

人工智慧期末報告-第五組
00957003王丞頤 00957046賴建宇 00957049林家安

(一)策略研究:

王丞頤: (蒙地卡羅學習position heuristics):

蒙地卡羅是一種透過大量模擬去決定下一步的一種方法, 首先我蒙地卡羅的寫法是先設定一個整數的vector去紀錄所有步數的勝率, 再來輸入模擬次數, 並在每次模擬先由使用者可以下的步數中隨機去選擇一步, 並將下玩這一步的棋盤拿去模擬輸贏, 如果經過模擬的結果為贏的話則對應的score其+1, 最後去選擇贏面最大的步去當下一步, 因此我的heuristics為勝率也就是輸贏, 但我在實戰後發現其實蒙地卡羅並不強經常會被大比分, 所以感覺可能需要用到蒙地卡羅樹會好一點, 並且由於是隨機選擇所以也可會出現選擇相同步的情況導致模擬不平均等等, 造成蒙地卡羅並非是一個良好的策略選擇。

林家安:(minimax + 蒙地卡羅)

同王同學的蒙地卡羅法, 本策略是採取minimax和蒙地卡羅混和的方式。首先先測試模擬次數(700~1300)對蒙地卡羅法的影響, 發現除了執行時間有明顯的差異外, 結果沒有太大的變化。因此此在後續使用此方法時使用模擬1000次, 因為能夠在目標時間3秒內給予決策。

再來是將兩種方法結合。在這邊採取的策略是, 前2/3對局採用minimax, 而後1/3對局使用蒙地卡羅。原因是研究後發現minimax已經是很強的決策方法了, 所以給予其較多的決策權。且在剩下1/3的棋局時, 蒙地卡羅需要模擬的時間也比初始盤面來的少很多, 所以如此安排。

最後在測試時, 同王同學發現的一樣, 往往輪到蒙地卡羅法做決策時, 會給出一些明顯的壞手。也許是因為模擬次數太少所以導致模擬不平均(根據維基百科須放置30000個隨機點才能與真實的pi有0.07%相異)。也許棋局的模擬次數需要達到上萬次或是加入現實棋局的棋譜加以模擬才有可能得到一個稍微靠譜的決策。

賴建宇: (minimax with alpha-beta pruning)

minimax演算法假設對手擁有完美的決策能力, 從對對手最有利的局面中找出對自己最有利的局面。若對手不選擇對玩家的最壞情況, 則玩家面臨的情況一定比最壞解還要更佳; 因此雖然minimax不尋找最佳解, 但一定能確保在搜尋的深度內能找到對玩家最有利的情況。

由於minimax本質上是一種窮舉式的方法, 導致搜尋深度增加時需要非常多的時間及算力。alpha-beta pruning可防止minimax搜尋不必要的局面。把minimax畫成一棵樹, 對自己的兄弟節點來說, 若有兄弟能提供給父親更好的局面, 那自己這個節點就不必繼續搜尋了; 對自己父親輩的節點來說, 若有叔叔能提供給爺爺更好的局面, 那自己父親這個節點以下就不必繼續搜尋了。這

些複雜的判斷都能由alpha-beta兩個值定義判斷，將當前節點的情況記錄在alpha與beta兩個值中依次傳下去，就能由此判斷搜尋樹上目前的情況。並且alpha與beta的值還會不斷搜縮，進而讓剪枝的效果越來越好。

minimax對黑白棋來說是個很好的策略，節點的稀少性(能下的地方很有限)讓搜尋樹展開不至於太過龐大。然而對於如圍棋類有非常多可能性的遊戲來說，窮舉的負擔太過龐大，並不適合使用minimax演算法。

(二)對戰結果研究：

蒙地卡羅(模擬1000)VS蒙地卡羅(模擬2000):

經由實際去互相對戰可以發現模擬2000的不論先手或後手均比模擬1000的來的厲害並且經過本機上測驗發現五戰三勝得知後手的蒙地卡羅較強可以贏對面較多的棋子數。

```
0 0 0 0 0 X
0 0 0 X 0 0
0 0 0 0 X X
0 0 0 0 0 X
經過電腦(2000)下完後的棋盤
X X X X X X
0 0 0 0 0 +
0 0 0 0 0 X
0 0 0 X 0 0
0 0 0 0 X X
0 0 0 0 0 X
經過電腦(1000)下完後的棋盤
X X X X X X
0 0 0 0 0 X
0 0 0 0 X X
0 0 0 X 0 0
0 0 0 0 X X
0 0 0 0 0 X
模擬2000獲勝並且贏了10個棋子
=====
Process exited after 9.65 seconds with return value
```

