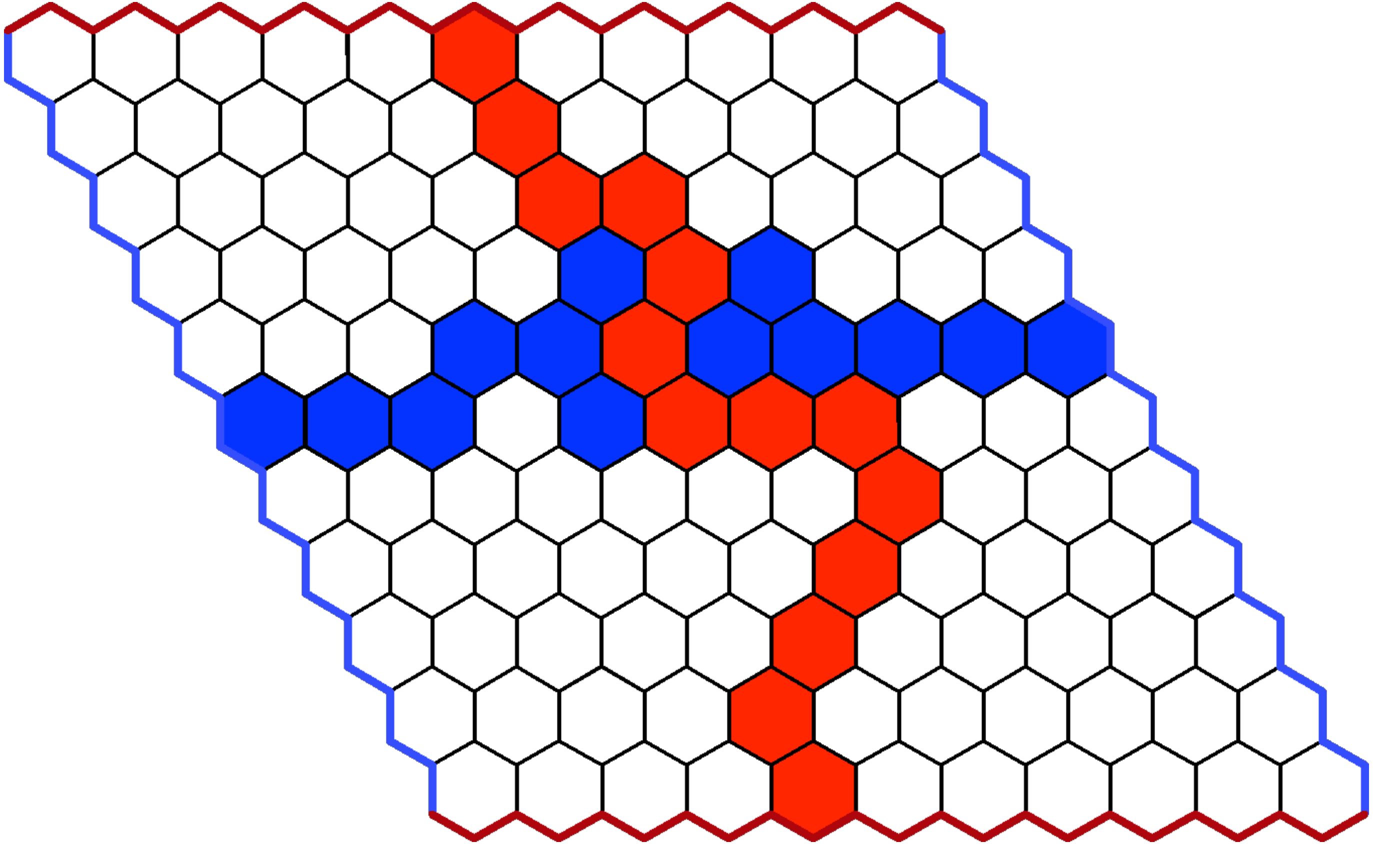
Monte-Carlo Tree Search for Poly-Y

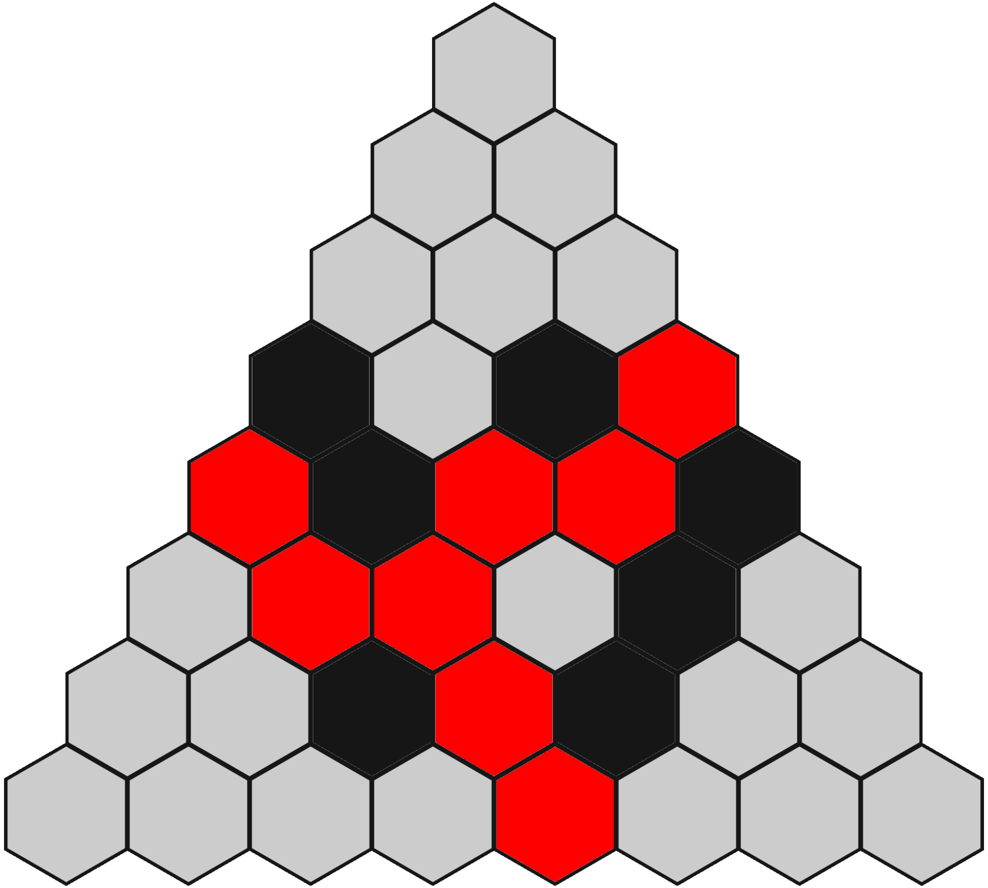
Lesley Wevers Steven te Brinke
University of Twente

