

Solution Search Simulation The Shortest Step On Chess Horse Using Algorithm

Ade Bastian^{1,*}, Rezha Nugraha²

Informatic Engineering Program, Technic Faculty, Majalengka University, Majalengka West Java, Indonesia

¹ *adb@ft.unma.ac.id**; ² *rezhanoe@yahoo.com*

** corresponding author*

ARTICLE INFO

Article history:
Received: 2018-02-25
Revised: 2018-05-05
Accepted: 2018-06-06

Keywords:
A horse beans
Chess board
Simulation programming
Breadth-first search

ABSTRACT

Horse seed in the chess board movement resembles the letter L. The chess pieces are one of very hard-driven beans, and grains are often also the most dangerous if not carefully considered every change. Simulation of this problem provides a chess board size $n \times n$. Target (goal) of this problem is to move horse beans of a particular position on a chess board position to the desired destination with the shortest movement simulates all possible solutions to get to the goal position. This problem is also one of the classic problems in artificial intelligence (AI). Settlement of this problem can use the help system and tree production tracking. Therefore, designed simulation applications by utilizing several techniques of simulation programming and Breadth-First Search method. With this method, all nodes will be traced, and the nodes at level n will be visited first before visiting the nodes at level $n + 1$. The purpose of this study is to design a software that can find all the solutions for the shortest movement toward the goal position by using the system of production and tracking tree. Results from this paper are that the software can find all solutions shortest movement a horse beans from the initial position to the goal position and displays the simulation of the evolution of the horse in the chess board.

Copyright © 2017 International Journal of Artificial Intelligence Research.
All rights reserved.

I. Introduction

Artificial intelligence or AI includes a relatively young field of science. In the 1950s scientists and researchers began to think of how the machine could do its work as human beings could [1]. Alan Turing, a British mathematician, first proposed a test to see whether or not a computer is said to be intelligent[2]. The test results are then known as Turing Test, where the machine is disguised as someone in a game that can respond to a series of questions asked. Turing assumes that, if the network can make a person believe that he or she is capable of communicating with others, then it can be said that the engine is intelligent, just like a human being [3].

To solve a problem in AI, adequate knowledge is required. Not only that, but the system must also have the ability to reason. The knowledge base and the ability to motivate can be the essential part of a system that uses AI [4]. Although a system has a lot of knowledge, if it cannot suppose, it would be useless. Vice versa, if a system has a powerful ability to motivate, but its knowledge base is not enough, then the solution obtained becomes not maximal[5].

The most important thing in determining the success of AI-based systems is a success in search and matching. There are two search and tracking techniques used, namely Blind Search and Heuristic Search [6]. In the Breadth-First Search method, all nodes at level n will be visited first before visiting nodes at level $n + 1$. The search starts from the root node and continues to the 1st level from left to right, then moves to the next level. And so on until the discovery of a solution [7].

Chess is a two player game played on a square board. Generally chess is performed on board with eight rows and eight columns; however, we will deal with the more general case where the

board contains m rows and m columns. For this paper, we will assume the orientation of a board is fixed so that we can label all the squares [8]. Let column 1 denote the leftmost column and row 1 indicate the top row. The game of chess is a game that requires intelligence because it takes a strategy to regulate the number of possible movements that can be done. In computer science, to be able to create a computer system that has the same ability as the world chess players is a challenge that many in-demand [9]. Problems that exist, among others, the system created to develop the knowledge it has through the continuous learning process. The learning process in this chess game system requires excellent resources and intricate complexity[10].

Horse seeds on a chessboard have movements resembling the letter L. This chess is one of the seeds is very difficult to move and is often also the most dangerous seed if not carefully observed every movement [11]. The simulation of this problem provides an $n \times n$ chessboard. The goal of this problem is to move a horse seed from a particular position on the chessboard to the desired destination position by simulating all possible shortest possible movement solutions to get to that goal position. This problem is also one of the classic problems in artificial intelligence (AI). Solving this problem can use the help of a search algorithm and tracking tree [12].

II. Method

The problem of horse step movement on a chessboard is one of the classic issues in AI. In the game of chess, horse seeds have changes resembling the letter L. This chess seed is one of the most difficult seeds to move and is often also the most dangerous seed if not carefully considered every movement [13]. This problem provides an $n \times n$ chessboard. The goal of this problem is to move a horse seed from a particular position on the chessboard to the desired destination position by simulating all possible moves to get to the goal position without violating the prescribed rules [14].

