Heuristic Analysis

Guo Qiang

August 9, 2017

In the heuristic evaluation part, I implement three methods custom_score, custom_score2 and custom_score3. The aim is to compare improved_score and the three self-defined heuristic functions to see their functionality. First the basic analysis of these three methods is offered and the ideas of implementation are introduced. Second the winning rate data is listed to show the performance of the methods. Finally I recommend a method and gives the reasons.

1 Basic Ideas

Custom_score is inspired by improved_score in sample_players and add a penalty factor 2 to decrease the score of a player if his opponent has more legal moves.

Custom_score_2 is inspired by center_score. In center_score only the current player's concentration on center is evaluated. In custom_score_2, the basic idea is to integrated with the evaluation of current position of the players and the evaluation of their future moves. I think the current position of a player should be scored more if it is in the center position of the board since it usually has more direction available. And the length of his legal next moves is added to the evaluation of the player's future move possibilities. Likewise we subtract the opponent's score to add penalty factor.

Custom_score_3 is trying to calculate the length of a player's legal moves too but this time the next move is taken into account. It is assumed that a player will make his next move according to the utilities the move will bring to him. And the utility is positively related with the length of possible moves that the chosen move brings to him. So he will choose the move that brings him the maximum length of moves in the future. Also, the player's score is penalized by the opponent's score.

2 Performance Analysis

Table 1 gives the result of tournament. Compared with alpha beta improved score method, custom_score draws a tie with an identical win rate of 60% and the custom_score_2 is a little bit under-performed with a win rate of 58.6% and custom_score_3 wins the top giving a rate of 66.7%. But since the result differs every time we run the program, three samples are draws from execution and the average score of the four methods are 62.5%, 60.0%, 61.5% and 62.8% respectively.(See Table 2)

3 Conclusion and Recommendation

Here I recommend the score method custom_score_3 with the following reasons:

1. From the four samples we draw, for all the executions custom_score_3 wins the highest average score. It is of highly possibility that custom_score_3 will top other three methods if more samples are drawn.

2. For all the competitions custom_score_3 compete with benchmark method alpha beta improved method, it wins or draws a tie three out of four times. It is of high probability that the method will offer the best result.

3. Custom_score_3 is effective and it is also intuitive. Since moves made by players are all trying to let him to have more possible moves in the future and at the same time limit the moves of the opponent player. If we just judge the current move by the length of this move it brings us, the score function might be too short sighted. So I add the one step further score to the current utility so it reflects more of the score gaining in the future.

Table 1: Competition Result

Match	Opponent	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
		Won . Lost	Won . Lost	Won . Lost	Won . Lost
1	Random	9 . 1	9 . 1	10 . 0	8 . 2
2	MM_Open	7 . 3	6 . 4	4 . 6	4 . 6
3	MM_Center	7 . 3	8 . 2	7 . 3	10 . 0
4	MM_Improved	5 . 5	5 . 5	6 . 4	8 . 2
5	AB_Open	4 . 6	5 . 5	4 . 6	6 . 4
6	AB_Center	5 . 5	5 . 5	4 . 6	4 . 6
7	AB_Improved	5 . 5	4 . 6	6 . 4	6 . 4
	Win Rate:	60.0%	60.0%	58.6%	65.7%

Table 2: Three Samples

Match	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
1	60.0%	60.0%	58.6%	65.7%
2	61.4%	60.0%	62.9%	62.9%
3	64.3%	58.6%	61.4%	58.3%
4	64.3%	61.4%	62.9%	64.3%
average	62.5%	60.0%	61.5%	62.8%