OBSTACLE AVOIDANCE AND PATH PLANNING IN MOBILE ROBOTICS

A MAJOR PROJECT- II REPORT
SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
INFORMATION SYSTEMS

Submitted by

VIKAS P DAMLE **2K20/ISY/24**

Under the supervision of

DR. SEBA SUSAN
Professor

DEPARTMENT OF
INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi - 110042

MAY - 2022

DELHI TECHNOLOGICAL UNIVERSITY

STUDENT DECLARATION

I hereby declare that the work presented in this report entitled "OBSTACLE AVOIDANCE AND PATH PLANNING IN MOBILE ROBOTICS", in fulfillment of the requirement for the award of the MASTER OF TECHNOLOGY degree in Information Systems submitted in Information Technology Department at DELHI TECHNOLOGICAL UNIVERSITY, New Delhi, is an authentic record of my own work carried out during my degree under the guidance of Dr. Seba Susan.

Date: 26 - May - 2022

Place: New Delhi

VIKAS P DAMLE

2K20/ISY/24

DELHI TECHNOLOGICAL UNIVERSITY

CERTIFICATE

This is to certify that VIKAS P DAMLE (2K20/ISY/24) have completed the seminar titled “OBSTACLE AVOIDANCE AND PATH PLANNING IN MOBILE ROBOTICS” under my supervision in partial fulfillment of the MASTER OF TECHNOLOGY degree in Information Systems at DELHI TECHNOLOGICAL UNIVERSITY.

Dr. SEBA SUSAN
(Supervisor)

ABSTRACT

Robotics has been one of the emerging fields in the current date scenario. Robotics are being employed in various application sectors like the medical, industries, tourism, agriculture, etc. Among them, autonomous driving systems have been one of the major problems where active research is been taking place. Among many, two of the major sub-sections under them will be obstacle avoidance and path planning. Obstacle avoidance usually works in the higher level of path planning. Mobile robots should find an optimal or near-optimal path from the starting state to the target state based on one or some performance indicators. In path planning, we will try to build an algorithm to dynamically deal with the cost constraint.

ACKNOWLEDGEMENT

I am very thankful to Dr. Seba Susan (Professor, Department of Information Technology) and all the faculty members of the Department of Information Technology of DTU. They all provided me with immense support and guidance for the seminar.

I would also like to express my gratitude to the university for providing me with the laboratories, infrastructure, testing facilities and environment which allowed me to work without any obstructions.

I would also like to appreciate the support provided to me by my lab assistants, seniors and my peer group who aided me with all the knowledge they had regarding various topics.

VIKAS P DAMLE

2K20/ISY/24

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 MOTIVATION

In earlier days, use of mobile robot was restricted to the industries. However, it is now widely employed in entertainment, medical, mining, rescue operations, aerospace, education,defense, agriculture, and many other. The robot is fitted with several intelligent pieces of equipment necessary to trace the environment and localize, control the navigation, identify impediments, and avoid obstacles while completing the task of navigation. The most fundamental role of any navigation technique is to discover a safe path from the starting point to the required end point by recognizing and avoiding obstacles. As a result, when working in an environment, choosing the proper navigational strategy is an important phase in the course planning.

Path planning has been mainly identified into local and global path planners. In local path planner, the device has limited understanding of the navigational environment (either partially known or unknown). In global path planner, the robots have comprehensive data about the surrounding, allowing it to approach the objective by following a predetermined path. However, due to their inability to adapt in unknown environments, global path planning methods have limited uses, whereas local path planner can adapt to known/unknown situations to provide an optimal path. In this, we are studying path planners where they have prior knowledge of the environment and they have been divided into the classical and reactive approaches.

CHAPTER 2
PATH PLANNING

2.1 WHAT IS PATH PLANNING

It is the process of finding an efficient path from the source to destination. It plays a prime role in the safe navigation of the robot in an environment. The path must meet certain robot requirements based on its model and obstacle placements. It should also provide better results for functions like time and distance. Dynamic and sampling-based approaches are the commonly used algorithms.

It can be done based upon different scenarios like industrial equipment, autonomous vehicles, and other robotic applications. Usually, the environments are static. But we can add some extra features where it can adapt to a dynamic environment, for things like obstacle avoidance and many more.

2.2 TYPES OF PATH PLANNING

They have been divided into several categories based upon the geometric path built [1]

- Roadmap method [2]
- Cell decomposition method [3]
- Artificial potential field method [4]

2.3 GLOBAL PATH PLANNER

A path planner which already has the information of an environment using any of the SLAM [5] algorithms are called a Global Path Planner [6]. In SLAM, the surrounding is being mapped while parallelly the robot is localized. This helps the robot to know the uncertain environment it is present in and have an idea of the obstacles beforehand. This way the robot can travel from start to endpoint in an optimal path while avoiding the obstacles. This path is depends on the algorithm chosen by the user based upon the type of environment and other constraints.

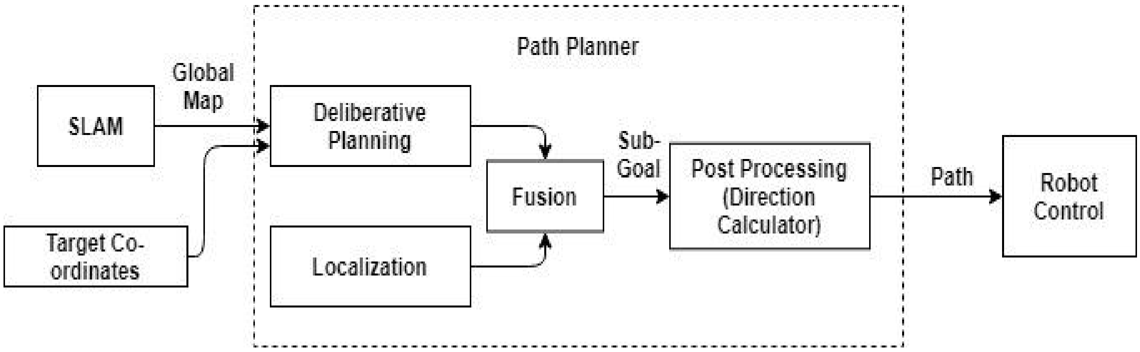


Fig 2.1 Process followed in global path planning

The Global Path Planners can be categorized into

- Graph Search Algorithm
- Random Sampling Algorithm
- Intelligent Bionic Algorithm

