AIND-Build a Game-Playing Agent

Heuristic function analysis

Cheng Wang

Three types of functions are explored. For each score, tournament is run for 10 times and the results are shown in Table 1 with max, min, median, mean, and standard deviation for ID_Improved and Student. The label and formula of scores are also listed in Table 1 at the end of this report. The games were performed on HPC cluster with Intel Xeon E5-2690v4 2.6GHz CPUs and CentOS 7.2 system.

Type 1

Type 1 is started from the improved score (score = own – opp, where *own* is the number of moves for player itself, and *opp* is the number of moves for opponent player) from lecture. This score takes account of own moves and opponent moves together with equal contribution. A series of scores are designed by function form:

score = c1*own – c2*opp

The improved score is denoted as EF1 with c1 = 1 and c2 = 1. EF2 to EF5 are scores with more contribution from own moves, while EF6 to EF9 are scores with more contribution from opponent moves. EF2 is aggressive on maximize own moves by only considering own moves and EF6 is aggressive on minimize opponent move by only considering opponent moves. By varying c1 and c2, we get different contribution from own and opponent. The results are shown in Figure 1.

In addition to success rate, number of student wins over ID_improved among 10 tournaments is also shown as in Figure 1(b) by comparing the success rate. We can see that EF8 (score = own – 4 opp) has the highest median (76.4%) as in boxplot in Figure 1(a) and wins 8 out of 10 times as in bar plot in Figure 1(b). Then this score function is considered for type 2 design.

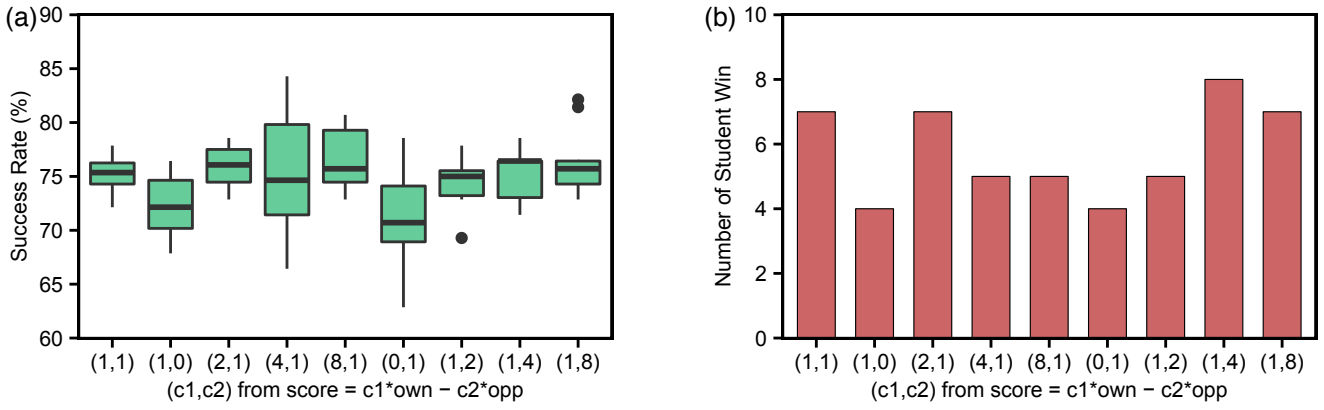

Figure 1. (a) Boxplot of success rates for scores EF1 to EF9. (b) Bar plot of number of student wins for score EF1 to EF9. The score is c1*own – c2*opp and the x-axis is parameter for c1 and c2.

Type 2

In type 1, the same score is used over entire game. In general, we think the game opening and game closing should have different strategies. A simple way to do that is use different scores. Here two scores EF1 (score = own – opp) and EF8 (score = own – 4opp) are considered. The time to start new score depends on the empty space of the game board. Scores EF10 to EF15 are designed based on this rule. The score functions are switched when the number of blank spaces reaches different fraction (such as 1/2, 1/4, and 3/4) of board size. For example in EF10, the score is

```
if emptyspace < boardsize/2:
        score = own – 4opp
else:
        score = own – opp
```

The results are presented in Figure 2 and none of them is better than EF8. The median of EF14's success rate (76.4%) is similar to EF8 (76.4%). Scores in type 2 may be not necessary. One thing should take into consideration is the code running time. The time limit is 150 ms. Complex score takes longer than simple score and this could limit the search depth.

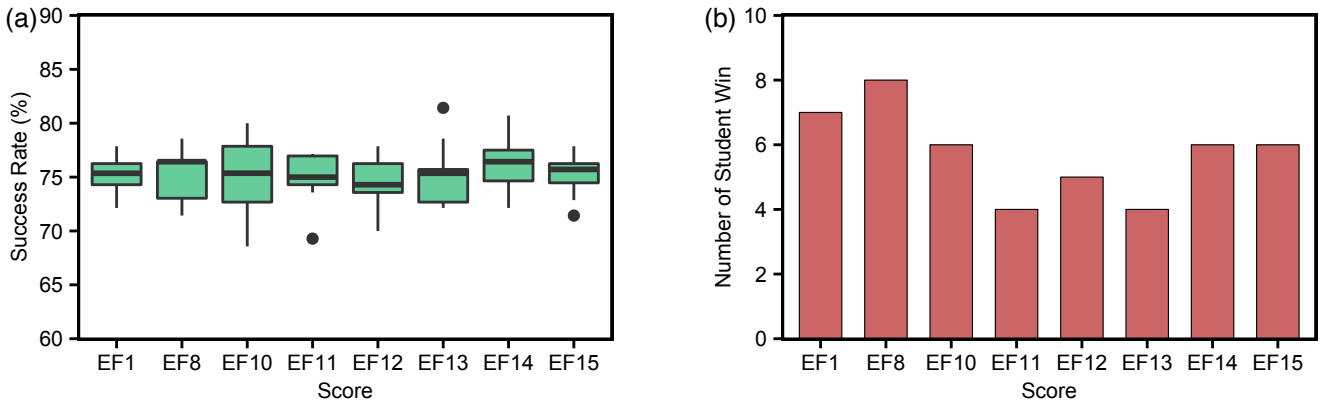

Figure 2. (a) Boxplot of success rates for scores EF1, EF8, and EF10 to EF15. (b) Bar plot of number of student wins for score EF1, EF8, and EF10 to EF15.

