ShipPy

Jordan Lipson, Alfa Budiman, Wesley Howe, David Gayowsky

February 12th, 2023 for uOttaHack 5

Motivation

InnovaPost:

- How do we optimize package deliveries for multiple addresses and trucks?
- Equivalent to traveling salesman: NP hard!
- However… we can solve this!

Project Breakdown

Assumptions and Considerations:

- Time taken to complete route:

  Weather & traffic - use Google maps data!

- Number of mail trucks:

  How many do we have available?

- Computational time:

  Fine tune parameters to affect computational speed.

Project Breakdown

Four major components:

- Recipient Identification
- Recipient Allocation with K-Means Clustering
- Delivery Sequencing with Metropolis Algorithm
- Visualization

Technologies Used

- Code writing and running:
  - Python, Jupyter Notebook
- Route Data and Visualization:
  - Google Maps API, Folium
- Data Management and Evaluation:
  - Numpy, SKLearn, Pandas

Recipient Identification

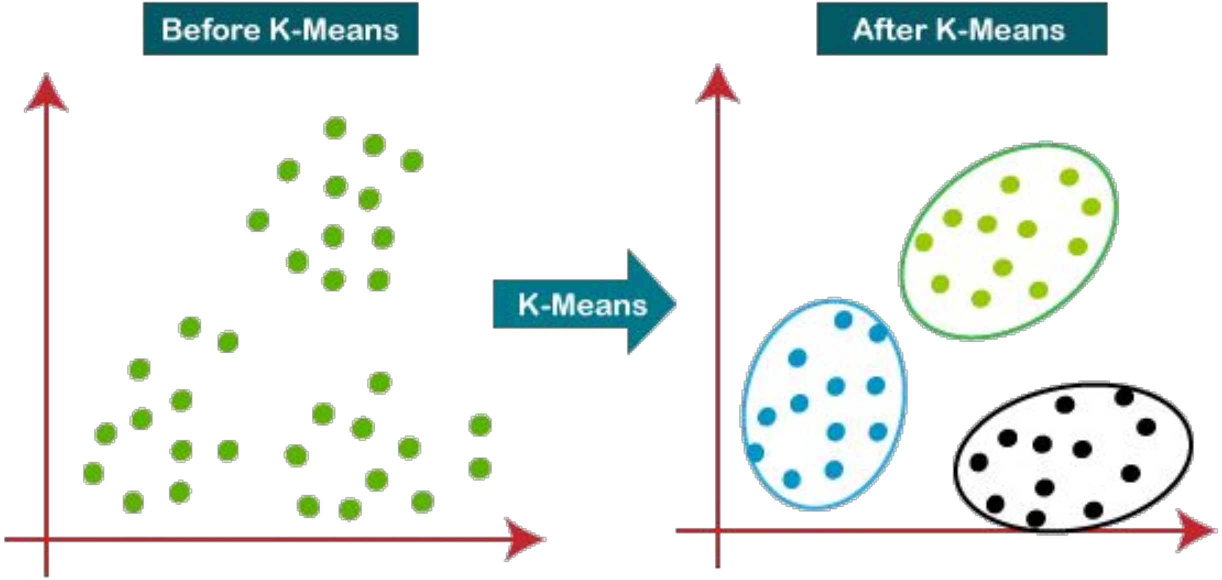
Delivery data - where are we going?

- Addresses from file,
- Google Maps API: retrieve address coordinates,
- Then send to K-Means clustering.

K-Means Clustering

- Groups physical locations of recipients based on proximity to each other,
- Grouped into *n_clusters,* number of delivery trucks,
- Each clusters’ locations correspond to allocated recipients for a delivery truck.

