IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 20, Issue 3, Ver. III (May. - June. 2018), PP 20-23
www.iosrjournals.org

Cyclomatic Complexity Analysis of Boundary Iterative Deepening Depth First Search (Bides)

[1]olowoye A. O., [1]oyeleye C. A., [1]oladosu J. B, [2]amumeji O. T., [1]olabiyisi S. O.

[1]Dept.of Computer Science and Engineering, Ladoke Akintola University of Technology (LAUTECH), Ogbomoso, Nigeria.
[2]Dept. of Mathematical Sciences, Ondo state University of Technology, Okitipupa, Nigeria.
Corresponding author: olowoye A. O

Abstract: *Boundary iterative-deepening depth-first search (BIDDFS) developed by [1] is used to allow searching from the starting node and the goal node simultaneously, yielding higher pathfinding efficiencies. It is based on the idea that faster pathfinding will result if the search expansions of a pathfinding algorithm is to be performed from both the starting node and goal node concurrently, due to the resulting expansion area being smaller than that of a unidirectional search. Hence, bidirectional search will also lead to a reduction of memory footprint. Experiments in [1] showed that the proposed algorithm was able to record drastic improvements in pathfinding speeds compared to the standard BIDDFS. Likewise, the fast bidirectional BIDDFS recorded significant speed improvements over the parallel bidirectional BIDDFS. In this paper, the cyclomatic behavior of the algorithm presented in [1] is described. It is shown that the cyclomatic analysis of the algorithm shows that the algorithm is simple and doesn't contain much risk.*

Keywords: *pathfinding; visually impaired; uninformed search; bidirectional search, Computational Complexity.*

Date of Submission: 31-05-2018 Date of acceptance: 16-06-2018

I. Introduction

Human navigation is defined as the process for a human to travel between two points on a location, usually associated with the act of walking or running. Navigation systems are electronic devices that assist human navigation by planning and guiding the user along a route during navigation. Therefore, the process of navigation starts by identifying the position of the user (positioning); and then a route is planned for the user to take during travelling (pathfinding); and finally, the user guided by the planned path, travels to the destination (guidance).

Pathfinding in navigation is defined as the plotting by a navigation system to find the best route between two points [2]. Routes from pathfinding are normally optimized for one more of the following: shortest distance, fastest travelling time, or lowest cost. Navigation system employs pathfinding algorithms to perform pathfinding [3]. The bidirectional BIDDFS offers faster pathfinding by searching from both the starting and the goal until both searches meet. This effectively reduces the search area compared to unidirectional search algorithms. As a result, time and memory consumption is reduced. In this paper, the asymptotic time complexity of the new algorithm is analyzed.

II. Boundary Iterative Deepening Depth First Search (Bides)

According to [1], to index the expanded nodes from each direction (starting node direction and goal node direction), a new variable is introduced as a two-dimensional matrix with the same dimensions of the field map, with each value on the matrix representing direction, and their locations reflecting the location on the field map. This new variable is required so that the algorithm can determine whenever and wherever the two expansion directions meet, the pathfinding process will then terminate.

The location where the two expansion directions meet is called the meeting node. For each completed pathfinding process, there are two meeting nodes – each representing the direction it originates [4]. Each meeting node will plot its way back in the opposite direction of the other using the calculated least cost path to ensure the shortest path from the meeting node to the starting or goal node. Once the route plotting is completed, there would be a path connecting the starting node to the ending node, which is the resultant route [4]. This search pattern is as illustrated in Figure 1, with the line connecting the two starting nodes being the resultant path, with the expansions from each originating node as shown.

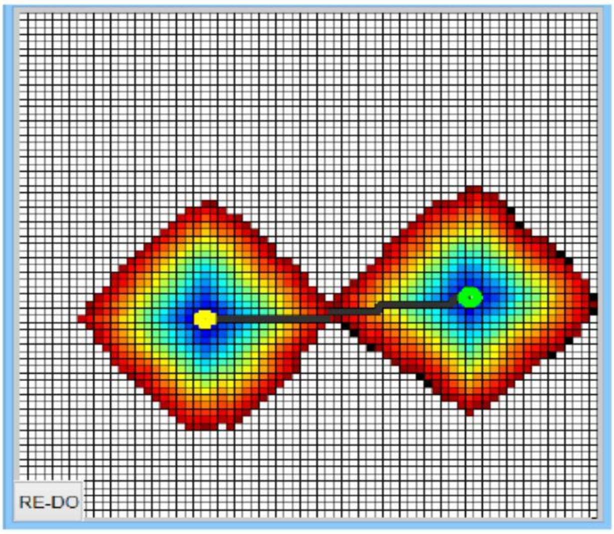

Figure 1: Search pattern of the bidirectional BIDDFS [1]

Pathfinding for navigation can be optimized for faster speeds. For example, assumptions such as BIDDFS will perform on uniform grid maps allow the cost across the map to be uniform, simplifying calculations. Figure 2 shows the flowchart of the bidirectional BIDDFS.

