ROUTE PLANNING AND RECOMMENDATION SYSTEM

Data Structure and Algorithm - CSL2020

Instructor Name : Suchetana Chakraborty
Mentor TA: Debottam Bhunia

Team Members
Deepinder Deep Singh - B23MT1015
Ashwani Pratap Singh - B23CI1007
Patitapaban Sahoo - B23ME1045
Uday Shaw - B23CH1045

PROBLEM STATEMENT

PROBLEM

Finding the shortest path between cities considering dynamic traffic conditions.

Problem:

Finding the shortest path between cities considering dynamic traffic conditions.

Domain:

- Smart Transportation Systems
- Intelligent Route Planning

Importance:

- Reduces travel time and congestion.
- Helps logistics and transportation services.
- Real-time route optimization is crucial for efficiency.

Challenge:

- Traffic is time-dependent (morning, afternoon, evening).
- Traditional shortest path algorithms (Dijkstra, Bellman-Ford) assume static edge weights.

Why Data-Driven:

- Traffic data can be modeled and adjusted dynamically.
- Machine-readable formats like JSON make updates seamless.

CURRENT STATUS

1. EXISTING SOLUTIONS:

- Google Maps and Waze use advanced live traffic data.
- Research papers suggest dynamic algorithms like Time-Dependent Dijkstra and modified A*.

2. KEY HIGHLIGHTS:

- Many algorithms are for live updating traffic data.
- Traditional graphs assume static weights.

3. LIMITATIONS IN EXISTING APPROACHES:

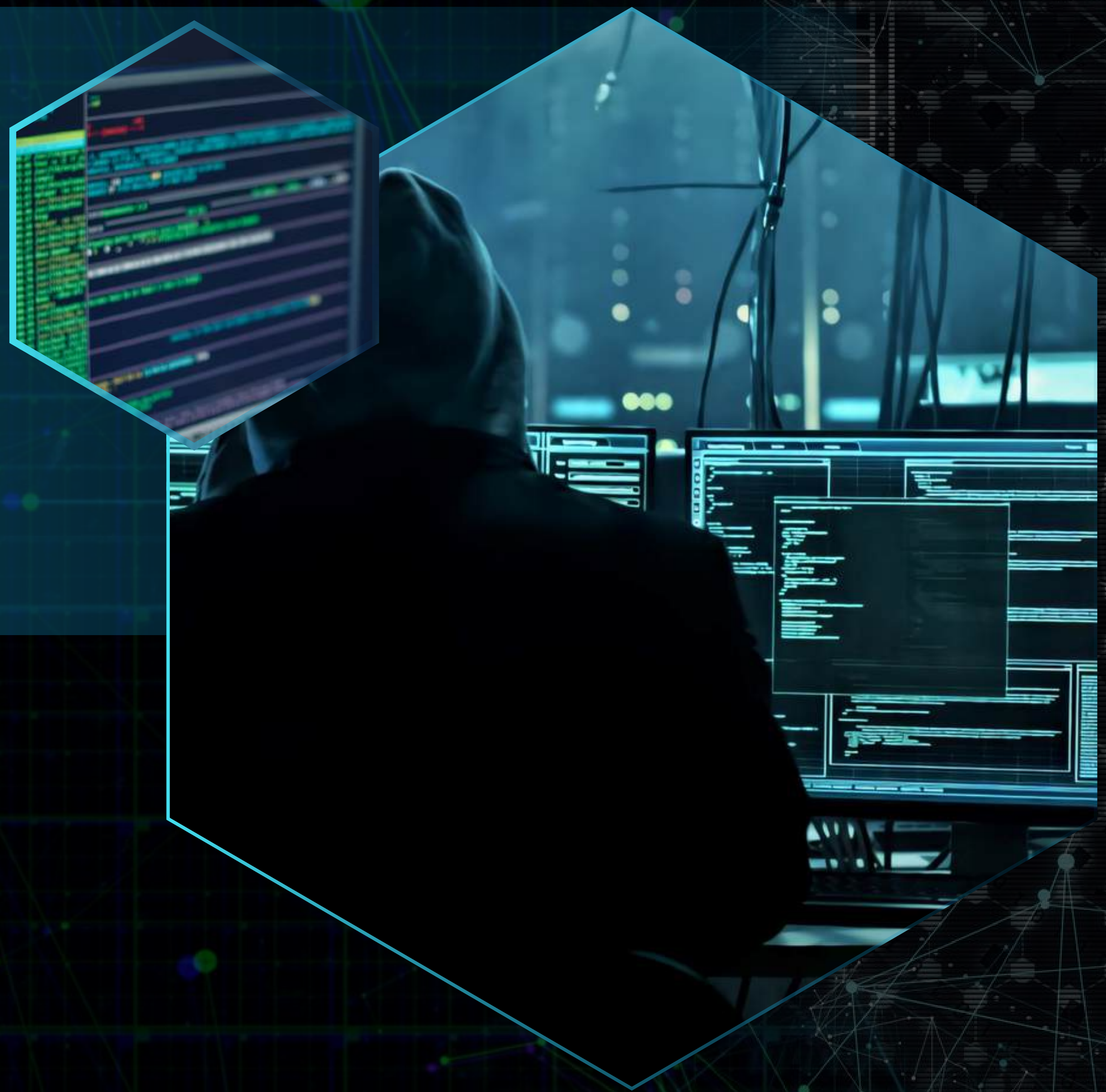
- Real-time traffic is expensive to model.
- High reliance on live data feeds.
- Dijkstra algorithm does not handle negative weights (which arise when traffic improves).
- Existing methods are not suitable for offline usage.
- Bellman-Ford is preferred but slower.