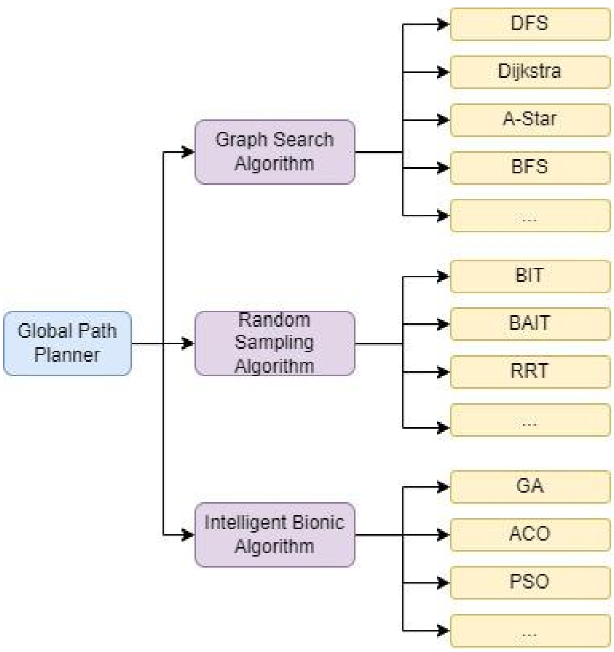


Fig 2.2 Different categories of global path planning algorithms

- Graph search algorithm

In graph search algorithms [7], a graph is used for general path discovery. It will try to visit as many nodes of the graph as possible. But, it cant be assured that the solution obtained will be optimal. They are usually used as a base for other algorithms.

- Random sampling algorithm

A random sampling algorithm [8] finds the paths by sampling random points in the given environment. The exploration space and the bias are usually increased with the help of heuristics. This results in faster results, but not efficiency.

- Intelligent bionic algorithm

The intelligent Bionic Algorithm is a search-based algorithm built using the concept of genetics and natural selection for optimization. They are used to search an efficient solution for complicated problems. This has algorithms like GA, PSO, ASO, etc.

2.4 OPTIMIZATION TECHNIQUES

When a path planner creates a path it should meet specific optimization factors like the shortest route, efficient energy consumption, smoothness of path, least execution time, and maximum safety. For a problem to be classified as NP-hard, it must satisfy at least two functions. All the basic multi-objective functions are listed below.

- Cost of the path
- Computation time
- Smoothness in the path
- Amount of energy consumed

2.4.1 Artificial Intelligence (Heuristic Approach)

- Genetic Algorithm (GA)

GA is a popular search-based algorithm based on genetics and natural selection used for optimization. It was initially designed by H. J. Bremermann [9]. Holland [10] introduced it to computer science. Its applications range in many fields of engineering. For optimizing during a complex problem in which the objective function value must be maximized or minimized. The population must be allocated a fitness value based on the objective function for the given issue in this strategy. GA works well in an unknown environment as it requires very little knowledge about the environment. Xiao et al. [11] used this strategy to tackle the main functions of path planning like length, smoothness, and obstacle avoidance.

- Fuzzy Logic (FL)

Lotfi Zadeh proposed the fuzzy logic system for the first time in 1965. They provide a formal idea for communicating and execution of the heuristic information obtained from observation-based behaviors of hominid experts. Fuzzy logic is based on the premise that humans don't think in terms of numbers, but rather in terms of ideas. There are several vulnerability aspects in depicting the environment in self-ruling mobile robot route planning where the scenario cannot be straightforwardly grouped into a single situation. During these cases, a fuzzy logic notion is welcomed. Fuzzy logic has the advantages of no new changes between states and robustness during long execution as they are directly derived from hominid experience eventually solving the issue of local minimization. Zavlangas et al. [12] propose a fuzzy (Sugeno) based omnidirectional mobile robot navigation. For effective navigation, Castellano et al. [13] devised a fuzzy rule generation approach that is autonomous for obstacle avoidance.

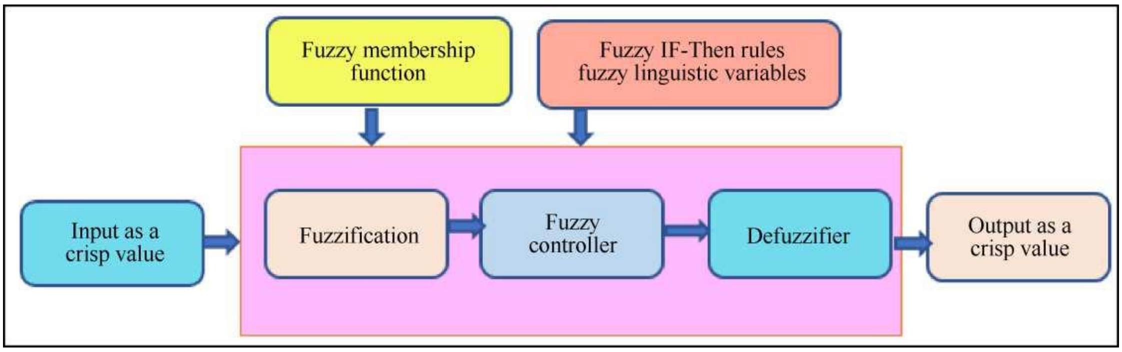


Fig 2.3 Basic FL Controller

- Neural Network (NN)

