Isolation: Heuristic analysis

Jaehwan Park

The game agent of isolation is developed by minimax algorithm, alpha-beta algorithm, iterative deepening and three heuristic score methods.

This report is focused on these three heuristics.

1) **custom_score: penalties of corners and only one move**

   Default score is calculated by distance from center and number of legal moves.

   Default score = square of opponent distance from center – square of player distance from center + player number of legal moves – opponent number of legal moves

   If player is close to center, the player has more chance to move any directions. So, player should stay on center and opponent should be on corner.

   If player can limit path of opponent, player have a great advantage on this game. So, it is calculated by difference of legal moves between player and opponent.

   If player is on corner or (±1. ±1) from corner, its penalty score is -5. Because player has only a few place to move on corner. For opponent, score is added by +5.

   If there is no move for player or opponent, this game will be ended. If player can limit only one place to move for opponent, player has a advantage and can attack opponent. If opponent has only one move, player get 10 points. But, if player has only one move, player get -10 penalty points.

   *Final score = default score - 5 * (is player on corner) + 5 * (is opponent on corner) – 10 * (player has only one move) + 10 * (opponent has only one move)*

2) **custom_score_2: far from corner and attack on same quadrant**

   *Score = difference of distance from center * factor + difference of number of legal moves*

   Custom_score_2 method introduces factor to adjust distance and number of legal moves.

   Let's divide 4 spaces for a board. If both player and opponent are on same quadrant, a player more close to center can attack its opponent. The player has a chance to push its opponent on the corner, and it can limit number of opponent's legal moves, and finally opponent has no place to move.

If both player and opponent stay on same quadrant, the factor is 1, and if not, the factor is 0.1.

3) custom_score_3: score of spaces

This heuristic focuses to spaces. If player stayed on center, player can have many chances to move, and if on corner, there is a few place to move. Each space has different chances to move.

For (y, x) position, score was given like below.

Space score = (half_board_width - abs(half_board_width - 0.5 - x)) *

(half_board_height - abs(half_board_height - 0.5 - y)

From this space score, below final score is calculated.

Score = sum of space scores of player legal moves –

sum of space scores of opponent legal moves

Here is a result of these three heuristics.

NUM_MATCHES = 25 # number of matches against each opponent

TIME_LIMIT = 150 # number of milliseconds before timeout

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	42	8	46	4	49	1	46	4
2	MM_Open	41	9	37	13	38	12	40	10
3	MM_Center	42	8	43	7	45	5	46	4
4	MM_Improved	41	9	39	11	37	13	40	10
5	AB_Open	26	24	23	27	23	27	27	23
6	AB_Center	28	22	27	23	30	20	33	17
7	AB_Improved	25	25	18	32	27	23	28	22

| | Win Rate: | 70.0% | | 66.6% | | 71.1% | | 74.3% | |

1) custom_score: penalties of corners and only one move

AB_Custom: alpha-beta algorithm + custom score heuristic

AB_Improved: alpha-beta algorithm + improved heuristic (difference of number of legal moves)

Winning rate (66.6%) of AB_Custom is lower than winning rate (70.0%) of AB_Improved. AB_Custom has adjustment of corner and only one move, but it is not effective if compared with AB_Improved.

Even though adjustment is adopted, it can decrease performance not like expectation.

2) custom_score_2: far from corner and attack on same quadrant

AB_Custom_2: alpha-beta algorithm + custom score 2 heuristic

AB_Open: alpha-beta algorithm + open heuristic (number of player legal moves)

Winning rate (71.1%) of AB_Custom_2 is similar with winning rate (70.0%) of AB_Improved. Just 1 ~ 2 % difference is too small to conclude which score method is better. Also, 350 samples are too small, because whenever tested, winning rate is changed. If NUM_MATCHS increase like 500, this result is more accurate, but the number is limited by environment of time and computer performance.

Also, AB_Custom_2 is defeated by AB_Open (23 : 27). At least, we cannot say that AB_Custom_2 is better than AB_Open.

From this result, factor of difference of distance from center and number of legal moves cannot increase performance significantly.

3) custom_score_3: score of spaces

AB_Custom_3: alpha-beta algorithm + custom score 3 heuristic

Winning rate (74.3%) of AB_Custom_3 is better than winning rate (70.0%) of AB_Improved. Also, AB_Custom_3 defeat all agents significantly.

AB_Custom_3 is best agent for the Isolation game among given agents. So, custom score 3 heuristic is most effective among all given heuristics.

<Recommendation>

I recommend AB_Custom_3 (custom score 3 heuristic: score of spaces) agent for this game.

First, this agent has best winning rate as 74.3% in the tournament. It reveals that this agent is strongest among all given and three custom agents.

Second, this heuristic wasn't defeated by any agents. It means that this agent has no weak point against all given agents at least.

Third, this heuristic has advantage from heuristics of AB_Improved and AB_Center agents without long time consumption. If space score is 1 for all spaces, this heuristic is exactly equal to improved heuristic. So, this heuristic is update version of improved heuristic by adding space score, and improved the performance by defeating AB_Improved.

Also, this heuristic gives high score on center and low score on corner. So, this heuristic has characteristic of center heuristics, and has better performance than center heuristics by defeating AB_Center.

Before applying cache for space score, AB_Custom_3 couldn't win AB_Improved. I think that custom score 3 heuristic is better than improved heuristic, so assumed that time is key. If heuristic has long processing time, alpha-beta algorithm cannot go deeper by iterative deepening. Cache increase processing time of this heuristic dramatically, and AB_Custom_3 finally defeated AB_Improved.

So, this heuristic has good performance with low processing time.