Custom Heuristic Functions

Prabath Peiris

Heuristic functions

$$score = (my\_moves) - (2.0 \times opponent\_moves) \tag{1}$$

Match	Student vs	Result
1	Random	18 to 2
2	MM_Null	20 to 0
3	MM_Open	15 to 5
4	MM_Improved	13 to 7
5	AB_Null	19 to 1
6	AB_Open	13 to 7
7	AB_Improved	14 to 6
Evaluating Student using equation 1: 80.00%		

Equation 1 calculate the different between player 1 and player 2 available moves by doubling the available moves for the opponent. During the entire play, the coefficient that used here is constant. This makes the score value negative (opponent has more moves) much quicker. This makes the Player 1 (student) to make more aggressive moves towards winning the game and lead to 80% success.

$$score = D \times distance(player1, player2) \tag{2}$$

Match	Student vs	Result
1	Random	20 to 0
2	MM_Null	18 to 2
3	MM_Open	14 to 6
4	MM_Improved	14 to 6
5	AB_Null	15 to 5
6	AB_Open	13 to 7
7	AB_Improved	10 to 10
D = 2.0		
Evaluating Student using equation 2: 74.29%		

Equation 2 calculate the distance between the two player locations. This is the Euclidean distance between two points. The euclidean distance between two points is always a positive value. To make it more dynamic, when the distance is greater than 2, we make the distance a negative value $score = \{-score | score >$

2}. This allows the player to take more aggressive moves when there is a larger distance between pieces; however, this reduces the success rate of the player 1 compare the to the equation 1.

$$score = (progress) \times my\_moves - (1 - progress) \times opponent\_moves \quad (3)$$

$$progress = (my\_moves + opponent\_moves)/(board\_width * board\_height) \quad (4)$$

Match	Student vs	Result
1	Random	20 to 0
2	MM_Null	16 to 4
3	MM_Open	14 to 6
4	MM_Improved	13 to 7
5	AB_Null	19 to 1
6	AB_Open	13 to 7
7	AB_Improved	14 to 6
Evaluating Student using equation 3: 77.86%		

In equation 3 *progress* is define equation 4. When there are more moves available for both players the value of the *progress* scaling factor have a value close to one. This allow score value to be positive larger value. As the game progress, the value of the *progress* will keep getting smaller and smaller where when calculating *score* value will have a negative value. This allow the player 1 to choose more aggressive moves.

Choise of Heuristic Function

Comparing the success rated of all of the above functions, I can say the best heuristic function is given by the equation 1 (success rate 80.00%). This function produces negative values in very early in the game where the player 1 start taking more aggressive moves toward winning the game. This function also has much less complexity when compare to the other two equations.