Route Navigation Optimizer

A Project Report

Submitted By

Rishav Parasar - 2114110034
Tanishk Sonani - 2114110046
Shreesh Kulkarni - 2114110053
Aditya Mishra - 2114110058

In partial fulfilment of the requirement

For the course of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

Prof. Shital Dumbre

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)

COLLEGE OF ENGINEERING, PUNE- 43

2024-25

BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY)

COLLEGE OF ENGINEERING, PUNE- 43

CERTIFICATE

This is to certify that the Project Based Learning report titled **Route Navigation Optimizer**, submitted by **RISHAV PARASAR (2114110034)**, **TANISHK SONANI (2114110046)**, **SHREESH KULKARNI (2114110053)**, **ADITYA MISHRA (2114110058)**, to the Bharati Vidyapeeth (Deemed to be University), College of Engineering, Pune - 43 for the award of the degree of **BACHELOR OF TECHNOLOGY** in Computer Science and Engineering is a bonafide record of the PBL work done by them under my supervision.

Place: Pune

Prof. Shital Dumbre

Date: 30/09/2024

Abstract

The project, **"Route Navigation Optimizer,"** presents a simulation-based application designed to optimize ride-sharing routes in a grid-based environment using the *A (A-star)** pathfinding algorithm. The system demonstrates the process of navigating a vehicle to service multiple customers by picking them up at specific locations and dropping them off at their respective destinations, all while avoiding obstacles on a 2D grid.

The application is implemented using Java's **Swing** framework for the graphical user interface (GUI) and backend simulation, creating an interactive platform where users can visually observe the route optimization process. By allowing users to dynamically add customer requests through mouse interaction, the system dynamically adjusts the vehicle's route based on proximity and predefined barriers.

The heart of the system lies in the A* algorithm, which calculates the shortest path from the vehicle's current position to the next destination. The grid environment features obstacles, making the simulation more realistic by forcing the pathfinder to navigate around barriers. Each customer's journey involves a pickup and a drop-off point, with the vehicle continuously seeking the nearest customer using Manhattan distance as a heuristic.

In addition to route optimization, the simulation includes visual indicators such as color-coded traces that mark the vehicle's movement and changing colors to signify completed pickups and drop-offs. The system can also reset, clearing the current simulation and allowing new scenarios to be tested.

This project showcases how real-time path optimization techniques can be applied to problems in ride-sharing, logistics, or other scenarios where optimal route calculation is critical. The project also provides a platform for further exploration into route optimization, traffic management, and the efficiency of algorithms like A* in constrained environments.

Table of Contents

1| Introduction

The **Route Navigation Optimizer** project addresses a fundamental problem in modern transportation systems—efficient route planning in dynamic, obstacle-filled environments. With the increasing demand for ride-sharing services such as Uber, Lyft, and Ola, optimizing routes is crucial to minimize travel time, reduce fuel consumption, and improve overall efficiency. This project simulates an environment where an autonomous vehicle navigates through a grid-like map, picking up and dropping off customers while avoiding barriers.

At the core of this system lies a *hybrid approach combining the **A (A-star) pathfinding algorithm** with a **Genetic Algorithm (GA)**** to further enhance optimization. A* is one of the most popular algorithms for calculating the shortest path between two points. It combines the concepts of Dijkstra's algorithm and greedy best-first search, using both the distance traveled and an estimated cost to the destination (heuristic) to guide the path. In this project, the heuristic used is the Manhattan distance, which is particularly suited for grid-based maps since it reflects the movement restrictions of city streets—only horizontal and vertical movements are allowed.

However, to optimize paths across multiple pickup and drop-off requests efficiently, a Genetic Algorithm is applied on top of A* to address the **multi-objective optimization**. The GA explores different route sequences by simulating the process of natural selection—mutating and recombining possible paths to evolve the most efficient route plan. This approach balances the trade-off between finding the shortest individual paths (using A*) and optimizing the overall route across multiple stops, a challenge similar to the traveling salesman problem.

The user interface of the Route Navigation Optimizer is built using Java Swing, offering a simple and interactive way for users to add customers dynamically. Users can click on the grid to place customer pickup and drop-off points, and the system immediately calculates the most efficient path for the vehicle to service these requests. The vehicle navigates the grid, avoiding obstacles that simulate real-world barriers such as buildings or traffic jams. The combined A* + Genetic Algorithm ensures the vehicle finds not only the shortest route but also the most optimal sequence to service each customer, even as new requests come in.

