Heuristic Analysis

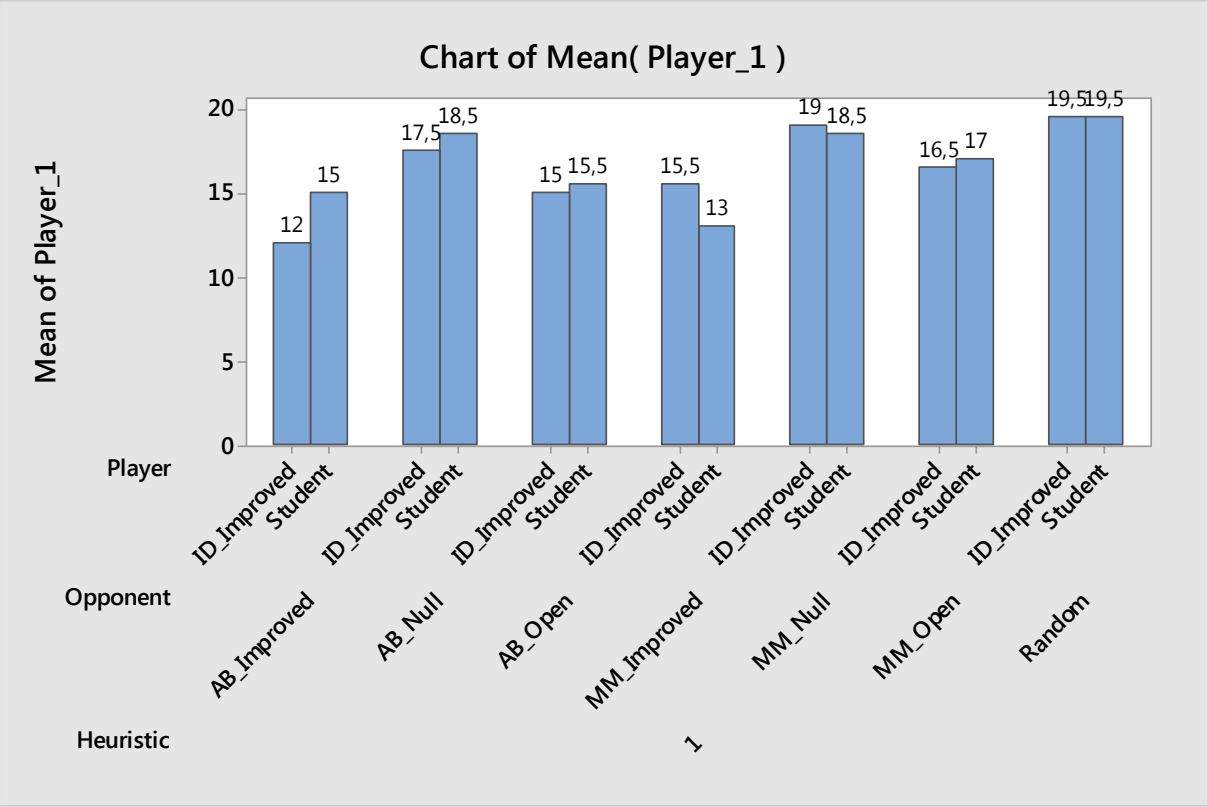
Heuristic 1 – Offensive and select same moves when having more options

Description: It is offensive(regarding number of opponent moves much more) and if number of my player's move options is higher, it is keen to select shared moves with opponent as next action.

Code:

```
a = 1.0
b = 2.5
c = 50.0

if my_moves>opp_moves:
    return a*my_moves - b*opp_moves + c*same_moves
else:
    return a*my_moves – b*opp_moves,
```

Result for Heuristic 1:



Heuristic 2 – Offensive and Hate-Edges

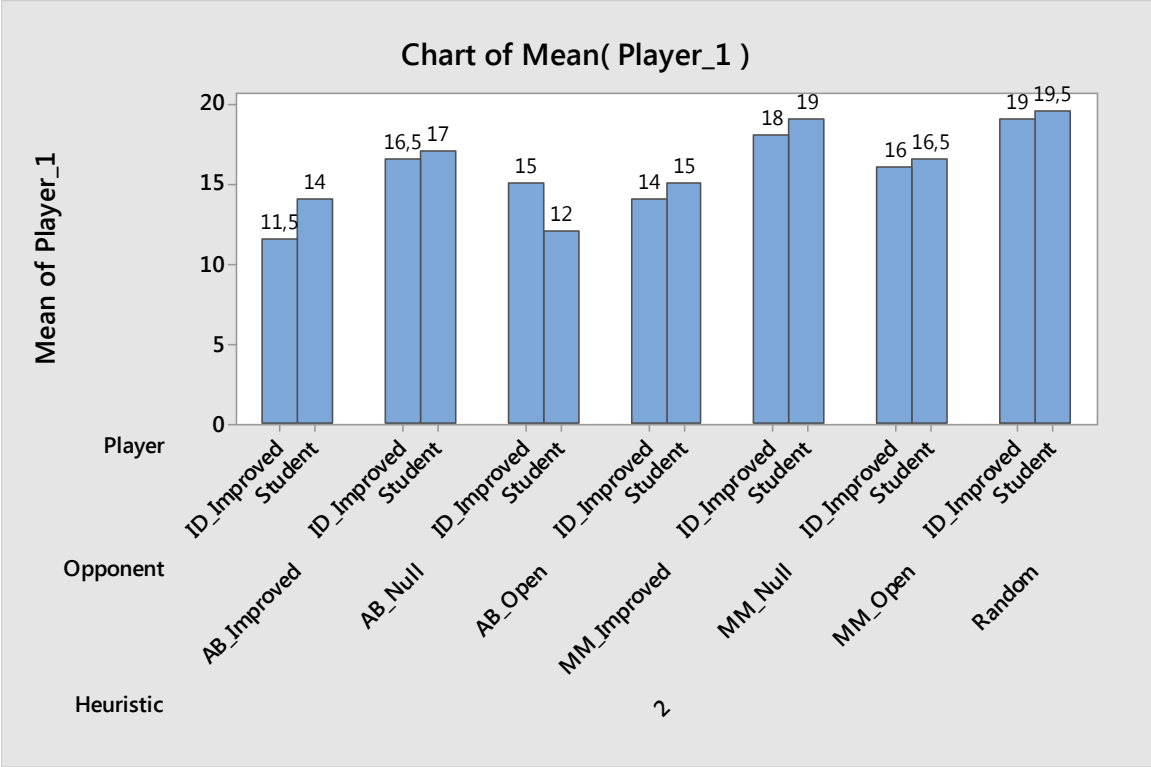
Description: It is offensive and it is prevent my player to go edges when opponent is more likely to go edges.

