A2

Maxime Mahdavian

Common to both heuristics

- I cheat a little bit for both heuristics. When going through the different states of the board with both alpha-beta and minimax, the algorithm checks if the state is a goal state (for either players) and returns an appropriately big value.
  Like this, I can quickly give a heuristic value to some state, without having to evaluate it with a function.
- The value given to a state if this happens is ±(n*n+1)

E1

- Tries to make clusters of the same pieces. The more and bigger those clusters become, the better for the heuristic.
- Uses DFS to get the state of the neighbours to find islands of size bigger than one.
- Assuming that player 1 is using the heuristic the value is calculated by: #clusters_p1 - #clusters_p2
-

E2

- E2 takes goes through every possible winning set of moves on the board and checks how close or far the player is to winning on that set.

- Their own pieces gives them 2 points, opposing pieces subtracts 2 points, empty gives them 1 point. (It is useful since it could lead to a win eventually) and a block piece removes 1 point.

Scoreboard-4435

- Here, performance of both algorithms is very bad at the beginning of the game. Going at depth 6 with minimax is very costly. It eventually gets going after a couple moves.
- This favours e2,since e1 doesn’t have time to create good clusters at the beginning. E2 doesn’t have this limitation and can make better use of a bad beginning.
- It seems important for e1 to have a good beginning.
- We can see the slow start in the gameTrace and also see how the computation quickly gets higher as the board gets filled (less state to go through).

Scoreboard-4431

- Total win for e1.
- This is because it is faster overall than e2. Which means that it goes deeper than e2 with the tight time limit.
- E1 also seems to be doing better on smaller boards. This is most probably because it is easier for the opponent to break up clusters, causing more of them to naturally build up and improve the performance of e1.
- However, this wins seems to be only happen because of minimax where e2 can’t go deep enough in the tree to make a informed decision.

Scoreboard-5441

- Here we see that the two heuristics tie all the time on medium boards.
- The smaller depth of e1 doesn’t affect it very much, it seems like it can quickly find a good solution. This is most probably because it doesn’t take much computation to make clusters and e2 doesn’t tend to be very big clusters, going in and out of potential winning situations. The shortcut to quickly find losing situations also help it tremendously here.
- We see that both heuristics usually go the distance since most of the evaluations are at depth 6 and 2. Alpha-beta is the main cause here.

Scoreboard-5445

- Again, all games ended in ties. Not very surprising given the last scoreboard.
- The increase of time does not affect the results, since both heuristics went all the way down most of the time.
- We can confirm here the finding that the results of e1 doesn’t seem to find better results when going deeper . Like with I mentioned in the last slide, it is probably due to the simplicity of making clusters.

Scoreboard-8551

- The situation from the previous board seems to be the same when having a bigger board. Ties all the time.
- The shortcut for losing situation is very useful for e1 as it blocks the attempts of e2 to win. We can see this since e1 has a lot of moves where it only does one evaluation.
- The time constraint is surprisingly not an issue here. Alpha-beta seems very good at pruning the tree and giving an answer fast. However, e2 does seem to get cutoff, especially at the beginning of the game.

Scoreboard-8555

- All ties here again.
- Not much different than the last example. The lose situation shortcut is working wonders for both heuristic as it seems to completely crush the attempts of the opposition to win.

Scoreboard-8651

- Streak of all ties stop here. E2 gets one win over e1.
- Here, we see one of e1 weakness truly emerge. This heuristic seems pretty decent on smaller boards, but as boards and winning line up sizes become bigger, the heuristics seem to find some problems finding good moves. This seems to be because of DFS taking too long to find the appropriate clusters within the tight time constraint of this problem. This allows e2 to place blocks much better and win some games.
- The flat time of e2 is an advantage here because it ends up caring much less about the size of the winning line up size.

Scoreboard-8655

- Here, I can confirm some of my assumptions about the last scoreboard.
- E1 gets two wins over e2. The more time allocated to find a move allows e1 to form better clusters. This seems to overwhelm e2. (clusters have more chance of having more than one winning move per turn)
- The opposite seems to be happening here, the increase in the size of the winning line up seems to be affecting e2. Since it is a flat time, it cannot get better here unlike e1.