

# 電腦對局理論 Homework 2 Report

B01902032 江東峻

Policy:

## 1. UCT

$UCB = \text{win rate} + C * \sqrt{\log(N0) / \text{Visit number}};$

$N0 = \text{parent's Visit number}$

$C = 0.1126$

## 2. progressive pruning

$rd = 1.5$  犧牲一點時間，增加AI的強度

$VAR\_EQUAL = 0.5$  根據投影片，0.2到0.5的score差距不大，速度卻差1秒，所以選擇0.5應該是划算的。

因為simulation次數較少，基本上砍不到東西。

## 3. move type

將move分成 normal, eat, connect, attack, boundary等幾種類型，將UCB乘上系數作為一個node的score，在expansion時，選score最高的。

$\{1, 1, 1, 1, 0.7, 1, 1, 0.9, 0.6\};$

=Normal, eat, conn, attack, bound, eat&conn, conn&att, att&bound, lib

eat = 可以吃別人棋子的move

conn = 當周圍有敵人棋子且自己有兩顆(連)，或敵人有兩顆(斷)

attack = move的周圍8個點有敵人的棋子

bound = 在邊界的棋子

lib = 放了只有一氣的棋子，放了也會被敵人馬上吃掉，所以係數要低

用一個tell\_move判斷是哪種類型，順序是：

$\text{eat} = \text{conn} > \text{attack} > \text{bound} > \text{normal}$

其中bound的係數很低，所以bound幾乎只有在有作用的時候才會判斷要下

## 4. repeat number

同move type，根據不同的move type有不同的simulation次數。

$\{30, 30, 30, 30, 30, 30, 30, 30, 30\};$

=Normal, eat, conn, attack, bound, eat&conn, conn&att, att&bound, lib

原本是要用來增加eat等的次數，但是根據UCB的算法，次數越多分數就越小，所以後來選用了比較平均的分佈。

## 6. check\_legal\_move

在simulation時，一開始空格很多，所以如果隨便猜一個位置，很容易就會是legal move，而不用去算全部的legal move後再隨機選一個。

在game\_length超過一定值後，就改成算全部的move在裡面挑，

方法是：找出全部的空白格，再用rand\_shuffle將順序打亂，然後一個個試(使用check\_legal\_move)

=>減少simulation一次的時間。

將NEXT\_BOARD改掉，用原本的Board，若有repeat\_move的話，將Board還原成GameRecord的重複盤面的上一個。

## 7. domain knowledge

前兩子會搶33的位置，而搶的順序是：沒有敵子>1敵子，若有兩個敵子的33則放棄。