ISSN: 2633-4828 Vol. 4 No.1, June, 2022

*International Journal of Applied Engineering & Technology*

# BIDIRECTIONAL PATH PLANNING FORMATION IN TWO DIMENTIONAL OBSTACLE CONSTRAINTS CONFIGURATION

**Narayan Kumar[1] and Amit Kumar[2]**

[1]Research Scholer and [2]Professor, Department of Mechanical Engineering, National Institute of Technology, Patna

## ABSTRACT

*It is beneficial to give mobile robots the ability to function independently. It enables the removal of the need for human operators, which could have positive effects on safety and the economy. Most of the time, in order for a robot to be autonomous, path planners must be used, allowing the robot to carefully consider the best course of action when moving from one place to another. Global interest in autonomous surface vehicles is growing as a result of the advantages of increased efficiency and safety. This has increased interest in creating path planning techniques that can save expenses and time expenditure. The path planning started with very simple constraint of static obstacles is solved first with one start and one goal point. Some dynamic obstacles included further and analyzed the solution by the algorithm with one start and one goal point. The numbers of turnings and the angle of turn have been minimized. Finally moving toward the proposed solution bidirectional path planning in the paper.*

## I. INTRODUCTION

Robotics is a field that deals with the design, construction, operation, and application of robots. Robots are machines capable of performing tasks autonomously or semi-autonomously, often mimicking human actions or executing tasks in environments that are hazardous, tedious, or difficult for humans to handle. One of the fundamental challenges in robotics is efficient path planning, which involves determining the optimal or feasible path for a robot to reach its destination while avoiding obstacles and adhering to various constraints. Path planning is a critical aspect of robotic navigation and motion control. It enables a robot to move from one location to another in its environment effectively. The process of path planning involves; analysing the robot's surroundings, identifying obstacles, and finding a path that maximizes efficiency and minimizes potential risks. Several key elements are essential for understanding path planning in robotics like the presentation of the environment, the algorithm used for path planning, motion constraints, dynamic environment, and mode of path planning: local or global path planning.

Robots play an important role in our daily lives. It makes a variety of tasks easier and more precise to execute. Robots can function in both static and dynamic environments by mapping their work areas. Airports, hospitals, post offices, and train platforms are just a few of the places where robots can be used. Companies use robots to do specific tasks such as welding, material handling, joining machine parts, and so on. Handling potentially hazardous substances, fires, and toxic gaseous environments are just a few examples of extraordinary scenarios where robots are the only viable solution. Robots are being used in research to investigate harsh environments and the availability of rare minerals on asteroids, deep beneath the oceans, satellites, and in space. Path planning is a typical problem that involves determining the shortest path between two vertices, or start and finish points, in a given space. The route depicts the cheapest path from the starting location to the desired condition. Finding the shortest path between two points, on the other hand, is frequently insufficient. People may need to identify the shortest path between two or more points. This type of problem includes the travelling salesman problem [1], in which the path must pass through every vertex in the graph, vehicle routing problems [2], in which the vehicle delivers items to several predetermined places, and path planning for tourist sites [3]. These kinds of problems are classified as multi-point path planning problems.

Environment representation deals with, before planning a path, the robot needs to perceive its surroundings. This is typically achieved through sensors such as cameras, LIDAR (Light Detection and Ranging), or ultrasonic sensors. The data collected is then used to create a representation of the environment, often in the form of a map or grid. Once the environment is represented, path-planning algorithms come into play. These algorithms determine the best path for the robot to follow. Common approaches include Dijkstra's algorithm, A* (A-star) algorithm, Rapidly Exploring Random Trees (RRT), Probabilistic Roadmaps (PRM), and many more. Each algorithm has its strengths and weaknesses, making them suitable for different scenarios. Path planning also considers the robot's physical limitations and constraints. These include the robot's maximum speed, turning radius, and other mechanical limitations. The path planner ensures that the generated path is feasible for the robot to execute. In some cases, the robot operates in dynamic environments, where obstacles or the robot's surroundings change over time. In dynamic path planning, algorithms continuously update the robot's path to adapt to these changes and avoid collisions. Figure 1 shows the environmental classification of path planning problem.

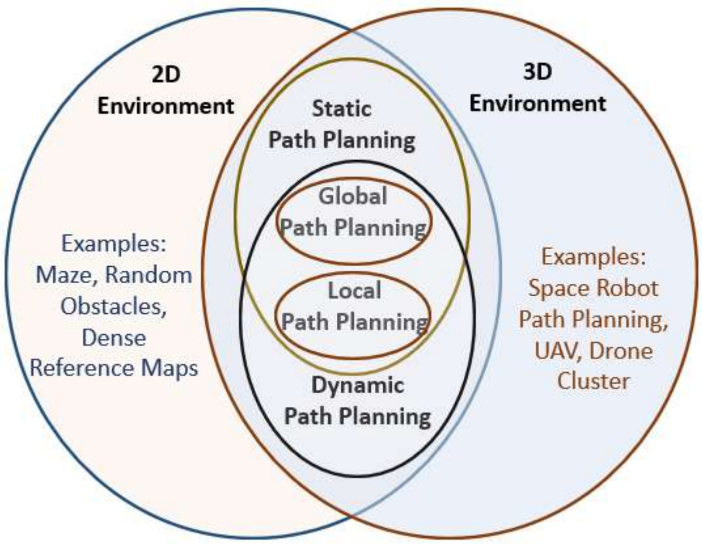



**Figure 1**: Venn diagram representing environmental classification of path planning problems [4]

Path planning can be divided into global and local planning. Global planning involves computing a path from the robot's initial position to its final goal. Local planning, on the other hand, focuses on handling immediate obstacles and fine-tuning the trajectory within the global path. Path planning is crucial in various robotic applications, including autonomous vehicles, industrial robots, delivery drones, and robotic exploration missions. By enabling robots to navigate safely and efficiently, path planning contributes to the advancement and integration of robotics into various industries and everyday life. As technology and research progress, path-planning algorithms continue to evolve, becoming more efficient and capable of handling increasingly complex and dynamic environments, thus driving the progress of robotics as a whole.

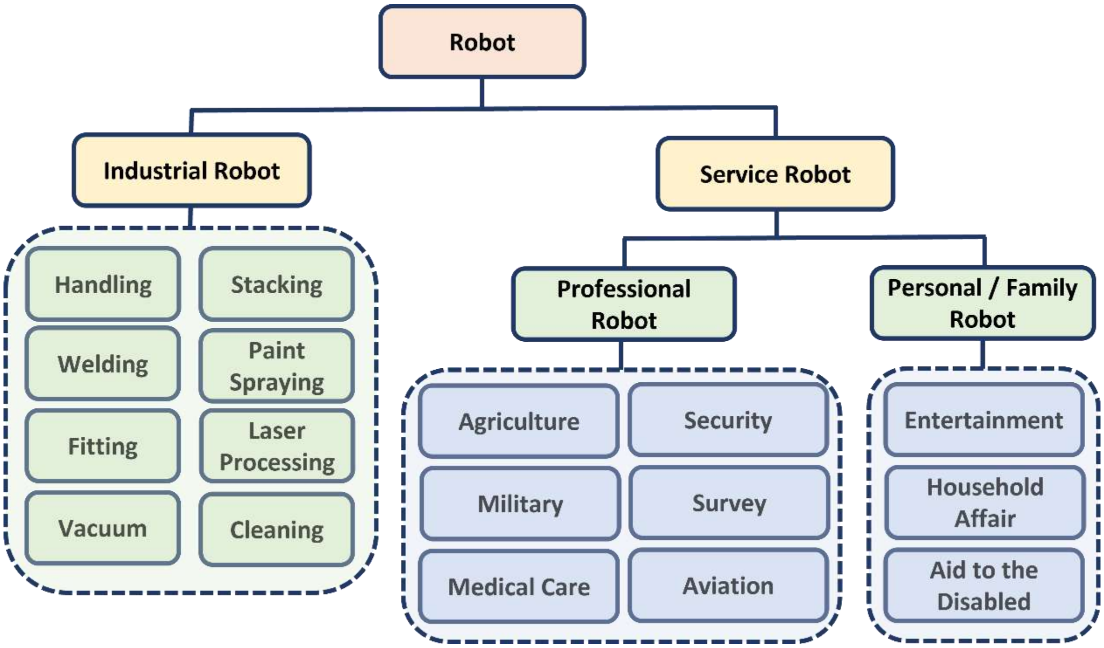



**Figure 2:** Classification of Robot Applications [5]

Robotics is a multidisciplinary field that deals with the design, construction, operation, and application of robots. A robot is a mechanical device that can be programmed to perform a range of tasks autonomously or semi-autonomously. The field of robotics combines elements of computer science, engineering mechanics, electronics, and artificial intelligence (AI) to create intelligent machines that can interact with and manipulate the physical world. Broadly the robots are classified in two categories of industrial robot and service robot. The classification of robot applications is shown in figure 2. The importance of robotics in various applications can be reviewed in manufacturing industries [6], healthcare, exploration of space, agriculture, logistics, warehousing, transportation, environment monitoring, education, research, entertainment, defence, and many more.