The simulation also visually displays the car's path, marking the trail it takes as it moves from one point to another. When a customer is picked up or dropped off, the color of the trace changes, giving a visual indication of progress. The system runs in real-time, adjusting as new customers are added, obstacles are encountered, or route sequences evolve.

2 | Literature Study

[1] A Matching Algorithm for Dynamic Ridesharing

Authors: Maximilian Schreieck et al.
Conference: International Scientific Conference on Mobility and Transport, mobil.TUM 2016
Relevance to Project:
This study provides a foundation for understanding dynamic ridesharing in urban contexts. It addresses the need for real-time matching algorithms that facilitate efficient ride-sharing among multiple users. The implementation of a high-performance algorithm that can process a large number of ride requests quickly is particularly relevant to optimizing the backend of a ridesharing application.

Key Contributions:

- Development of an automated matching algorithm that efficiently connects drivers and passengers in urban settings.
- Evaluation of the algorithm's performance using real data, demonstrating its potential for scalability in similar applications.

[2] Grid-Based Path-Finding

Author: Peter Yap
Relevance to Project:
This paper discusses the foundational principles of pathfinding using grid-based representations, crucial for implementing navigation in ridesharing applications. The comparative analysis of different grid representations and search algorithms, particularly A*, informs the design choices for route optimization in the project.

Key Contributions:

- Examination of various grid structures and their impact on pathfinding efficiency.
- Introduction of a new grid representation (texes) that could enhance route optimization for dynamic environments like urban ridesharing.

[3] Scalable Ride-sharing through Geometric Algorithms and Multi-hop Matching

Author: Yixin Xu
Relevance to Project:
This thesis tackles the challenges of vehicle dispatching in ride-sharing applications, focusing on improving efficiency and response times. The proposed algorithms that utilize geometric properties to filter potential vehicle candidates are directly applicable to enhancing the performance of the ridesharing application developed in this project.

Key Contributions:

- Innovative methods for pruning infeasible vehicles, thereby streamlining the matching process.
- Introduction of multi-hop ride-sharing algorithms that increase flexibility and reduce overall travel distances, addressing real-world complexities in urban transport scenarios.

[4] Fast Grid-Based Path Finding for Video Games

Authors: William Lee and Ramon Lawrence
Relevance to Project:
The insights from this study on efficient grid-based pathfinding algorithms have direct applications in optimizing routes for ridesharing vehicles. The DBA* algorithm's ability to balance memory and computation time while maintaining low suboptimality rates offers a practical approach for real-time route optimization.

Key Contributions:

- Presentation of the DBA* algorithm, which effectively combines pre-computed databases with real-time pathfinding.
- Empirical results demonstrating the algorithm's efficiency in reducing both memory usage and search time, valuable for enhancing user experience in a ridesharing application.

[5] Genetic Algorithm: A Literature Review

Authors: Annu Lambora, Kunal Gupta, Kriti Chopra
Publisher: IEEE
Publication Date: February 16, 2019
Relevance to Project:
This literature review provides a comprehensive overview of Genetic Algorithms (GAs), discussing their principles, basic workflow, and various applications in optimization and machine learning. It highlights the analogy of GAs to biological processes, emphasizing the importance of selection, crossover, and mutation in generating effective solutions. Understanding the foundational concepts and performance metrics of GAs is essential for their application in optimizing route navigation and other complex problems.

Key Contributions:

- Detailed explanation of the genetic algorithm's operational framework, including initial population generation, fitness evaluation, and evolutionary processes.
- Exploration of the historical context of GAs, tracing their development from John Holland's foundational work in the 1970s.
- Discussion of various applications of GAs in optimization problems, illustrating their versatility in research and development across multiple fields.

3 | Problem Statement

In modern urban environments, efficient route optimization has become a critical component for various transportation services, especially in the rapidly growing domain of ride-sharing platforms like Uber, Ola, and Lyft. The primary challenge lies in navigating through densely populated cities with various obstacles, traffic conditions, and the dynamic nature of customer requests. Each ride-sharing vehicle must efficiently pick up and drop off multiple passengers while minimizing overall travel time, fuel consumption, and wait times for customers.

