Go - Project

Jude Haris

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — May 2019

Chapter 1

Introduction

The ancient game of Go is regarded as the oldest board game still being played in the modern day. The origins of Go are not completely known but it is said to be invented in China 3000-4000 years ago. The game is formed by very simple rules that is played out most commonly in a 19x19 board. Each player takes turns placing a black or white stone on the intersections of the grid lines on the board. The board size is very crucial for the project. Due to increased board size compared to most western board games, a computer will take significantly more time to process through each possible move in every turn of the game to determine the final outcome. On the other side the basic principles of the game are much simpler to program and execute in software. Some complicated rules such as self-capture, and ko rule do exist. While implementing these rules can be simple,the effect of them being implemented correctly is great. For instance without implementing the ko rule it would be impossible to make progress in the game search tree in certain situations where moves could be played so that a never ending battle for the same point on the board takes place. Brute force methods to solve board positions would get stuck in these situation and loop infinitely.

The simplicity of the game's rules are deceiving as the game's strategy and tactics used are very complex. The possible amount of different games of Go to occur is inconceivably large. It has been said to learn the game it takes little time but to become an expert or remotely good at it takes years maybe even decades. To allow a computer to calculate the perfect move via brute force would take too long to be considered viable. The current best AI built to play Go is AlphaGo, it has defeated one of the world's best Go player Kie Jie in 2017 at Future of Go Summit. AlphaGo uses Monte Carlo Tree Search used with neural networks to evaluate the board and calculate the best move. With that in mind AlphaGo is yet to be anywhere near perfectly solving Go due to the enormous complexity and possibility of moves.

The aim of this project was to create a AI to play against users in Go, which can choose the correct move given a board position. Specifically, the project's aim was to create a program which given a Go life and death problem it would solve it, if solvable or return as unsolvable. First milestone was to achieve perfect AI which would choose the correct move given infinite time or smaller problem with less valid moves. The goal was to use brute force search method to look through every possible board state that could occur given the current state and choose the move which will lead to victory for the computer if there is any. The final goal of the project was to alter and enhance the AI which was achieved through the first milestone to be able to solve larger problems in the same time scale. And to do so, use heuristics to determine how favourable a board position is after searching for certain depth into the possible line of choices and then picking the line which leads to the most favourable board position. This final aim was set so that more realistic problems could be solved using the final product as without larger problems would simply take too long to play through. The motivation behind the project was to create a tool for Go players to use to improve tactical skill in Go via an interactive problem solver which helps the user figure out the solutions to life and death problems by having the computer play against them.