## II. RELATED WORK

In the field of robotics, to mitigate the dependency of human intervention for flowless output; autonomous navigation is a very helpful asset in mobile robots. Path planning is one of the most important factors considered in autonomous navigation. Path planning is about finding the best route to the desired state from the start state. In the given condition there may be many possible paths from which the robot leads to the desired location. Path planning algorithms search the optimal paths to the given constraints. The sense of the optimal path is to minimize or optimize one or many objective functions at once. The objective functions may be in terms of the amount of time, distance of travel, number of turns in the path, energy of the robot, smoothness of path, and many more. Other constraints in path planning may be considered as the nature of obstacles, effectiveness of the algorithms, terrains, etc. The robots involved in the search and rescue [7] of victims of any disaster are critically important concerning the amount of time to save the life. In other situations, like planetary exploration, the energy of the robot, irregular gravity field, and complex terrain [8] are important factors to be considered as the rovers have limited energy resources available.

Path planning is an active and important research topic in the field of robotics. Generally, it starts with two-dimensional planner path-finding solutions but it is not restricted to these planner solutions only. The planner environment model is classified as static and dynamic. In practice, the environment is usually dynamic in nature; e.g. the passengers on the railway platforms, people at shopping complexes, students in schools, and many more. In this section, we have listed a few works of researchers who designed or improved algorithms for path planning concerning static and dynamic environmental conditions.

**General Classification**

The suggested classification takes into account the behaviour of path-planning algorithms. Many previous studies distinguished between two types of path planners: online and offline, depending on whether the environment is dynamic or not. A mobile robot's path planning is used to establish a collision-free path from a starting point to a goal point while optimizing a performance criterion such as distance, time, or energy, with distance being the most typically used criterion. Global path planning refers to offline path planning for robots in contexts where comprehensive knowledge about stationary obstacles and the trajectory of moving objects is available ahead of time. When complete information about the surroundings is not accessible ahead of time, the mobile robot gathers information as it goes through the environment using sensors. This is referred to as online or local path planning [9]. Essentially, online path planning starts its original path offline but switches to online mode when it detects fresh changes in obstacle scenarios.

Because of its great computational speed, a non-replanning approach might be employed online. The opposite could also occur. For example, the Dynamic Window Approach (DWA) is a Reactive Computing approach that is commonly used for local planning [10], but can also be utilized for global planning. A novel evaluation function is constructed using the result of global path planning as a reference trajectory, ensuring the optimal trajectory [11]. Furthermore, the study proposes three assessment sub-functions: direction, smoothing speed, and acceleration, which ensure the direction, smoothness, and speed of movement, respectively. Figure 3 shows the general classification of path planning of mobile robots.

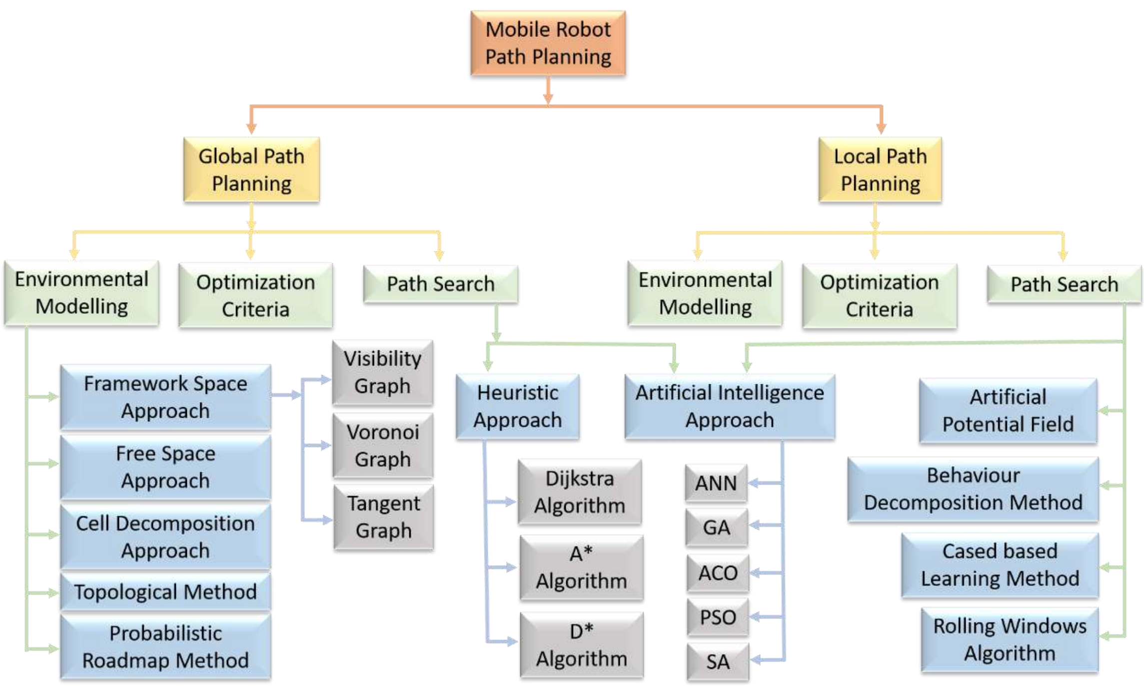



**Figure 3**: Classification of path planning [10]

Autonomous surface vehicles are gaining popularity worldwide because of the potential benefits of increased safety and efficiency. This has sparked interest in creating path-planning techniques that can reduce the probability of collisions, groundings, and stranding mishaps at sea, as well as expenses and time spent. An interesting distinction exists between algorithms that require a preliminary map representation (Classic) and those that do not (Advanced). Classic contains Graph Search methods, whereas Advanced focuses on Soft Computing and Sampling-Based algorithms. Figure 4 shows visual representation of the distinction of path planning terms in the literature [12]. Path planning classifications are clear and reasonable [13]: according to the robot model (holonomic, non-holonomic, kinodynamic); according to the map model requirement (needed or not beforehand); according to the replanning capability (offline or online); and according to whether the algorithm always calculates the same solution, according to preliminary configuration parameters (deterministic or probabilistic).

