Monte-Carlo Tree Search for Poly-Y

Lesley Wevers  Steven te Brinke
University of Twente

BNAIC 2014
Hex
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

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.

<table>
<thead>
<tr>
<th></th>
<th>Win rate versus previous</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCTS UCT</td>
<td>-</td>
</tr>
<tr>
<td>+ AMAF</td>
<td>94.3% ± 0.48%</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th></th>
<th>Win rate versus previous</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCTS UCT</td>
<td>-</td>
</tr>
<tr>
<td>+ AMAF</td>
<td>94.3% ± 0.48%</td>
</tr>
<tr>
<td>+ Bridges</td>
<td>98.8% ± 0.16%</td>
</tr>
</tbody>
</table>
Playout Heuristics: Higher-level VC

<table>
<thead>
<tr>
<th>Win rate versus previous</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCTS UCT</td>
</tr>
<tr>
<td>+ AMAF</td>
</tr>
<tr>
<td>+ Bridges</td>
</tr>
<tr>
<td>+ Higher-level</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th></th>
<th>Win rate versus previous</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCTS UCT</td>
<td>-</td>
</tr>
<tr>
<td>+ AMAF</td>
<td>94.3% ± 0.48%</td>
</tr>
<tr>
<td>+ Bridges</td>
<td>98.8% ± 0.16%</td>
</tr>
<tr>
<td>+ Higher-level</td>
<td>70.4% ± 0.64%</td>
</tr>
<tr>
<td>+ Fillboard</td>
<td>58.7% ± 0.69%</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th></th>
<th>Playing as white</th>
<th>Playing as black</th>
</tr>
</thead>
<tbody>
<tr>
<td>No book vs. no book</td>
<td>24.4%</td>
<td>75.6%</td>
</tr>
<tr>
<td>Book vs. no book</td>
<td>65.1%</td>
<td>78.0%</td>
</tr>
<tr>
<td>Book vs. book</td>
<td>57.4%</td>
<td>42.6%</td>
</tr>
</tbody>
</table>
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