Performance Analysis of BFS and DFS Algorithms for Food Serving Robot in an Eatery

P Navya and R Ranjith

May 27, 2021

Performance Analysis of BFS and DFS Algorithms for Food Serving Robot in an Eatery

P Navya
Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
navyapulimi@gmail.com

R Ranjith
Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
r_ranjith@cb.amrita.edu

Abstract— **Technological advancements in the industry4.0 has huge out-turn in the automation of food industry to an incredible extent. Consequently, the serving robot in an eatery has all the aspects of critical importance in terms of fast and safe food delivery to the customers. Highlighting on that note, this paper has the comparative study on the performance of two major algorithms namely BFS and DFS and they are implemented on a 10x10 virtualized grid floor plan of an eatery which are simulated in MATLAB. The comparison is done on the performance attributes like optimality, completeness, time complexity and space complexity in which both Breadth First Search (BFS) and Depth First Search (DFS) are equally optimal and complete to find a path for serving robot. It is observed that the average time consumption and average space consumption is lesser in the BFS algorithm. The time complexity and space complexity being the key attributes had a significant variation in the statistical results and concluded BFS as the better algorithm for the application of serving robot in an eatery.**

Keywords— food automation, serving robot, BFS, DFS, graph-based planning, optimal path, time complexity and space complexity

I. INTRODUCTION

The use of control frameworks, such as PCs or robots, and information technologies for dealing with numerous cycles and machineries in an industry to replace a person is known as industrial automation. As this is the hour of industry 4.0, robotized advancement has been associated in various points of human life. Automation trends in India are geared to centralization by robots replacing humans in many industrial firms for better productivity and safety. Putting robots in benefit trade such as robot serving in an eatery is exceptionally promising industry within the close future [3]. Eatery could be a place where individuals come, sit and eat dinners that are cooked and served by the waiters separately. In traditional eatery framework, orders are taken by a server and they bring the nourishment when it is prepared. This framework depends on expansive numbers of labour to handle client reservation, request around them, ordering food and arranging food on the tables. So, the automated serving robot in the eatery will drastically reduce the investment and saves time with minimal human interference in the present framework [2]. In this work a platform is created on the assumption of eatery floor with the placements of tables to find the optimal path planning algorithm for an automated mobile robot.

According to the known degree of the robot environment, the versatile robot path planning maybe divided into two types: global map path planning and local map path planning. The global map path planning can produce the way beneath the totally known predefined environment which includes two parts: establishment of the environmental global map as the occupancy grid model and the path planning strategy as (BFS) and Depth First Search (DFS) respectively for serving robot in an eatery. The grid-based path planning method appears to be a viable alternative to traditional methods that rely on minimum environmental data [4]. BFS and DFS are thought to be capable of improving the performance of a serving robot at an eatery by finding the shortest path based on their features. In this work a 10x10 grid environment has been created in MATLAB for simulating the behavior of a mobile robot under the decision-making logic of BFS and DFS algorithms to plan and follow paths. The main goal of this project is to implement and analyze both BFS and DFS algorithms with performance attributes like completeness, optimality, time complexity and space complexity.

The following chapters make up this paper: chapter 2 contains the literature review; chapter 3 discusses the system overview and methodology. The simulation findings are presented in chapter 4 and the project's conclusion is described in chapter 5.

II. LITERATURE REVIEW

A. Background of robotic eateries:

The present scenario, advancement and strategies of robotics in food industry is clearly discussed in [1] with accurate statistical results. In the fourth age, we tend to experience vital advances globally with artificial intelligence within the robotic industry [2,3]. In the food trade, AI is additionally attracting attention as a result of assists to scale back expenditure by increasing outcome in terms of service fastness, ordering approach and manufacturing environment. [4,5], therefore their utilization is widely implemented in robotic restaurants worldwide.

Pieska in [4] have described about a social service robot called CENTRIA. The author developed a Robosoft's Komapi-robot with Finnish voice recognition and created a touch screen application for restaurants. The service menu also has different applications with video-calling. The author states that robot's service and navigation has been tested in two Finnish restaurants. Now, in this work the kind of navigation for robot and eatery floor plan is assumed on an occupancy grid environment with seating tables placed as obstacles in MATLAB.

