# Heuristic Analysis

Stefan Heidekrüger

August 11, 2017

This document contains some analysis on the position evaluation heuristics that have been tried and ultimately selected for the *Knight-Isolation* game playing agent. We will briefly describe the methodology as well as the findings of our analysis.

## Methodololgy

The goal of an evaluation heuristic $v$ is to approximate the *value* $v^*$ of the game (from the point of view of the agent player.) As such, it's always the goal to win the game and avoid losing it, so it makes sense to set $v(\text{winning position}) = \infty$ and $v(\text{losing position}) = -\infty$. (These values are well-defined, as our alpha-beta-tuning algorithm solely relies on *ordinal* comparisons of values; if we instead employed Monte-Carlo tree search or other similar algorithms, we would need to use finite values in order to *weigh* moves somehow —and also worry about interval scaling.)

More importantly, let's consider positions that are neither clearly winning or losing. We decided that the heuristic should be primarily decided based on the *number of possible moves for the playing agent $n$* and the *number of possible moves for the opposing player $m$*. Other measures, such as centrality, were briefly considered but were quickly discarded when they didn't show much promise in the first experiments.

As it's obvious that $m$ should affect the heuristic negatively (and $n$ positively), we considered polynomial functions of the form

$$v(n, m) = an^b - cm^d$$

with $0 \leq a, c \lessapprox 10$, $0 < b, d \lessapprox 3$ (a value of 0 for $b$ or $d$ would be irrelevant due to translation invariance).

Listing 1: Sample output of `tournament.py` for one tournament (T3 in tables 1 , 2) for heuristics introduced below.

```
                         *************************
                              Playing Matches
                         *************************

 Match #    Opponent    AB_Improved   AB_Custom    AB_Custom_2  AB_Custom_3
                         Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1        Random       8  |  2     10  |  0      9  |  1     10  |  0
    2        MM_Open      4  |  6      4  |  6      7  |  3      8  |  2
    3       MM_Center     6  |  4      8  |  2      9  |  1      8  |  2
    4      MM_Improved    5  |  5      6  |  4      7  |  3      7  |  3
    5        AB_Open      5  |  5      7  |  3      6  |  4      5  |  5
    6       AB_Center     4  |  6      4  |  6      5  |  5      5  |  5
    7      AB_Improved    6  |  4      7  |  3      6  |  4      5  |  5
--------------------------------------------------------------------------
           Win Rate:      54.3%        65.7%        70.0%        68.6%
```

Due to time-constraints, we did not perform an exhaustive parameter-search in this space, but rather manually tried different combinations for a very superficial search by hand. In each case, decisions were made on a handful of cases of round-robin tournaments as provided in `tournameny.py` as no resources for more robust methods were available. A sample output of such a tournament is given in . It should be noted that the results contained "a lot" of variance across the tournaments and any analysis result should therefore be taken with a grain of salt and only be seen as a rough draft rather than thoroughly tested insights on valuable heuristics. The given implementation `AB-Improved` served as the main performance benchmark and has thus been included as a possible heuristic for direct comparison.

### Narrowing down the parameter space

From the first few experiments (results omitted), it quickly became clear that the following should hold:

- $a > 0, c > 0$
- $n$ should be weighed stronger than $m$, roughly speaking $cm^d = \mathcal{O}(an^b)$
- Ideally, each of the parameters should be reasonably small - otherwise one of the terms would dominate and render the other obsolete. This however was found not to be desirable.

After some experimentation, we finally considered the following 3 candidate heuristics for a somewhat more formal comparison across 5 tournaments, which each adher to the properties above:

- $v_0$ baseline `AB-Improved`
- $v_1 = 2n^2 - m\sqrt{m}$ `AB-Custom`
- $v_2 = n\sqrt{n} - 2m$ `AB-Custom2`
- $v_3 = 2n - m$ (with additional check: If player's only possible move can be blocked in opponent's next move, then $v_3 = -100$) `AB-Custom3`

As the main critera for comparison of these candidate heuristics, we consider repeated round-robin-performance as well as each agent's one-on-one track record against the oponent's instance of `AB-Improved` –both across a series of 5 consecutive tournaments. Detailed results of the round-robin-scores and direct performance against the baseline are given in tables 1 and 2, respectively.

Table 1: Round-Robin performance of selected heuristics over a series of 5 tournaments.

| Agent | T1 | T2 | T3 | T4 | T5 | avg |
|---|---|---|---|---|---|---|
| $v_0$ (`AB-Improved`) | 52.9 | 60.0 | 54.3 | 60.0 | 54.3 | 56.30 |
| $v_1$ | 51.4 | 57.1 | 65.7 | 64.3 | 71.4 | 61.98 |
| $v_2$ | 65.7 | 64.3 | 70.0 | 71.4 | 55.7 | **65.42** |
| $v_3$ | 70.0 | 51.4 | 68.6 | 65.7 | 57.1 | 62.56 |

Table 2: Direct comparison of agents (lhs) to opponent instance of `AB-Improved` (rhs).

| Agent | T1 | T2 | T3 | T4 | T5 | total | win rate |
|---|---|---|---|---|---|---|---|
| $v_0$ (`AB-Improved`) | 5 : 5 | 4 : 6 | 6 : 4 | 5 : 5 | 6 : 4 | 26 : 24 | 52% |
| $v_1$ | 4 : 6 | 4 : 6 | 7 : 3 | 6 : 4 | 6 : 4 | 27 : 23 | 54% |
| $v_2$ | 5 : 5 | 5 : 5 | 6 : 4 | 8 : 2 | 6 : 4 | 30 : 20 | **60%** |
| $v_3$ | 6 : 4 | 2 : 8 | 5 : 5 | 5 : 5 | 5 : 5 | 28 : 22 | 56% |

### Results and Final Heuristics

In the experiment all of our custom heuristics outperformed `AB-Improved`, both in terms of RR-tournament scores and in direct comparison. Moreover, we chose $v_2$ as our final heuristic, as it outperformed all other heuristics in several ways:

- $v_2$ achieved the highest average RR-score in the experiment.
- $v_2$ also achieved the highest direct comparison score against the baseline.
- Finally, $v_2$'s Round-Robin scores dominated the baseline's RR-scores in each individual tournament, a feat that wasn't achieved by any of the other candidate heuristics.

In conclusion, this lead to us chosing the following final heuristic function:

$$v_2 = n^{1.5} - 2m$$