BNAIC 2014

Hex

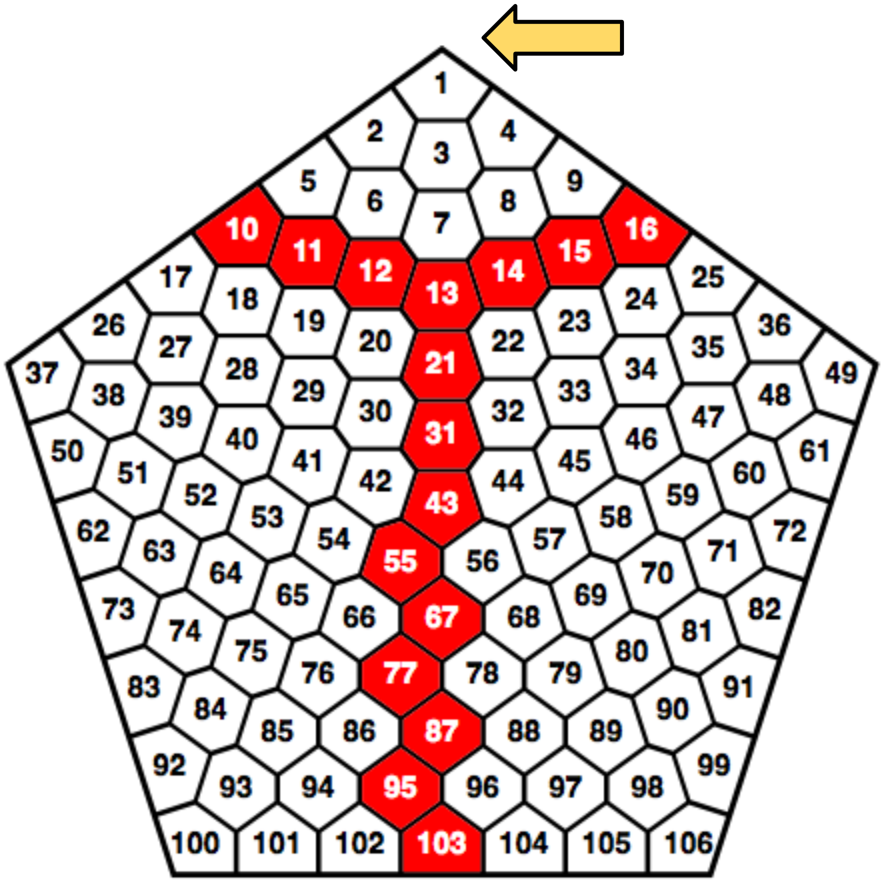


Y



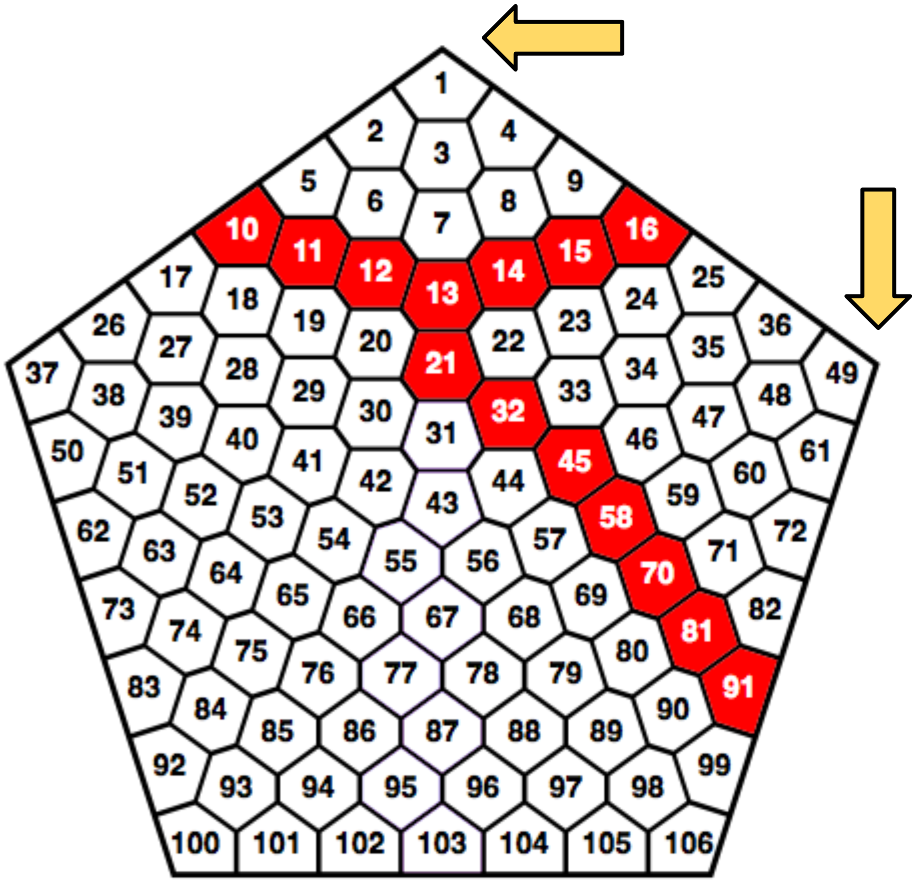
Poly-Y

- A Y-structure captures a corner.
- You win if you capture three corners.



