CS 440 Assignment 2

R section 3 credits

Group members:
Chendi Lin, Zhuoyue Wang

1.Overview

In this assignment, we try to use backtracking searching and several algorithms to solve constraint satisfaction problems and games.

In the first part,

In the second part, we implements a simple two-player zero-sum game called Breakthrough, and use minimax search and alpha-beta search to simulate two players' actions. Besides these, we also create two evaluation functions, "Defensive" one and "Offensive" one to help agents find best-fit actions.

2.Work distribution

Chendi Lin: Flow Free problems, report

Zhuoyue Wang: Breakthrough game problems, report

3. Part 1: CSP - Flow Free

3.1. Smaller inputs

3.2. Bigger inputs (bonus)

3.3. Bonus

4. Part 2: Game of Breakthrough

In this section, we try to implements a simple two-player zero-sum game called Breakthrough, and use minimax search and alpha-beta search to simulate two players' actions. Besides these, we also create two evaluation functions, "Defensive" one and "Offensive" one to help agents find best-fit actions.

There are our evaluation functions:

Defensive Heuristic 1:

6*(my remaining pieces) + random()

Offensive Heuristic 1:

6*(30-opponent’s remaining pieces) +random()

Defensive Heuristic 2:

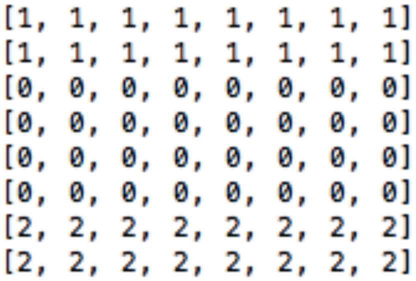
6*(my remaining pieces) - 2*(opponent’s remaining piece)+random()

Offensive Heuristic 2:

2*(my remaining pieces) - 6*(opponent’s remaining pieces)+random()

In our Offensive and Defensive Heuristic 2, we start to consider the difference between opponent’s remaining pieces and my remaining pieces to avoid the unstable cases that both have many pieces or both have few pieces.

2.1 Minimax and alpha-beta agents



```
[1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[2, 2, 2, 2, 2, 2, 2, 2]
[2, 2, 2, 2, 2, 2, 2, 2]
```

1 —> Black

2 —> White

0 —> empty

1. Minimax (Offensive Heuristic 1) vs Alpha-beta (Offensive Heuristic 1)

Winner:

White: Alpha-beta (Offensive Heuristic 1)

White:

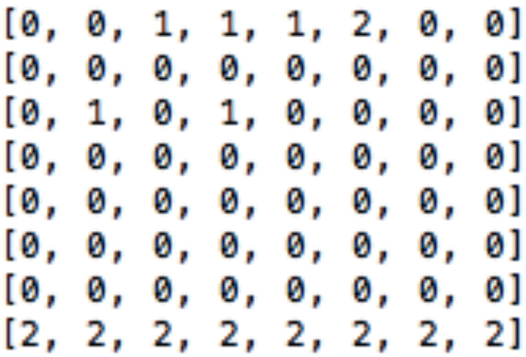
Total steps: 25

Total expand game tree nodes: 13895961

Average expanded nodes per move: 555838.4

Average time to make a move: 13.8

Number of opponent captured: 11



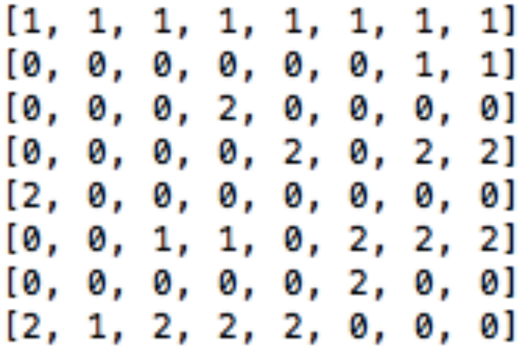
```
[0, 0, 1, 1, 1, 2, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 0, 1, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[2, 2, 2, 2, 2, 2, 2, 2]
```

Black:
Total steps: 25
Total expand game tree nodes: 349520
Average expanded nodes per move: 13980.8
Average time to make a move: 0.17
Number of opponent captured: 7

