

TCG HW2 Report

How to compile your code into an agent?

- `make shiro`
- 執行檔名：Shiro_agent

What algorithms and heuristics you've implemented.

- Monte-Carlo Tree Search with UCB score
 - Setting
 - $C = 1.18$ (模擬次數夠大，所以選擇paper推薦的參數)
 - 每個新長出的leaf所設定simulate的次數：`COUNT/pv_node->childs.size()` * `COUNT=3600` (leaf越少simulate越仔細)
- Progressive Pruning. `shiro_node::progressivePruning()`
- Some moving rules
 - 在Random simulate以及長樹時分別實作以下的rules

長樹選winRate / 走路選nonLoseRate
在可以吃子時要限制ruleTree生長
ruleTree要有逃脫選項
對方棋子很少的時候不能用rule (可能吃掉後對方會和我們平手)
不隨便吃掉在終點的棋子(當有棋子可以吃到對面終點的時候不使用ruleTree)
敵人沒達陣前我不達陣
但如果不達陣必定會造成把己方"最小子"吃掉時選擇最小子達陣

Experiment results and findings of your implementation.

Findings & Implementation

- Board
 - 由於上課提供的board在 `find (Color who, int num)`時要用 $O(n^2)$ 的複雜度找到特定子的位置，會造成simulate速度緩慢，因此我自己做了一個board，以確保在find的時候可以用 $O(1)$ 的時間找到特定子的位置，主要的方法是在cube上也紀錄子的位置，並在移動時同步更新board的指針及cube上的x,y座標
- Cube
 - 在cube中存著x,y座標以及棋子編號num
 - 在實作時為了用正負號分辨對手和己方旗子因此設定棋子編號分別為
 - 紅方：-1 ~ -6

- 藍方 : +1 ~ +6

```
class CUBE{
public:
    int x;
    int y;
    int num;
    CUBE(int px, int py, int pos) : x(px), y(py), num(pos){};
    CUBE() : x(7), y(7), num(0){};
};
```

• Random Simulate

- 9~9.5秒的simulate盤數(初始盤面)約為 550000 盤
- Random
 - 在實作random simulate時為了減少使用random的次數，我在遊戲開始時預先建立0~5共6!的排列組合陣列，因此每個move只需要呼叫一次random即可決定隨機落子之順序，移動方向的部分也使用類似方式用一個random決定所有子的移動方向 (共計一個random move需要呼叫2次random)
 - 此種方式雖然減少了呼叫random的次數，但作為代價，也降低了random的隨機性
- Rule base
 - 由於降低了抽樣的隨機性，因此我決定加上rule base來提升抽樣品質作為補救
 - 在純Random的狀況下會有許多不合理的move產生 (例如：當可以吃子時不吃子卻走其他沒用的步 / 在大優勢時把自己的大子走到終點，讓敵人獲勝) 因此在Random simulate時我實作了一些rule避免掉一些不合理的移動，大致上實作於此的rule如下：
 - 當可以吃敵人時傾向吃敵人(除非吃完後會被對手反吃且互換後是己方虧)
 - 不在敵人達陣前自己先達陣，除非沒路走或此步必勝，當沒路可走時，就讓號碼最小的達陣

• 長樹 `board::move_list()`

- 在長樹時有些不合理的步也應該不長出來，才不會浪費計算資源，大致上實作的長樹rule如下：
 - 可以吃子時，只長吃子步以及逃脫步
 - 上面的rule在有一方的棋子小於3或是該吃子步為達陣步時不使用此rule
 - 若有一方的棋子小於3還用rule可能會害死自己或是讓對方和我們平手
 - 若為達陣步還吃子可能會造成輸棋
 - 對方還沒達陣時不長達陣步的node，除非沒有其他路或是會造成自己的最小子被自己的大子吃掉時才選擇達陣

