Evolutionary Learning of Policies for MCTS Simulations

James Pettit, David Helmbold

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

July 2012

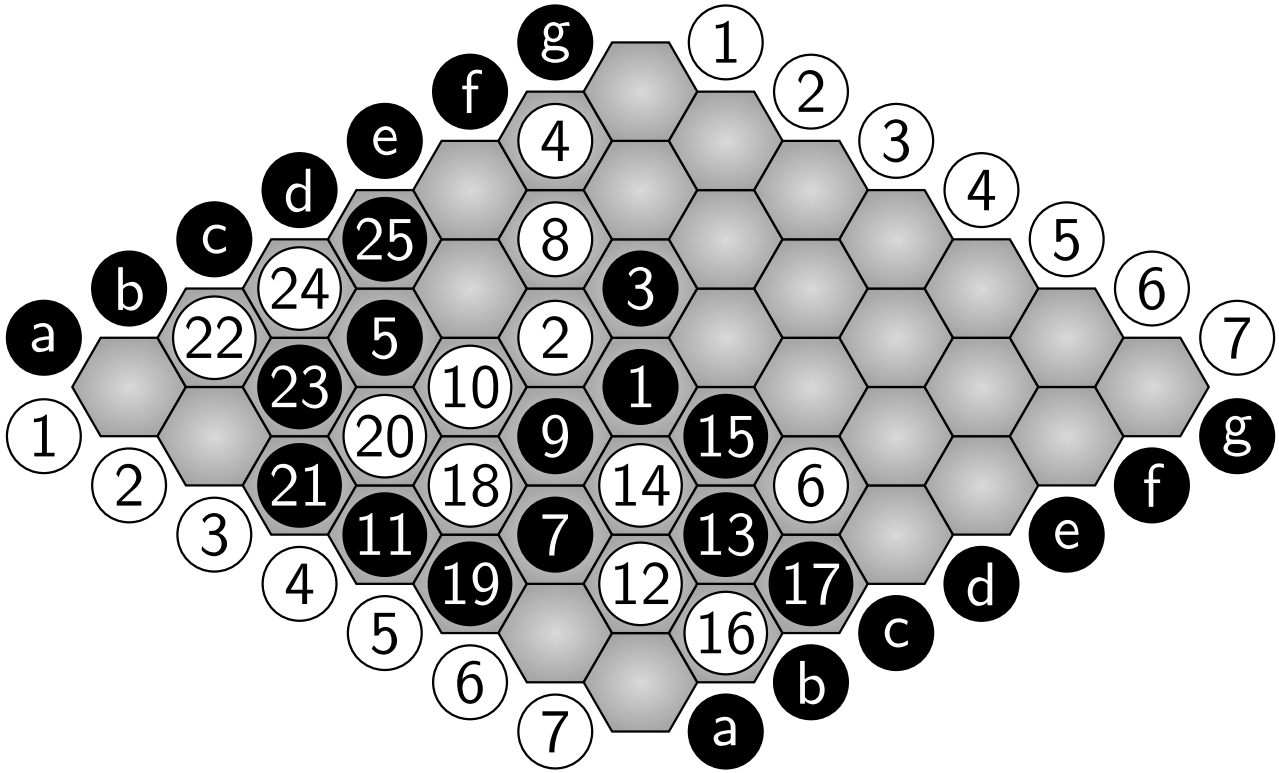
Overview

1. The Game of Hex
2. Monte-Carlo Tree Search (MCTS)
3. Apply Evolutionary Learningto MCTS+Hex
4. Results and Future Work

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 (Hivemind)

