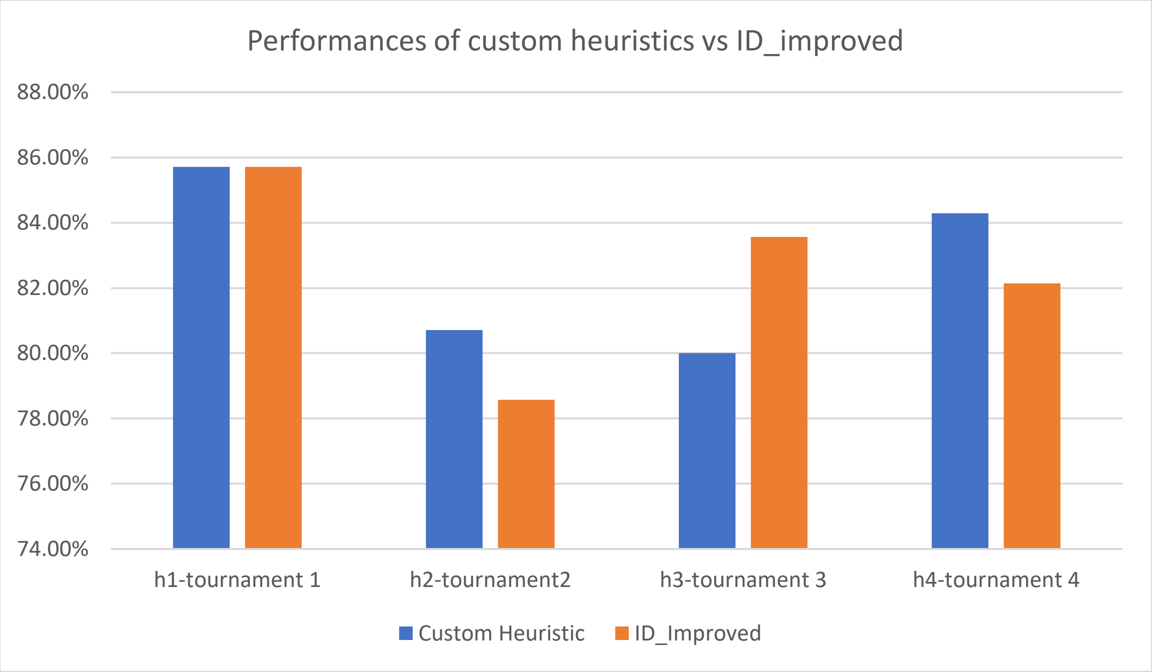
Evaluation of the performance of custom heuristics for Isolation game playing agent

Comparison of the different custom heuristic functions

Heuristic function name	Score	Score of ID_improved	Chosen?	Reason for choice
number_of_moves (h1)	85.71%	85.71%	No	Not better than ID_improved
number_of_moves_with_bias (h2)	80.71%	78.57%	No	Practically tied with the best function in terms of results but the chosen function has a slightly better performance
number_of_unique_moves (h3)	80.00%	83.57%	No	Performed less well than ID_improved
number_of_free_adjacent_cells (h4)	84.29%	82.14%	Yes	Best performing function relatively to ID_improved

Bar chart showing the performance of each custom heuristic vs ID_improved in the tournaments



Basic heuristic function: number_of_moves

This function calculates the difference between the number of moves left for each player and returns the result. It's a simple heuristic that is based on the intuition that the player with the most number of legal moves left is more likely to win.

For each player, the function calculates the number of legal moves left and then returns the different between the number of moves of the player and the opponent. Higher values mean that the player has more moves left than the opponent and is more likely to win.

Performance of the agent for the tournament test

This function performed as well as the ID_improved agent with an overall success rate of 85.71%. This result indicates that this is a good function however, a better heuristic could outperform ID_improved.

Custom heuristic function: number_of_moves_with_bias

This function uses the basic heuristic of the difference between the number of moves of the player and the opponent, with an added bias for the player. The player moves have a weight of 75% while the opponent moves have a weight of 25%.

For each player, this function takes the same formula as number_of_moves but returns a value that is more dependent on the player's number of move than the opponent's number of moves. The intuition behind this is, if it's the player's turn to play, then any move the player makes could affect the remaining number of moves of the opponent.

Performance of the agent for the tournament test

This function performed better than the ID_Improved agent with a score of 80.71% compared to 78.57% for ID_Improved. This shows that it's a better heuristic than number_of_moves based on the tournament result.

Custom heuristic function: number_of_unique_moves

This function is a version of the basic heuristic where moves shared by both players are not counted. This heuristic favors the player with the most unique moves. For each player, the total number of free moves is calculated then the number of moves that are common to the player and the opponent are subtracted. For the player, this means that only the moves that are not legal for the opponent are counted. The idea was to consider the cases where blocking is not possible.

Performance of the agent for the tournament test

This function performed less well than ID_improved with an 80.00% success rate compared to 83.57% for ID_improved. This heuristic has the worst performance of all three tested so far.

Custom heuristic function: number_of_free_adjacent_cells

This heuristic function is a version of the basic heuristic function which favors a player with the most adjacent cells that are free. For each player, this function counts not only the number of legal moves but also counts the ones that are adjacent cells meaning they are the 8 positions around the current position of the player. The number of free adjacent cells is added to the number of free moves for the player and the opponent. Consequently, the number of free adjacent cells is counted twice. The idea behind this is that the player with the highest number of free adjacent cells is more likely to win.

Performance of the agent for the tournament test

This function performed better than the ID_improved agent with a success rate of 84.29% compared to 82.14% for ID_improved. This heuristic performed practically as well as number_of_moves_with bias, compared to ID_improved.

