Individual Assignment

For: Data Analysis and Programming for Operations Management 2021-2022.2a

Master Technology & Operations Management, Faculty of Economics & Business

University of Groningen

Content:

i. **Description of work**
ii. **Appendix 1 – examples of graphical output**
iii. **Appendix 2 – mathematical programming models for the optimization problem**

I. Description of work

Summary of requirements

- This is an individual assignment and the grade given will be your course grade.
- The deadline for this assignment is Monday April 4, 2022 (noon, 12:00).
- An oral exam is part of the assessment.
- The oral exams will be scheduled in the exam weeks after the deadline.
- You can enroll for a time slot for the oral exam through Nestor. It will be announced on Nestor when the enrollment opens, before the exam weeks will start.
- The text part of the report should not exceed 3 pages (this text part is excluding the pictures, the tables, and the code).

Introduction

The idea for this individual assignment is that you will (pretend to) play the role of a data analyst and operations consultant. The (imagined) company that you advise is a business developing the infrastructure for intra-city end-point delivery via sustainable transport (i.e. supporting the whole range of small electric vehicles, from e-bikes to small e-vans). For this particular project, they are planning to deploy a network of facilities aimed to simplify the delivery task for potentially all the fresh grocery delivery companies that operate sustainable delivery in the city of Groningen. This network – its structure being the goal of your work - will consists of refrigerated lockers placed at strategic locations in the city.

The deliverable of the assignment is a concise report that includes the choices you made and the activities that you carried over in your assignment. The report should not be just a collection of notes; it needs to have a proper flow (introduction, conclusions, graphs, etc.). The final source code needs to be properly modularized, commented, and formatted - and included in the appendix of this report.

Submit the report on Nestor – this implying a plagiarism check.

Also, send your code as .py files attached to an email to the lecturer (n.b.szirbik@rug.nl).

The assessment criteria for the assignment are the following:

- Solid reasoning explaining your modeling choices, for the organization of your data, and the LP models
- Appropriate use of the learned programming techniques and Python coding skills
- Use of the Gurobi and Elastic packages
- Quality of the report (layout, narrative, structure)
- Quality of the oral defense

By doing this assignment, you will show that you have a beginner's understanding of programming related areas that were practiced in the previous weeks (e.g. CSV files, data structures, Elasticsearch facilities, LP in Gurobi). Note that many different possible solutions exist for the programming problems at hand. The techniques and libraries used during the practicals provide guidelines to find your answers for various parts of this assignment, yet you are free to explore other methods and use additional libraries.

Please consider that working together and helping each other is understandable and it is allowed, but also be aware that the assignment is individual. Hence, plagiarism is not allowed. The work has to be yours, and you need to make your own choices, decisions, visualizations, be in total command of the material, and be able to defend it at the oral exam.

During the oral exam, we will test your understanding of your own report and code as well as your understanding of the DAPOM topics in general. The questions at the oral exam largely depend on your report and code. You must be able to explain the goals and functionality of the libraries you used. You are, however, not expected to memorize all parameters of Gurobi or Elasticsearch libraries. Typical examples of questions are:

- Where in your program do you determine the distance between two postcodes?
- Could you explain your choice of coding on the top part of page 8 of your report?
- How would your optimization model change if there were a lower bound on the number of hubs to be installed in each district?
- *Teacher pointing at a piece of code.* Could you explain what this piece of code is doing?

The oral exam will take half an hour. The exam will be done with two students simultaneously to entice discussion.

Case description

In this assignment, you will consult a delivery-support infrastructure-building company in the city of Groningen. The company is currently providing support for ready-to-cook-items delivery services via electric vehicles from various grocery shops. Multiple delivery companies can use it in the same time, on a first come, first served basis. Real-time information about the available space in the lockers placed in strategically chosen pickup points for fresh items is to be provided to the delivery companies, which will manage the communication with the end-customers. The deliverer can decide in advance to deliver to the door, or, if the customer agrees, to put the grocery bag in the cooled locker – for the end customer to pick it up later. These lockers are relatively small (10-30 slots for

separate grocery bags which can be accessed individually by deliverers and customers) and are fully automatic (the deliverer and, customer has code-based access to top and respectively pick the grocery bag). The tradeoff for a locker is its space vs. its cost, meaning that the utilization rate (how full the locker is on average) shall be high.

