PROJECT 15: BKK – STATION VISITING

Vatsa Kush – I3B326

Description

Imagine, that a group of travelling salesman wants to visit all public transport stations in Budapest. They want to accomplish this task as quickly as possible. So they hook up the map of Budapest, locate the stations and start to plan their visits. They can use only public transport to reach the stations except in the first step, where they can start at any station.

In the first planning round they build the network of the stations by connecting that ones where at least one public transport line connects the two stations. On this networks they try to find the optimal starting nodes (stations) from which they can reach all nodes with the minimal number of steps. The steps are counted by the number of stations that they enter or travel through as they use the public transport lines.

Depending on the size of the group (number of salesmen in the group) try to make a station-visiting plan, which results the minimal number of steps, which is needed to visit all the stations for the group! Try to minimize to sum of the steps taken by all members of the group, and try to minimize the number of steps, which is taken by the member who makes to most steps in the group!

Compare your result with a reshuffled network!

The Data

- agency - Basic information about service provider.
- calendar_dates - Service patterns that don't repeat such as for a one-time special event.
- feed_info - Basic information about the BKK service.
- pathways - Contains all paths connected via directional -bidirectional route.
- routes - Contains a list of all the routes with their names, short description and color coding for the online map.
- shapes - Contains collections of GPS coordinates.
- stop_times – contains information about trip_id, stop_id, arrival_time, departure_time and other attributes.
- stops - Contains a list of all the stops with their location(GPS coordinates) and other attributes.
- trips - Contains which shape corresponds to which route.

General Transit Feed Specification (gtfs)

- agency.txt
- calendar_dates.txt
- feed_info.txt
- pathways.txt
- routes.txt ✓
- shapes.txt
- stop_times.txt ✓
- stops.txt ✓
- trips.txt ✓

The data was downloaded from the BKK Website

Data Preprocessing

- The main problem: lots of attributes containing null data.
- Removing the unnecessary columns.

Steps for Merging the data

- Firstly, Merging routes and trips dataset on trip_id.
- Secondly, Merging stops and stop_times dataset on stop_id.
- Lastly, Merging all data obtained from previous two datasets on trip_id.

The Main Data

The main data consists of 5419163 rows and 12 columns. Moreover, contracted_nodes network function and other basic functions helps in removal of the duplicate routes.

	route_id	trip_id	direction_id	route_short_name	route_desc	route_color	stop_id	stop_name	stop_lat	stop_lon	arrival_time	departure_time
0	0078	B864691	0	7G	Cinkotai autóbuszgarázs / Újpalota, Nyírpalota út	009FE3	008569	Cinkotai autóbuszgarázs	47.498194	19.237086	03:50:00	03:50:00
1	0078	B864691	0	7G	Cinkotai autóbuszgarázs / Újpalota, Nyírpalota út	009FE3	043341	György utca	47.530940	19.171561	04:05:00	04:05:00
2	0078	B864691	0	7G	Cinkotai autóbuszgarázs / Újpalota, Nyírpalota út	009FE3	F02984	Rákospalotai határút	47.537837	19.164790	04:08:00	04:08:00
3	0078	B864691	0	7G	Cinkotai autóbuszgarázs / Újpalota, Nyírpalota út	009FE3	F02985	Szentmihályi út	47.537200	19.162956	04:09:00	04:09:00
4	0078	B864691	0	7G	Cinkotai autóbuszgarázs / Újpalota, Nyírpalota út	009FE3	F03148	Erdőkerülő utca	47.541151	19.151023	04:10:00	04:10:00
...	...	...	...	...	...	...	...	...	...	...	...	...
5419158	H5	H80001_4	0	H5	Batthyány tér / Szentendre	8A236C	F04688	Pannóniatelep	47.652488	19.065294	24:57:00	24:57:00
5419159	H5	H80001_4	0	H5	Batthyány tér / Szentendre	8A236C	F04690	Pomáz	47.643188	19.032032	24:52:00	24:53:00
5419160	H5	H80001_4	0	H5	Batthyány tér / Szentendre	8A236C	F04692	Budakalász, Lenfonó	47.621692	19.046912	24:48:00	24:48:00
5419161	H5	H80001_4	0	H5	Batthyány tér / Szentendre	8A236C	F04694	Budakalász	47.616133	19.055013	24:46:00	24:46:00
5419162	H5	H80001_4	0	H5	Batthyány tér / Szentendre	8A236C	F04793	Szentistvántelep	47.629301	19.043159	24:50:00	24:50:00

5419163 rows × 12 columns

M4 Metro

	stop_name
0	Kelenföld vasútállomás
1	Bikás park
2	Újbuda-központ
3	Móricz Zsigmond körtér
4	Szent Gellért tér - Műegyetem
5	Fővám tér
6	Kálvin tér
7	Rákóczi tér
8	II. János Pál pápa tér
9	Keleti pályaudvar

- Visualizing the directional path for the M4 Metro line from Kelenföld vasútállomás to Keleti pályaudvar.
- It consists of ten stations, connects the southwestern Kelenföld vasútállomás located in Buda, and the eastern Keleti pályaudvar in Pest
- **Start Station:** Kelenföld vasútállomás
- **End Station: (végállomás):** Keleti pályaudvar

Algorithm approach

- The three versions of the network is created based on BKK time schedule.
- Sorted the dataset based on arrival time and created unique trip ids for all networks.

