City-Planner: An Optimization Framework for Sustainable Urban Resource Allocation

Team Name: CivicSense
Team Members: Vishal T , K B Prathul , Yellamilli Thanmai Sri

Abstract

Rapid urbanization strains infrastructure, leading to congestion and inefficient service. Cities struggle to optimize roadways and railways for growing population centers, impacting economic productivity and sustainability. Current planning relies on static models, lacking dynamic, data-driven insights for sustainable infrastructure development. This project addresses the urgent need for intelligent tools to guide urban growth aligned with global sustainability goals.

Inspired by UNESCO's SDG 9 and 11, our motivation is to build resilient urban environments by optimizing transportation infrastructure based on population distribution. We aim to enhance urban mobility, reduce commute times, and minimize environmental impact. Existing GIS/CAD tools offer visualization but lack advanced multi-objective optimization for integrating diverse data (population, network capacity, demand).

Our objective is to design a system modeling urban areas as a graph, using data structures and algorithms to identify optimal layouts for roadways and railways. Primary goals include maximizing connectivity between population centers, minimizing travel times, and optimizing resource utilization for development.

The system will leverage graph theory (nodes for population centers, edges for links with weighted attributes), shortest path algorithms (Dijkstra's, A*), and custom greedy/heuristic optimization for placement and network design. Key features include modular data integration, a robust computation backend, and basic visualization.

Future scope includes incorporating real-time data, optimizing other urban services (e.g., utility networks), integrating machine learning for predictive analytics, and developing interactive web-based visualization for collaborative planning.

Individual Contributions

- **Vishal**: Handled the Python implementation and contributed to initial idea generation.
- **Prathul**: Contributed to idea generation and implemented the necessary Abstract Data Types (ADTs).
- **Thanmai**: Helped with implementation of Abstract Data Types (ADTs) and created the project's flow chart.

Key Takeaways

- **ADT Application**: Gained a clear understanding of specific, prominent use cases for various ADTs, recognizing how each is ideal for a particular scenario.
- **Python & Libraries**: Learned the Python language and discovered the ease of data manipulation using its extensive libraries.
- **Modular Programming**: Realized the critical importance and utility of a modular programming approach, especially for managing large-scale projects.

- **Development Tools**: Acquired practical skills in version control (e.g., Git) and the creation of Makefiles for project builds.

Future Scope

- **Data-Driven Optimization**: With access to more precise population density data, the algorithm for calculating optimal new roads can be significantly enhanced.
- **Map Integration**: A future goal is to visually add the newly calculated optimal road back into the map data.
- **Algorithmic Efficiency**: Research and implement more advanced and efficient algorithms to improve the speed and accuracy of finding optimal road placements.

References

- **Youtube**: It was used to help in initial idea generation and help with coding in some places.
- **Google**: Helped in solving syntax or library related doubts when needed.
- **Gemini and ChatGPT**: Helped in solving errors in the program and making the code more clear and readable.

Detailed System Structure

1. **Foundation: Abstract Data Type (ADT) Implementation**
   - **Graphs** are the most critical component here, as they are the natural way to represent a road network (where intersections are vertices/nodes and roads are edges).
   - **Stacks, Queues, and Heaps** are supporting structures for the Graph algorithms. For example, Queues are used for Breadth-First Search (BFS), Stacks for Depth-First Search (DFS), and Heaps (as priority queues) are essential for efficient algorithms like Dijkstra's to find the shortest paths.
2. **Data Layer: Importing & Processing**
   - **Transport**: This represents the road map data (the existing network of roads, highways, and intersections).
   - **Services**: This represents key amenities such as schools and hospitals.
3. **Interaction Layer: Graphical User Interface (GUI)**
   - The GUI serves as the project's control panel.
   - It allows users to visualize the current map and, most importantly, initiate the analysis to find new roads.
4. **Core Logic: Reachability-Based Approach**
   - The program analyzes the graph to see how well-connected different areas are.
   - It likely calculates which "Service" areas (or high-population-density zones) are difficult to reach from the main "Transport" network. It identifies "underserved" or "isolated" spots.
5. **Final Output: Identify New Optimal Roads**

- Based on the analysis from the "Reachability" step, the program produces a concrete, actionable result.
- It identifies and suggests specific locations for new roads that would most effectively and optimally connect the underserved areas to the main transport network.

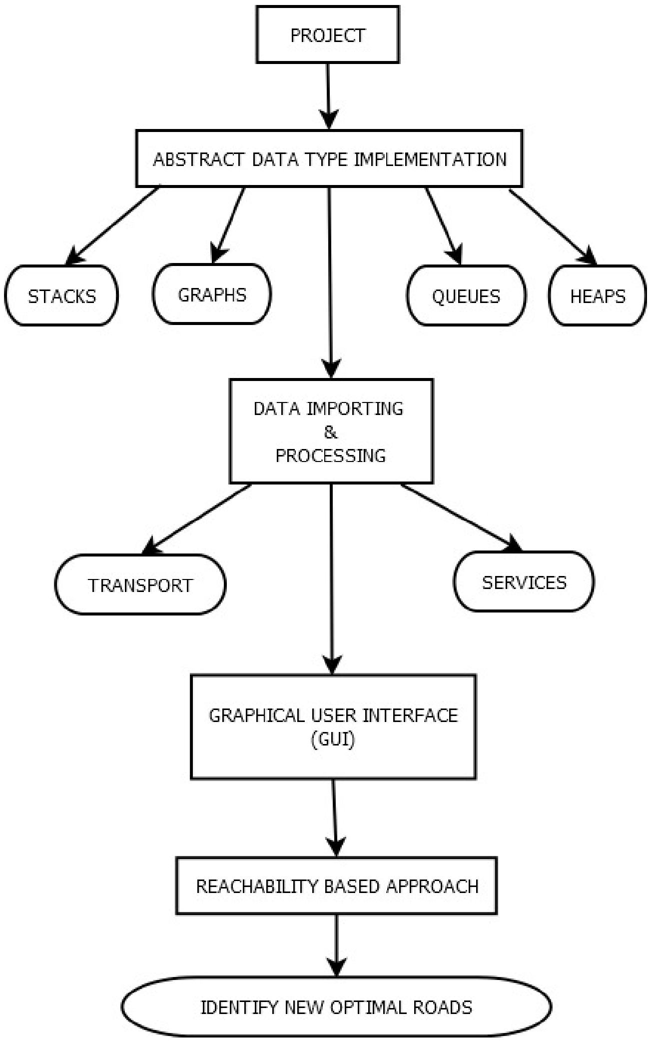

Figure 1: Flow chart detailing the system's logic.