Code:

```
a = 1.0
b = 2.5
```

```
    if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):
        my_edge_factor = -10
    if (row_o == 0 or row_o == (game.height-1) or column_o == 0 or column_o == (game.width-1)):
        opp_edge_factor = 10

    return a*my_moves – b*opp_moves+my_edge_factor+opp_edge_factor
```

Result for Heuristic 2:



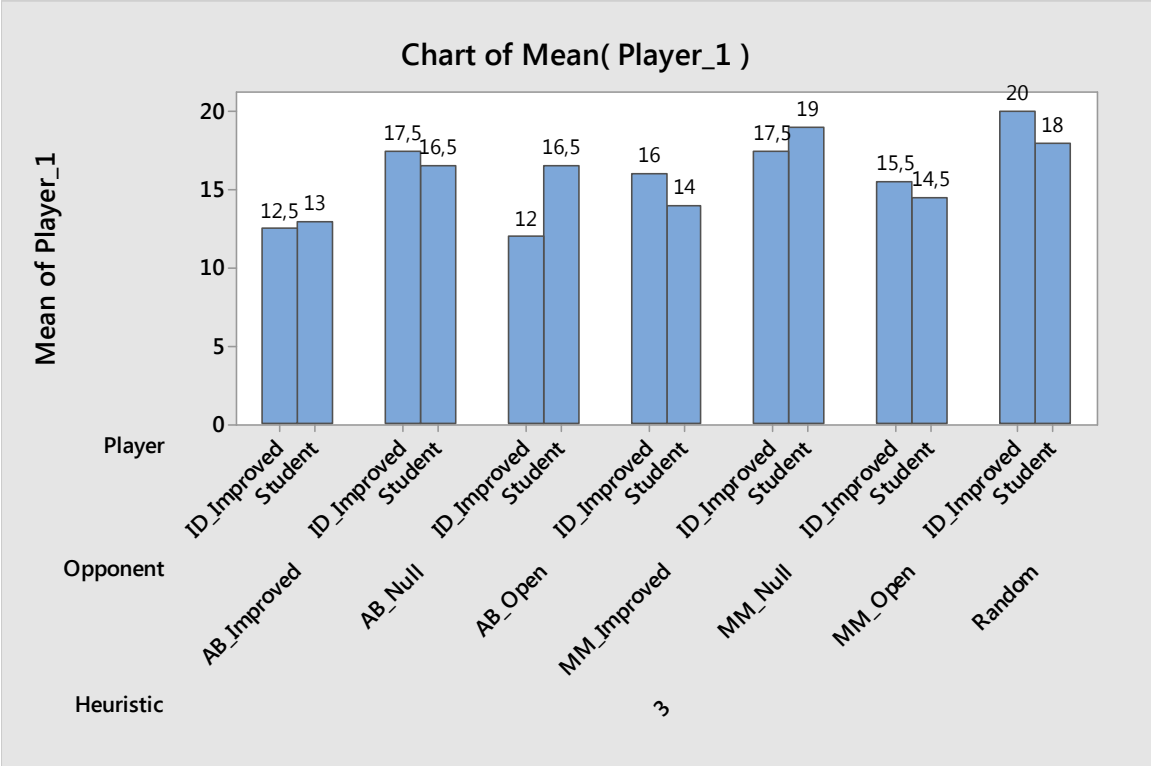
Heuristic 3 – Offensive and Hate-Corners

Description: It is offensive and it is prevent my player to go corners when opponent is more likely to go corners.

Code:

```
    a = 1.0
    b = 2.5
    if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):
        my_corner_factor = -10
    if ((row_o == 0 or row_o == (game.height-1)) and (column_o == 0 or column_o ==
(game.width-1))):
        my_corner_factor = 10

    return a*my_moves – b*opp_moves+my_corner_factor+opp_corner_factor
```

Result for Heuristic 3:



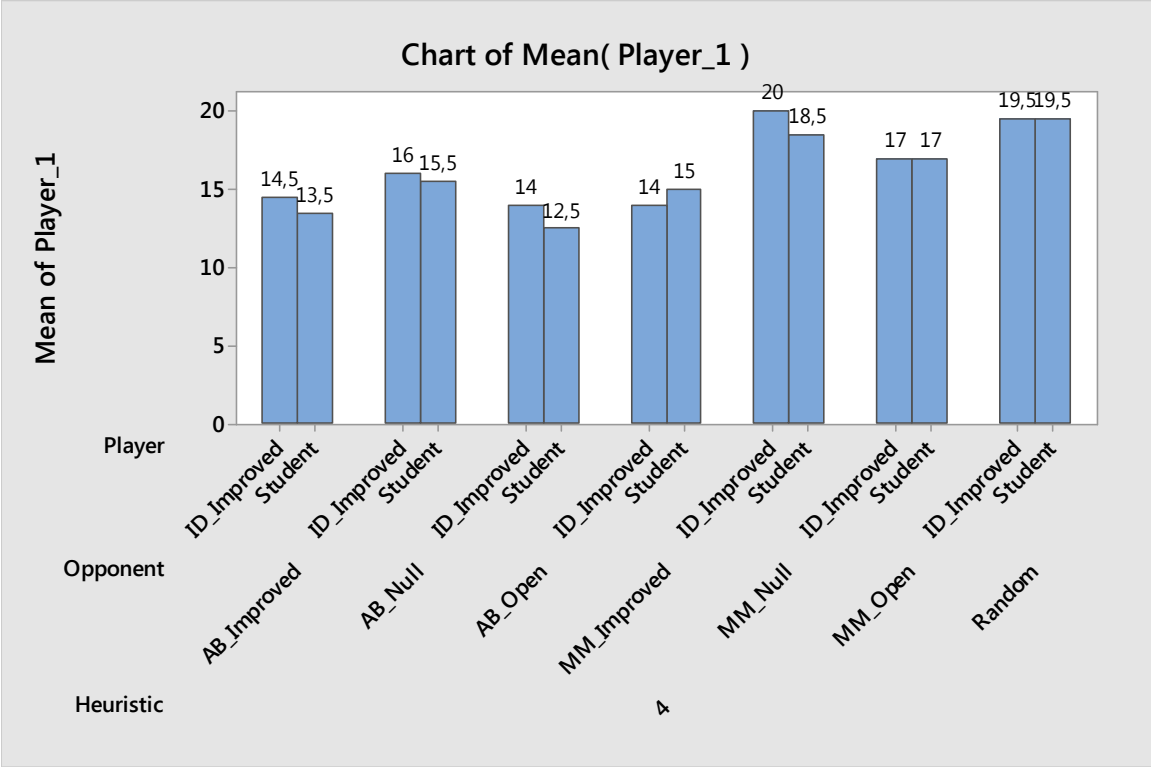
Heuristic 4 – Game Progress Monitored(2-part) Offensive, Same Moves and Hate-Edges

Description: It is offensive. Importance of same movement is not significant at first.In progress of game, it becomes important. Also preventing my player to go edges when opponent is more likely to go edges is more important at start and becomes less important over time. Game is regarded 2 parts.

Code:

```
a = 1
b = 3
c = 3

if game_progress >= 0.15:

    c = 50.0*game_progress

if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):
    my_edge_factor = -20.0
if (row_o == 0 or row_o == (game.height-1) or column_o == 0 or column_o == (game.width-1)):
    opp_edge_factor = 20 .0

return a*my_moves - b*opp_moves +c*same_moves+my_edge_factor+opp_edge_factor
```

Result for Heuristic 4:



Heuristic 5 - Game Progress Monitored(3-part) Increasing Offensive and Hate-Edges Like-Centered

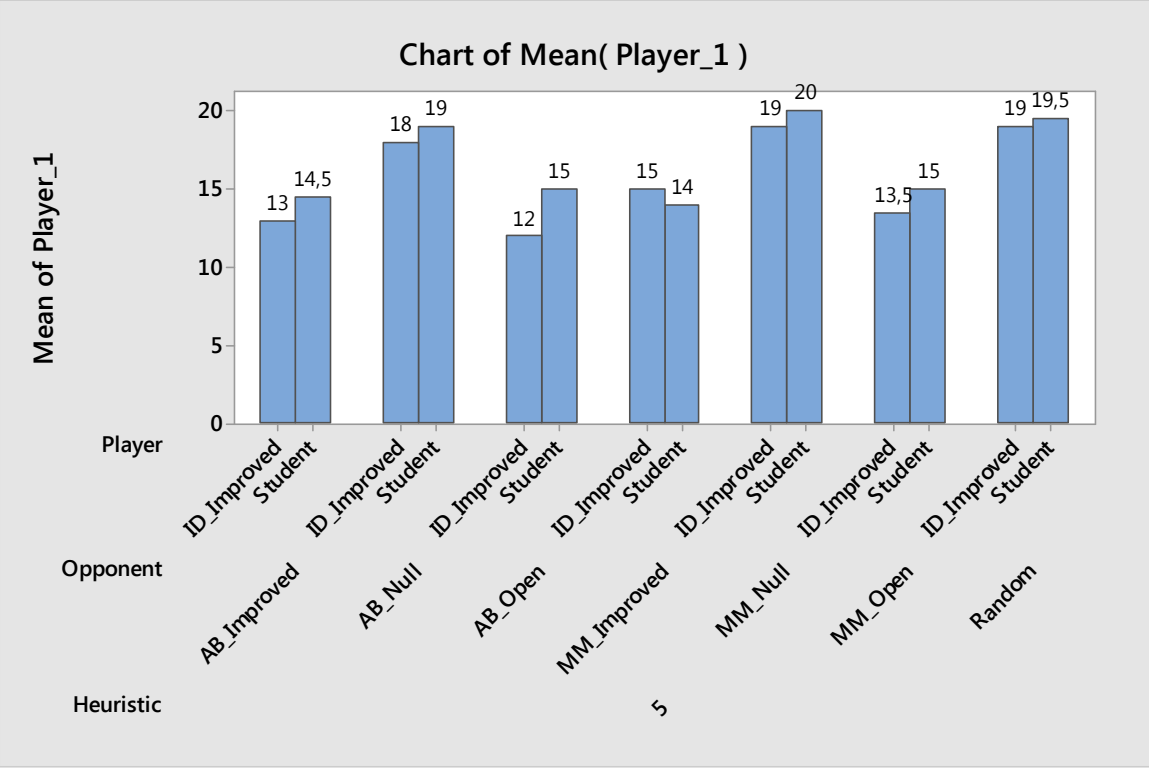
Description: It is offensive at first. Game is regarded 3 parts. Importance of center factor is decreasing with progress,in first part. In second part, preventive edge factor become important with progress.
Offensiveness decreased sharply at the beginning of the part and then increase with time. At the last part, more balanced.