蒙地卡羅(模擬1000)VS蒙地卡羅(模擬2000):

經由實驗得知先手後手總共完成的時間並非差很多，但在不同的模擬次數間對戰，其對戰總時常會隨者模擬次數而直線上升在雙方均為模擬兩千次對戰時總時長就需要將近一分鐘了與模擬1000次相比整整多了兩倍多一點的時間。

```
Mante-Carlo to learn heuristics:

當雙方均為蒙地卡羅AI
模擬1000次
黑旗vs白旗 勝方:黑旗 先功:黑棋 總時間為28.6秒
黑旗vs白旗 勝方:黑旗 先功:白棋 總時間為29.6秒

模擬1200次
黑旗vs白旗 勝方:白旗 先功:黑棋 總時間為34.22秒
黑旗vs白旗 勝方:黑旗 先功:白棋 總時間為38.5秒

模擬2000次
黑旗vs白旗 勝方:白旗 先功:黑棋 總時間為64.92秒
黑旗vs白旗 勝方:黑旗 先功:白棋 總時間為60.431秒
```

蒙地卡羅+MINMAX(模擬次數與時長的關係)

經由實驗發現模擬次數越長對時間的影響基本不大的在模擬次數為1000深度為6時期完成一局的時間大約為2秒多一點而在2000也是一樣，且除此之外在對戰間可以發現先攻對於戰局得影響是很大的，先攻的勝率會比較高一點。

```
Mante-Carlo to MinMax:
當一方為蒙地卡羅+MinMax AI 另外一方為randminmax
模擬1000次
黑旗vs白旗 勝方:黑旗 先功:黑棋 總時間為2.63秒
黑旗vs白旗 勝方:黑旗 先功:白棋 總時間為2.65秒
s
模擬1200次
黑旗vs白旗 勝方:白旗 先功:白棋 總時間為1.6秒
黑旗vs白旗 勝方:白旗 先功:白棋 總時間為1.5秒

模擬2000次
黑旗vs白旗 勝方:白旗 先功:白棋 總時間為2.33秒
黑旗vs白旗 勝方:黑旗 先功:黑棋 總時間為2.374秒
```

MINMAX間的比較:

深度	勝方	黑方花時	白方花時	勝子數
6	黑	0.66	0.462	3
8	白	2.651	2.041	9
9	黑	5.656	9.889	5
10	黑	31.245	46.415	3



以上是雙方雙方對戰一局的情況，而以下將模擬局數增加到100以上。

深度為8的情況下雙方對戰100場，先手的黑棋贏下61局，平均贏19子；後手的白棋贏下37局，平均贏20子；和局2場。

深度為7的情況下雙方對戰100場，先手的黑棋贏下57局，平均贏22子；後手的白棋贏下37局，平均贏15子；和局6場。

深度為6的情況下雙方對戰500場，先手的黑棋贏下32.4%局，平均贏16子；後手的白棋贏下63.2%局，平均贏20子；和局4.4場。

深度為5的情況下雙方對戰500場，先手的黑棋贏下45%局，平均贏17子；後手的白棋贏下52.2%局，平均贏19子；和局2.8%場。

深度為4的情況下雙方對戰500場，先手的黑棋贏下29.4%局，平均贏15子；後手的白棋贏下69%局，平均贏20子；和局1.6%場。

深度為4的情況下雙方對戰500場，先手的黑棋贏下38.4%局，平均贏14子；後手的白棋贏下58.4%局，平均贏18子；和局3.2%場。

可以看的出來，在模擬深度為偶數的情況下，先攻的黑棋表現較佳；反之在模擬深度為奇數的情況下，後攻的白棋表現較佳。不過隨著模擬深度增加，情況對先手的黑棋越來越有利，除了能贏下更多局外，每局贏下的目數也在逐漸增加。另外和局數也基本隨著模擬深度增加而增加。

