A Decision Tree-Based Approach to Optimal Move Selection in Chess Middlegame Using Minimax Algorithm with Stockfish Engine

Ibrahim Ihsan Rasyid - 135220181

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

113522018@std.stei.itb.ac.id

*Abstract*—*Chess is*.

*Keywords*—chess engine, middlegame, minimax algorithm, tree.

# I. Introduction

Chess is one of the most popular board games since its discovery around the 7th century, evolving into competitive play on the world stage for the first time in 1834 up to the present modern era. In chess, the term “middlegame” is recognized, referring to the phase where majority of player’s pieces and pawns have been developed. Most of the times, players find it challenging to determine the best next move due to the multitude of possibilities that can unfold after each move. Poor move calculation can have long-term effects and lead to defeat in the end. It is not easy for humans to perform calculations to determine the best move.

Therefore, humans strive to develop chess engines to assist in analyzing previously completed games. Numerous chess engines have already been developed, so many that since 2010, the Top Chess Engine Championship (TCEC) has been held, which is a competition among chess engines. To date, TCEC has conducted 25 seasons, with one of the chess engines, Stockfish, emerging victorious in 15 seasons and currently reigning as the best chess engine.

The algorithm employed by Stockfish, eventually replaced by NNUE (Efficiently Updated Neural Networks) due to its widespread adoption across various chess engines, includes the Minimax Algorithm and Alpha-Beta Pruning. This paper focuses solely on the Minimax Algorithm and the application of tree concepts in constructing this algorithm.

The Minimax Algorithm is a decision-making rule designed to minimize the potential for the worst-case scenario. This algorithm is commonly applied to zero-sum games such as tic-tac-toe, checkers, shogi, and of course chess. In the context of these games, minimax entails maximizing the potential for the smallest advantages. Simply put, one player makes moves to gain maximum advantage for themselves, while the opposing player makes moves to diminish the advantage of their opponent. Hence, the evaluation of the board position in a game typically holds a positive value for one player and negative value for their opponent. In chess, the advantage of the white player is reflected in a positive evaluation, and vice versa. An infinite value signifies that one player can win the game within a few moves, usually written as M4, means there is a checkmate in 4 moves.

# II. Theoretical Foundation

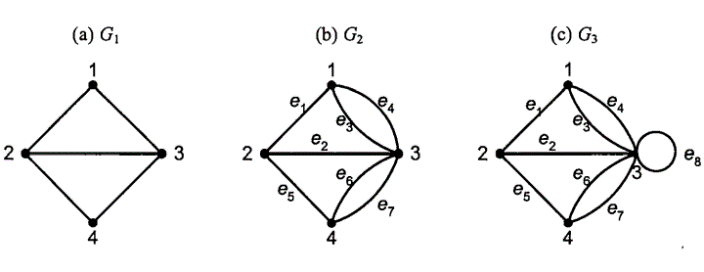
## A. Chess

Chess is a board game which is played between two opponents (light-colored/white and dark-colored/black) who move their pieces alternately, with the objective is to place the opponent’s king in “under attack” in such a way that the opponent does not have any legal move. The player who achieves this goal is said to have “checkmated” the opponent’s king and to have won the game, while the opponent king whose king has been checkmated has lost the game.

The middlegame in chess is the phase between the opening and the endgame. Typically, the middlegame begins when players have developed a significant portion of their pieces, brought their king to safety, and concludes when only a few pieces remain on the board. Middlegame theory is considered less developed compared to opening and endgame theories. This is because the positions of pieces in the middlegame often vary significantly from one to another, making it much more challenging to memorize the various variations compared to openings and endgames.