The simplest way to represent a board is to create an 8×8 two-dimensional array. There are 13 different entities on the chessboard: 6 different pieces for white and black and an empty square. This implies that a byte (short integer) would be enough for one element representation. Each array element identifies which entity is occupying the square on the chessboard. The next problem is the encoding of these entities [15].

The container where the horse movement is a chessboard with the size of $m \times n$. This is so that the user can restrict the movement of the horse, for example only in the space of motion: 4×4 . Chessboard 8×8 as shown in Figure 1 below :

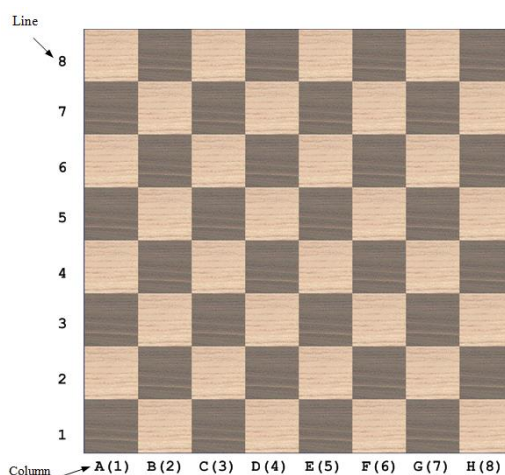


Fig. 1. Chess Board 8×8

The problem faced is that the horse must be moved from its position to the king's position in a chessboard $m \times n$. The possible rules of horse seed movement are :

Horses can be moved to the left side. Operation: Row = row + 2, Column = column - 1. Rule: The column variable value after operation must be > 0 , the row variable value $\leq m$, and there is no pawn seed in that position.

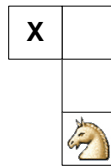


Fig. 2. Rule of Movement-1

Horses can be shifted to the right. Operation : $Row = row + 2, Column = column + 1$.
 Rule: The column variable value after operation should be $\leq n$ and the line after surgery should be $\leq m$, and there is no pawn seed in that position.



Fig. 3. Rule of Movement-2

Horses can be shifted to the upper left. Operation : $Line = row + 1, Column = column - 2$.
 Rule: The column variable value after operation must be > 0 , the row variable value $\leq m$, and there is no pawn seed in that position.

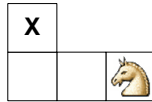


Fig. 4. Rule of Movement-3

Horses can be moved to the left side down. Operation : $Line = row - 1, Column = column - 2$.
 Rule: The value of column variable and row after operation must be > 0 , and there is no pawn seed in that position.



Fig. 5. Rule of Movement-4

Horses can be shifted to the upper right. Operation : $Line = row + 1, Column = column + 2$.
 Rule: The column variable value after operation should be $\leq m$, and the line after operation should be $\leq n$, and there is no pawn seed in that position.



Fig. 6. Rule of Movement-5

Horses can be moved to the right side down. Operation : $Line = row - 1, Column = column + 2$.
 Rule: The column variable value after operation should be $\leq n$, the value of the row variable must be > 0 , and there is no pawn seed in that position.



Fig. 7. Rule of Movement-6

Horses can be moved down the left side Operation : $Line = row - 2, Column = column - 1$.
 Rule: The value of column variable and row after operation must be > 0 , and there is no pawn seed in that position.

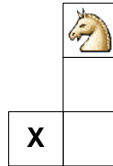


Fig. 8. Rule of Movement-7

Horses can be shifted down the right side. Operation : Line = row - 2, Column = column + 1.
 Rule: The column variable value after operation must be $\leq n$, the value of the row variable must be > 0 , and no other chess seeds in that position.

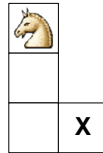


Fig. 9. Rule of Movement-8

The solution of the horse problem is by using the tracking tree and the Breadth-First Search (BFS) method. BFS method is used because the solution of the problem can be more than one and will get the solution with the shortest path (shortest path).

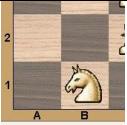
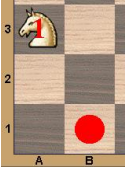
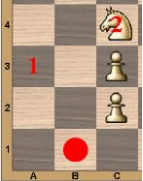

The steps to find solutions to the problem are as follows: Set the initial position as the root node (node 1) of the tree. White Horse (B1) = Initial Position. White Pawns (C2, C3, C5, E4, F3) = Obstacles. Black King (E5) = Purpose.



