

Analysis Of Shortest Path Determination By Utilizing Breadth First Search Algorithm

Jonhariono Sihotang, Siska Simamora

¹Magister Teknologi Informasi, STMIK Mikroskil, Medan Sumatera Utara

²Lembaga Publikasi Bersinar

Email: jonharionsihotang45@gmail.com

Received : 21 Juli 2020

Revised : 18 Agustus 2020

Accepted :20 September 2020

Keywords :

BFS

Shortest Path

Search

Abstract. The rapid development of science requires the public to keep up with the development of such technology. The use of computers as one of the tools used to facilitate work and improve the efficiency and effectiveness of work is also high, this we can see from the development of such technology. Artificial Intelligence (AI) is one part of computer science that learns about how to make computers can do the job as humans do. At the beginning of its creation, the computer was only functioned as a counting tool. But along with the development of the times, the role of computers increasingly dominates the life of mankind. Computers are no longer only used as a calculation tool, more than that, computers are expected to be empowered to do everything that can be done by humans. People can be good at solving all problems in this world because people have knowledge and experience. Knowledge is gained from learning. The more knowledge possessed by a person is certainly expected to be more able to solve problems. But the provision of knowledge alone is not enough, people are also given the sense to reason, draw conclusions based on their knowledge and experience.

1. INTRODUCTION

Artificial Intelligence (AI) is a branch of computer science that deals with the automating of intelligent behavior. This statement can also be used as a definition of AI [1]. This definition indicates that AI is part of a computer so it should be based on the application principles of its field. These principles include the data structures used in the representation of knowledge, the algorithms needed to apply that knowledge, as well as the languages and programming techniques used in implementing it [2].

Tracking tree is a tree that can be applied to solve problems in the field of *Artificial Intelligence* (AI), where the root of the tree in the form of the initial state of the problem and branches (branches) in the form of circumstances that may occur from the previous state and leaves are the final state of the problem. These final circumstances can be a solution to the problem or perhaps nothing can be used as a solution to the problem [3].

Track search in the cartesian field, is a problem that applies the tracking tree method to find solutions by expanding the root *nodes* to the next levels, all possible movements, not breaking the conditions and resulting in new conditions being developed to the maximum extent possible. The search ends when there are no more *nodes* or new conditions that can be developed. All *nodes* that are the destination position are solutions[4].

The problem of AI that requires the application of tracking trees is in finding a trajectory that can be passed by an ant in the Cartesian field. This problem can be described as follows, it is known that an ant will move from the center point (0,0) to point A(m,n). The ant should only deflect at *grid* points and always step parallel to the x-axis or y-axis. The ant must not cross the path it has passed and must not cross the point it has been through. After that, a series of provisions are provided that limit the movement of the ants. The question is how the ants can form the trajectories using the conditions set out above[5].

Breadth First Search algorithm is one of the solution node search algorithms in a graph or tree. The hallmark of this algorithm is that the search starts from the root and then continues with a gradual search level by level, checking all nodes at a certain depth before going to a deeper level until 13 goals are found [6]–[8].

2. METHODS

2.1 Research Work Steps

The basis of research used as material for the completeness of data and information and the steps of making this software include:

1. Study Library
Collect data by reading and studying books related to *Artificial Intelligence* (AI).
2. Report Studies
This is done by collecting data directly on the application of tracking trees to solve the problem of finding a trajectory that can be passed by an ant in the cartesian field.
3. Algorithm Implementation
Stages of Implementation of Algorithms Is implementation by utilizing case study examples.
4. Exposure Results
An important part of this activity is to display the shortest path search results according to the issues outlined.

2.2 Search Method Breadth-First Search (BFS)

The most important thing in determining the success of ai-based systems is success in search and matching. The search used is *Breadth-First Search*, with the working concept In this method, all *nodes* at level n will be visited first before visiting the *nodes* at level $n + 1$ (next level). The search starts from the root *node* to the 1st level from left to right, then moves to level-2 and so on. Trees were developed until the discovery of solutions [9], [10].

