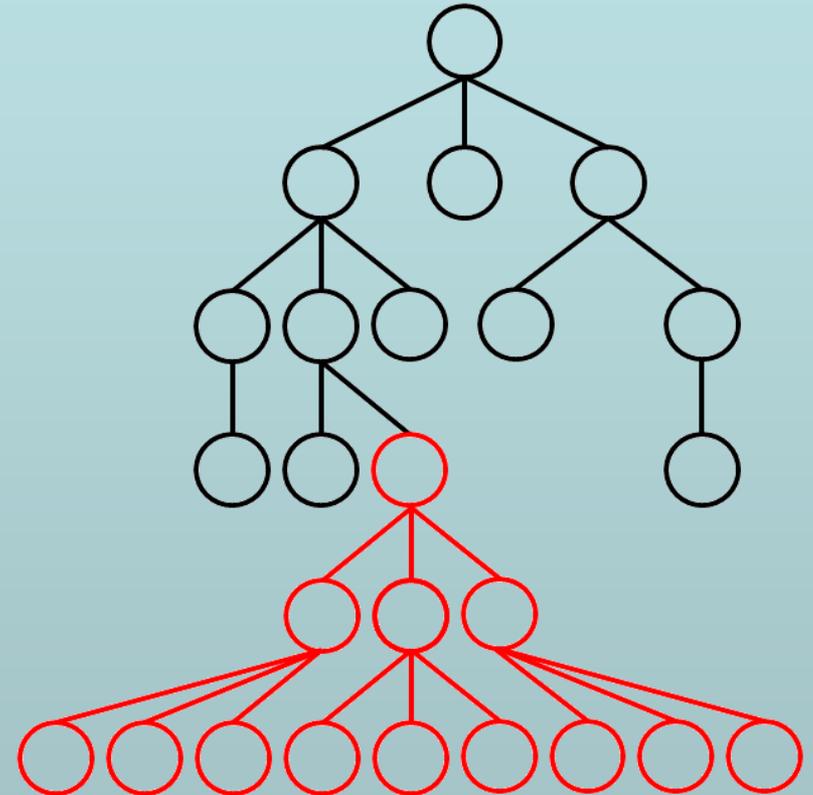


MCTS-Minimax Hybrids



Hendrik Baier · Mark H.M. Winands

Overview

- Motivation

Overview

- Motivation
- MCTS-Minimax Hybrid Algorithms

Overview

- Motivation
- MCTS-Minimax Hybrid Algorithms
 - Test Domains

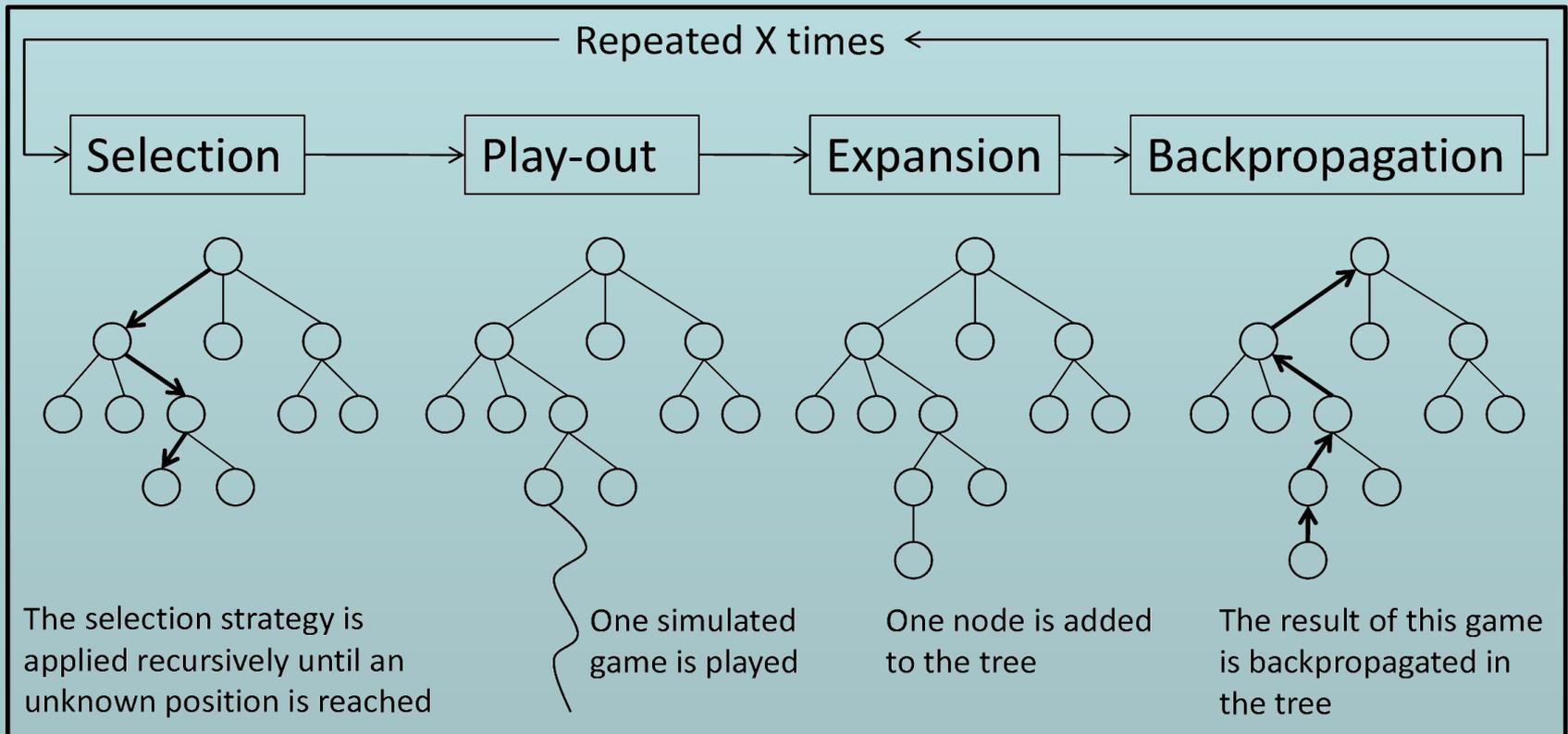
Overview

- Motivation
- MCTS-Minimax Hybrid Algorithms
 - Test Domains
 - Experimental Results

