A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Santhosh Shanmugasundaram |
| *Student Number* | Sba24100 |
| *Module Title* | AI Concepts to Implementation |
| *Assessment Title* | CA2 |
| *Assessment Due Date* | 5th January 2025 |
| *Date of Submission* | 5th January 2025 |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

CA1 – AI Concepts to Implementation

Contents

[1. Task1: Two Artificial Intelligence Strategy that are used to play Chess. 2](#_Toc182156774)

[1.1 Introduction: 2](#_Toc182156775)

[1.2 Why Chess for AI Research: 2](#_Toc182156776)

[1.3 Why Artificial Intelligence for Chess Game: 2](#_Toc182156777)

[1.4 Working Principles of AI Chess Engine: 2](#_Toc182156778)

[1.5 Min-Max Searching 3](#_Toc182156779)

[1.5.1 Key Terminologies 3](#_Toc182156780)

[1.5.2 Steps 3](#_Toc182156781)

[1.5.3 Example 4](#_Toc182156782)

[1.5.4 Key Properties of Min-Max Algorithm 4](#_Toc182156783)

[1.6 Alpha-Beta Pruning 5](#_Toc182156784)

[1.6.1 Key Terminologies 5](#_Toc182156785)

[1.6.2 Key Steps 5](#_Toc182156786)

[1.6.3 Condition for Pruning 5](#_Toc182156787)

[1.6.4 Example 6](#_Toc182156788)

[2 Task2: 7](#_Toc182156789)

[2.1 Find the Best Route from Tipperary to Sligo 7](#_Toc182156790)

[2.1.1 Route Finding Using Depth-First Search (DFS) 7](#_Toc182156791)

[2.1.2 Route Finding Using Dijkstra’s Algorithm 7](#_Toc182156792)

[2.1.3 Route Finding Using A\* Algorithm 8](#_Toc182156793)

[2.1 Conclusion 8](#_Toc182156794)

[2.2 Two AI Strategies Used in Modern Games 8](#_Toc182156795)

[2.2.1 Monte Carlo Tree Search (MCTS) in AlphaGO Zero: 9](#_Toc182156796)

[2.2.2 Behavior Tree Algorithm in Halo 2 10](#_Toc182156797)

[3 Task3: 10](#_Toc182156798)

[3.1 Characteristics of AIBO 11](#_Toc182156799)

[3.2 AIBO – Scope for Improvement 12](#_Toc182156800)

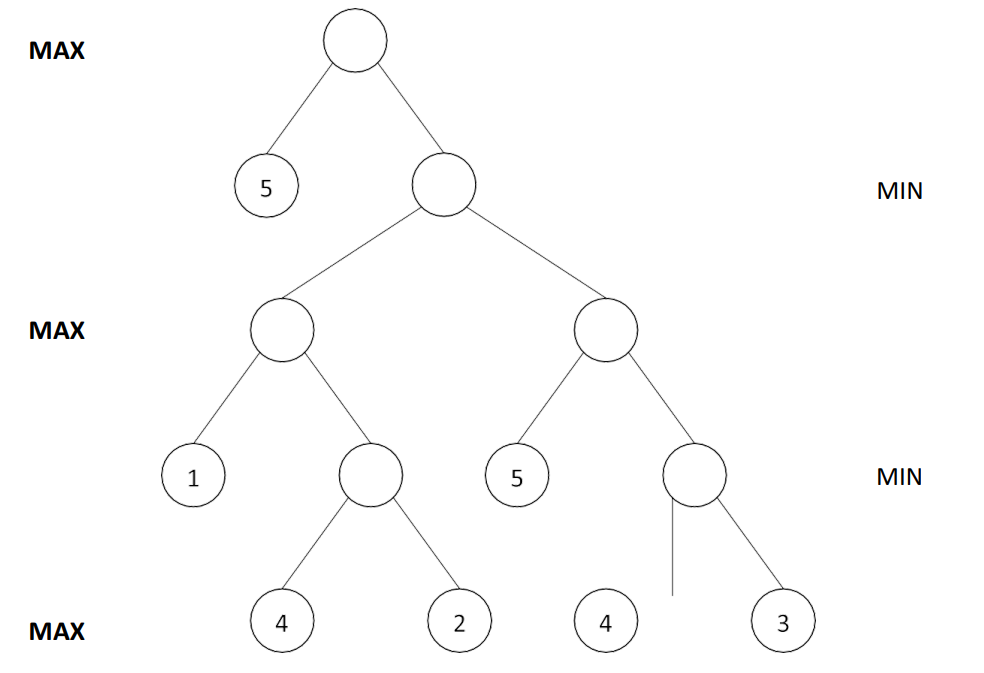
[4 References 12](#_Toc182156801)

[5 GitHub Link 15](#_Toc182156802)