在大量模擬前我以為minimax的特性會讓雙方的目數差距縮小，但事實上大部分的場次輸贏雙方的目數都相差巨大，還經常出現一方將另一方全殲的局面。

蒙地卡羅VS蒙地卡羅&MINMAX:

```

XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
+XXXXXXXXX
=====
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
+XXXXXXXXX
=====
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
+XXXXXXXXX
=====
Simulation time: 36.64 seconds

Process exited after 36.96 seconds with return value 0
請按任意鍵繼續 . . . . .

=====
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
+XXXXXXXXX
=====
Simulation time: 32.942 seconds

Process exited after 33.27 seconds with return value 0
請按任意鍵繼續 . . . . .

=====
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
+XXXXXXXXX
=====
Simulation time: 27.154 seconds

=====
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX
+XXXXXXXXX
=====
Simulation time: 27.154 seconds

```

模擬次數(2000 蒙地卡羅 先攻)

```
+X X X++
=====
++++++
O+X X++
O+X X O+
O X O O+
++X+O+
+X X X++
=====
++++++
O+X X++
O+X X X+
O X X X X
++X+X+
+X X X++
+X X X++
Simulation time: 18.177 seconds
-----
Process exited after 18.88 seconds with return value 0
請按任意鍵繼續 . . .
```

```
000XXX
XXXXXX
XXXXXX
XXXXXX
XOX OXX
XOX OXX
XOX OXX
XOX OXX
Simulation time: 31.293 seconds
-----
Process exited after 31.61 seconds with return value 0
請按任意鍵繼續 . . .
```

```
XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX
XXXXXX
Simulation time: 28.192 seconds
-----
Process exited after 28.52 seconds with return value 0
請按任意鍵繼續 . . .
```

蒙地卡羅VS MINMAX+AlphaBeta pruning:

由我們實際測的結果顯示發現蒙地卡羅均被MINMAX給死死的克制，在MINMAX深度為4時蒙地卡羅模擬次數為2000時，當蒙地卡羅為先攻時雖然還是輸但可以剩下比較多的棋子的可能性比較大，比較不會接近全吃光，但一到了深度為6基本上，能剩下的棋子數都不多甚至可以說很少，這個情況是在不論先手或後手時均相同，因此可以很容易猜測出MINMAX一定在對戰上會比蒙地卡羅來的好，除非可能需要蒙地卡羅的模擬次數再往上才有可能會比較強一點，因此比起蒙地卡羅我們會選擇MINMAX+AlphaBeta pruning去作為我們的最佳對戰策略。

成果圖:

黑色:蒙地卡羅 白色:MINMAX

模擬次數(2000 深度4 MINMAX 先攻)

```
-----
a b c d e f
A _ _ . . . _
B . _ . . . .
C . . . . . .
D . . . . . .
E . . . . _ _
F . . . . _ _
-----
white wins
black wins-28pieces
black spent=0
white spent=0
```

```
-----
a b c d e f
A . . . . . _
B . . . . . .
C . . . . . .
D . . . . . .
E . . . . . .
F . . . . . .
-----
white wins
black wins-31pieces
black spent=0
white spent=0
```

```
-----
a b c d e f
A . . . . . .
B . _ . . . .
C . . . . . .
D . . . . . .
E . . . . . .
F . . . . . .
-----
white wins
black wins-31pieces
black spent=0
white spent=0
```

模擬次數(2000 深度4 蒙地卡羅 先攻)

```
-----
a b c d e f
A _ _ . . . _
B . _ . . . .
C . . . . . .
D . . . . . .
E . . . . _ _
F . . . . _ _
-----
white wins
black wins-28pieces
black spent=0
white spent=0
```

```
-----
a b c d e f
A . . . . . .
B . . . . . .
C . . . . . .
D . . . . . .
E . . . . . .
F . . . . . .
-----
white wins
black wins-22pieces
black spent=0
white spent=0
```

```
-----
a b c d e f
A . . . . . .
B . . . . . .
C . . . . . .
D . . . . . .
E . . . . . .
F . . . . . .
-----
white wins
black wins-24pieces
black spent=0
white spent=0
```

模擬次數(2000 深度6 MINMAX 先攻)

	a	b	c	d	e f
A	●	●	●	●	● ○
B	●	●	●	●	● ●
C	●	●	●	●	● ●
D	●	●	●	●	● ●
E	●	●	●	●	● ●
F	●	●	○	●	● ●

white wins					
black wins-32pieces					
black spent=12.931					
white spent=0.373					

	a	b	c	d	e f
A	●	●	●	●	○ ●
B	●	●	●	○ ●	● ●
C	●	●	○ ●	○ ●	● ●
D	●	●	○ ●	○ ●	● ●
E	●	●	○ ●	● ●	● ●
F	●	●	●	●	● ●

white wins					
black wins-26pieces					
black spent=15.677					
white spent=0.364					

	a	b	c	d	e f
A	-	●	-	●	-
B	-	-	●	●	●
C	●	-	●	●	-
D	●	●	●	●	-
E	●	●	●	●	-
F	●	●	●	●	●

white wins					
black wins-26pieces					
black spent=11.53					
white spent=0.289					

