Breadth First Search Approach for Shortest Path Solution in Cartesian Area

Robbi Rahim¹*, Dahlan Abdullah², Saiful Nurarif³, Mukhlis Ramadhan³, Badru Anwar³, Muhammad Dahria⁴, Surya Darma Nasution⁵, Tengku Mohd Diansyah⁶ and Mufida Khairani⁶

¹School of Computer and Communication Engineering, Universiti Malaysia Perlis, Malaysia
²Department of Informatics, Universitas Malikussaleh, Jl. Medan-Banda Aceh, Aceh Utara, Indonesia
³Department of Computer System, STMIK Triguna Dharma, Jl. Jenderal Abdul Haris Nasution No.73, Medan 20219, Indonesia
⁴Department of Information System, STMIK Triguna Dharma, Jl. Jenderal Abdul Haris Nasution No.73, Medan 20219, Indonesia
⁵Department of Informatics, STMIK Budidarma, Medan, Indonesia
⁶Department of Informatics, Universitas Harapan Medan, Medan, Indonesia

*usurobbi85@zoho.com

Abstract. Determining the shortest path is one problem that is much discussed using some algorithm like Dijkstra, Floyd Warshall and in this research an algorithm Breadth First Search are used. Breadth First Search algorithms in this study is used to determine the shortest route and optimal from a Cartesian field, the best and optimal route search experiment of cartesian areas using Breadth First Search algorithm can be perform very well and gets some route options from the best to the longest route.

1. Introduction

The development of computer-based technology is currently experiencing rapid development, and the development has also covered various aspects of the field of life with the aim to facilitate the completion of tasks charged, Artificial intelligence is the most developed field today.

Shortest Path is a pretty much solved problem with algorithms like Dijkstra, Depth First Search and Breadth First, Yan ZHOU[1] conducted a shortest path study using the Breadth First Search algorithm by approaching the search from the starting point to the destination point by examining the layer by layer against all the possibilities that can be passed and remove all the paths that are not possible to pass. Similar to Yan ZHOU this experiment try to finding the shortest path that can pass through the Cartesian field. This issue could be explained as follows; it is identified that an object will move from the center point (0,0) to point A (m, n). The object should only bend at the grid points and always step in parallel with the x-axis or y-axis. The object must not cross the path that has been passed and should not cross the point that has been crossed. After that provided a set of provisions that limit the movement of the object. The question is how the trajectories that the object can pass through using the terms defined above.
Breadth First Search is one solution that can be used to solve the above problems. Breadth-first search is an algorithm that performs a wider search by visiting the node preorder i.e. visiting a node and then visiting all the nodes adjacent to the node first[2][3]. Next, unvisited nodes and neighbors of the previously visited nodes, so it is doubtful that the visited node will be revisited [5][6].

This study attempts to show some of the path solutions from the initial location to the destination location, using the Breadth First Search algorithm. The given solution can consist of many solutions with the path meet the criteria of the problem already mentioned, Research conducted by previous research Yan ZHOU, this research not only provides the shortest path solution but also provides another form of optimal search if there are obstacles that block the process of the path being traversed.

2. Theory

A. Breadth first search

In breadth first search method, all nodes at level n will be visited first before visiting nodes at level n+1 (next level). The search starts from the root node and continues to the 1st level from left to right, then moves to level-2 and so on[2][3][7]. As an illustration of the first search breadth process could be seen in the following Tree process until the discovery of the solution [8].

```
For N = 0 To UBound(Solution(Row).Step)
    Step.Matrix(N + 1, 0) = N
    Step.Matrix(N + 1, 1) = "(" & _
    Solution(Row).Step(N).X & "," & _
    Solution(Row).Step(N).Y & ")"
    'Ant Movement
    If N > 0 Then
        'Initial
        LineS(N).X1 = ConvX(Solution(Row).Step(N - 1).X)
        LineS(N).Y1 = ConvY(Solution(Row).Step(N - 1).Y)
        'Goal
        LineS(N).X2 = ConvX(Solution(Row).Step(N).X) + 15
```
LineS(N).Y2 = ConvY(Solution(Row).Step(N).Y) + 15

End If
Next N

B. Cartesian field trajectory
One of the issues that can put into the topic of discussion of AI is the search for trajectories in the field of Cartesian \([14][15][16]\). This problem applies the tracking tree method using the Breadth First Search algorithm to find possible solutions.

This issue is described with an object moving from the center point \((0,0)\) to point \((m, n)\) on the Cartesian plane provided that it must move parallel to the x-axis or y-axis (horizontal or vertical) and should not cross the path Which he has been through and should not cross the point he has been through.

Implementation of the tracking tree is as follows, and we first set the initial state as the root node (root node). Next, do the development of child nodes that may lead to the destination situation by complying with and not violating existing rules.