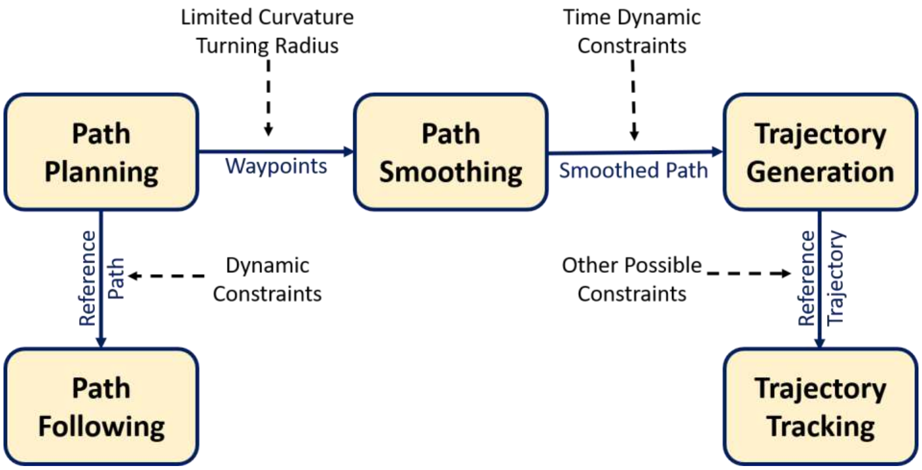



**Figure 4**: Visual representation of the distinction of path planning terms used in the literature [12]

Robot autonomous navigation is an attractive research subject due to its wide range of applications. Navigation consists of four key requirements: perception, location, cognition, path planning, and motion control, with path planning being the most significant and intriguing. The proposed path-planning strategies are divided into two categories: classical and heuristic methods [14]. The classical methods include cell decomposition, potential field method, subgoal network, and road map. The procedures are simple, but they frequently need expensive computation and may fail when the robot encounters ambiguity. In contrast, heuristic-based algorithms for robot path planning include neural networks, fuzzy logic, nature-inspired algorithms, and hybrid algorithms. The current research in mobile robots focuses on path planning algorithms and optimization in both static and dynamic situations. Mobile robot path-planning strategies can be divided into two categories: classical methods and heuristic methods. Subcategories include (i) analytical methods, (ii) enumerative methods, (iii) evolutionary methods, and (iv) meta-heuristic methods [15]. Each of the aforementioned strategies has both advantages and disadvantages. However, the fundamental issue is that analytical approaches are too difficult for intangible applications, whereas enumerative methods are concerned with the size of the search area. However, when the search space of a path-planning technique is too big, many evolutionary strategies fail. To overcome these disadvantages, meta-heuristic methods have sparked a lot of interest in this large field of research. Path planning for mobile robots has seen the development of numerous techniques around the world. The navigation across static and dynamic situations is investigated (for single and multiple robot systems), and it is discovered that reactive approaches are more robust and perform well in every terrain than classical approaches. It has also been observed that reactive approaches are employed to increase the performance of classical approaches in the form of hybrid algorithms. As a result, reactive techniques for mobile robot path planning are becoming increasingly popular and widely employed [16]. The term "heuristic" has been used to apply to both evolutionary and artificial intelligence systems, as well as Graph Search-based planners.

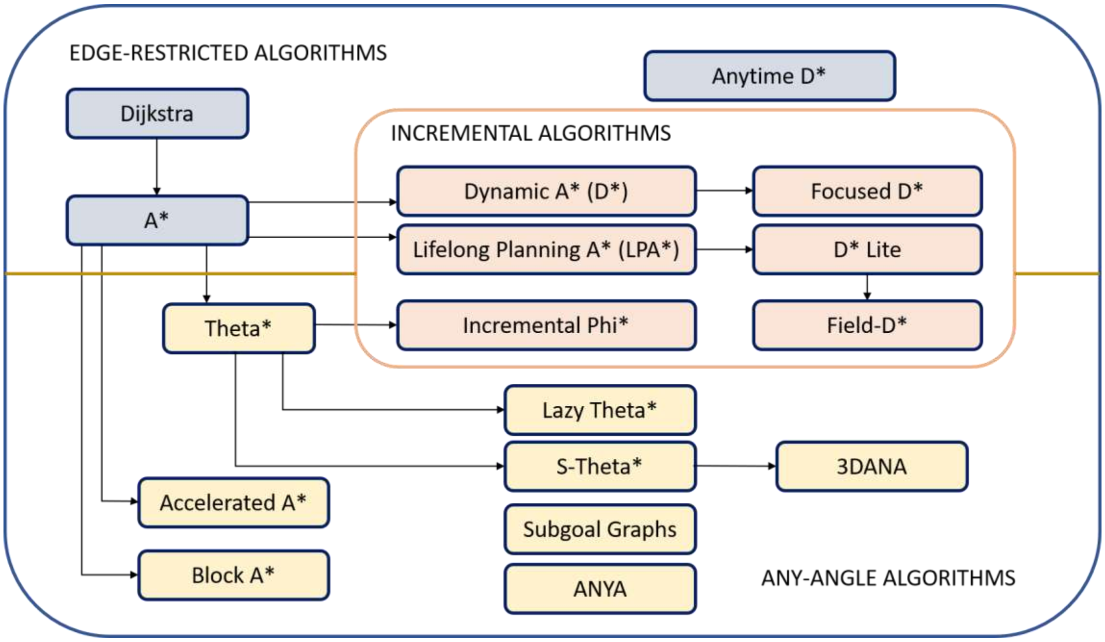



**Figure 5**: Overview of many different approaches to Graph Search path planning. The arrows indicate how most of them rest on older approaches yet introduce significant improvements [17].

## III. CELL NOMENCLATURE

Robot mobility is impossible without path planning developments. Different path-planning techniques have been put forth and shown to work on the robots and AGVs (automated guided vehicles) with numerous applications during the past few decades. A* algorithm is the most popular approach for path planning, which uses graph searching as its foundation. Convergence in the A* algorithm produces the ideal path. The speed and robustness of the planned path are two key indicators of the A* algorithm's performance. There are still several flaws, such as the close proximity of the path to obstructions and right-angle curves that slow down the speed. These elements contribute to the intended path's declining resilience. Other topics under investigation that affect the algorithm's effectiveness and speed of mobile robot are path planning speed and path smoothness. The suggested approach incorporates a bidirectional search strategy to speed up path design. This approach simultaneously searches from the start point to the target point.

Depending on the available environmental parameters, path planning algorithms are further divided into global path planning and local path planning. Global path planning, which seeks the optimum path given a significant amount of environmental data, works effectively when the environment is static and the robot fully comprehends it. As a result, static path planning is also known as global path planning. Local path planning, often referred to as dynamic path planning, is more frequently used in unforeseen or uncertain situations. In static environments such as storage and logistics, path planning requires the robot to be able to see its surroundings.

### Immediate Cells And Distances

Here it is considered that the robot may opt for four possible movements or eight possible movements from the current location on the grid as shown in figure 6. Here the distance of the corner cell is 1.41 units and for rest cell is one unit for the simplification of calculations.

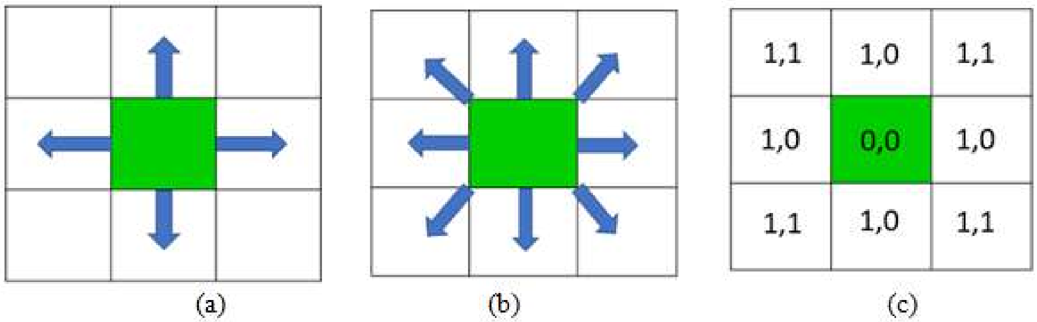



**Figure 6**: Possible movement of a robot from current location (denoted by green colour) on two-dimensional surface: (a) four possible movements and (b) eight possible movements (c) Numeric notation for possible movement from current location (0,0) for two-dimensional surface.

From the current node (0,0), the robot can transverse at any adjacent node. The current node is surrounded by total eight node thus the robot have maximum eight node to go.

In three-dimensional scenario there are twenty-six possible movement from the current node. The planes are involved in three-dimensional problems for immediate next step. The current node lie in middle plane and having eight possible movements in the same plane whereas there are nine possible movements to each lower and upper planes. In other words the current node surrounded by twenty-six nodes where it go. There are three types of steps involved in three-dimensional problems. One with one unit, second with 1.41 unit and third for 1.73 units.

**Numbering of Planner Grid**

Here we have considered a rectangular grid and each square cell is of one unit. For uniformity the cell with green colour depicts the start point and the rest colour yellow, black, and white depict goal point, obstacles, and free space respectively as shown in figure 7. The obstacles have been taken randomly in disorganised manner to make the algorithm more efficient.