A data processing concept known as a neural network is based on how biological sensory systems, like the brain. The brain functions are complex, non-linear and follow the concept of parallel PC. The novel structure of the processed data handling framework is the driving force behind this concept. For dealing with specific challenges, a large number of interconnected handling components (neurons) are generated. The robot's operating environment is confined to a 2-D plane. They are completely opaque to the nature of snags and are not curved to the status of impediments. Using a neurological system about a nearby connection to describe the arrangement space of the mechanism's work, each neuron and its local has a comparable association in the neural system. The neurological system is designed in a fairly parallel manner, with two-way data exchange between neurons. Each node features an activation function. A system of weighted connections then transports these patterns to buried levels for processing. To create the desired output, these hidden layers communicate with the output nodes. The qualities of NN that make it beneficial in mobile robot navigation are generalisation, enormous parallelism, distributed representation, learning ability, and fault tolerance. D. Janglova [14] showed the use of neural networks to guide a mobile robot across an unfamiliar area. For the building of a collision-free route, he employed two NN-based techniques. The first neural mechanism uses sensory input to discover clear space, while another NN avoids the nearest impediment to find a safe path.

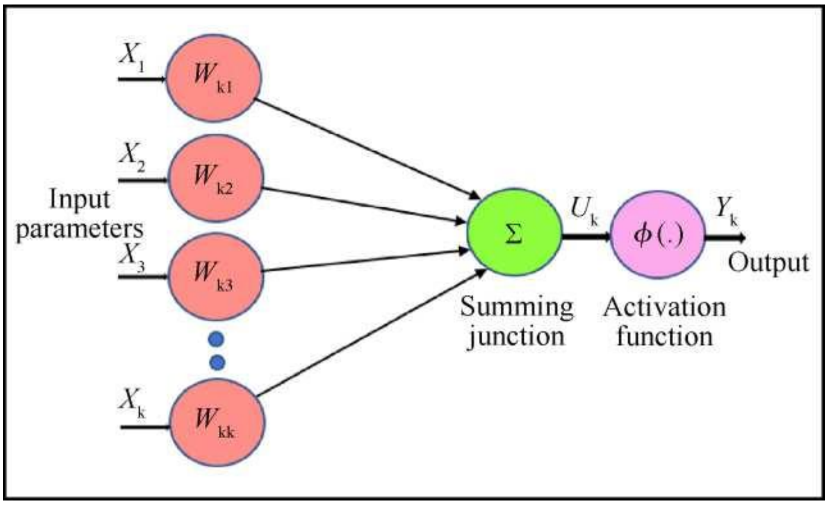


Fig 2.4 Architecture of NN

- Firefly Algorithm (FA)

X. S. Yang [15] created this in 2008 and called it after the firefly's light flashing behavior. Random states and generic identification are used. This meta-heuristics technique is modeled after the trial and error process performed by fireflies in nature. Fireflies belonging to the Lampyridae family utilize its light to find a mate, communicate, and occasionally drive away predators. They have recently been employed for optimization most sectors of engineering, which also includes robot navigation. A. Hidalgo-Paniagua, A. Miguel, Vega–Rodriguez, J. Ferruz and N. Pavon [16] developed an FA-based mobile robot navigating strategy where they achieved the three primary navigational goals of length, smoothness, and safety.

- Particle Swarm Optimization (PSO)

PSO a metaheuristic algorithm is known to be an nature based as they mimic the behavior of animals like fish or birds. In 1995 Eberhart and Kennedy [17] found an tool to optimize scientific problems that imitate the social animal behavior not requiring a group leader to attain the goal. The birds achieve their goal of finding food by successfully communicating and later following one of the members who are closest to the meal rather than the leader. In PSO every particle present a possible solution. They are often utilized in robot navigation. X. Tang, L. Li and B. Jiang [18] solved the challenges of robot navigation in an unfamiliar environment using a multi-agent particle filter. It helps to reduce calculations and the perseverance of convergence characteristics.

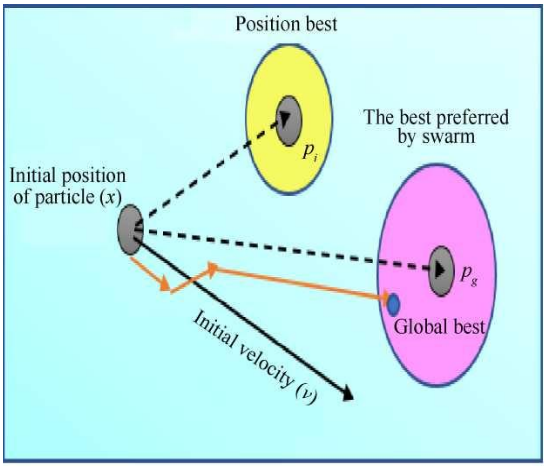


Fig 2.5 Basics of PSO

➢ Ant Colony Optimization (ACO)

Marco Dorigo presented the Ant system for tackling computational issues. The ant colony optimization which is probabilistic and condensed in nature is used to find an efficient path in graphs. ACO is derived from the behaviour of an ant colony. The related connection among the ants, which is at the heart of their behavior, allows them to find a shortest path between their house and food sources. The optimal path is determined by measuring the number of pheromones which is an chemical essence deposited on its courses by ants. These features/characteristics are obtained using the ACO approach to solve a discrete optimization issue. As a result, the ACO is better suited to challenges when the source and purpose are well defined and specific.The following is the main ACO method:

- Produce the ant(s) and loop for each ant until the work is accomplished.
- Pheromone is stored on visited states/sites that are ant-infested (s).
- Pheromone dissipation and daemon exercises

