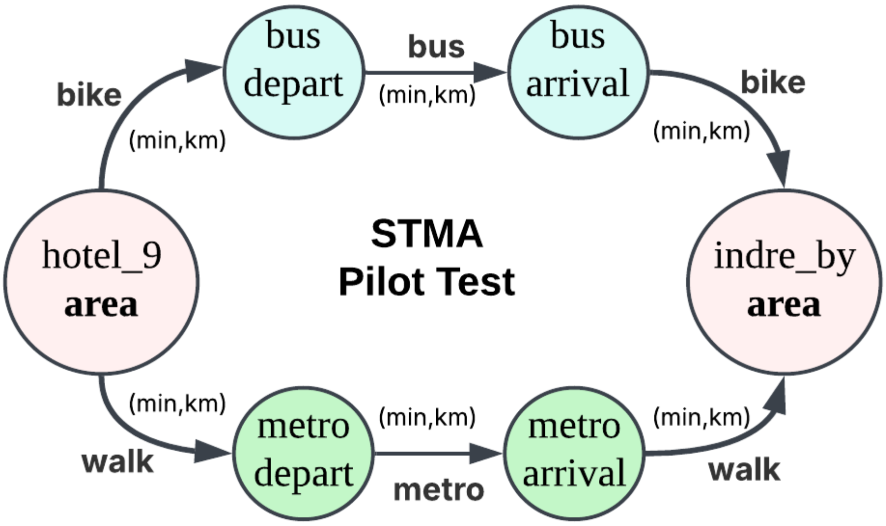
Pilot Test of STMA

Matthew Papesh - April 13th, 2025

Overview:

The pilot was a test to collect sampled locations near hotel 9 and the city center of Copenhagen. The hotel 9 area of locations were taken within a radius of 0.5 kilometers and the city center was taken within a 1.0 kilometer radius. Two trip types were considered: (1) Bike->Bus->Bike (BBB), and (2) Walk->Metro->Walk (WMW) from the hotel 9 area of n=30 locations, to the city center of the same number of locations. Considering BBB, a random location at hotel 9 was routed to the nearest bus stop by biking, and then the bus was taken to the closest bus stop to a random destination location in the city center, from which biking continued to the destination from the bus arrival stop. Considering WMW, a random location at hotel 9 was routed to the nearest metro stop by walking, and then the metro was taken to the nearest stop to a random destination in the city, from which walking continued to the destination from the metro arrival stop. Time in minutes and distance in kilometers was estimated from all connecting sub trips between origin locations near hotel 9, transit stops, and destination locations in the city center. A random trip was found by choosing a sub trip between sampled locations and estimating random trips to the city center from hotel 9 by both BBB and WMW.