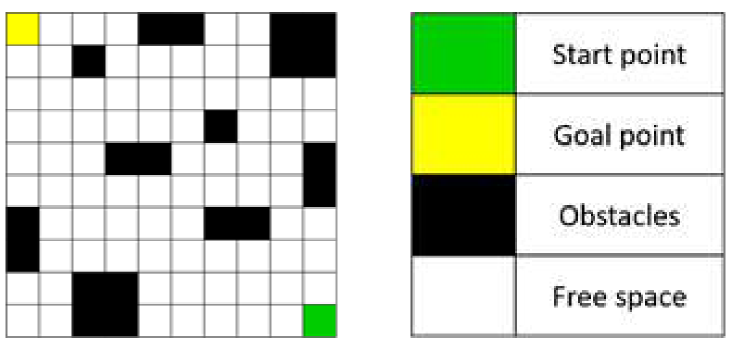



**Figure 7**: Path planning grid depicting start point, goal point, obstacles, and free space.

Numbering the cells gives the information about number of steps and distance to reach at desired cell. This is the key system to find the path. Numbering of grids is shown in figure 8. The start cell numbered as '0' as shown in figure 8 (a). The immediate surrounding cells will be numbered as '1' which are at cell positions (10,9), (9,9) and (9,10) as shown in table 1.

**Table 1:** Numbering of cells from cell number '0' to immediate forward surrounding cell positions.

| Start cell position | Start cell number | Forward surrounding cell position | Forward surrounding cell number | Distance |
|---|---|---|---|---|
| (10,10) | 0 | (10,9) | 1 | 1 |
| | | (9,9) | 1 | 1.41 |
| | | (9,10) | 1 | 1 |

As the cells numbered with '1' completed and no other possibility left for cell number '1'. Then for the next cell number '2', cell number '1' will be the start position and immediate surrounding cells will be numbered by '2'. In this case the start cell numbered as '1' as shown in figure 8 (b). The immediate surrounding cells will be numbered as '2' which are at cell positions (10,8), (9,8) (8,8), (8,9) and (8,10) as shown in table 2.

**Table 2:** Numbering of cells from cell number '1' to immediate forward surrounding cell positions.

| Start cell position | Start cell number | Forward surrounding cell position | Forward surrounding cell number | Distance |
|---|---|---|---|---|
| (10,9) | 1 | (10,8) | 2 | 1 |
| | | (9,8) | 2 | 1.41 |
| (9,9) | 1 | (10,8) | 2 | 1.41 |
| | | (9,8) | 2 | 1 |
| | | (8,8) | 2 | 1.41 |
| | | (8,9) | 2 | 1 |
| | | (8,10) | 2 | 1.41 |
| (9,10) | 1 | (8,9) | 2 | 1.41 |
| | | (8,10) | 2 | 1 |

Here cell position (10,8) is in immediate surrounding of (10,9) or (9,9). Table 2 shows, if we move from (10,9) to (10,8) then the distance of travel will be "1 unit" while if we move from (9,9) to (10,8) then the distance of travel will be "1.41 unit". Here destination is (10,8) is same but we get two possible distances of travel. Similarly (9,8) can be approached from (10,9) or (9,9) having distance of travel "1.41 unit" and "1 unit" respectively. Cell position (8,9) can be approached from (9,9) or (9,10) and cell position (8,10) can be approached from (9,9) or (9,10).

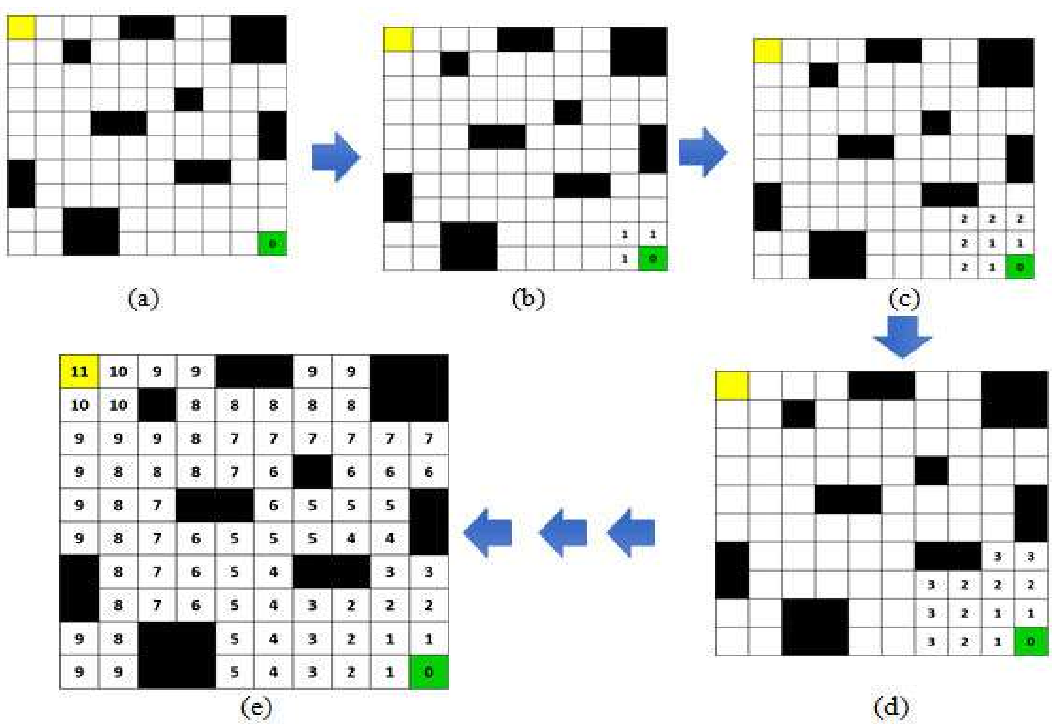



**Figure 8**: Numbering the grid with respect to number of steps required to reach from start point to each cell.

The numbers in the cells of figure 8 indicates the minimum number of steps to reach from start point to desired point. The goal point, yellow coloured cell is numbered as eleven, that means robot have to take minimum of eleven steps to reach from start point, green coloured cell. This is valid, not only for goal point but for all numbered cells. For example, the cell numbered with nine means from start position minimum 9 steps have to be taken to reach at this cell. The problem of this nomenclature is that the steps sizes are not uniform in all directions. If the robot takes one step in diagonal direction, then the step size will be of approximately 1.41 unit and if the robot takes one step in non-diagonal direction, then the step size will be of one unit.

**Modified Numbering of Cells**

To remove the problem of different step sizes in distance calculations, numbering of cells modified (figure 9). In the modification, each cell contains two numbers separated by comma. The first numbers show number multiplied by one unit distance while the second number shows number multiplied by 0.41 unit distance. Thus getting the total distance by summing up both the distances. The modified numbering system also come into picture because the numbers in figure 8 did not having any information that how it approached there diagonal and non-diagonal steps are of same length.

$$d_{(i,j)-(i+m,j+n)} = 1*((i+m)-i) + \sqrt{2}*((j+n)-j) \qquad ...(1)$$

The equation 1 consist of two distances; namely diagonal distance and non-diagonal distance. First part of the formula indicates about non-diagonal movement whereas second part of the formula indicates about diagonal movements. Numbering of grids is shown in figure 9. The start cell numbered as (0,0) as shown in figure 9 (a). The immediate surrounding cells will be numbered as (1,0) or (1,1) which are at cell positions (10,9), (9,9) and (9,10) as shown in table 3. The first number of cell number indicate about non-diagonal movement and second number indicates about diagonal movenent. For example the minimum distance of travel from cell number (0,0) to (4,3) will be calculated as:

$$d_{(0,0)-(4,3)} = 1*(4-0) + \sqrt{2}*(3-0)$$
$$= 4 + 3\sqrt{2} = 5.23 unit$$

**Table 3:** Numbering of cells from cell number '(0,0)' to immediate forward surrounding cell positions.

| Start cell position | Start cell number | Forward surrounding cell position | Forward surrounding cell number | Distance |
|---|---|---|---|---|
| (10,10) | (0,0) | (10,9) | (1,0) | 1 |
| | | (9,9) | (1,1) | 1.41 |
| | | (9,10) | (1,0) | 1 |

As the first number of cells numbered with '1' completed and no other possibility left for the first number of cell numbered with '1'. Then for the first number of next cell number will satrt from '2'. The start positions will be the closed one which first number of cell number were '1' and first number of immediate surro unding cells will be numbered by '2'. In this case the start cell numbered as '1' as shown in figure 9(b). The immediate surrounding cells will be numbered as '2' which are at cell positions (10,8), (9,8) (8,8), (8,9) and (8,10) as shown in table 4.

