



Multi Modal Route Planning

Presented By :

Anoop Toffy - MT2016016

Harshit Joshi - MT2016059

Mehul Singh - MT2016083

Mohd Awais - MT2016086

Introduction

For a traveler it is often reliable to use different modes of transport for a travel whenever it is available. But there are very few efficient systems to find such a travel plan, there are systems that aid traveler to choose different modes of transport, such as Google Transit.

Multimodal Route Planning(MRP) is a system that provides the traveler with optimal, feasible and personalized route plan between a given source and destination in Bangalore city. The system is designed to provide combinations of various public travel approaches like bus and metro.

Related Work

- ▶ Multi-Modal Route Planning [1] by Thomas Pajor performs multimodal route planning on a large scale networks involving roads, railways and flights. He combined the different realistic models for road, railway and flight networks into multi-modal graphs. He used dijkstras algorithms with labelled constraints.
- ▶ *In another paper Multi-Modal Journey Planning in the Presence of Uncertainty* by Adi Botea et la (2013) [2] solves the multimodal planning problem by introducing a heuristic search planner, based on Weighted AO* (WAO*).

Proposed Solution

► Dataset used

Bus Network:

BMTC Bus network data in csv format.

Files:

- Routes.csv: Description of all bus routes in Bangalore city.
- UniqueBusStops.csv: List of unique bus stops along with Longitude and Latitude information

Metro Network:

- Data scraped from BMRC web page.(manually)

Proposed Solution (cont...)

• Modeling The Network

- **Road Network:**

Nodes: Bus stops.

Edge between two bus stations is inserted if there exists a route between them.

Edge weight: Haversine Distance.

Total Bus stops: 5338

- **Metro Network:**

Nodes: Metro station.

Edge between two metro stations is inserted if there exists a route.

Edge weight: Haversine Distance.

Total Metro stations: 42

Proposed Solution (cont...)

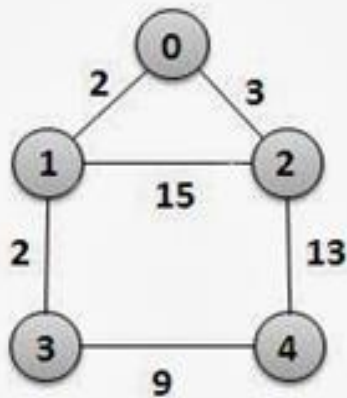
► Combining The Unimodal Networks

Both unimodal networks were combined to the nearest node (<1km distance) which gives the multimodal network.

Proposed Solution (cont...)

Data Structures Used

A. Graph



	0	1	2	3	4
0	0	2	3	0	0
1	2	0	15	2	0
2	3	15	0	0	13
3	0	2	0	0	9
4	0	0	13	9	0

**Adjacency Matrix Representation of
Weighted Graph**

Proposed Solution (cont...)

Data Structures Used (cont..)

B. Priority Queue

It is implemented as a binary heap which supports following operations:

- FindMin in $\Theta(1)$
- DeleteMin in $\Theta(\log n)$
- Insert $\Theta(\log n)$
- Update Key $\Theta(\log n)$

Proposed Solution (cont...)

► Our Graph Node

Each node of graph stores following information required for further processing:

- Bus Stop/ Metro Stop Names
- Longitude and Latitude Information
- List of routes passing through the given nodes

Proposed Solution (cont...)

