A Comparative Study of Solvers for Amazons Endgames

Kloetzer Julien
Iida Hiroyuki
Bouzy Bruno

The game of the Amazons
- Created by Walter Zambikus (Argentina) in 1988
- Intermediate between Chess (movement) and Go (territory)
- In turn, each player moves an Amazon (like a queen in Chess), then shoots an arrow from the landing position in the same way; this square is blocked
- No further shot or move can pass over or land on a blocked square
- The first player unable to move loses the game
- The score of the game is the size of the territory of the winner, that is the number of moves he can still play after his opponent has passed

Amazons problems
- Unlike other more famous games (Chess, Othello, Shogi...), there is few knowledge about the game of Amazons. If we want a problem database, we will have to build it
- What is an Amazon problem? It is a portion of the board containing Amazons of both players as well as arrows, considered isolated from the rest of the board (edges of the position should be either edges of the board or arrows). Its solution is a move or a set of move maximizing the final territory of the winning player
- How to get these sub-positions? By extracting them from game records

Extraction procedure
- Read a game record until a sub-part of the board is isolated
- Check if this sub-part has a set of predetermined features; if yes, copy the sub-position and store it in the database

Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFPN</td>
<td>81.84</td>
</tr>
<tr>
<td>WPNS</td>
<td>87.74</td>
</tr>
<tr>
<td>Alpha/Beta</td>
<td>98.52</td>
</tr>
<tr>
<td>MCTS</td>
<td>87.51</td>
</tr>
</tbody>
</table>

Results analysis
Considering its “standard” status, the performances of DFPN are disappointing
The performance of WPNS are encouraging, but needs further study
Alpha/Beta clearly outperforms every other method, confirming the necessity of an evaluation function
MCTS performs surprisingly well, considering it is not at all designed to solve problems

Problem 1: Use 2-valued solvers for a multi-valued problem?
DFPN and WPNS are 2-valued solvers; how to adapt?
- Instead of answering to “Is this position winnable?”, we answer to “Can I win this position with at least N points?” (or the reverse question)
- How? With an initial result R to prove
  * Consider leaf nodes with better results than R as YES nodes, others as NO nodes
  * Do an iterative search on all possible results, starting from the worst

Problem 2: How to evaluate Game-playing methods?
Game-playing methods just give a move, not the exact value of a position; how to compute the value?
- Get a move from the method; then, use a judge program (e.g. real or solver) to compute the exact value of the position given by playing the move in the initial position.
  * If the computed value is equal to the value of the initial position, the move is optimal
  * If it is inferior, the move has failed by <difference between the two evaluations> points

Problem 3: The solving method’s procedure needs more time
Set the limit of time for the game-playing methods to find their move-answer, and give that same period of time to each iteration of the solving method process

Problem 4: Both classes of methods are not evaluated in the same way
Use the same judge program to evaluate both game-playing methods and solving methods

Results

Percentage of problems solved for each assessed methods

Statistics on unsolved problems

% DFPN % WPNS % Mc % AB

Problems extracted
- Game record: auto-play from our program (Camyra) with 3 minutes per side
- Problems that were Too small or too simple were discarded
- ~900 problems generated