2. Alpha-beta (Offensive Heuristic 2) vs Alpha-beta (Defensive Heuristic 1)

Winner:
Black: Alpha-beta (Offensive Heuristic 2)

White:
Total steps: 25
Total expand game tree nodes: 6583669
Average expanded nodes per move: 263346.7
Average time to make a move: 13.3
Number of opponent captured: 3

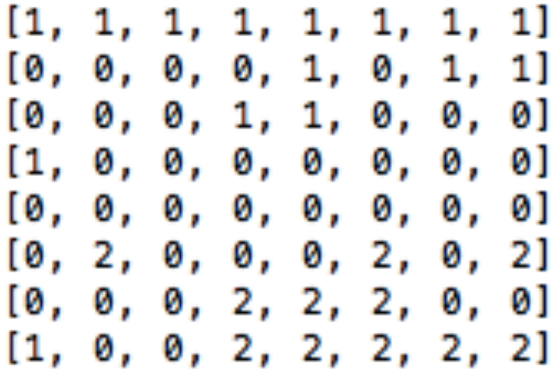


```
[1, 1, 1, 1, 1, 1, 1, 1]
[0, 0, 0, 0, 0, 0, 1, 1]
[0, 0, 0, 2, 0, 0, 0, 0]
[0, 0, 0, 0, 2, 0, 2, 2]
[2, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 1, 0, 2, 2, 2]
[0, 0, 0, 0, 0, 2, 0, 0]
[2, 1, 2, 2, 2, 0, 0, 0]
```

Black:
Total steps: 26
Total expand game tree nodes: 3563706
Average expanded nodes per move: 137065.6
Average time to make a move: 9.1
Number of opponent captured: 3

3. Alpha-beta (Defensive Heuristic 2) vs Alpha-beta (Offensive Heuristic 1)

Winner:
Black: Alpha-beta (Defensive Heuristic 2)



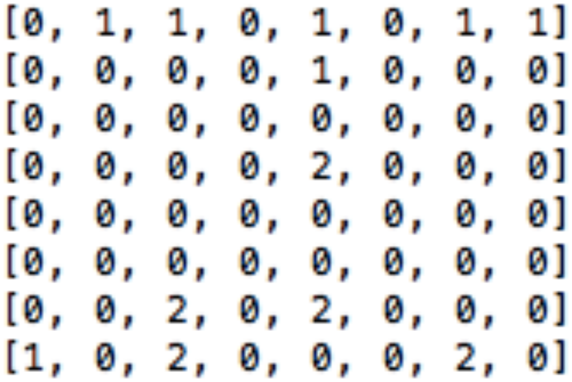
```
[1, 1, 1, 1, 1, 1, 1, 1]
[0, 0, 0, 0, 1, 0, 1, 1]
[0, 0, 0, 1, 1, 0, 0, 0]
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 2, 0, 0, 0, 2, 0, 2]
[0, 0, 0, 2, 2, 2, 0, 0]
[1, 0, 0, 2, 2, 2, 2, 2]
```

White:
Total steps: 14
Total expand game tree nodes:6854829
Average expanded nodes per move: 489630.6
Average time to make a move: 14.7
Number of opponent captured: 1

Black:
Total steps: 15
Total expand game tree nodes: 914810
Average expanded nodes per move: 60987.3
Average time to make a move: 4.0
Number of opponent captured: 5

4. Alpha-beta (Offensive Heuristic 2) vs Alpha-beta (Offensive Heuristic 1)

Winner:
Black: Alpha-beta (Offensive Heuristic 2)



```
[0, 1, 1, 0, 1, 0, 1, 1]
[0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 2, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 2, 0, 2, 0, 0, 0]
[1, 0, 2, 0, 0, 0, 2, 0]
```

White:
Total steps: 35
Total expand game tree nodes: 14955900
Average expanded nodes per move: 427311.4
Average time to make a move: 12.3
Number of opponent captured: 9

Black:

Total steps: 36

Total expand game tree nodes: 4410448

Average expanded nodes per move: 122512.4

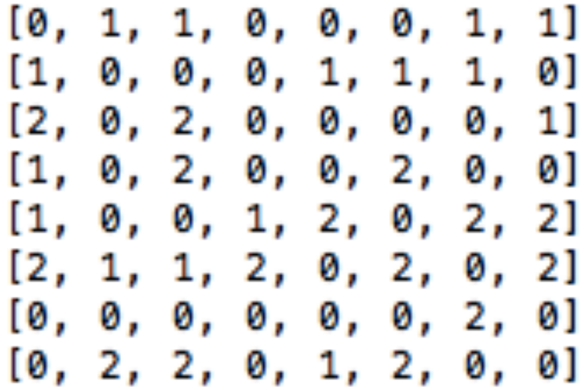
Average time to make a move: 5.7

Number of opponent captured: 11

5. Alpha-beta (Defensive Heuristic 2) vs Alpha-beta (Defensive Heuristic 1)

Winner:
Black: Alpha-beta (Defensive Heuristic 2)

White:
Total steps: 30
Total expand game tree nodes: 12209874
Average expanded nodes per move: 406995.8
Average time to make a move: 19.4
Number of opponent captured: 1

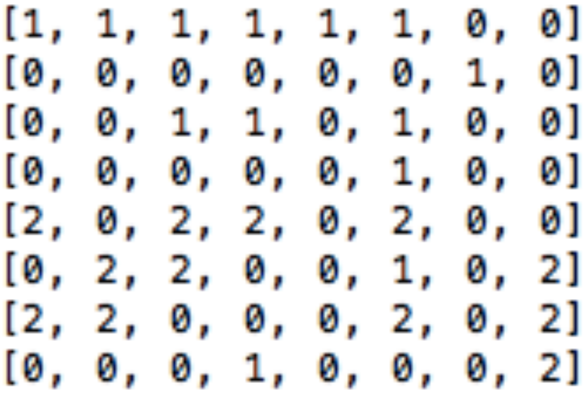


```
[0, 1, 1, 0, 0, 0, 1, 1]
[1, 0, 0, 0, 1, 1, 1, 0]
[2, 0, 2, 0, 0, 0, 0, 1]
[1, 0, 2, 0, 0, 2, 0, 0]
[1, 0, 0, 1, 2, 0, 2, 2]
[2, 1, 1, 2, 0, 2, 0, 2]
[0, 0, 0, 0, 0, 0, 2, 0]
[0, 2, 2, 0, 1, 2, 0, 0]
```

Black:
Total steps:
Total expand game tree nodes: 31
Average expanded nodes per move: 7444547
Average time to make a move: 240146.7
Number of opponent captured: 1

6. Alpha-beta (Offensive Heuristic 2) vs Alpha-beta (Defensive Heuristic 2)

Winner:
Black: Alpha-beta (Offensive Heuristic 2)



```
[1, 1, 1, 1, 1, 1, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0]
[0, 0, 1, 1, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 1, 0, 0]
[2, 0, 2, 2, 0, 2, 0, 0]
[0, 2, 2, 0, 0, 1, 0, 2]
[2, 2, 0, 0, 0, 2, 0, 2]
[0, 0, 0, 1, 0, 0, 0, 2]
```

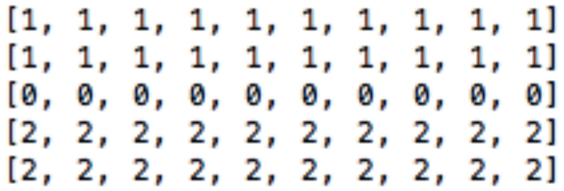
White:
Total steps: 25
Total expand game tree nodes: 13849007

Average expanded nodes per move: 553960.2
Average time to make a move: 22.9
Number of opponent captured: 4

Black:
Total steps: 26
Total expand game tree nodes: 23022326
Average expanded nodes per move: 885474.1
Average time to make a move: 32.2
Number of opponent captured: 4

According to the results, we can see that firstly using Alpha-beta search can defeat minimax search based on the same heuristic function. Secondly, our Heuristic 2 functions defeat all of Heuristic 1 functions, which means it is desirable to evaluate the decision based on both side’s current performance. Moreover, when we run the last game Alpha-beta (Offensive Heuristic 2) vs Alpha-beta (Defensive Heuristic 2) for several times, we find Offensive Heuristic 2 has 80% chance to win (4/5). We believe it is because it expands more nodes