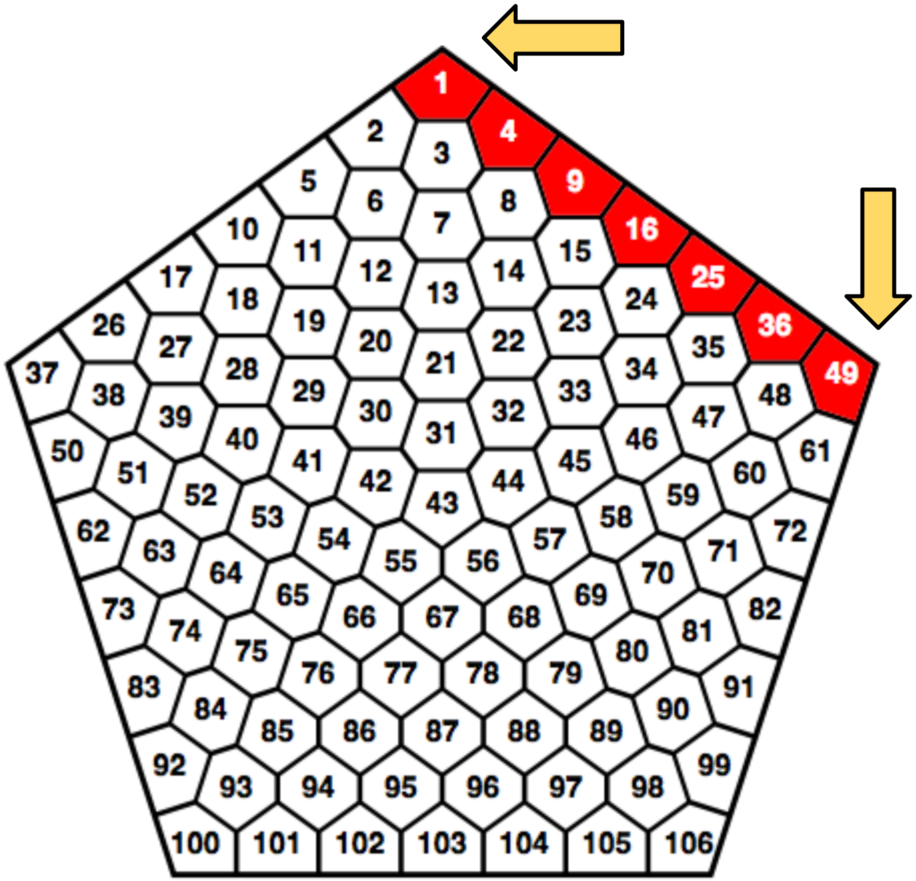
Poly-Y

- A Y-structure captures a corner.
- You win if you capture three corners.



Poly-Y

- A Y-structure captures a corner.
- You win if you capture three corners.

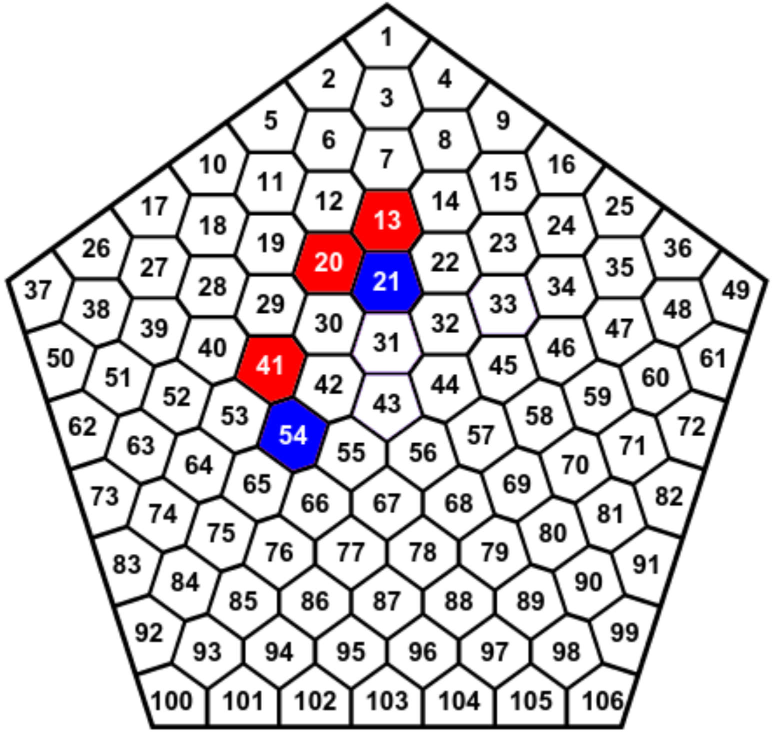


Poly-Y: Example game

Demo: http://archive.codecup.nl/2014/23/showgame_qga_e80349.html

Challenges

- Large branching factor
- Difficult to evaluate the strength of states
- Codecup: only 30 seconds per player



