Implementation of A* Motion Planning in ROS 2 for Obstacle Avoidance in a 2D Occupancy Grid

Mohamed Eljahmi
Department of Robotics Engineering
Worcester Polytechnic Institute
Worcester, Massachusetts

***Abstract*—Motion planning is a fundamental capability for autonomous systems, enabling robots to compute safe trajectories in environments with obstacles. Many applications, from mobile navigation to surgical robotics, can be modeled as graph search problems where configurations are represented as nodes and feasible transitions as edges. This project proposes a motion planning framework in Robot Operating System 2 (ROS 2) that generates collision-free paths on two-dimensional occupancy grids. The framework implements the A* algorithm as a baseline, subscribing to occupancy grid maps and start/goal poses, and publishing paths for visualization in RViz. Evaluation will be performed in synthetic environments with static obstacles. The resulting ROS 2 package demonstrates the principles of heuristic graph search while providing a foundation for extensions to additional algorithms and application domains.**

***Keywords*—motion planning, A* algorithm, Dijkstra algorithm, BFS, DFS, graph search, surgical robotics**

I. Introduction

Motion planning is a central problem in robotics, concerned with computing feasible and collision-free trajectories in environments that contain obstacles. A wide range of robotic applications rely on effective planning, including mobile robots operating in warehouses, aerial vehicles navigating cluttered airspace, and surgical robots executing delicate procedures in confined anatomical regions. In such high-stakes contexts, reliable planning is essential to ensure safety and task success.

A unifying feature of many planning problems is that they can be abstracted as graph search. Configurations or waypoints are modeled as nodes, and feasible transitions between them are represented as edges. Classical algorithms such as Dijkstra's method guarantee optimal solutions but are computationally expensive for large graphs. The A* algorithm extends this approach by introducing heuristics that direct the search toward the goal, reducing computation while maintaining optimality under consistent heuristics [1]. As a result, A* has become a standard baseline for deterministic motion planning in robotics.

This project aims to implement a modular motion planning framework in the Robot Operating System 2 (ROS 2) that accepts graph-based environment representations and computes paths using heuristic search. A* will serve as the initial demonstration algorithm, producing collision-free paths on two-dimensional occupancy grid maps. The framework will subscribe to occupancy grid and start/goal inputs, compute feasible paths, and publish results for visualization in RViz. By focusing on a general graph-based design, the system provides both a complete, reproducible demonstration of A* and a foundation for extending the framework to alternative search methods and diverse application domains.

A motivating application scenario is warehouse robotics, where mobile robots must navigate between shelves to transport items. This environment highlights the importance of efficient 2D occupancy grid planning under time and space constraints, making A* a suitable baseline method for reliable pathfinding.

This project builds on the concepts explored in Homework 1, where grid-based environments and simple traversal patterns such as the Victor-Sierra search were implemented to illustrate the fundamentals of search. Those exercises emphasized visualization and coverage but did not guarantee optimal paths. The present project advances this foundation by implementing the A* algorithm in ROS 2, enabling optimal path planning on occupancy grids, standardized message passing, and visualization in RViz. The work therefore transitions from classroom exercises to a reproducible motion planning framework aligned with robotics software practices.

II. Project Team

This project is an individual effort by Mohamed Eljahmi. All tasks — including literature review, algorithm implementation, ROS 2 integration, visualization, evaluation, and documentation — will be performed by the proposer. The project is scoped appropriately for a solo submission, focusing on a complete A* motion planning system within ROS 2.