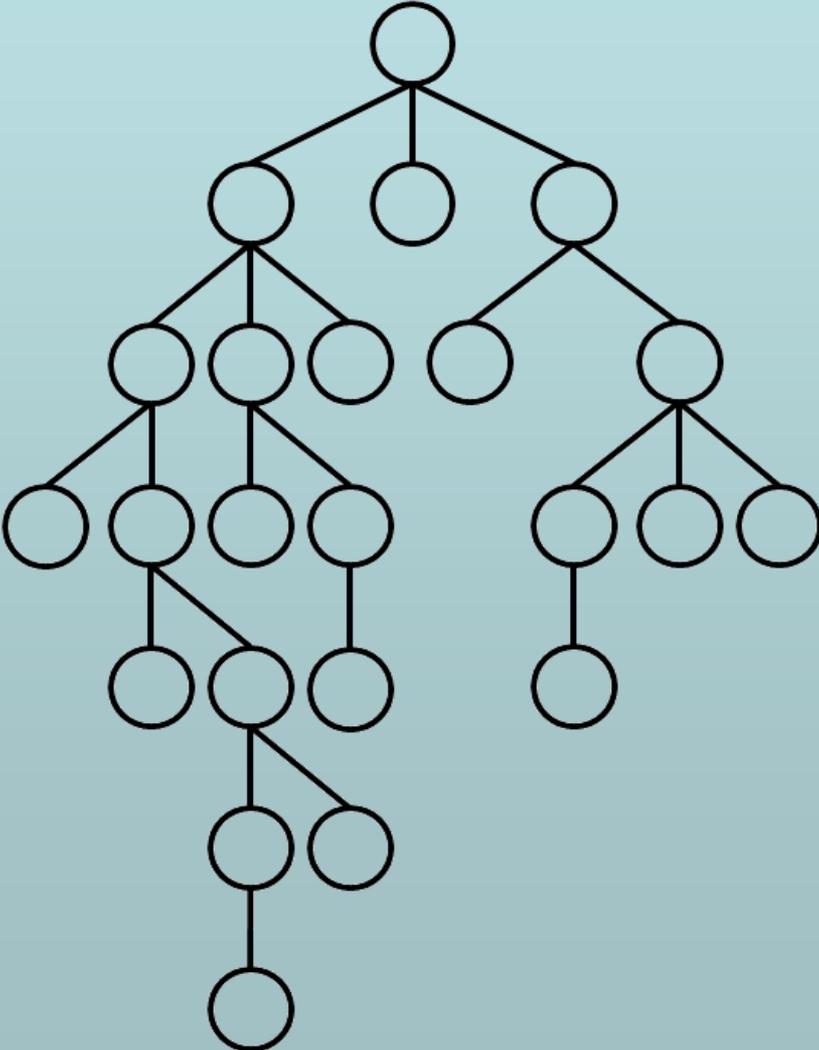
Overview

- Motivation
- MCTS-Minimax Hybrid Algorithms
 - Test Domains
 - Experimental Results
- Conclusions and Future Work

