Heuristic Analysis

Ilya Nikokoshev

January 1, 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

2 Final Choices **3**

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 (suitably rescaled) arctan or sigmoid function $(1 + e^{-x})^{-1}$ as the function f.

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 is $z + \alpha(n_A - n_B)$. Naively, such a position is likely to be winning for A if and only if $n_A > n_B$. If we want the value to be equal to the utility of -1 when $n_A = n_B = 0$, and 1 when $n_A = 1, n_B = 0$, we should select $\alpha = 2, z = -1/2$..

1.1.3 Validating the parameter choices

For comparison, we provide results of the competition 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 against 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	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 above, 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 even for the average result column, these deviations of $1 - 3\%$ could be possible even for the opponents of perfectly equal strength. Therefore, the confidence in these results is not very high.

In combination with the theoretical analysis above, we therefore make a choice to take both $D(0, 1, \text{arctan})$ and $D(-1/2, 2, \text{arctan})$ as the preferred distance heuristics.

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

2 Final Choices

We combine the results of 1.1.3 to select three heuristic functions and present the results of a competition between them and `improved_score` heuristics in Table 2.

Table 2: Final custom score heuristics

	Function	Heuristic	1	2	3	Improved
1	`custom_score`	$D(0, 1, \text{arctan})$				
2	`custom_score_2`	$D(-1/2, 2, \text{sigmoid})$				
3	`custom_score_3`	$D(-1/2, 2, \text{sigmoid})$				