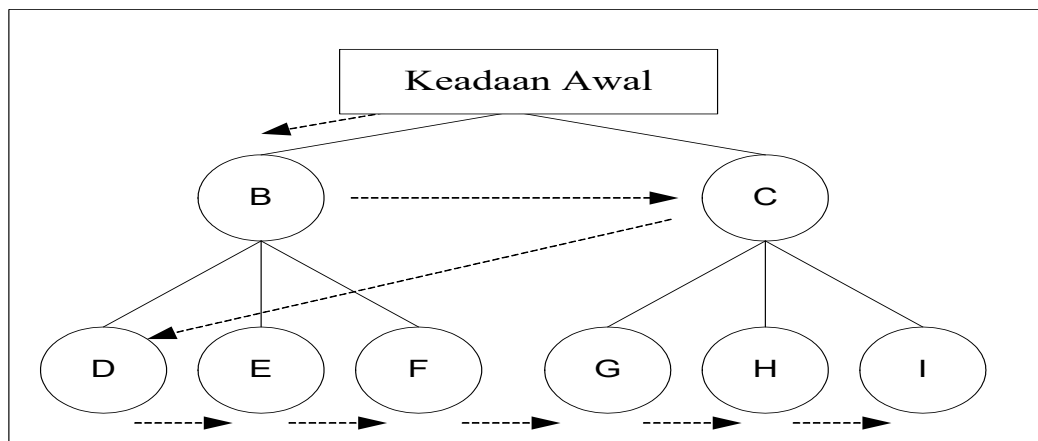


Figure 1. First Widening Search Procedure (BFS)

The advantage of this method is that it will not end in stalemate. If there is more than one solution, then *breadth-first search* will find it. And if there is more than one solution, then a minimum solution will be found. While the drawback of this method is that it requires a considerable amount of memory, because it stores all *nodes* in one tree and takes a long time, because it will test all nodes at the n -level to get a solution at the 1st level ($n + 1$).

The advantage of this method is that it will not end in stalemate. If there is more than one solution, then *breadth-first search* will find it. And if there is more than one solution, then a minimum solution

will be found. While the drawback of this method is that it requires a considerable amount of memory, because it stores all *nodes* in one tree and takes a long time, because it will test all nodes at the n -level to get a solution at the 1st level ($n + 1$).

3. RESULTS AND DISCUSSION

The solution to this problem can be more than one and the software is required to be able to display all the paths that the Object can pass from the starting position to the destination position by complying with the applicable provisions. Because the solution obtained can be more than one, then the search method used to find the solution to this problem is the first widened search method *breadth-first search* (BFS).

Starting from the initial position as the root *node*, the *breadth-first search* (BFS) method seeks a solution by expanding the root *node* to the next levels, all possible movements, not breaking the conditions and resulting in new conditions being developed to the fullest extent possible. The search ends when there are no more *nodes* or new conditions that can be developed. All *nodes* that are the destination position are solutions. The allowed movement of objects is:

1. Move to the right ($x + 1$).
2. Move left ($x - 1$).
3. Move up ($y + 1$).
4. Move down ($y - 1$).
5. All movement of the above Objects is legal if it complies with the following provisions:
 1. Do not cross the position of the obstacle point.
 2. Not crossing a point that's been passed before.
 3. Does not exceed the maximum movement limit of objects(x or y).
 4. Do not cross quadrants that are not allowed to cross. The quadrants in question are quadrant I (where x is positive and y is positive), quadrant II (where x is negative and y is positive), quadrant III (where x is negative and y is negative) and quadrant IV (where x is positive and y is positive).

An example of a tracking tree structure developed based on *the breadth-first search* (BFS) search method can be seen in figure 2

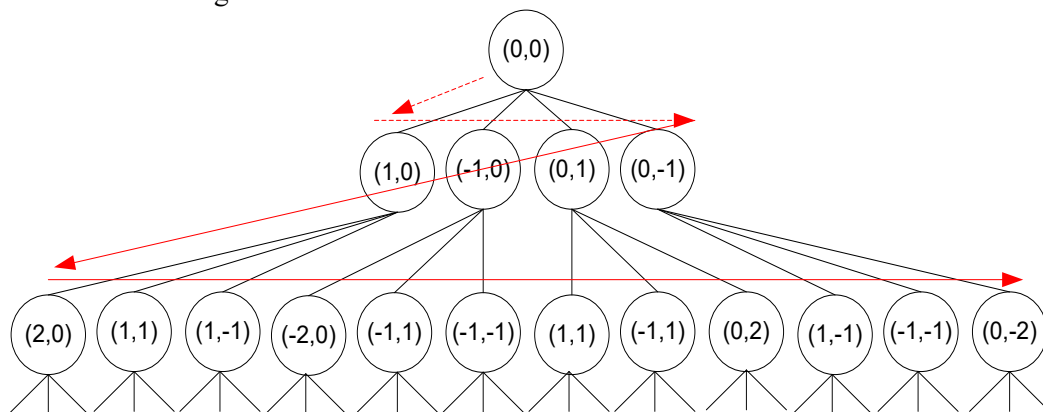


Figure 2. Example of tree structure tracking with *breadth* method -*firstsearch* (BFS)