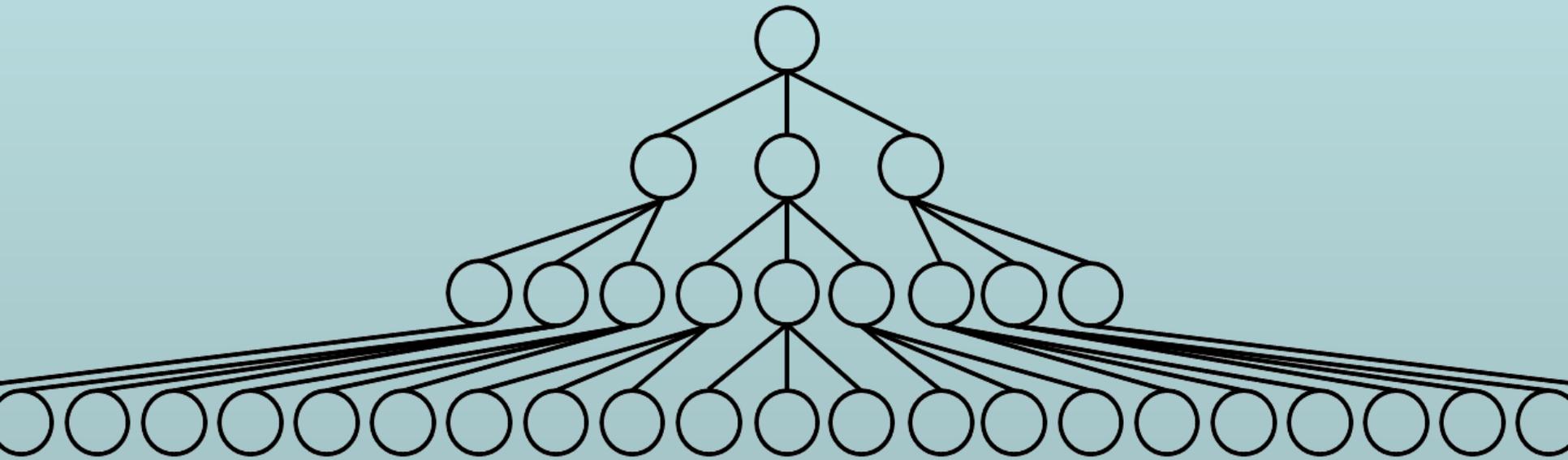
Motivation: Monte-Carlo Tree Search



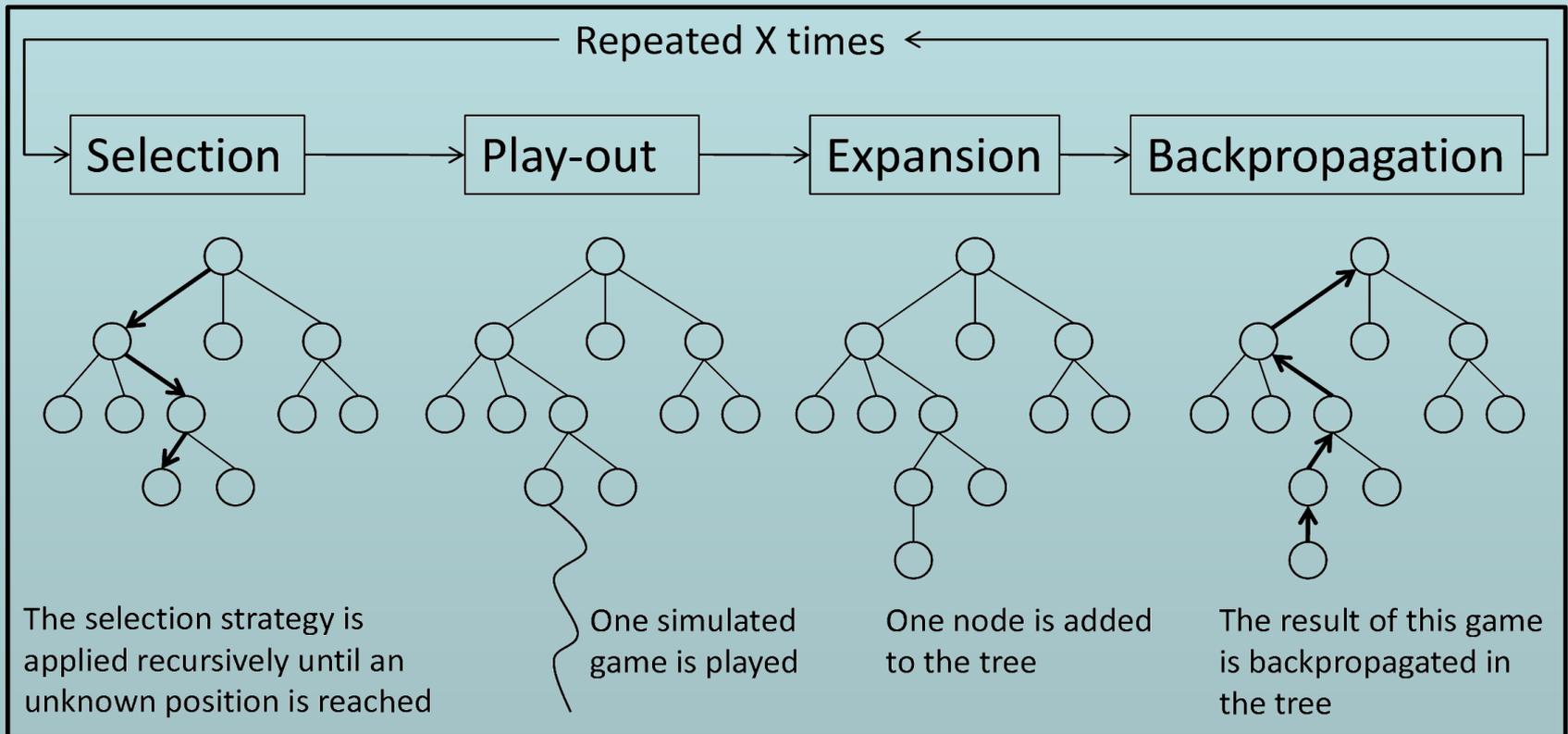
Motivation: Monte-Carlo Tree Search



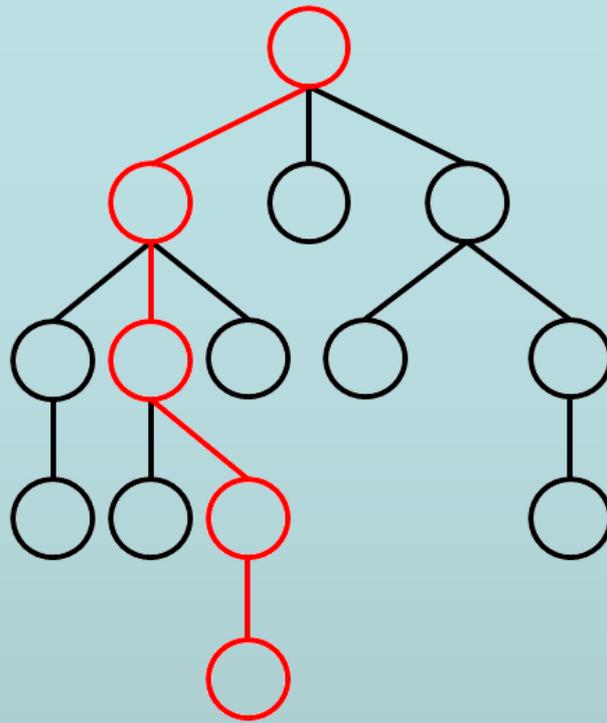