**Table 4**: Numbering of cells from cell number '1' to immediate forward surrounding cell positions.

| Start cell position | Start cell number | Forward surrounding cell position | Forward surrounding cell number | Distance |
|---|---|---|---|---|
| (10,9) | (1,0) | (10,8) | (2,0) | 1 |
| | | (9,8) | (2,1) | 1.41 |
| (9,9) | (1,1) | (10,8) | (2,2) | 1.41 |
| | | (9,8) | (2,1) | 1 |
| | | (8,8) | (2,2) | 1.41 |
| | | (8,9) | (2,1) | 1 |

| | | | | |
|---|---|---|---|---|
| | | (8,10) | (2,2) | 1.41 |
| (9,10) | (1,0) | (8,9) | (2,1) | 1.41 |
| | | (8,10) | (2,0) | 1 |

Here cell position (10,8) is in immediate surrounding of (10,9) or (9,9). Table 4 shows, if we move from (10,9) to (10,8) then the distance of travel will be “1 unit” while if we move from (9,9) to (10,8) then the distance of travel will be “1.41 unit”.

$$d_{(2,0)-(1,0)} = \sqrt{(2-1)^2 + (0-0)^2} = 1\,unit$$

$$d_{(2,2)-(1,1)} = \sqrt{(2-1)^2 + (2-1)^2} = \sqrt{2}\,unit$$

Here destination is (10,8) is same but we get two possible distances of travel. Previously in table 2 start cell numbers were always '1' and forward surrounding cell numbers were always '2' which didn't pass information about diagonal movement or non-diagonal movements. But here in table 4 start cell number and forward surrounding cell numbers pass information about diagonal or non-diagonal movements. For example cell position (10,9) having cell number (1,0) tells that there will be one non-diagonal movement (the first element is '1') and no diagonal movement (the second element is '0') minimally with respect to start point (0,0). And cell position (9,9) having cell number (1,1) tells that there will be one non-diagonal movement (the first element is '1') and one diagonal movement (the second element is '1') minimally with respect to start point (0,0).

Cell position (10,8) having two cell numbers as (2,0) when it is approached from cell position (10,9) and (2,2) when it is approached from cell position (9,9). We choose the best cell number as different cell numbers indicates different distances with respect to start point. In this case cell number (2,0) indicates '2 unit' of distance while (2,2) indicates '2.82 unit' of distance. We choose best cell number for optimized paths. Similarly (9,8) can be approached from (10,9) or (9,9) having distance of travel “1.41 unit” and “1 unit” respectively. Cell position (8,9) can be approached from (9,9) or (9,10) and cell position (8,10) can be approached from (9,9) or (9,10).

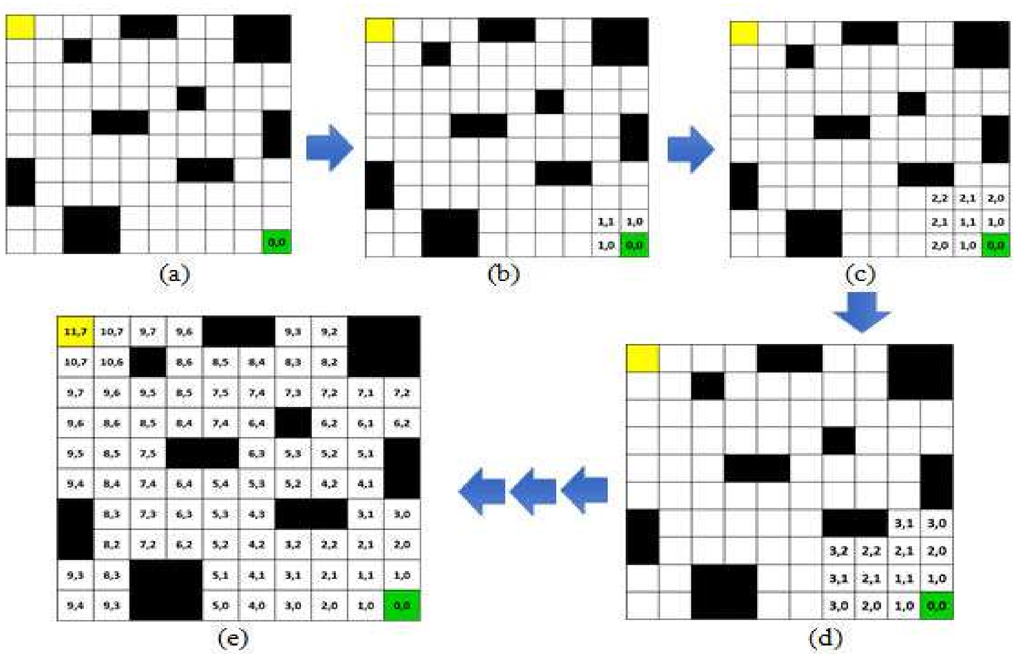



**Figure 9**: Modified numbering the grid with respect to number of steps required to reach from start point to each cell.

Here in figure 10 cell position (2,7) can be assigned from the cell positions (3,6), (3,7) and (3,8). If we move from cell position (3,6) to (2,7) then the cell position (2,7) will be numbered as (8,5) while if we move from cell position (3,7) to (2,7) or (3,8) to (2,7) then the cell position (2,7) will be numbered as (8,3). The optimum number for cell position (2,7) will be (8,3) as for this number distance from the start point will be minimum. The problem is that there would be still two start cell which approach optimum number cell. In that case the optimum path may be two but we need to check for the minimum number turns for the system.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 11,7 | 10,7 | 9,7 | 9,6 | | | 9,3 | 9,2 | | |
| 10,7 | 10,6 | | 8,6 | 8,5 | 8,4 | 8,3 | 8,2 | | |
| 9,7 | 9,6 | 9,5 | 8,5 | 7,5 | 7,4 | 7,3 | 7,2 | 7,1 | 7,2 |
| 9,6 | 8,6 | 8,5 | 8,4 | 7,4 | 6,4 | | 6,2 | 6,1 | 6,2 |
| 9,5 | 8,5 | 7,5 | | | 6,3 | 5,3 | 5,2 | 5,1 | |
| 9,4 | 8,4 | 7,4 | 6,4 | 5,4 | 5,3 | 5,2 | 4,2 | 4,1 | |
| | 8,3 | 7,3 | 6,3 | 5,3 | 4,3 | | | 3,1 | 3,0 |
| | 8,2 | 7,2 | 6,2 | 5,2 | 4,2 | 3,2 | 2,2 | 2,1 | 2,0 |
| 9,3 | 8,3 | | | 5,1 | 4,1 | 3,1 | 2,1 | 1,1 | 1,0 |
| 9,4 | 9,3 | | | 5,0 | 4,0 | 3,0 | 2,0 | 1,0 | 0,0 |

**Figure 10**: Modified numbering the grid with respect to number of steps required to reach from start point to each cell indicating optimum cell number of (8,3) for cell position (2,7).

Total number of start cells that start with the first number '7' is eleven and total number of forward surrounding cells that start with first number '8' is fourteen. While in table total thirty one entity which start with the first number '8'. That means some of these thirty one entity some are repeated where chances of multiple solution exist, some of these are not optimum and some of them are uniquely optimum. Cell position (3,10) has no any forword surrounding cell.

## IV. BIDIRECTIONAL PATH PLANNING

In this modelling two types of motion have been taken in consideration. First from initial cell to moving towards final cell and second from final cell to moving towards initial cell. The wave started from initial cell is termed up-front wave as it is approaching towards final cell and the wave started from final cell is termed down-front wave as it is approaching towards initial cell. The cell numbering start from initial first and then from final cell until both the wavefronts intersects.

Numbering of cell positions surrounding the current start position is forward numbering if it originates from start position and move towards goal position.