K-Means Clustering

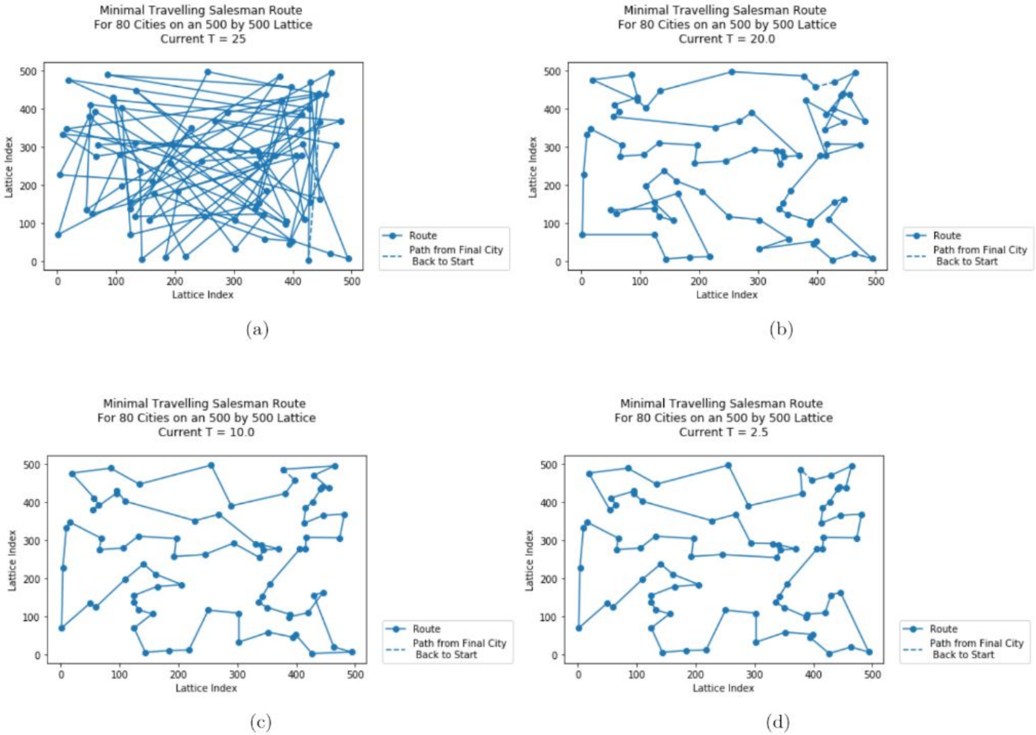


Base Algorithm: Metropolis

Concept: Simulated annealing to solve “traveling salesman problem”.

- Generate trial route,
- Compute difference in route length,
- Accept or reject trial route,
- Vary temperature.

Base Algorithm: Metropolis



(a) (b) (c) (d)

Example: Evolution to traverse 80 cities on 500 x 500 lattice, at:

(a) T = 25 (initial route),

(b) T = 20,

(c) T = 10, and

(d) T = 2.5 (final route).

Base Algorithm: Metropolis

In detail: magic math!

- Generate trial route by swapping order of two randomly chosen destinations e.g.:

$$\{n_a, n_b, n_c\} \rightarrow \{n_b, n_a, n_c\}$$

- If new route is shorter: accept without condition,
- If new route is longer: compute probability at current T:

$$P = e^{-\Delta d/T}$$

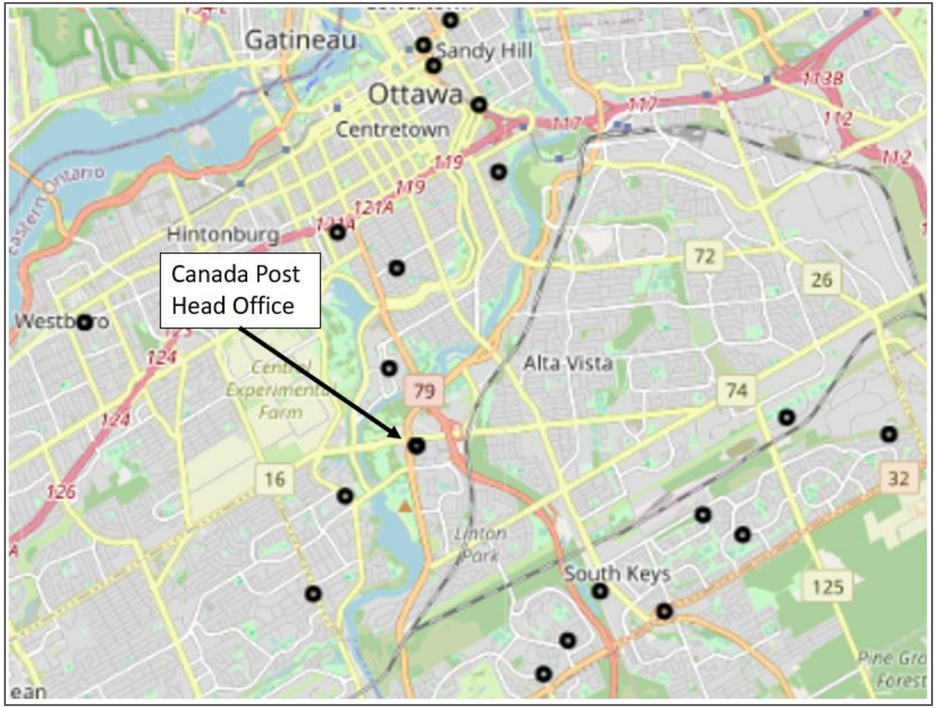
If P > randomly generated $u \in U(0,1)$, accept.

Visualization

Showing our routes on map:

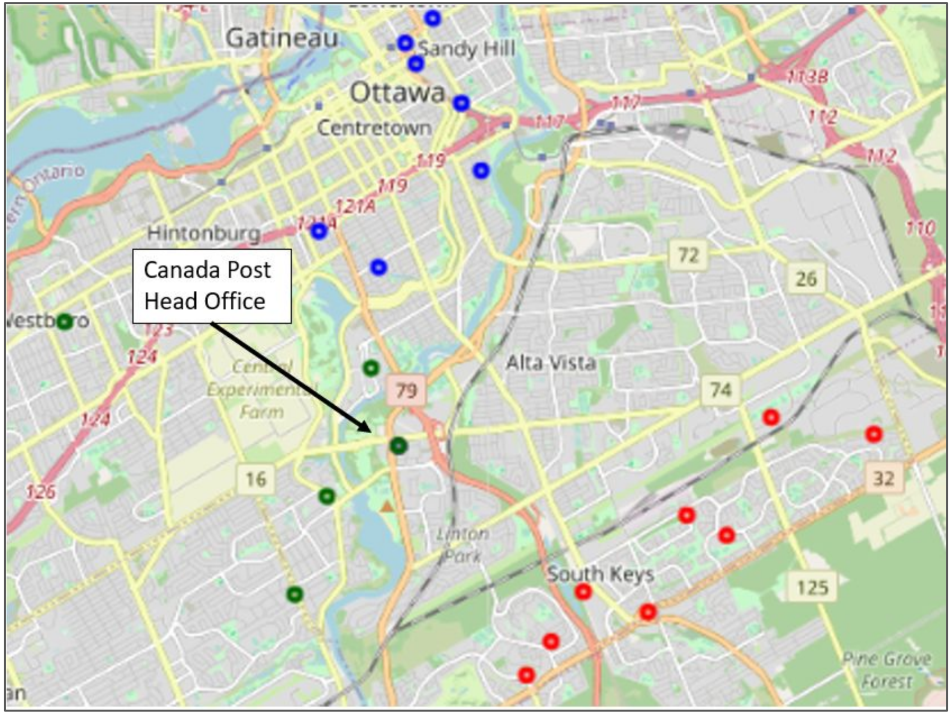
- Folium, interactive map.
- Google Directions API to generate route for each list of waypoints.