Type 3

In addition to linear combination of own and opponent moves, score (EF16) is defined as own/opp. This maximizes own moves and minimizes opponent moves. The median of EF16 (76.1%) is slightly smaller than EF8 as in Figure 3.

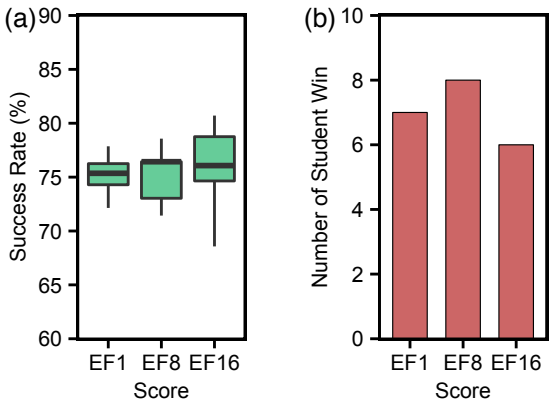

Figure 3. (a) Boxplot of success rates for scores EF1, EF8, and EF16. (b) Bar plot of number of student wins for score EF1, EF8, and EF16.

ID_Improved

The success rate of a certain score varies in multiple runs. For ID_Improved, the success rate ranges from 64.3% to 82.9% in 160 runs. The average is 74.8% and standard deviation is 3.1%. The distribution of success rate is shown in Figure 4. The CPU type is homogeneous in cluster, but the speed varies from my experience from other tasks. Therefore, the performance varies. In addition to CPU, the code to calculate the score could also lead to the performance difference.

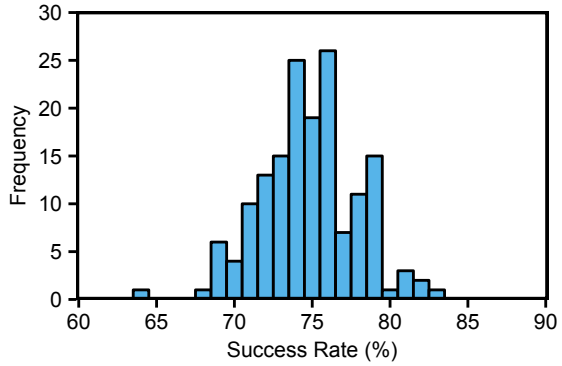

Figure 4. Histogram of success rates for ID_Improved for 160 tournaments.

Summary

Given the result above, the score EF8 is used as my final score. It should be noted that none of designed score is significant better than ID_Improved from t-test with $p < 0.05$.

Table 1. Results of Tournament for Different Scores

			ID_Improved					Student				
type	label	function form[a]	max	min	median	average	sd	max	min	median	average	sd
type 1	EF1	own-opp	78.6	72.1	75.4	74.9	2.3	77.9	72.1	75.4	75.1	1.7
	EF2	own	80.0	68.6	75.0	74.6	3.5	76.4	67.9	72.2	72.2	2.7
	EF3	2own-opp	78.6	67.9	74.7	74.1	3.5	78.6	72.9	76.1	75.9	2.0
	EF4	4own-opp	82.1	69.3	74.3	74.4	3.4	84.3	66.4	74.7	75.1	5.5
	EF5	8own-opp	78.6	72.9	74.6	75.2	2.2	80.7	72.9	75.7	76.6	2.7
	EF6	-opp	78.6	68.6	73.9	73.8	3.0	78.6	62.9	70.7	71.1	4.1
	EF7	own-2opp	79.3	72.1	75.4	75.4	2.6	77.9	69.3	75.0	74.4	2.2
	EF8	own-4opp	81.4	70.0	73.2	73.9	3.2	78.6	71.4	76.4	75.3	2.5
	EF9	own-8opp	79.3	70.0	74.3	74.4	2.9	82.1	72.9	75.7	76.2	3.1
type 2	EF10	If es < bs/2: EF8 else: EF1	79.3	70.7	75.0	75.0	2.9	80.0	68.6	75.4	75.2	3.5
	EF11	If es < bs/2: EF1 else: EF8	82.1	71.4	77.1	77.1	3.2	77.1	69.3	75.0	74.9	2.3
	EF12	If es < bs/4: EF8 else: EF1	82.9	72.1	73.6	74.9	3.1	77.9	70.0	74.3	74.4	2.3
	EF13	if es < bs/4: EF1 else: EF8	78.6	70.7	75.0	75.4	2.1	81.4	72.1	75.4	75.3	2.8
	EF14	If es < 3bs/4: EF8 else: EF1	79.3	70.0	75.4	75.0	3.0	80.7	72.1	76.4	76.1	2.3
	EF15	if es < 3bs/4: EF1 else: EF8	78.6	64.3	75.0	74.2	3.7	77.9	71.4	75.7	75.3	1.9
type 3	EF16	own/opp	81.4	70.7	73.6	74.5	3.3	80.7	68.6	76.1	76.1	3.4

[a]own is the number of moves for player itself; opp is the number of moves for opponent player; es is the number of empty space in the board; bs is the boardsize.