The problem becomes more complex when considering real-world factors such as:

1. **Dynamic Customer Requests:** In a typical ride-sharing service, customer requests for pickups and drop-offs are constantly changing. A system must be capable of adapting its route in real-time to account for new customers without significantly deviating from its current optimal path.
2. **Obstacle Navigation:** Urban environments are filled with obstacles such as buildings, traffic, or roadblocks that require vehicles to find alternative paths. Any route optimization system must account for these obstacles and ensure that the vehicle can navigate through the city while avoiding these hindrances.
3. **Shortest Path Optimization:** One of the most fundamental requirements of such a system is to ensure that the vehicle follows the shortest or most efficient route from one point to another, balancing speed and distance. This becomes particularly challenging when multiple destinations (pickups and drop-offs) need to be handled simultaneously, which involves complex calculations to find the optimal sequence of routes.
4. **Multiple Customers:** The system must also manage multiple customer pickups and drop-offs in a single trip, finding an optimal route that minimizes travel distance while servicing multiple requests in an efficient order. The vehicle must determine whether to pick up additional customers or drop off current passengers based on proximity and time.

Given these challenges, the key problem is how to develop a system that efficiently navigates a vehicle in a grid-based city environment while handling real-time customer requests and avoiding obstacles. The solution must be computationally efficient, able to dynamically adjust to new information (like incoming customer requests), and capable of computing the shortest path quickly and effectively.

Goal:

The **Route Navigation Optimizer** aims to address the challenges of efficient route planning by implementing a solution that combines the A* pathfinding algorithm with a Genetic Algorithm (GA) to navigate a vehicle through a city-like grid. The objective is to pick up and drop off passengers as efficiently as possible while avoiding barriers and optimizing for the shortest route among multiple locations. The system simulates real-time ride-sharing scenarios where customer requests and obstacles can dynamically alter the landscape, necessitating continuous recalculation of optimal paths. In this context, the project demonstrates how the hybrid approach of A* and GA can effectively solve complex route optimization problems relevant to real-world ride-sharing services and autonomous vehicle navigation systems.

4 | Objectives

[1] Implement Efficient Route Optimization:

- Develop a system that combines the A* pathfinding algorithm with a Genetic Algorithm (GA) to calculate the shortest and most efficient route for a vehicle navigating through a grid-based city environment.
- Ensure the vehicle optimally picks up and drops off multiple customers while avoiding barriers and obstacles in the grid, balancing between route efficiency and sequence optimization.

[2] Real-Time Dynamic Pathfinding:

- Integrate A* with GA to dynamically adjust the vehicle's route based on real-time changes, such as new customer pickup or drop-off requests.
- Continuously recalculate the optimal path as new obstacles or customer requests are introduced during the simulation, with GA improving the sequence of multiple stops.

[3] Simulate Multiple Customer Requests:

- Allow users to add multiple customers to the simulation, each with distinct pickup and drop-off points.
- Utilize the hybrid A* + GA system to manage multiple ride requests efficiently, ensuring the vehicle services customers with minimal delays and travel distance.

[4] Visual Representation of the Simulation:

- Create an intuitive graphical user interface using Java Swing that visually displays the grid, obstacles, customers, vehicle, and its optimized path.
- Provide clear visual feedback for users when customers are picked up, dropped off, and when the vehicle is en route.

[5] Handle Obstacles in the Environment:

- Simulate a city environment with obstacles representing real-world barriers such as buildings, roadblocks, or traffic congestion.
- Ensure the vehicle avoids these obstacles and dynamically adjusts its route using A* + GA to efficiently reach its destination.

[6] Provide a Reset and Simulation Control Mechanism:

- Allow users to start, stop, and reset the simulation to easily manage different scenarios.
- Reset the simulation environment, vehicle position, and customer requests for repeated testing and demonstrations.

[7] Demonstrate the Efficiency of A* + GA:

- Showcase the efficiency of combining A* and GA in solving real-world routing problems through this simulation.
- Provide a practical example of how hybrid route optimization algorithms can be applied to modern transportation systems, such as ride-sharing and autonomous vehicles, particularly in congested cities like those in India.

5 | System Architecture

The system architecture for the Route Navigation Optimizer consists of multiple components working together to simulate efficient route optimization in a grid-based city environment. The architecture is divided into the following key layers and modules:

[1] User Interface (UI) Layer

- **Purpose:** Provide a graphical interface where users can interact with the simulation.
- **Components:**
  - **Java Swing Components:**
    - **Main Frame (Home1 Class):** Displays the application window, title, and control buttons.
    - **Control Buttons:** Includes buttons for "Start Simulation" and "Reset Simulation" to control the state of the simulation.
    - **Grid Panel (PathFinder Class):** Displays the grid environment, vehicles, customers, and paths dynamically. This panel visualizes the system's operations.
    - **Updates Panel:** Displays the current status of the simulation, such as "Simulation in progress" or "Simulation reset."
  - **User Interaction:**
    - Users can add customer pickup and drop-off points by clicking on the grid with the mouse.
    - Visual feedback for customer locations (green for pickup, red for drop-off) and vehicle movements (blue for the vehicle, magenta for traces) is provided.