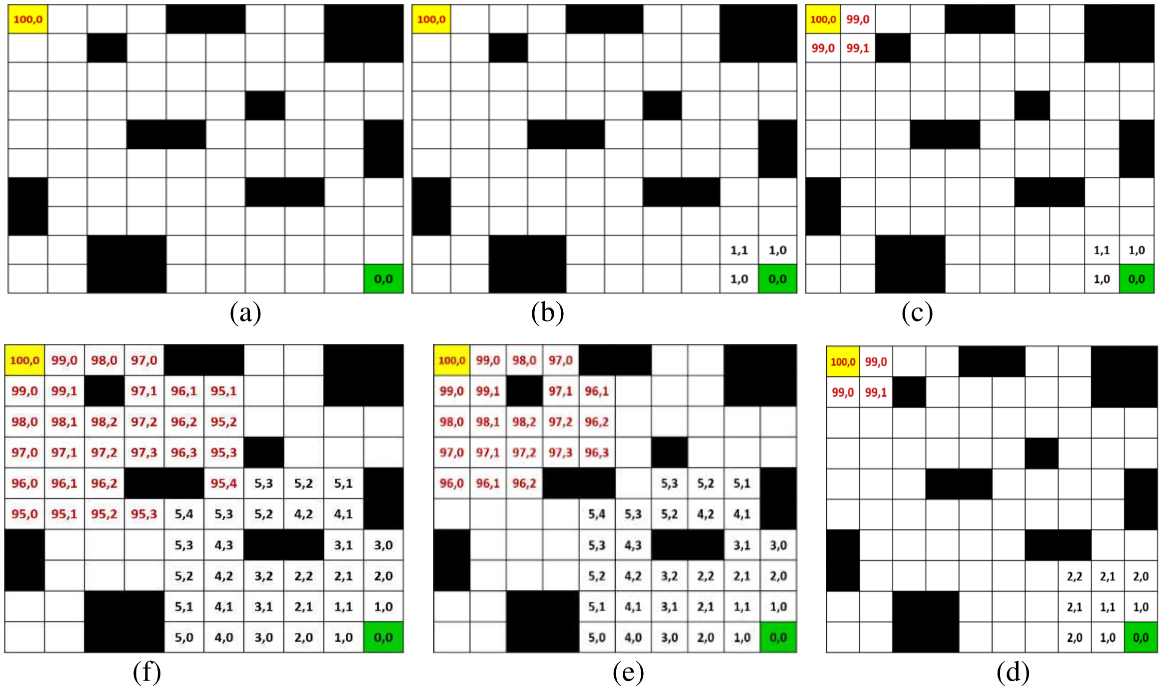


**Figure 11**: Numbering the cells from start first and then from last simultaneously until it intersects.

As shown in figure 11; initial cell is termed as (0,0) while for the final position it is termed as (100,0). In the first step as it starts from start cell first, so the numbering those cells which surrounds (0,0) cells is shown in table 5.

**Table 5**: Forward surrounding cell numbering from cell position (10,10).

| Start cell position | Start cell number | Forward surrounding cell position | Forward surrounding cell number | Best cell number w.r.t to cell number (0,0) |
|---|---|---|---|---|
| (10,10) | (0,0) | (10,9)-1 | (1,0) | (1,0) |
| | | (9,9)-1 | (1,1) | (1,1) |
| | | (9,10)-1 | (1,0) | (1,0) |

As soon as immediate forward cells of cell number (0,0) covered and final cell not met then numbering will start to surrounds the cell number (100,0), shown in table 6.

Numbering of cell positions surrounding the current start position is backward numbering if it originates from goal position and move towards start position.

**Table 6**: Backward surrounding cell numbering from cell position (1,1).

| Start cell position | Start cell number | Backward surrounding cell position | Forward surrounding cell number | Best cell number w.r.t to cell number (0,0) |
|---|---|---|---|---|
| (1,1) | (100,0) | (2,1)-1 | (99,0) | (99,0) |
| | | (2,2)-1 | (99,1) | (99,1) |
| | | (1,2)-1 | (99,0) | (99,0) |

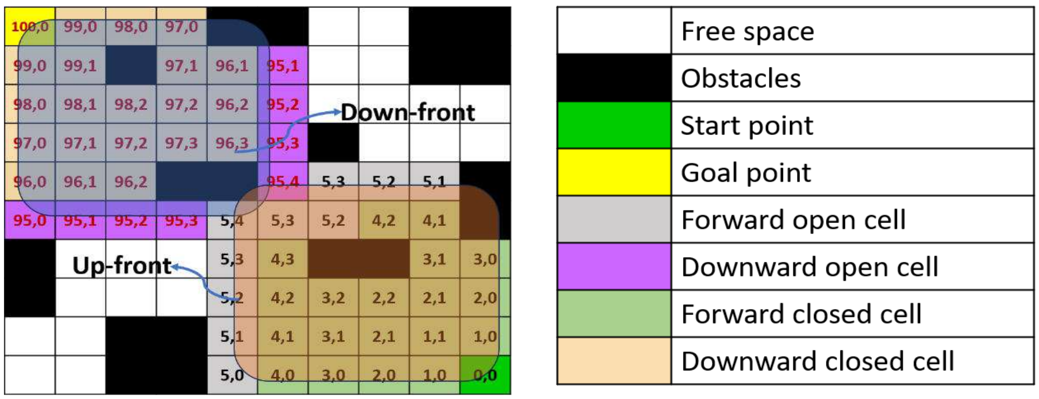
As soon as immediate backward cells of cell number (100,0) covered and final cell not met with the surrounding cells of cell number (0,0), then numbering will start to surrounds the cells which cell number start with '1', shown in table 6.

ISSN: 2633-4828 Vol. 4 No.1, June, 2022

International Journal of Applied Engineering & Technology

After numbering the grid the cell may be divided into different group cells. There are still many cells which were not numbered and it is good sign to have the calculations in the lesser time for the algorithm, those cell are free space represented as white cell in figure 12. Obstacles are represented by black colour and rest cells are numbered which are further divided into open cell and closed cell. Closed cells are those cells which are surrounded by upcoming numbers, those cells have not intersected with forward numbering and backward numbering. Open cells are those cells which are not surrounded by upcoming numbers because of occurance of intersection with forward numbering and backward numbering.

Up-front wave constitutes of open cells and closed cells of upward numbers. The closed cells of up-front wave will not directly interact with cells of down-front wave where as the open cells of up-front wave have the opportunity to directly interact with down-front wave.



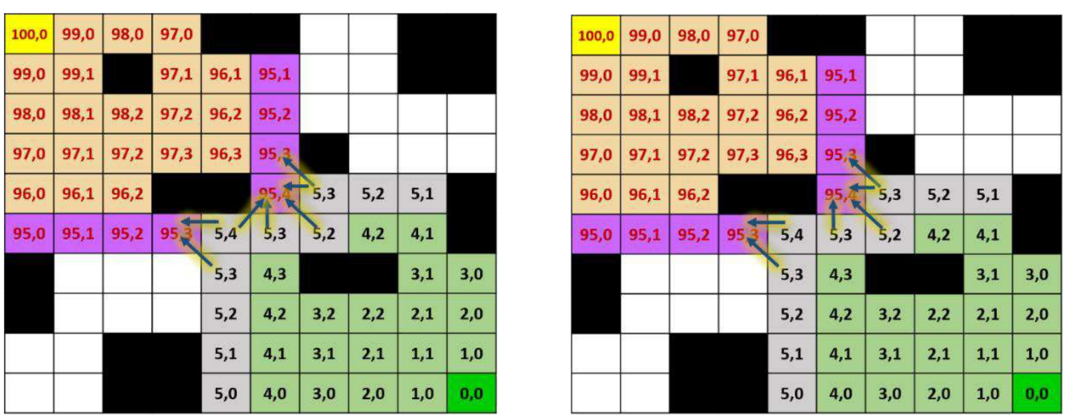

(a) (b)

**Figure 12**: (a) Up-front and down-front wave, (b) Corresponding colours

Down-front wave constitutes of open cells and closed cells of downward numbers. The closed cells of down-front wave will not directly interact with cells of up-front wave where as the open cells of down-front wave have the opportunity to directly interact with down-front wave.

## V. EXPERIMENTAL RESULTS

In the experiments, the static obstacle problem was solved on 10*10 grids. Multiple paths were investigated and the ones with the fewest turnings were chosen to get the shortest distance travelled from the start position to the goal position. The optimal path from the start (10, 10) to the goal (1, 1) points.



(a) (b)

**Figure 13**: (a) Possible paths at intersection of forward and backward wavefronts (b) Minimum distance covered intersections.

Open cells of up-front wave may interact with open cells of down-front wave and vis-a-versa. Some of these cell which interacts open the opportunity to find the optimum path. Table 7 listed the open cells of up-front wave and down-front wave of figure 13.

