Dimitri Masetta 306130

Quixo Report

Notable features:

- Space of states = 3^{25} as an upper limit, some states are unreachable.
- Space of actions = 44, max limit reachable only on the first move.
- Symmetry = 4 flips (horizontal, vertical, diagonal left, diagonal right) and 4 rotations.

Minmax agent:

The defined minmax agent explores all the solution tree by recursively evaluating each possible move until a defined depth. Alpha beta pruning was implemented to improve performance in terms of time.

As expected, the results are great for this kind of game with almost 100% of won matches against a random player. The only drawback is the execution time ~ 3s per games.

Genetic programming agent:

A genetic programming algorithm was developed, capable of evolve individuals that have a graph as their genome. The nodes of the graph consist of a function set and a terminal set inspired by [**Competitive Environments Evolve Better Solutions for Complex Tasks,** Peter J. Angeline and Jordan B. Pollack].

The recombination function can be:

- One node crossover
- One node mutation
- Reproduction.

Various mechanisms for selection and evaluation of individuals in the population have been developed:

Fitnessless coevolution, inspired by [**Fitnessless Coevolution,** Wojciech Ja«skowski, Krzysztof Krawiec, Bartosz Wieloch]

- Fitness evaluation against a random players or pre-trained ones followed by roulette fitness selection or tournament fitness selection.

After many tests on all the major configuration, the one leading to the best individuals is as follows:

- Population size = 50
- Selection size = 25
- Crossover probability = 0.2 (despite the usual 0.8, no particular bloating detected)
- Reproduction probability = 1 - Crossover probability
- Mutation probability = 0.4 (despite the usual 0.1)
- Round against random = 30
- Fitness evaluation without coevolution against random
- Fitness selection roulette wheel

With the above parameters with 100 generations, we arrive at agents that win on average 91% of the time against a random agent, with a match time of only ~2ms.

Try it for yourself:

To try the agents and see the results described above try the notebook *play_the_game.ipynb*