Motivation: Minimax

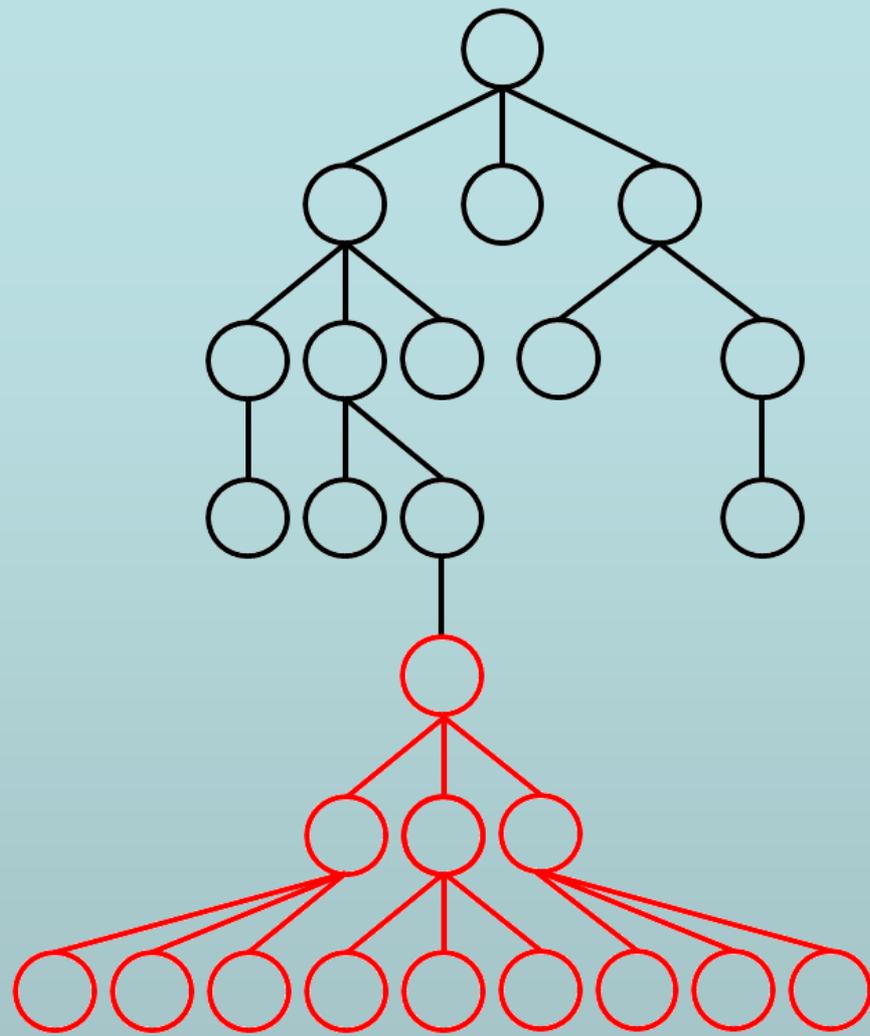


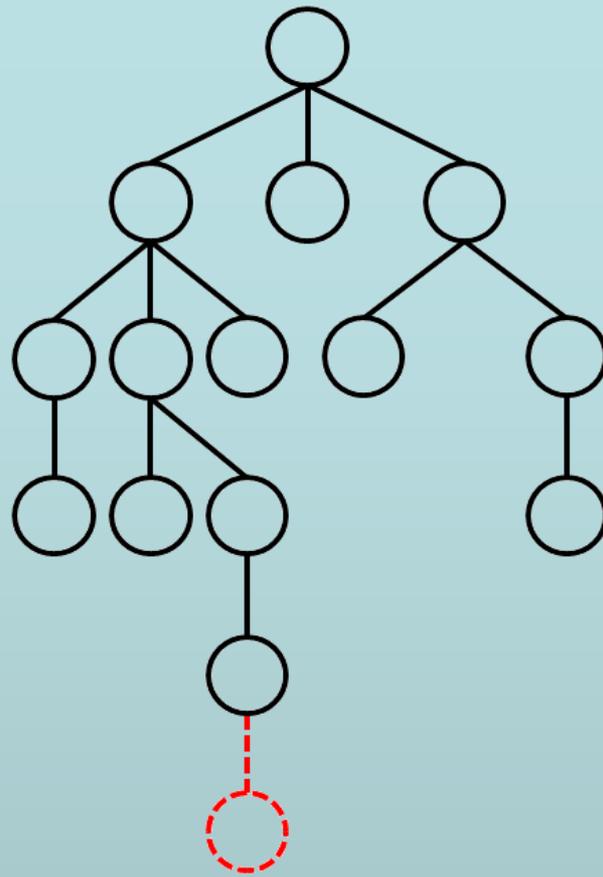
MCTS-Minimax Hybrids

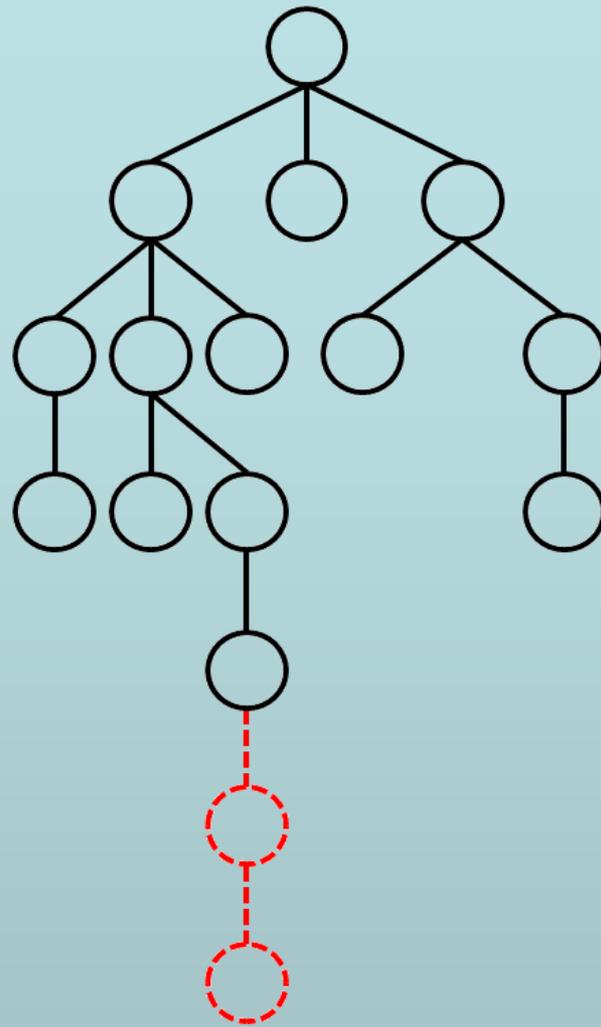


MCTS-Minimax Hybrids: MCTS-MR (Minimax Rollouts)