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

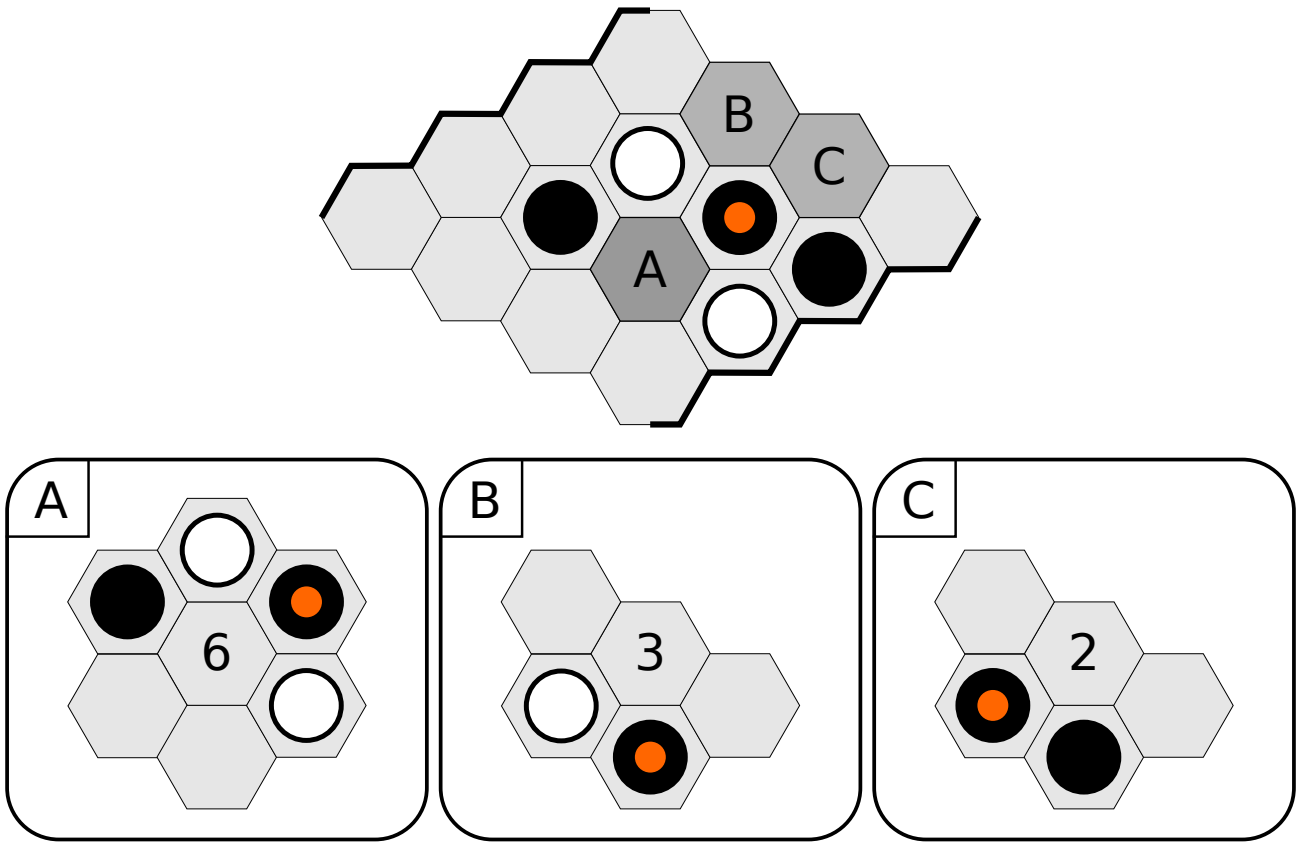
Evolution

1. Generate n individuals (population)
2. Evaluate fitness
3. Rank by fitness
4. Select top c children, $c < n$
5. Recombine children into n new individuals (next generation)
6. Mutate population
7. Repeat from 2

