Heuristic Analasys

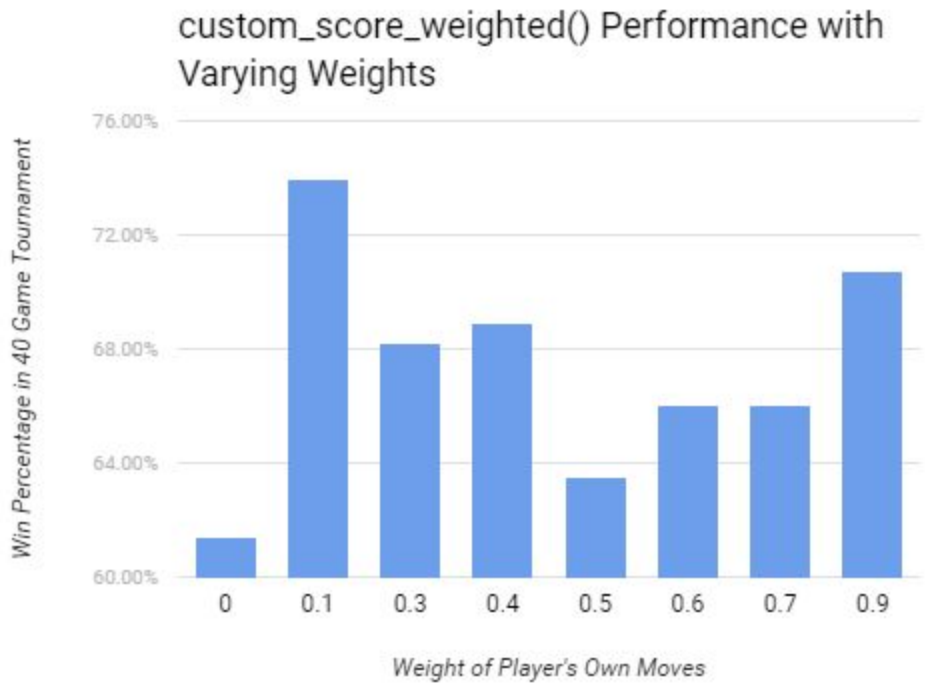
Note: originally, the tournament.py test plays 20 matches between players. To reduce variance in scores, while maintaining a reasonable runtime, all data in this analysis is taken with 40 games played between players.

custom_score_distance()

This heuristic takes into account the euclidian distance to the opponent based on the theory that if you can stay further from the opponent, it is less likely they will block your moves. The original heuristic scored 63.53% and this heuristic improved that to 74.64% by favoring moves further from the opponent just slightly.

custom_score_weighted()

I tested the original “improved heuristic” but assigned different weights to “player moves” and “opponent moves”. The weights always sum to 1 The performance can be seen in the data and chart below (note that the “improved score” is equivalent to the weight (.5, .5) in this case. Across these 8 tests of 40 games against each opponent, the mean score for the improved score was 63.53 with a variance of 3.08%:



Weight of Player Moves	Weight of Opponent Moves	Score
0	1	61.43%
.1	.9	73.93%

.3	.7	68.21%
.4	.6	68.93%
.5	.5	63.53%
.6	.4	66.07%
.7	.3	66.07%
.9	.1	70.71%

There is not a clear trend to this data except that the two weights which weighed one of the variables as important and took the other into slight consideration performed better. The (.1, .9) pair is far outside of the variance seen for the improved score and is likely a better metric.

custom_score_symmetry()

This heuristic takes advantage of the following fact: if we are playing second, and start in a position that is either horizontally, vertically or diagonally symmetric to our opponent's move, we can win the game if we move to maintain that symmetry and if our opponent does not break the symmetry. In these games, the second player always has the last move. On even-sized boards, it is impossible for our opponent to break symmetry, so we will always win these games if we go second starting at a symmetric board state. The probability of starting in a symmetric state depends on the size of the board. If the state is not symmetric, the heuristic can fall back on another heuristic.

In testing, the "improved heuristic" and the symmetric heuristic scored about the same on an 8x8 board. This may be due to the fact that symmetric board states are unlikely at this size, however in a 4x4 board, the results were the same - this may mean that in the majority of symmetric games, the fallback heuristic (being the improved heuristic) is already winning the game, and it is the non-symmetric games that the fallback heuristic struggles with. In any case, moving based on symmetry is clearly a strictly better heuristic on even-board sizes where symmetry applies, even if the probability of a symmetric game is low.

In this assignment, the board size is 7x7, so the opponent always has the opportunity to break symmetry by moving into the center space, and in horizontally and vertically symmetric setups, the opponent can break the symmetry by moving into the center row or column respectively. Because minimax assumes that our opponent will try to minimize our value, most searches will result in our opponent breaking the symmetry, so the strategy is not pursued. Therefore, it is typically worth pursuing this heuristic only on even boards.

This held true in testing - the symmetric heuristic scored worse than the "improved heuristic" (57.5% vs 63.53%) probably because the few times that it did pick a move based on symmetry, it was because it had not searched far enough to see that the opponent would break that symmetry, making the "symmetric move" non-optimal.

The second drawback of this heuristic is that it takes more time to compute a score - on a slower system this could cost the player an iteration of

The takeaway here is that if the board is even, and we can select our starting position to be symmetric, we could always win by going second.

Conclusion

This assignment asks that we select a single Heuristic to recommend, but I would recommend "custom_score_final()," a combination of the three. "custom_score_final()" uses the symmetric heuristic on even-sized boards and fall back on the weighted heuristic with weights (.1, .9), favoring moves with tied scores based on the distance heuristic. I'd recommend this for three reasons:

1. The symmetric strategy, as mentioned earlier is strictly better than other heuristics on even-boards; if it applies to a board, the second player always wins, otherwise the original heuristic will perform as well as it would without the symmetric heuristic.
2. The (.1, .9) weighting of player moves to opponent moves provided the best performance of any weight pair.
3. Tie-breaking based on distance provides a large boost which is orthogonal to the weighted score heuristic, therefore it can be applied to improve performance. This is reflected in the score of this algorithm when implemented on a 7x7 board where the symmetric strategy would not come into play. (the final score is 79.64%)