模擬次數(2000 深度6 蒙地卡羅 先攻)

	a	b	c	d	e f
A	●	●	●	●	●
B	●	●	○	●	●
C	●	●	●	●	●
D	●	●	●	●	●
E	●	●	●	●	●
F	○	○	○	●	●

white wins					
black wins-28pieces					
black spent=15.725					
white spent=0.395					

	a	b	c	d	e f
A	●	●	●	●	-
B	●	●	○	●	-
C	●	●	●	●	●
D	●	●	●	●	●
E	●	●	●	●	●
F	●	●	●	●	●

white wins					
black wins-34pieces					
black spent=11.462					
white spent=0.319					

game over from way					

	a	b	c	d	e f
A	●	●	●	-	●
B	●	●	●	-	●
C	●	●	●	-	●
D	●	●	●	●	●
E	●	●	●	●	●
F	-	●	●	●	●

white wins					
black wins-32pieces					
black spent=15.304					
white spent=0.194					

MINMAX+AlphaBeta pruning VS 蒙地卡羅+MINMAX :

將蒙地卡羅+MINMAX的MINIMAX深度限制在6層，觀察兩個策略間對戰的情況。在MINMAX+AlphaBeta pruning的深度為4. 先攻的情況下蒙地卡羅+MINMAX整體還能保持部分優勢，以2:1的分數取得勝利。但到了MINMAX+AlphaBeta pruning後攻時，蒙地卡羅+MINMAX毫無還手之力。由此可知蒙地卡羅不是個好策略，甚至會拖累原本的minimax。

在MINMAX+AlphaBeta pruning的深度為6時，更是幾乎每局都虐殺蒙地卡羅+MINMAX。另外蒙地卡羅+MINMAX的時間花費都是 MINMAX+AlphaBeta pruning的十倍以上。

成果圖：

白色:蒙地卡羅 黑色:MINMAX

模擬次數(2000 深度4 MINMAX 先攻)1:2

	a	b	c	d	e f
A	○	○	○	○	○ ●
B	●	●	○	○	● ●
C	●	●	○	○	○ ●
D	●	○	○	○	○ ●
E	●	○	●	●	● ●
F	●	●	●	●	● ●

white wins					
white wins 6pieces					
black spent= 0.04sec					
white spent= 8.09sec					

	a	b	c	d	e f
A	○	○	○	○	○ ●
B	○	○	○	○	○ ○
C	○	○	○	○	○ ○
D	○	○	○	○	○ ●
E	○	○	○	○	○ ●
F	○	○	○	○	○ ●

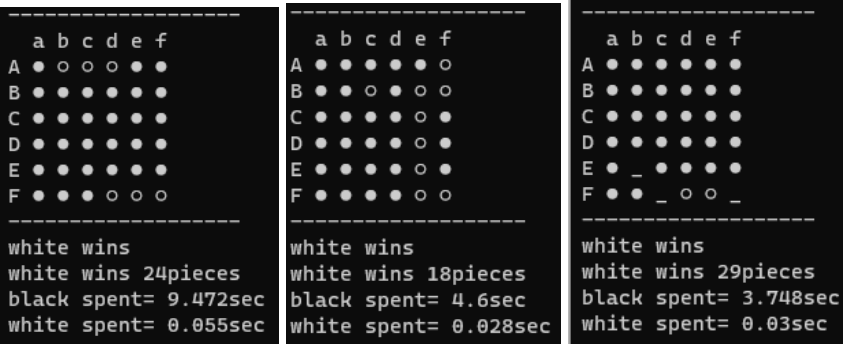
black wins					
black wins 6pieces					
black spent= 0.026sec					
white spent= 5.412sec					

	a	b	c	d	e f
A	○	○	○	○	○ ●
B	●	●	●	●	● ●
C	●	●	●	●	● ●
D	●	●	●	●	○ ●
E	●	●	●	●	○ ●
F	●	●	●	●	○ ●

