Karolis Jankauskas, March 2017

Heuristic analysis

Introduction

A custom heuristic and several versions (Table 1) of it have been implemented and tested in attempts to improve the score of the *ID_Improved* agent which is based on iterative deepening alpha-beta tree search and *improved_score* (Table 2) heuristic.

For the non-terminal states, the custom heuristic considers not only the number of available legal moves for both the player and the opponent but also the squared Euclidean distance, $d^2(p, q)$, between the two players – the larger the $d^2(p, q)$ the larger the penalty. The goal of the custom heuristic is to have the tree search algorithm select the game states with the largest number of player moves, smallest number of opponent moves and smallest $d^2(p, q)$ between the positions of the two players. Theoretically, this should make the game agent more aggressive, corner and deny the opponent blank spaces on the board. The custom heuristic based game agent relies on iterative deepening alpha-beta tree search.

The custom heuristic was further explored by assigning a weight of 2 to the different terms in the evaluation function, and measuring the impact on the winning rate against the game agents based on the default heuristics (Table 2).

Table 1. A list of the custom heuristics

custom1	*#own_moves – #opp_moves – sqrd_euclidean_distance*
custom2	*#2 x own_moves –#opp_moves – sqrd_euclidean_distance*
custom3	*#own_moves – 2 x #opp_moves – sqrd_euclidean_distance*
custom4	*#own_moves – #opp_moves – 2 x sqrd_euclidean_distance*

Table 2. A list of the default heuristics

null_score	*no knowledge of non-terminal states*
open_move_score	*#own_moves*
improved_score	*#own_moves – #opp_moves*

Karolis Jankauskas, March 2017

Results

The game agent based on the “vanilla” version of the custom heuristic *custom1* won more games than the *ID_Improved* agent against the 6 out 7 default game agents, achieving a 3.79% higher average winning rate (67.43% versus 63.64%).

Placing more emphasis on either the number of legal moves available for the player or penalizing the game states with more available legal moves for the opponent did not seem to have a significant effect on the performance of the custom heuristics *custom2* and *custom3*. The difference between the average winning rates of the first three custom heuristics *custom1, custom2,* and *custom3* was negligible.

