ME44206 – Quantitative Methods for Logistics
GROUP ASSIGNMENT Q2
Due Date: December 23, 2022

Consider a vehicle routing problem where you have a number of locations in a region to be visited in order to deliver parcels. Vehicles start and finish at the designated depot. The vehicle capacity is limited in volume. Each location has a time window (between the earliest time and latest time) where they will be available for the delivery of the parcel.

The datasets for this problem consists of multiple rows; each row defines a delivery task with the following information:

- LOC_ID: identification number for a location; the first row (with DEMAND = 0) represent the depot
- XCOORD: x coordinates of the location
- YCOORD: y coordinates of the location
- DEMAND: parcel volume to be delivered at each location (for the depot: zero)
- READYTIME: earliest time for delivery (for the depot: opening time)
- DUETIME: latest time for delivery (for the depot: closing time)
- SERVICETIME: time needed for a delivery at the location

You are provided with several data files, each with different number of locations to be visited. Use the correct file for each question. You can assume Euclidean distances between nodes and 1 distance unit corresponds to 1 time unit.

For this problem multiple questions must be answered. Read each question carefully to identify the relevant constraints for that question. In case a formulation is not clear, please post your question on Brightspace. For specific questions about your own work, please contact us (me44206@tudelft.nl) for clarification *mentioning your group number* and *adding your draft mathematical model (even if it is just a start)*!

In the initial part of the assignment, ignore the capacity of the vehicle and focus on a single vehicle only as in the case of a **TSP problem**. Find a route that visits all the locations in time to return at the depot before closing time.

a. **[20 points]** Formulate the mathematical model for this TSP problem considering the time windows provided for each location with the objective of minimizing the total distance traveled.
b. **[20 points]** Implement the model from question a. in python and solve with gurobi using the data file "**data_small.txt**". You can include a big enough capacity of the single vehicle to represent the TSP. Report your results with the total distance traveled, the locations visited by the vehicle together with the sequence (i.e., route of the vehicle), the time of visit and the load of the vehicle at each location. Also draw a 2D plot of the solution. When referring to the *results* in other parts of the assignment these mentioned outputs should be the standard minimum and in each part there will be additional results added to this list.
Tip: spent some time on formatting the outputs of your implementation, so that it shows not only the objective value but also gives insight in the decisions! This is very useful for debugging during implementation, verification and reporting of results. You can also make use of "numpy" and "matplotlib.pyplot" in your implementation.

Now consider that you can have multiple vehicles which have a maximum capacity, i.e., you are moving to a **capacitated VRP** with time windows.

c. **[10 points]** Update the formulation indicating any changes in the model (sets, parameters, variables, objective function, constraints etc.).
d. **[5 points]** Implement the model in part c and <u>experiment with the following three cases</u> using "**data_small.txt**":
    1. 1 vehicle with a capacity of 120
    2. 2 vehicles with a capacity of 60 each
    3. 6 vehicles with a capacity of 20 each

    Evaluate the results of these three cases in comparison to each other and in comparison to part b, provide your insights on the differences.

Now, consider that you can have **split deliveries** in the capacitated VRP such that the locations can be visited multiple times to complete the delivery task.

e. **[10 points]** Update the formulation indicating any changes in the model (sets, parameters, variables, objective function, constraints etc.).
f. **[5 points]** Implement the model in part c and <u>experiment with the three cases in part d</u> using "**data_small.txt**". Discuss the differences between the VRP with split deliveries to the one in part c (where it is not allowed) considering the three cases.
g. **[5 points]** Now run your implementation with "**data_large.txt**" with the assumption that the capacity of vehicles is 200 for each. How many vehicles you would need? Does the model make use of the split delivery option? Why / Why not? Report the results as usual with the total distance traveled and the route of the vehicles.
h. **[5 points]** Now run your implementation with "**data_large.txt**" with the assumption that the capacity of vehicles is 100. Are you able to get results in 30 min? Limit the computational time by 30 min and <u>experiment with different capacities</u>. Discuss the computational challenge associated with this case explaining the reasons behind. Report the number of used vehicles, the use of split deliveries together with the usual outputs of the total distance traveled and the routes of the vehicles.

Now, building upon the capacitated VRP with split deliveries, consider that you have a mix fleet of vehicles with different capacities, i.e., **heterogeneous fleet**. Those different vehicles are associated with different **fixed costs** such that there is a one-time payment for each vehicle according to the capacity of that vehicle.

i. **[10 points]** Update the formulation with the objective of both minimizing the total distance travelled and the total fixed costs indicating any changes in the model (sets, parameters, variables, objective function, constraints etc.) Discuss your approach for combining the two objective functions and how you formulate it in your mathematical model.
j. **[10 points]** Implement the model in part i. Note that you are required to have the objective function explicitly implemented rather than using the *objectiveN* command of Gurobi. Run your implementation with "**data_large.txt**":
    1. 25 vehicles: 10 vehicles with a capacity of 20 and a fixed cost of 100 euros, 15 vehicles with a capacity of 100 and a fixed cost of 4000 euros. You can consider that those small vehicles are like cargo bikes that do not cost you much to have them available and the bigger ones are vans that cost much more. Report the results as usual with the total distance traveled and the route of the vehicles, the number of used vehicles as well as the use of split

delivery option together with your insights. Compare the total distance traveled in this part to that of part h (with a capacity of 100 for each of the vehicles in the fleet) and discuss the differences, if any, together with your insights.

2. Experiment with different relative costs of the vehicles by changing the cost of the small vehicles using “**data_large.txt**”. Justify the values you choose for experimenting with the costs. Provide your insights on the impact of the costs on the results (number of used vehicles, the use of split delivery, total distance traveled, routes of vehicles). Consider the tipping point for the cost of small vehicles when they become no longer beneficial to have.

Submission

Please submit a report (not more than 15 pages) and your .py files for the implementation for the versions in b, d, f, j separately. Keep the reporting of your mathematical models compact; use short (but clear) definitions and avoid formula derivations or explaining text (if that is needed, report that in a separate section, not in the section with the mathematical model itself). You can still refer to the template we provided for the airplane cargo example. On the first page of your report include your group number and members’ names.

Notes

- Solve the problems to optimality if the computational time is within 30 min and report the computational time. When it goes beyond 30 min set a time limit of 30 min (=1800 s) in order to get solutions (by using “model.Params.timeLimit = 1800”). If you do that report the optimality gap together with your results.
- In order to deal with the time dimension of these routing problems, you may define a variable for the start time of the service at each node. This variable then facilitates the definition of related constraints.
- Some common VRP formulations repeat the depot (node 0) as the last node (node n+1). Such modifications are not obligatory but can be helpful. In any case, **you MUST use the input file as it is** and you can implement such things after reading datafile in python within your python implementation!
- Include a **contribution statement** where for each of the group members you list down what was the contribution from this member, e.g., data handling, mathematical formulation for a specific part, python implementation, discussion of results, report writing etc. The idea is that you as a group are all aware and transparent about the share of the work. We may ask questions to verify this contribution statement as part of the grading.