Heuristic Analysis

Ilya Nikokoshev

January 7, 2018

Abstract

As part of the Project 2 in the Artificial Intelligence Nanodegree Program, we implement three heuristics for score computation.

Contents

1 Heuristic Descriptions **1**
1.1 Distance Heuristic $D(z, \alpha, f)$ 1
1.1.1 Definition 1
1.1.2 Theoretical analysis 2
1.1.3 Validating the parameter choices 2
1.2 Free Moves Heuristic $F(d, u, \beta)$ 3
1.2.1 Definition 3
1.2.2 Selection of β 3
1.2.3 Selection of d and u 4

2 Final Choices **4**

1 Heuristic Descriptions

All of our heuristics attempt to express the goodness of a position as a value that would be larger whenever there seem to be more choices for the first player (denoted A) and fewer choices for the opponent (denoted B).

1.1 Distance Heuristic $D(z, \alpha, f)$

1.1.1 Definition

For each blank space s on the board, we find whether it is accessible from the current positions of players A and B. If it is not accessible by either, this square does not contribute to the heuristic value.

Otherwise, let d_s^A and d_s^B denote the minimum number of steps required to move to this square from the positions of, respectively, player A and B, or $+\infty$ if this square is inaccessible. We set the contribution of s to the heuristic then as $\alpha f(d_s^B - d_s^A)$, where f is a suitably chosen monotonic function with the property P that $f(+\infty) = 1$ and $f(-\infty) = -1$.

The total value of the heuristic will be therefore computed as

$$D(z, \alpha, f) = z + \alpha \sum_{s \in A} f(d_s^B - d_s^A),$$

where A is the set of accessible spaces on the board, d_s^P is defined above, and z provides the initial value.

1.1.2 Theoretical analysis

We will select either arctan or sigmoid function $(1 + e^{-x})^{-1}$ as the function f (both of them, of course, suitably rescaled).

The condition P ensures that if the board is divided into two disjoint parts, so that A only has access to n_A cells, and B only has access to some other n_B cells, the value of the heuristic $D = z + \alpha(n_A - n_B)$. Naively, such a position is likely to be winning for A if and only if $n_A > n_B$. Moreover, for the cases when $n_A = n_B = 0$ (A loses immediately), and $n_A = 1, n_B = 0$ (B loses after A's move), the utility is known to be equal to -1 and 1. If we want the value of D to be equal to the utility in these two cases, we should select $\alpha = 2, z = -1/2$.

1.1.3 Validating the parameter choices

For comparison, we provide results of the competition[1] of matches between selected distance heuristics with different values of parameters and `improved_score` heuristic in Table 1. In the table, cells corresponds to the percentage of wins of an alphabeta search agent with a row heuristic against the column heuristic in 100 matches, and the average is computed from rows as well as columns (thus, it is an average win in 700 parties against other opponents).

Table 1: Distance heuristics competition

	z	α	f	Wins against 1	2	3	4	Improved	Average
Variant 1	0	1	arctan	52%	52%	51%	58%	57%	53.0%
Variant 2	-1/2	1	arctan	50%	52%	51%	47%	45%	49.3%
Variant 3	-1/2	2	arctan	48%	51%	55%	48%	51%	48.9%
Variant 4	-1/2	2	sigmoid	49%	45%	54%	49%	49%	49.1%

The table suggests that, contrary to the theoretical analysis in 1.1.2, the Variant 1 heuristic is the strongest, and 3 and 4 are the weakest.

Unfortunately, the standard deviation for the case of tossing a perfect coin 700 times comprises about 2%, which means that the results in the average column cannot rule out the hypothesis that all of the opponents have equal strength. Thus the confidence in the results above is not very high.

In an attempt to further examine the question of parameters, we put Variant 1 against Variant 4 in two other competitions with 500 matches, not only on a standard 7×7, but also on a 13×13 boards. The results, presented in Table 2, do not strongly support selecting one choice over the other.

[1] Raw data is available at `https://github.com/ilyannn/AIND-Isolation/tree/master/tournaments`

Table 2: Further comparison of distance heuristic variants 1 and 4

Board size	Wins of 4 over 1
7×7	52.6%
13×13	50.6%

In combination with the theoretical analysis, we therefore make a choice to take $D(-1/2, 2, \text{sigmoid})$ as the preferred distance heuristic.

1.2 Free Moves Heuristic $F(d, u, \beta)$

1.2.1 Definition

We will attempt to perform a random unroll of the board and compute the amount of free moves available at each step. Naturally, the moves available to the opponent B should count as a negative for the player A. We will need to restrict us to certain maximal depth d, so $F(d, u, \beta)$ will be computed recursively starting from $F(0, u, \beta) = 0$.

For a given board $\mathfrak{b}$ and $d > 0$, the *free moves score* will be therefore defined as

$$F_{\mathfrak{b}}(d, u, \beta) = n - \frac{\beta}{\min(v, n)} \sum_{\mathfrak{b}' \in S(\mathfrak{b}, u)} F_{\mathfrak{b}'}(d - 1, u, \beta),$$

where n is the number of available moves for the current player, $S(\mathfrak{b}, u)$ is the set of no more than u boards obtained by selecting a random subset of the possible moves, and β is a positive number that is part of the definition of the score function.

1.2.2 Selection of β

We will first consider the simplest case of looking forward a single step ahead and attempt to find a preferred value for parameter β by comparing a few of the heuristics in 200 matches against alphabeta improved agent. According to the results in Table 3, values of β around 1.2 seem to be the most promising (variant numbers correspond to those given in `FreeMoveBetaCompetition.png`).

Table 3: Free move heuristics competition ($d = 2$, $u = 3$, variance of β)

	β	Wins against Improved
Variant 2	1	44.0%
Variant 3	0.8	49.5%
Variant 4	1.2	52.5%
Variant 5	1.4	47.5%

This corresponds to our intuition that making an aggressive move that takes away the choices from the opponent is slightly more important than keeping choices for yourself.

1.2.3 Selection of d and u

Unrolling the position tree deeper or considering more moves at each depth would be likely to give us a better heuristic value at the cost of time: the total number of positions analysed in a typical computation $T = 1 + u + u^2 + \ldots + u^{d-1}$. Table 4 compares several agents that increase d or u in a competition of 500 matches each and the standard timeout of 250ms.

Table 4: Free move heuristics competition ($\beta = 1.2$)

	d	u	T	Wins against 4	Improved
Variant 4	2	3	4	48.4%	52.8%
Variant 7	2	10	11	49.2%	51.0%
Variant 8	3	4	21	50.0%	49.8%
Variant 9	4	2	15	50.2%	53.0%

The results seem to suggest that the variants 4 and 9 are the strongest scores. We will take them as two final heuristic choices.

2 Final Choices

We combine the results of 1.1.3 and 1.2.3 to select three heuristic functions and present the results of a competition between them and `improved_score` heuristics in Table 5.

Table 5: Final heuristics competition

	Function Name	Heuristic	1	2	3	Open	Center	Improved
1	`custom_score`	$D(-1/2, 2, \text{sigmoid})$						
2	`custom_score_2`	$F(2, 3, 1.2)$						
3	`custom_score_3`	$F(4, 2, 1.2)$						