Rides Scheduling Problem

After analyzing a lot of data and deciding which rides will satisfy the maximum demand of a transportation company's users. The problem is how to optimally assign buses to the rides. Each bus has a single garage location and operates in a specific time interval. Each ride has location and time for bothride start and end. The problem is to find a plan for each bus, which is the ordered sequence of rides assigned to this bus. The optimality of a solution s is measured by maximizing an objective function defined as follows:

$$F(s) = 60A(s) - 3B(s) - C(s)$$

where:

- $A(s)$ is the number of successfully assigned trips.
- $B(s)$ is the total wasted travel distance (in kilometers) over all buses (i.e. from/to garage or between rides, but not during a ride).
- $C(s)$ the total bus waiting time (in minutes) over all buses (if it arrived before the trip start time).

There are some constraints that should hold in the solution:

- Bus can start the day from his garage any time after 6 AM (**360** minutes from midnight).
- Bus must arrive at his garage by 8 PM (**1200** minutes from midnight).
- No bus can do more than one trip at the same time.
- No ride is assigned to more than one bus.
- The bus must start the trip on time.
- Each bus can do at most **4** trips.

Some notes to consider:

- If a bus arrives before the start time of a trip, it waits until the ride starts.
- A bus can have zero rides assigned to it. In this case, the bus will remain stationary at its garage the whole day and will not contribute to the objective function.
- Some rides can remain unassigned.

You can assume that the travel distance between any two locations is equal to the haversine distance and that all busses have a constant speed of **60 km/h**.
See this link for how to calculate the haversine distance between two LatLng points.

Input Format

- Your code should take the input from a file.
- The first line contains two space-separated integers n, m representing the number of rides and the number of buses , respectively.
- Then n lines follow, where the i^{th} line describes a ride with six space-separated numbers $slat_i$, $slng_i$, st_i, $elat_i$, $etlng_i$, et_i where:
  - $slat_i$, $slng_i$ is the ride start location (LatLong point)
  - $elat_i$, $elng_i$ is the ride end location (LatLong point)
  - st_i is the ride start time
  - et_i is the ride end time

  All times are given as the number of minutes since midnight.
- Then m lines follow, where j^{th} line describes bus j with two space-separated numbers lat_j, lng_j where:
  - lat_j, lng_j is the garage location of bus j (LatLong point)

Output Format

- Your code should produce an output file with the solution.
- The first line should contain one integer c where c is the best solution cost calculated using the objective function.
- Then m lines should follow, the i^{th} of which represents the plan of bus i. Each plan is represented by ordered space-separated integers k_i, $r_{i,1}$, $r_{i,2}$, ...$r_{i,k}$ where:
  - k_i is the number of rides assigned to bus i, and $0 \leq k_i \leq 4$
  - $r_{i,j}$ is the index of the j^{th} ride in the plan of bus i.
  - all $r_{i,j}$ values are unique and $1 \leq r_{i,j} \leq n$

Sample Input

```
5 2
30.02 30.05 480 32.04 31.05 600
30.05 32.02 480 32.03 32.00 600
32.07 32.02 840 30.01 30.00 900
32.04 31.03 660 32.05 32.07 780
32.01 32.02 660 30.05 32.03 780
30.04 30.07
30.01 32.04
```

Sample Output

```
40.4
3 1 4 3
2 2 5
```

Sample Explanation

- Bus 1 is assigned to rides 1, 4 and 3:

- Starts from his garage (30.04, 30.07) at minute 477 (7:57 AM).
- Travels 2.942 km to start ride 1 at (30.02, 30.05). Arrives at 480, and waits 0 minutes for ride start time.
- Finishes ride 1 at (32.04, 31.05) at minute 600.
- Travels 1.885 km to start ride 4 at (32.04, 31.03). Arrives at 602, and waits 58.115 minutes for ride start time.
- Finishes ride 4 at (32.05, 32.07) at minute 780.
- Travels 5.21 km to start ride 3 at (32.07, 32.02). Arrives at 785, and waits 54.79 minutes for ride start time.
- Finishes ride 3 at (30.01, 30.0) at minute 900.
- Travels 7.52 km to his garage.

- Bus 2 is assigned to rides 2 and 5:
  - Starts from his garage (30.01, 32.04) at minute 475 (7:55 AM).
  - Travels 4.847 km to start ride 2 at (30.05, 32.02). Arrives at 480, and waits 0 minutes for ride start time.
  - Finishes ride 2 at (32.03, 32.0) at minute 600.
  - Travels 2.916 km to start ride 5 at (32.01, 32.02). Arrives at 603, and waits 57.084 minutes for ride start time.
  - Finishes ride 5 at (30.05, 32.03) at minute 780.
  - Travels 4.551 km to his garage.

Total assigned rides = 5
Total wasted distance = 29.87 km
Total waiting time = 169.99 minutes
Best objective value = 60 * 5 - 3 * 29.87 - 169.99 = 40.4

Solution approach

The proposed to model to solve this problem is to consider it as a variety of Vehicle Routing Problem VRP (see figure below). Such problem is known as NP-hard: there is no known algorithm for solving this problem in polynomial time, and even small instances may require long time. Therefore researchers use all the time heuristics and metaheuristics algorithms to solve it in acceptable times although the fact that it does not ensure the optimality all the time, but it gives mostly good solutions. As instance genetic algorithm (GA) is the most famous metaheuristic or Iterative local search (ILS).