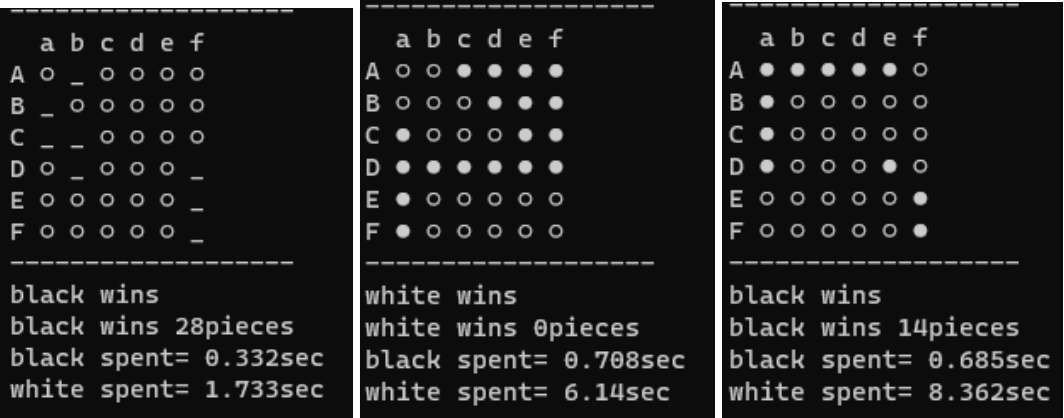
white wins					
white wins 28pieces					
black spent= 0.032sec					
white spent= 6.161sec					

黑色:蒙地卡羅 白色:MINMAX

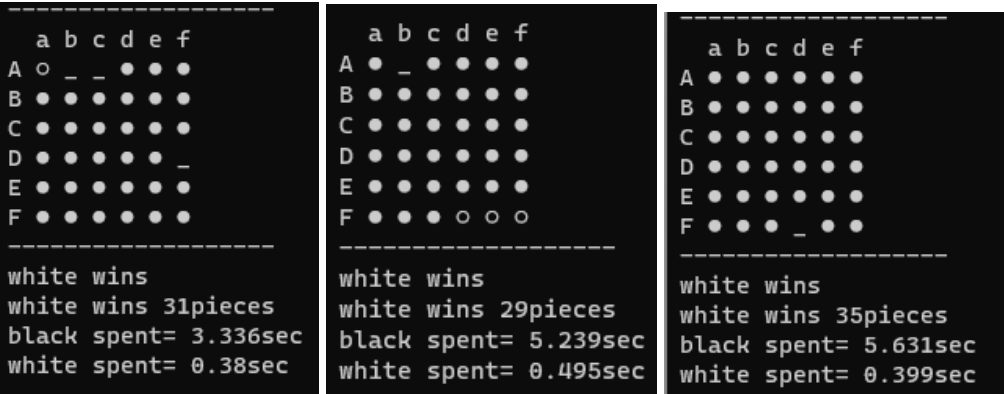
模擬次數(2000 深度4 蒙地卡羅 先攻)0:3



白色:蒙地卡羅 黑色:MINMAX
模擬次數(2000 深度6 MINMAX 先攻) 0:2 1局平手



黑色:蒙地卡羅 白色:MINMAX
模擬次數(2000 深度6 蒙地卡羅 先攻) 0:3



(三)最佳策略以及先手後手之影响

我們經過實際的兩兩對戰中，覺得MMINMAX+AlphaBeta pruning 會是我們的最佳策略，並且隨著深度增加在先手的情況贏面會比較大。

我們也寫了一個和人類和minimax電腦對戰的程式，在讓我們和電腦對戰，並限制我們每一步的思考時間為一分鐘。在測試中，我們勉強能跟深度為4~5層的minimax有一戰之力，不過在上去就打不贏了。再來我們拿這個程式和Play商店中的黑白棋遊戲電腦對戰，發現遊戲中難度最難的電腦大概和深度為8層的minimax表現差不多互有勝負，先後手都相同。

接著我們使用深度為7層的minimax程式去挑戰網路上的其他人類玩家。令人意外的是，居然有不少人類玩家能夠打贏，和我們原先預測的相差甚遠，必盡minimax等於把7層內的所有可能性都看過一次並找到最佳解。

我們觀察到minimax輸給人類的地方通常在於邊角的掌控，通常minimax在開局都能下出優勢的局面，然而棋局進行到中斷，minimax常常會下一些錯誤的邊角給對方製造機會，或許這和深度為7的視野有關。