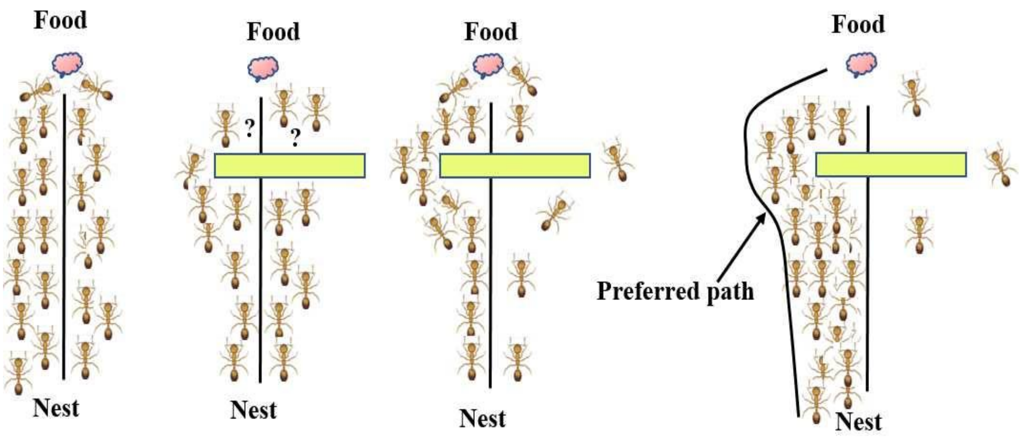


Fig 2.6 How the ant behaves while searching for food

T. Guan-Zheng, HE Huan and S. Aaron [19] described the usage of ACO for path planning in real-time. This method had improved performance over previous algorithms.

CHAPTER 3
PREVIOUS WORK

3.1 GAZEBO SIMULATOR

The Gazebo [24] is an open source simulator where the package consists of in-house physics engine ODE, OpenGL visuals, sensor simulation, and actuator control support code. ODE, Bullet, and other high-performance physics engines are all supported by Gazebo (the default is ODE). It creates realistic environments with high-quality details. It houses sensors such as lasers, cameras, kinect sensors, and others which helps in seeing the environment. Many challenges have employed gazebos such as

- DARPA Robotics Challenge[25]
- NASA Space Robotics Challenge[26]
- Toyota Prius Challenge[27]
- Agile Robotics for Industrial Automation Competition[28]
- DARPA Subterranean Challenge[29]

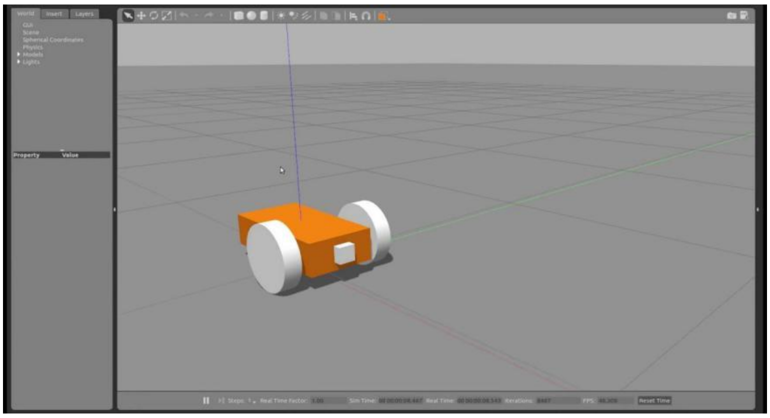


Fig 3.1 Simple robot in a Gazebo environment

3.2 EVALUATION OF PERFORMANCE OF ALGORITHMS

In a previous study, the algorithms were built using python. First, the environment will be built using python functions. Later the starting, destination, and obstacles were defined. The algorithm has to find the shortest path from the required starting point to the end point. This final optimal path was plotted. The algorithms were executed for 10 iterations and the following results are the average of those. The algorithms that were taken into consideration were BFS, DFS, Dijkstra's, and A-Star

➢ BFS

The Breadth First Search (BFS) [20] algorithm to search path or data makes use of graph and tree data structure. It explores all the node at the existing level before exploring the nodes present in next level.

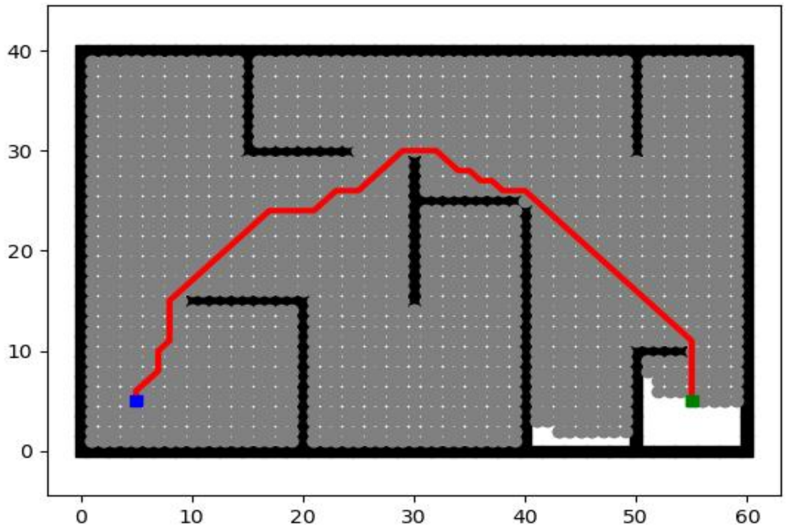


Fig 3.2 Path traced using BFS algorithm

➢ DFS

The depth-first search [21] , is a recursive method that employs the backtracking. It requires searching all nodes thoroughly, advancing ahead if possible and retracing if required. To go to the next node, remove the top node from the stack and stack all of its neighbours.

