Evolutionary Learning of Policies for MCTS Simulations

James Pettit, David Helmbold

University of California, Santa Cruz
jpettit@soe.ucsc.edu

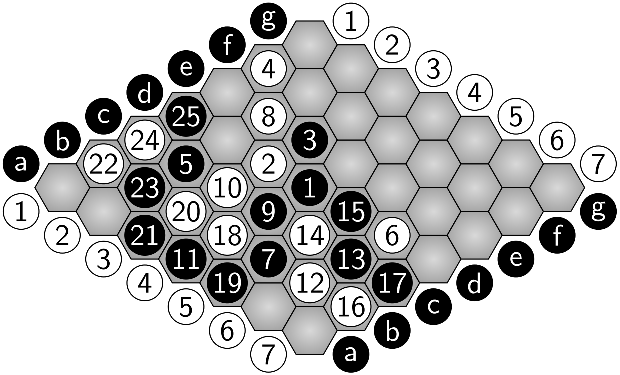
July 2012

Overview

The Game of Hex

- 2 player, perfect information
- 6-sided hexagons on a parallelogram board
- Common sizes: 11, 13

The Game of Hex - Example Board



The Game of Hex - Good for AI

- Easy to program
- Clear-cut winning condition
- Large problem space
- Solved for boards up to 7x7

Tree Search

- Game tree grows exponentially
- Symmetry can halve space
- Limited opportunities for provable pruning
- No good position ranking heuristic

Monte Carlo Tree Search

- Use random playouts to estimate minimax value
- Large enough playouts will converge to true minimax value
- Seems dumb, actually works
- Computationally very expensive

Monte Carlo Tree Search - Playout Policy

- Naively “improving” the strength of the playout policy can hurt overall performance
- Requires careful and expensive testing to verify improvement

Evolutionary Learning

- Idea: Evolve playout policy
- Individual policies compete in a tournament to reproduce
- Self-play yield a self-bootstrapping system

Encoding

- Individual policy is a collection of evolvable weights
- One weight, many interpretations
- Explicitly requires domain knowledge
- Implicitly limits the system

Results

- Strong and robust

Future Work

- Different encodings
- General game play
- Encode expert knowledge in weights