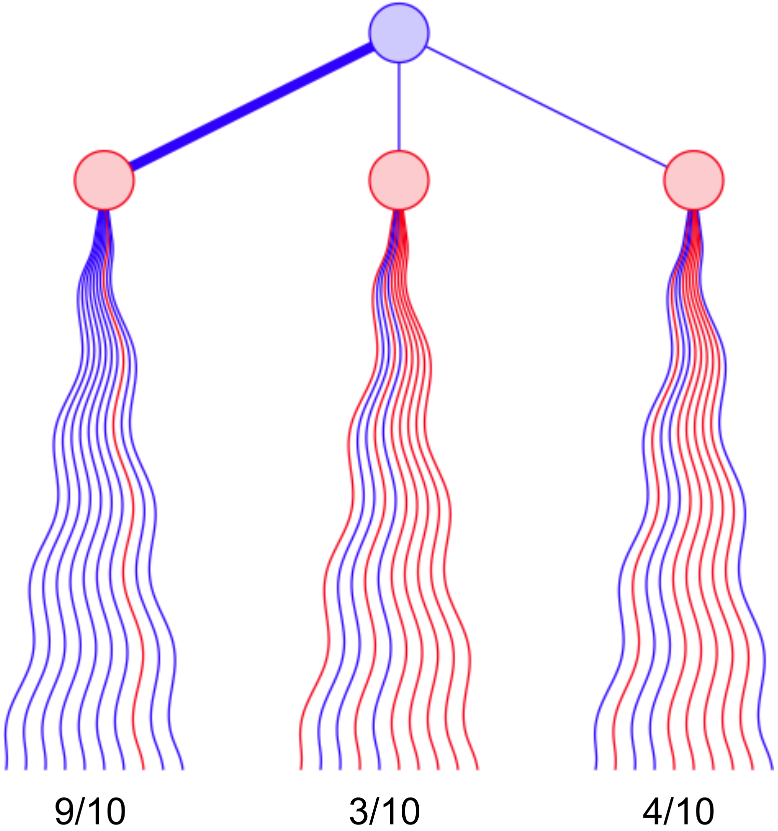
Approach

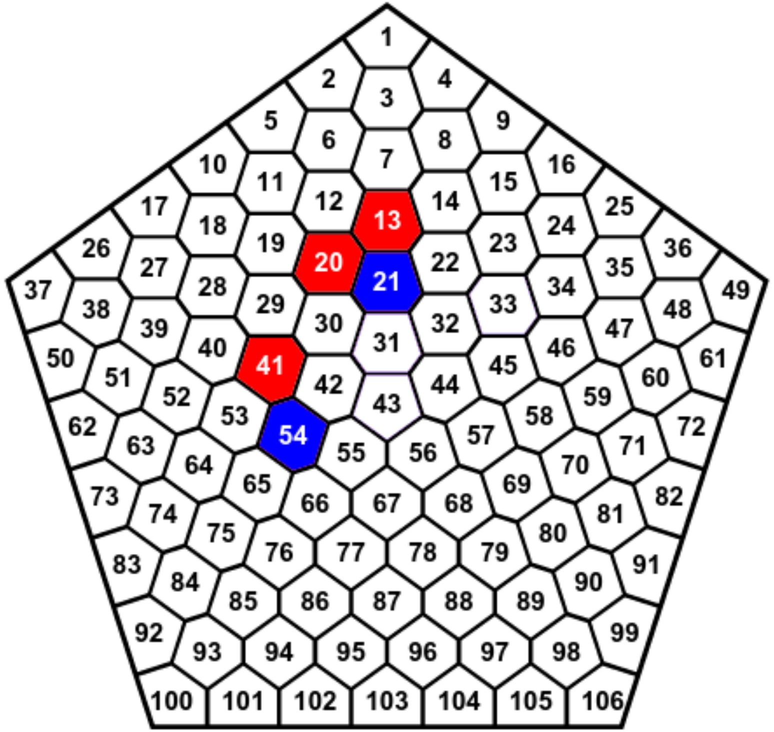
Monte-Carlo Tree Search has been successful in Hex.

Our approach:

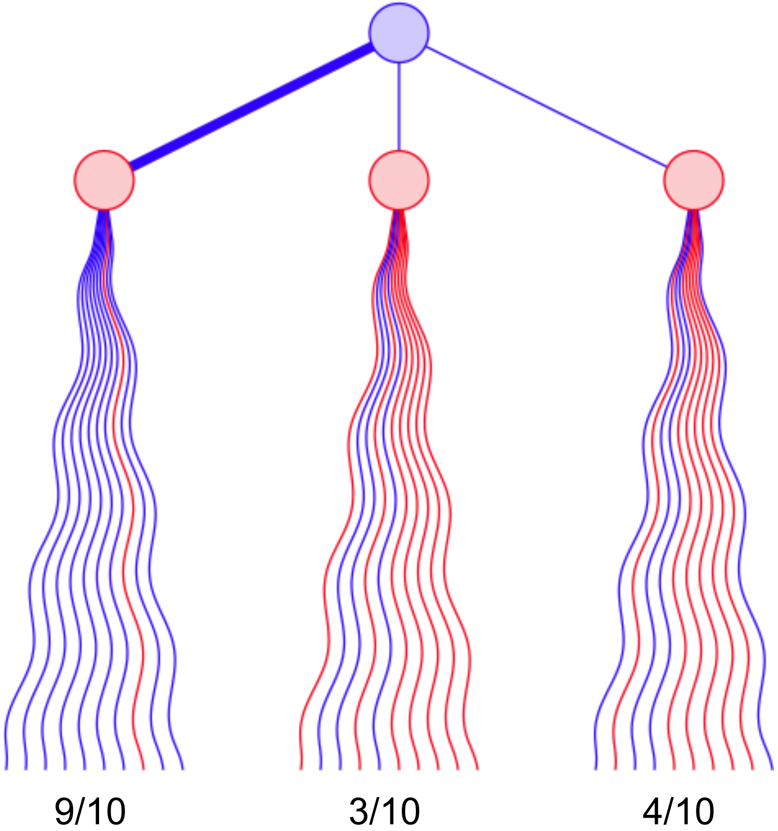
- Leverage MCTS from Hex to Poly-Y
- Develop heuristics for Poly-Y
- Construct an opening book for Poly-Y