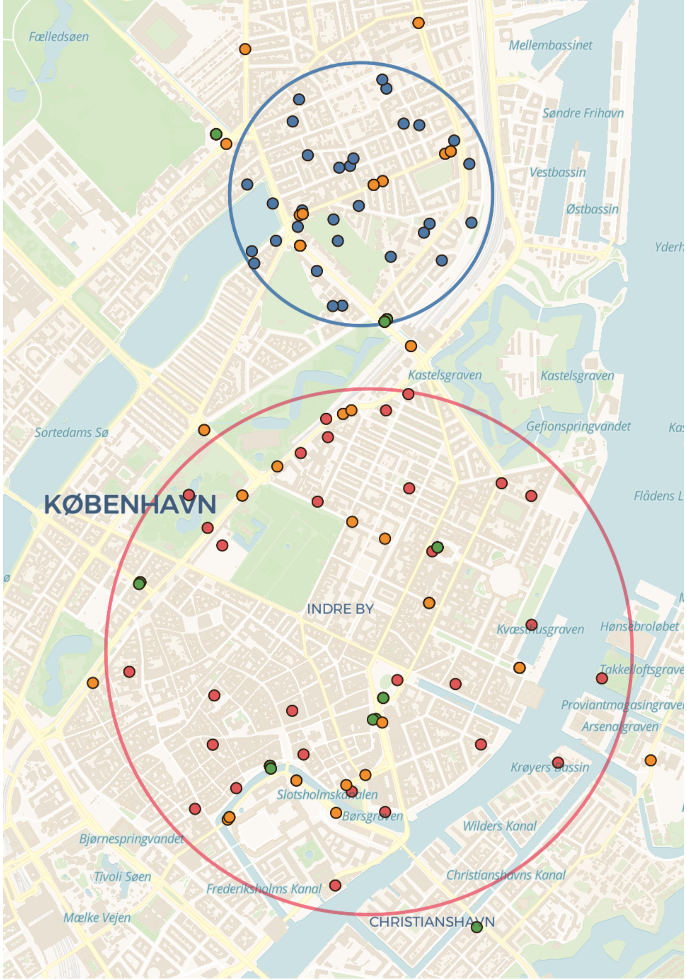


The design of this test can be represented by Individual Mobility Network (IMN). An IMN connects nodes of different areas of sample connections with links that represent the different mobilities. The links represent the different ways to travel between the nodes. The IMN is represented below. The diagram above illustrates the trips taken through each node between the start and end areas; the areas of hotel 9 and indre by respectively. Directed paths are followed from hotel 9 by a specific mobility, either walking or biking, and then transit of either metro or bus, followed by either more walking or biking. Each link in the graph has a sample of sub strips between its two connecting nodes. Each sub trip sample has sampled distance and time data. The sub trip data can be summed across many links to represent the travel across many links and nodes for travel between hotel 9 and indre by. Travel efficiency of distance (km) over time (min) can be computed for each trip sampled. The results of travel efficiency of both mobilities: BBB and WMW can be seen further below.

Network Code:

The network mentioned above can be described by creating `MobilityNode` / `MNode` and `MobilityProfile` / `MProfile` instances. `MNode` for nodes, such as indre_by, and `MProfile` for connections, such as the bus link between departure and arrival stops. Node samples were taken as sizes of n=30 locations each. Bus and metro stops were determined based on whichever ones were closest to locations in either the hotel 9 or indre by node areas. After data is queried for sub trip data, the data is written to the *PILOT_osterport_connections.csv* and *PILOT_osterport_nodes.csv* files. The connections file represents nearly all possible sub trip combinations for trips between hotel 9 and indre by. The node file represents all GPS locations and qualitative data on locations.

The code for this can be seen on the final page below.