Networks:

- Network 1 – consists of the Morning data.
- Network 2 – consists of the Evening data.
- Reshuffled Network – consists of Random data.

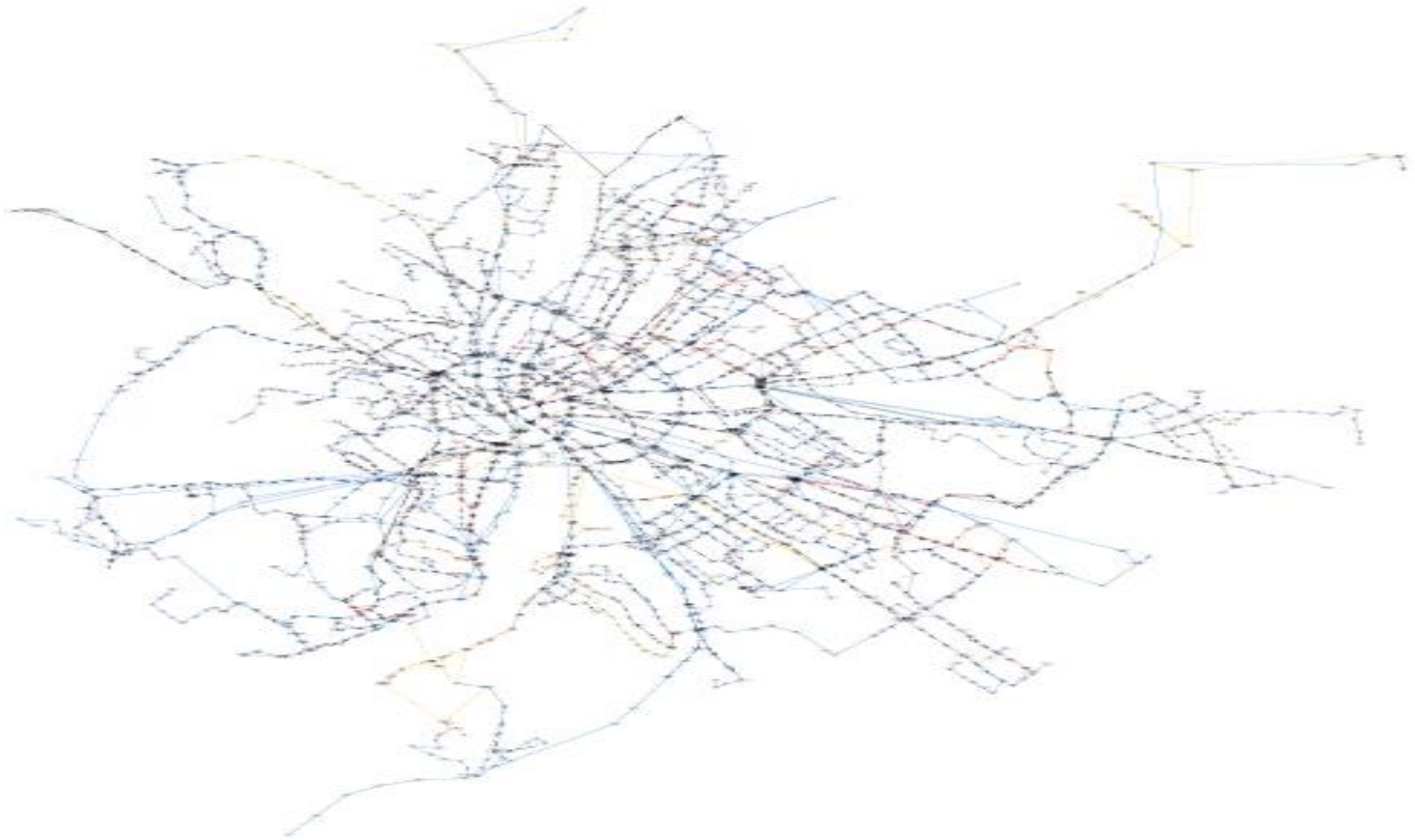
Creation of Networks

- Undirected Graph - It is clearly depicted from the data that the routes are bidirectional, It is a two way relationship, each edge can be traversed in both directions.
- The weights is calculated as the time difference between two stations. i.e, Arrival time and Departure Time.