Code:

```
    if game_progress <= 0.1:
        a=1.0
        b=3.0
        c= 1
        if (row == (game.height/2) or row == (game.height/2+1) or row == (game.height/2-1)) and
(column == (game.width/2) or column == (game.width/2+1) or column == (game.width/2-1)):
            my_center_factor = 20.0*(1.0/game_progress**2)
        if (row_o == (game.height/2) or row_o == (game.height/2+1) or row_o == (game.height/2-1))
and (column_o == (game.width/2) or column_o == (game.width/2+1) or column_o ==
(game.width/2-1)):
            opp_center_factor = -20.0*(1.0/game_progress**2)
    elif game_progress <= 0.40:
```

```
        a=1.0
        b=1.0*4.0*(game_progress)*(opp_moves/(my_moves+0.1)) #4.0,10.0
        c=80*(game_progress**2)
        if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):
            my_edge_factor = -50.0 * (game_progress**2) #-500
        if (row_o == 0 or row_o == (game.height-1) or column_o == 0 or column_o == (game.width-
1)):
            opp_edge_factor = 50.0 * (game_progress**2) #500

    else:

        a=5.0
        b=5.0
        c=1.0
        if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):
            my_edge_factor = -300.0 * (game_progress**2)
        if (row_o == 0 or row_o == (game.height-1) or column_o == 0 or column_o == (game.width-
1)):
            opp_edge_factor = 300.0 * (game_progress**2)



    return a*my_moves - b*opp_moves
+c*same_moves+my_edge_factor+opp_edge_factor+my_center_factor+opp_center_factor
```

Result for Heuristic 5:



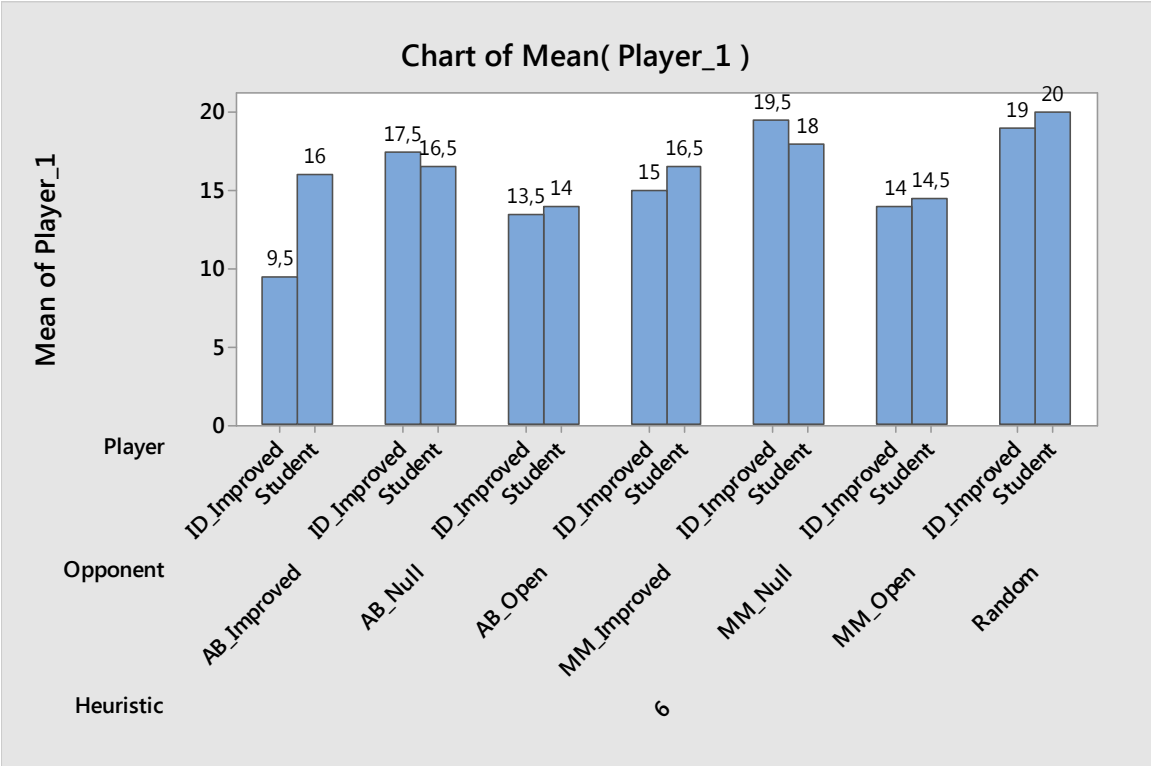
Heuristic 6- Game Progress Monitored(3-part) Increasing Offensive and Hate-Edges Like-Centered 2

Description: It looks like as Heuristic 5 but not offensive at first. Getting more offensive with time. Some coeffients and also progress frames are different.

Code:

```
        if game_progress <= 0.1:
        a=1.0
        b=1.0
        c= 1
        if (row == (game.height/2) or row == (game.height/2+1) or row == (game.height/2-1)) and
(column == (game.width/2) or column == (game.width/2+1) or column == (game.width/2-1)):
            my_center_factor = 20.0*(1.0/game_progress)
            opp_center_factor = -20.0*(1.0/game_progress**2)
    elif game_progress <= 0.25:
        a=1.0
        b=1.0*10.0*(game_progress)
        c=800*(game_progress**2)
        if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):
            my_edge_factor = -500.0 * (game_progress**2)
        if (row_o == 0 or row_o == (game.height-1) or column_o == 0 or column_o == (game.width-
1)):
            opp_edge_factor = 500.0 * (game_progress**2)


    else:
        a=1.0
        b=2.5
        c=800.0*(game_progress**2)
        if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):#
            my_edge_factor = -300.0 * (game_progress**2)
        if (row_o == 0 or row_o == (game.height-1) or column_o == 0 or column_o == (game.width-
1)):
            opp_edge_factor = 300.0 * (game_progress**2)

    return a*my_moves - b*opp_moves +c*same_moves #+
my_edge_factor+opp_edge_factor+my_center_factor
```

Result for Heuristic 6:



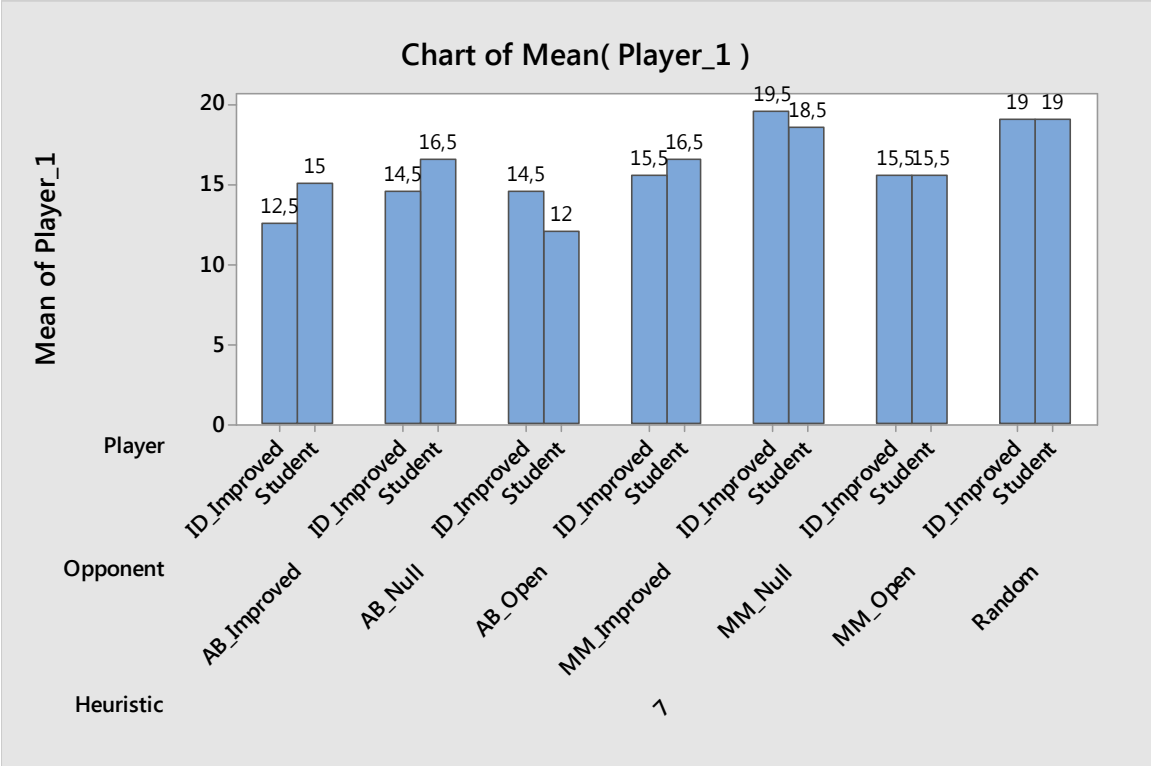
Heuristic 7 – Dynamic Opponent Moves Coefficient with Additive Same Movement Coefficient

Description:
Conditional offensive. Same move coefficient change with position of move.(edge,center)

Code:

```
   my_center_factor = 1.0
    my_edge_factor=1.0
    a=1.0
    k=0.0
    b=3.0 * (opp_moves/(my_moves+0.1))
    c=1
    if game_progress >= 0.4:
        k = 20.0  #additive importance for shared moves
    if (row == (game.height/2) or row == (game.height/2+1) or row == (game.height/2-1)) and
(column == (game.width/2) or column == (game.width/2+1) or column == (game.width/2-1)):
        my_center_factor = 2.0
    if (row == 0 or row == (game.height-1) or column == 0 or column == (game.width-1)):
        my_edge_factor = -2.0
    return a*my_moves - b*opp_moves +(c*my_edge_factor*my_center_factor+k)*same_moves
```

Result for Heuristic 7:



SUMMARY RESULT

Heuristic	Mean of Student Win Game	Mean of Improved Win Game
1	16,143	15,857
2	16,143	15,714
3	15,929	15,857
4	15,929	16,429
5	16,714	15,643
6	16,500	15,429
7	16,143	15,857

CONCLUSION

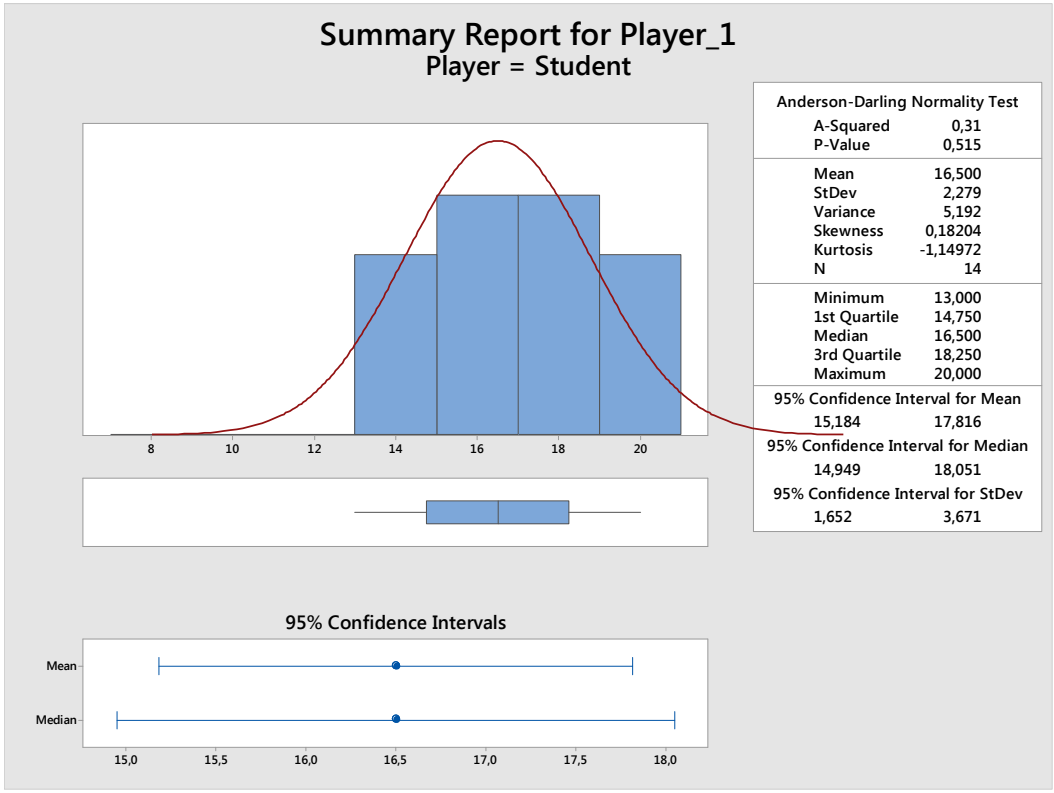
Heuristic 5 and **Heuristic 6** are succesful according to results. **Heuristic 5** wins 6 of 7 opponents.