## B. Graph

Graph G is defined as a pair of sets (V, E), expressed using the notation G = (V, E), where V is a non-empty set of vertices/nodes, and E is a set of edges connecting pairs of nodes. Geometrically, a graph is depicted as a collection of nodes in a two-dimensional plane connected by a set of lines



*Fig. 2.1 (a) simple graph, (b) unsimple graph, (c) pseudo-graph*

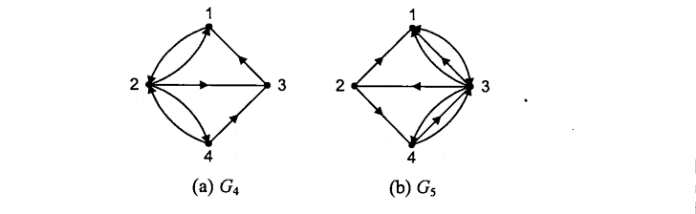
Based on the presence or absence of loops or multiple edges in a graph, graphs can generally be classified into two types:

1. Simple Graph  
   A simple graph is a graph that does not have loops or multiple edges. Examples can be seen in Fig. 2.1 (a)
2. Unsimple Graph  
   An unsimple graph is a graph that can have loops and/or multiple edges. Here we can classify an unsimple graph into:

* Multi-graph  
  Multigraph is a graph that contains multiple egdes, those are two or more edges that connect the same pair of nodes. Examples can be seen in Fig. 2.1 (b)
* Pseudo-graph  
  Pseudo-graph is a graph that contains loops, that is an edge that connects a node to itself. Examples can be seen in Fig 2.1 (c)

Another way to categorize graphs is based on the orientation of their edges. Using this approach, graphs are generally classified into two types:

1. Undirected Graph  
   A graph whose edges have no directional orientation is referred to as an undirected graph. In an undirected graph, the order of the pairs of nodes connected by an edge is not considered. Thus, for an edge connecting two nodes *u* and *v* in a graph, (*u*, *v*) is equivalent to (*v*, *u*). Examples can be seen in Fig 2.1.
2. Directed Graph  
   A graph in which each edge is given a directional orientation is termed as a directed graph. In a directed graph, for an edge connecting two nodes *u* and *v*, (*u*, *v*) and (*v*, *u*) represent distinct pairs. In the case of the edge (*u*, *v*), node *u* is referred to as the origin node, and node *v* is termed the terminal node. Examples can be seen in Fig 2.2.



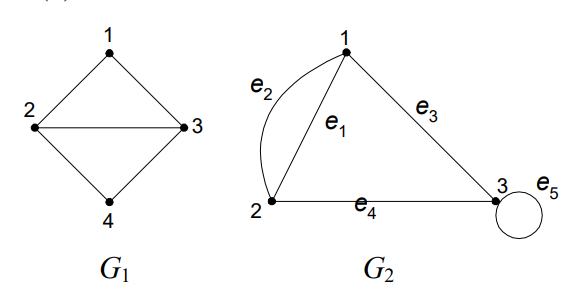
*Fig. 2.2 Examples of directed graph*

There are several terminologies related to graphs. Below are some terms that will be used in this paper.

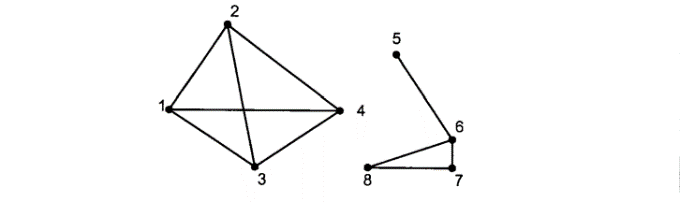
1. Path  
   A path of length n from the initial vertex *v0* to the destination vertex *vn* in the graph G is a sequence alternating between vertices and edges in the form *v0*, *e1*, *v1*, *e2*, …, *vn-1*, *en*, *vn*, such that *e1* = (*v1*, *v2*), …, *en* = (*vn-1*, *vn*) are the edges of the graph G.

If the graph under consideration is a simple graph, then we only need to express the path as a sequence of vertices. For example, refer to Fig. 2.3. We can express the path for graph G1 as 1, 2, 3, 4 instead of 1, 12, 2, 23, 3, 34.

1. Circuit  
   A path that starts and ends at the same vertex is called a circuit. Refer to Fig 2.3 for an example. Note that graph G1 has a path 1, 2, 4, 3, 1. Because the path starts and ends at the same vertex, graph G1 is considered has a circuit. Similarly in graph G2, we can see that G2 has a path 1, *e2*, 2, *e4*, 3, *e3*, 1 that starts and ends at the same vertex. It means that G2 has a circuit.