The use of locker-based pickup points has several advantages. For the delivery companies, it leads to economies of scale in deliveries as the time and effort to drop off multiple grocery bags to a locker-based pickup point is much smaller than individual drop-offs at each customer. For the customers, it provides flexibility given by asynchronicity, as it is no longer necessary to stay at home for a parcel delivery; albeit at the expense of self-pickups from a local locker pickup point—which is often only a few hundred meters away.

The success of the company's venture depends on the selection of the strategic points where locker pickup points are (and how big they are) such that a sizable portion of total deliveries is made via voluntary self-pickups. Based on earlier experience the company management has decided to proceed with the following basic principles:

- It would be a tedious and expensive task to operate a dense and tightly interconnected network of locker-based pickup points. Hence, the city should be partitioned into a number of districts, with distinct geographical boundaries, each with its own small network of lockers.
- In order to be economically viable, it is necessary to shift at least 25% of deliveries to self-pickups in each district. The aim should be to reach this objective with the minimum number of pickup points possible.

The company has recently run a market analysis among its customers to get a better understanding of their perception regarding the concept of locker-based pickup points. The results of this analysis reveal the following:

- Customers may opt for self-pickups up to 35% percent of their parcels—depending on how far the designated pickup point is to their addresses. It is estimated that the percentage will drop by 5% for each additional minute of cycling time. For instance, if the pickup point is at 2 minutes of cycling distance from a customer's address, then she will use it for $35 - 2 * 5 = 25\%$ of her parcels, and, but if it is further than 7 minutes she will not use it at all.
- The lockers are in fact slightly more complicated and secure fridges. For simplicity, we consider that we have only one size, and this can provide for 24 pickups in a day.

As a consultant, your duty is to identify by data analysis and quantitative methods the strategic points in a city-wide network of locker-based pickup points. In consultation with the logistics manager, you have decided to do this based on the following building blocks:

- The districts will be defined by their distinct 4-digit postcodes. There are around 30 such postcodes in Groningen.
- It is often hard and also unnecessary to consider each and every possible delivery address. To facilitate the analysis, you will consider 6-digit postcodes rather than individual addresses (e.g. as if all individual addresses that share the same postcode are aggregated into a single

address). This is a very reasonable approximation because these postcodes cover addresses in very close proximity, encompassing each like 30 individual addresses on average.

- The analysis will require a forecast for the number of grocery bags that the delivery companies will deliver per day at each 6-digit postcode. To that end, you will use the forecasted numbers based on data collected from the last years and estimations derived from the market research.

Available data

`forecasts_2023.csv`: All forecasted grocery bag deliveries for the entire 2023. Each delivery contains a forecasted date/time based on previous deliveries and the postcode of the destination address. Please note that these data are randomly generated for reasons of privacy. It is totally fictitious and not even based on real data.

`geo_coordinate_per_each_location.csv`: Coordinates of all postcodes in Groningen.

`distances_between_locations.csv`: The traveling time (in seconds) and distance (in meters) in between all pairs of postcodes in Groningen. It is generated with the Open Source Routing Machine using a bicycle profile (http://project-osrm.org/).

Before proceeding with your analysis, examine the structure of the data files and make sense of what each column and/or row stand for. In principle all three files can be read directly in Python by various csv readers. However, especially for the files `forecasts_2023.csv` and `distances_between_locations.csv`, the use of Elasticsearch is mandatory in this assignment. This is a critical aspect because it is assumed that the company will have to repeat the same analysis with larger cities and data set following this pilot in Groningen. Keep in mind that the distance file in particular will take some time to ingest in the corresponding Elasticsearch index. On a regular laptop, it should take approximately half an hour. If it takes longer and you do not trust whether the import is actually running, use Kibana to check the count, or from a browser go to

`http://localhost:9200/<insert-your-index-name-here>/_count`

(refresh regularly the browser page). The count should increase. The total count for the postcode file should be around 30 million. The code you have to use for loading the file data is provided by us in the loaders.py file, on Nestor, Course Documents/Practicals Week 4/Practical B, item "faster loader module". Do not try to use the elasticsearch-loader from pypi (https://pypi.org/project/elasticsearch-loader/) because this does not work with the new elasticsearch versions.

