SnooCODE

Imperial College London

Building a route optimization system that takes elevation into consideration

Jinsong Dong

Supervisors:

Sesinam Dagadu

Dr. Yves Plancherel

Catalogue

SnooCODE Imperial College London

1. Background
2. Methodology
3. Result
4. Limitation
5. Conclusion

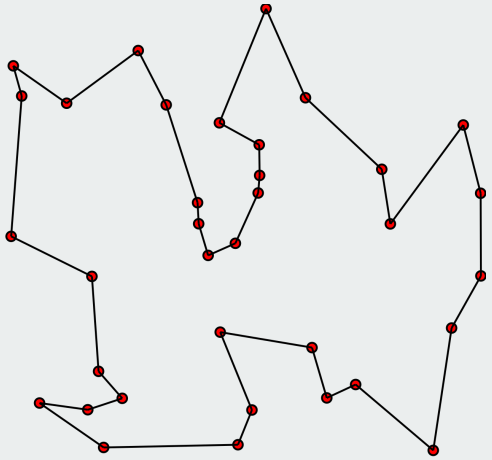
Background

SnooCODE

Imperial College London

- System
  - Elevation
  - No Internet

- Traveling salesman problem
- Carbon footprint



Methodology

- Integration

2D-Road Json file

Point 1 $\{x_1, y_1\}$

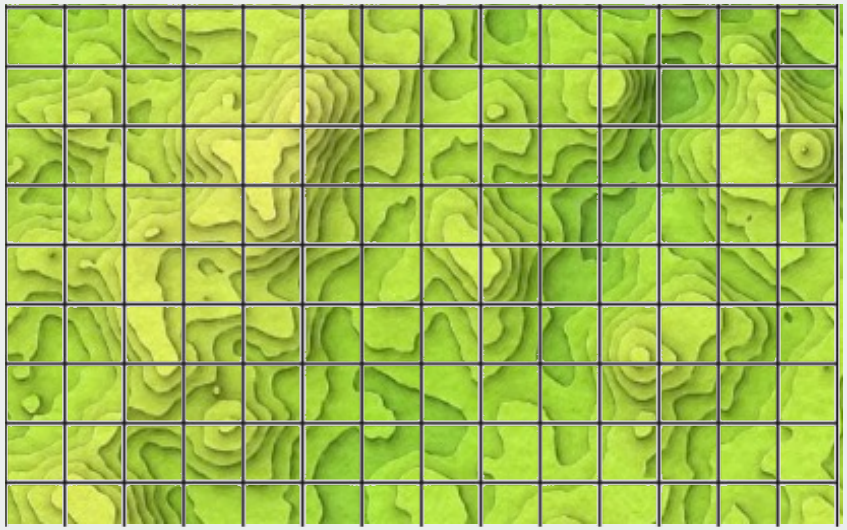
Point 2 $\{x_2, y_2\}$

Point 3 $\{x_3, y_3\}$

Point 4 $\{x_4, y_4\}$

Elevation raster file

z_a	z_b	z_c
z_d	z_e	z_f
z_g	z_h	z_i



Methodology

- Integration

SnooCODE

Imperial College London

3D-Road Json file