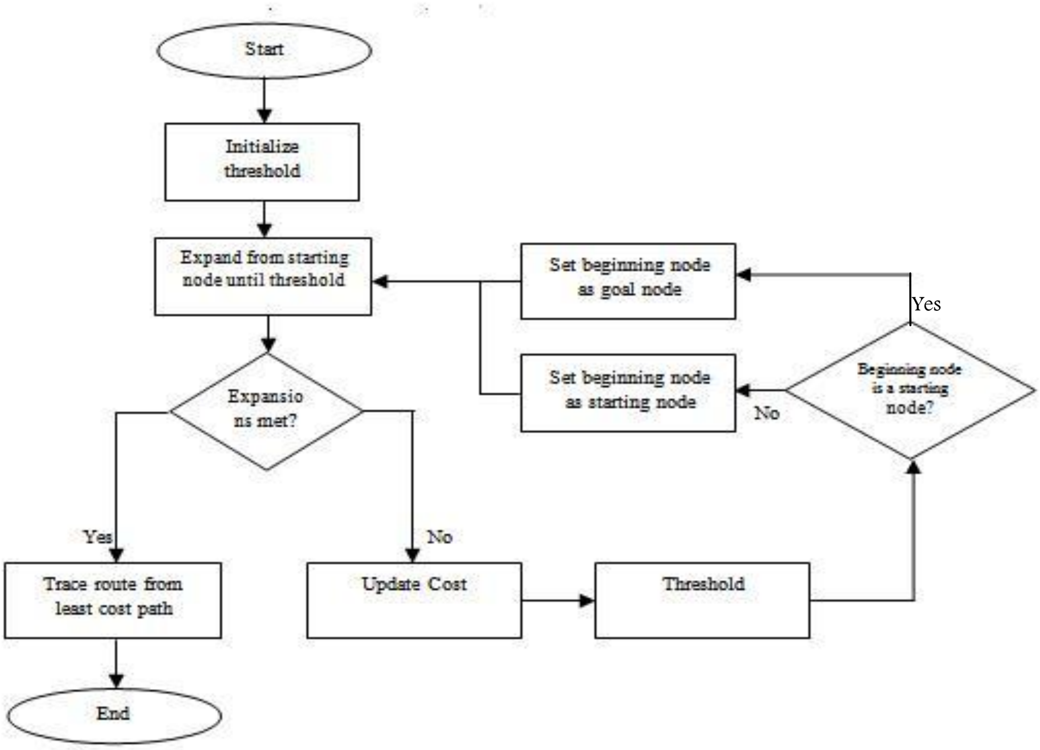

Figure 2: Flowchart of the bidirectional BIDDFS [1]

III. Cyclomatic Complexity

Software complexity metric is quantitative measure of the quality of software or measure of the difficulty of testing, understanding, or maintaining a piece of software or measure of ease of using software [5, 6, 7 and 8].

Software Complexity is a measure of the resources which must be expended in developing, maintaining, or using a software product. A measure of the complexity of a program was developed by [9]. He developed a system which he called the cyclomatic complexity of a program. This system measures the number of independent paths in a program, thereby placing a numerical value on the complexity. In practice it is a count of the number of test conditions in a program. The cyclomatic complexity (CC) of a graph (G) may be computed according to the following formula:

CC(G) = Number (edges) - Number (nodes) + 2 1 The results of multiple experiments [10] suggest that modules approach zero defects when McCabe's Cyclomatic Complexity is within 7 ± 2. A study of PASCAL and FORTRAN programs [11] found that a Cyclomatic Complexity between 10 and 15 minimized the number of module changes. A common application of cyclomatic complexity is to compare it against a set of threshold values. One such threshold set is in Table 1:

Table 1: Cyclomatic Complexity

Cyclomatic Complexity	**Risk Evaluation**
1-10	a simple program, without much risk
11-20	more complex, moderate risk
21-50	complex, high risk program
greater than 50	Un-testable program (very high risk)

3.1 Theorem

The cyclomatic complexity for the bidirectional boundary iterative-deepening depth first search is simple without much risk.