Fig. 10. Initial Positioning

Develop all possible child nodes (level 2 nodes) from the root node by making all possible moves and obeying the movement rules. The rules of movement in question are the 8 points of movement rules described above plus the rule that the horse shall no longer occupy previously occupied positions in the same path. Each searches 8 points; each vacant position will be stored in an array of horses. When developing a node and found that the node being examined and to be developed is currently the destination position, then save the node level to shortest and store it in the shortest solution array.

Table 1. Finding the Shortest Solution

No.	Step	Figure
1	Root Node (Initial) Position (B1)	
2	Node Level 2 Position (A3)	
3	Node Level 3 Position (C4)	
4	Node Level 4 Position (E5) Save As (nShortest = 3 Steps) (Temporary solution) Position (E5)	

If the node being examined and developed now has a level one level more excellent than the shortest variable value then the Check = True set for all nodes (meaning there will be no child nodes to be checked and developed.) The search ends. Repeat step b and step c for all nodes to generate level nodes 3, 4 and so on. Child node development stops when no nodes can be expanded again.

The result of the solution of horse problem above by using the Breadth-First Search method is as follows :

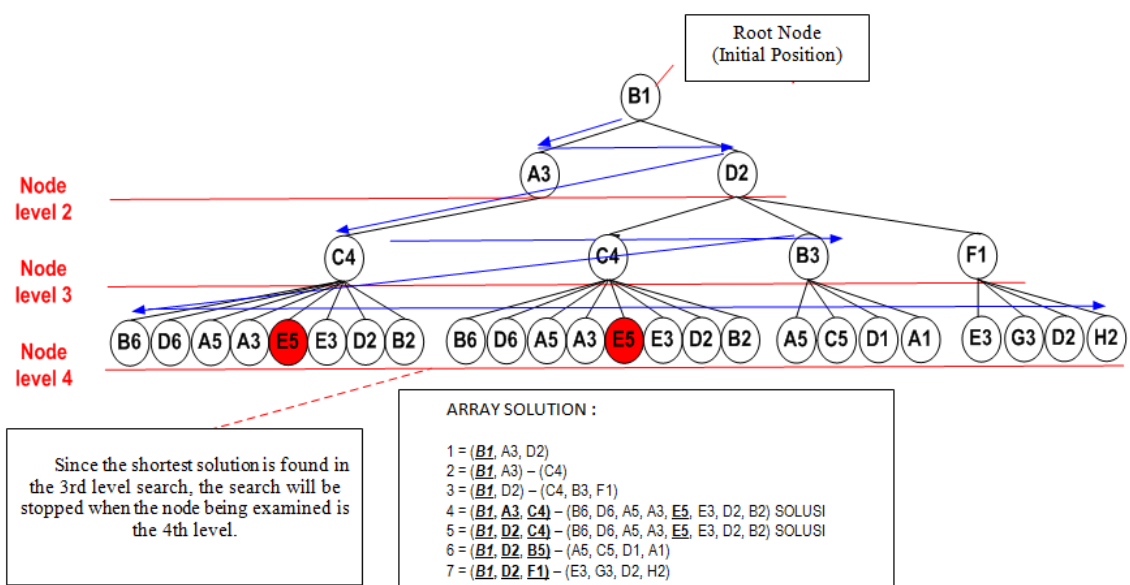


Fig. 11. BFS search on horse problem

III. Result

The design algorithm of horse step simulation software in the chessboard can be divided into 2 (two) parts, namely solution search algorithm and supporting function Algorithm. For example, the input data is as follows :

- a) Chess board size = 8×8 .
- b) The position of horse seed = C2.
- c) The position of seed king = G7.
- d) Pawn seed position = B5, D3, D4, D5, F3, F5, H3, and H5.

There are eight operations to develop each node in the tracking tree :

- e) Slide the seed horse to the left. (Row = row + 2, Column = column - 1).
- f) Slide the seed horse to the right. (Line = row + 2, Column = column + 1).
- g) Slide the horse seeds to the top left. (Row = row + 1, Column = column - 2).
- h) Slide the seeds of the horse to the lower left. (Line = row - 1, Column = column - 2).
- i) Move the horse to the right side up. (Line = row + 1, Column = column + 2).
- j) Move the horse to the right side down. (Line = row - 1, Column = column + 2).
- k) Slide the horse down on the left. (Row = row - 2, Column = column - 1).
- l) Slide the horse down to the right. (Line = row - 2, Column = column + 1).