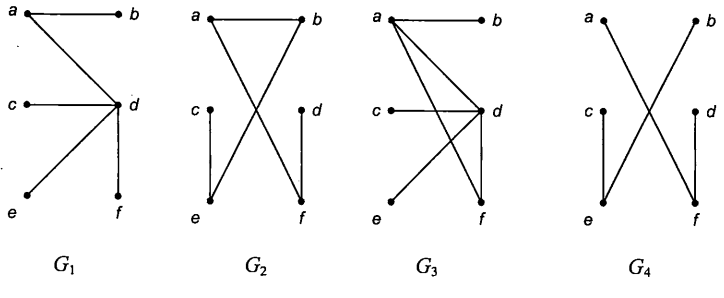
*Fig. 2.3 Two graphs G1 and G2*

1. Connected  
   An undirected graph G = (V, E) is called connected if, for every pair of vertices *u* and *v* within the set V, there exists a path from *u* to *v*. If not, G is referred to as a disconnected graph. For example, in Fig. 2.4 below, we can see that there is no path from vertex 1 to 5. It means that the graph below is a disconnected graph.  
     
   

*Fig. 2.4 Example of disconnected graph*

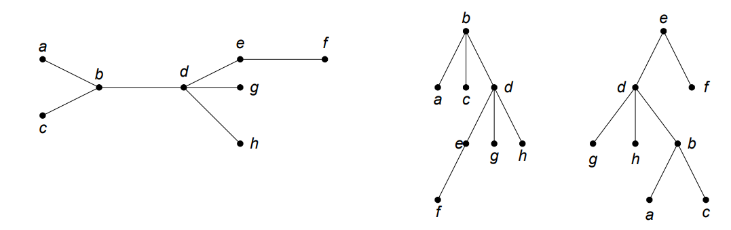
## C. Tree

A tree is defined as a connected undirected graph that does not contain any circuits. In this definition, we can observe two essential properties of a tree: connectedness and absence of circuits. Notice the examples in Fig. 2.5 below. We can see that both graphs G1 and G2 are trees, as they are both connected graphs and do not contain any circuit. While graph G3 is not a tree because it contains a circuit a, d, f, a, graph G4 also not a tree because it is not a connected graph



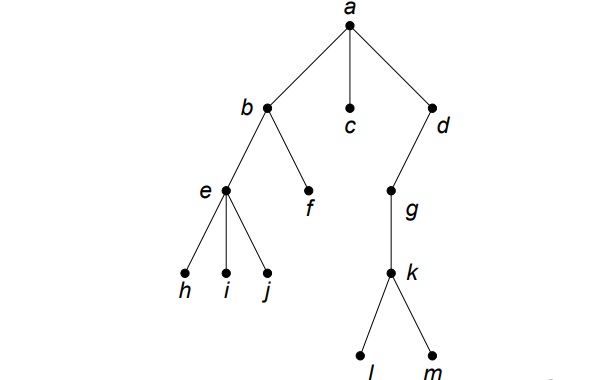
*Fig. 2.5 Examples of tree (G1 and G2) and not-tree (G3 and G4)*

A tree in which one of its vertices is treated as the root, and its edges are directed away from the root, is called a rooted tree. In many applications of trees, a specific vertex is designated as the root, forming a rooted tree. Any arbitrary unrooted tree can be transformed into a rooted tree by selecting a vertex as the root. Refer to the example in Fig. 2.6 for illustration. Here we can transform the tree into a rooted tree by selecting vertex b or vertex e as the root.



*Fig. 2.6 A tree and its rooted tree produced by choosing a vertex as the tree root*

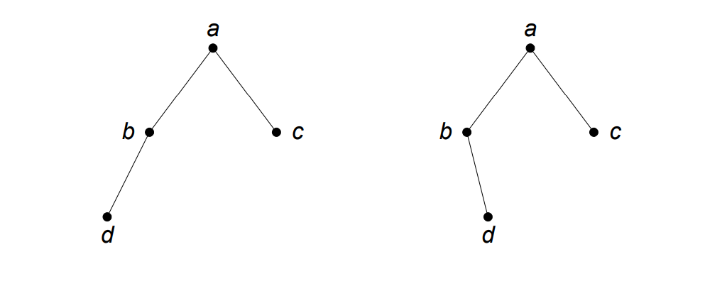
There are several terminologies related to rooted tree. We will discuss some terms that used in this paper.



