CS165A MP2 Report

Lan Liu CSIL username: liulan19920228

Python version: 2.7.11

To run the code: **./Gobang –n size –l**

The command line argument is optional: the default value of size will be 11, and the default setting will be human is dark player, and dark player moves first.

Architecture:

1. My code consists of board.py, players.py, Gobang.py. There are two classes,

class Gameboard (defined in board.py)

class Human_Player and class AI_Player (defined in players.py)

2. The class Gameboard has the following functionality:

It can set stones and remove stones on the board with specified color, track the position which will make the player to win, and also generate the set of next moves based on current state of the board. It can also display the board based on the standard output requirement. In detail, as follows:

def is_cell_legal(self, pos)   Tell if a given position is empty or not, and if the position is out of range.
def othercolor(self, val) For an input color, return the other color. White->Black, Black->White
def index(self,row,col) For a given row and column, return the position, which is a 1D array.
def marker_color For white stone, put value -1, and for black stone, put value 1.
def get_cell(self, pos) Return the value of a position, 0—empty, 1—black, -1—white
def do_move(self,pos, val) Place a new stone to the board on position with color 'val'.
def undo_move(self, pos) Remove the stone on position
def move_set(self) The possible moves can be made legally, and in order to save memory and running time at the minimax process, I restrict the move_set to be adjacent with the current stones.
def win_status(self, pos, val) If put a stone with 'val' on the given position, will win the game or not?
def win_step(self,val) The first position which makes the player wins.
def _*str*_(self) print and display the game board using basic ASCII art.

3. The class Human_Player and AI_Player

In the class Human_Player, it can read the move input from keyboard, and tell if the given move is legal or not, if so, it will play that move on the board.

In the class AI_player, I defined several patterns, and a function to count the number of given patterns in the board, so that get an evaluation function based on the assigned weight for different patterns. I also implement alpha beta pruning and minimax algorithm in the code, with cut off depth to be 2.

Search:

I found the best move by the following order:

First, I test if there is any location will make Ai wins or Human wins, if so, then I will set stone to that position, since by this step, either Ai will win or this is a necessary defensive step. Then I used minimax algorithm and applied alpha beta pruning with cut off depth 2. The heuristic function is the

evaluation function (AI) –defensive_weight* (evaluation(human))

I make defensive_weight to be a little bit larger than 1 (1.1).

The evaluation function is sum(weight * pattern), based on the following form (let 1 denote the stone)

Pattern	Weight
Two connected stone 01100 or 00110	10
Three connected stone 01110	1000
Four connected stone 11101 11011 10111	1000
Five connected stone 11111 011110	10000
Two two connected	50

• **Challenges:**
I am not a CS major and have no much experience on coding, this MP2 is much more demanding in coding skill than MP1, I took a super long time to make my code to run properly, this is the most challenging part personally.

The next challenge is to balance the pattern types and running time. If I considered more pattern types to modify the heuristic function, then the computational time will be increasing a lot and might exceed the given time. At the end, I optimize it that when I have Two connected pattern more than 1 then I will assign some extra weight, and if there are more than 1 three connected pattern, then it must lead to a winning, so the weight assigned to it should be win weight. But still, it hard to say what weights will be the best to set up.

• **Weaknesses:**
The weaknesses in my code is that it will run a lot slower if I deepen the level in the search process, so I can only try the cut off level as 2, if the cut off level is larger, than the move generated should be better. And also, I am very wondering how to make the code runs faster, the given executable code always runs faster than me.