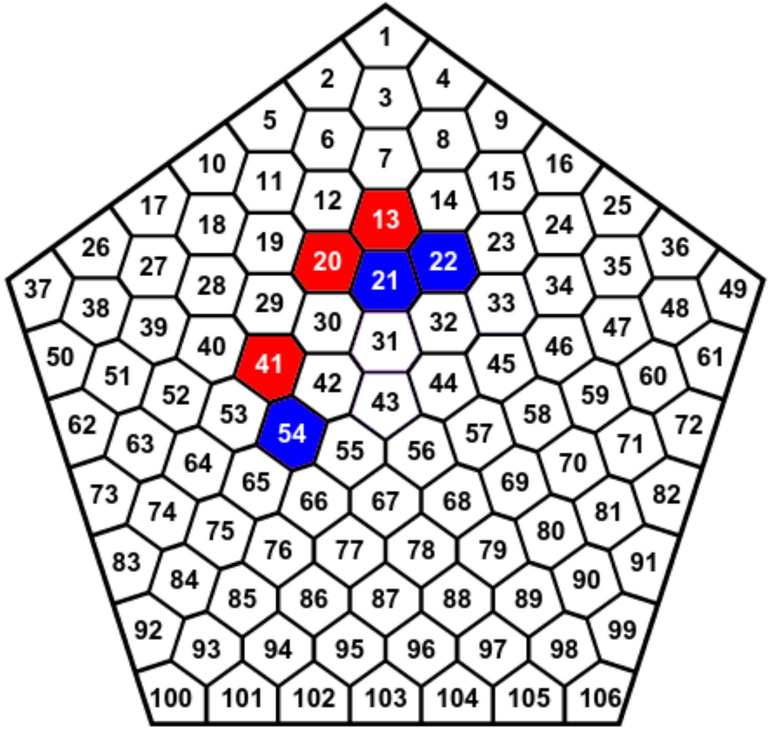
Background: Monte-Carlo Search



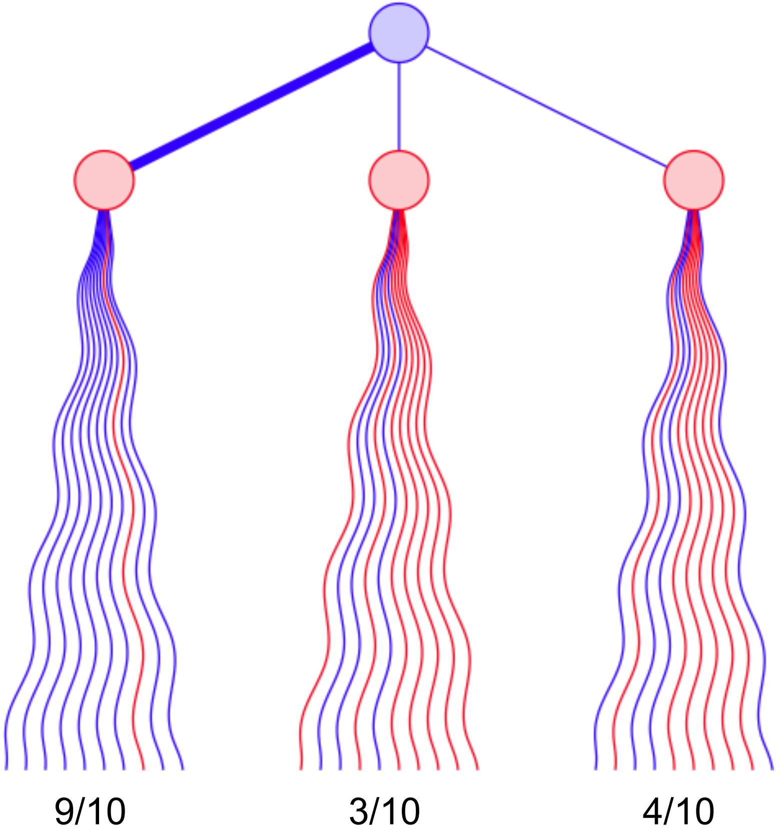


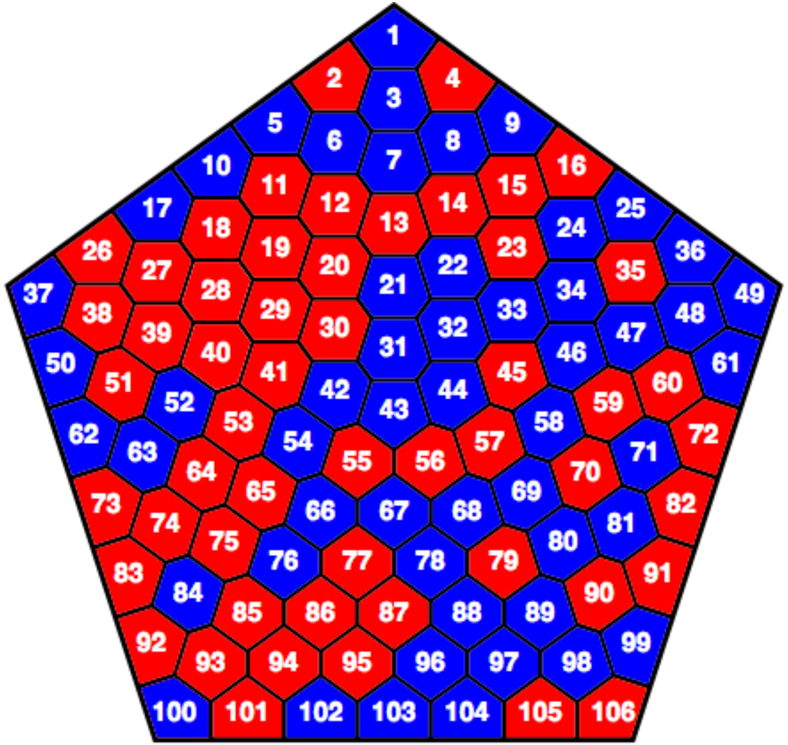
Background: Monte-Carlo Search





Background: Monte-Carlo Search