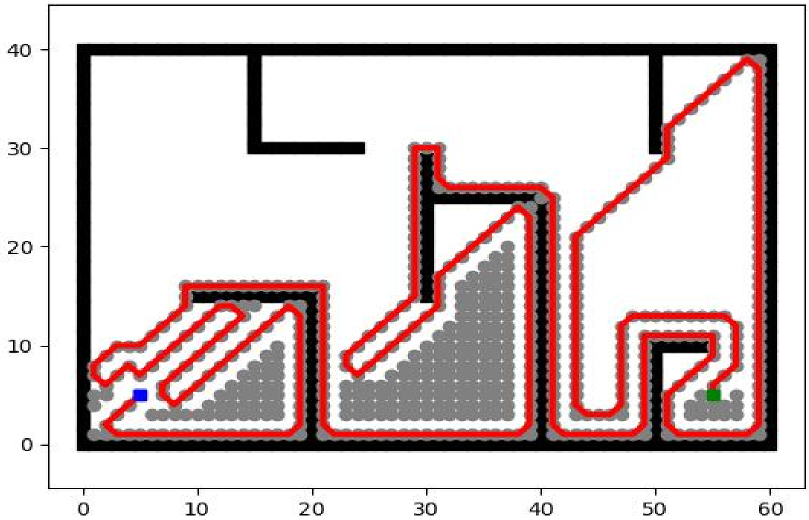


Fig 3.3 Path traced using DFS algorithm

➢ Dijkstras

A single-source shortest route algorithm Dijkstra [22], refers to the fact that there is only one source and we must discover the lowest cost path from the source to all nodes.

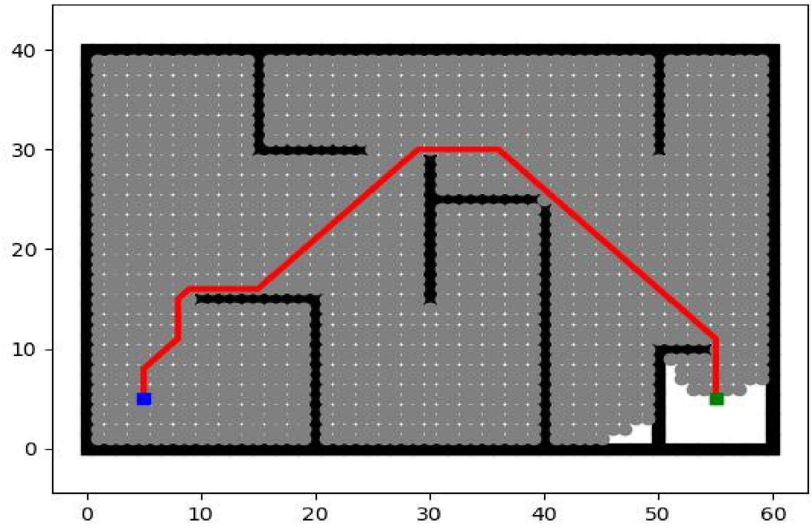


Fig 3.4 Path traced using Dijkstra algorithm

➢ A*

When the objective is to find the lowest cost path between two nodes, A* [23] algorithm which is based on Dijkstra' produce a speedier solution. It accomplishes this by incorporating a heuristic element that aids in the selection of the next node to examine as the path progresses.

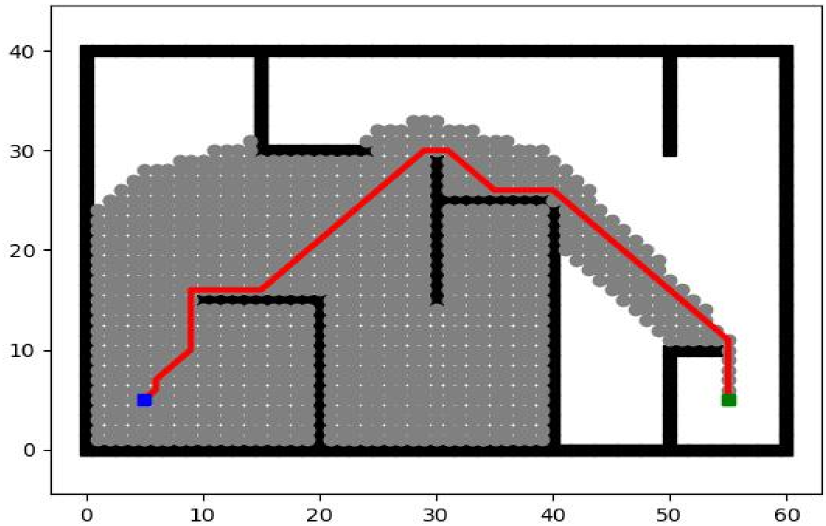


Fig 3.5 Path traced using A* algorithm

Table 3.1 Results of the execution of algorithms

Algorithm Name	**Path Length in Units**	**Execution Time in seconds**
BFS	64	0.08 second
DFS	361	2.74 seconds
Dijkstra	64	0.05 seconds
A*	64	0.04 seconds

After evaluating several algorithms we can observe that we need to compromise on one of the factors like time or distance A* performed better than the others.

CHAPTER 4
IMPLEMENTATION

4.1 EXPERIMENT SETUP

Using python programming the algorithm and the environment has been has been developed. The obstacles have been placed. And the environment is is not dynamic. The information present are the source,destination and the checkpoints. Checkpoints can be considered as recharge stations in the real world conditions. A* is the main algorithm that has been used to built main global path.