[2] Core Logic Layer

- **Purpose:** Handle all core logic related to route optimization, customer handling, pathfinding, and vehicle movement.
- **Components:**
  - **PathFinder Class:**
    - *A Pathfinding Algorithm:** Calculates the shortest path from the vehicle's current position to customer pickup/drop-off points while avoiding obstacles. It uses a heuristic-based search to optimize route selection.

- **Customer Management:** Manages customer requests (pickup/drop-off), customer states (whether picked up or not), and ensures they are serviced in the optimal order.
- **Dynamic Obstacle Handling:** Includes a method for creating barriers on the grid, ensuring the vehicle navigates around obstacles.
- **Vehicle Movement:** Moves the vehicle along the optimal path step-by-step while leaving a colored trace for visual representation.
- **Simulation Control:** Handles starting and resetting the simulation, dynamically adjusting the paths based on real-time changes in customer requests and the environment.

[3] Data Layer

- **Purpose:** Store and manage the data associated with the simulation, including customer locations, vehicle position, and obstacles.
- **Components:**
  - **Customer Class:**
    - Represents each customer with pickup and drop-off locations and a unique ID.
    - Tracks whether the customer has been picked up and dropped off.
  - **Car Position:** Stores the current position of the vehicle on the grid.
  - **Grid Structure:**
    - **Grid Size:** The environment is represented by a 2D grid with specific dimensions (31x63 cells), where each cell is 20x20 pixels.
    - **Barriers:** The grid contains obstacles represented as barriers, which prevent the vehicle from moving through certain cells.

[4] Algorithm Layer

- **Purpose:** Implement efficient algorithms to find the shortest path, navigate obstacles, and optimize route selection.
- **Components:**
  - *A Search Algorithm:**
    - **Open and Closed Lists:** Keeps track of nodes to explore and those already explored during the pathfinding process.
    - **Heuristic Function (Manhattan Distance):** Estimates the cost of reaching the destination by summing the horizontal and vertical

distance between two points. This ensures quick convergence towards the goal.

        - **Node Expansion:** Expands neighboring nodes to explore alternative routes and selects the one with the lowest combined cost of travel (gCost + hCost).

    - **Path Reconstruction:** After the goal (pickup or drop-off) is reached, the algorithm reconstructs the optimal path and feeds it to the vehicle to follow.