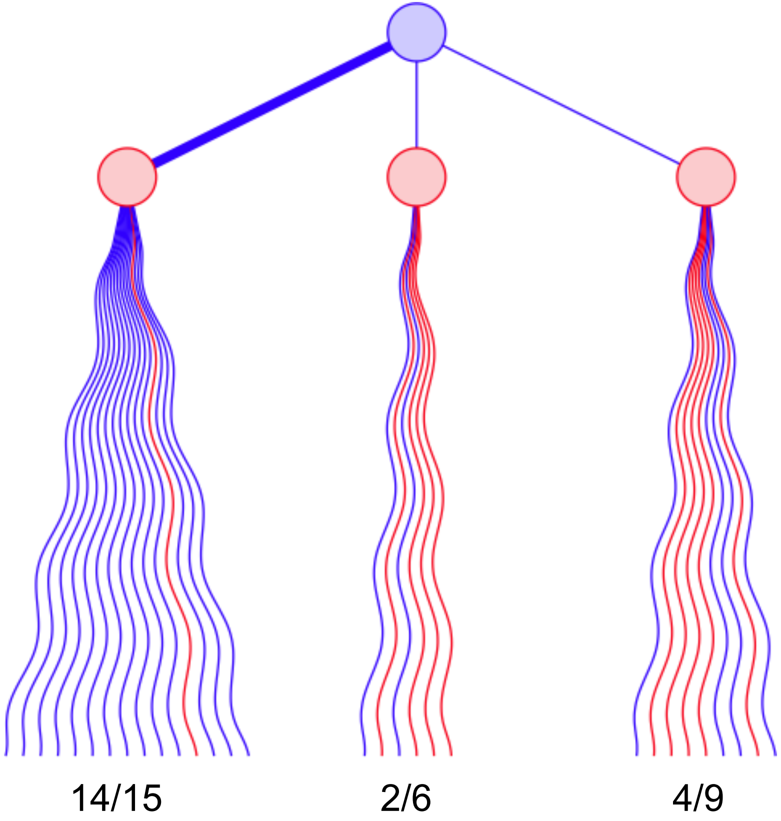
Background: Directed MC-Search

Find balance between:

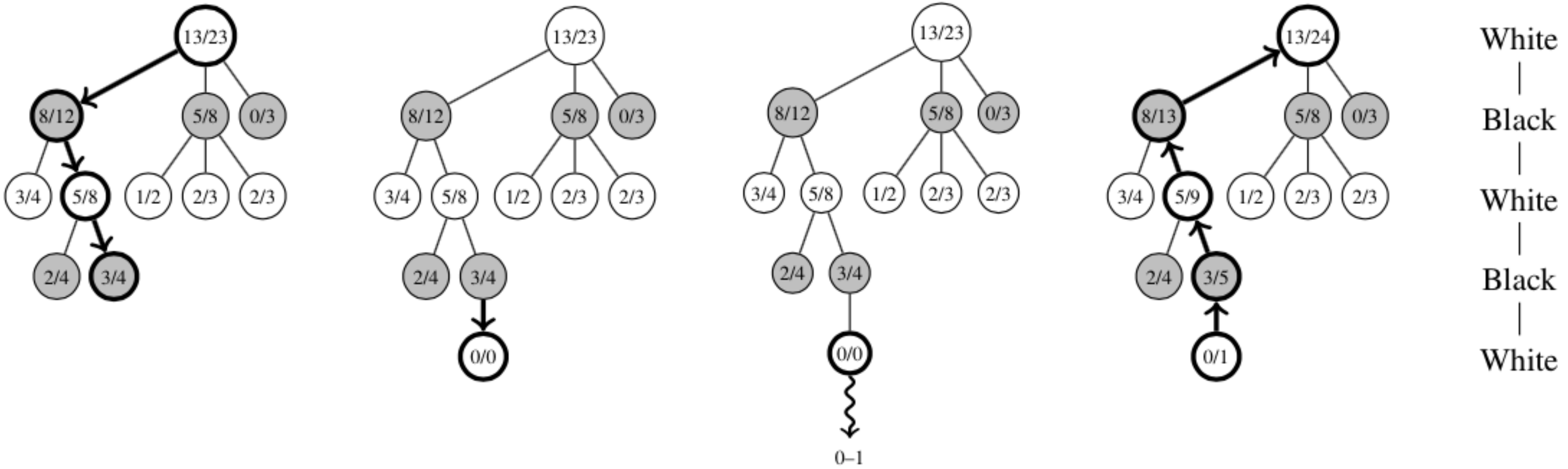
- **Exploration** of moves with few samples
- **Exploitation** of good moves

Multi-Armed Bandit Problem:

- UCT algorithm

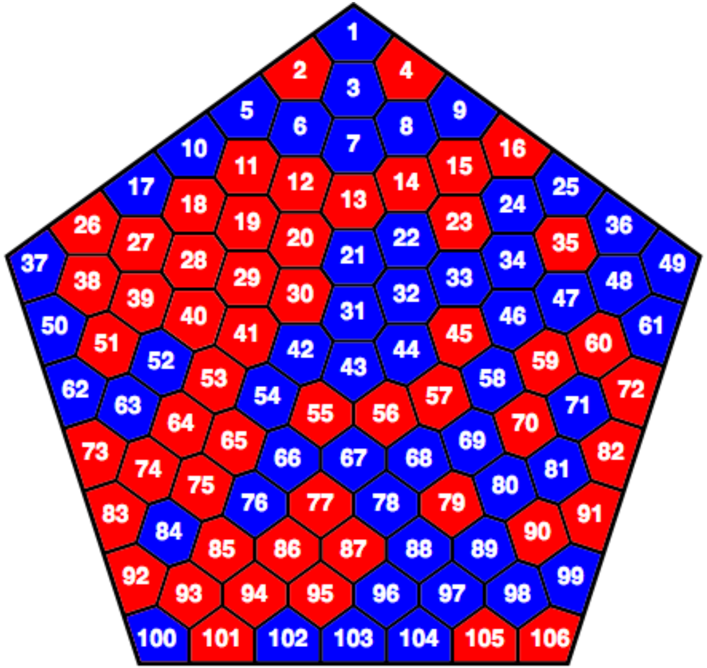


Background: MC Tree Search



(a) Selection (b) Expansion (c) Playout (d) Backpropagation Player

Background: All Moves as First

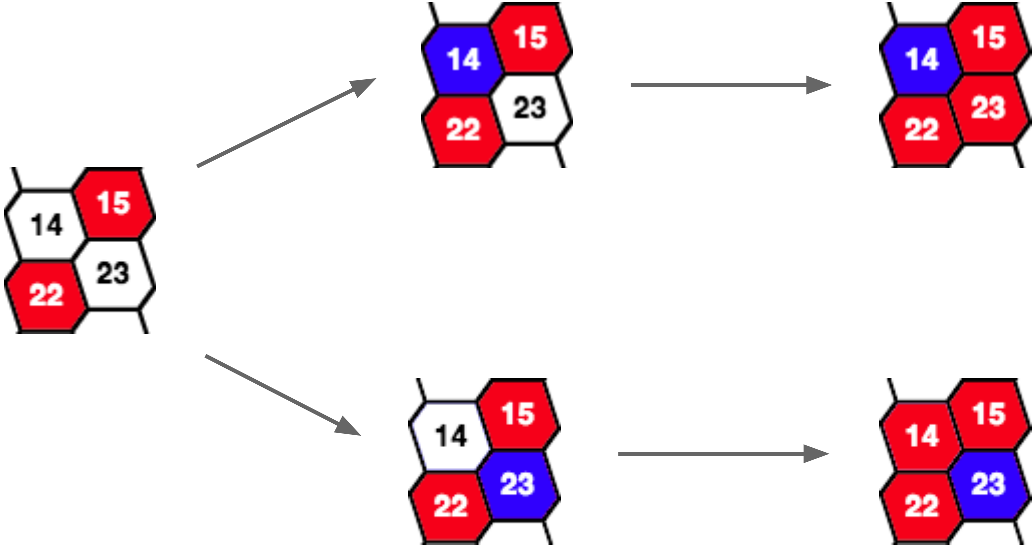


One random playout gives information about many states in the search tree.

	Win rate versus previous
MCTS UCT	-
+ AMAF	94.3% ± 0.48%

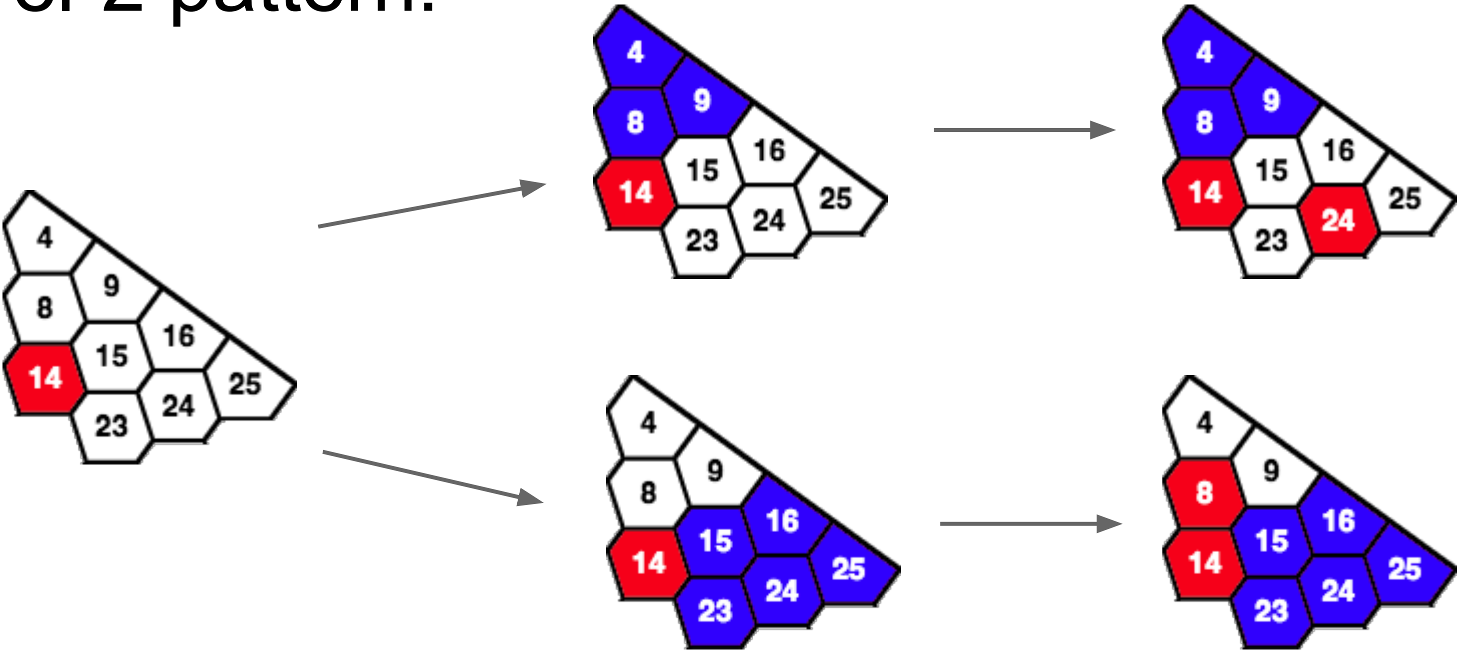
Background: Virtual Connections

Bridge pattern:



Background: Virtual Connections

Level-2 pattern:



Background: Playout Heuristics

How can we use virtual connections?