4.2 ALGORITHM

The basic algorithm of the implementation is shown below

Step 1 - Obtain a path from starting point to the ending point with no restrictions on distance. Begin your journey

Step 2 - Check if you can get to your target within the time/cost constraints given your present location

Step 3 - If you are unable to reach your goal from your present location, look for a non-visited checkpoint that is closer to the destination with an available distance

Step 4 - Choose the checkpoint with the lowest path cost to the destination + the cost to the checkpoint from your present position if several checkpoints are available

Step 5 - Build an ideal way to the specified checkpoint, then form a path from the checkpoint to the destination, reset, mark the visited checkpoint, and continue from step 2

Step 6 - If you can't reach the endpoint or a checkpoint, proceed to the next place on the most recent path and deduct the cost from the available cost; then repeat step 2 until you reach the checkpoint or the destination

Step 7 - Terminate if the distance/cost limit is exceeded, there are no checkpoints available, and the endpoint has not yet been reached

Fig 4.1 Algorithm of dynamic path planning

4.3 FLOWCHART

The basic implementation flowchart is shown below

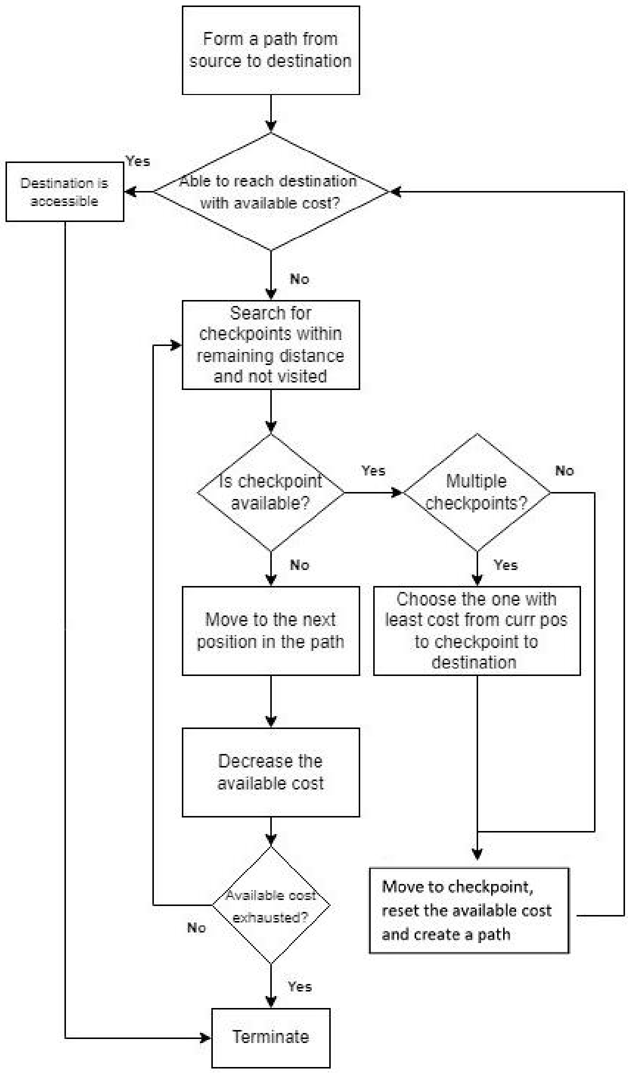
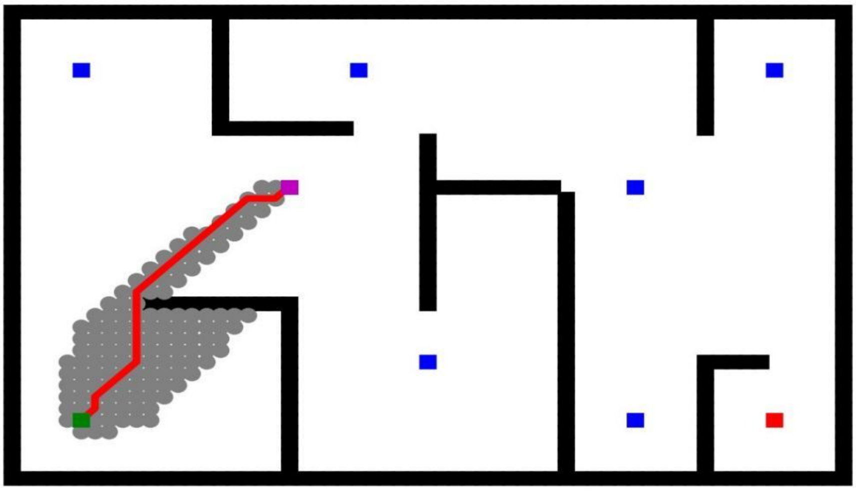
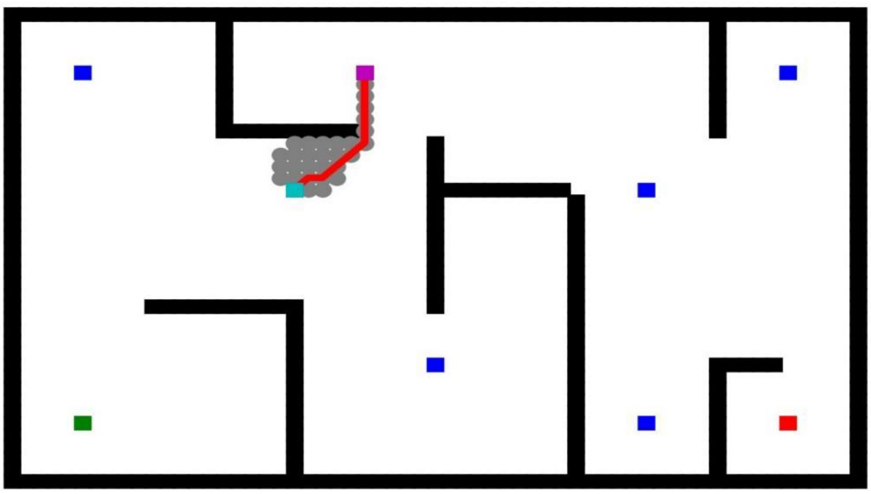


Fig 4.2 Flowchart of dynamic path planning

CHAPTER 5
RESULTS

The environment used for caparison was reused and carried out for 10 iterations. The max path length specified was 25 units. The results obtained are the average of those results. The environment had checkpoints at the fixed position, starting point, endpoint, and the obstacles defined beforehand. The below figure is the path taken by the algorithm where the final path length was 77 units. Initially, it calculated if the destination was reachable. As it was not reachable within the given limit, it chose a checkpoint within the given range and chose the one the nearest to the destination and was not visited. This process carried out till the endpoint was reached. In the path traversed, three checkpoints were visited before reaching the endpoint.





