

Introduction

There are 2 objectives in this project. The first one is to implement the Minimax and Alpabeta pruning algorithms. The second part is to try different heuristics and see how they behave against some other players. The aim of this document is to describe this heuristics.

For testing the performace of the heuristics I used the code from tournament.py. Since there are some randomness involved in the games played I increased the number of matches played to 1,000 for each heuristic.

Heuristics

For testing the 3 different heuristics 4 bots have been created. One as benchmark and 3 other using each heuristic. Those bots have competed against 7 simpler bots. The number of wins for every bot can be seen in Table 1.

	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
Random	860	864	885	866
MM_Open	508	515	538	519
MM_Center	708	691	687	705
MM_Improved	497	553	524	511
AB_Open	511	521	522	498
AB_Center	710	708	699	669
AB_Improved	504	520	503	466
Total	61.40%	62.46%	62.26%	60.49%

Table 1: Number of wins for each heuristic

With these stats we can see that bot 1 and 2 are slightly better than the benchmark bot. So the heuristics they are using are better (Details in Figure 1).

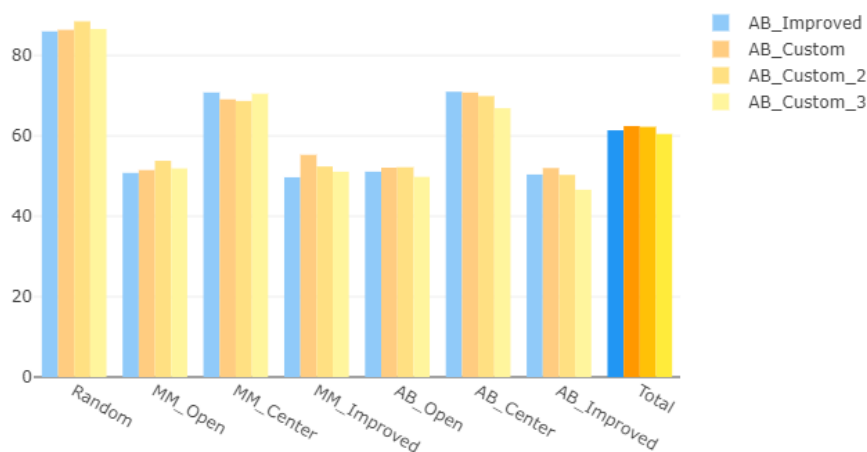


Figure 1: Percentage of wins for each heuristic

Common aspects of each heuristic

First of all the first for the first move the bot will try to go to the center of the board. If the bot it is not the first player it will check if it is allowed to go there. After some research this slight change has proven to be very useful.

The other thing the 3 heuristics share is that they will work differently depending on how many blank space are left in the board. By using that they can be aware of some problems that can happen at the end of the game.

Heuristic 1 (AB_Custom)

If less than 25% of the squares have been occupied the bot will take into account 2 things:

- Number of moves - number of opponent's moves
- A factor that stats how far the player is going from the center

In general the first factor will be the more useful. The second is intended only to differentiate between similar options, given a best results to the one that are in the center.

For the last part of the game the bot will also be penalized if it tries to go to the corner. The idea is that in the corners is more common to get trapped.

Heuristic 2 (AB_Custom_2)

This bot works similar to the first one but is simpler.

When less than 50% of the squares have been occupied it will do the same as in the first part of heuristic 1. It will forget the part about going to the center in order to only focus on surviving and trying to trap the other player.

Heuristic 3 (AB_Custom_3)

This last heuristic is the simplest one. For the first 50% of the game it will only take into account his number of movements and the ones of the other player.

For the last part of the game it will be more focused on surviving rather than trying to trap the opponent.