# Heuristic Analysis

For the heuristic analysis we will evaluate 3 heuristics in the custom\_scores and comment about the ID\_Improved.

## Three heuristics

The heuristics are:

### Difference between number of player’s movements minus number of opponent’s movements – Heuristic 1

Here it is the code:

return (float(len(game.get\_legal\_moves(player))) - len(game.get\_legal\_moves(game.get\_opponent(player))))

This is the simplest heuristic we analysed. The results were 66.71% student against 66.25% ID\_Improved. In the end evaluating this heuristic against the ID\_Improved, they are really calculating the same thing.

The difference in performance is due to the Student code to be faster being able to evaluate deeper positions.

### Average difference between number of player’s movements minus number of opponent’s movements for one move ahead – Heuristic 2

Code:

score = 0

for move in game.get\_legal\_moves(player):

game\_copy = game.forecast\_move(move)

score += float(len(game\_copy.get\_legal\_moves(player))) - len(game\_copy.get\_legal\_moves(game.get\_opponent(player)))

division = (len(game.get\_legal\_moves(player)) if len(game.get\_legal\_moves(player)) != 0 else 0.001)

return score / division

The performance here was 67.21% Student and 67.96% ID\_Improved. When we go one movement ahead we are seeing more into the future than the ID\_Improved but since we are spending more time the result is approximately the same.

### Average difference between number of player’s movements minus number of opponent’s movements for two moves ahead – Heuristic 3

Code:

score = 0

division = 0

for move in game.get\_legal\_moves(player):

game\_copy = game.forecast\_move(move)

for second\_move in game\_copy.get\_legal\_moves(game.get\_opponent(player)):

game\_second\_copy = game\_copy.forecast\_move(second\_move)

score += (

float(len(game\_second\_copy.get\_legal\_moves(player))) -

len(game\_second\_copy.get\_legal\_moves(game.get\_opponent(player))))

division += 1

division = (division if division != 0 else 0.001)

return score / division

The performance here was 58.82% Student and 67.00% ID\_Improved. When we increased the code the time to run it became too high, in the end we couldn’t perform better than ID\_Improved. We lost too much time performing the custom\_score and were not seeing deeper enough in the end.

## Table with runs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Heuristic 1 | | Heuristic 2 | | Heuristic 3 | |
| ID\_Improved | Student | ID\_Improved | Student | ID\_Improved | Student |
| 65.71 | 62.86 | 68.57 | 65.00 | 67.14 | 58.57 |
| 66.43 | 65.00 | 64.29 | 65.00 | 69.29 | 57.14 |
| 68.57 | 60.00 | 69.29 | 70.00 | 70.00 | 59.29 |
| 62.86 | 71.43 | 71.43 | 65.00 | 64.29 | 62.14 |
| 65.00 | 65.71 | 64.29 | 61.43 | 71.43 | 56.43 |
| 63.57 | 70.00 | 68.57 | 65.71 | 67.86 | 56.43 |
| 60.00 | 62.86 | 67.86 | 67.14 | 70.71 | 63.57 |
| 67.14 | 67.14 | 66.43 | 62.86 | 67.14 | 62.86 |
| 68.57 | 67.86 | 77.14 | 71.43 | 70.00 | 62.14 |
| 63.57 | 67.14 | 70.00 | 68.57 | 61.43 | 55.71 |
| 61.43 | 71.43 | 69.29 | 70.71 | 61.43 | 60.71 |
| 67.86 | 71.43 | 60.71 | 67.14 | 72.14 | 60.00 |
| 69.29 | 68.57 | 70.71 | 65.71 | 72.86 | 57.86 |
| 73.57 | 68.57 | 69.29 | 69.29 | 67.14 | 55.71 |
| 70.00 | 64.29 | 66.43 | 69.29 | 66.43 | 59.29 |
| 67.86 | 62.14 | 67.86 | 68.57 | 65.00 | 55.00 |
| 62.86 | 74.29 | 65.00 | 67.14 | 67.14 | 59.29 |
| 62.86 | 61.43 | 64.29 | 65.00 | 62.86 | 60.71 |
| 66.43 | 70.00 | 68.57 | 75.71 | 55.71 | 61.43 |
| 71.43 | 62.14 | 69.29 | 63.57 | 70.00 | 52.14 |

## Conclusion

In the end the heuristic choosen is the one that see one move ahead. It is similar to the ID\_Improved but it is more prepared to the end of the game, when one move can be the difference between winning and losing.