Of the solutions found, there are optimum solutions. In its implementation, the optimum solution is the *shortest path* found using the tracking tree. In the case of Object movement, the optimum solution (the shortest movement without hindrance) can be determined by criteria. See the following example, if the initial position of the Object is (x_1, y_1) and the object's destination position is (x_2, y_2) then the most optimum solution (the shortest step) to the destination position can be calculated by the formula: $(abs(x_2 - x_1), abs(y_2 - y_1))$. This means that the movement along the x-axis (horizontal) is as big as

$abs(x_2 - x_1)$ step and the movement along the y axis (vertical) is as big as $abs(y_2 - y_1)$. Suppose the initial position of the Object is (-1, -2) and the object's destination position is (4, 5), then the most optimum solution is a solution that has the following criteria:

1. Movement along the x -axis (horizontal) is $abs(4 - (-1)) = 5$ steps.
2. Movement along the y -axis (vertical) is $abs(5 - (-2)) = 7$ steps.

However, if there are obstacles to the movement of objects (which causes all the most optimum solutions to be impassable, note figure 3.3), then the criteria for the optimum solution do not apply. Therefore, the optimum solution is the shortest solution found by the tracking tree and no criteria can formulate that a solution is the most optimum solution. In the software, to find or obtain the most optimum solution, the maximum movement of objects is limited by the above formula and there are no obstacles that limit all movement of existing Objects.

For more details, consider the example of the following problem.

1. Object starting position : (2,1).
2. Object destination position : (5,5).
3. Maximum Movement of Objects , $x = 6$ and $y = 6$.
4. Obstacle positions, namely:
 - 1.Hurdle-1: (2,3).
 - 2.Hurdle-2: (4,3).
 - 3.Hurdle-3: (3,5).

All quadrants can be traversed by Objects. The most optimum solution is a solution that has the following criteria:

- 1.Movement along the x axis = $abs(5 - 2) = 3$.
- 2.Movement along the y axis = $abs(5 - 1) = 4$.

So in accordance with the search method used and the applicable conditions, there are several solutions. Examples of the two most optimum solutions can be seen in figure 2

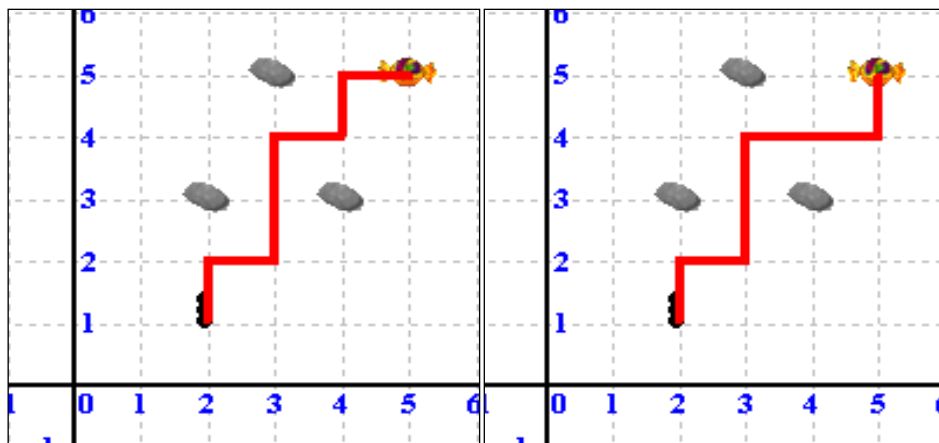


Figure 3 The most optimum solution-1

The solution in figure 2 is the most optimum solution because it is the solution with the *shortest path* and meets the criteria of the most optimum solution formula, which is to have the movement of the x -axis as much as 3 steps and *the* y -axis as much as 4 steps. Examples of two solutions not optimum can be seen in figure 3