**Table 7**: Open cells of Up-front wave and down-front wave from figure 13.

| Up-front open cell position | Intersecting / Non-Intersecting | Down-front open cell position | Intersecting / Non-Intersecting |
|---|---|---|---|
| (10,5) | Non-Intersecting | (6,1) | Non-Intersecting |
| (9,5) | Non-Intersecting | (6,2) | Non-Intersecting |
| (8,5) | Non-Intersecting | (6,3) | Non-Intersecting |
| (7,5) | Intersecting | (6,4) | Intersecting |
| (6,5) | Intersecting | (5,6) | Intersecting |
| (6,6) | Intersecting | (4,6) | Intersecting |
| (6,7) | Intersecting | (3,6) | Non-Intersecting |
| (5,7) | Intersecting | (2,6) | Non-Intersecting |
| (5,8) | Non-Intersecting | | |
| (5,9) | Non-Intersecting | | |

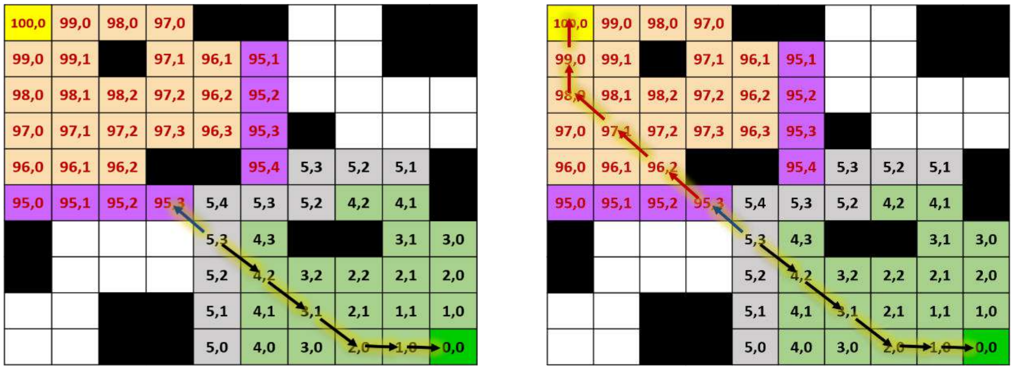
Out of those in interactive paired cells of up-front wave and down-front wave have the opportunity of best path. Table 7, shows the combination of intersecting cells from open cells of up-front wave and down-front wave.

The intersection may occure diagonally or non-diagonally. This is very important to know because of two reasons. The first reason lie on the distance between the interactive cells. If intersection occure diagonally then the distace covered will be of '1.41 unit' which means (1,1) with respect to numbers, i.e. first number '1' indicates one unit of distance in addition with mulpilication of second number '1' with the multiplyier (1.41-1) which results into '1.41 unit' of distance. If intersection occure non-diagonally then the distace covered will be of '1 unit' which means (1,0) with respect to numbers, i.e. first number '1' indicates one unit of distance in addition with mulpilication of second number '0' with the multiplyier (1.41-1) which results into '1 unit' of distance.

**Table 8**: Combination of intersecting cells from open cells of up-front wave and down-front wave.

| Up-front open cell position | Up-front open cell number | Down-front open cell position | Down-front open cell number | Diagonal / Non-diagonal matching | Non-diagonal distance | Diagonal distance | Distance | Minmun / Not minimum |
|---|---|---|---|---|---|---|---|---|
| (7,5) | (5,3) | (6,4) | (95,3) | D | 10 | 6 | 11+7*0.41 | Minmun |
| (6,5) | (5,4) | (6,4) | (95,3) | ND | 10 | 7 | 11+7*.41 | Minmun |
| (6,5) | (5,4) | (5,6) | (95,4) | D | 10 | 8 | 11+9*0.41 | Not-minmun |
| (6,6) | (5,3) | (5,6) | (95,4) | ND | 10 | 7 | 11+70.41 | Minmun |
| (6,7) | (5,2) | (5,6) | (95,4) | D | 10 | 6 | 11+7*0.41 | Minmun |
| (5,7) | (5,3) | (5,6) | (95,4) | ND | 10 | 7 | 11+7*0.41 | Minmun |
| (5,7) | (5,3) | (4,6) | (95,3) | D | 10 | 6 | 11+70.41 | Minmun |

The second reason to know intersection is that if it intersects diagonally then the path selection in closed cells of up-front and down-front will be first seacrched in the same direction of diagonal. And if it intersects non-diagonally then the path selection in closed cells of up-front and down-front will be first seacrched in the same direction of non-diagonal. There are toal four diagonal and four non-diagonal direction of approaching the end cells or goal cells.




(a) (b)

**Figure 14**: Searching the path from intersecting cell positions (7,5) to (6,4) in (a) up-front wave and (b) down-front wave.

The first step in the path making is to find the intersecting cells for the minimum distance of travel. The minimum distance will be meet if we choose the perfect cell number ahead in up-front wave front and down-front wave front. If we not choose the appropriate cell number then the distance will be increased. In selection of cells from intersecting cell to up-front cells to make the path complete. The first priority is to make the path of minimum distance and the second priority is to choose the cell which is in direction of intersecting cells.

For finding the cells in up-front wave, let take a cell from the intersecting cells which is part of open up-front wave. Suppose the cell number is (x, y), then to choose next cell the first number of the cell number must be 'x-1' and the second number is diversified with options. The second number should be searched first in the direction of intersecting cells, if it not met in any surrounding cell then it will go for the other directions. The first number of cell number will stop the iteration if this not find next reduced first number by one unit.

**Table 9**: For up-front when wave fronts intersect through cell position (7,5) and (6,4)

| Up-front cell position | Up-front cell number | Possible movement cell position | Possible movement cell number | Diagonal / Non-diagonal |
|---|---|---|---|---|
| (7,5) | (5,3) | (7,6) | (4,3) | |
| | | (8,6) | (4,2) | Diagonal |
| (8,6) | (4,2) | (8,7) | (3,2) | |
| | | (9,7) | (3,1) | Diagonal |
| (9,7) | (3,1) | (8,8) | (2,2) | |
| | | (9,8) | (2,1) | |
| | | (10,8) | (2,0) | Diagonal |
| (10,8) | (2,0) | (9,9) | (1,1) | |
| | | (10,9) | (1,0) | Non-diagonal |
| (10,9) | (1,0) | (10,10) | (0,0) | Non-diagonal |

Open up-front cell position (7, 5) having cell number (5, 3) in figure 14. There are two possibilities from which cell position (7, 5) can be reached from the first number of cell number consideration. The first possibility is from (7, 6) having cell number (4, 3) and from (8, 6) having cell number (4, 2). Out of these two options both making same optimum distance, so the second possibility with cell position (8, 6) having cell number (4, 2) will be chosen as it makes the diagonal intersection in direction of open intersecting cells. Once we select the cell position (8, 6) having cell number (4, 2), the next cell should be searched with recent direction formed. In this fashion we move from cell position (8, 6) to cell position (9, 7) diagonally then move from cell position (9, 7) to cell position (10, 8) diagonally.

Now, there are two possibilities from which cell position (10, 8) can be reached from the first number of cell number consideration. The first possibility is from (9, 9) having cell number (1, 1) and from (10, 9) having cell number (1, 0). Out of these two options only second possibility making same optimum distance, so the second possibility with cell position (10, 9) having cell number (1, 0) will be chosen but it makes the non-diagonal intersection which is not in direction of previous intersecting cells because of first number of cell number (2, 0) consideration. Then from cell position (10, 9) to cell position (10, 10) is only option to move in non-diagonal direction. The iteration stops here for the up-front wave as there is no first number of '-1' of cell number to continue the next iteration.

For finding the cells in down-front wave, let take a cell from the intersecting cells which is part of open down-front wave. Suppose the cell number is (u, v), then to choose next cell the first number of the cell number must be 'u+1' and the second number is diversified with options. The second number should be searched first in the direction of intersecting cells, if it not met in any surrounding cell then it will go for the other directions. The first number of cell number will stop the iteration if this not find next enhanced first number by one unit.