*Fig. 2.7 A rooted tree*

1. Child and Parent  
   Let *x* be a node in a rooted tree. Node *y* is said to be the child of *x* if an edge from *x* to *y* exists. Therefore, *x* is parent of *y.* As an example, in Fig. 2.7, node d is a child of node a, and node a is the parent of node b, c, and d.
2. Path  
   Path from vertex *v*1 to vertex *v*k is a series of vertices *v*1, *v*2, …, *v*k such that *v*i is the parent of *v*i+1 for 1 <= *i* < *k*. From Fig. 2.7, the path from a to h is a, b, e, h.
3. Descendant and Ancestor  
   If there exists a path from vertex *x* to vertex *y*, then *x* is an ancestor of *y*, and *y* is a descendant of *x*. In Fig. 2.7, d is an ancestor of m and therefore m is a descendant of d.
4. Subtree  
   Let *x* be a node in a tree T. A subtree with x as the root is a subgraph *T’ = (V’, E’)* such as *V’* contains *x* and its descendant and *E’* contains the edges of every path that starts from *x*. As an example, in Fig. 2.7, we can take node b as the root of our subtree so that we have subtree *T' = (V’, E’)* with *V’* = {*b, e, f, h, i, j*} and E’ = {(*b, e*), (*b, f*), (*e, h*), (*e, i*), (*e, j*)}.
5. Degree  
   The degree of a node *x* in a rooted tree is the number of children of *x.* In Fig. 2.7, node d has a degree of 1, and node b has a degree of 3.
6. Leaf  
   A node that has a degree of zero, meaning it has no children, is called a leaf. In Fig. 2.7, node h, i, j, l, m are leaves.
7. Level and Height  
   The root of a tree is at level 0, while the level of any other node is defined as 1 plus the length of the path from the root to that node, while height of a tree is the maximum length of the path from the root to a leaf. In Fig. 2.7, the height of the tree is 4.

A rooted tree in which each node has at most n children is called an n-ary tree. A binary tree is an n-ary tree with n = 2. In a binary tree, the children of a node are referred to as the left child or the right child. A subtree that is the left child of a tree is called the left subtree, and a subtree that is the right child of a tree is called the right subtree.



*Fig. 2.8 A different binary tree*

## D. Minimax Algorithm in Game Theory

A strategic game is a model of interactive decision-making, where each participant selects their course of action definitively, and these decisions occur simultaneously. The model includes a finite set *N* of players, and for each player *i*, there exists a set *Ai* of actions along with a preference relation on the set of action profiles. An outcome is referred to as an action profile *a = (aj)j∈N*, and the set of outcoumes x*j∈NAj* is denoted as A.

A strategic game is strictly competitive if

# III. Helpful Hints

## A. Figures and Tables

Large figures and tables may span both columns. Place figure captions below the figures; place table titles above the tables. If your figure has two parts, include the labels “(a)” and “(b)” as part of the artwork. Please verify that the figures and tables you mention in the text actually exist. Please do not include captions as part of the figures. Do not put captions in “text boxes” linked to the figures. Do not put borders around the outside of your figures. Use the abbreviation “Fig.” even at the beginning of a sentence. Do not abbreviate “Table.” Tables are numbered with Roman numerals.

Figure axis labels are often a source of confusion. Use words rather than symbols. As an example, write the quantity “Magnetization,” or “Magnetization *M*,” not just “*M*.” Put units in parentheses. Do not label axes only with units. As in Fig. 1, for example, write “Magnetization (A/m)” or “Magnetization (Am1),” not just “A/m.” Do not label axes with a ratio of quantities and units. For example, write “Temperature (K),” not “Temperature/K.”