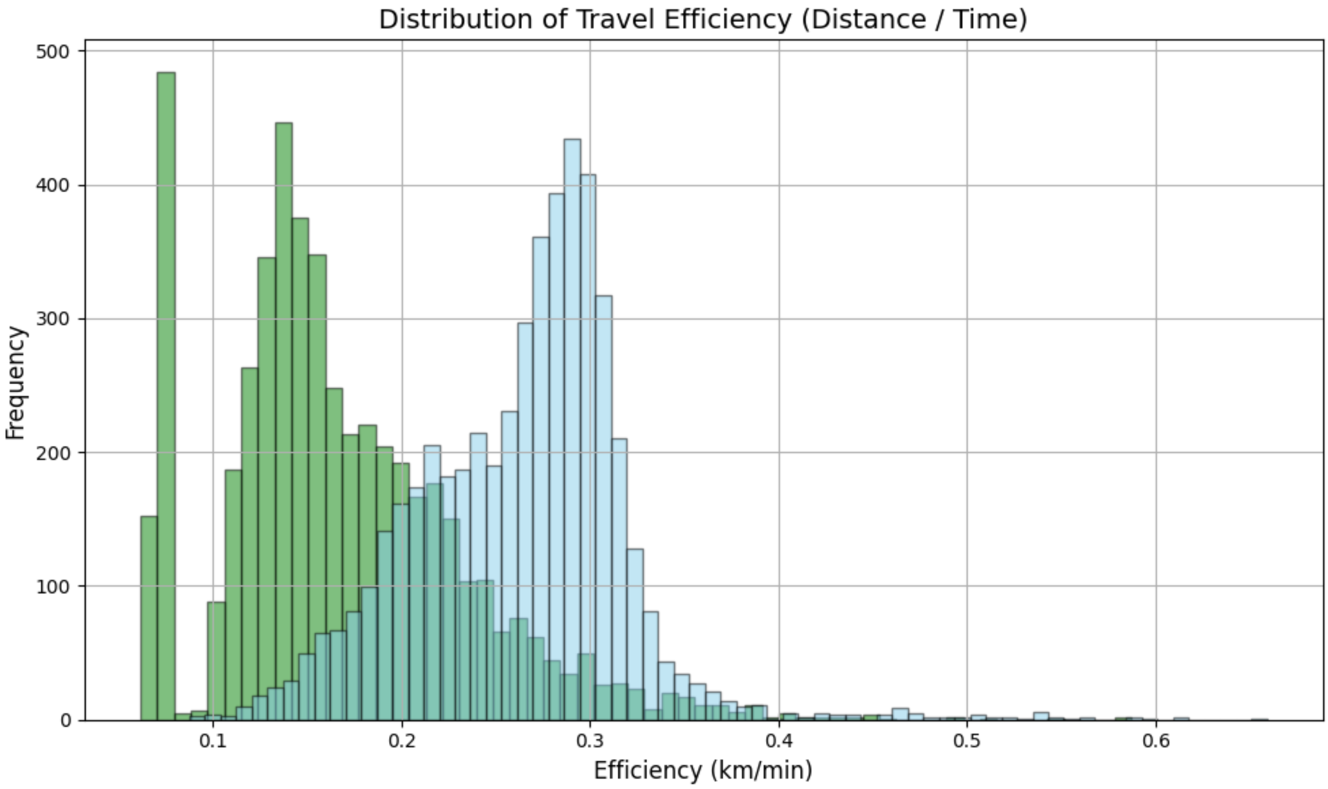


The sampled node locations can be seen above. A light blue circle represent the area of hotel 9 vicinity sampled, and the red circle perimeter represents the same for indre by - city center. The points of the same colors as their corresponding area perimeter are the locations for that node sample. The orange points are the nearby bus stops for departure and arrival in the hotel 9 and indre by areas. The same is true for the green points, but these locations are represent the nearby metro stops. Trips are found by considering a hotel 9 point, traveling to either the closest bus or metro point, and then traveling by transit to the transit stop closest to the destination point in the indre by area.

More bus stops were present than metro stations and stops. Some transit stops were across the canal nearby the indre by area, and a few transit stops were even outside the perimeters of the hotel 9 and indre by areas.Results are posted below.

Results:

Travel efficiency was found by computing the summed distance in kilometers over time in minutes for each trip by method of either BBB or WMW. The results are displayed below. The BBB method is shown in blue and the WMW in green for km/min for 5k possible routes considered by both methods from hotel 9 to the city center.



IMN STMA Python Code Implementation:

```
from internal.mobility_network_base import *
import matplotlib.pyplot as plt
import numpy as np

class OsterportNetwork(MobilityNetworkBase):
    def __init__(self):
        super().__init__(name="osterport")
        # create nodes
        self.hotel_9_node = self.node_30(node_id="hotel_9", gps=
(55.69719963723292, 12.584430781731745), radius=0.5)
        self.osterport_node =self.node_30(node_id="osterport", gps=
(55.69403577472335, 12.586553673755699), radius=1.0)
        self.indre_by_node = self.node_30(node_id="indre_by", gps=
(55.68154580802538, 12.584904309902273), radius=1.0)

        # network connections
        self.metro_conn = self.connection_metro(conn_id="ost_transit",
origin_node=self.hotel_9_node, origin_mobility=self.connection_walk(),
dest_node=self.indre_by_node, dest_mobility=self.connection_walk(),
metro_line=metros.all.value)
        self.bus_conn = self.connection_bus(conn_id="ost",
origin_node=self.hotel_9_node, origin_mobility=self.connection_bike(),
dest_node=self.indre_by_node, dest_mobility=self.connection_bike(),
bus_line=buses.all.value)
        # query data for first run
        #self.QUERY()
        # read data after querying

        self.READ()

        # metro mobilities
        bus_trip_sample = self.sample_trip(n=5000, connections=
[self.bus_conn[0], self.bus_conn[1], self.bus_conn[2]])
        metro_trip_sample = self.sample_trip(n=5000, connections=
[self.metro_conn[0], self.metro_conn[1], self.metro_conn[2]])

        bus_trip_eff_sample = [trip[1]/trip[0] for trip in bus_trip_sample]
        metro_trip_eff_sample = [trip[1]/trip[0] for trip in
metro_trip_sample]

        plt.figure(figsize=(10, 6))
        plt.hist(metro_trip_eff_sample, bins='auto', alpha=0.5,
label='metro', color='green', edgecolor='black')
        plt.hist(bus_trip_eff_sample, bins='auto', alpha=0.5, label='bus',
color='skyblue', edgecolor='black')
        plt.title('Distribution of Travel Efficiency (Distance / Time)',
fontsize=14)
        plt.xlabel('Efficiency (km/min)', fontsize=12)
        plt.ylabel('Frequency', fontsize=12)
        plt.grid(True)
        plt.tight_layout()
        plt.show()

if __name__ == "__main__":
    network = OsterportNetwork()
```