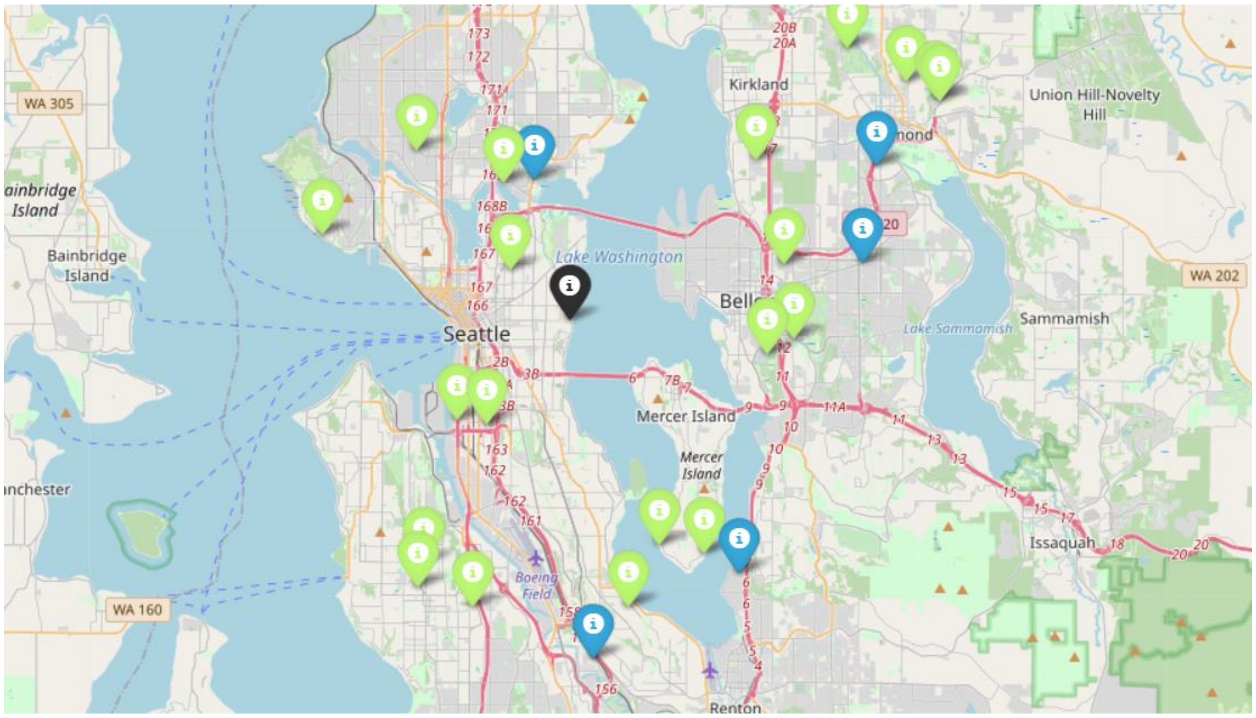
LUNCASU BOGDAN, ANDA BUINOSCHI
11/29/20

TSP with multiple drones

This paper showcases a solution for solving the 'TSP with multiple drones' problem. As this problem is known to be NP-hard, a deterministic approach for this problem would give a far from the optimum solution, a way too much time-consuming solution. For this reason, we are using a Genetic Algorithm to solve our problem.



Data Modelling

- Chromosome = ($permutation$, $drone\_visits$) = (p, v)
- P$ermutation$ p: a permutation of the list [1, . . . , n − 1], appending a 0 at the beginning and the ending of the permutation => p = [0, i1,i2, . . . , in−1, 0]
- Nodes visited by drone = v
    - A list of vehicle ids, each vehicle is mapped to the corresponding node from p
    - The mapping is done by index
        - e.g., the node p[idx] is visited by vehicle v[idx]

Genetic algorithm

For the genetic algorithm we are using the following probabilities:

CHROMOSOME_MUTATION_PROBABILITY=5%

TOUR_MUTATION_PROBABILITY=1%

DRONE_VISIT_MUTATION_PROBABILITY=5%

CROSSOVER_PROBABILITY=20%

The crossover and mutation are done in two steps: one for tour permutations and one for drone visits.