2.2 Extended rules (bonus)



```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
[2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

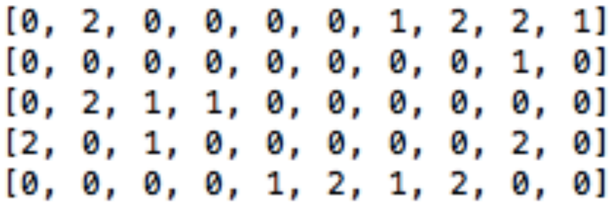
1 —> Black
2 —> White
0 —> empty

In this part, to compare the difference between both parts, we use the same evaluation functions as the previous part. In addition, For convenience, we will only show the matches with different result from previous ones.

1. Alpha-beta (Offensive Heuristic 2) vs Alpha-beta (Defensive Heuristic 2)

Winner:
White: Alpha-beta (Denfensive Heuristic 2)

White:
Total steps: 30
Total expand game tree nodes: 5709081
Average expanded nodes per move: 190302.7
Average time to make a move: 8.0
Number of opponent captured: 12



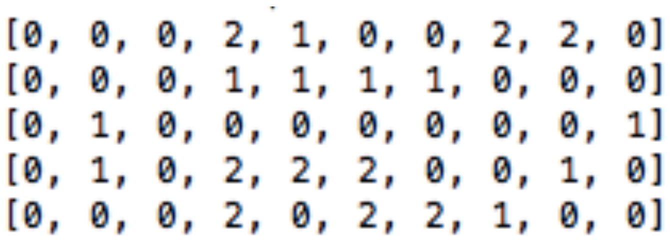
```
[0, 2, 0, 0, 0, 0, 1, 2, 2, 1]
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 2, 1, 1, 0, 0, 0, 0, 0, 0]
[2, 0, 1, 0, 0, 0, 0, 0, 2, 0]
[0, 0, 0, 0, 1, 2, 1, 2, 0, 0]
```

Black:
Total steps: 30
Total expand game tree nodes: 8103393
Average expanded nodes per move: 270113.1
Average time to make a move: 10.3
Number of opponent captured: 14

2. Alpha-beta (Defensive Heuristic 2) vs Alpha-beta (Defensive Heuristic 1)

Winner:
White: Alpha-beta (Denfensive Heuristic 1)

White:
Total steps: 27
Total expand game tree nodes: 5539002
Average expanded nodes per move: 205148.2
Average time to make a move: 9.0
Number of opponent captured: 11



```
[0, 0, 0, 2, 1, 0, 0, 2, 2, 0]
[0, 0, 0, 1, 1, 1, 1, 0, 0, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1]
[0, 1, 0, 2, 2, 2, 0, 0, 1, 0]
[0, 0, 0, 2, 0, 2, 2, 1, 0, 0]
```

Black:
Total steps: 27

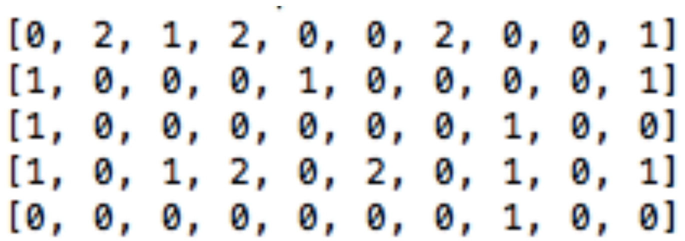
Total expand game tree nodes: 4749158
Average expanded nodes per move: 175894.7
Average time to make a move: 8.6
Number of opponent captured: 11

2. Alpha-beta (Offensive Heuristic 2) vs Alpha-beta (Defensive Heuristic 1)

Winner:
White: Alpha-beta (Denfensive Heuristic 1)

White:
Total steps: 29
Total expand game tree nodes: 5197675
Average expanded nodes per move: 179230.2
Average time to make a move: 7.8
Number of opponent captured: 8

Black:
Total steps: 29
Total expand game tree nodes: 8264231
Average expanded nodes per move: 284973.4
Average time to make a move: 10.8
Number of opponent captured: 15



```
[0, 2, 1, 2, 0, 0, 2, 0, 0, 1]
[1, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 1, 0, 0]
[1, 0, 1, 2, 0, 2, 0, 1, 0, 1]
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
```

By these results, we can see that the interesting difference that Denfensive Heuristic 1 defeats other evaluation functions. We guess the reason is that in such condition with extended rule and the smaller board, the player should give priority to only consider his remaining piece to achieve victory.