CSE422: Artificial Intelligence

Lab Assignment 03: MiniMax using Alpha-beta Pruning

Problem 1: Chess Masters [7]

In the 2018 Chess World Championship finals, Magnus Carlsen and Fabiano Caruana faced off in a series of 12 classical games. Although both players demonstrated some of the finest chess skills we have ever seen, all 12 games ended in draws [Carlsen later won in the tie-breaks]. However, analysis using super computer chess engines like Stockfish and Leela Chess Zero (LCZero) showed that both players had missed some opportunities in several positions. This happens because we humans cannot evaluate every possible move and calculate its long-term consequences that only become obvious 20, 25, or 30 moves later. Chess engines, on the other hand, can systematically explore millions of positions and identify the best move with high precision.

Traditional chess engines like Stockfish use a highly optimized version of alpha-beta pruning to efficiently search through vast game trees [although neural networks are becoming increasingly popular]. In this problem, we will implement alpha-beta pruning to simulate a simplified game of chess.

Game Description

We will simulate **four** games using minimax algorithms with alpha-beta pruning.
- **Players:** Magnus Carlsen [0] and Fabiano Caruana [1].
- **Turn based gameplay:** Each player will take turns to make a move.
- **Turns:** The player specified by the first input number will take the first turn as the maximizing player. The other player will be the minimizing player. For example, if input is 0, then in Game-1 Carlsen will be Max make the first move, and Caruana will be Min and make the second move, and so on. In Game-2, roles will reverse; so Caruana will play as Max and make the first move. Similarly, Carlsen will play as Max in Game-3 and Caruana will be Max in Game-4.
- **Moves in each turn:** Although in a game of chess a player has many possible moves, in this problem each player will have only two possible moves in each turn. So **branching factor = 2.**
- **Game depth:** The maximum depth of the game tree will be **5.** After that, you will have to create an appropriate number of leaf nodes and assign them utility values.
- **Winner:** For each game, you will decide on the winner by using the minimax algorithm with alpha beta pruning and print it out. If the final value is positive, then Max is the winner. If it is negative, then Min is the winner; and the game is drawn if the value is 0. After four games, you will declare the overall winner. Check the sample output for better understanding.

Utility Function
You will assign utility values to each leaf node using the following function:

$$utility = strength(maxV) - strength(minV) + (-1)^i \frac{random(1,10)}{10}$$

Here,

- **maxV:** base strength of the Max player in a game, taken as input
- **minV:** base strength of the Min player in a game, taken as input
- **i:** either 0 or 1, chosen at random
- **random(1,10):** a random integer between 1 and 10, inclusive

The strength function is defined as:

$$strength(x) = log_2(x + 1) + \frac{x}{10}$$

The base strengths will be taken as input, and the same values will be used in all four games. For example, suppose the base strength of Carlsen and Caruana are given as 10 and 8, respectively, and Carlsen goes first. Then, in game-1, maxV = 10 and minV = 8. In game-2 roles reverse, so maxV = 8 and minV = 10. Similar for games 3 and 4. Check the sample input for better understanding.

Sample I/O

[Note that, due to the random function, the output values are unlikely to match exactly.]

Input 1

```
Enter starting player for game 1 (0 for Carlsen, 1 for Caruana): 0
Enter base strength for Carlsen: 9
Enter base strength for Caruana: 8
```

Explanation: The first input [0] denotes that in the first game Carlsen will play as Max and make the first move, while Caruana will play as Min. In game 2 Caruana will play as Max and Carlsen will play as Min. The second [9] and third [8] inputs denote that Carlsen's base strength value is 9 while Caruana's is 8. So, in game 1, maxV = 9, minV = 8. In game 2, maxV = 8, and minV = 9.

Output 1

```
Game 1 Winner: Magnus Carlsen (Max) (Utility value: 0.45)
Game 2 Winner: Magnus Carlsen (Min) (Utility value: -0.15)
Game 3 Winner: Magnus Carlsen (Max) (Utility value: 0.15)
Game 4 Winner: Fabiano Caruana (Max) (Utility value: 0.15)

Overall Results:
Magnus Carlsen Wins: 3
Fabiano Caruana Wins: 1
Draws: 0
Overall Winner: Magnus Carlsen
```

Input 2

```
Enter starting player for game 1 (0 for Carlsen, 1 for Caruana): 1
Enter base strength for Carlsen: 9
Enter base strength for Caruana: 8
```

Explanation: Game-1: Caruana Max, Carlsen Min, maxV = 8, minV = 9. Game-2: Carlsen Max, Caruana Min, maxV = 9, minV = 8.