1- Seperating by defining different conditions for one game helps to gain success.
2- Increase or decrease offensiveness during the game is also advantage.
3. Dynamic coefficient according to move number helps to create more flexible agent.

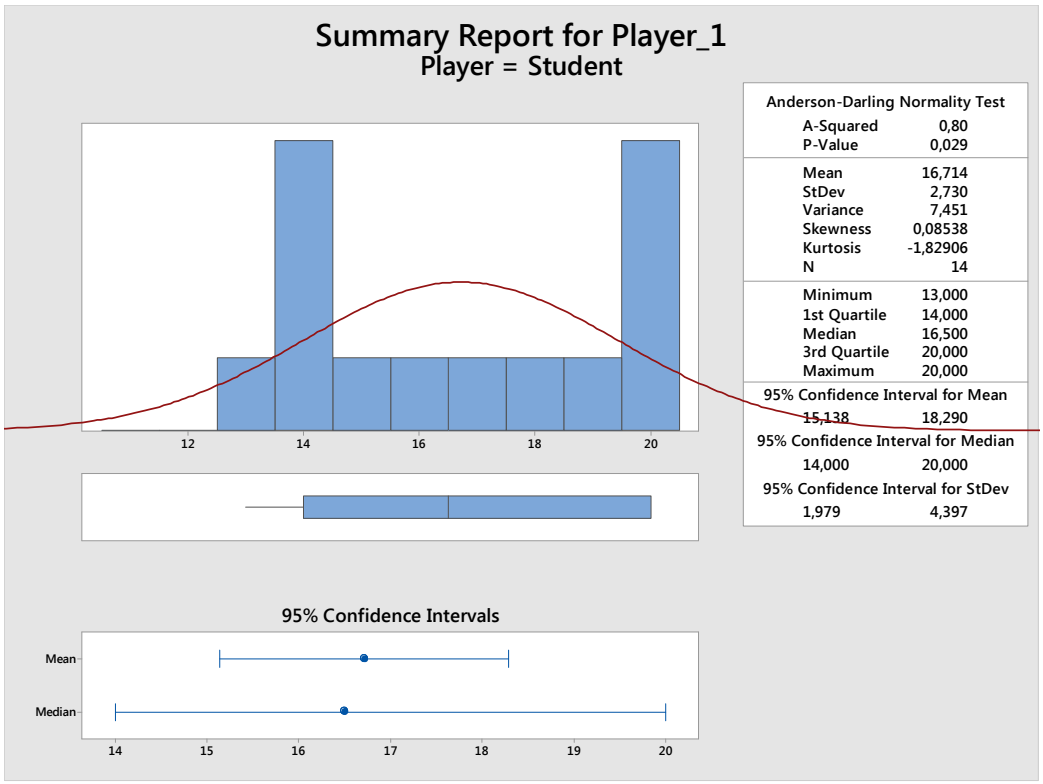
They are not statistically different from each other according to t-test. However I prefered to choose **Heuristic 6** due to the fact that its number of wins is normal distrubuted and more balanced.

APPENDIX. DETAILED RESULTS

Heuristic 6 Student:



Heuristic 5 Student:



Heuristic	Iteration	Player		Opponent		Player	Opponent	Result
1	1	Student	vs	Random	Result:	19 to	1	
1	1	Student	vs	MM_Null	Result:	20 to	0	
1	1	Student	vs	MM_Open	Result:	17 to	3	
1	1	Student	vs	MM_Improved	Result:	14 to	6	83.57%
1	1	Student	vs	AB_Null	Result:	19 to	1	
1	1	Student	vs	AB_Open	Result:	14 to	6	
1	1	Student	vs	AB_Improved	Result:	14 to	6	
1	1	ID_Improved	vs	Random	Result:	19 to	1	
1	1	ID_Improved	vs	MM_Null	Result:	18 to	2	
1	1	ID_Improved	vs	MM_Open	Result:	17 to	3	
1	1	ID_Improved	vs	MM_Improved	Result:	15 to	5	80.71%
1	1	ID_Improved	vs	AB_Null	Result:	17 to	3	
1	1	ID_Improved	vs	AB_Open	Result:	14 to	6	
1	1	ID_Improved	vs	AB_Improved	Result:	13 to	7	
1	2	Student	vs	Random	Result:	20 to	0	
1	2	Student	vs	MM_Null	Result:	17 to	3	
1	2	Student	vs	MM_Open	Result:	17 to	3	
1	2	Student	vs	MM_Improved	Result:	12 to	8	83.57%
1	2	Student	vs	AB_Null	Result:	18 to	2	
1	2	Student	vs	AB_Open	Result:	17 to	3	
1	2	Student	vs	AB_Improved	Result:	16 to	4	
1	2	ID_Improved	vs	Random	Result:	20 to	0	
1	2	ID_Improved	vs	MM_Null	Result:	20 to	0	
1	2	ID_Improved	vs	MM_Open	Result:	16 to	4	
1	2	ID_Improved	vs	MM_Improved	Result:	16 to	4	83.57%
1	2	ID_Improved	vs	AB_Null	Result:	18 to	2	
1	2	ID_Improved	vs	AB_Open	Result:	16 to	4	
1	2	ID_Improved	vs	AB_Improved	Result:	11 to	9	
2	1	Student	vs	Random	Result:	19 to	1	
2	1	Student	vs	MM_Null	Result:	18 to	2	
2	1	Student	vs	MM_Open	Result:	19 to	1	
2	1	Student	vs	MM_Improved	Result:	12 to	8	80.71%
2	1	Student	vs	AB_Null	Result:	18 to	2	
2	1	Student	vs	AB_Open	Result:	13 to	7	
2	1	Student	vs	AB_Improved	Result:	14 to	6	
2	1	ID_Improved	vs	Random	Result:	20 to	0	
2	1	ID_Improved	vs	MM_Null	Result:	17 to	3	
2	1	ID_Improved	vs	MM_Open	Result:	15 to	5	
2	1	ID_Improved	vs	MM_Improved	Result:	14 to	6	78.57%
2	1	ID_Improved	vs	AB_Null	Result:	17 to	3	
2	1	ID_Improved	vs	AB_Open	Result:	15 to	5	
2	1	ID_Improved	vs	AB_Improved	Result:	12 to	8	
2	2	Student	vs	Random	Result:	20 to	0	80.71%
2	2	Student	vs	MM_Null	Result:	20 to	0	
2	2	Student	vs	MM_Open	Result:	14 to	6	
2	2	Student	vs	MM_Improved	Result:	18 to	2	
2	2	Student	vs	AB_Null	Result:	16 to	4	
2	2	Student	vs	AB_Open	Result:	11 to	9	