• 在挑pv時使用WinRate以及UCB score，但在選擇move時選擇nonLossRate最高者

- 如果在選擇move時選擇WinRate最高者會造成可以平手的棋下到輸

• Progressive Pruning

- 嘗試實作了兩種計分方式

- 終盤步數
- 終盤殘子數 / 殘子號碼
- 測試過後兩種都可以某種程度的切到勝率很低的node，但還是有切到好步的風險，由於我的 simulate 次數為 550000 次，評估後應該有足夠的能力搜尋，因此最終決定調高 PP 的門檻，以降低砍到不該砍的 node 的風險。

最終設定的參數為：

- PP_EFFECTIVE_NUM : 3000
- SIGMA_E : 15
- RD : 2

Experiment results

最後，提供最後一週測試時的結果統計，最後提交的版本為 shiro1219

(提交時拿掉了 log 和 debug 的 code 並更名為 Shiro_agent)

- 對 greedy 的勝率為 96% (96 勝 2 和 2 負)
- 對 conservative 的勝率為 100%

My version	Enemy	Total	Win	Lose	Draw	WinRate	Score(every 10 games)
shiro1214-ruleTree-v3	agentMCTUnevenTotalTrialPrune-log	16	16	0	0	100.00%	10
shiro1214-ruleTree-v3	shiro1214-ruleTree-v1	50	25	22	3	50.00%	0.6
shiro1214-ruleTree-v3	conservative	16	16	0	0	100.00%	10
shiro1216-ruleTree-v4	shiro1214-ruleTree-v3	44	17	22	5	38.64%	-1.136363636
shiro1216-ruleTree-newSimulate-v1	shiro1214-ruleTree-v3	20	20	0	0	100.00%	10
shiro1216-ruleTree-newSimulate-v1	greedy	30	28	2	0	93.33%	8.666666667
shiro1216-ruleTree-newSimulate-v1	conservative	20	20	0	0	100.00%	10
shiro1218-ruleTree-newSimulate-v2	shiro1214-ruleTree-v1	20	19	1	0	95.00%	9
shiro1218-ruleTree-newSimulate-v2	conservative	20	20	0	0	100.00%	10
shiro1218-ruleTree-newSimulate-v2	greedy	20	18	2	0	90.00%	8
shiro1218-ruleTree-newSimulate-v2	shiro1216-ruleTree-newSimulate-v1	100	38	27	35	38.00%	1.1
shiro1218-ruleTree-newSimulate-v3	greedy	50	47	0	3	94.00%	9.4
shiro1218-ruleTree-newSimulate-v3	shiro1216-ruleTree-newSimulate-v1	50	24	17	9	48.00%	1.4
shiro1218-final-pp15	greedy	50	47	0	3	94.00%	9.4
shiro1218-final-pp20	greedy	50	45	1	4	90.00%	8.8
shiro1218-final-pp15	shiro1218-final-pp20	50	14	23	13	28.00%	-1.8
shiro1218-ruleTree-newSimulate-v3	shiro1218-final-pp20	50	14	21	15	28.00%	-1.4
shiro1218-ruleTree-newSimulate-v3	shiro1218-final-pp15	50	25	14	11	50.00%	2.2
shiro1218-final-pp20	shiro1216-ruleTree-newSimulate-v1	50	16	24	10	32.00%	-1.6
shiro1218-final-pp15	shiro1216-ruleTree-newSimulate-v1	50	15	23	12	30.00%	-1.6
shiro1219	shiro1216-ruleTree-newSimulate-v1	70	34	20	16	48.57%	2
shiro1219	shiro1218-final-pp15	70	26	16	28	37.14%	1.428571429
shiro1219	shiro1218-final-pp20	370	138	94	138	37.30%	1.189189189
shiro1219	shiro1218-ruleTree-newSimulate-v3	72	36	12	24	50.00%	3.333333333
shiro1219	greedy	100	96	2	2	96.00%	9.4
shiro1219	conservative	100	100	0	0	100.00%	10