Group - Selecting service only for one day with different timezones.

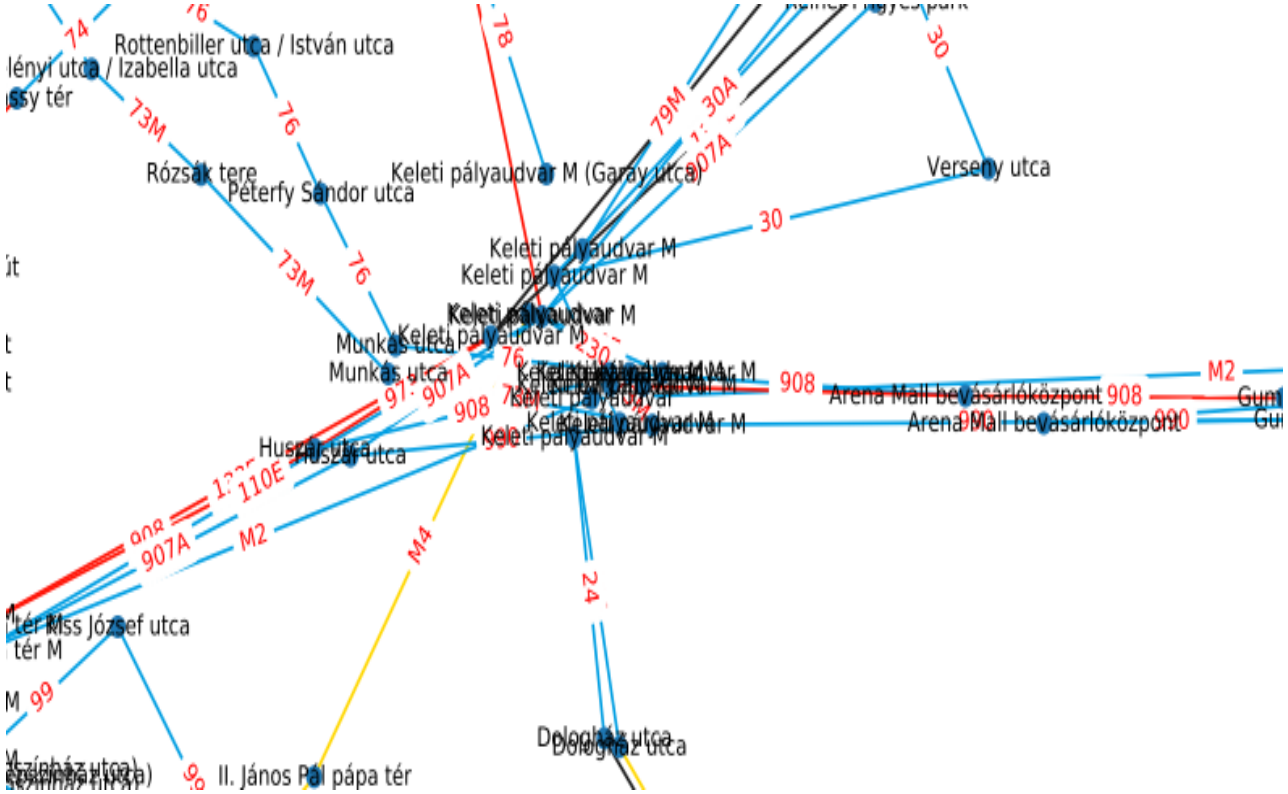
- The generation of network includes station_name_labels, edges, edge_labels, edge_colors, station_positions, node_labels.
- For visualization of Network graphs, Pickle and matplotlib packages were used.
- For each networks, the steps (Number of stations visited) is calculated via shortest path algorithm for a group of travelling salesman.