In November 2018, JD.com, the world's largest online Chinese retailer, unveiled the world's first smart robotic restaurant which encompassed the entire procedures of food manufacturing, processing and table service. The theoretical and managerial intentions on the robotic restaurant management have been seen in [6]. By 2020, the business opened up to more than 1000 locations.

Jamshed in [7] states that, food service sector has the greatest R&D potential. This paper has a long research study on the forecast of robotics in automating the food industry. In comparison to manual production systems, the- state-of-the-art demonstrates that the realm of robots has dramatically enhanced productivity. This also contains the detailed explanation for the need of robotics in food industry as well as the types of robots which can be used in food industries.

B. Study on graph-based path planning methods:

The complete definition and types of path planning in accordance with automation of robots in food industry has been discussed in "Robotics and Automation in the food industry" [8]. This book also provided the knowledge on current trends in food industry like Optical sensors, live spectroscopy, gripper technologies and wireless sensor networks (WSN).

In [9], Galceran claims that a serving robot goes via all points of uncertain nodes in order to find the goal. It is said to be used for a variety of tasks, including painting, cleaning and operating an underwater vehicle. Here, a grid-based technique with nodes in linear or circular dimension is used. The likelihood is determined by threats along the path, distance and other factors. For each targeted site, beacons are deployed to survey a certain region. This project is primarily concerned with the building of a grid- based map for an eatery environment.

Grid-based planning approach was conferred by Vaibhav Jain in [10]. It can be used in material handling applications to convey objects from one node to other node and the nodes are supposed to be in any dimension of 2D or 3D. A line follower robot is employed in this work, along with sensors to detect the path. In a grid context, concerns like detection of obstacles as the consideration of points in a coordinate system was well explained.

The knowhow of performance analysis of a particular algorithm called A* for the application of firefighting robot is described in [11]. The two algorithms named A* and Dijkstra are simulated and analyzed with performance attributes called completeness, optimality, time complexity and space complexity with statistical results performed on the different zones of goals. The considered robot was a hexapod robot with KRPAI 2016 map as floor plan.

In conclusion, it can be summarized with a few key themes such as the necessity for robot automation in the food industry and current trends in the food automation. The importance of serving robots can be mainly considered on the factors of productivity and time saving. The path finding algorithms have a vital role in providing productivity in terms of serving foods in less time. The discussed methods of path planning and grid-based navigation provided the knowledge on using BFS and DFS for this paper.

III. System Overview and Methodology

A. Robots in an eatery environment:

The real-world scenarios are casted as graphs to find the optimal path finding algorithms. In this paper the graph of an eatery is represented as occupancy grid-based floor plan with color notations. The proposed environment consists of seating tables arranged in a discrete manner with serving robots placed at starting positions on 10x10 grid-based floor plan. These robots' position is updated when it moves to successive node. So, the start positions are serving robots places and goal positions for the serving robots are table positions with respective arrangement on the occupancy grid of proposed environment. The use case for this system is observed when a customer reaches eatery and approaches the setting tables in the proposed manner. The proposed work focuses solely on determining the most effective path planning algorithm for a serving robot in a grid-based environment. The two considered algorithms for performance analysis are BFS and DFS respectively. The grid-based eatery plan is illustrated in the figure 1. With T1, T2, T3 …... T16 depicts the table arrangement and serving robots at the waiting area.

