Senior Project Proposal (CPSC 490)

David Liu
Advisor: Dr. James Glenn

Quoridor is an abstract strategy board game which was invented in 1997 by gigamic and previously held the honor of Mensa’s Game of The Year title in the USA, France, Canada, and Belgium. This game involves a 9 by 9 grid-styled board, and the objective of both players is to get his walker piece to the opposite side of the board. The grid lines on the board are indented such that a wall can be placed. Walls have a length of 2 squares, and each player only has 10 walls. Walls can be placed anywhere as long as it’s spanning two squares, and as long as both players have a path to victory. This is a turn based game and during one’s turn, the player could either move his walker piece into an edge adjacent square, or place a wall. The first player to reach the opposite end wins.
For my senior project, I plan to build a computer player of Quoridor that will be evaluated by the metrics specified below. Quoridor has a complexity that’s similar to that of Chess -- it has an average branching factor of 60, a game-tree complexity of 162 (log 10), and a state-space complexity of 42 (log 10).

Few people have attempted to build a strong computer player for Quoridor due to the relative newness of the game, and none have been able to build a computer player strong enough to beat a median Quoridor human player. Mertens and Glendenning are two individuals who have done research involving constructing a strong Quoridor computer agent, and they explored approaches in the topics of evolutionary computation, 1-layer neural network, and various search methods using different combinations of features. I will aim to incorporate their research into my strategizing on what kind of system could potentially lead to a strong computer Quoridor player.

**My Approach**

Preliminarily, my plan of attack is to first explore baseline search methods (e.g. alpha beta search) to gain a stronger understanding of the deficiencies in that elementary approach and figure out how to approach those weaknesses. While doing that, I aim to come up with better features than Mertens and Glendenning devised. Once I have a few search strategies explored and tested, I’d like to explore a neural network strategy. Since my background in these subjects is nearly nonexistent, I’m unable to elaborate here on the specifics of these search and neural network approaches that I’ll attempt; however, in consultation with my computer science classmates and Dr. Glenn, I know that taking these approaches will at least prove to be interesting and is a good direction to start.

**Likely Challenges**

Due to the complex strategies on the macro-level of this game that’s similar to Go, it’s likely that non-neural network approaches will face the same challenges that they faced when processing Go. To address this challenge, I plan on using reduced sized boards to reduce the complexity both for non-neural network approaches and neural network approaches. Although this changes the game non-trivially, exploration of the reduced game could yield insights that may lead to a better understanding of the game, potentially allowing for improved performance of various strategies on the full board.

It will also be difficult to find good features or heuristics due to the limited amount of literature and analysis on this game. Whereas other games have books and books of common strategies or ways of thinking, Quoridor is largely a blank slate and discovering these strategies may take a lot of game play on my part.

Furthermore, another challenge is that I might not have enough compute time or compute resources to truly gauge whether the neural network approach is a good one. It may be that the designed neural network system is a good one that has the potential to play well, but I may not have enough time to have it play against itself enough times or be properly trained. I hope that the reduced board size tests will partially alleviate this issue.

**Metrics**

I aim to compare my computer player against the existing computer players for Quoridor that are available online. Mertens’ computer player seems to be reproducible, but Glendenning’s will likely not be.
There are also one or two other computer players online that I have yet to investigate. One quantifiable metric that I’ll look at is how long (in terms of turns) the computer player survives against a human player. My experience playing this game suggests that games are much faster when there’s a huge level discrepancy between players; however, more investigation of this metric is needed. Another straightforward metric is how far the losing player is from victory at the end of the game. This metric can be problematic in that players could make bets with one or two moves that can swing the distance to victory by a lot. For instance, one wall placement can increase a player’s path to victory tremendously and the players can be betting on whether that wall will be formed. As a result, a good player that’s a close match to the other player can still ending up with a long path to victory at the end. Furthermore, the question of whether this computer player will be able to beat the median human player will be an evaluation point. It’s very unlikely that the computer player will be able to beat the median human player, but that is nevertheless something that will be kept in mind. Given enough time at the end, I’d also like to have a web application version of my Quoridor player online so that others can play against it, and I can gather data on how my computer player fares against other humans of varying levels.

**Deliverables**

- A Quoridor application that allows for two human players to play against each other, and for a human to play against my computer player.
  - Two human players version. *Goal: March 15th*
  - Human vs Basic Computer Player. *Goal: March 25th*
  - Human vs Stronger Computer Player. *Goal: May 4th*
- A report on the method(s) used in trying to create a strong Quoridor player. This will also include an assessment of how the computer program does with respect to the metrics specified. *Goal: May 4th.*
- Test results data. *Goal: May 4th.*