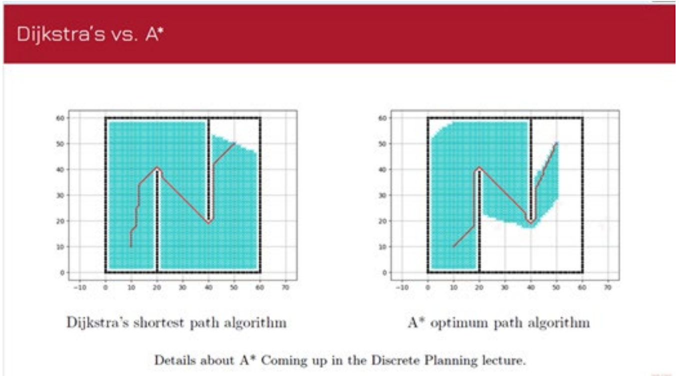
III. Background

Graph-based search methods provide some of the earliest solutions to robotic motion planning. In these approaches, the environment is represented as a discrete graph, with vertices corresponding to feasible configurations and edges representing allowable transitions. Classical algorithms such as Breadth-First Search (BFS) and Depth-First Search (DFS) guarantee completeness in finite graphs but are inefficient in terms of computational cost, particularly when applied to large or dense state spaces [1].

Dijkstra's algorithm improves upon these methods by assigning weights to edges and expanding nodes according to cumulative path cost. This ensures optimality of the computed path but requires uniform exploration of the search space, which may result in unnecessary expansions [2]. To address this limitation, Hart, Nilsson, and Raphael introduced the A* algorithm, which augments uniform-cost search with a heuristic function that directs the search toward the goal state [3]. When admissible heuristics are applied, A* maintains optimality while reducing computational effort compared to Dijkstra's method.

As emphasized in the course material [4], these algorithms illustrate a progression from uninformed search (BFS and DFS), to cost-based optimal search (Dijkstra), to heuristic-guided optimal search (A*). This conceptual progression was first explored in Homework 1, where grid-based traversal patterns were implemented. The current project extends that foundation by applying A* in ROS 2 to compute optimal paths on occupancy grids with standardized message interfaces.

These graph search techniques have seen widespread adoption across domains such as autonomous driving, warehouse logistics, and surgical robotics. In the latter application, planners must generate precise, collision-free paths in highly constrained anatomical regions where computational efficiency and reliability are paramount [5]. The simplicity, predictability, and optimality guarantees of graph-based planners make them an effective foundation for robotic motion planning and a natural starting point for more advanced methods.



Dijkstra's vs. A*

Dijkstra's shortest path algorithm

A* optimum path algorithm

Details about A* Coming up in the Discrete Planning lecture.

Fig. 1. Comparison of Dijkstra's shortest path algorithm and the A* algorithm. A* incorporates a heuristic function, which reduces the number of expanded nodes while maintaining optimality [4].

IV. Proposed Methods

The proposed system will be implemented as a modular motion planning framework in the Robot Operating System 2 (ROS 2). The framework will operate on graph-based representations of the environment, allowing flexibility in both algorithm selection and application domain. In this formulation, vertices correspond to discrete configurations or grid cells, and edges represent feasible transitions between them.

The initial implementation will focus on the A* algorithm as the baseline planner. A* maintains two sets of nodes—an open set of candidates for expansion and a closed set of visited nodes—while using a heuristic to guide the search toward the goal. The cost function combines the cumulative cost of the path from the start node with an estimate of the remaining distance to the goal. For occupancy grid maps, this estimate will be based on Euclidean or Manhattan distance. The planner will terminate when the goal node is reached, at which point the path is reconstructed through backtracking of parent pointers.The ROS 2 package will consist of the following components:

- **Inputs:**
- *Occupancy grid:* Subscribed from the /map topic (nav_msgs/OccupancyGrid). Cells above a specified threshold will be treated as obstacles.
- *Start and goal poses:* Subscribed from /start and /goal topics (geometry_msgs/PoseStamped).
- **Core algorithm:**
    - Implementation of A* with configurable neighbor connectivity (four- or eight-connected grids).
    - Heuristic function selection (Euclidean or Manhattan distance).
    - Obstacle-aware node expansion.
- **Outputs:**
    - A planned path published on the /planned_path topic (nav_msgs/Path). The path will consist of waypoints reconstructed from the search tree.
- **Visualization:**
    - Visualization of the occupancy grid, start and goal poses, and the computed path in RViz.
    - Optional visualization of search expansion for educational purposes.
    - (Figure 1: Example occupancy grid with obstacles and a planned path — to be added once implementation begins.)
- **Evaluation:**
    - Experiments in synthetic environments with varying obstacle densities.
    - Metrics including path length, number of expanded nodes, and computation time.
    - Optional extension to physical hardware for validation with real sensor data.