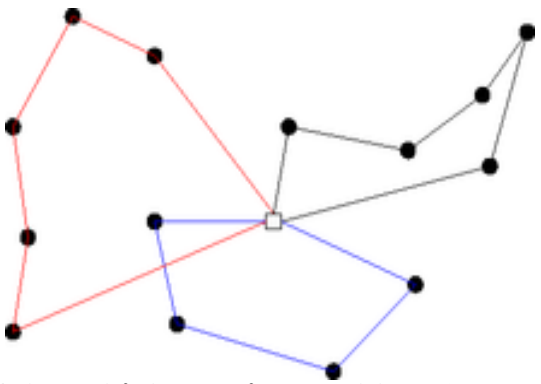
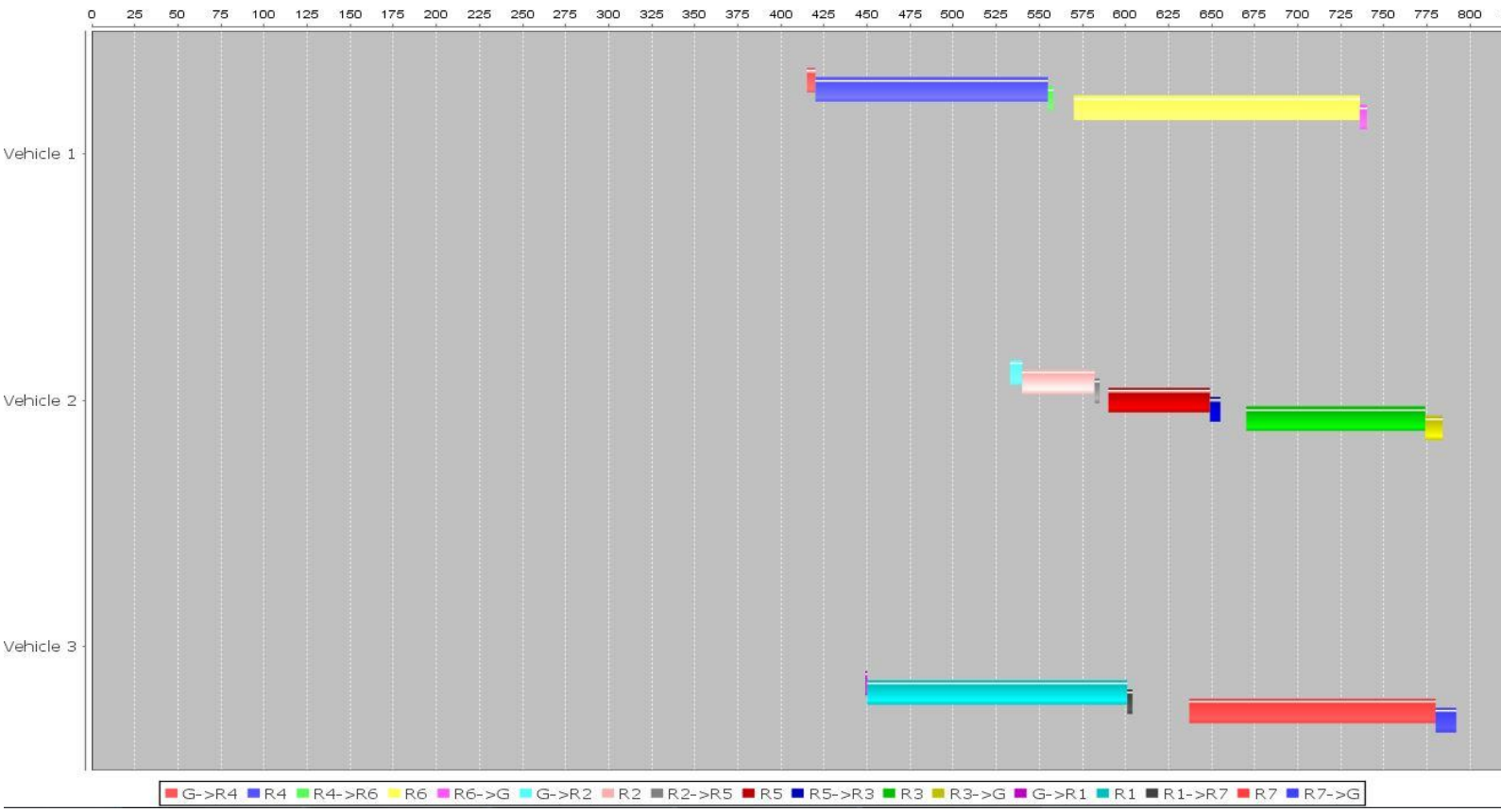


Illustration of the vehicle routing problem (Source: Wikipedia)

I can describe the solution as follow: a set of rides presented by fictitious nodes dispatched in the map (the node represents the whole ride not only the pickup or drop-off location, that's why it is considered as fictitious). Nodes need to be visited by a set of buses or vehicles. The depot in the middle of the illustration represent the garage except that in our model there is many garages, which makes the problem sharing feature with multi depot vehicle routing problem.

The goal of the algorithm is to define the order of all rides (called giant tour) and the rides assignments to vehicles. This last is provided by an adapted Split algorithm that clusters optimally the giant tour to feasible routes by taking into consideration as constraint a maximum number of rides made by one bus. This method is called route-first cluster-second approach for VRPs. Feasible route represent in this case the trip made by every vehicle like if we solve a parallel machine scheduling problem (see figure below).



Buses schedule for s1.in instance (Source: Java code)