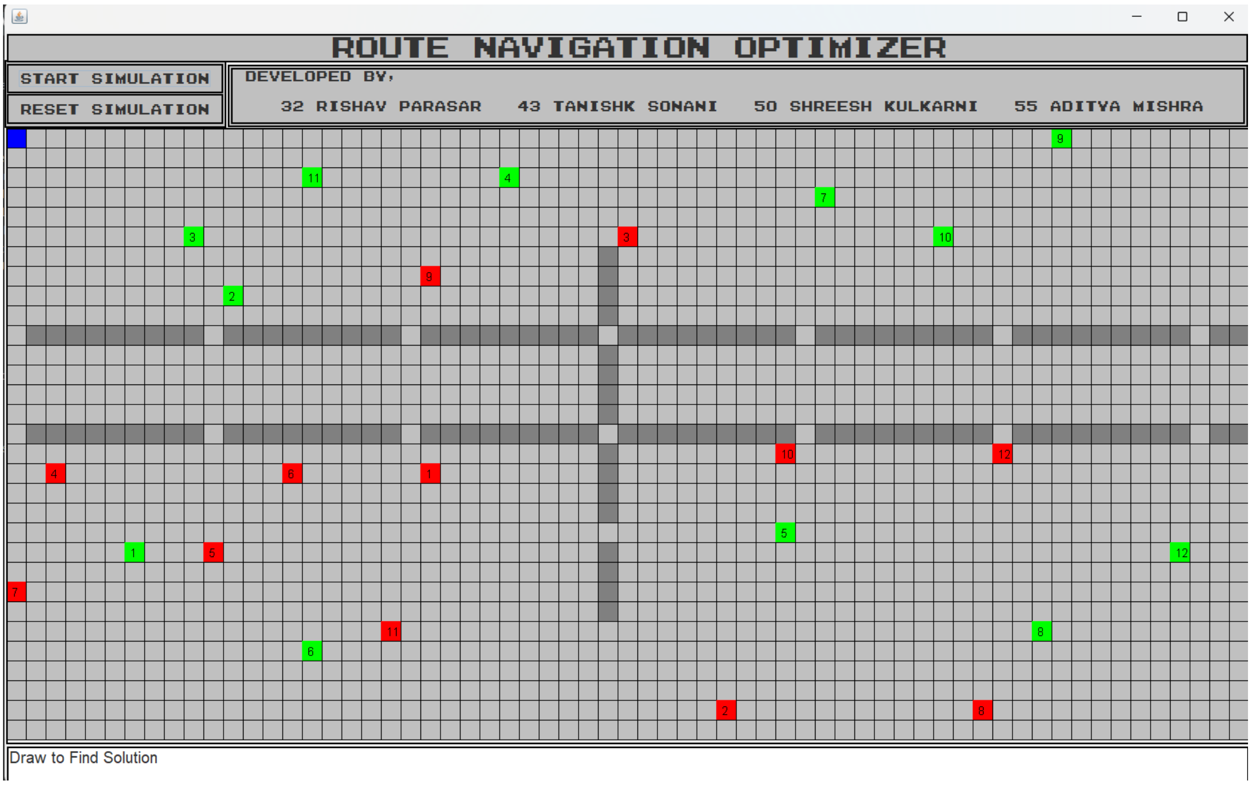
[5] Simulation Layer

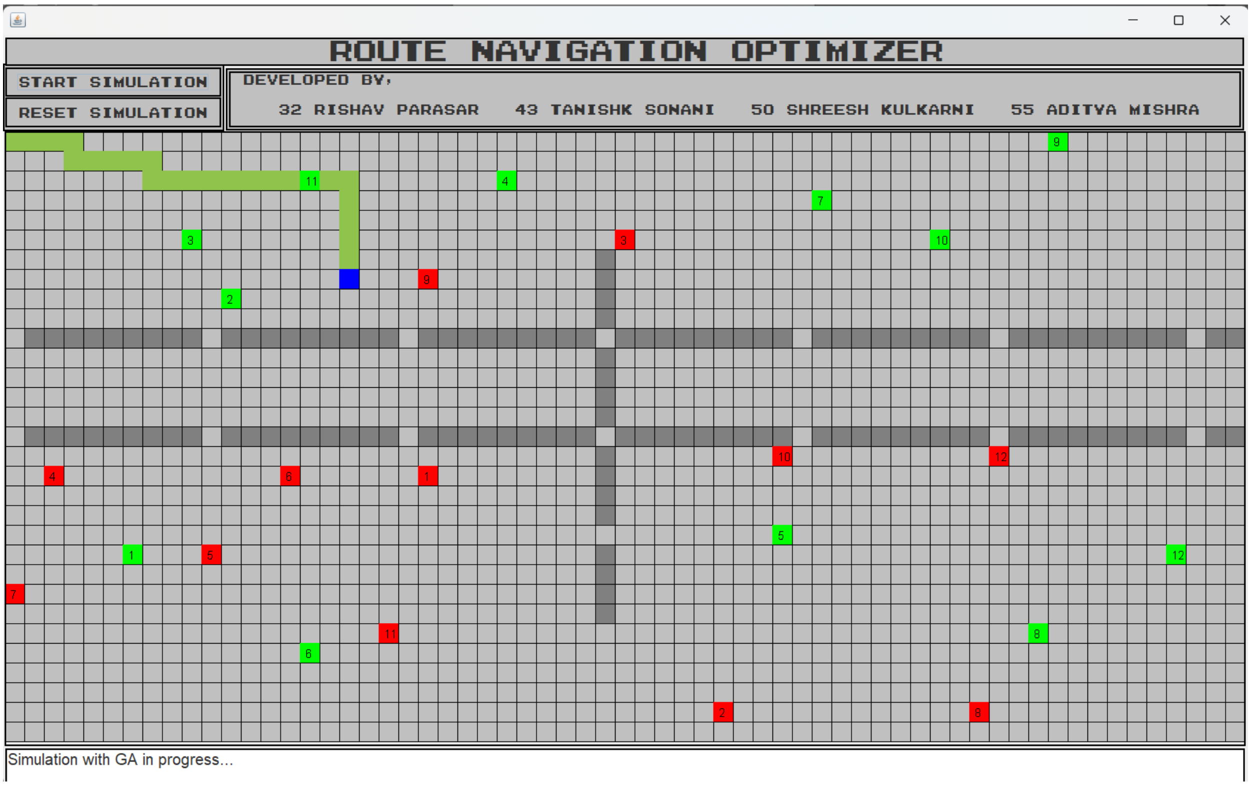
- **Purpose:** Simulate the real-time movement of the vehicle, customer pickups, drop-offs, and changes in the environment.
- **Components:**
    - **Ride-Sharing Simulation:** The vehicle moves along the calculated path to pick up and drop off customers in real-time. The simulation continuously updates as new customers are added or the environment changes.
    - **Real-Time Updates:** The simulation dynamically adjusts the vehicle's route as new requests are introduced. The vehicle recalculates the optimal path based on the current state of the grid.

