Heuristic Analysis

By Akash Chauhan

We have tested 3 types of heuristic functions and find the custom1 to be working best among the heuristics.

1. custom\_score:

This heuristic is a combination of two heuristic function

a. center heuristic the player which is closer to the center can do better comparing to the case when he is far from the center

b. left legal moves: that states the condition of player comparing to the opponents 50% more moves.

Heuristic change point: when player is closer to the center than the opponent, if the player is far then the b heuristic comes into play.

# TODO: finish this function!

if game.is\_loser(player):

return float("-inf")

if game.is\_winner(player):

return float("inf")

own\_position = game.get\_player\_location(player)

opp\_position = game.get\_player\_location(game.get\_opponent(player))

# Heuristic 1

own\_distance\_from\_center = math.sqrt((own\_position[0] - game.width/2)\*\*2 + (own\_position[1] - game.height/2)\*\*2)

opp\_distance\_from\_center = math.sqrt((opp\_position[0] - game.width/2)\*\*2 + (opp\_position[1] - game.height/2)\*\*2)

if (own\_distance\_from\_center < opp\_distance\_from\_center):

return game.width - own\_distance\_from\_center

# Heuristic 2

own\_moves = len(game.get\_legal\_moves(player))

opp\_moves = len(game.get\_legal\_moves(game.get\_opponent(player)))

return float(own\_moves - 1.5 \* opp\_moves)

# raise NotImplementedError

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 20 to 0

Match 2: ID\_Improved vs MM\_Null Result: 16 to 4

Match 3: ID\_Improved vs MM\_Open Result: 6 to 14

Match 4: ID\_Improved vs MM\_Improved Result: 10 to 10

Match 5: ID\_Improved vs AB\_Null Result: 13 to 7

Match 6: ID\_Improved vs AB\_Open Result: 7 to 13

Match 7: ID\_Improved vs AB\_Improved Result: 7 to 13

Results:

----------

ID\_Improved 56.43%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 19 to 1

Match 2: Student vs MM\_Null Result: 14 to 6

Match 3: Student vs MM\_Open Result: 13 to 7

Match 4: Student vs MM\_Improved Result: 9 to 11

Match 5: Student vs AB\_Null Result: 11 to 9

Match 6: Student vs AB\_Open Result: 10 to 10

Match 7: Student vs AB\_Improved Result: 7 to 13

Results:

----------

Student 59.29%

2. custom\_score\_2:

Center Heuristic: it states that the player near to the center has better chances to win the game.

def custom\_score\_2(game, player):

if game.is\_loser(player):

return float("-inf")

if game.is\_winner(player):

return float("inf")

own\_position = game.get\_player\_location(player)

opp\_position = game.get\_player\_location(game.get\_opponent(player))

# Center Heuristic

own\_distance\_from\_center = math.sqrt((own\_position[0] - game.width/2)\*\*2 + (own\_position[1] - game.height/2)\*\*2)

opp\_distance\_from\_center = math.sqrt((opp\_position[0] - game.width/2)\*\*2 + (opp\_position[1] - game.height/2)\*\*2)

return game.width - own\_distance\_from\_center

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 19 to 1

Match 2: ID\_Improved vs MM\_Null Result: 15 to 5

Match 3: ID\_Improved vs MM\_Open Result: 8 to 12

Match 4: ID\_Improved vs MM\_Improved Result: 10 to 10

Match 5: ID\_Improved vs AB\_Null Result: 11 to 9

Match 6: ID\_Improved vs AB\_Open Result: 8 to 12

Match 7: ID\_Improved vs AB\_Improved Result: 7 to 13

Results:

----------

ID\_Improved 55.71%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 20 to 0

Match 2: Student vs MM\_Null Result: 16 to 4

Match 3: Student vs MM\_Open Result: 10 to 10

Match 4: Student vs MM\_Improved Result: 9 to 11

Match 5: Student vs AB\_Null Result: 8 to 12

Match 6: Student vs AB\_Open Result: 6 to 14

Match 7: Student vs AB\_Improved Result: 5 to 15

Results:

----------

Student 52.86%

custom\_score\_3:

Legal moves left heuristic: this one is intuitive, one with the more moves available has more chances to win.

def custom\_score\_3(game, player):

if game.is\_loser(player):

return float("-inf")

if game.is\_winner(player):

return float("inf")

# 2

own\_moves = len(game.get\_legal\_moves(player))

opp\_moves = len(game.get\_legal\_moves(game.get\_opponent(player)))

return float(own\_moves - 2 \* opp\_moves)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: ID\_Improved

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: ID\_Improved vs Random Result: 20 to 0

Match 2: ID\_Improved vs MM\_Null Result: 20 to 0

Match 3: ID\_Improved vs MM\_Open Result: 8 to 12

Match 4: ID\_Improved vs MM\_Improved Result: 10 to 10

Match 5: ID\_Improved vs AB\_Null Result: 10 to 10

Match 6: ID\_Improved vs AB\_Open Result: 9 to 11

Match 7: ID\_Improved vs AB\_Improved Result: 7 to 13

Results:

----------

ID\_Improved 60.00%

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Evaluating: Student

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Playing Matches:

----------

Match 1: Student vs Random Result: 19 to 1

Match 2: Student vs MM\_Null Result: 14 to 6

Match 3: Student vs MM\_Open Result: 11 to 9

Match 4: Student vs MM\_Improved Result: 8 to 12

Match 5: Student vs AB\_Null Result: 11 to 9

Match 6: Student vs AB\_Open Result: 9 to 11

Match 7: Student vs AB\_Improved Result: 4 to 16

Results:

----------

Student 54.29%