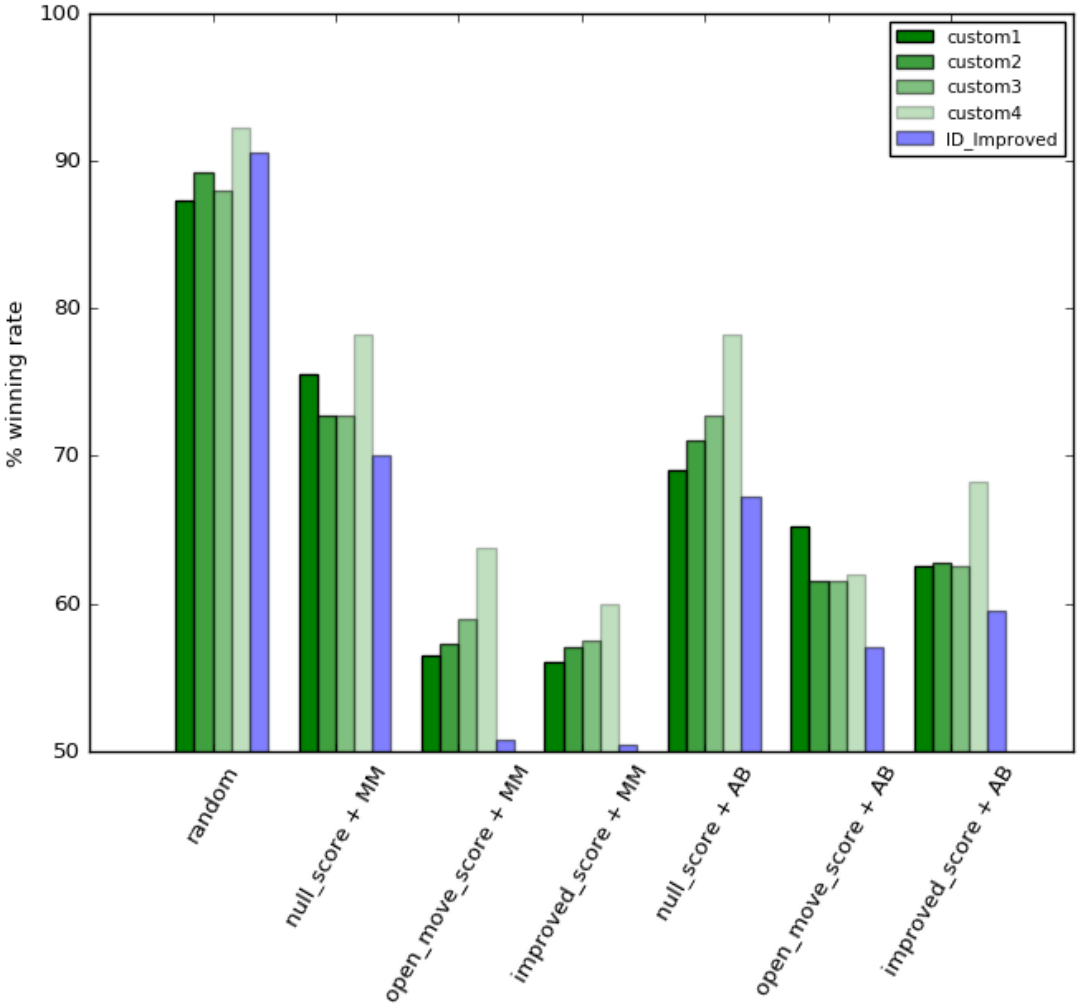

Figure 1. Percentage of games won by the different versions of the custom heuristic and the *ID_Improved* game agent against the default game agents (x-axis)

Table 3. A comparison of the winning rates between the different versions of the custom heuristic and the default game agents

	custom1	***custom2***	***custom3***	***custom4***
random	349 to 51	357 to 43	352 to 48	369 to 31
null_score + MM[1]	302 to 98	291 to 109	291 to 109	313 to 87
open_move_score + MM	226 to 174	229 to 171	236 to 164	255 to 145
improved_score + MM	224 to 176	228 to 172	230 to 170	240 to 160
null_score + AB[2]	276 to 124	284 to 116	291 to 109	313 to 87
open_move_score + AB	261 to 139	246 to 154	246 to 154	248 to 152
improved_score + AB	250 to 150	251 to 149	250 to 150	273 to 127
Overall winning rate	**67.43%**	**67.35%**	**67.71%**	**71.82%**

[1] Minimax tree search

[2] Alpha-beta tree search

However, placing more emphasis on the $d^2(p, q)$ term (multiplying it by 2) significantly improved the winning rate of the custom heuristic *custom4.* The average winning rate of this version was 4.39% higher than that of the base version, and 8.18% higher compared to the winning rate of the *ID_Improved* game agent. *Custom4* won more games against all the default game agents than any other version of the custom heuristic and the *ID_Improved* game agent. Furthermore, *custom4* won 57.75% of games (231 to 169) against the *ID_Improved* game agent (see Figure 2).

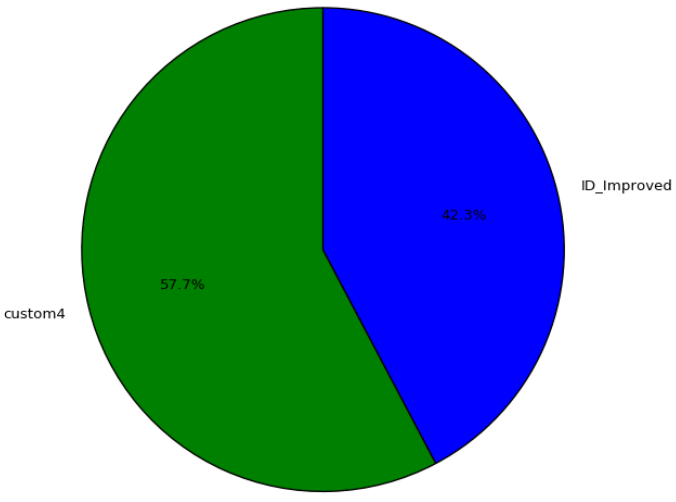

Figure 2. The proportion of games won by the custom heuristic *custom4* and *ID_Improved* game agent

Table 4. A comparison of the winning rates between the best performing custom heuristic and the *ID_improved* agent

	ID_improved	***custom4***
random	362 to 38	369 to 31
null_score + MM	280 to 120	313 to 87
open_move_score + MM	203 to 197	255 to 145
improved_score + MM	202 to 198	240 to 160
null_score + AB	269 to 131	313 to 87
open_move_score + AB	228 to 172	248 to 152
improved_score + AB	238 to 162	273 to 127
Overall winning rate	**63.64%**	**71.82%**

Both *custom4* and *ID_Improved* took approximately 5 min to finish playing 20 matches against each default game agent.

Conclusions

The decision to not only include the $d^2(p, q)$ term in the state evaluation function but also giving more weight to it had a significant positive impact on the average winning rate of the custom heuristic *custom4* compared to the *ID_Improved* game agent. *Custom4* not only had a higher average winning rate than the *ID_Improved* agent but also outperformed it against every single default game agent. This was also achieved without any significant computational cost. Therefore, the recommendation is to use the *custom4* state evaluation function instead of the *improved_score.*