Heuristic Description

After spending some time optimizing the search algorithm, I experimented with several different heuristics. The three I will compare here are described below

1. “Student3”: A heuristic that favors the center of the board for the player and the edges of the board for the opponent.
    a. distance(opponent_location, center) - distance(player_location, center)
    b. This was based on observations that most games end with a player trapped in the outer edges of the board, and some games end very quickly with one player trapped in a corner.
2. “Student8”: A heuristic that tries to maximize the player’s distance from the opponent.
    a. distance(player_location, opponent_location)
    b. Based on a simple idea to run away from the opponent to try to outlive it.
3. “Student12”: A modification of “ID_Improved” that takes into account the number of moves that have been played in the game along with some coefficients for both the “player” term and the “opponent” term.
    a. own_coef * move_count * own_moves - opp_coef * move_count * opp_moves
    b. After many observations, it seemed likely that the number of moves taken in a match have a large effect on the state of the game and possibly the strategy of the next move. However it was unclear how to “weight” the effect of the move count for each term.

These three were chosen because they provide a good variation in approaches and performance results.

Heuristic Development

Out of the three heuristics, the majority of testing was focused on “Student 12” since the optimal coefficient values were unknown. In an attempt to find optimal coefficient values, an optimization loop was defined based loosely on concepts from genetic algorithms. Basically, five Student12 agents were initialized with random weights uniformly distributed between 0 and 2. Each agent then played a round against an unchanging set of higher difficulty agents. After a tournament run, the Student12 agents were re-initialized with values similar to the top performing Student12 agents from the previous iteration. Ideally this process would converge on optimal coefficient values resulting in the highest performance.

The difficulty in this process stemmed from finding a balance between performance accuracy and the amount of time it took to run and iteration. A higher number of matches between players resulted in a higher accuracy, but also slower iterations, however the iterations needed to be fast enough to actually converge on optimal values. The lowest match number that seemed to provide a decent amount of accuracy (~+/- 1.5%) was 100 (400 matches

between each opponent in the tournament code). However this seemed to take a sub-optimal amount of time (~5-7 hours depending on cpu). To strike a balance, 18 total instances of the optimization code were deployed on 3 separate Google Cloud Compute Engine VM instances, each with 8vCPUs. This in effect reduced the number of iterations required for each individual instance since the top results between instances could be compared, giving a better overall perspective of which coefficients seemed most promising.

After several iterations the most promising coefficients were chosen and compared against other heuristics in two ways: first by playing against the default set of “test” agents in the tournament code, then by playing each other in a round robin tournament. The results can be seen below. (NOTE: the ID_Improved agent is the default heuristic using basic aplhabeta pruning while the other agents employ search optimizations discussed earlier)

Results

Student12: (own_coef = 1.492220782479327, opp_coef = 0.7729218598739231)

TOURNAMENT 1 - VS DEFAULT OPPONENTS

```
*************************
 Evaluating: ID_Improved
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random      Result: 363 to 37
  Match 2: ID_Improved vs   MM_Null     Result: 335 to 65
  Match 3: ID_Improved vs   MM_Open     Result: 276 to 124
  Match 4: ID_Improved vs MM_Improved   Result: 271 to 129
  Match 5: ID_Improved vs   AB_Null     Result: 316 to 84
  Match 6: ID_Improved vs   AB_Open     Result: 290 to 110
  Match 7: ID_Improved vs AB_Improved   Result: 260 to 140


Results:
----------
ID_Improved         75.39%

*************************
Evaluating: ID_Improved_Optimized
*************************
```

```
Playing Matches:
----------
  Match 1: ID_Improved_Optimized vs   Random           Result: 376 to 24
  Match 2: ID_Improved_Optimized vs   MM_Null          Result: 348 to 52
  Match 3: ID_Improved_Optimized vs   MM_Open          Result: 288 to 112
  Match 4: ID_Improved_Optimized vs MM_Improved        Result: 288 to 112
  Match 5: ID_Improved_Optimized vs   AB_Null          Result: 338 to 62
  Match 6: ID_Improved_Optimized vs   AB_Open          Result: 311 to 89
  Match 7: ID_Improved_Optimized vs AB_Improved        Result: 282 to 118


Results:
----------
ID_Improved_Optimized     79.68%

**************************
 Evaluating: Student3
**************************

Playing Matches:
----------
  Match 1: Student3    vs   Random      Result: 373 to 27
  Match 2: Student3    vs   MM_Null     Result: 361 to 39
  Match 3: Student3    vs   MM_Open     Result: 291 to 109
  Match 4: Student3    vs MM_Improved   Result: 284 to 116
  Match 5: Student3    vs   AB_Null     Result: 331 to 69
  Match 6: Student3    vs   AB_Open     Result: 290 to 110
  Match 7: Student3    vs AB_Improved   Result: 271 to 129


Results:
----------
Student3           78.61%

**************************
 Evaluating: Student8
**************************

Playing Matches:
----------
  Match 1: Student8    vs   Random      Result: 380 to 20
  Match 2: Student8    vs   MM_Null     Result: 360 to 40
  Match 3: Student8    vs   MM_Open     Result: 274 to 126
```

```
  Match 4: Student8    vs MM_Improved   Result: 259 to 141
  Match 5: Student8    vs   AB_Null     Result: 334 to 66
  Match 6: Student8    vs   AB_Open     Result: 274 to 126
  Match 7: Student8    vs AB_Improved   Result: 265 to 135


Results:
----------
Student8           76.64%

*************************
Evaluating: Student12
*************************

Playing Matches:
----------
  Match 1: Student12    vs   Random     Result: 380 to 20
  Match 2: Student12    vs   MM_Null    Result: 359 to 41
  Match 3: Student12    vs   MM_Open    Result: 319 to 81
  Match 4: Student12    vs MM_Improved  Result: 289 to 111
  Match 5: Student12    vs   AB_Null    Result: 342 to 58
  Match 6: Student12    vs   AB_Open    Result: 308 to 92
  Match 7: Student12    vs AB_Improved  Result: 281 to 119


Results:
----------
Student12          81.36%
```