This operation must comply with the following rules :

- m) Position after shift should not be outside the chessboard.
- n) Position after shift should not be equal to the area of pawn seed (hindrance).

Position after the shift has never been traversed before (on the same track in the tracking tree). This ensures that horse seeds will not occupy previously occupied positions. Display the input form for the case example as shown in Figure 12 below :

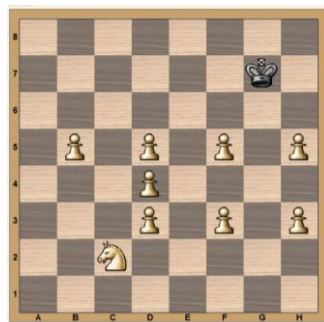


Fig. 12. Solution Form (example case 1)

The simulation of the resulting solution is as shown in Figure 13 below :

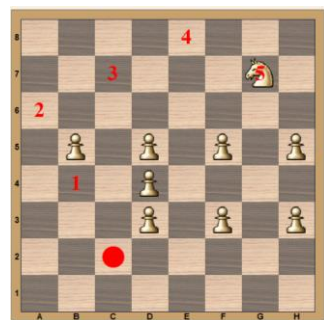


Fig. 13. Simulation from Solution-1

IV. Conclusion

After completing the simulation software of the legal movement of the horse in the chess board, the authors draw the following conclusions Breadth First Search (BFS) search method can be used to search shortest step solution on horse chess. The software can implement the concept of a tracking tree based on the idea of Artificial Intelligence (AI) in solving a problem using the Breadth First Search (BFS) search method. The software can simulate all the steps of any shortest solution found.

References

- [1] R. Zhou and E. A. Hansen, "Breadth-first heuristic search," *Artif. Intell.*, 2006.
- [2] F. A. Csaszar and N. Siggelkow, "How Much to Copy? Determinants of Effective Imitation Breadth," *Organ. Sci.*, 2010.
- [3] S. Beamer, K. Asanović, and D. Patterson, "Direction-optimizing breadth-first search," in *Scientific Programming*, 2013.
- [4] Z. Fu, H. K. Dasari, B. Bebee, M. Berzins, and B. Thompson, "Parallel breadth first search on GPU clusters," in *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, 2014.
- [5] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, 2017.
- [6] E. Aprile *et al.*, "First Dark Matter Search Results from the XENON1T Experiment," *Phys. Rev. Lett.*, 2017.
- [7] J. L. Ferreras-Méndez, S. Newell, A. Fernández-Mesa, and J. Alegre, "Depth and breadth of external knowledge search and performance: The mediating role of absorptive capacity," *Ind. Mark. Manag.*, 2015.
- [8] A. K. Zobel, B. Lokshin, and J. Hagedoorn, "Formal and informal appropriation mechanisms: The role of openness and innovativeness," *Technovation*, 2017.
- [9] B. P. Abbott *et al.*, "First targeted search for gravitational-wave bursts from core-collapse supernovae in data of first-generation laser interferometer detectors," *Phys. Rev. D*, 2016.
- [10] S. Terjesen and P. C. Patel, "In Search of Process Innovations: The Role of Search Depth, Search Breadth, and the Industry Environment," *J. Manage.*, 2017.
- [11] R. Pearce, M. Gokhale, and N. M. Amato, "Multithreaded asynchronous graph traversal for In-Memory and Semi-External Memory," in *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2010*, 2010.
- [12] I. Im, J. Jun, W. Oh, and S.-O. Jeong, "Deal-Seeking Versus Brand-Seeking: Search Behaviors and Purchase Propensities in Sponsored Search Platforms.," *MIS Q.*, 2016.
- [13] S. Paruchuri and S. Awate, "Organizational knowledge networks and local search: The role of intra-organizational inventor networks," *Strateg. Manag. J.*, 2017.
- [14] S. Sallinen, A. Gharaibeh, and M. Ripeanu, "Accelerating direction-optimized breadth first search on hybrid architectures," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015.
- [15] K. Ueno, T. Suzumura, N. Maruyama, K. Fujisawa, and S. Matsuoka, "Extreme scale breadth-first search on supercomputers," in *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, 2016.