An overview of the analysis

In the following, you will find a detailed overview of each step you will conduct in your analysis—where larger area 4- and smaller area 6-digit postcodes are referred to as larger *districts* and smaller *locations*, respectively.

Step 1—Analyze locations and districts

The first step involves a descriptive analysis of locations and districts.

1. Load all locations with their respective geo positioning data, and load the forecasted deliveries for 2023 (each line represents a forecasted grocery bad to be delivered at the time stamp indicated).
2. Plot all locations and visualize them geographically with the toolkit of your choice (smopy, folium, google geolocation services, etc.). In your plot, indicate the limits of each district by using a different color for the location points such that all locations that lie within this district are of this color. As an added feature in folium for example, you can use clickable pop up messages for a location icon, showing for example the 6 character code and its daily (or monthly, or yearly) forecasted demand.
3. Compute the forecasted daily average deliveries for each location as well as the total daily average deliveries for each district. In doing so, exclude public holidays as they have very specific delivery patterns that do not reflect on usual daily operations.
   Hint: To detect which days are holidays over a time interval, you can use the python package holidays (`https://pypi.org/project/holidays/`).
4. Plot total average deliveries for each district on a bar chart. The graph will have as many bars as the number of districts (around 30).
5. Briefly comment on your observations based on the maps and graphs you have produced.

Step 2—optimization

The second step involves finding an optimal network. This will require solving an optimization problem separately for each district (as each district will have an independent network of lockers).

1. Load the distance between locations data.
2. The network design problem can be modeled as a variant of the capacitated facility location problem (see for example the JOR paper at https://doi.org/10.1016/S0377-2217(99)00464-6). The complete description of the model is provided in appendix II. Study this model in detail.
3. Implement this model using the facilities in the `gurobipy` package, to find an optimal network for a district. The solution of the model shall provide you with the location of each locker-based pickup point and the assignment of each location to one of the pickup points.
4. For all possible (customer) location and (pickup) location pairs in your district, compute the percentage of grocery bags that will be delivered via self-pickups. This, together with the data that you have compiled thus far is sufficient for the optimization of the locker-based pickup points for this specific district. NOTE: In the mathematical description of the model, the percentages are forming the W matrix and the average number of daily deliveries for locations are forming the D vector.
   Hint: The size (and the time to solve) of the facility location problem you need to solve grows quadratically w.r.t number of locations in a district (for $\mathbf{n}$ locations, the computation time to solve the problem becomes proportional to $\mathbf{n^2}$). Be aware that your model will include tens of thousands of variables and constraints for an average district. It should not yet take more than a few minutes to solve such a problem with `gurobi`. If your computational times are exceeding that, it is a sign that there is something wrong with your model. (for all the 30 districts, on relatively powerful computer, it takes in total like 5 minutes to finish the optimization process).

Proceed with the following, once you have solved the optimization problems for each district and hence obtained the district-wide optimal networks of locker-based pickup points.

5. Plot the locations of all pickup points with the toolkit of your choice (smopy, folium, google geolocation services, etc.), along with all locations (as you did in the first step). Make sure that locations with lockers are clearly visible within all locations. You may use graphical means like lines to show the hub-and-spoke structure of the pick-up point and its associated locations it serves. An example of results with maps obtained using the folium packages is presented at the end of this document.
6. Plot the number of pickup points in each district on a bar chart.
7. Briefly comment on your observations based on the maps and graphs you have produced.

Step 3—sensitivity analysis

1. Pick one of the districts (preferably one that has higher forecasted demand).
2. In your analysis thus far, you have used a constraint that considers that max of 35% of deliveries to be self-pickups in each district (adjusted negatively by the individual distances from each location to the pickup point). Conduct a sensitivity analysis and iteratively solve the network design problem considering values from 10% up to 40% with increments of 5%, but only for the district you have randomly picked. For each of these experiments, record the optimal number of pickup points. (be aware that it is possible that for some districts and values solutions are not feasible)
3. Plot the optimal number of pickup points for different self-pickup percentages on a bar chart.
4. Briefly comment on your observations based on the graph you have produced.