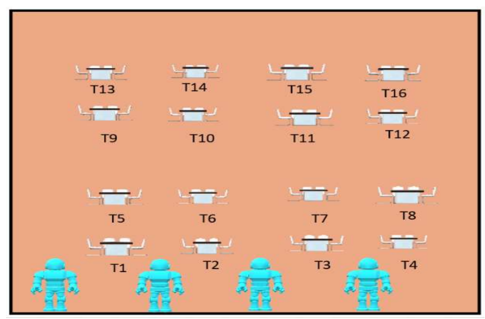

Fig.1: Floor plan of an eatery

B. DFS algorithm:

DFS is one of the main graph transversal algorithms which is used to search a goal node in a graph. The functionality of DFS is to go as deep as possible in search of a node and traceback once the node is found. This algorithm can be implemented by using stack operations in which a stack is created and origin is marked as visited as it is pushed into stack. Iterate while stack is not empty and pop a node from stack by pushing all its adjacent nodes into stack. For a vertex to get pushed into stack it shouldn't be visited. The pseudocode for DFS algorithm is shown in fig.2.

```
DFS method (source):
    Create a stack S
    Add origin to stack S
    Mark origin as visited
    while S is not empty:
        take an element from the stack S called A
        for each vertex B adjacent to A in the Graph:
            if B has not been visited:
                Mark as visited B
                Insert B into the stack S
```

Fig.2: Pseudocode for DFS algorithm

The recursive method of DFS algorithm yields better results where the process is recursively repeated until the goal node is found. The pseudocode for recursive DFS algorithm is shown in fig.3.

```
DFS method (source):
    Mark origin as visited
        for each vertex B adjacent to origin in the graph:
            if B has not been visited:
                Mark as visited B
                Recursively call DFS (B)
```

Fig.3: Pseudocode for recursive DFS algorithm

C. *BFS algorithm:*

Another graph-based planning approach is the BFS algorithm, which is mostly used to discover the shortest path between two nodes is shown in fig.4 with pseudocode It explores all the nodes at the present level before exploring the nodes at the next level. This algorithm can be implemented using queue operations in which a queue is created and origin is marked as visited. Iterate while queue is not empty and remove a node from queue by enqueueing all its adjacent nodes into queue. For a node to get inserted into queue it shouldn't be visited.

```
BFS method (Graph, origin):
    Create a queue Q
    Add origin to queue Q
    Mark origin as visited
    While Q is not empty:
        Remove an element from the Q queue called A
        for each vertex B adjacent to A in the Graph:
            if B has not been visited:
                Mark as visited B
                Insert B into the queue Q
```

Fig.4: Pseudocode for BFS algorithm

IV. Implementation and Results

In general, the eatery is constructed in populous urban areas which is easily navigable by customers. In order to replace human servers with serving robots the floor plan has to be well equipped and well designed. A well-designed floor plan acts as a blueprint for better planned eatery business. The purpose of serving robot in an eatery is fulfilled only when it serves food in adequate time. So, by the virtual analysis of real-life eatery this experimental eatery's floor plan is depicted in methodology. In order to provide outstanding food delivery functions, the serving robot has to follow optimal path finding algorithm. The optimal path finding algorithm between DFS and BFS was observed by following this implementation process.

The implementation process starts with the creation of occupancy grid of 10x10 in MATLAB and by allotting the table arrangements as obstacle nodes in the grid as shown in the floor plan fig.1. The positioning of start node (robot waiting position) and the positioning of goal node (seating table position) are customized, but the number of seating tables are limited to 16 in this implementation which can further be included and one mobile serving robot is considered to reach the goal node to serve food. The implementation of BFS and DFS can be visually analysed from the flowchart fig.5, where the process flow of BFS is a bit varied from DFS process. The difference between them is the addition of nodes to the fringe set. In DFS, new nodes are added at start of the fringe set as it follows stack operations whereas in BFS, new nodes are added to the end of the fringe set as it follows queue operations.