Point 1 $\{x_1, y_1, z_1\}$

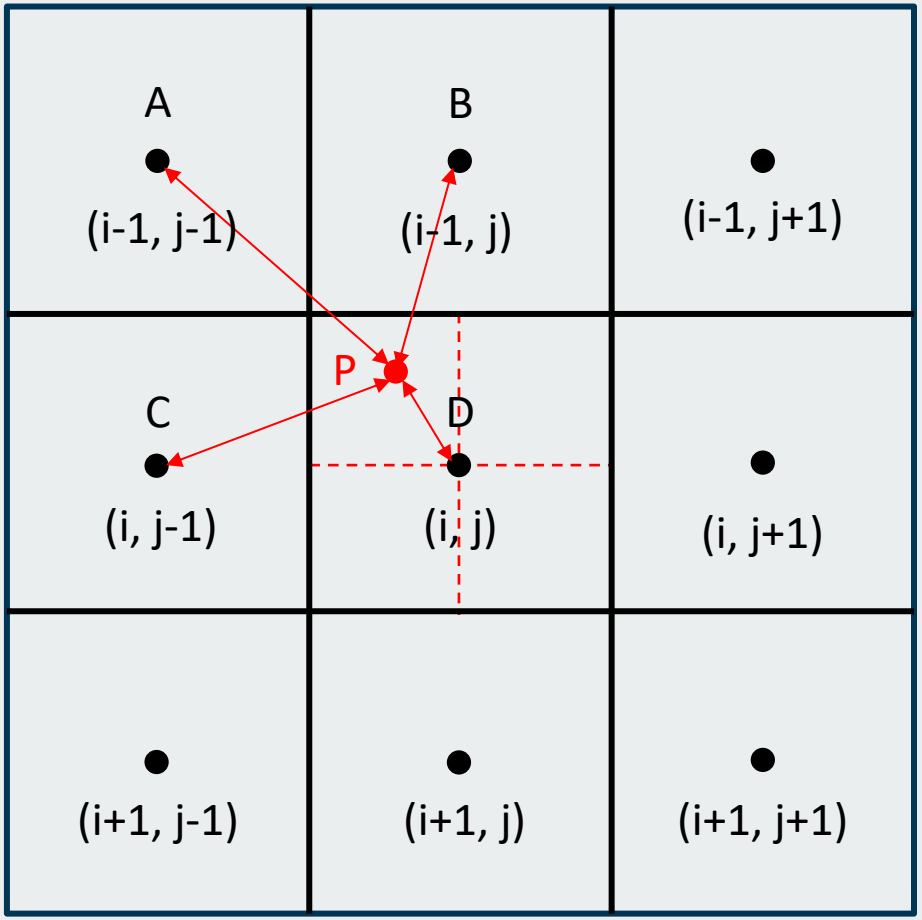
Point 2 $\{x_2, y_2, z_2\}$

Point 3 $\{x_3, y_3, z_3\}$

Point 4 $\{x_4, y_4, z_4\}$

Methodology

- Integration



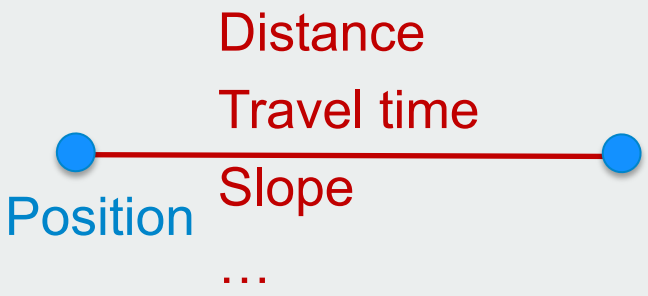
1. Find the grid the point p belongs to.
2. Find the subregion
3. Use the elevation data of four grids surrounding the point p to interpolate the elevation.
4. Calculate the weight for the point p. For example: the weight of point A is: w_A = 1/distance(P, A)
5. Calculate the elevation of P

Methodology

SnooCODE

Imperial College London

- Create Network graph



Node: Store the position info

Edge: Connect nodes

Store attributes

Methodology

SnooCODE

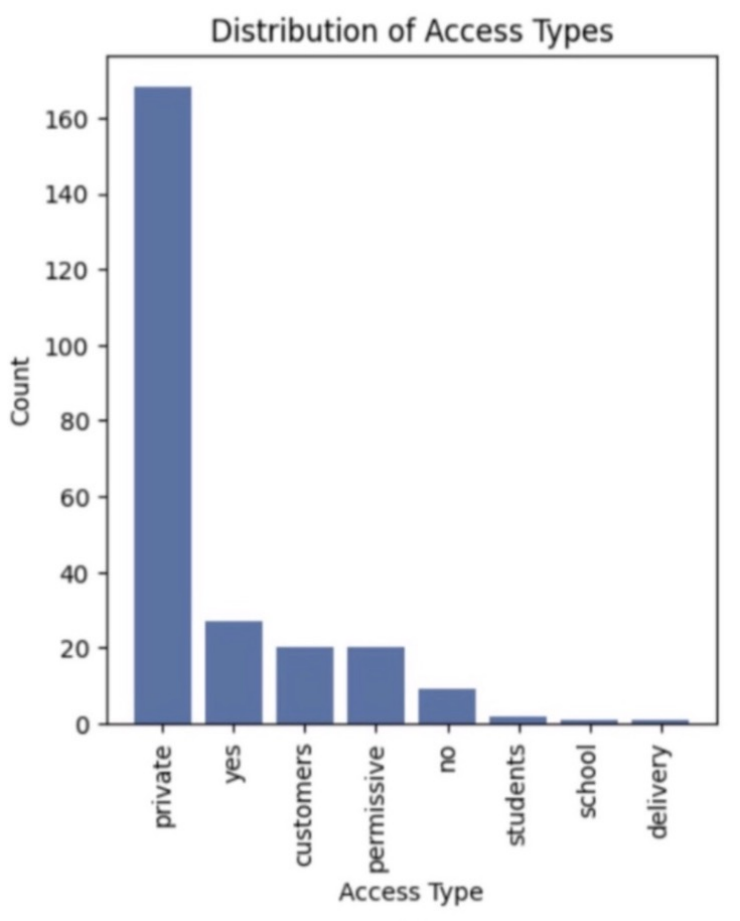
Imperial College London

- Create Network graph

Considering:

1. Road type
2. Access type
3. One-way road

Road type	Speed limit (km/h)
Primary	50
Secondary	50
Tertiary	50
Trunk	50
Residential	20
Motorway	80
Service	20
Unclassified	50

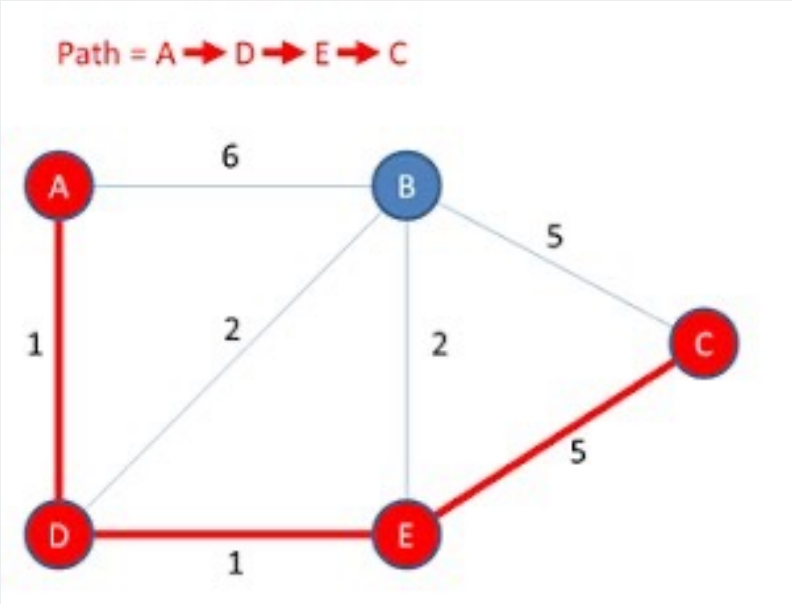


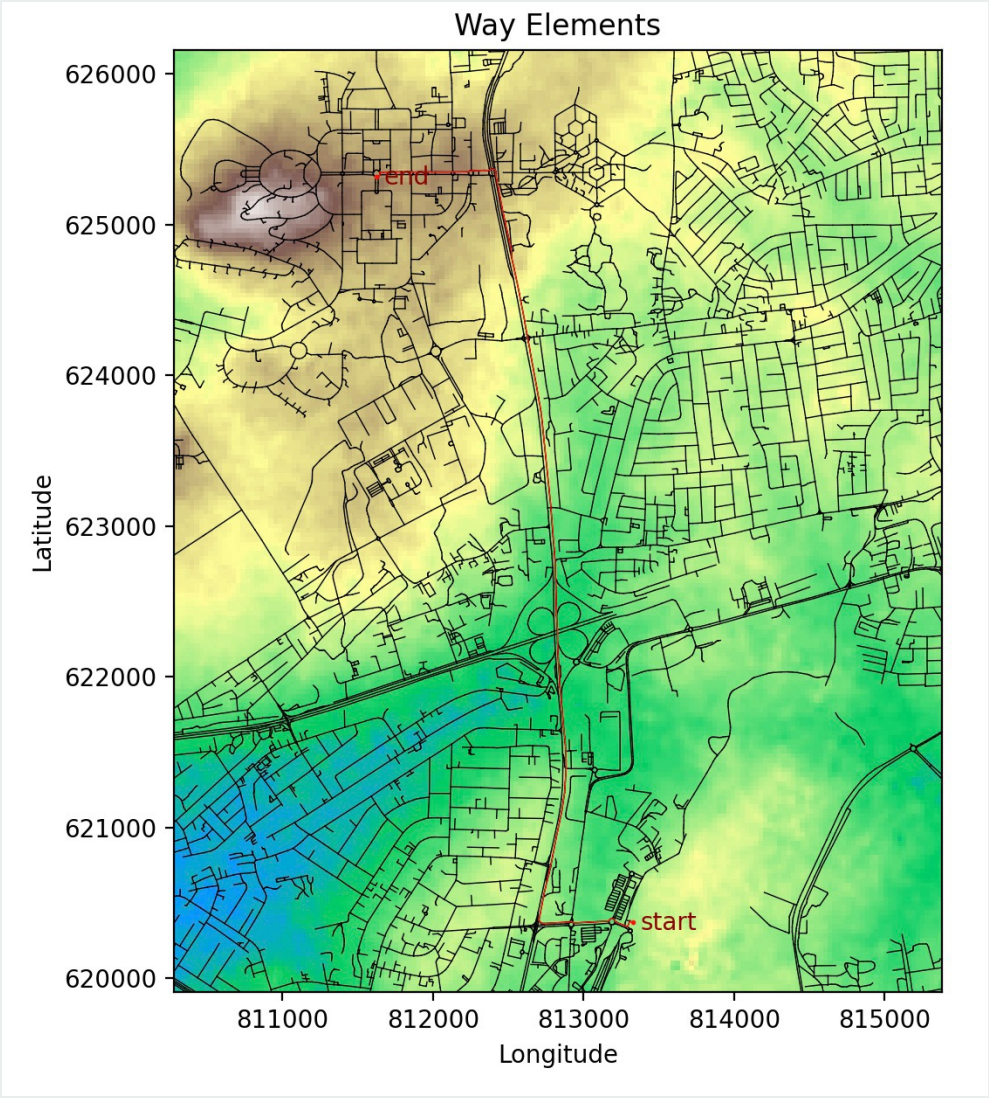
Methodology

- Route planning

Dijkstra algorithm

Find the path with least distance/time