Reasons for choosing this heuristic

I ultimately chose this as my final heuristic for the following reasons:

1. It performed better than ID_improved in the tournament
2. It takes at least 2 elements into consideration, adjacency and availability.
3. It produced better results than the other custom heuristics in the tournament. The difference between its performance and that of ID_improved in percentage is 2.15 while the difference between number_of_moves_with_bias and ID_improved is 2.14. (It is a very small difference and may not be significant. However, having to choose only one function I can use this result to favor number_of_frre_adjacent_cells over number_of_moves_with_bias.)

Appendix – Tournament results

Tournament test output for number_of_moves

```
*************************
 Evaluating: ID_Improved
*************************


Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 18 to 2
  Match 2: ID_Improved vs   MM_Null     Result: 16 to 4
  Match 3: ID_Improved vs   MM_Open     Result: 14 to 6
  Match 4: ID_Improved vs MM_Improved   Result: 12 to 8
  Match 5: ID_Improved vs   AB_Null     Result: 20 to 0
  Match 6: ID_Improved vs   AB_Open     Result: 20 to 0
  Match 7: ID_Improved vs AB_Improved   Result: 20 to 0



Results:
----------
ID_Improved         85.71%

*************************
   Evaluating: Student
*************************


Playing Matches:
----------
  Match 1:   Student   vs   Random      Result: 19 to 1
  Match 2:   Student   vs   MM_Null     Result: 16 to 4
  Match 3:   Student   vs   MM_Open     Result: 14 to 6
  Match 4:   Student   vs MM_Improved   Result: 11 to 9
  Match 5:   Student   vs   AB_Null     Result: 20 to 0
  Match 6:   Student   vs   AB_Open     Result: 20 to 0
  Match 7:   Student   vs AB_Improved   Result: 20 to 0



Results:
----------
```

```
Student          85.71%
```

Tournament test output for number_of_moves_with_bias

```
*************************
 Evaluating: ID_Improved
*************************


Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 16 to 4
  Match 2: ID_Improved vs   MM_Null     Result: 15 to 5
  Match 3: ID_Improved vs   MM_Open     Result: 12 to 8
  Match 4: ID_Improved vs MM_Improved   Result: 7 to 13
  Match 5: ID_Improved vs   AB_Null     Result: 20 to 0
  Match 6: ID_Improved vs   AB_Open     Result: 20 to 0
  Match 7: ID_Improved vs AB_Improved   Result: 20 to 0


Results:
----------
ID_Improved       78.57%

*************************
   Evaluating: Student
*************************


Playing Matches:
----------
  Match 1:   Student   vs   Random      Result: 17 to 3
  Match 2:   Student   vs   MM_Null     Result: 17 to 3
  Match 3:   Student   vs   MM_Open     Result: 9 to 11
  Match 4:   Student   vs MM_Improved   Result: 10 to 10
  Match 5:   Student   vs   AB_Null         Result: 20 to 0
  Match 6:   Student   vs   AB_Open      Result: 20 to 0
  Match 7:   Student   vs AB_Improved   Result: 20 to 0


Results:
----------
Student          80.71%
```

Tournament test output for number_of_unique_moves

```
*************************
 Evaluating: ID_Improved
*************************


Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 16 to 4
  Match 2: ID_Improved vs   MM_Null     Result: 14 to 6
  Match 3: ID_Improved vs   MM_Open     Result: 15 to 5
  Match 4: ID_Improved vs MM_Improved   Result: 12 to 8
  Match 5: ID_Improved vs   AB_Null     Result: 20 to 0
  Match 6: ID_Improved vs   AB_Open     Result: 20 to 0
  Match 7: ID_Improved vs AB_Improved   Result: 20 to 0


Results:
----------
ID_Improved        83.57%


*************************
   Evaluating: Student
*************************


Playing Matches:
----------
  Match 1:   Student   vs   Random        Result: 20 to 0
  Match 2:   Student   vs   MM_Null       Result: 15 to 5
  Match 3:   Student   vs   MM_Open       Result: 8 to 12
  Match 4:   Student   vs MM_Improved  Result: 9 to 11
  Match 5:   Student   vs   AB_Null         Result: 20 to 0
  Match 6:   Student   vs   AB_Open       Result: 20 to 0
  Match 7:   Student   vs AB_Improved   Result: 20 to 0

Results:
----------
Student           80.00%
```

Tournament test output for number_of_free_adjacent_cells

```
*************************
 Evaluating: ID_Improved
```

```
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random           Result: 17 to 3
  Match 2: ID_Improved vs   MM_Null          Result: 17 to 3
  Match 3: ID_Improved vs   MM_Open          Result: 13 to 7
  Match 4: ID_Improved vs MM_Improved   Result: 8 to 12
  Match 5: ID_Improved vs   AB_Null          Result: 20 to 0
  Match 6: ID_Improved vs   AB_Open          Result: 20 to 0
  Match 7: ID_Improved vs AB_Improved      Result: 20 to 0

Results:
----------
ID_Improved        82.14%


*************************
   Evaluating: Student
*************************

Playing Matches:
----------
  Match 1:   Student   vs  Random           Result: 18 to 2
  Match 2:   Student   vs  MM_Null          Result: 13 to 7
  Match 3:   Student   vs  MM_Open          Result: 14 to 6
  Match 4:   Student   vs  MM_Improved  Result: 13 to 7
  Match 5:   Student   vs  AB_Null          Result: 20 to 0
  Match 6:   Student   vs  AB_Open          Result: 20 to 0
  Match 7:   Student   vs AB_Improved     Result: 20 to 0


Results:
----------
Student           84.29%
```