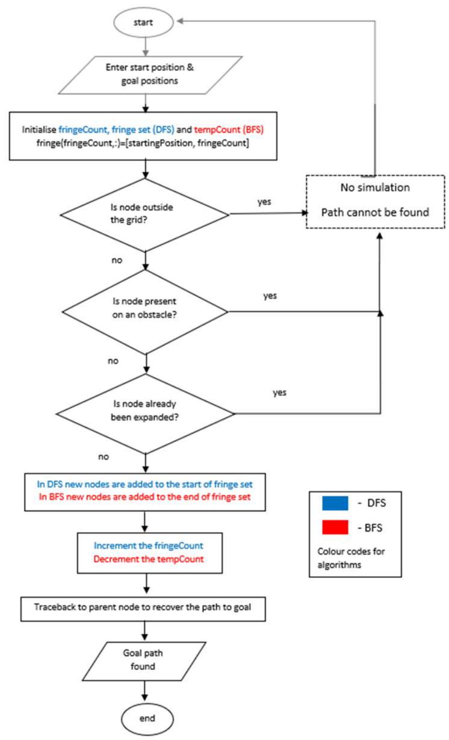

Fig.5: Flowchart for BFS and DFS algorithms

The color codes are considered in the flow chart for referring DFS with blue color and BFS with red color. The color highlighted text depicts the variations in the implementation process of two considered algorithms. Results are evaluated for different test conditions by considering different sets of start node and goal node coordinates. The respective considered sets of nodes are tabulated as below:

Table 1: paths notations for different test conditions in the simulation

Start node	Goal node	Path notations
(1,1)	(5,4)	Path A
(1,3)	(4,7)	Path B
(4,2)	(7,9)	Path C
(6,1)	(2,5)	Path D
(4,5)	(8,3)	Path E

The results are statistically tabulated with the attributes of performance analysis of an algorithm. The considered attributes are completeness, optimality, time complexity and space complexity.

A. Completeness:

From the results of BFS and DFS simulations shown in below figures.6, 7, both the methods are complete as they provide paths to every goal node.

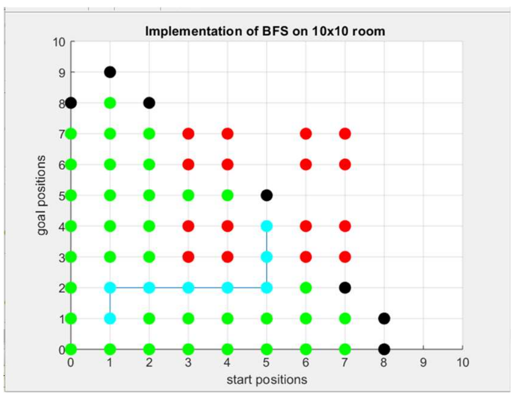

Fig.6: Simulation result of BFS in MATLAB

B. Optimality:

BFS is optimal since it offers the shortest path from the start node (robot position) to the goal node (seating table). The results shown in figures 6,7 are simulated with (5,4) coordinate point as table position in both BFS and DFS.

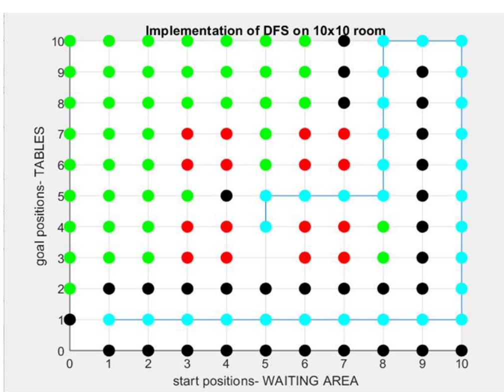

Fig.7: Simulation result of DFS in MATLAB

C. Time Complexity:

The time complexity for both BFS and DFS are measured by the time consumed to find the path from start node to goal node. The consumed time results are tabulated as below:

Table 2 Average execution time for simulation

Path notations	BFS (seconds)	DFS (seconds)
A	4.065	5.079
B	4.740	6.056
C	8.619	8.986
D	5.650	7.122
E	5.763	6.755

D. Space Complexity:

The space complexity for both BFS and DFS are measured by the space occupied by them in bytes to form the path from start node to goal node. The occupied space results are tabulated as below:

Table 3 Average memory consumption for simulation

Path notations	BFS(Kbytes)	DFS(Kbytes)
A	1848	2520
B	1632	2520
C	2280	2520
D	1368	1368
E	1200	2328