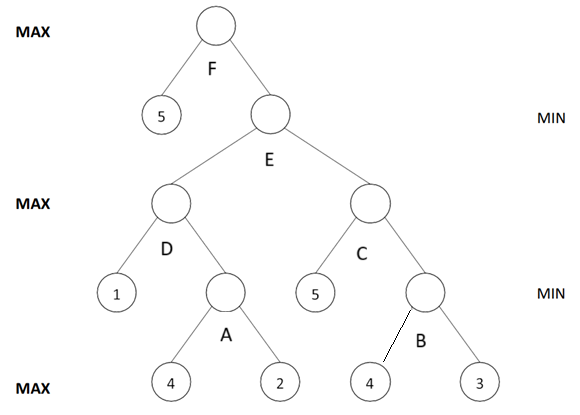
# Question1: Min-Max Algorithm with Alpha Beta Pruning.

## Task1: Determine Min-Max Value

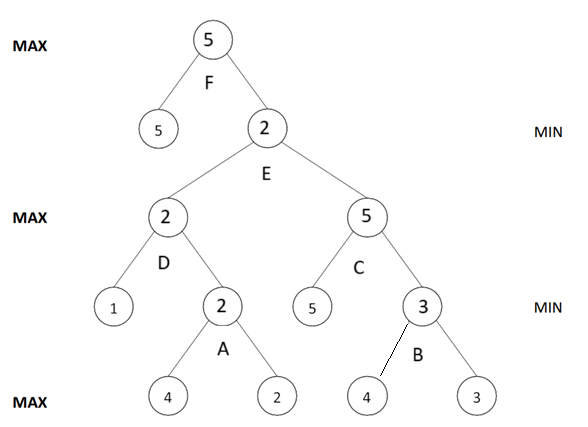
Task is to implement the min-max algorithm for the below tree and find the values at each node.

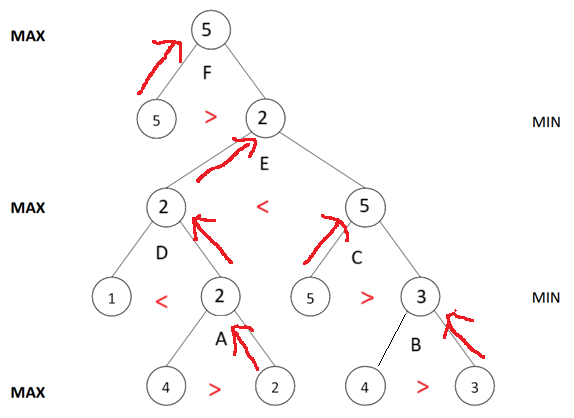


The Min-Max algorithm involves several key steps, executed recursively until the optimal move is determined. This algorithm applies Depth First Search (DFS); therefore, we must go till the terminal node. Each step involved in execution of min-max algorithm for the above tree is listed below and to start with, each node has been named from A to F starting from leaf node at bottom to top.



1. Start from leaf node, which is at the bottom, let’s try to compare the leaf values corresponding to node A.
2. Since node A and B corresponds to min node, we try to fetch the lowest of value from its associated leaf value.
3. Therefore, A is assigned as 2 (lowest of 4 & 2) and B is assigned as 3 (lowest of 4 and 3).
4. Next nodes are D and C which corresponds to Max node; therefore, we try to fetch the highest value from its associated child node/leaf node.
5. Node D is assigned as 2 (highest of 1 & 2) and C get’s 5 (highest of 5 & 3).
6. Node E corresponds to Min node; therefore, we take the lowest of D and C which results in value 2 (lowest of 2 & 5).
7. Now we arrive to the topmost/root node of the tree which contains a leaf node with value 5 and child node E whose value is 2. Taking the max value of 5 and 2, we assign 5 to F.





## Task2: Determine Pruned Nodes using Alpha Beta Pruning Technique

Alpha-beta pruning is a technique used to improve the efficiency of the minimax algorithm by eliminating the evaluation of certain nodes that has no significance to the result.

**Alpha (α)**:   
Represents the best value the maximizing player can guarantee so far. The initial value of alpha is -Infinity (-∞).

**Beta**: Represents the best value the minimizing player can guarantee so far. The initial value of Beta is +Infinity (+∞)

## Min-Max Searching

The Mini-Max algorithm is a decision-making algorithm used particularly in game theory and computer games. In a two-player game, one player is the maximiser, aiming to maximize their score, while the other is the minimizer, aiming to minimize the maximiser’s score. The algorithm operates by evaluating all possible moves for both players, predicting the opponent's responses, and choosing the optimal move to ensure the best possible outcome.