Methodology

- Pointer network

Adjacent Matrix

$$\begin{matrix} 0 & D_{12.} & D_{13} & D_{14} \\ D_{21} & 0 & D_{23} & D_{24} \\ D_{31} & D_{32} & 0 & D_{34} \\ D_{41} & D_{42} & D_{43} & 0 \end{matrix}$$

D_{ij} : Actual distance from *i* to *j*

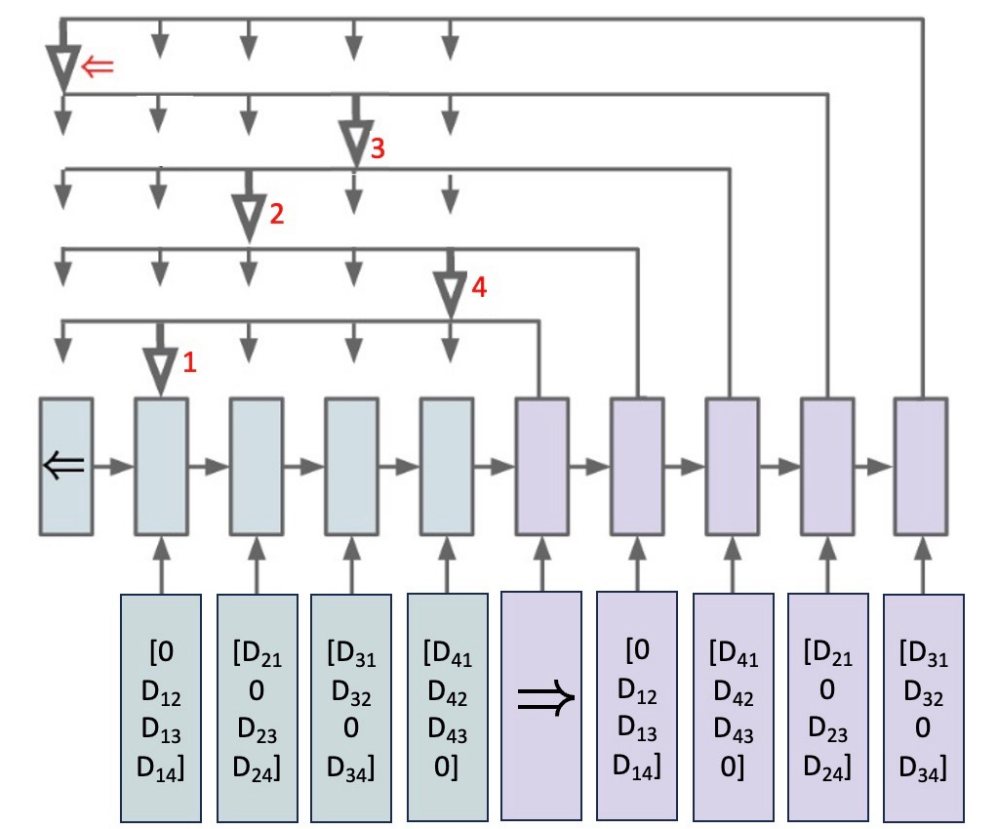
To handle TSP in real-world street:

❌ Simple straight-line distance

✅ Actual distance along the street

Methodology

- Pointer network



Input:
Adjacent Matrix

0	$D_{12.}$	D_{13}	D_{14}
D_{21}	0	D_{23}	D_{24}
D_{31}	D_{32}	0	D_{34}
D_{41}	D_{42}	D_{43}	0

Output:
List of optimised route
[1,4,2,3]

Result

SnooCODE Imperial College London

- Integration and Network

case unit	distance km	distance(Google) km	Relative Error	time minute	time(Google) minute	Relative Error
1	6.55	6.5	0.77%	12.85	13	1.15%
2	6.94	7.7	9.9%	14.32	15	4.5%
3	7.5	7.4	1.4%	13.62	13	4.8%
4	5.4	6.8	20.6%	12.18	22	44.6%
5	9.97	9.7	2.7%	18.89	21	10.04%

100 random paths are tested:

RMSE of distance: 11.95%
RMSE of travel time: 23.73%

Result

- Pointer network

Criterion

Accuracy: The values and order of the output list need to be correct

Mean route length ratio: The ratio between length of predicted route and of optimal route

Accuracy: 67%
Mean route length ratio: 1.02

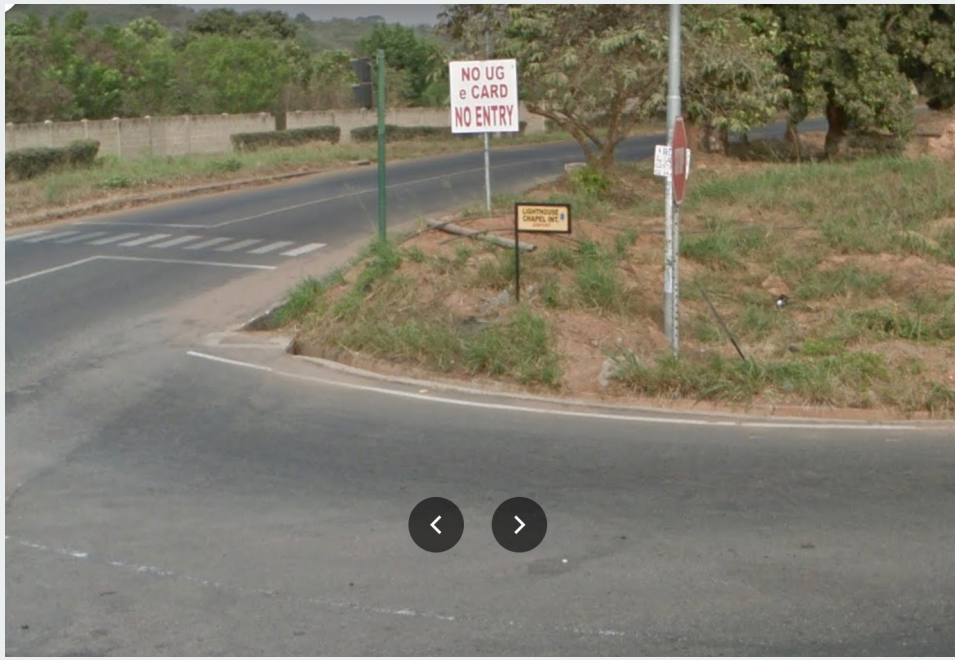
Optimal route: [0 4 1 2 3]

✅ Predicted route: [0 4 1 2 3]

❌ Predicted route: [0 4 1 3 2]

Limitation

- Data quality



Some roads are one-way roads but do not show up in the road data

Limitation

SnooCODE

Imperial College London

- Assumptions

1. The ratio of traffic to capacity is 0.5 for all road segments.
2. No Traffic light
3. Clear road

Error in travel time

Conclusion

SnooCODE Imperial College London

The route planning function:

- 11.95% relative error for finding the shortest path
- 23.73% relative error for finding the most time-saving path compared to Google Maps.

Pointer Network:

- 67.0% of the accuracy of the model
- 1.02 of average route length ratio compared to the optimal route length.

Conclusion

SnooCODE

Imperial College London

Thanks for listening!