All Recipients



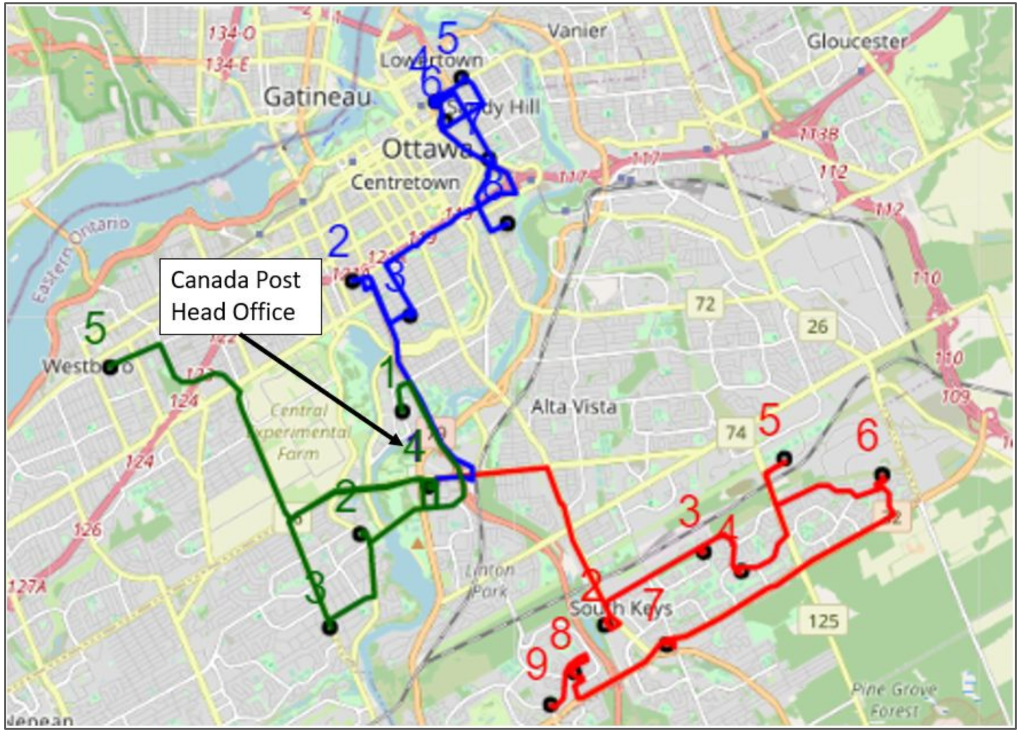
Canada Post Head Office

Assignment of Recipients for 3 Delivery Trucks (KMeans Clustering)

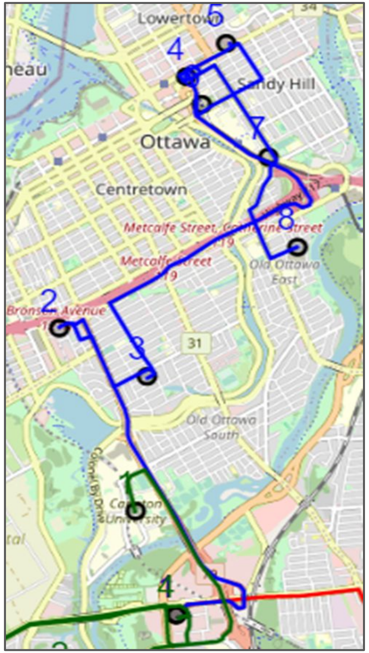


Canada Post Head Office

Route Planning & Recipient Sequencing (Metropolis Algorithm)