OUR IDEA

The core idea behind this project is to develop a dynamic route optimization system that considers time-dependent traffic conditions — specifically for morning, afternoon, and evening — using a modified Bellman-Ford algorithm. Unlike traditional shortest path problems where road weights are static, we introduced dynamic traffic weights for every edge in the graph, allowing the system to adapt travel times based on the time of day.

We designed a flexible and extensible system where cities and routes are stored in a simple JSON file format. This makes it very easy to update the city graph without recompiling code — ideal for scaling to more cities or integrating live data later. The use of hash maps (unordered_map) to manage traffic conditions ensures efficient lookup and management of dynamic weights. Another major advantage of our solution is that it is designed to be completely offline. Once the JSON data is loaded, users can find the best routes without requiring any internet connection. This makes it highly suitable for remote areas, travelers with limited connectivity, or offline navigation applications — a sharp contrast to solutions like Google Mapsand Apple Maps, which rely heavily on continuous online data access and cloud-based updates.

GITHUB REPO LINK



OUR IDEA

Key improvements over traditional approaches include:

- Handling time-dependent dynamic weights efficiently.
- Offline capability without any external dependency on internet or live servers.
- Compatibility with negative weight edges where traffic can reduce travel time (e.g., empty roads during off-peak hours), which most traditional Dijkstra-based systems cannot handle.

Technical Details:

- Algorithm Used: Modified Bellman-Ford to accommodate dynamic edge weights based on time of day.
- Data Structures:
  - Graph represented as an Adjacency List.
  - Hash Maps for fast traffic weight lookups.
  - JSON Parsing with nlohmann/json library.
- Input Data: Structured JSON file containing cities, coordinates, edges, and traffic variations.

GITHUB REPO LINK

OUR IDEA

Results:

- Successful dynamic shortest path calculation with user-friendly interface.
- Flexible, portable, and future-ready architecture.

Cost Analysis:

- Space Complexity: **O(V + E)**
- Time Complexity: **O(V × E)** per query (manageable for small to medium networks).

Trade-off:

- Bellman-Ford is slower than Dijkstra but necessary here because of possible negative traffic adjustments.
- Sacrificing a bit of speed enables higher flexibility and offline usability.

GITHUB REPO LINK

CONCLUSION

Through this project, we successfully demonstrated a dynamic routing system capable of handling real-world traffic variations based on time of day. By leveraging a modified Bellman-Ford algorithm and JSON-based data storage, we achieved flexible, scalable, and completely offline navigation. The project provided deep insights into working with dynamic graphs, optimized data structures, and real-world applications of algorithms.

1. KEY LEARNINGS:

- Application of Bellman-Ford algorithm to dynamic traffic modeling.
- Efficient parsing and handling of JSON data.
- Practical exposure to offline systems.

2. FUTURE SCOPE:

- Integration with live traffic APIs.
- Expansion to 3D route planning (adding elevation factors).
- Dynamic updates without restart.

3. INDIVIDUAL CONTRIBUTION:

- Deepinder Deep Singh - Developed core algorithm and Bellman-Ford integration. Managed data validation and ensured consistency of the edge data.
- Ashwani Pratap Singh - Designed JSON structure and handled file parsing. Integrated the system with the graph visualization and traffic-based edge weights.
- Patitalaban Sahoo & Uday Shaw- Conducted testing, validation, and documentation and presentation. Developed the time-of-day parsing mechanism. Implemented the logic for edge-case handling like invalid cities or time input.

4. ACKNOWLEDGEMENTS:

- nlohmann::json library for JSON parsing.
- StackOverflow, GeeksforGeeks for reference material.
- Course materials and TA guidance.

THANK YOU