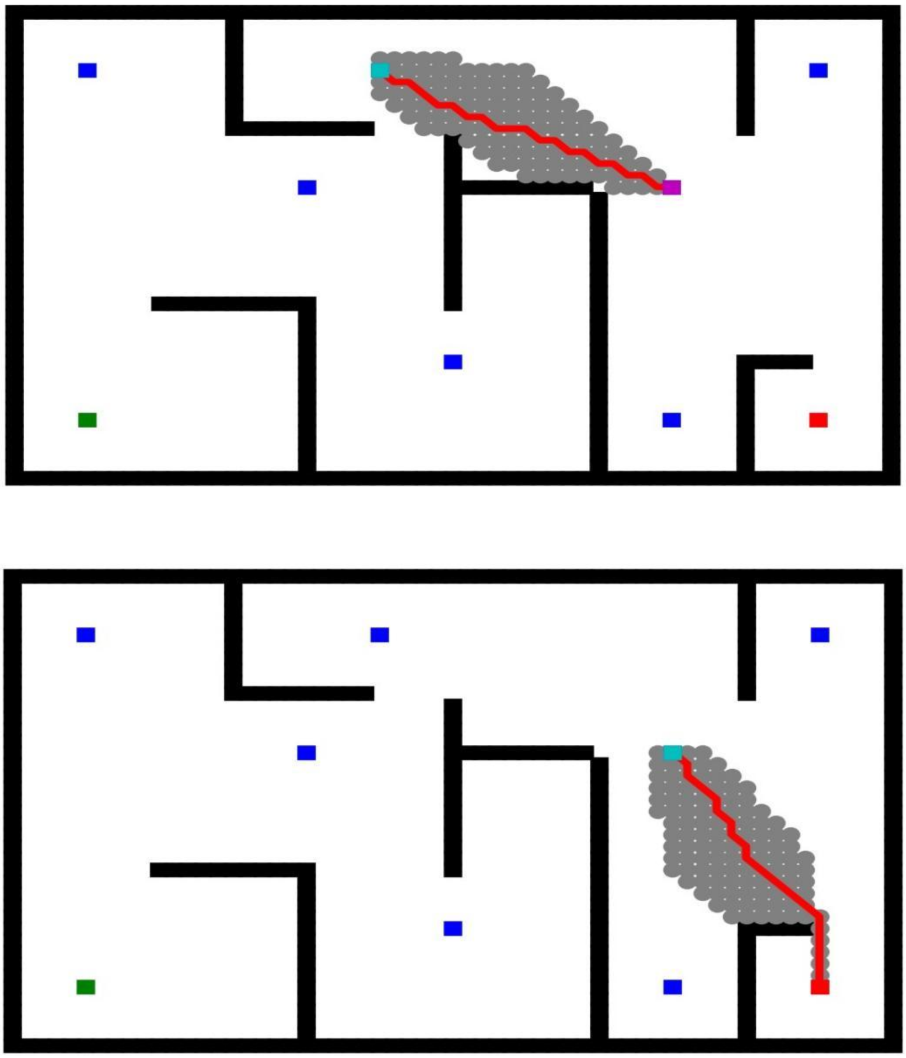


Fig 5.1 Path traversal of dynamic algorithm using A*

CHAPTER 6
CONCLUSION & FUTURE WORK

In this, we compared various path planning algorithms and what are the roles of each stage. A global path planner is mainly used when the environment is globally known where no dynamic entities are present. After evaluating both algorithms for several iterations of the experiment we can conclude that the algorithm built only around perform A* performed optimally when compared to only A*. But this also depends on the environment and location of the checkpoints. This algorithm can be used for real-world problems where we can find battery constraints. This way, the battery level, and other factors of the distance can be predicted by which path can be dynamically built based on that. This algorithm can be further improved to absorb various other dynamic factors like traffic, friction, and obstacles like animals, humans, etc. This can be further integrated with evolutionary algorithms to be optimized [30].

CHAPTER 7
REFERENCES

[1] Gasparetto, Alessandro, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. "Path planning and trajectory planning algorithms: A general overview." Motion and operation planning of robotic systems (2015): 3-27

[2] Amato, Nancy M., and Yan Wu. "A randomized roadmap method for path and manipulation planning." In Proceedings of IEEE international conference on robotics and automation, vol. 1, pp. 113-120. IEEE, 1996.

[3] Gonzalez, Ramon, Marius Kloetzer, and Cristian Mahulea. "Comparative study of trajectories resulted from cell decomposition path planning approaches." In 2017 21st International Conference on System Theory, Control and Computing (ICSTCC), pp. 49-54. IEEE, 2017.

[4] Warren, Charles W. "Global path planning using artificial potential fields." In 1989 IEEE International Conference on Robotics and Automation, pp. 316-317. IEEE Computer Society, 1989.

[5] Taketomi, Takafumi, Hideaki Uchiyama, and Sei Ikeda. "Visual SLAM algorithms: a survey from 2010 to 2016." IPSJ Transactions on Computer Vision and Applications 9, no. 1 (2017): 1-11.

[6] Zhu, Zexuan, Jun Xiao, Jian-Qiang Li, Fangxiao Wang, and Qingfu Zhang. "Global path planning of wheeled robots using multi-objective memetic algorithms." Integrated computer-aided engineering 22, no. 4 (2015): 387-404.