NETWORK 1 GRAPH



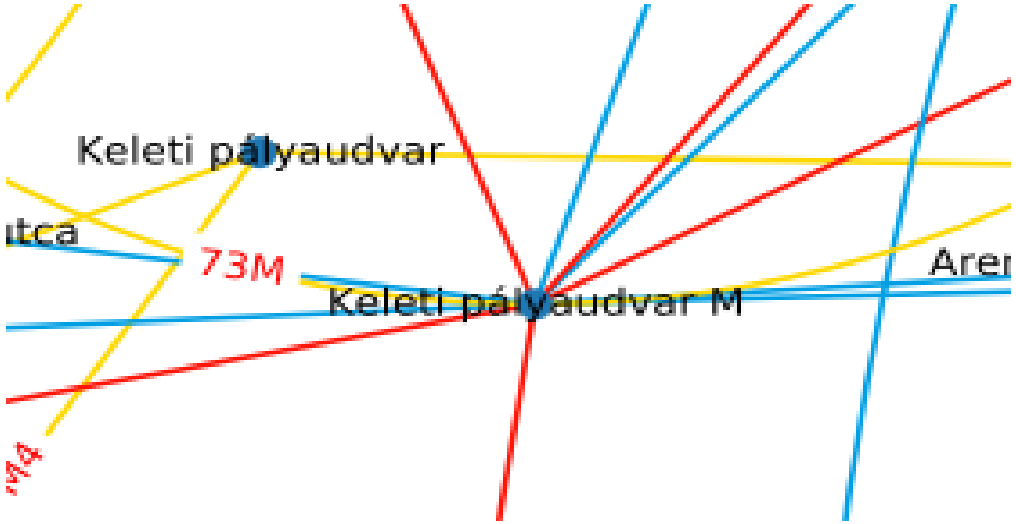
Problem

Data inconsistency: It leads to same station names with multiple times**.**



Solution

- A stop_id column can work but only if the distance between other nodes is zero, if we merge all those stops ids into one, this will leads to less efficient networks, which results in most of nodes in the network have no edges or have edges which is not useful for connecting to other nodes.
- To solve this problem, Firstly, calculate the distance with respect to the longitude and latitude coordinates. And, checks whether a node is close to a set of other nodes.
- Secondly, contracted_nodes – networkx function, It identifies the two nodes as a single node incident to any edge that was incident to the original two nodes.



Results - Nodes

After applying two function results in decrease of nodes approximately 40%.

Networks	Previous Nodes	New Nodes
Network 1	3581	2147
Network 2	4093	2276
Reshuffled Network	4041	2291

Results - Edges

Similarly, The result obtained after applying functions, results in decrease of edges approximately 25 - 30%.

Networks	Previous Edges	New Edges
Network 1	4135	3053
Network 2	4745	3277
Reshuffled Network	4671	3292

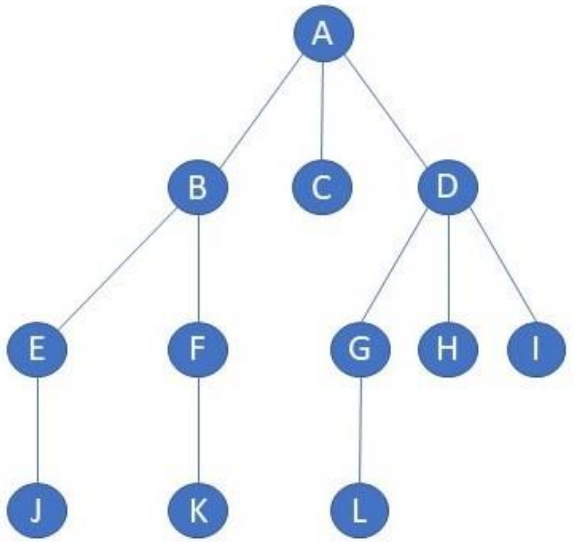
Traversing Graph

Breadth First Search is used for traversing. BFS is a traversing algorithm where we should start traversing from a selected node (source or starting node) and traverse the graph layerwise thus exploring the neighbour nodes (nodes which are directly connected to source node). And, then move towards the next-level neighbour nodes.

Shortest Path : The BKK - Station Visiting

- Networkx package - BFS function is used.
- Weight calculation – The weight of each edge calculated as follows:

Weight = $Arrival\ Time\ -Departure\ Time$



Results

The starting point for the group is Deák Ferenc tér M and It is same for all networks. The results obtained with non filtered and filtered networks is reduction in steps approximately 35 to 40%.

Networks	Previous Steps (Number of Stations)	New Steps (Number of Stations)
Network 1	3318	2102
Network 2	3797	2258
Reshuffled Network	3803	2273

References

- https://bkk.hu/apps/gtfs
- https://networkx.org/
- Mátrai, T., Ábel, M., & Kerényi, L. S. (2015, June). How can a transport model be integrated to the strategic transport planning approach: A case study from Budapest. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)* (pp. 192-199). IEEE.
- Wessel, N., & Farber, S. (2019). On the accuracy of schedule-based GTFS for measuring accessibility. *Journal of Transport and Land Use*, *12*(1), 475-500.
- Kurant, M., Markopoulou, A., & Thiran, P. (2010, September). On the bias of bfs (breadth first search). In *2010 22nd International Teletraffic Congress (ITC 22)* (pp. 1-8). IEEE.
- ELTE Network Science course lectures and practices.

Thank You