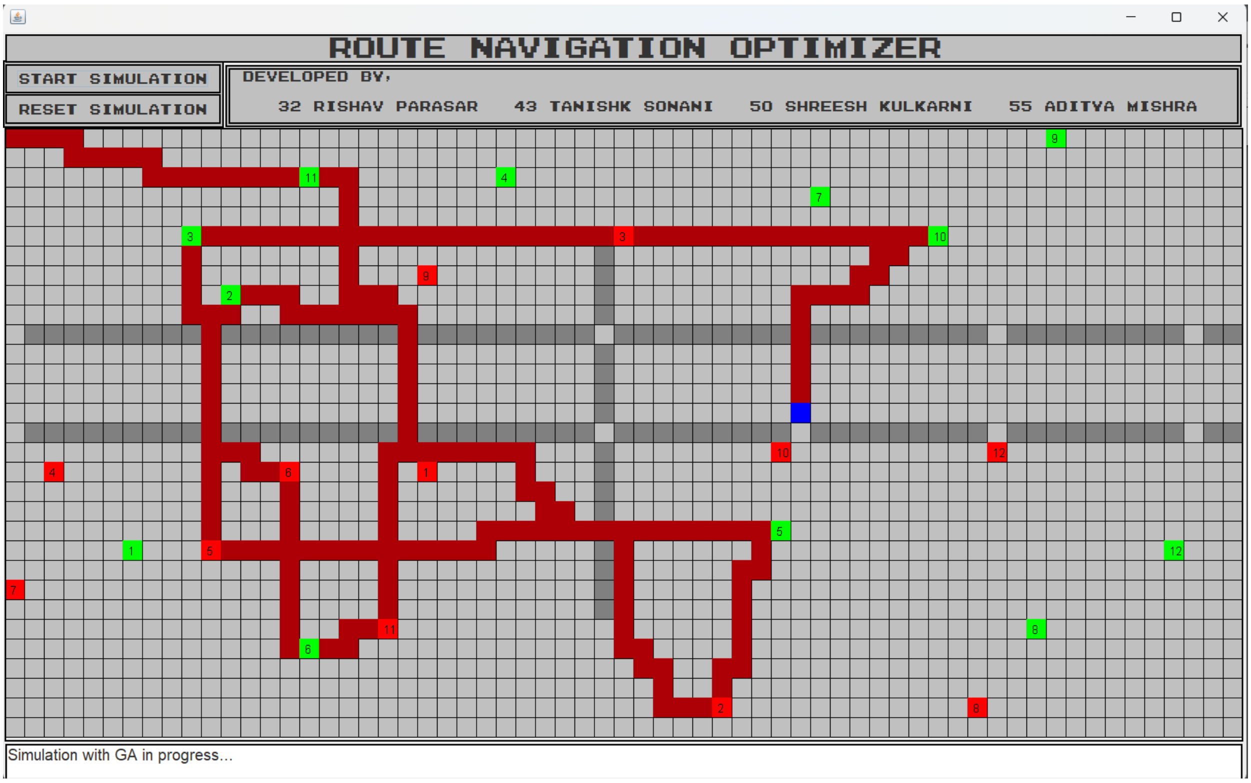
6. Diagram of the System Architecture

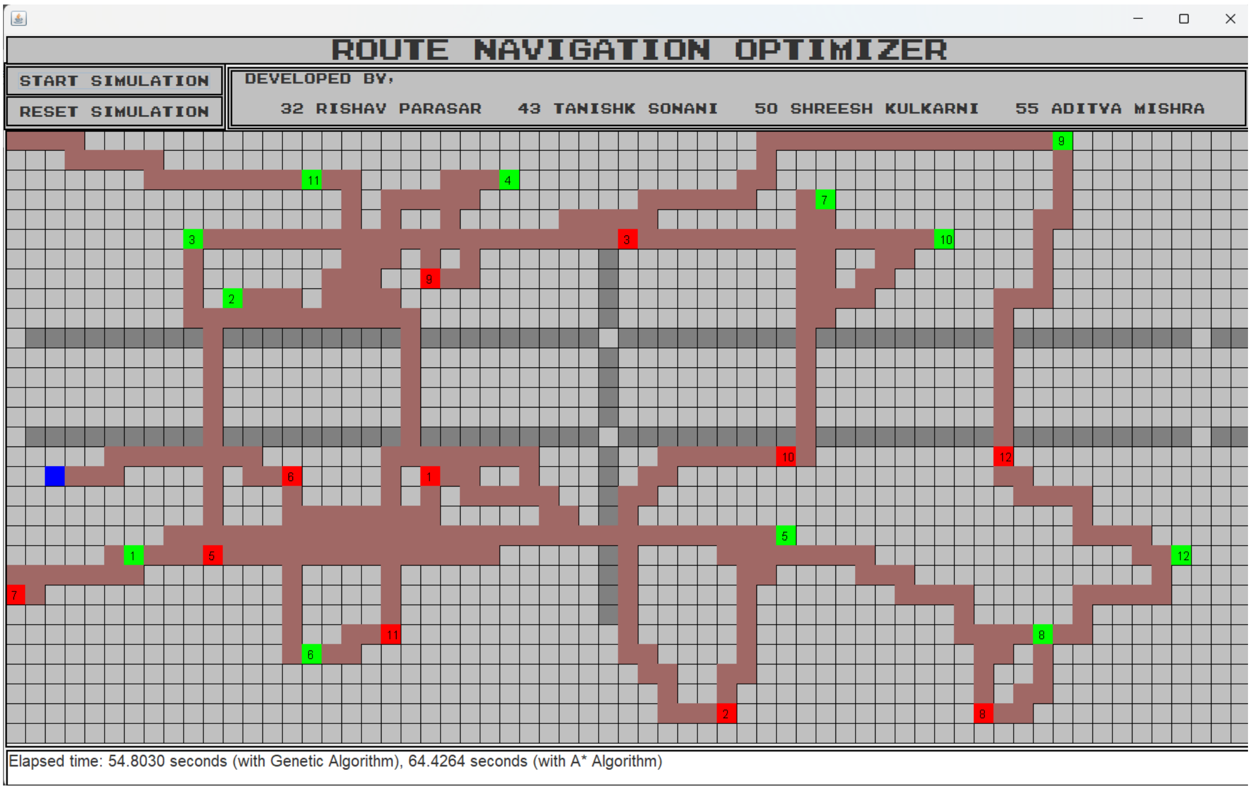
```
+------------------------------------------------------------+
|                 User Interface (UI) Layer                  |
|  +------------------+   +-----------------------------+    |
|  | Home1 (Main Frame) |   | Control Buttons             |  |
|  |                  |   | (Start/Reset Simulation)    |    |
|  +------------------+   +-----------------------------+    |
|  +------------------+   +-----------------------------+    |
|  | Grid (PathFinder) |   | Updates Panel               |   |
|  |   - Customers     |   |   - Simulation Status       |   |
|  |   - Vehicle       |   |                             |   |
|  +------------------+   +-----------------------------+    |
+------------------------------------------------------------+
|                   Core Logic Layer                         |
|  +------------------------------------------------------+  |
|  |  PathFinder (Core Logic for Simulation, Customer, and | |
|  |  Vehicle Management)                                  | |
|  |   - Customer Management                               | |
|  |   - Vehicle Movement                                  | |
|  |   - Obstacle Handling                                 | |
|  +------------------------------------------------------+  |
+------------------------------------------------------------+
|                    Data Layer                              |
|  +------------------------------------------------------+  |
|  |   - Customer Data (pickup, drop-off, ID, state)       | |
|  |   - Car Position                                      | |
|  |   - Grid Structure (Barriers, Grid Size)              | |
|  +------------------------------------------------------+  |
+------------------------------------------------------------+
|                   Algorithm Layer                          |
|  +------------------------------------------------------+  |
|  |  A* Pathfinding Algorithm                             | |
|  |   - Open/Closed Lists                                 | |
|  |   - Heuristic Function (Manhattan Distance)           | |
|  |   - Path Reconstruction                               | |
|  +------------------------------------------------------+  |
+------------------------------------------------------------+
|                   Simulation Layer                         |
|  +------------------------------------------------------+  |
|  |   - Real-Time Ride Sharing Simulation                 | |
|  |   - Dynamic Path Updates                              | |
|  +------------------------------------------------------+  |
+------------------------------------------------------------+
```

6 | Implementation Snapshots






ROUTE NAVIGATION OPTIMIZER
START SIMULATION
RESET SIMULATION
DEVELOPED BY,
32 RISHAV PARASAR 43 TANISHK SONANI 50 SHREESH KULKARNI 55 ADITYA MISHRA
Simulation with GA in progress...