Encoding

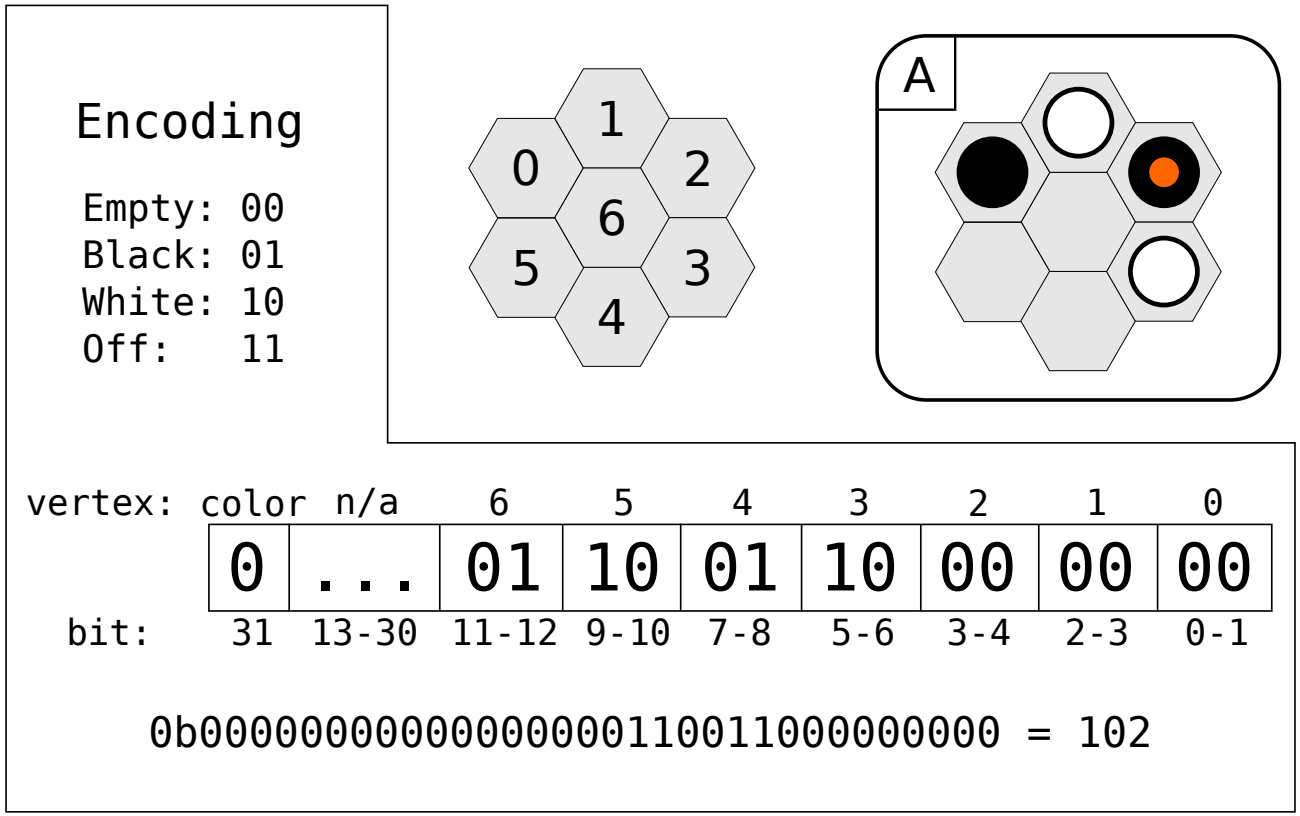
- Individual policy is a collection of evolvable weights
- One weight, many interpretations
- Explicitly requires domain knowledge
- Implicitly limits the system

Encoding - Example



- Weight moves local to last-played move
- If local area is filled, use default policy

Encoding - Example



- Map from 32-bit integer to floating point weight
- Evolution (mutation/recombination) operates on individual's map entries

Results (Self)

player	opponent		
	default	uniform	uniform (tenuki)
uniform	70.50%		
uniform (tenuki)	61.00%	50.00%	
learned	90.00%	84.00%	86.00%

All-play-all tournament of the 4 Hivemind variants.
Each element is the percent win-rate of the row variant versus the column variant.

Results (Self)

	default		uniform	local
	11x11	13x13	11x11	13x13
learned win %	92.5 %	94 %	88.5%	85%

Results (MoHex 7x7)

	Learned policy	Uniform local	Default policy
win %	42%	26%	11.75%

Results (MoHex 11x11)

Hivemind	Win-rate	Relative CPU
Default	0.5%	510%
Local	0.0%	545%
Learned	11%	620%

Future Work

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