General Board Game Playing for Education and Research in General AI Game Learning

Wolfgang Konen
TH Köln – University of Applied Sciences

Why? We have GGP!

GGP with GDL: learns an unknown game at run time.
This is a tough task and fantastic endeavor in logic reasoning!

But as a consequence there are also some limitations [Swiechowski2015] of GGP:
• Simulations in GDL are slow
• Cannot compare & compete with the best game-specific agents (e.g. Othello → Edax, Hex → Hexy)
• CI agents (TD and deep learning) are difficult to integrate (slow learning and combinatorial explosion)

GBG Advantages

• Easier for education: The complicated agents can be reused for new games
• First generic implementation of TD-n-tuple agents: Arbitrary games, arbitrary number of players
• GBG allows fast game simulation (10,000 – 90,000 moves per second) for CI agents
• Comparison with game-specific agents (strong or perfect players, e.g. Othello → Edax) is possible
• Generic inclusion of game symmetries
• Game-specific visualization and inspection → get deeper insights
• Human – agent play

Disadvantages of GBG:
• Does not allow new games at run time. But new games can be added at compile time.

What is GBG?

• Let various agents play on all board games
• Standardized interfaces: game states, agents, …

Table 1: Games in GBG (N: # player, D: deterministic, ND: non-deterministic)

<table>
<thead>
<tr>
<th>Game</th>
<th>N</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>TicTacToe</td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>Hex (scalable)</td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>2048</td>
<td>1</td>
<td>ND</td>
</tr>
<tr>
<td>Connect-4</td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>Nim (scalable)</td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>Othello</td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>Sim (Color blind)</td>
<td>2</td>
<td>D</td>
</tr>
<tr>
<td>Sim (Opts)</td>
<td>2,3</td>
<td>D</td>
</tr>
</tbody>
</table>

Table 2: Agents in GBG

<table>
<thead>
<tr>
<th>Agent</th>
<th>Game Description</th>
<th>Game</th>
<th>Moves/second during:</th>
<th>...game learning</th>
<th>...game play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expectimax-N</td>
<td>Max-N (Minimax) for N-player</td>
<td></td>
<td>2048</td>
<td>67.000</td>
<td>94.000</td>
</tr>
<tr>
<td>MCTS</td>
<td>Monte Carlo Tree Search</td>
<td></td>
<td>2048</td>
<td>66.000</td>
<td>9.000</td>
</tr>
<tr>
<td>MCTS-Expectimax</td>
<td>TD-n-tuple</td>
<td></td>
<td>2048</td>
<td>7.000</td>
<td>40.400</td>
</tr>
<tr>
<td>TD</td>
<td>Temporal Difference</td>
<td></td>
<td>Connect-4</td>
<td>7.000</td>
<td>40.400</td>
</tr>
<tr>
<td>Sarsa</td>
<td>Sarsa with n-tuple features</td>
<td></td>
<td>Sim (Beta)</td>
<td>7.000</td>
<td>5.100</td>
</tr>
</tbody>
</table>

GBG allows fast game simulation / game play
The table shows moves/second for various agents and games on a single core.

Conclusion

Educational Benefits

Benefits for students:
• Agents readily available
• First results within days or weeks
• Code better re-usable

Evaluation (questionnaire, 3 students):
• “Better than starting from scratch?” – Strong agree
• “Time to get familiar with GBG?” – 2 days (median)
• “Enough documentation?” – Strong agree

Their wishlist:
• More GUI-elements for game-related settings
• More “How-To”-cases in documentation

Some Results

Example Games

Expectimax-N

<table>
<thead>
<tr>
<th>Score: 1,073,800</th>
<th>left</th>
<th>up</th>
<th>right</th>
</tr>
</thead>
<tbody>
<tr>
<td>5215,2,416</td>
<td>5319,148</td>
<td>5319,481</td>
<td>5360,171</td>
</tr>
</tbody>
</table>

GDG benefits

Educational perspective:
• Much easier for students to use complex CI agents
• Standardization of code development
• Attracts students to fascinating area of game learning

Research perspective:
• Fast simulation of CI agents
• TD-n-tuple successful on diverse games: 2048, Connect-4, Hex, Othello, Nim
• Success against strong or perfect-playing agents (Connect-4, Hex7x7, Nim) → advantage over GGP
• Not yet: Othello, Hex above 7x7