This modular design enables the framework to support alternative algorithms such as Dijkstra, BFS, or DFS with minimal changes to the overall architecture. The initial implementation of A* therefore serves not only as a proof of concept but also as a foundation for expanding the framework to other graph search methods and broader application domains.

V. Goals and Expected Results

The primary goal of this project is the development of a modular motion planning framework in ROS 2 that demonstrates the principles of graph-based search through a complete implementation of the A* algorithm. The system is expected to integrate seamlessly with standard ROS 2 message types, compute collision-free paths in occupancy grid environments, and provide visualization tools for analysis and demonstration.

Specific objectives include:

- Implementing the A* search algorithm in Python as a ROS 2 node.
- Subscribing to occupancy grid maps and start/goal poses, and publishing resulting paths as nav_msgs/Path.
- Supporting configurable parameters such as connectivity (four- or eight-neighbor search) and heuristic function (Euclidean or Manhattan distance).
- Visualizing both the generated path and the underlying occupancy grid in RViz.
- Evaluating algorithm performance in synthetic environments with varying obstacle densities.
- Recording quantitative results, including path length, computation time, and number of expanded nodes.
- Providing qualitative results through RViz visualizations and recorded demonstrations.
- Designing the framework such that alternative algorithms (e.g., Dijkstra, BFS, DFS) could be incorporated in future work with minimal modifications.

The expected outcome is a reproducible ROS 2 package that produces collision-free paths in 2D occupancy grids and demonstrates the benefits of heuristic graph search. The project will also establish a foundation for future extensions, including deployment on physical hardware and adaptation to application domains requiring high precision, such as surgical robotics.

VI. Planned Schedule

The project will be completed over the course of the semester in structured phases to ensure steady progress and timely completion. Each phase builds upon the previous, beginning with algorithmic design and culminating in evaluation and documentation. A summary of the planned schedule is provided below:

- **Weeks 1–2:** Review of literature on graph search algorithms, including BFS, DFS, Dijkstra, and A*. Setup of the ROS 2 development environment and validation of required packages.
- **Weeks 3–4:** Implementation of the A* algorithm as a standalone Python module, including open and closed list management, cost computation, and heuristic functions.
- No implementation has been started yet; development will begin in this phase according to the planned timeline.
- **Week 5:** Integration of the A* module into an ROS 2 node, subscribing to occupancy grids and start/goal poses and publishing paths.
- **Weeks 6–7:** Testing of the planner in synthetic occupancy grid environments with varying obstacle densities. Visualization of results in RViz and refinement of algorithm parameters.
- **Weeks 8–9:** Evaluation of algorithm performance using quantitative metrics (path length, computation time, nodes expanded). Optional extension to hardware deployment if resources permit.
- **Week 10:** Compilation of results, preparation of video demonstration, and completion of final report and presentation.

This timeline ensures that the core implementation and simulation experiments are completed by the midpoint of the semester, allowing sufficient time for evaluation, optional extensions, and the preparation of deliverables.

VII. Conclusion

This project addresses the fundamental problem of motion planning by implementing a graph-based framework in ROS 2 with the A* algorithm as the primary demonstrator. The proposed system is designed to generate collision-free paths in occupancy grid environments, publish results using standard ROS 2 message types, and visualize outcomes in RViz. Through experiments in synthetic maps, the framework is expected to highlight the efficiency and optimality of heuristic search while providing a foundation for extensions to other algorithms such as Dijkstra, BFS, and DFS. The project further establishes a platform for future applications in domains where precise and reliable planning is critical, including surgical robotics.

A GitHub repository link will be provided as code development progresses, ensuring transparency and reproducibility.

References

[1] H. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.

[2] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[4] RBE-550 Lecture Notes, “Graph Search,” Worcester Polytechnic Institute, Fall 2025.