► Our Transit Algorithms:

We used following two approaches for finding shortest path:

- A* algorithm with haversine heuristics
- Using simple Dijkstra's Algorithm

Dijkstra Algorithm (Pseudo Code)

```
dist[s]  $\leftarrow$  0                                (distance to source vertex is zero)
for all  $v \in V - \{s\}$ 
    do dist[v]  $\leftarrow \infty$                 (set all other distances to infinity)
S  $\leftarrow \emptyset$                             (S, the set of visited vertices is initially empty)
Q  $\leftarrow V$                                 (Q, the queue initially contains all vertices)
while Q  $\neq \emptyset$                             (while the queue is not empty)
do u  $\leftarrow$  mindistance(Q, dist)              (select the element of Q with the min. distance)
    S  $\leftarrow S \cup \{u\}$                     (add u to list of visited vertices)
    for all  $v \in \text{neighbors}[u]$ 
        do if dist[v] > dist[u] + w(u, v)      (if new shortest path found)
            then d[v]  $\leftarrow$  d[u] + w(u, v)  (set new value of shortest path)
                                                (if desired, add traceback code)

return dist
```

Complexity Analysis

- ▶ Time complexity of algorithm = $O(E \log V)$
- ▶ Total number of nodes = 5380
- ▶ Time taken for result = $5380 * 5380$ tp ≤ 2 sec (Approx)

Demo

Multi Model Routing

Source:

Magadi Road Metro Station
Choose a source from the list

Destination:

Magadi Road Metro Station
Choose a destination from the list

Source	Destination	Bus	Distance (in Kms.)
14th main hsr layout	koramangala	K-5	0.678
koramangala	jn of 80 and 100 feet road	171G	1.709
jn of 80 and 100 feet road	Halasuru Metro Station	171G, 171J, 163A, 166, K-5, 173A, 164, 201K, 201, 171, 201L, 201N, 171E, 201A	0.893
Halasuru Metro Station	lingarajapura	Metro2	0.352
lingarajapura	hennur depot	293A, G-11	1.275
hennur depot	nagavara junction	293B, 293, 293K, 296F, 293D, 293M, 294B, 294A, 294, 294C, 293F, 293C, 296D, 295, 412, 295E, 294D, 293L, 292, 292B, 295C, 293H, 293P, 293A, 293J, 292C, MF-4, 295A, 296N, G-11, 294F, 293E, 294E, 296A, 296B, 296K	0.421
nagavara junction	manyatha tech park nagavara in front of ibm manyata embassy business park	290N, MBS-14, 290B, 415H, 290Q, 290, 290S, 290E, 290R, 291H, 290J, 290T, 290A, 291P, 291J, G-10, 290RA, 415E, 291A, BC-9, 290Z, 296K	0.797
manyatha tech park nagavara in front of ibm manyata embassy business park	dasarahalli dasarahalli beside annaporna temple	291	0.736
dasarahalli dasarahalli beside annaporna temple	18th cross srirampura amruthahalli in front of open area	291L	0.697

Total Distance : 7.556 Kms.

*Metro1: Green Line, *Metro2 : Purple Line

App in Motion

Technologies Used

▶ Back-End

- ▶ Python Programming Language
- ▶ Django Web Development Framework
- ▶ Django Rest API

▶ Front-End

- ▶ Django Web Development Framework
- ▶ Bootstrap Framework

Key Challenges

- ▶ A* algorithm with haversine distance as heuristics criteria didn't work well since the heuristic criteria is not found admissible. The algorithm takes exponential time in unbound space, $O(b^d)$ and depends on the heuristics.
- ▶ Combining unimodal networks road and metro was challenging.

Future Work

- ▶ Incorporating Pedestrians
- ▶ Duration Constraint
- ▶ Caching Repetitive queries
- ▶ User location detection
- ▶ Expanding model to other cities

References

1. *Adi Botea IBM Research, Evdokia Nikolova Dept. of Computer Science and Engineering Texas A&M University, Michele Berlingiero IBM Research - "Multi-Modal Journey Planning in the Presence of Uncertainty" 2013 .*
<https://www.aaai.org/ocs/index.php/ICAPS/ICAPS13/paper/view/6023>
2. *Thomas Pajor, Universität Karlsruhe (TH), Germany "Multi-Modal Route Planning" 2009* <http://i11www.iti.uni-karlsruhe.de/extra/publications/p-mmrp-09.pdf>

Questions ?

Thank You!