ROUTE NAVIGATION OPTIMIZER
START SIMULATION
RESET SIMULATION
DEVELOPED BY,
32 RISHAV PARASAR 43 TANISHK SONANI 50 SHREESH KULKARNI 55 ADITYA MISHRA
Elapsed time: 54.8030 seconds (with Genetic Algorithm), 64.4264 seconds (with A* Algorithm)

7 | Conclusion

The Route Navigation Optimizer successfully demonstrates the integration of advanced pathfinding algorithms within a user-friendly interface, addressing the complexities of dynamic ride-sharing environments. By utilizing the hybrid approach of the A* search algorithm combined with a Genetic Algorithm (GA), the system efficiently computes optimal routes while accommodating real-time customer pickups and drop-offs, navigating obstacles in a grid-based layout.

This project highlights several key outcomes:

1. **Efficiency in Route Optimization:** The implementation of the A* + GA approach allows for rapid calculation of the shortest paths while optimizing the sequence of multiple customer pickups and drop-offs. This reduces wait times for customers and enhances the overall user experience. The ability to dynamically adjust routes based on real-time customer requests showcases the system's adaptability.
2. **User Engagement through Interactive Interface:** The graphical interface designed with Java Swing provides an intuitive platform for users to interact with the simulation. Users can easily add customer requests and visualize the vehicle's optimized movements, making the system accessible even to those with limited technical expertise.
3. **Scalability and Future Enhancements:** While the current implementation operates within a defined grid size and a set number of barriers, the architecture allows for scalability. Future enhancements could include expanding the grid, implementing more complex road networks, or integrating real-world data to reflect urban traffic patterns and optimize the GA’s performance.
4. **Educational Value:** This project serves as a valuable educational tool, illustrating core concepts in algorithms, data structures, and graphical user interfaces. It provides a practical application of theoretical knowledge in computer science and software engineering, particularly in algorithm optimization.

In conclusion, the Route Navigation Optimizer not only fulfills its objective of providing an efficient ride-sharing solution but also lays the groundwork for further research and development in the realm of intelligent transportation systems. The integration of real-time data, user feedback, and additional features could propel this project into a more comprehensive platform for future applications.

8 | References

[1] Schreieck, M., Safetli, H., Siddiqui, S. A., Pflügler, C., Wiesche, M., & Krcmar, H. (2016). A Matching Algorithm for Dynamic Ridesharing. In *Proceedings of the International Scientific Conference on Mobility and Transport, mobil.TUM 2016* (pp. xx-xx). Munich, Germany. https://www.researchgate.net/publication/312248731_A_Matching_Algorithm_for_Dynamic_Ridesharing

[2] Yap, P. (2013). Grid-Based Path-Finding. In *Canadian AI 2013* (LNAI 7884, pp. 100–111). Springer-Verlag. https://svn.sable.mcgill.ca/sable/courses/COMP763/oldpapers/yap-02-grid-based.pdf

[3] Xu, Y. (2021). Scalable Ride-sharing through Geometric Algorithms and Multi-hop Matching. *PhD Thesis, School of Computing and Information Systems, The University of Melbourne*. https://rest.neptune-prod.its.unimelb.edu.au/server/api/core/bitstreams/4ed189ea-75d9-5c2e-8f87-abafe2b050cb/content

[4] Lee, W., & Lawrence, R. (2013). Fast Grid-Based Path Finding for Video Games. In *Proceedings of the Canadian AI 2013* (LNAI 7884, pp. 100–111). Springer-Verlag. https://page-one.springer.com/pdf/preview/10.1007/978-3-642-38457-8_9

[5] Lambora, A., Gupta, K., & Chopra, K. (2019). Genetic Algorithm- A Literature Review. In Proceedings of the IEEE Conference. IEEE. https://ieeexplore.ieee.org/abstract/document/8862255

[6] OpenAI. "ChatGPT." https://chat.openai.com/chat