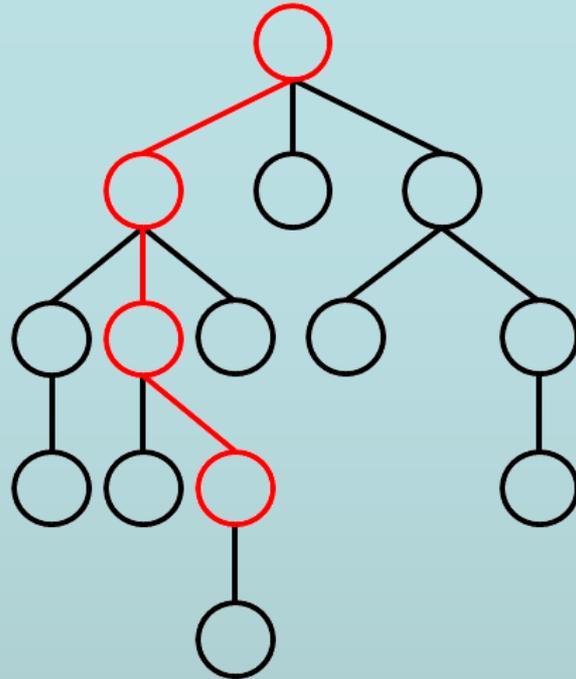


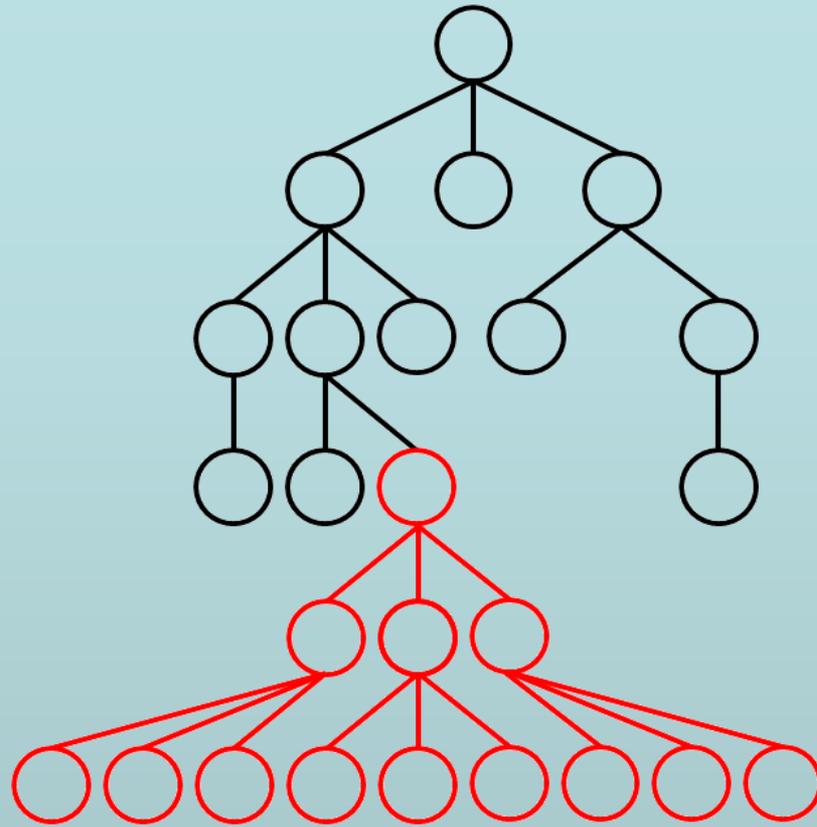


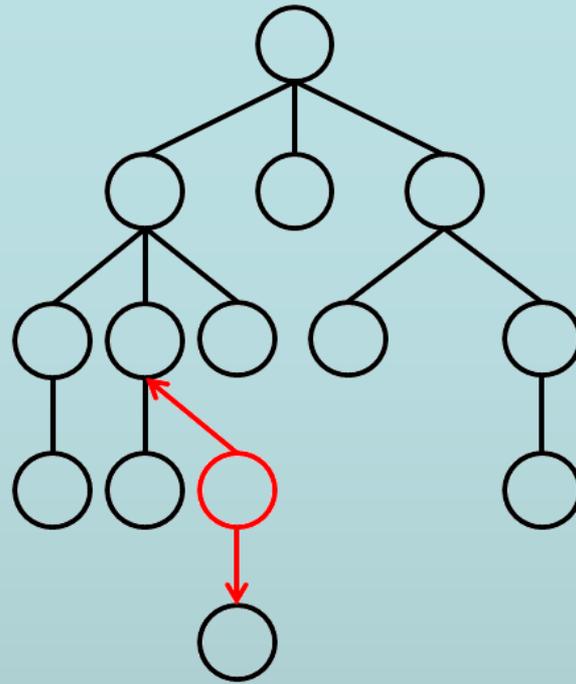




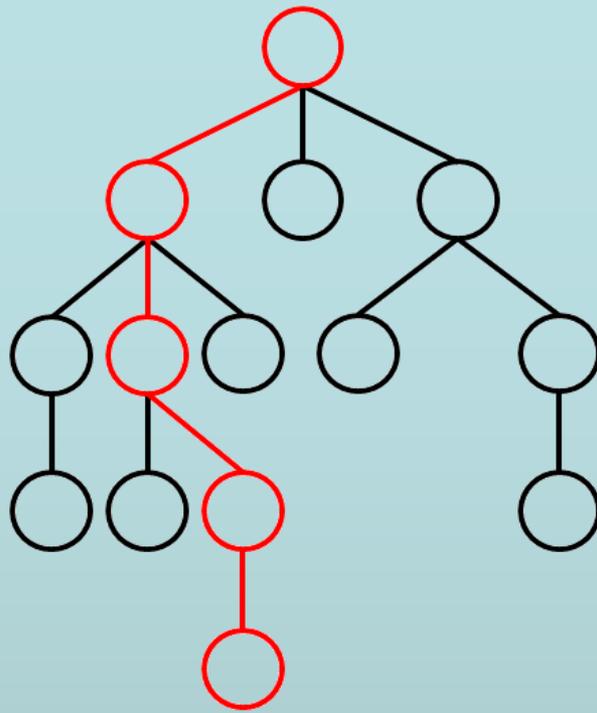
MCTS-Minimax Hybrids: MCTS-MS (Minimax Selection)