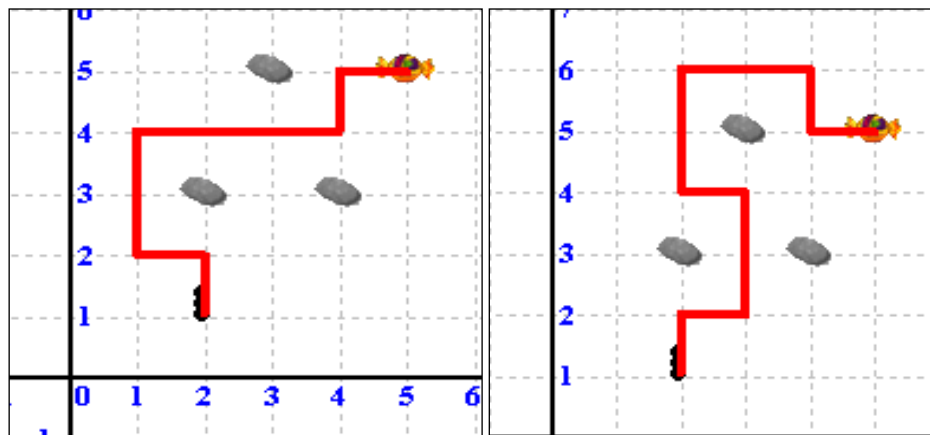


Figure 4 Solution is not optimal-1

The solution in figure 4 is not an optimum solution because it is not the *shortest path*.

4. CONCLUSION

After completing the application of the tracking tree in finding the path that can be passed by an object in the cartesian field with *breadth first search* method (BFS) stated that the most optimum solution because it is the solution with the shortest path (*shortest path*) and meets the criteria of the most optimum solution formula, which has the movement of the x-axis as much as 3 steps and the y-axis as much as 4 steps.

REFERENCE

- [1] Nasri, "Kecerdasan buatan (Artificial Intelligence)," *Artif. Intell.*, vol. 1, no. 2, 2014.
- [2] A. N. Khomarudin and L. Efriyanti, "PENGEMBANGAN MEDIA PEMBELAJARAN MOBILE LEARNING BERBASIS ANDROID PADA MATA KULIAH KECERDASAN BUATAN," *J. Educ. J. Educ. Stud.*, vol. 3, no. 1, 2018, doi: 10.30983/educative.v3i1.543.
- [3] H. A. Ramadhan and D. A. Putri, "Big Data, Kecerdasan Buatan, Blockchain, dan Teknologi Finansial di Indonesia," *Direktorat Jenderal Apl. Inform. Kementeri. Komun. dan Inform.*, 2018.
- [4] I. Aprilia, "Analisis dan Penyelesaian Permainan River Crossing Ultimate Menggunakan Algoritma BFS dan DFS," *Edisi*, vol. 6, no. 2, 2016.
- [5] E. G. Masala, I. P. Saputro, and R. T. B. Turang, "PERBANDINGAN ALGORITMA BREADTH FIRST SEARCH DAN DEPTH FIRST SEARCH PADA GAME MUMMY MAZE DELUXE," *J. Ilm. Realt.*, vol. 14, no. 2, 2018, doi: 10.52159/realtech.v14i2.46.
- [6] R. Priyana and E. T. E. Handayani, "Perancangan Game 'Heroes Surabaya' Sebagai Edukasi Pengetahuan Sejarah Menggunakan Algoritma BFS Berbasis Android," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 4, no. 1, 2019, doi: 10.37438/jimp.v4i1.192.
- [7] Santi, "Penerapan Algoritma Best First Search (BFS) dalam Pencarian Lokasi Apotek K-24 Berbasis Android di Kota Makassar," *SNIF*, 2015.
- [8] M. A. Adiputra, "Penerapan Algoritma BFS dan DFS untuk Penjadwalan Rencana Studi," *Makal. IF2211 Strateg. Algoritm. Sem. II Tahun 2017/2018*, 2018.
- [9] H. A. Humansyah, I. Fitri, and A. Rubhasy, "Aplikasi Mobile Presensi Siswa Terintegrasi SMS Gateway Menggunakan Algoritma Breadth-First Search (BFS)," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 5, no. 2, 2021, doi: 10.35870/jtik.v5i2.203.
- [10] Z. Fachrina, "Pencarian Barang Dalam Basis Data Online Shop Menggunakan Algoritma Bfs ," *Strateg. Algoritm.*, 2015.