Selection

For the selection part we are using roulette selection where the best chromosomes get a higher chance of being selected.

Crossover

For the crossover operator we are using Partially Matched Crossover (PMX) to generate new permutations for tour and two point cut for drone assignments.

Mutation

For tour mutation we are using Shuffle Index Mutation which swaps two random cities from p(tour). The mutation on vehicle assignments is done by replacing the current vehicle with a different one from existing pool.

Educate

After mutation there is a high chance of having infeasible chromosomes, the educate step uses hill climbing to search for feasible solutions. This operation is done only on the vehicle array(v)

The neighbors are selected by replacing each node assignment with a different drone/truck. We keep trace of the drone's assignments so a drone will not visit two intermediary nodes between starting node i and rendezvous node k

Objective function

The objective function represents the time spent to deliver all parcels by starting from depot and returning to depot.

We split the evaluation by computing the time spent from node i (launching node) to node k (rendezvous node). This is done by taking the maximum amount of time needed for all vehicles to reach node k.

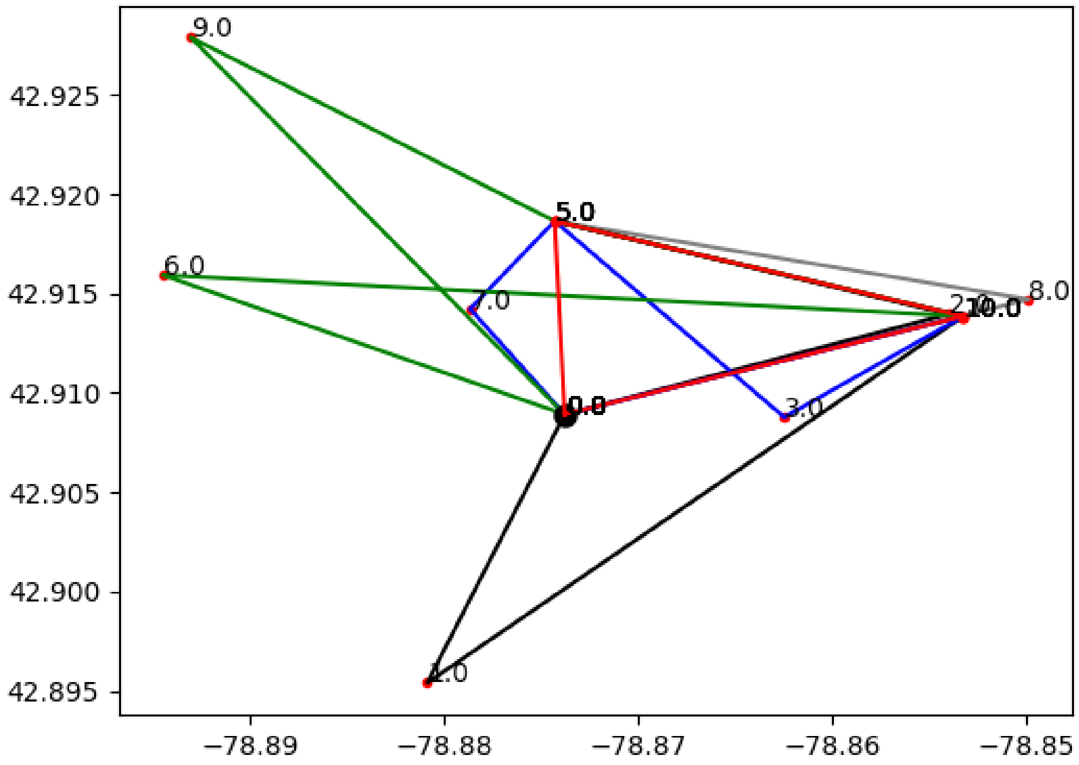
Results

This section showcases the results of the algorithm on three data sets which varies from 10 nodes to 50 nodes.