2	2	Student	vs	AB_Improved	Result:	14	to	6
2	2	ID_Improved	vs	Random	Result:	18	to	2
2	2	ID_Improved	vs	MM_Null	Result:	19	to	1
2	2	ID_Improved	vs	MM_Open	Result:	17	to	3
2	2	ID_Improved	vs	MM_Improved	Result:	14	to	6 78.57%
2	2	ID_Improved	vs	AB_Null	Result:	16	to	4
2	2	ID_Improved	vs	AB_Open	Result:	15	to	5
2	2	ID_Improved	vs	AB_Improved	Result:	11	to	9
3	1	Student	vs	Random	Result:	18	to	2
3	1	Student	vs	MM_Null	Result:	20	to	0
3	1	Student	vs	MM_Open	Result:	15	to	5
3	1	Student	vs	MM_Improved	Result:	16	to	4 81.43%
3	1	Student	vs	AB_Null	Result:	17	to	3
3	1	Student	vs	AB_Open	Result:	16	to	4
3	1	Student	vs	AB_Improved	Result:	12	to	8
3	1	ID_Improved	vs	Random	Result:	20	to	0
3	1	ID_Improved	vs	MM_Null	Result:	16	to	4
3	1	ID_Improved	vs	MM_Open	Result:	17	to	3
3	1	ID_Improved	vs	MM_Improved	Result:	16	to	4 81.43%
3	1	ID_Improved	vs	AB_Null	Result:	19	to	1
3	1	ID_Improved	vs	AB_Open	Result:	13	to	7
3	1	ID_Improved	vs	AB_Improved	Result:	13	to	7
3	2	Student	vs	Random	Result:	18	to	2
3	2	Student	vs	MM_Null	Result:	18	to	2
3	2	Student	vs	MM_Open	Result:	14	to	6
3	2	Student	vs	MM_Improved	Result:	12	to	8 77.86%
3	2	Student	vs	AB_Null	Result:	16	to	4
3	2	Student	vs	AB_Open	Result:	17	to	3
3	2	Student	vs	AB_Improved	Result:	14	to	6
3	2	ID_Improved	vs	Random	Result:	20	to	0
3	2	ID_Improved	vs	MM_Null	Result:	19	to	1
3	2	ID_Improved	vs	MM_Open	Result:	14	to	6
3	2	ID_Improved	vs	MM_Improved	Result:	16	to	4 77.14%
3	2	ID_Improved	vs	AB_Null	Result:	16	to	4
3	2	ID_Improved	vs	AB_Open	Result:	11	to	9
3	2	ID_Improved	vs	AB_Improved	Result:	12	to	8
4	1	Student	vs	Random	Result:	20	to	0 78.57%
4	1	Student	vs	MM_Null	Result:	18	to	2
4	1	Student	vs	MM_Open	Result:	17	to	3
4	1	Student	vs	MM_Improved	Result:	14	to	6
4	1	Student	vs	AB_Null	Result:	15	to	5
4	1	Student	vs	AB_Open	Result:	14	to	6
4	1	Student	vs	AB_Improved	Result:	12	to	8
4	1	ID_Improved	vs	Random	Result:	19	to	1
4	1	ID_Improved	vs	MM_Null	Result:	20	to	0
4	1	ID_Improved	vs	MM_Open	Result:	15	to	5
4	1	ID_Improved	vs	MM_Improved	Result:	14	to	6 81.43%
4	1	ID_Improved	vs	AB_Null	Result:	16	to	4
4	1	ID_Improved	vs	AB_Open	Result:	14	to	6
4	1	ID_Improved	vs	AB_Improved	Result:	16	to	4
4	1	Student	vs	Random	Result:	19	to	1 80.71%
4	1	Student	vs	MM_Null	Result:	19	to	1