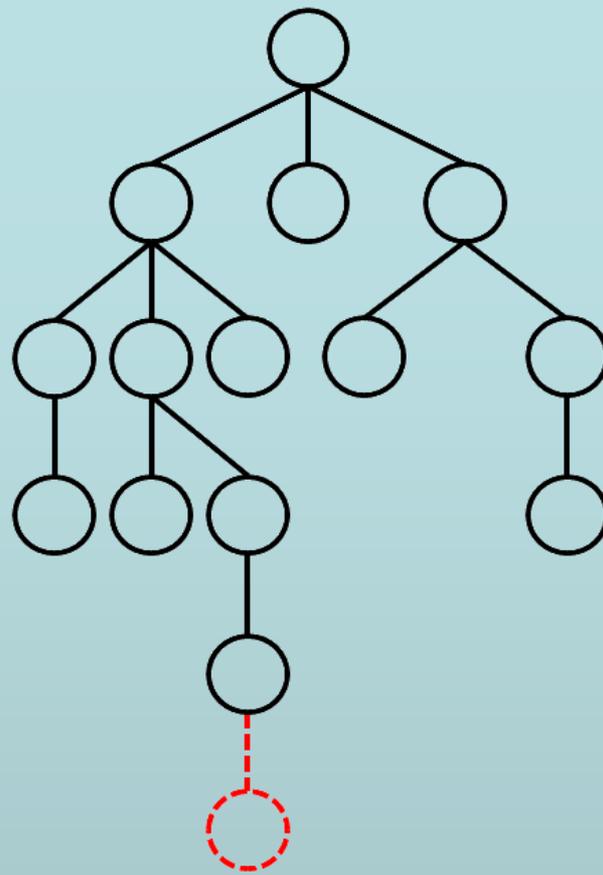


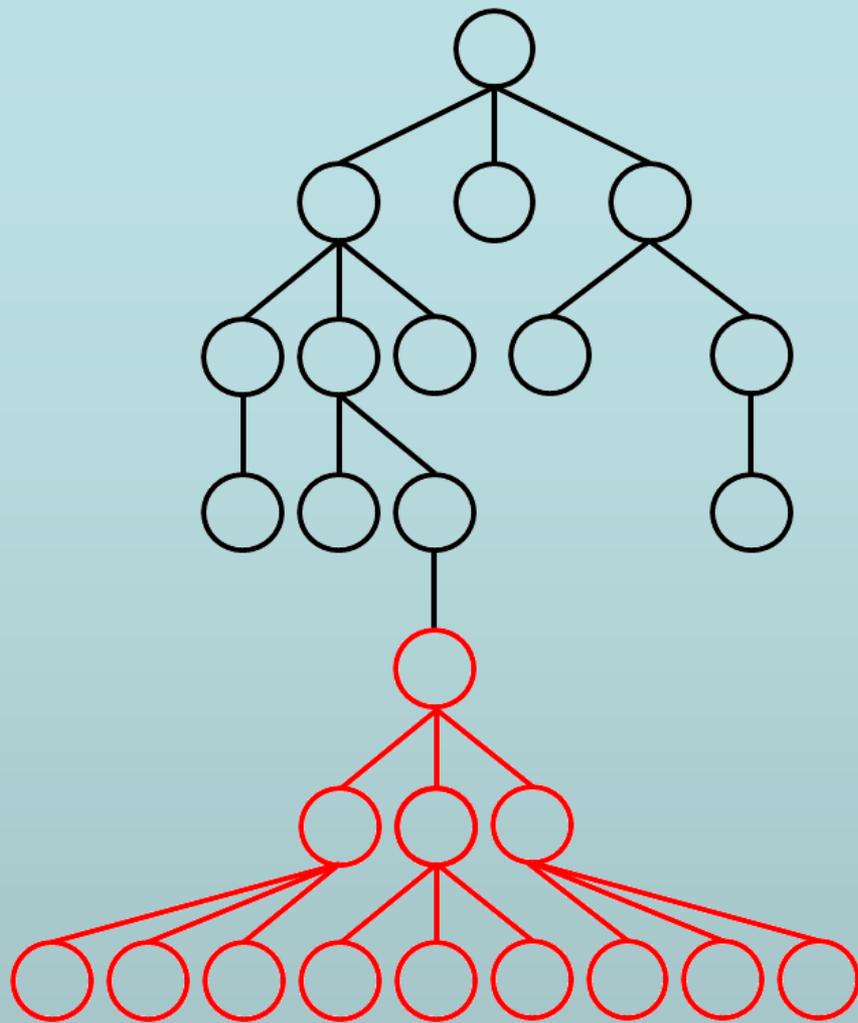


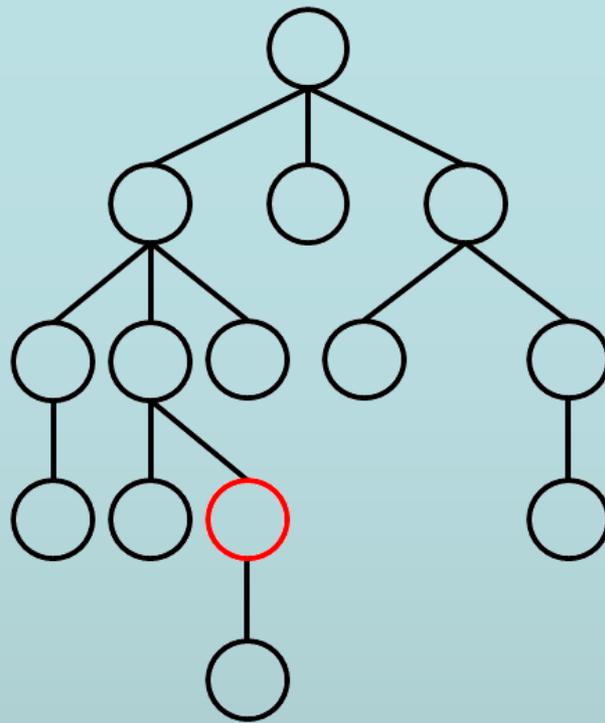


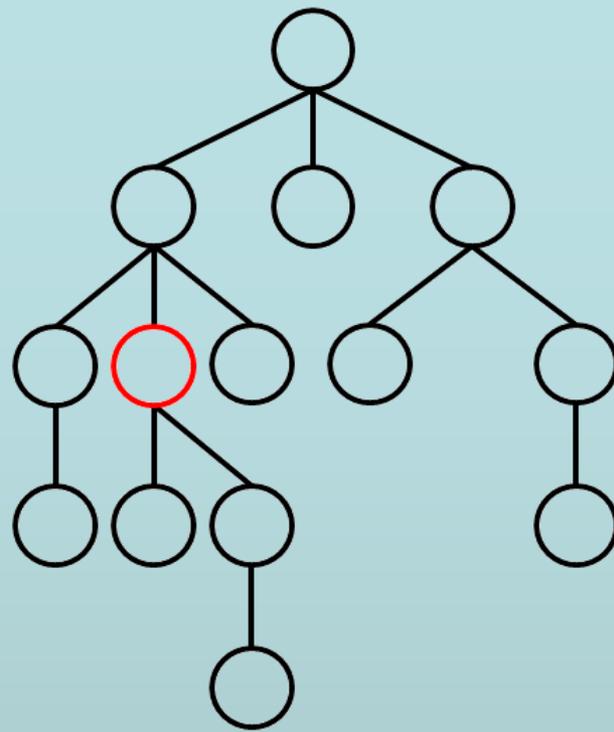
MCTS-Minimax Hybrids: MCTS-MB (Minimax Backpropagation)