For example, the object's destination point is the point \((m, n) = (3, 2)\). Rules, objects may only move within quadrant I (positive x-axis and positive y-axis), the movement of the object should not exceed the maximum limit, \(x = 3\) and \(y = 2\), then with the tree obtained the possible movement toward the state of destination is as follows:

![Figure 2. (a) First Solution, (b) Second Solution](image)

3. Result and Discussion
The path search that can pass by an object in the Cartesian field could be accomplished with the help of a tracking tree, then at the point to go through a series of rules that restrict the movement of the object. Restrictive provisions are the maximum permissible limit of movement, the position of the obstacle to which the object should not pass, the position where the object must pass and cross only the allowed quadrant to pass, starting from the initial position as a root node, the BFS algorithm then finds a solution by developing root nodes to the next levels, all possible moves, not violating the terms and producing new conditions as fully developed. Search ends when no new nodes or conditions could be revealed. All the nodes that are the destination position are the solutions.

The solution of the shortest path finding solution using BFS method in this research is different from the previous research, the search process is perform in Cartesian area and every point in cartesian area will be calculated as the possibility of the passing path so the search process will be more optimal and from the traceable solution can be known. See the movement process of objects that are allowed are:

a. Move right \((x + 1)\).

b. Move left \((x - 1)\).

c. Move up \((y + 1)\).
d. Move down (y - 1).
e. All object movements above are legal if they comply with the following provisions:
   1) Not crossing the point of obstacle positions.
   2) Not crossing the post of a point that has been pass before.
   3) Does not exceed the maximum movement limit of the object (x or y).
   4) Not crossing quadrants that are not allowed to pass. The quadrant in question is 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).

The process of searching the solution search using the Breadth First Search algorithm are performed in the following Cartesian case:

![Figure 3. Sample case](image)

Based on tests carried out, several solutions can be taken to achieve the desired goal, here is a solution obtained:

| No | Steps | Few Solution Step |
|----|-------|-------------------|
| 1  | 7 Step Found 8 Solution | ![Solution 1](image) |
| 2  | 9 Step Found 61 Solution | ![Solution 2](image) |
Testing was accomplished in the area of Cartesian to locate the path through which the object produces some options that could be passed, with the most optimal solution consists of 7 steps.

4. Conclusion
The Breadth First Search algorithm applied to determine the path from the original location to the destination location in the Cartesian field yields several alternative solutions based on the tests performed. Thus the Breadth First Search algorithm applied to the Cartesian field can produce some optimal solution (shortest path) and solution which is non-optimal.
5. References

[1] Y. Zhou, W. Wang, D. He, and Z. Wang, “A fewest-turn-and-shortest path algorithm based on breadth-first search,” Geo-Spatial Inf. Sci., vol. 17, no. 4, pp. 201–207, 2014.

[2] J. Silvela and J. Portillo, “Breadth-first search and its application to image processing problems,” IEEE Trans. Image Process., vol. 10, no. 8, pp. 1194–1199, 2001.

[3] P. Ossona de Mendez and P. Rosenstiehl, “Coding Properties of Breadth-First Search Orderings,” Electron. Notes Discret. Math., vol. 5, pp. 253–255, 2000.

[4] S. Beamer, K. Asanovic, and D. Patterson, “Searching for a Parent Instead of Fighting Over Children: A Fast Breadth-First Search Implementation for Graph500,” Tech. Rep. UCB/EECS-2011-117, EECS Dep. Univ. California, Berkeley, pp. 1–9, 2011.

[5] D. Ajwani, R. Dementiev, V. Osipov, and U. Meyer, “Breadth first search on massive graphs,” shortest path Probl. ninth DIMACS implementation Chall., vol. 74, pp. 291–307, 2009.

[6] A. S. M. Lumenta, “Perbandingan Metode Pencarian Depth-First Search, Breadth-First Search Dan Best-First Search Pada Permainan 8-Puzzle,” e-journal Tek. Elektro dan Komput., 2014.

[7] S. Beamer, A. Buluc, K. Asanovic, and D. Patterson, “Distributed memory breadth-first search revisited: Enabling bottom-up search,” in Proceedings - IEEE 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum, IPDPSW 2013, 2013, pp. 1618–1627.

[8] A. V. Goldberg, S. Hed, H. Kaplan, R. E. Tarjan, and R. F. Werneck, “Maximum flows by incremental breadth-First search,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2011, vol. 6942 LNCS, pp. 457–468.

[9] K. L. Wright, J. I. Hamilton, M. A. Griswold, V. Gulani, and N. Seiberlich, “Non-Cartesian parallel imaging reconstruction,” Journal of Magnetic Resonance Imaging, vol. 40, no. 5. pp. 1022–1040, 2014.

[10] F. Porikli, “Integral histogram: A fast way to extract histograms in Cartesian spaces,” in Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 2005, vol. I, pp. 829–837.

[11] J. Wang, J. Wang, G. Zeng, R. Gan, L. Shipeng, and B. Guo, “Fast neighborhood graph search using cartesian concatenation,” in Multimedia Data Mining and Analytics: Disruptive Innovation, 2015, pp. 397–417.