4	1 Student	vs	MM_Open	Result:	17 to	3	
4	1 Student	vs	MM_Improved	Result:	16 to	4	
4	1 Student	vs	AB_Null	Result:	16 to	4	
4	1 Student	vs	AB_Open	Result:	11 to	9	
4	1 Student	vs	AB_Improved	Result:	15 to	5	
4	1 ID_Improved	vs	Random	Result:	20 to	0	
4	1 ID_Improved	vs	MM_Null	Result:	20 to	0	
4	1 ID_Improved	vs	MM_Open	Result:	19 to	1	
4	1 ID_Improved	vs	MM_Improved	Result:	14 to	6	82.86%
4	1 ID_Improved	vs	AB_Null	Result:	16 to	4	
4	1 ID_Improved	vs	AB_Open	Result:	14 to	6	
4	1 ID_Improved	vs	AB_Improved	Result:	13 to	7	
5	1 Student	vs	Random	Result:	19 to	1	
5	1 Student	vs	MM_Null	Result:	20 to	0	
5	1 Student	vs	MM_Open	Result:	14 to	6	
5	1 Student	vs	MM_Improved	Result:	14 to	6	83.57%
5	1 Student	vs	AB_Null	Result:	18 to	2	
5	1 Student	vs	AB_Open	Result:	17 to	3	
5	1 Student	vs	AB_Improved	Result:	15 to	5	
5	1 ID_Improved	vs	Random	Result:	19 to	1	
5	1 ID_Improved	vs	MM_Null	Result:	18 to	2	
5	1 ID_Improved	vs	MM_Open	Result:	15 to	5	
5	1 ID_Improved	vs	MM_Improved	Result:	15 to	5	79.29%
5	1 ID_Improved	vs	AB_Null	Result:	18 to	2	
5	1 ID_Improved	vs	AB_Open	Result:	13 to	7	
5	1 ID_Improved	vs	AB_Improved	Result:	13 to	7	
5	2 Student	vs	Random	Result:	20 to	0	
5	2 Student	vs	MM_Null	Result:	20 to	0	
5	2 Student	vs	MM_Open	Result:	16 to	4	
5	2 Student	vs	MM_Improved	Result:	14 to	6	83.57%
5	2 Student	vs	AB_Null	Result:	20 to	0	
5	2 Student	vs	AB_Open	Result:	13 to	7	
5	2 Student	vs	AB_Improved	Result:	14 to	6	
5	2 ID_Improved	vs	Random	Result:	19 to	1	
5	2 ID_Improved	vs	MM_Null	Result:	20 to	0	
5	2 ID_Improved	vs	MM_Open	Result:	12 to	8	
5	2 ID_Improved	vs	MM_Improved	Result:	15 to	5	77.14%
5	2 ID_Improved	vs	AB_Null	Result:	18 to	2	
5	2 ID_Improved	vs	AB_Open	Result:	11 to	9	
5	2 ID_Improved	vs	AB_Improved	Result:	13 to	7	
6	1 Student	vs	Random	Result:	20 to	0	
6	1 Student	vs	MM_Null	Result:	19 to	1	
6	1 Student	vs	MM_Open	Result:	15 to	5	
6	1 Student	vs	MM_Improved	Result:	16 to	4	82.14%
6	1 Student	vs	AB_Null	Result:	18 to	2	
6	1 Student	vs	AB_Open	Result:	13 to	7	
6	1 Student	vs	AB_Improved	Result:	14 to	6	
6	1 ID_Improved	vs	Random	Result:	18 to	2	76.43%
6	1 ID_Improved	vs	MM_Null	Result:	19 to	1	
6	1 ID_Improved	vs	MM_Open	Result:	15 to	5	
6	1 ID_Improved	vs	MM_Improved	Result:	16 to	4	
6	1 ID_Improved	vs	AB_Null	Result:	16 to	4	

6	1	ID_Improved	vs	AB_Open	Result:	15	to	5
6	1	ID_Improved	vs	AB_Improved	Result:	8	to	12
6	2	Student	vs	Random	Result:	20	to	0
6	2	Student	vs	MM_Null	Result:	17	to	3
6	2	Student	vs	MM_Open	Result:	14	to	6
6	2	Student	vs	MM_Improved	Result:	17	to	3 82.86%
6	2	Student	vs	AB_Null	Result:	15	to	5
6	2	Student	vs	AB_Open	Result:	15	to	5
6	2	Student	vs	AB_Improved	Result:	18	to	2
6	2	ID_Improved	vs	Random	Result:	20	to	0
6	2	ID_Improved	vs	MM_Null	Result:	20	to	0
6	2	ID_Improved	vs	MM_Open	Result:	13	to	7
6	2	ID_Improved	vs	MM_Improved	Result:	14	to	6 77.86%
6	2	ID_Improved	vs	AB_Null	Result:	19	to	1
6	2	ID_Improved	vs	AB_Open	Result:	12	to	8
6	2	ID_Improved	vs	AB_Improved	Result:	11	to	9
7	1	Student	vs	Random	Result:	19	to	1
7	1	Student	vs	MM_Null	Result:	18	to	2
7	1	Student	vs	MM_Open	Result:	15	to	5
7	1	Student	vs	MM_Improved	Result:	17	to	3 80.71%
7	1	Student	vs	AB_Null	Result:	17	to	3
7	1	Student	vs	AB_Open	Result:	11	to	9
7	1	Student	vs	AB_Improved	Result:	16	to	4
7	1	ID_Improved	vs	Random	Result:	18	to	2
7	1	ID_Improved	vs	MM_Null	Result:	20	to	0
7	1	ID_Improved	vs	MM_Open	Result:	15	to	5
7	1	ID_Improved	vs	MM_Improved	Result:	16	to	4 76.43%
7	1	ID_Improved	vs	AB_Null	Result:	15	to	5
7	1	ID_Improved	vs	AB_Open	Result:	13	to	7
7	1	ID_Improved	vs	AB_Improved	Result:	10	to	10
7	2	Student	vs	Random	Result:	19	to	1
7	2	Student	vs	MM_Null	Result:	19	to	1
7	2	Student	vs	MM_Open	Result:	16	to	4
7	2	Student	vs	MM_Improved	Result:	16	to	4 80.71%
7	2	Student	vs	AB_Null	Result:	16	to	4
7	2	Student	vs	AB_Open	Result:	13	to	7
7	2	Student	vs	AB_Improved	Result:	14	to	6
7	2	ID_Improved	vs	Random	Result:	20	to	0
7	2	ID_Improved	vs	MM_Null	Result:	19	to	1
7	2	ID_Improved	vs	MM_Open	Result:	16	to	4
7	2	ID_Improved	vs	MM_Improved	Result:	15	to	5 82.14%
7	2	ID_Improved	vs	AB_Null	Result:	14	to	6
7	2	ID_Improved	vs	AB_Open	Result:	16	to	4
7	2	ID_Improved	vs	AB_Improved	Result:	15	to	5