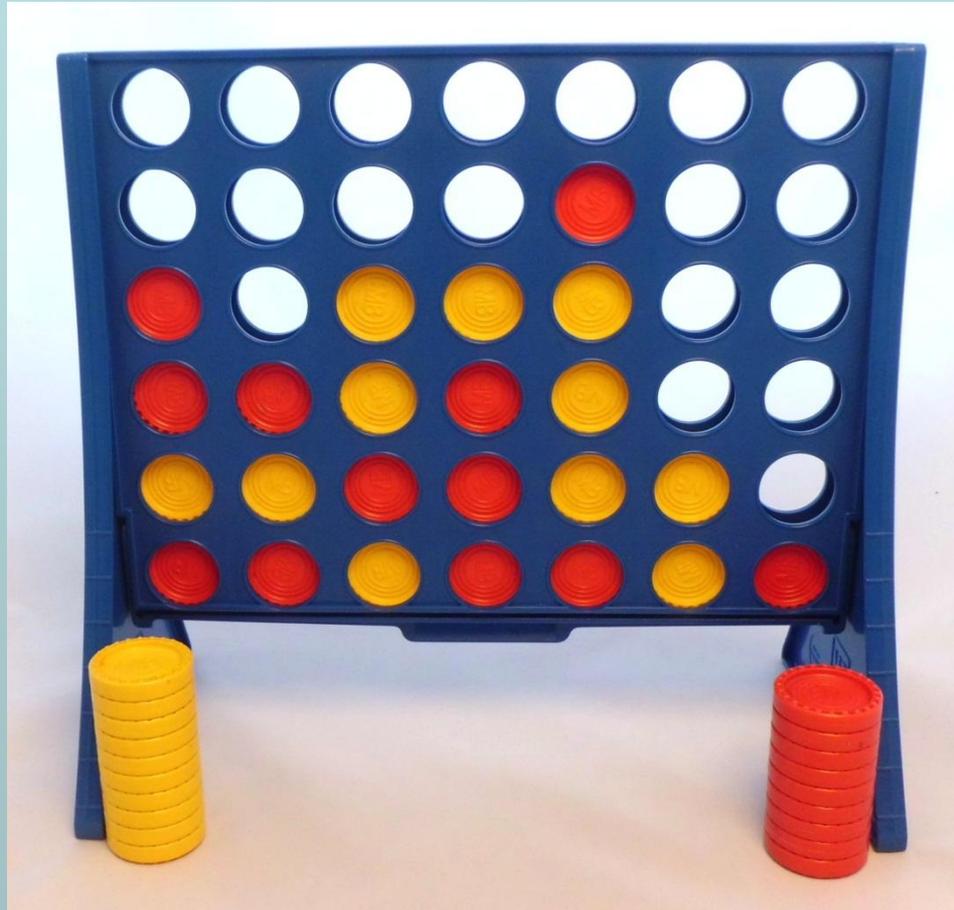








Test Domains: Connect-4



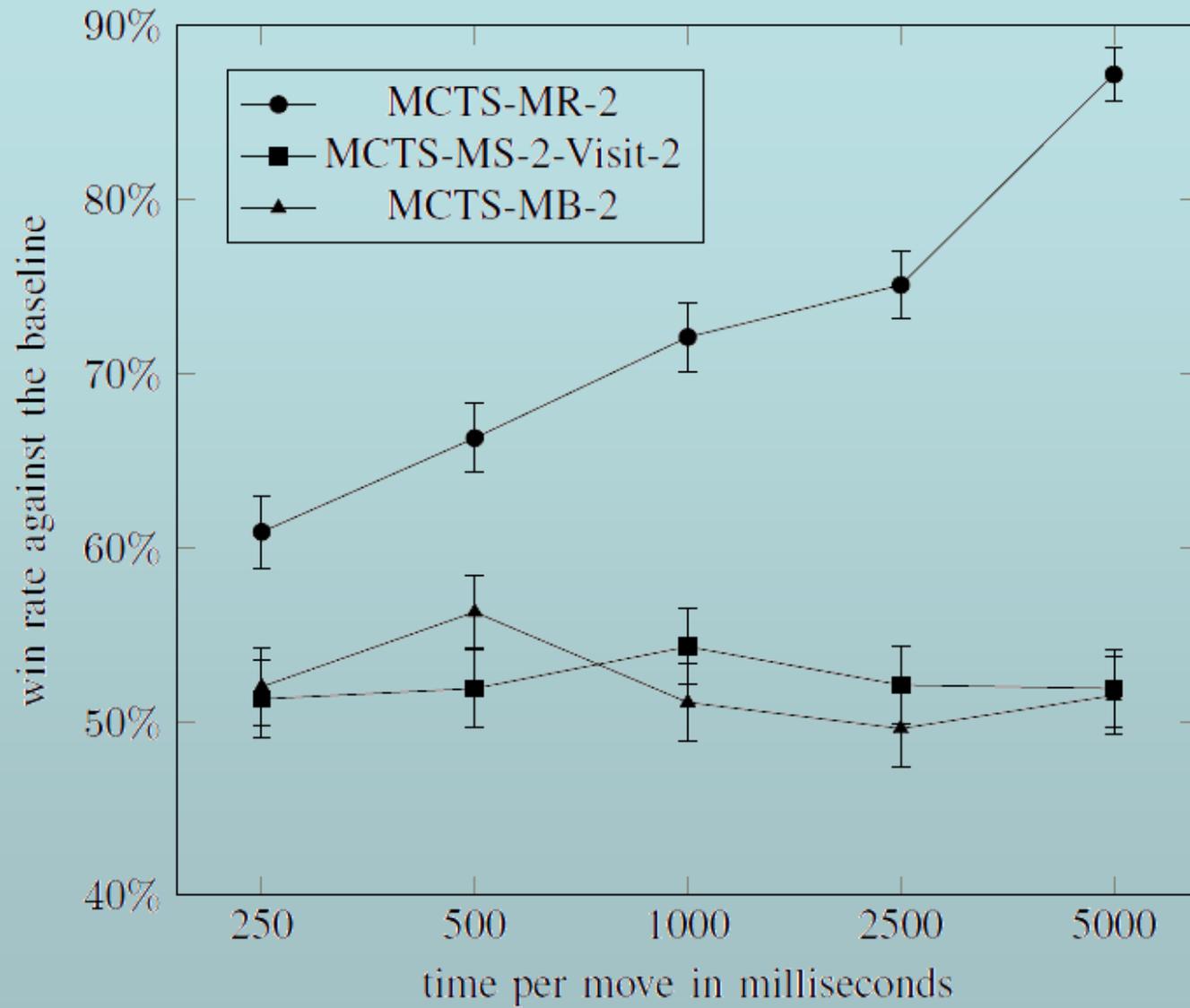
Test Domains: Breakthrough

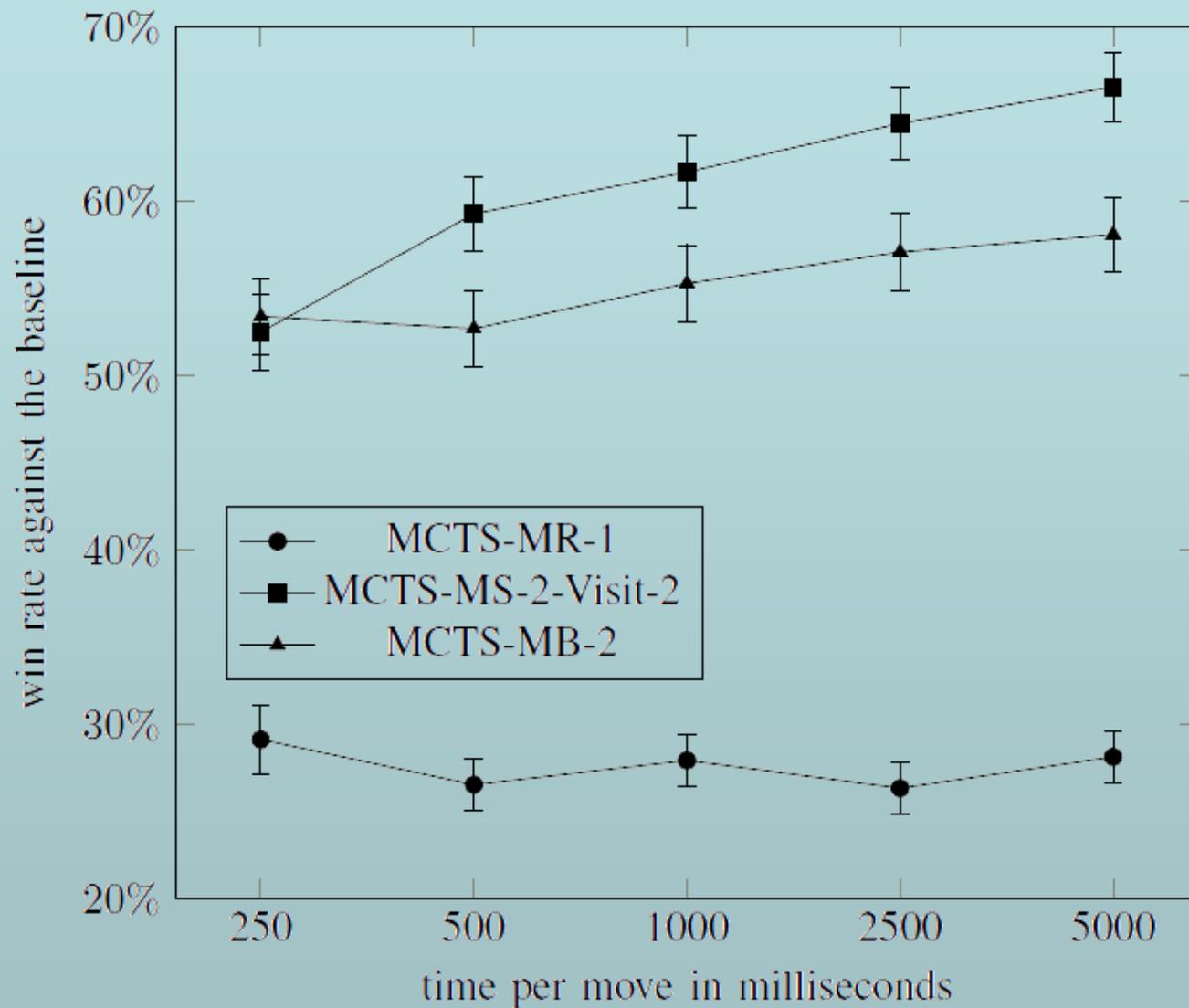


Experimental Results

	Connect 4	Breakthrough
MCTS-MR	73.2%	27.0%
MCTS-MS	53.4%	62.2%
MCTS-MB	52.1%	55.0%

Experimental Results: Influence of Time Controls





Conclusions

- Three knowledge-free ways of integrating minimax into MCTS

Conclusions

- Three knowledge-free ways of integrating minimax into MCTS
 - Newly proposed MCTS-MB and MCTS-MS significantly outperform regular MCTS(-Solver)

Conclusions

- Three knowledge-free ways of integrating minimax into MCTS
 - Newly proposed MCTS-MB and MCTS-MS significantly outperform regular MCTS-Solver
- MCTS-MR seems to be more sensitive to differences between search spaces (at least when used without knowledge)

Future Work

