Euchre AI

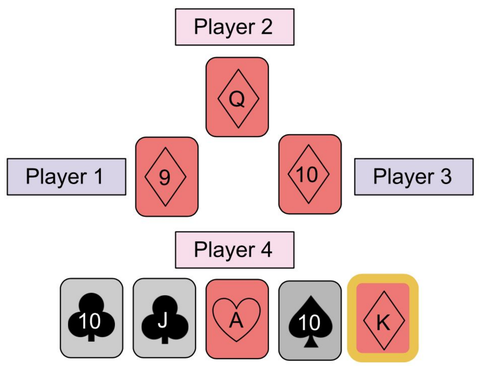
Computer Science Senior Capstone Project
August Nord aanord20@earlham.edu
Earlham College

Intro

Games are often studied in CS as they handle complex problems on a manageable scale. Euchre likewise is a card game with aspects that make it interesting to study:

- Imperfect Information
  - The complete game state is hidden from players
  - As opposed to perfect information games like Chess
- Partner Play
  - It is advantageous to let your partner win sometimes
- The Calling Phase
  - A single decision made first in the round that heavily influences the results of the round

The Game

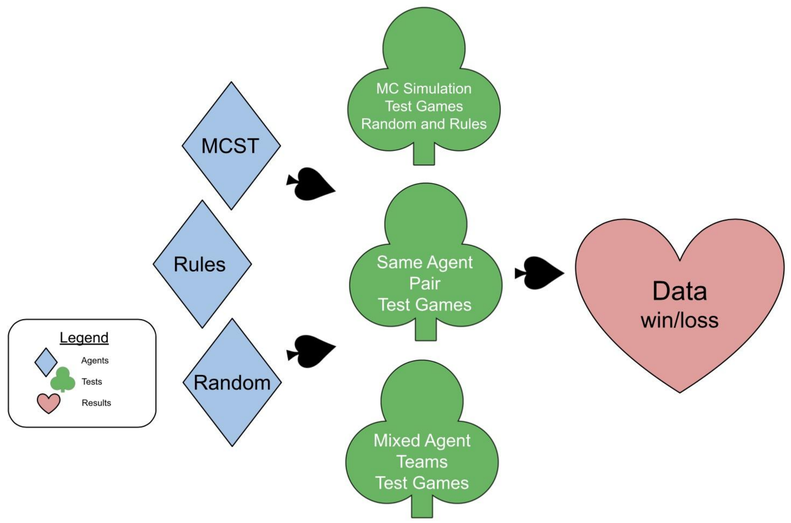


*Player 4 must follow suit and wins the trick with the King

Each game is made of several hands divided into five tricks

- Partners earn points together
  - Players 1 and 3, Players 2 and 4
- Play proceeds clockwise
- Players play the same suit that was led
  - If unable to follow suit, a player can throw a card or ‘trump in’ with a high-ranking card of that hand’s trump suit
- The highest ranking card wins the trick

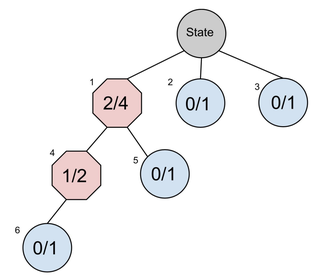
Data Architecture Diagram



Methods

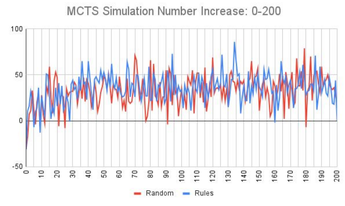
Our work focused on three types of agents:

- Random
  - Selects a playable card at random
- Rules
  - Plays using some basic Euchre strategies
- Monte Carlo Search Tree (MCST)
  - Plays simulation games from each decision point and updates probability of winning accordingly

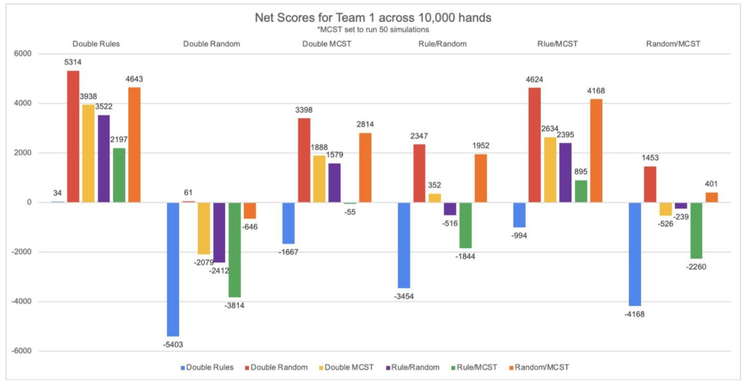


Above: A Monte Carlo Search Tree after 6 simulations. The red nodes are those which, when explored, led to a win. Blue node simulations were losses. Note that the wins and losses of the child nodes affect the parent node’s win probability.

Below: Tests playing Double MCTS against Double Random on 100 hands with increasing simulation number. Tested



Results



Pairwise tests of Team 1 (top) against Team 2 (bottom/color)

- Double Rules consistently performs best
- Double Random consistently performs worst
- Double MCST does fairly well

Future Work

Ideas for future work include:

- More testing on MCST Agent
  - Experiment with exploration weight
- Develop Deep Q-Learning Agent
  - Train an agent with a neural net
- Implement special game cases
  - Add functionality for “going alone”
- Create an interactive interface
  - Develop an interface that lets the human user play against these computer agents

Acknowledgements

I would like to thank David Barbella, Micah Nord, and Charlie Peck for their support and contributions to this project.