Artificial Intelligence – Udacity Nanodegree
Project: Build an Adversarial Game Playing Agent

Student: Carlos Alberto dos Santos

Experiment: *(Option 1) Develop a custom heuristic (must not be one of the heuristics from lectures, and cannot only be a combination of the number of liberties available to each agent)*

- **For this project, the MINIMAX WITH ALPHA-BETA PRUNING AND ITERATIVE DEEPNING algorithm was used as a basis, at first with the heuristic #my_moves - #opponent_moves and with the depth of the research was limited to 5 levels.**

- **Afterwards, other heuristics with aggressive and defensive characteristics were applied with the base algorithm, being applied 100 games for each test.**

HEURISTIC BASELINE: **"heuristic1" - #my_moves - #opponent_moves**

OPPONENT	**WINNING MATCHES (%)**
MINIMAX	43%
SELF	53%
GREEDY	63%
RANDOM	87%

In the first applications, the base algorithm was tested, which obtained good results against the opponents "GREEDY" and "RANDOM", however it did not obtain a good result against the opponent "MINIMAX".

"heuristic2" - #my_moves – 3*#opponent_moves

OPPONENT	**WINNING MATCHES (%)**
MINIMAX	60 %
SELF	53 %
GREEDY	75%
RANDOM	93%

Seeking a variation of the first approach, a change was made to a more aggressive approach, which resulted in improved performance against all opponents.

“heuristic3” - #my_moves

OPPONENT	WINNING MATCHES (%)
MINIMAX	33%
SELF	51%
GREEDY	73%
RANDOM	97%

Tests were carried out with a more conservative and defensive approach. When compared to the base algorithm, this heuristic obtained a better performance in relation to the opponents "RANDOM" and "GREEDY", but it worsened in relation to the opponent "MINIMAX".

“heuristic4” - #my_moves – 3*#opponent_moves + (own_liberties and opp_liberties)

OPPONENT	WINNING MATCHES (%)
MINIMAX	54%
SELF	49%
GREEDY	70%
RANDOM	93%

Considering that the second heuristic had presented the best result, an attempt was made to modify the approach, with the inclusion of the common free frames in the function, however there was a drop in performance when compared to the second heuristic.

QUESTIONS

What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?

As a base, the minimax algorithm was used with alpha-beta pruning, with depth limit and iterative search, due to the computational limits and the game time. I believe that this approach allows the search of the agent to advance as much as possible, respecting the restrictions already mentioned.

In the project's heuristics, the agent's own freedoms, the number of liberties of the opponent and the number of freedoms in common were used. In this way, the custom agent tries to maximize its own number of freedoms while minimizing the opponent's freedoms.

Comparing the test results, it was found that the most aggressive approaches showed a better result.

Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

I believe that the balance between the time and depth factors of the research had an impact on the result.
Regarding the time in the game, the agents had a fixed time limit (150 milliseconds by default) to research the best movement and respond, otherwise they would lose the game.
The depth of the search brings the agent closer to the best moves towards the end of the game, however there is the computational cost and time spent.
In this project, several levels of depth were tested, and in the first levels the result was very weak, especially in relation to the MINIMAX opponent.
With the increase in the depth limit, the agent showed an improvement in the results to the point where it presented a satisfactory improvement (in the case of the project, I think that the limit at level 5 has already provided a reasonable improvement).