### Key Terminologies

* Maximizing Player (Max):   
  Chooses the move that leads to the highest possible utility value, assuming the opponent will play optimally.
* Minimizing Player (Min):  
  Selects the move that results in the lowest possible utility value for the maximiser, assuming the opponent will play optimally.

The interplay between these two players is central to the Min-Max algorithm, as each player attempts to outthink and counter the other's strategies.

### Steps

The Min-Max algorithm involves several key steps, executed recursively until the optimal move is determined. This algorithm applies Depth First Search (DFS); therefore, we must go till the terminal node.

**Step1**: Generate the Game Tree  
Create a tree structure representing all possible moves at the current state of game.

**Step2**: Evaluate Terminal States  
Assign utility values to the terminal nodes of the game tree.

**Step3**: Propagate Utility Values Upwards  
Starting from the terminal nodes, propagate the utility values upwards through the tree.  
For each non-terminal node, if it's the maximizing player's turn, select the maximum value from the child nodes.  
If it's the minimizing player's turn, select the minimum value from the child nodes.

**Step4**: Select Optimal Move  
At the root of the game tree, the maximizing player selects the move that leads to the highest utility value. (GeeksforGeeks)

### Example

A simple game where each player can choose between two moves at each turn. Here's a basic game tree:

Max

/ \

Min Min

/ \ / \

+1 -1 0 +1

At the leaf nodes, the utility values are +1, -1, 0, and +1.

The minimizing player will choose the minimum values from the child nodes: -1 (left subtree) and 0 (right subtree).  
The maximizing player will then choose the maximum value between -1 and 0, which is 0.

Thus, the optimal move for the maximizing player, considering optimal play by the minimizer, leads to a utility value of 0.

### Key Properties of Min-Max Algorithm

**Complete**- Min-Max algorithm is Complete. It will find a solution (if exist), in the finite search tree.

**Optimal**- Min-Max algorithm is optimal if both opponents are playing optimally.

**Time complexity**- As it performs DFS for the game-tree, so the time complexity of Min-Max algorithm is O(bm), where b is branching factor of the game-tree, and m is the maximum depth of the tree.

**Space Complexity**- Space complexity of Mini-max algorithm is also like DFS which is O(bm).

## Alpha-Beta Pruning

Alpha-beta pruning is a technique used to improve the efficiency of the minimax algorithm by reducing the number of nodes that need to be evaluated in a game tree. In minimax, every possible move and countermove is evaluated, but many of these moves don’t affect the final decision.   
Alpha-beta pruning works by eliminating branches that are guaranteed to not influence the outcome. If the algorithm finds that a particular branch can’t improve the final outcome for either player, it “prunes” that branch, meaning it stops further evaluation of that part of the tree. By doing so, the algorithm avoids unnecessary calculations, allowing it to search deeper into the tree more quickly. (Saxena)

### Key Terminologies

* Alpha:   
  Represents the best value the maximizing player can guarantee so far. The initial value of alpha is -Infinity (-∞).
* Beta: Represents the best value the minimizing player can guarantee so far. The initial value of Beta is +Infinity (+∞)

### Key Steps

Since Alpha-Beta Pruning is an extension of Min-Max algorithm, following steps are in addition to the steps followed by min-max algorithm.

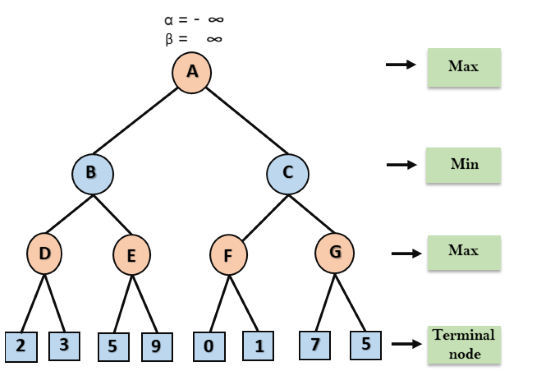
* For all maximizing players turn, update the Alpha value with node value.
* For all minimizing players turn, update the Beta value with node value.
* When backtracking the tree, node value will be passed to Alpha and Beta of upper node.
* When moving from top to bottom in the tree, pass the alpha and Beta value to all child nodes.

### Condition for Pruning

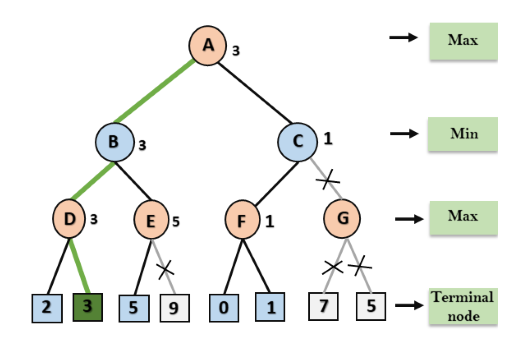
* The minimizing player finds a value that is lower than or equal to alpha (β ≤ α) then prune the branch.
* The maximizing player can prune a branch when they find a value higher than or equal to beta (α ≥ β)

### Example

Consider the below search tree with root node and 2 child nodes and terminal node.



