Heuristic analysis

One of the most difficult parts about game playing algorithms is designing a good evaluation function. In this brief analysis I will go through the evaluation functions used in designing a game playing agent for the board game Isolation.

$$v(s) = MyMoves - (3 * OpponentMoves)$$

This is function is an extension to the function mentioned in the lecture in which the opponent's available moves were multiplied by 2 and subtracted from the agent's own moves, $v(s) = MyMoves - (2 * OpponentMoves)$. Multiplying the opponent's moves by 3 increases the emphasis on chasing the opponent. This function was chosen because it performed better than the function mentioned in the lecture.

$$v(s) = MyMoves - (4 * OpponentMoves)$$

This function is not much different from the one above but it merely increases the pressure to chase the opponent. This function was also chosen because of performance shown in tests.

$$v(s) = MyMoves - (OpponentMoves^2)$$

In the this evaluation function the score is calculated by subtracting the number of moves the opponent has left squared from the number of moves of the agent. The idea behind designing this function is to put a stronger emphasis on decreasing the moves available to the opponent than multiplying it by a constant. which would lead to a behaviour similar to chasing the opponent and trying to corner it.

Results:

All 3 of the evaluation functions mentioned above outperform ID_improved $(MyMoves - OpponentMoves)$ consistently. The results from the tournaments are shown in table 1.

Table 1: this table shows the results of one competition with each pair playing a 100 matches between each other. For brevity "my moves" is abbreviated in the table as "mm", which represents the number of legal moves left for the agent, and "opponent moves" is abbreviated to "om" which represents the number of legal moves left for the opponent.

# Matches	Opponent	AB_improved	$mm - (3 * om)$	$mm - (4 * om)$	$mm - om^2$
	Won | Lost	Won | Lost	Won | Lost	Won | Lost	Won | Lost
1	Random	89 | 11	87 | 13	92 | 8	92 | 8
2	MM_Open	61 | 39	55 | 45	60 | 40	59 | 41
3	MM_Center	84 | 16	83 | 17	89 | 11	81 | 19
4	MM_Improved	54 | 46	57 | 43	67 | 33	61 | 39
5	AB_Open	48 | 52	52 | 48	59 | 41	51 | 49
6	AB_Center	46 | 54	49 | 51	54 | 46	48 | 52
7	AB_Improved	48 | 52	51 | 49	47 | 53	53 | 47
	Win Rate:	61.4%	62.0%	66.9%	63.6%

Table 1 demonstrates the strength of the 3 evaluation functions compared to the AB_improved evaluation function. Table 1 only shows the results of 1 tournament. Even though a 100 matches are played between each pair of agents, 1 competition is not enough to judge the strength of the evaluation functions. Therefore more competitions were conducted and their results are shown in a single table (table 2) with only the win rate for brevity.

Table 2: shown above are the wine rate from each tournament

# Tournament	AB_improved	$mm - (3 * om)$	$mm - (4 * om)$	$mm - om^2$
1	61.1%	61.9%	64.6%	63.6%
2	60.9%	64.0%	64.6%	64.6%
3	58.1%	61.7%	64.0%	63.1%

4	61.0%	63.6%	65.0%	64.3%
5	61.4%	64.0%	62.9%	63.0%
6	60.4%	64.3%	64.0%	65.6%
7	61.4%	62.0%	66.9%	63.6%
8	62.3%	64.0%	64.1%	63.1%
Average:	60.83%	63.19%	64.51%	63.86%

Table 2 shows that the 3 evaluation functions outperform AB_improved consistently. And out of the 3, $mm - (4 * om)$ seems to perform the best on average.

My recommendation would be to select $mm - (4 * om)$ as the evaluation function because:

1. It gives the highest win rate among all the evaluation functions tested
2. It is very cheap to compute
3. It seems to finds the balance of chasing the opponent while still not getting itself in tight positions, chases more than $mm - (3 * om)$ but not as much as $mm - om^2$. which as shown above leads to the best results.

For the reasons mentioned above, $mm - (4 * om)$ is the evaluation function of my choice.