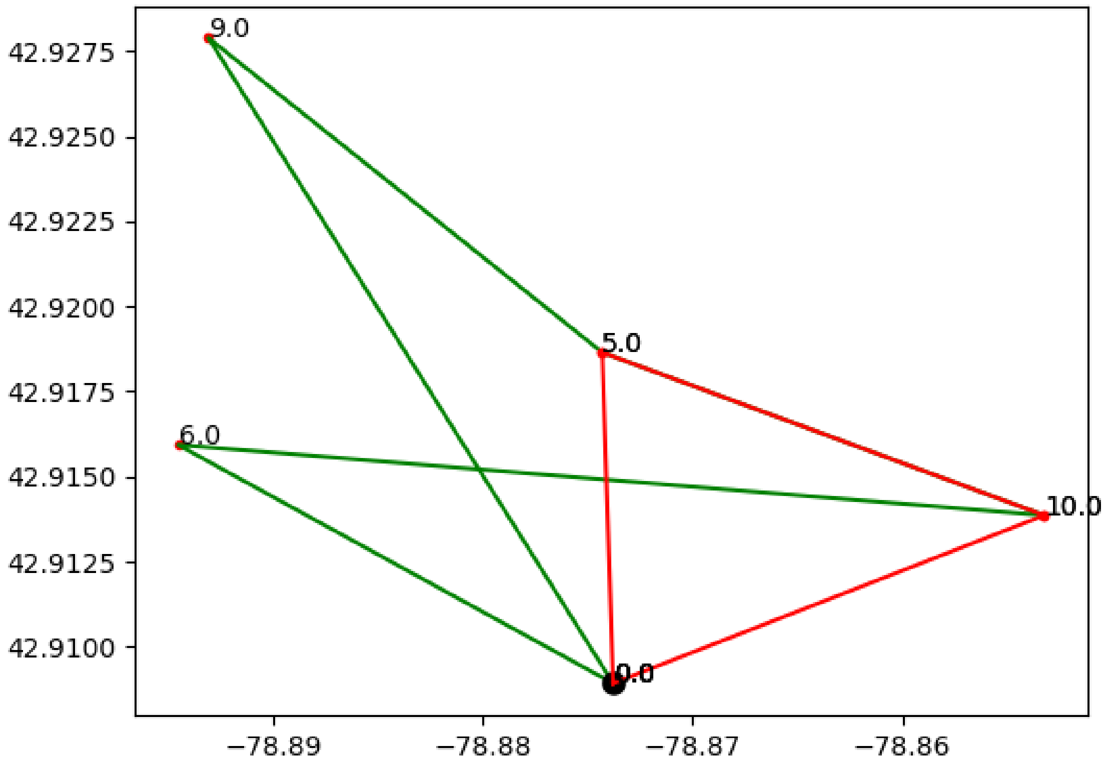
On the smallest dataset we can see below the best solution where the truck starts from node 0 goes to node 10, then to node 5 and back to depot.

```
Tour                0,6,1,10,3,5,8,9,7,2,0
Vehicle assignments 0,1,3, 0,2,0,4,1,2,3,0
```

We can visualize this solution in the figure below where the red lines represent the truck path, and the other colors are used for drone paths.



To better understand the picture above we limited the plot to truck path and a single drone path in the image below.



The score of this solution is 520.44 units of time.

The experiments listed in this paper were done with the following GA configuration:

NUMBER_OF_ITERATIONS=20

POPULATION_SIZE=100

Number of nodes	Number of drones	Best score	Average score	Average execution time
10	4	520.44	598.91	7.84s
25	4			
50	4			

Biography

A Hybrid Genetic Algorithm for the Traveling Salesman Problem with Drone

The Multiple Flying Sidekicks Traveling Salesman Problem (mFSTSP)