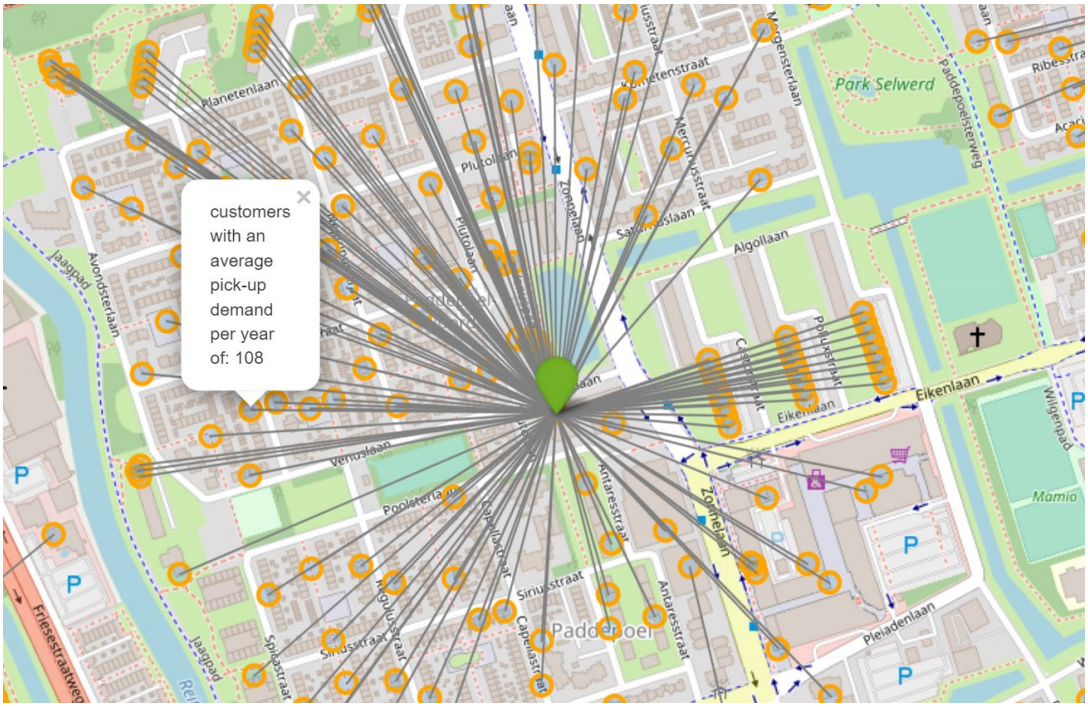
Bonus question —lockers of different sizes

We consider that all lockers have 24 slots. However, such a big size would be not necessary in some areas. This assumes that we have multiple sizes (for example, you can consider lockers with 6, 12, and 18 slots) for three types of lockers. Naturally, the cost increases with size; however, due to the technology used (a single cooling unit is necessary for the whole locker), the cost increase is not linear, but shows the economy of scale (for the above prices, the unitary cost increase will be from 1 for size 6, to 1.1 for size 12, and 1.15 for size 18).
Change the existing model by increasing the number of variables that represent lockers allocated to locations, include the costs in the objective function, and include the different capacity aspect in one of the constraints.
Implement this new model via `gurobipy` and optimize a chosen district (or all the districts if you have time remaining) and visualize on maps and graphs the results.
In the appendix I below, you may find a result of such an optimization, where each district has only one locker, considering that we have two types (small and big, which are indicated with markers of color green and red, respectively).