- Heuristically enforce simple virtual connections in random playouts.
- Idea: make playouts more realistic.

Issues:

- Searching for complex connections is expensive.
- Enforcing virtual connections can make MCTS weaker.

Playout Heuristics: Patterns

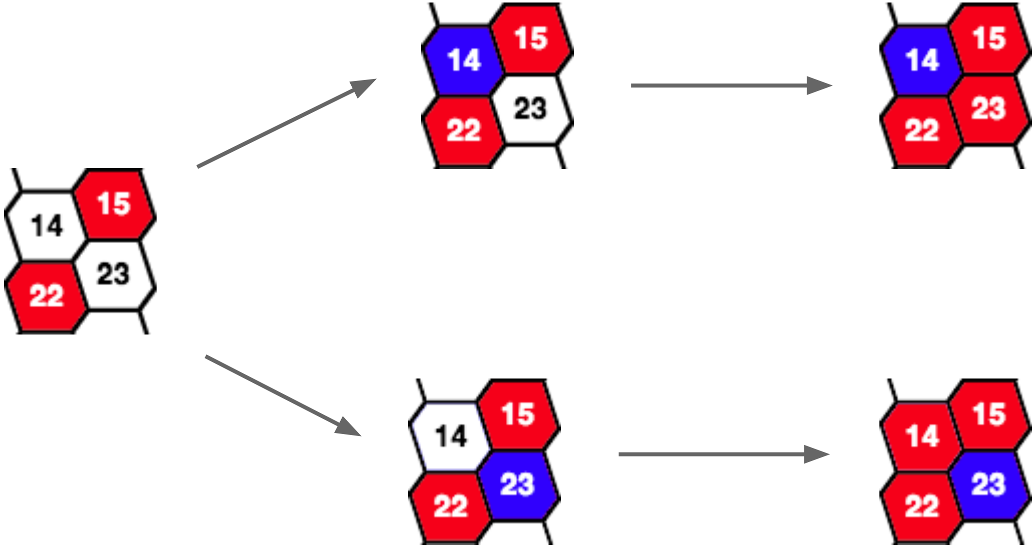
For every cell we define patterns as:

- Constraints on board (bitmasks)
- Move to perform if pattern matches

If multiple patterns match: pick a random one

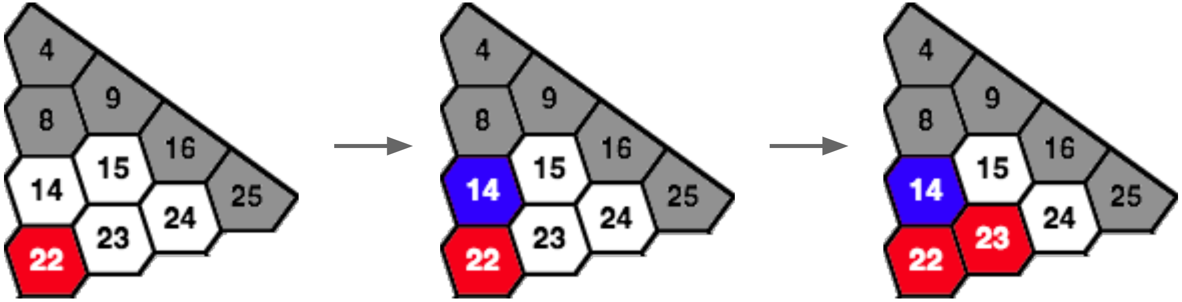
If no patterns match: do a random move

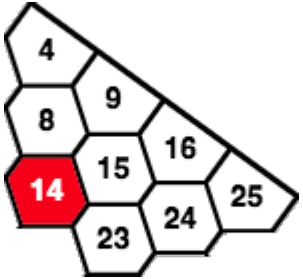
Playout Heuristics: Bridges



	Win rate versus previous
MCTS UCT	-
+ AMAF	94.3% ± 0.48%
+ Bridges	98.8% ± 0.16%

Playout Heuristics: Higher-level VC





	Win rate versus previous
MCTS UCT	-
+ AMAF	94.3% ± 0.48%
+ Bridges	98.8% ± 0.16%
+ Higher-level	70.4% ± 0.64%

Playout Heuristics: Fillboard

Don’t play at the edge early in the playouts.

- Playing at the edge is generally weak.
- This makes more patterns applicable.

	Win rate versus previous
MCTS UCT	-
+ AMAF	94.3% ± 0.48%
+ Bridges	98.8% ± 0.16%
+ Higher-level	70.4% ± 0.64%
+ Fillboard	58.7% ± 0.69%

Opening Book

Problem:

- MCTS is weak early in the game.
- Games can be decided in the first few moves.

Opening book:

- Move to states with a high win rate in self-play.

Opening Book

- Variable depth opening book: 2 to 4 moves deep
- Two weeks, 64-core machine

	Playing as white	Playing as black
No book vs. no book	24.4%	75.6%
Book vs. no book	65.1%	78.0%
Book vs. book	57.4%	42.6%

Conclusions

We won the CodeCup 2014 by:

- Leveraging techniques from MCTS Hex to Poly-Y
- Developing new playout patterns
- Developing an opening book

You can:

- Play against our player online:
  - https://maksverver.github.io/lynx/
- Download our source code
- Participate in the CodeCup 2015