UNIVERSITY OF MALAYA

Heuristic Analysis Report

For an Adversarial Game Playing Agent for Isolation

Name: Lukman Hakim Bin Azahari

Matric Number: WID170026 @ 17170546/2

Lecturer: Professor Dr. Loo Chu Kiong

Table of Contents

SYNOPSIS

The project studies the adversarial search agent that plays “Isolation”. This project is focusing on the heuristics to use in A* Search for minimax and alphabeta pruning. Isolation is a deterministic, two-player game of perfect information in which the players alternate turns moving a single piece from one cell to another on a board. Whenever either player occupies a cell, that cell becomes blocked for the remainder of the game. The first player with no remaining legal moves loses, and the opponent is declared the winner. This project uses a version of Isolation where each agent is restricted to L-shaped movements (like a knight in chess) on a rectangular grid (like a chess or checkerboard). The agents can move to any open cell on the board that is 2-rows and 1-column or 2- columns and 1-row away from their current position on the board. Movements are blocked at the edges of the board (the board does not wrap around), however, the player can "jump" blocked or occupied spaces (just like a knight in chess). Additionally, agents will have a fixed time limit each turn to search for the best move and respond. If the time limit expires during a player's turn, that player forfeits the match, and the opponent wins. These rules are implemented in the isolation. Board class provided in the repository.

CUSTOM HEURISTICS

1. My student7 Heurisitic

The heuristic is based on the logic that the difference between the player’s moves and opponent’s moves. It can be mathematically as :

len(my available moves)-2* len (available opponent moves)

CODE USED

```
if game.is_loser(player):
    return float("-inf")

if game.is_winner(player):
    return float("inf")

player_legal_moves = game.get_legal_moves(player)
player_total_moves = len(player_legal_moves)

for move in player_legal_moves:
    board = game
    board = game.forecast_move(move)
    player_total_moves += len(board.get_legal_moves())

opp_legal_moves = game.get_legal_moves(game.get_opponent(player))
opp_total_moves = len(opp_legal_moves)

for move in opp_legal_moves:
    board = game
    board = game.forecast_move(move)
    opp_total_moves += len(board.get_legal_moves())

return float(player_total_moves - 2 * opp_total_moves)
```

Evaluating Heuristics

The tournament.py script is used to evaluate the effectiveness of heuristic. The script measures relative performance of player in a round-robin tournament against several other pre-defined agents.
The performance of time-limited iterative deepening search is hardware dependent (faster hardware is expected to search deeper than slower hardware in the same amount of time). The script controls for these effects by also measuring the baseline performance of an agent called "ID_Improved" that uses Iterative Deepening and the improved_score heuristic from sample_players.py.
The tournament opponents are listed below:

- Random: An agent that randomly chooses a move each turn.
- MM_Null: CustomPlayer agent using fixed-depth minimax search and the null_score heuristic
- MM_Open: CustomPlayer agent using fixed-depth minimax search and the open_move_score heuristic
- MM_Improved: CustomPlayer agent using fixed-depth minimax search and the improved_score heuristic
- AB_Null: CustomPlayer agent using fixed-depth alpha-beta search and the null_score heuristic
- AB_Open: CustomPlayer agent using fixed-depth alpha-beta search and the open_move_score heuristic
- AB_Improved: CustomPlayer agent using fixed-depth alpha-beta search and the improved_score heuristic
- ID_Improved: CustomPlayer agent using iterative alpha-beta search and the improved_score heuristic
- Student7: CustomPlayer agent using iterative alpha-beta search and the custom_score_2

Since, running only a few matches gave different results, the number of matches is set to 5, with all group members agreed to it because faster execution time. Timeout value is 150 which is the default value

RESULTS

Results are compared between the baseline default (ID_Improved), 7 students and 3 of my group members. The results as follow:

Agent	Performance	Rank	Rank (Group Members
ID_Improved	55.71%	5	
Group Member3(AQIFF)	71.43%	1	1
Student7(LUKMAN)	64.29	4	4
Group Member1(FAWWAZ)	65.71%	2	2
Group Member2(SYAFIK)	65.00%	3	3

The custom heuristic performs better than Id_Improved by a reasonable margin which can be seen in the table.

APPENDICES

This is where performance of the custom heuristic function by comparing the strength of an agent using iterative deepening search with alpha-beta pruning against the strength rating of agents using other heuristic functions. ID_Improved agent is a baseline by measuring the performance of a basic agent using Iterative Deepening and the improved heuristic on the hardware. The “Student” agent except Student 7 as student 7 is using a different algorithm which is stated in the code as custom_score_2 is using the heuristic analysis given in the example.

```
*************************

 Evaluating: ID_Improved

*************************


Playing Matches:

----------

  Match 1: ID_Improved vs   Random          Result: 18 to 2

  Match 2: ID_Improved vs   MM_             Result: 12 to 8

  Match 3: ID_Improved vs   MM_Open         Result: 8 to 12

  Match 4: ID_Improved vs MM_Improved       Result: 8 to 12

  Match 5: ID_Improved vs   AB_Null         Result: 12 to 8

  Match 6: ID_Improved vs   AB_Open         Result: 10 to 10

  Match 7: ID_Improved vs AB_Improved       Result: 10 to 10


Results:

----------

ID_Improved       55.71%
```

```
*************************
 Evaluating: Student7
*************************

Playing Matches:
----------
  Match 1: Student7   vs   Random          Result: 14 to 6
  Match 2: Student7   vs   MM_             Result: 16 to 4
  Match 3: Student7  vs   MM_Open          Result: 12 to 8
  Match 4 Student7   vs MM_Improved        Result: 8 to 12
  Match 5 Student7   vs   AB_Null          Result: 14 to 6
  Match 6: Student7  vs   AB_Open          Result: 15 to 5
  Match 7 Student7   vs AB_Improved        Result: 11 to 9

Results:
----------
Student7        64.29%
```