[7] Imai, Hiroshi, and Takao Asano. "Efficient algorithms for geometric graph search problems." *SIAM Journal on Computing* 15, no. 2 (1986): 478-494.

[8] Barraquand, Jérôme, Lydia Kavraki, Jean-Claude Latombe, Rajeev Motwani, Tsai-Yen Li, and Prabhakar Raghavan. "A random sampling scheme for path planning." *The International Journal of Robotics Research* 16, no. 6 (1997): 759-774.

[9] H. J. Bremermann, "The evolution of intelligence. The Nervous system as a model of its environment," Technical Report no.1, contract no. 477(17), Dept. Mathematics, Univ. Washington, Seattle, July 1958.

[10] J. H. Holland, "Adaptation in natural and artificial systems. Ann Aebor," MI: University of Michigan Press.

[11] J. Xia, Z. Michalewicz, L. Zhang and K. Trojanowski, “Adaptive evolutionary planner/navigator for mobile robot,” IEEE Transcation on Evolutionary Computation, vol. 1, no. 1, Apr 1997.
[12] P. G. Zavlangas and S. G. Tzafestas, “Motion control for mobile robot obstacle avoidance and navigation: a fuzzy logic-based approach,” Systems Analysis Modeling Simulation, vol. 43, no. 12, pp. 1625-1637, 2003.
[13] G. Castellano, G. Attolico and A. Distante, “Automatic generation of fuzzy rules for reactive robot controllers,” Robotics and Autonomous Systems, vol. 22, no. 2, pp. 133-149, 1997
[14] D. Janglova, “Neural networks in mobile robot motion,” International Journal of Advanced Robotic Systems, vol. 1, pp. 15-22, 2004.
[15] X. S. Yang, “Nature-inspired metaheuristic algorithm,” Luniver press, 2008.
[16] A. Hidalgo-Paniagua, A. Miguel, Vega–Rodriguez, J. Ferruz and N. Pavon, “Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach,” Soft Computing, vol. 1-16, 2015.
[17] R. C. Eberhart and J. A. Kennedy, “A new optimizer using particle swarm theory”, Proceedings of the sixth international symposium on micro machine and human science, IEEE service center, Piscataway, NJ, NAGOYA, Japan, pp. 39-43, 1995.
[18] X. Tang, L. Li and B. Jiang, “Mobile robot SLAM method based on multi-agent particle swarm optimized particle filter,” The Journal of China Universities of Posts and Telecommunications, vol. 21, no. 6, pp. 78-86, 2014.
[19] T. Guan-Zheng, HE Huan and S. Aaron, “Ant colony system algorithm for real time globally optimal path planning of mobile robots,” Acta Automatica Sinica, vol. 33, no. 3, pp. 279-285, 2007.
[20] B.Awerbuch and R.Gallager, "An 0(v1.6log V+E) distributed BFS", preprint, November 1984.
[21] Awerbuch, Baruch. "A new distributed depth-first-search algorithm." *Information Processing Letters* 20, no. 3 (1985): 147-150
[22] Dijkstra, Edsger W. "A note on two problems in connexion with graphs." *Numerische mathematik* 1, no. 1 (1959): 269-271.
[23] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4, no. 2 (1968): 100-107.
[24] Koenig, Nathan, and Andrew Howard. "Design and use paradigms for gazebo, an open-source multi-robot simulator." In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), vol. 3, pp. 2149-2154. IEEE, 2004.

[25] Krotkov, Eric, Douglas Hackett, Larry Jackel, Michael Perschbacher, James Pippine, Jesse Strauss, Gill Pratt, and Christopher Orlowski. "The DARPA robotics challenge finals: Results and perspectives." Journal of Field Robotics 34, no. 2 (2017): 229-240.

[26] Hambuchen, Kimberly A., Monsi C. Roman, Amy Sivak, Angela Herblet, Nathan Koenig, Daniel Newmyer, and Robert Ambrose. "NASA's space robotics challenge: advancing robotics for future exploration missions." In AIAA SPACE and Astronautics Forum and Exposition, p. 5120. 2017.

[27] Bohren, Jonathan, Tully Foote, Jim Keller, Alex Kushleyev, Daniel Lee, Alex Stewart, Paul Vernaza, Jason Derenick, John Spletzer, and Brian Satterfield. "Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge." Journal of Field Robotics 25, no. 9 (2008): 598-614.

[28] Harrison, William, Anthony Downs, and Craig Schlenoff. "The agile robotics for industrial automation competition." AI Magazine 39, no. 4 (2018): 73-76.

[29] Rouček, Tomáš, Martin Pecka, Petr Čížek, Tomáš Petříček, Jan Bayer, Vojtěch Šalanský, Daniel Heřt et al. "Darpa subterranean challenge: Multi-robotic exploration of underground environments." In International Conference on Modelling and Simulation for Autonomous Systems, pp. 274-290. Springer, Cham, 2019.

[30] Susan, Seba, Rohit Ranjan, Udyant Taluja, Shivang Rai, and Pranav Agarwal. "Global best optimization of ANN trained by PSO using the non-extensive cross-entropy with Gaussian gain." Soft Computing 24, no. 23 (2020): 18219-18231.

CHAPTER 8
LIST OF PUBLICATIONS

[1] Vikas P Damle and Seba Susan. “Dynamic Algorithm for Path Planning using A-Star with Distance Constraint.” In *2022 IEEE 2ND International Conference for Intelligent Technologies (CONIT)*.

[2] Vikas P Damle and Seba Susan “Virtal Environment To Test Real-Life Systems Or Scenarios Using Gazebo A Review.” In *2022 IEEE 2ND Asian Conference on Innovation in Technology (ASIANCON).* (Communicated)