TOURNAMENT 2 - ROUND ROBIN

****
 Evaluating: ID_Improved
****

Playing Matches:

  Match 1: ID_Improved vs ID_Improved   Result: 195 to 205
  Match 2: ID_Improved vs ID_Improved_Optimized   Result: 204 to 196
  Match 3: ID_Improved vs Student3   Result: 182 to 218
  Match 4: ID_Improved vs Student8   Result: 247 to 153
  Match 5: ID_Improved vs Student12   Result: 190 to 210

Results:

ID_Improved   50.90%

****
Evaluating: ID_Improved_Optimized
****

Playing Matches:

  Match 1: ID_Improved_Optimized vs ID_Improved   Result: 210 to 190
  Match 2: ID_Improved_Optimized vs ID_Improved_Optimized   Result: 213 to 187
  Match 3: ID_Improved_Optimized vs Student3   Result: 235 to 165
  Match 4: ID_Improved_Optimized vs Student8   Result: 225 to 175
  Match 5: ID_Improved_Optimized vs Student12   Result: 160 to 240

Results:

ID_Improved_Optimized   52.15%

****
 Evaluating: Student3
****

Playing Matches:

  Match 1: Student3   vs ID_Improved   Result: 208 to 192

```
  Match 2: Student3    vs ID_Improved_Optimized         Result: 147 to 253
  Match 3: Student3    vs Student3      Result: 183 to 217
  Match 4: Student3    vs Student8      Result: 257 to 143
  Match 5: Student3    vs Student12     Result: 171 to 229


Results:
----------
Student3           48.30%

*************************
 Evaluating: Student8
*************************

Playing Matches:
----------
  Match 1: Student8    vs ID_Improved   Result: 168 to 232
  Match 2: Student8    vs ID_Improved_Optimized         Result: 146 to 254
  Match 3: Student8    vs Student3      Result: 150 to 250
  Match 4: Student8    vs Student8      Result: 202 to 198
  Match 5: Student8    vs Student12     Result: 183 to 217


Results:
----------
Student8           42.45%

*************************
Evaluating: Student12
*************************

Playing Matches:
----------
  Match 1: Student12    vs ID_Improved  Result: 229 to 171
  Match 2: Student12    vs ID_Improved_Optimized        Result: 257 to 143
  Match 3: Student12    vs Student3     Result: 206 to 194
  Match 4: Student12    vs Student8     Result: 227 to 173
  Match 5: Student12    vs Student12           Result: 211 to 189


Results:
----------
Student12          56.50%
```

Analysis

The results show that the “Student12” heuristic with semi-optimized coefficients outperformed the other heuristics by a small but noticeable margin, followed closely by ID_Improved_Optimized (the original ID_Improved heuristic running on an optimized alphabeta algorithm). Although performance against the “default” test agents between ID_Improved_Optimized and Student12 were within the margin of error for these results, the round robin results (an arguably more “realistic” set of opponents) displayed a clear performance advantage for Student12. In addition multiple similar tests showed consistent results, with Student12 outperforming the default heuristic.

Conclusion

Although the Student12 heuristic showed an improvement over the default heuristic, it is likely far from optimized. Given more processing time, better coefficient values could be discovered. In addition, the same process of adding/optimizing coefficients could be applied to all other heuristics, including the default heuristic, which could prove to outperform the Student12 heuristic.