CONCLUSION

The automation era results in the increase of robotic life by making human life easy. Food industry in industry 4.0 is an imperative trend for replacing humans with robots in the terms of serving of food [14]. The purpose of this paper is to correlate the performance of two graph-based algorithms, BFS and DFS for the use of serving robots in eateries. The results are analyzed for the respective floor plan of an eatery with serving robots to drive the useful conclusions from the performance attributes of BFS and DFS algorithms. It is clearly evident that the BFS is an optimal path finding algorithm as it provides the shorter path than DFS. The completeness of both BFS and DFS is equalized with the results as they provide paths for all goal nodes. Lastly, the average time consumed by BFS is 5.7674sec and for DFS is 6.7995sec. This means the time complexity of BFS is lesser than DFS proving that BFS is a faster algorithm. Even the space complexity of BFS is much lesser than DFS stating that BFS has average memory consumption of 1665.6Kbytes but the DFS has 2251.2Kbytes. The wide-reaching experimental results conclude BFS as an outstanding algorithm that helps the serving robots to find the paths in an eatery accordingly and serve the food in adequate time.

REFERENCES

[1] A. A. Hidayatullah, A. N. Handayani, and M. J. Fuady, “Performance analysis of A∗ algorithm to determine shortest path of fire fighting robot,” *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*. 2017, doi: 10.1109/siet.2017.8304108.

[2] A. Antony and P. Sivraj, “Food Delivery Automation in Restaurants Using Collaborative Robotics,” *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2018, doi: 10.1109/icirca.2018.8597280.

[3] P. Joshy and P. Supriya, "Implementation of robotic path planning using Ant Colony Optimization Algorithm," *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016, pp. 1-6, doi: 10.1109/INVENTIVE.2016.7823281.

[4] Pieska, Sakari, Mika Luimula, Juhana Jauhiainen, and Van Spiz. "Social service robots in wellness and restaurant applications." *Journal of Communication and Computer* 10, no. 1 (2013): 116-123.

[5] Asif, M., M. Sabeel, and K. Z. Mujeeb-ur Rahman. "Waiter robot-solution to restaurant automation." In *Proceedings of the 1st student multi disciplinary research conference (MDSRC), At Wah, Pakistan*, pp. 14-15. 2015.

[6] J. Hwang, K.-W. Lee, D. Kim, and I. Kim, “Robotic Restaurant Marketing Strategies in the Era of the Fourth Industrial Revolution: Focusing on Perceived Innovativeness,” *Sustainability*, vol. 12, no. 21. p. 9165, 2020, doi: 10.3390/su12219165.

[7] J. Iqbal, Z. H. Khan, and A. Khalid, “Prospects of robotics in food industry,” *Food Science and Technology*, vol. 37, no. 2. pp. 159–165, 2017, doi: 10.1590/1678-457x.14616.

[8] D. G. Caldwell, “Robotics and automation in the food industry.” 2013, doi: 10.1533/9780857095763.

[9] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12. pp. 1258–1276, 2013, doi: 10.1016/j.robot.2013.09.004.

[10] Gehlod, Lalit, Vaibhav Jain, Mala Dutta, and D. K. Lal. "A Grid Based Robot Navigation by Using Priority Algorithm." *International Journal of Advanced Research in Computer and Communication Engineering* 2, no. 8 (2013).

[11] Hwang, Jinsoo, Kwang-Woo Lee, Dohyung Kim, and Insin Kim. "Robotic Restaurant Marketing Strategies in the Era of the Fourth Industrial Revolution: Focusing on Perceived Innovativeness." *Sustainability* 12, no. 21 (2020): 9165.

[12] V. A. Prasad, V. Akshatha Prasad, and R. Ranjith, “Intelligent Chatbot for Lab Security and Automation,” *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2020, doi: 10.1109/icccnt49239.2020.9225641.

[13] R. Ranjith, N. Krishna Prakash, D. Prasanna Vadana, and A. S. Pillai, “Smart Home Energy Management System—A Multicore Approach,” *International Conference on Advanced Computing Networking and Informatics*. pp. 363–370, 2019, doi: 10.1007/978-981-13-2673-8_38.

[14] S. M.MasudurRahmanAl-Arif, S. M. MasudurRahmanAl-Arif, A. H. Iftekharul Ferdous, and S. H. Nijami, “Comparative Study of Different Path Planning Algorithms: A Water based Rescue System,” *International Journal of Computer Applications*, vol. 39, no. 5. pp. 25–29, 2012, doi: 10.5120/4817-7058.