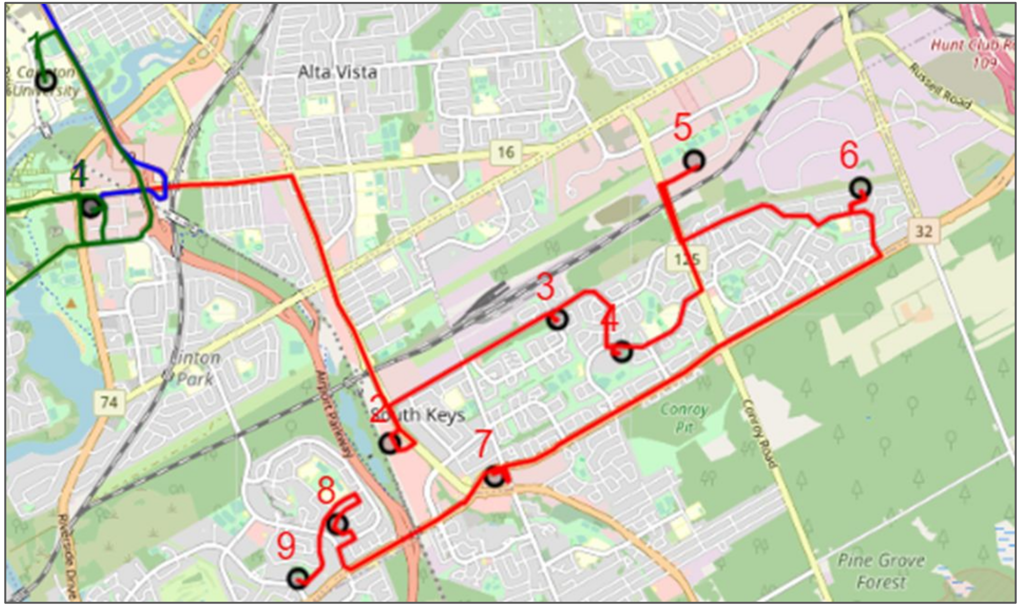
Truck 1 Sequence (Route)



Estimated Travel Time:

56 Minutes

Truck 2 Sequence (Route)

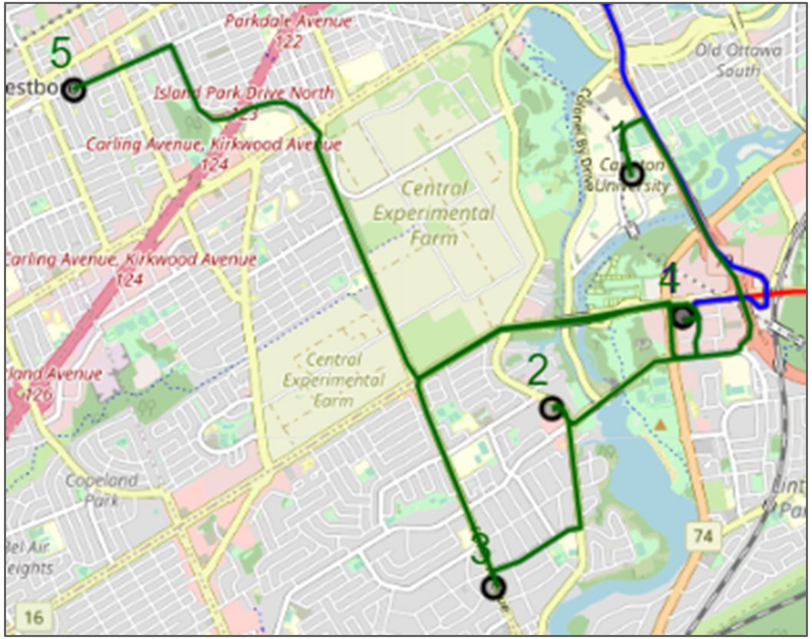


Estimated Travel

Time:

57 Minutes

Truck 3 Sequence (Route)



Estimated Travel

Time:

44 Minutes

Result Analysis

Overall: How did we do?

- Successfully grouped routes for N trucks,
- Algorithm converged to shortest routes based on Google maps data,
- Visualization showed route sequence and address grouping.

Further Work & Improvements

What’s next, and how can we make it better?

- Tuning algorithm & parameters to improve speed,
- Larger-scale test cases,
- Quantum annealing → using physical properties to improve speed with DWave!

Thank you for listening!

ShipPy

Any questions?