Multipliers can be especially confusing. Write “Magnetization (kA/m)” or “Magnetization (103 A/m).” Do not write “Magnetization (A/m)  1000” because the reader would not know whether the top axis label in Fig. 1 meant 16000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type.

## B. References

Number citations consecutively in square brackets [1]. The sentence punctuation follows the brackets [2]. Multiple references [2], [3] are each numbered with separate brackets [1]–[3]. When citing a section in a book, please give the relevant page numbers [2]. In sentences, refer simply to the reference number, as in [3]. Do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] shows ... .”

Number footnotes separately in superscripts (Insert | Footnote)[[1]](#footnote-1). Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes.

Please note that the references at the end of this document are in the preferred referencing style. Give all authors’ names; do not use “*et al*.” unless there are six authors or more. Use a space after authors' initials. Papers that have not been published should be cited as “unpublished” [4]. Papers that have been submitted for publication should be cited as “submitted for publication” [5]. Papers that have been accepted for publication, but not yet specified for an issue should be cited as “to be published” [6]. Please give affiliations and addresses for private communications [7].

## C. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have already been defined in the abstract. Abbreviations that incorporate periods should not have spaces: write “C.N.R.S.,” not “C. N. R. S.” Do not use abbreviations in the title unless they are unavoidable.

## D. Equations

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). First use the equation editor to create the equation. Then select the “Equation” markup style. Press the tab key and write the equation number in parentheses. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence, as in

 (1)

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (*T* might refer to temperature, but T is the unit tesla). Refer to “(1),” not “Eq. (1)” or “equation (1),” except at the beginning of a sentence: “Equation (1) is ... .”

## E. Other Recommendations

Use one space after periods and colons. Hyphenate complex modifiers: “zero-field-cooled magnetization.” Avoid dangling participles, such as, “Using (1), the potential was calculated.” [It is not clear who or what used (1).] Write instead, “The potential was calculated by using (1),” or “Using (1), we calculated the potential.”

Use a zero before decimal points: “0.25,” not “.25.” Use “cm3,” not “cc.” Indicate sample dimensions as “0.1 cm  0.2 cm,” not “0.1  0.2 cm2.” The abbreviation for “seconds” is “s,” not “sec.” Do not mix complete spellings and abbreviations of units: use “Wb/m2” or “webers per square meter,” not “webers/m2.” When expressing a range of values, write “7 to 9” or “7-9,” not “7~9.”

A parenthetical statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.) In American English, periods and commas are within quotation marks, like “this period.” Other punctuation is “outside”! Avoid contractions; for example, write “do not” instead of “don’t.” The serial comma is preferred: “A, B, and C” instead of “A, B and C.”

If you wish, you may write in the first person singular or plural and use the active voice (“I observed that ...” or “We observed that ...” instead of “It was observed that ...”). Remember to check spelling. If your native language is not English, please get a native English-speaking colleague to proofread your paper.

# IV. Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

# V. Appendix

Appendixes, if needed, appear before the acknowledgment.

# VI. Acknowledgment

All Praise and gratitude to Allah Subhanahu wa Ta’ala, for by His mercy, the paper titled “A Decision Tree-Based Approach to Optimal Move Selection in Chess Middlegame Using Minimax Algorithm with Stockfish Engine” has been successfully completed. Also, gratitude is extended to the lecturer for IF2120 Discrete Mathematics course, Dr. Nur Ulfa Maulidevi, S.T, M.Sc., for the guidance and motivation provided throughout her tenure in teaching the students

# References

1. handbook.fide.com. 2023. FIDE Laws of Chess. [online] Available at: <<https://handbook.fide.com/chapter/E012023>> [Accessed 7 December 2023 21:08]
2. Munir, Rinaldi. 2010. *Matematika Diskrit Edisi 3.* Bandung: Penerbit INFORMATIKA Bandung.

# Statement of Originality

I hereby declare that this paper is my own writing, not an adaptation, or translation of someone else's paper, and not plagiarized.

Bandung, 9 Desember 2023

Ibrahim Ihsan Rasyid - 13522018

1. It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date on the first page). Instead, try to integrate the footnote information into the text. [↑](#footnote-ref-1)