3.2 Proof

From the flowchart of bidirectional BIDDFS presented in Figure 2, the cyclomatic flow graph of the algorithm was derived, which is presented in Figure 3.

$$V(G) = e - n + 2$$
$$= 16 - 15 + 2$$
$$= 3$$

It can be seen that the cyclomatic complexity of the bidirectional BIDDFS is “3” which means that the algorithm is simple and it contains not much risk.

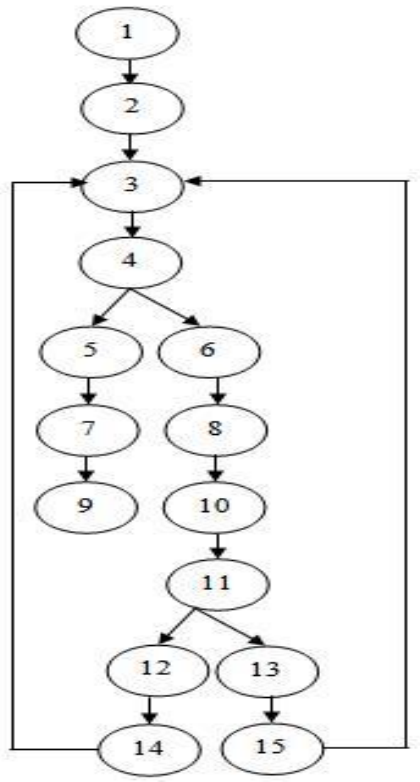

Figure 3: Flow graph of the bidirectional BIDDFS

IV. Discussion and Conclusion

This paper has examined the cyclomatic complexity of the bidirectional boundary iterative-deepening depth first search. The algorithm was analyzed and proved that the cyclomatic complexity of the algorithm is simple and without much risk.

References

[1]. Lim, K.L., Seng, K.P., Yeong, L.S., Ang, L-M. and Ch"ng, S.I. (2016) „Pathfinding for the navigation of visually impaired people", Int. J. Computational Complexity and Intelligent Algorithms, Vol. 1, No. 1, pp.99–114.

[2]. Di Giampaolo, E. (2010) „A passive-RFID based indoor navigation system for visually impaired people". Paper presented at the 2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL), 7–10 November.

[3]. Lim, K.L., Seng, K.P., Yeong, L.S., Ch"ng, S.I. and Ang, L-M. (2013) „The boundary iterative-deepening depth-first search algorithm", Paper presented at the Second International Conference on Advances in Computer and Information Technology – ACIT 2013, Kuala Lumpur, Malaysia.

[4]. Olabiyisi S.O., Akanmu T. A., Oyeleye C. A., Sobayo O. D. and Adelana J. B. (2007). „Complexity Analysis of a New Edge-Adaptive Zooming Algorithm for Digital Images". Journal of Research in Physical Sciences, Vol. 3, No. 2, pp. 67-71.

[5]. Olabiyisi S. O., Omidiora E. O., Adetunji A. B. and Dele Oluwade. (2003). „Asymptotic Time Complexity of Maximum Error Estimation (MEE) Algorithm for the Tau Numerical Method". An International Journal of Biological and Physical Sciences.Vol. 3, pp. 92-94.

[6]. Olabiyisi S. O., Okeyinka A. E., Omidiora S. O., Sobayo O. D. (2003). „Asymptotic Time Complexity of Polynomial Approximant Algorithm for the TAU Numerical Method". Journal of Science and Technology Research, Vol. 2, No. 3. Pp. 40-42.

[7]. Olabiyisi S.O. (2005). „Universal Machine for complexity measurement of computer programs". Unpublished Ph.D Thesis, Department of Pure and Applied Mathematics, LAUTECH, Ogbomoso.

[8]. Olabiyisi S.O., Omidiora E. O. Adigun A.A. and Adeosun O. O. (2005) „Asymptotic Time Complexity of Elements of Matrices A and B Generation Algorithm for the Tau Numerical Method". Journal of Applied Sciences Vol.813, pp 4935-4941.

[9]. T. J. McCabe (1976). „A Complexity Measure". Journal IEEE Transactions on Software Engineering archive. Vol. 2(4), pp 308-320.

[10]. V.S. Dimitrov, G.A. Jullien and W.C. Miller (2000). „Complexity and fast algorithms for multiexponentiations". IEEE Transactions on Computers Vol. 49(2), pp 141 – 147.

[11]. Randy K. Lind and K. Vairavan (1989). „An Experimental Investigation of Software Metrics and Their Relationship to Software Development Effort". IEEE Transactions on Software Engineering 15(5):649 – 653.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

olowoye A. O "Cyclomatic Complexity Analysis of Boundary Iterative Deepening Depth First Search (Bides)." IOSR Journal of Computer Engineering (IOSR-JCE) 20.3 (2018): 20-23.