Following the conditions and steps mentioned above, we arrive at the below result. Final result will be same after pruning four branches.



# References

(Rolls)

Rolls, Chris. “LinkedIn.” *Linkedin.com*, 14 July 2024, www.linkedin.com/pulse/thoughts-ai-chess-quality-engineering-chris-rolls-5eehc/. Accessed 8 Nov. 2024.

Available at:   
<https://www.linkedin.com/pulse/thoughts-ai-chess-quality-engineering-chris-rolls-5eehc/>

(Degni)

Degni, Riccardo. “The Ultimate Checkmate: AI and Chess Engines.” *Codemotion Magazine*, 20 Feb. 2023, www.codemotion.com/magazine/ai-ml/the-ultimate-checkmate-ai-and-chess-engines/.

Available at:   
<https://www.codemotion.com/magazine/ai-ml/the-ultimate-checkmate-ai-and-chess-engines/>

(GeeksforGeeks)

GeeksforGeeks. “Mini-Max Algorithm in Artificial Intelligence.” *GeeksforGeeks*, GeeksforGeeks, 27 June 2024, www.geeksforgeeks.org/mini-max-algorithm-in-artificial-intelligence/.

Available at:   
<https://www.geeksforgeeks.org/mini-max-algorithm-in-artificial-intelligence/>

(Saxena)

Saxena, Abhimanyu. “Alpha Beta Pruning in Artificial Intelligence.” *Applied AI Blog*, 25 Sept. 2024, www.appliedaicourse.com/blog/alpha-beta-pruning-in-artificial-intelligence/. Accessed 9 Nov. 2024.

Available at:   
<https://www.appliedaicourse.com/blog/alpha-beta-pruning-in-artificial-intelligence/>

(Ravikiran AS)

Ravikiran AS. “A\* Algorithm in Artificial Intelligence You Must Know in 2022.” *Simplilearn.com*, 22 July 2021, www.simplilearn.com/tutorials/artificial-intelligence-tutorial/a-star-algorithm.

Available at:   
<https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/a-star-algorithm>

(Sony Group Corporation)

Sony Group Corporation. “ERS-1000 | Help Guide | Getting to Know Aibo.” *Sony.net*, 2018, helpguide.sony.net/aibo/ers1000/v1/en-us/contents/TP0001970093.html. Accessed 10 Nov. 2024.

Available at:   
<https://helpguide.sony.net/aibo/ers1000/v1/en-us/contents/TP0001970093.html>

(Engati)

Engati. “What Is AI in Gaming Industry (40+ AI Powered Games in 2024 ).” *Engati*, 11 July 2024, www.engati.com/blog/ai-for-gaming#toc-what-are-the-benefits-of-ai-in-games-. Accessed 10 Nov. 2024.

Available at:   
[https://www.engati.com/blog/ai-for-gaming#toc-what-are-the-benefits-of-ai-in-games-](https://www.engati.com/blog/ai-for-gaming%23toc-what-are-the-benefits-of-ai-in-games-)

(Radke)

Radke, Parag. “Monte Carlo Tree Search: A Guide | Built In.” *Builtin.com*, 20 July 2023, builtin.com/machine-learning/monte-carlo-tree-search.

Available at:   
<https://builtin.com/machine-learning/monte-carlo-tree-search>

(Hui)

Hui, Jonathan. “Monte Carlo Tree Search (MCTS) in AlphaGo Zero.” *Medium*, 17 July 2020, jonathan-hui.medium.com/monte-carlo-tree-search-mcts-in-alphago-zero-8a403588276a.

Available at:   
<https://jonathan-hui.medium.com/monte-carlo-tree-search-mcts-in-alphago-zero-8a403588276a>

(Denis)

Denis, Sion. “The Craft of Behavior Trees - Sion Denis - Medium.” *Medium*, 25 Jan. 2024, medium.com/@sion\_denis/the-craft-of-behavior-trees-ab6b181ce21a. Accessed 10 Nov. 2024.

Available at:   
<https://medium.com/@sion_denis/the-craft-of-behavior-trees-ab6b181ce21a>

(Valdes)

Valdes, Robert. “The Artificial Intelligence of Halo 2.” *HowStuffWorks*, 17 Nov. 2004, electronics.howstuffworks.com/halo2-ai.htm.

Available at:   
<https://electronics.howstuffworks.com/halo2-ai.htm>

# GitHub Link

<https://github.com/santhosh-sba24100/CA1---AI-Concepts-to-Implementation>