**Table 10**: For down-front when wave fronts intersect through cell position (7,5) and (6,4)

| Down-front cell position | Down-front cell number | Possible movement cell position | Possible movement cell number | Diagonal / non-diagonal |
|---|---|---|---|---|
| (6,4) | (95,3) | (5,3) | (96,2) | Diagonal |
| (5,3) | (96,2) | (4,2) | (97,1) | Diagonal |
| | | (4,3) | (97,2) | |
| | | (4,4) | (97,3) | |
| (4,2) | (97,1) | (3,1) | (98,0) | Diagonal |
| | | (3,2) | (98,1) | |
| | | (3,3) | (98,2) | |
| (3,1) | (98,0) | (2,1) | (99,0) | Non-diagonal |
| | | (2,2) | (99,1) | |
| (2,1) | (99,0) | (1,1) | (100,0) | Non-diagonal |

Open down-front cell position (6, 4) having cell number (95, 3) in figure 14. There is only one possibility to move to cell number (5, 3) having cell number (96, 2) diagonally. From cell position (5,3) to next cell there are three possibilities from which cell position (5, 3) can be reached from the first number of cell number consideration. The first possibility is from cell position (4, 2) having cell number (97, 1), from cell position (4, 3) having cell number (97, 2) and from cell position (4, 4) having cell number (97, 3). Out of these three options all making same optimum distance, so the first possibility with cell position (4, 2) having cell number (97, 1) will be chosen as it makes the diagonal intersection in direction of previous intersecting cells. Once we select the cell position (4, 0) having cell number (97, 1), the next cell should be searched with recent direction formed. In this fashion we move from cell position (4, 2) to cell position (3, 1).

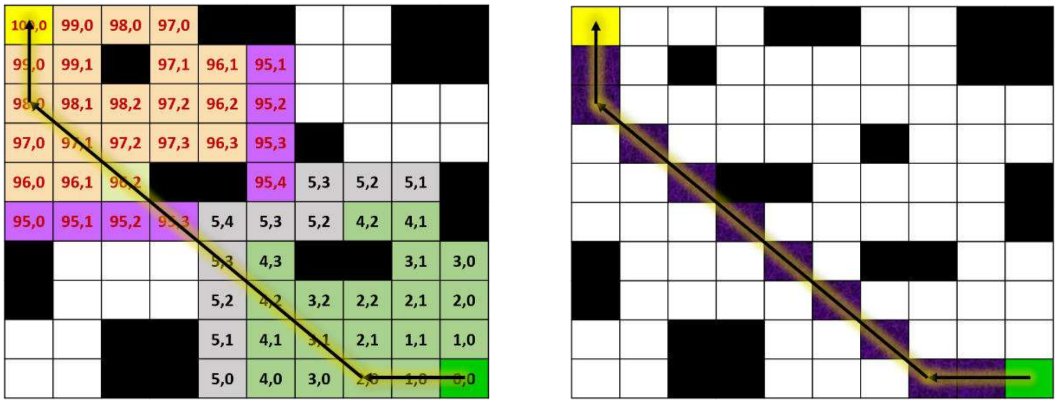



**Figure 15**: Complete path from open intersection of cell positions (7, 5) to (6, 4)

Now, there are two possibilities to move from cell position (3, 1), on the basis of first number of cell number consideration. The first possibility is (2, 1) having cell number (99, 0) and from cell position (2, 2) having cell number (99, 1). Out of these two options only first possibility making same optimum distance, so the first possibility with cell position (2, 1) having cell number (99, 0) will be chosen but it makes the non-diagonal intersection which is not in direction of previous intersecting cells because of first number of cell number (98, 0) consideration. Then from cell position (2, 1) to cell position (1, 1) is only option to move in non-diagonal direction. The iteration stops here for the down-front wave as there is no first number of '101' of cell number to continue the next iteration. Similrly, there are total six way of minimum distance in this manner but the number of turns is varying in some cases. The best path is that which have minimum distance with minimum number of turns.

## VI. CONCLUSION AND FUTURE SCOPE

This work created a simulation for a robot navigation system that relies on segmenting the image to speed up processing and has obstacles to avoid. In order to discover the best path based on the map of the environment, MATLAB software was used to construct the bidirectional search method and divide the bigger space into smaller portions. The field of mobile robot path planning has two branches: online and offline. As a result, many algorithms and methods are still being developed to offer the best answers for all the problems they encounter. The range of work can be further expanded by adding more service robots in the future. Additionally, by processing videos, the similar method can be developed for navigation.

The up-front and down-front are generated simultaneously, so the pre-installed webcams are needed. This can be seen as a limitation or an advantage. Limitation, if the system does not have webcams for the surveillance system, then it should be installed or an advantage if the system has pre-installed webcams for the surveillance, then no extra webcams are needed. The path with the fewest turnings helps the robot to maintain its trip continuity. Robots should use the path that has the fewest turns. With less turning, the robot may lower base motor activity, saving time and energy. In path planning issues, the algorithm returns a large number of pathways that travel the shortest distance. This thesis chooses the best ideal path among those with the same minimum distance between the start and goal points. The essential criterion for the solution is the least number of bots turns. This contributes significantly to the advancement of the bidirectional search algorithm in complicated maps for mobile robots, particularly in terms of shortest path length and overall least number of turns.

In future research, the proposed approach could be combined with various path-planning algorithms in a dynamic context. Service robots, rescue robots, and industrial robots are just some of the applications that can be introduced.

## REFERENCES

[1] Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I. and Dizdarevic, S., 1999. Genetic algorithms for the travelling salesman problem: A review of representations and operators. Artificial intelligence review, 13, pp.129-170.

[2] Wassan, N.A. and Nagy, G., 2014. Vehicle routing problem with deliveries and pickups: modelling issues and meta-heuristics solution approaches. International Journal of Transportation, 2(1), pp.95-110.

[3] Dibbelt, J., Konstantopoulos, C., Wagner, D., Gavalas, D., Kontogiannis, S., Zaroliagis, C., Kasapakis, V. and Pantziou, G., 2017, July. Multimodal route and tour planning in urban environments. In 2017 IEEE Symposium on Computers and Communications (ISCC) (pp. 214-219). IEEE.

[4] Sharma, K. and Doriya, R., 2020. Path planning for robots: An elucidating draft. International journal of intelligent robotics and applications, 4, pp.294-307.

[5] Wu, Q., Liu, Y. and Wu, C., 2018. An overview of current situations of robot industry development. In ITM Web of Conferences (Vol. 17, p. 03019). EDP Sciences.

[6] Rao, R.V. and Padmanabhan, K.K., 2006. Selection, identification and comparison of industrial robots using digraph and matrix methods. Robotics and Computer-Integrated Manufacturing, 22(4), pp.373-383.

[7] Alenezi, M.R. and Almeshal, A.M., 2018. Optimal Path Planning for a Remote Sensing Unmanned Ground Vehicle in a Hazardous Indoor Environment. Intelligent Control and Automation, 9(04), p.147.

[8] Jiang, J., Zeng, X., Guzzetti, D. and You, Y., 2020. Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures. Acta Astronautica, 171, pp.265-279.

[9] Raja, P. and Pugazhenthi, S., 2012. Optimal path planning of mobile robots: A review. International journal of physical sciences, 7(9), pp.1314-1320.

[10] Zhang, H.Y., Lin, W.M. and Chen, A.X., 2018. Path planning for the mobile robot: A review. Symmetry, 10(10), p.450.

[11] Zhang, F., Li, N., Xue, T., Zhu, Y., Yuan, R. and Fu, Y., 2019, December. An improved dynamic window approach integrated global path planning. In 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 2873-2878). IEEE.

[12] Ji, J., Khajepour, A., Melek, W.W. and Huang, Y., 2016. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. IEEE Transactions on Vehicular Technology, 66(2), pp.952-964.

[13] Souissi, O., Benatitallah, R., Duvivier, D., Artiba, A., Belanger, N. and Feyzeau, P., 2013, October. Path planning: A 2013 survey. In Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM) (pp. 1-8). IEEE.

[14] Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R., 2016. Heuristic approaches in robot path planning: A survey. Robotics and Autonomous Systems, 86, pp.13-28.

[15] Zafar, M.N. and Mohanta, J.C., 2018. Methodology for path planning and optimization of mobile robots: A review. Procedia computer science, 133, pp.141-152.

[16] Patle, B.K., Pandey, A., Parhi, D.R.K. and Jagadeesh, A.J.D.T., 2019. A review: On path planning strategies for navigation of mobile robot. Defence Technology, 15(4), pp.582-606.

[17] Victerpaul, P., Saravanan, D., Janakiraman, S. and Pradeep, J., 2017. Path planning of autonomous mobile robots: A survey and comparison. Journal of Advanced Research in Dynamical and Control Systems, 9(12), pp.1535-1565.