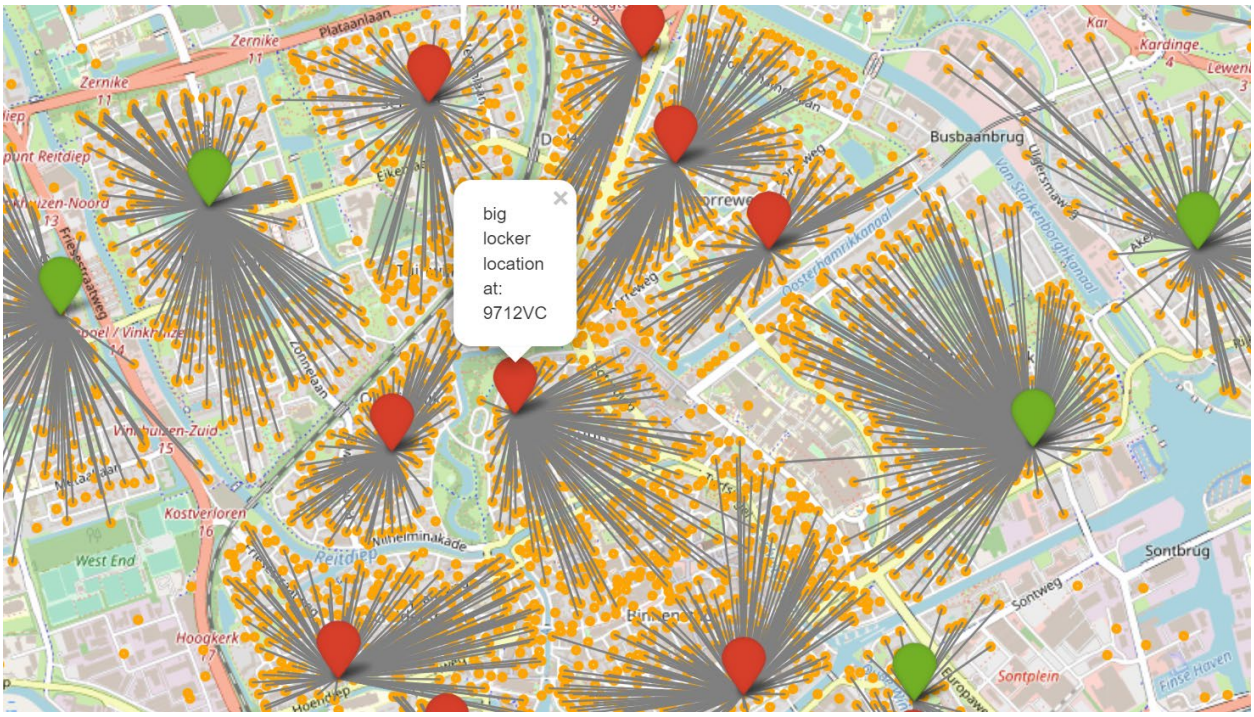
The bonus question will bring a grand total of 3 extra points (on a 1-10 scale) for the entire cohort. We will equally distribute these points among those students who provide a good answer. The bonus question does not count in the page limit.

Appendix I

An optimization result for unique locker size=24, with a result of 78 lockers allocated, serving 5361 locations. Each marker represents a locker, and the hub and spoke networks associated are visible.



A detail obtained by zooming into the first map (for the Paddepoel area) of a locker network that serves part of a district. Observe that folium allows popups for the iconography used. Each circle represents a 6-character postal code (and it is placed in its middle geo location).



For the bonus: An optimization result, where there is only one locker per district, and there are two size of locker (small=green, and big=red).

Appendix II

Network design

Intro. To design a network of local locker-based pickup points for fresh groceries, we need to decide in which locations to open pickup points—while considering in the same time the capacity limitations of pickup points and also the lowest admissible bound on the percentage of total deliveries to be finished via intermediate locker storage and self-pickups by the customer. An optimal network is the one with the minimum possible number of pickup points that satisfies a number of constraints. We model this problem as a linear programming optimization model (in particular as an integer program with binary variables). Before proceeding any further, let us first review what we actually know. We have in total n locations for delivery - including near the potential pickup points (in the model, they have the same index number). The number of daily forecasted grocery bag deliveries at the ith location is D_i. The percentage of these deliveries that will become self-pickups in the case this location uses a pickup point at the jth location is W_{ij}. The capacity of each locker-based pickup point is limited to at most C self-pickups. The percentage of the total number ofself-pickups should be at least P percent of the total number of deliveries. Any optimization model is defined by means of its decision variables, the objective function, and its constraints. In the following, we explain these for our model.

Decision variables. We decide in which locations to open locker-based pickup points with the binary decision variable y_j for $j = 1, \ldots, n$. That is, if $y_j = 1$ then a pickup is opened point at the jth location and $y_j = 0$, otherwise. We make the assignment of the (customer) locations to locker-based pickup points with the binary variable x_{ij} for $i = 1, \ldots, n$ and for $j = 1, \ldots, n$. That is, if $x_{ij} = 1$ then the ith location is assigned to use the pickup point at the jth location and $y_j = 0$ otherwise.