Output 2

```
Game 1 Winner: Magnus Carlsen (Min) (Utility value: -0.15)
Game 2 Winner: Magnus Carlsen (Max) (Utility value: 0.35)
Game 3 Winner: Magnus Carlsen (Min) (Utility value: -0.15)
Game 4 Winner: Magnus Carlsen (Max) (Utility value: 0.05)


Overall Results:
Magnus Carlsen Wins: 4
Fabiano Caruana Wins: 0
Draws: 0
Overall Winner: Magnus Carlsen
```

Input 3

```
Enter starting player for game 1 (0 for Carlsen, 1 for Caruana): 0
Enter base strength for Carlsen: 8
Enter base strength for Caruana: 10
```

Output 3

```
Game 1 Winner: Fabiano Caruana (Min) (Utility value: -0.09)
Game 2 Winner: Fabiano Caruana (Max) (Utility value: 0.79)
Game 3 Winner: Fabiano Caruana (Min) (Utility value: -0.19)
Game 4 Winner: Fabiano Caruana (Max) (Utility value: 0.79)


Overall Results:
Magnus Carlsen Wins: 0
Fabiano Caruana Wins: 4
Draws: 0
Overall Winner: Fabiano Caruana
```

Input 4

```
Enter starting player for game 1 (0 for Carlsen, 1 for Caruana): 0
Enter base strength for Carlsen: 9
Enter base strength for Caruana: 8.75
```

Output 4

```
Game 1 Winner: Magnus Carlsen (Max) (Utility value: 0.56)
Game 2 Winner: Magnus Carlsen (Min) (Utility value: -0.26)
Game 3 Winner: Fabiano Caruana (Min) (Utility value: -0.04)
Game 4 Winner: Fabiano Caruana (Max) (Utility value: 0.64)


Overall Results:
Magnus Carlsen Wins: 2
Fabiano Caruana Wins: 2
Draws: 0
Overall Winner: Draw
```

Problem 02: Chess Noobs with Magic [3]

Two players, **Light** and **L,** will play ONE game of chess but with a twist: whoever goes first has the magical ability to control the other player's mind. This way, if Light goes first, he can control which of the two possible moves L will make. L can do the same if he goes first. However, this magical ability comes at a cost **c** that will be subtracted from the final utility value, and also there will be the stigma of cheating!
Except for the twist, the game settings stay the same as before. Note that this time there will be only one game, not four. Only the MAX player can use mind control magic.

Sample I/O

Input 1

```
Enter who goes first (0 for Light, 1 for L): 0
Enter the cost of using Mind Control: 0.25
Enter base strength for Light: 10
Enter base strength for L: 8
```

Output 1

```
Minimax value without Mind Control: 0.79
Minimax value with Mind Control: 1.09
Minimax value with Mind Control after incurring the cost: 0.84

Light should NOT use Mind Control as the position is already winning.
```

Explanation: Here, Light will win with or without the mind control magic, so there is no need to use mind control magic and get his hands dirty.

Input 2

```
Enter who goes first (0 for Light, 1 for L): 0
Enter the cost of using Mind Control: 0.5
Enter base strength for Light: 9
Enter base strength for L: 9.5
```

Output 2

```
Minimax value without Mind Control: -0.32
Minimax value with Mind Control: 0.68
Minimax value with Mind Control after incurring the cost: 0.18

Light should use Mind Control.
```

Explanation: Here, Light will lose without the mind control but will win if he uses it, so he should.

Input 3

```
Enter who goes first (0 for Light, 1 for L): 0
Enter the cost of using Mind Control: 0.5
Enter base strength for Light: 7
Enter base strength for L: 9
```

Output 3

```
Minimax value without Mind Control: -0.42
Minimax value with Mind Control: 0.38
Minimax value with Mind Control after incurring the cost: -0.12


Light should NOT use Mind Control as the position is losing either
way.
```

Explanation: Here, Light will lose with or without the mind control, so there is no point in using the magic. Just get on with it!

Input 4

```
Enter who goes first (0 for Light, 1 for L): 0
Enter the cost of using Mind Control: 1
Enter base strength for Light: 9
Enter base strength for L: 8
```

Output 4

```
Minimax value without Mind Control: 0.45
Minimax value with Mind Control: 0.75
Minimax value with Mind Control after incurring the cost: -0.25


Light should NOT use Mind Control as it backfires.
```

Explanation: In this case, Light will win without the mind control but will lose if he uses it due to cost being high, so he must not use the weird magic.

Problem 03: Food for thought [0]

● In the first problem, will the stronger player always win? Does having the first move matter? First come up with an answer through reasoning only. Then run the algorithm with several inputs and see what happens.
● In the first problem, the utility function includes a random component. Do you think it is a wise choice? Why or why not? How much can the random factor affect the outcomes?
● Consider the implications of increasing the game tree depth or changing the branching factor. How would these changes affect the performance of the alpha-beta pruning algorithm and the accuracy of the final outcome?
● In minimax algorithms, is the first player always a maximizer? Can you think of any scenario where it will not be the case?
● Does alpha-beta pruning work in stochastic scenarios?