- Examine influence of algorithm properties such as speed and quality of rollouts (here uniformly random)

Future Work

- Examine influence of algorithm properties such as speed and quality of rollouts (here uniformly random)
- Examine influence of game properties such as branching factor, game length, terminal state density, trap density, etc.

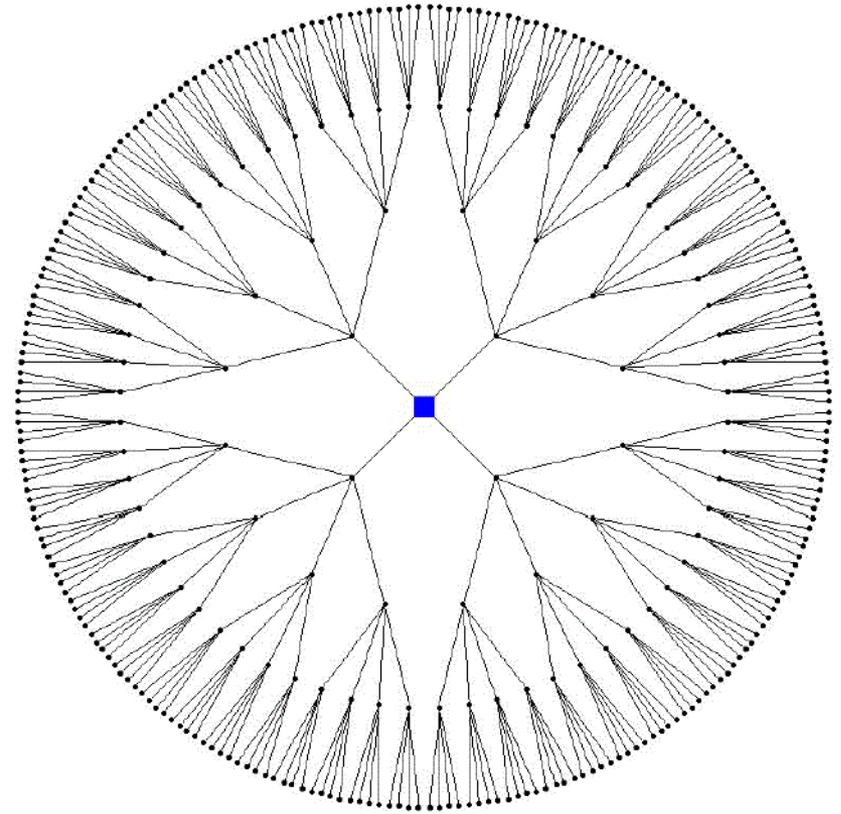
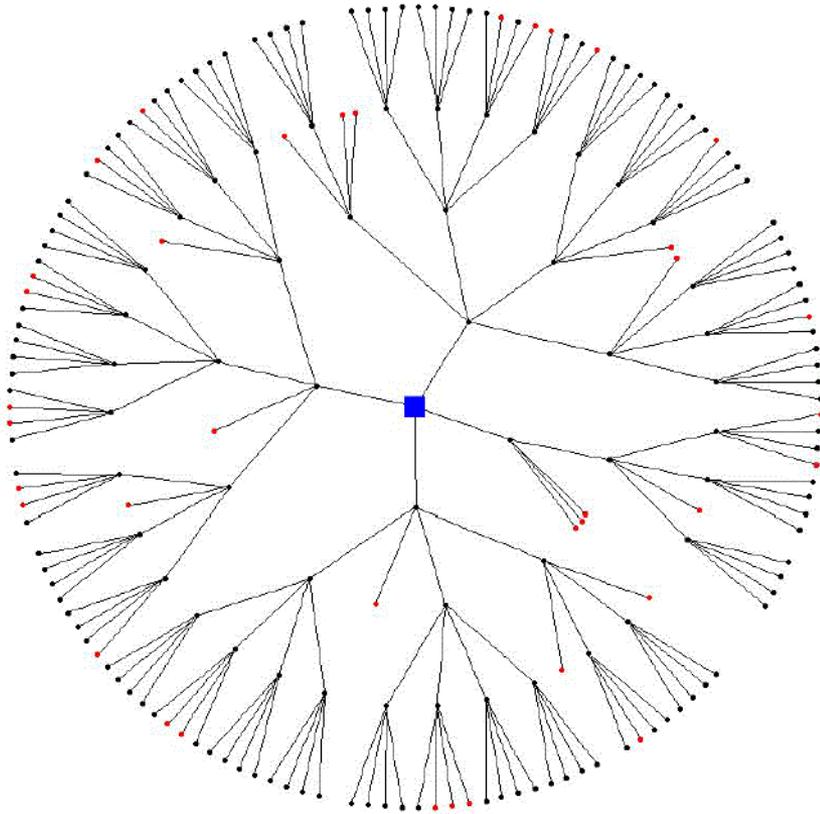
Future Work

- Examine influence of algorithm properties such as speed and quality of rollouts (here uniformly random)
- Examine influence of game properties such as branching factor, game length, terminal state density, trap density, etc.
- Incorporate knowledge in the form of evaluation functions – find ways of combining evaluation results with MCTS rollout returns

Additional Games



Terminal State Density



Questions?