Objective function. The objective of our model is to minimize the number of locker-based pickup points. We can simply write this as $\sum_{j=1}^{n} y_j$.

Constraints. There are several constraints we need to consider in the model. These are listed as follows.

1. Each location should be allocated to at most one pickup point. This can easily be written as $\sum_{j=1}^{n} x_{ij} \leq 1$ for $i = 1, \ldots, n$.
2. The number of self-pickups from each pickup point should be below the capacity of the locker. The number of self-pickups from the jth location can be written as $\sum_{i=1}^{n} D_i W_{ij} x_{ij}$. We need to make sure that this does not exceed the capacity. Obviously, we have capacity only if we open a pickup point at the jth location. Hence, we can write the capacity constraint as $\sum_{i=1}^{n} D_i W_{ij} x_{ij} \leq C y_j$ for $j = 1, \ldots, n$.
3. The total number of self pickups should be at least P percent of the total number of deliveries. From the previous constraint, we know that the self pickups from the jth location is $\sum_{i=1}^{n} D_i W_{ij} x_{ij}$. If we sum this over all locations we obtain the total number of self pickups. That is, $\sum_{i=1}^{n} \sum_{j=1}^{n} D_i W_{ij} x_{ij}$. The total number of deliveries is $\sum_{i=1}^{n} D_i$. Hence, the constraint reads $\sum_{i=1}^{n} \sum_{j=1}^{n} D_i W_{ij} x_{ij} \geq P \sum_{i=1}^{n} D_i$.

4. The ith location can be assigned to the pickup point at the jth location only if a locker-based pickup point is installed next to the the jth location. This can be written as $x_{ij} \leq y_j$ for $i = 1, \ldots, n$ and for $j = 1, \ldots, n$. Note that this constraint is redundant as it is implied by the capacity constraint. That is, if the capacity constraint holds, then this constraint also holds. Nevertheless, adding it has a positive impact on the computational speed, because it reduces the search space for the optimizer.

5. The ith location should not be assigned to the pickup point at the jth location if the associated percentage of self-pickups is zero — because it is too far. This avoids "null" allocations (allocations with no added self-pickups). It can be written as $x_{ij} = 0$ for $i = 1, \ldots, n$ and for $j = 1, \ldots, n$ where $W_{ij} = 0$. Observe that we already computed the value of W_{ij} when writing the model. This constraint is also redundant as such allocations do not contribute to the objective function and the rest of the constraints. It may yet, as the previous constraint, it has a positive impact on the computational speed.

Complete model with one size capacity for lockers

We have discussed the model in detail, yet its implementation is quite lean. The complete model—which you are required to implement in gurobi—is as follows.

$$\min \quad \sum_{j=1}^{n} y_j \tag{1}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij} \leq 1 \qquad i = 1, \ldots, n \tag{2}$$

$$\sum_{i=1}^{n} D_i W_{ij} x_{ij} \leq C y_j \qquad j = 1, \ldots, n \tag{3}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} D_i W_{ij} x_{ij} \geq P \sum_{i=1}^{n} D_i \tag{4}$$

$$x_{ij} \leq y_i \qquad i = 1, \ldots, n;\ j = 1, \ldots, n \tag{5}$$

$$x_{ij} = 0 \qquad i = 1, \ldots, n;\ j = 1, \ldots, n \text{ where } W_{ij} = 0 \tag{6}$$

$$x_{ij} \text{ binary} \qquad i = 1, \ldots, n;\ j = 1, \ldots, n \tag{7}$$

$$y_j \text{ binary} \qquad j = 1, \ldots, n \tag{8}$$

Model with various capacities for lockers - for the bonus points

We will not provide the derived model explicitly, you will have to figure it out yourselves. As a useful hint, think that the number of binary variables representing open pickup locations with lockers have to multiply with the number of possible capacity values. If you have 3 potential capacity as suggested (6, 12, 18), you will have to extend the set of the y variables threefold.

Each variant will have its own cost, albeit not growing linearly with capacity, due to economies of

scale. The capacity C is not anymore a single value, you have to make it a vector C_k, and you have to introduce a new vector for the cost.

The cost has to become a part of the objective function, in a way that the objective is not anymore to minimize the number of locker-enabled pickup points, but the objective is to minimize the overall cost of the district's network.