École Polytechnique Fédérale de Lausanne
Semester project
Master of Civil Engineering, Autumn 2024-2025

Real Data for a Drone-and-Bus Delivery System Scenario

Julien Ars

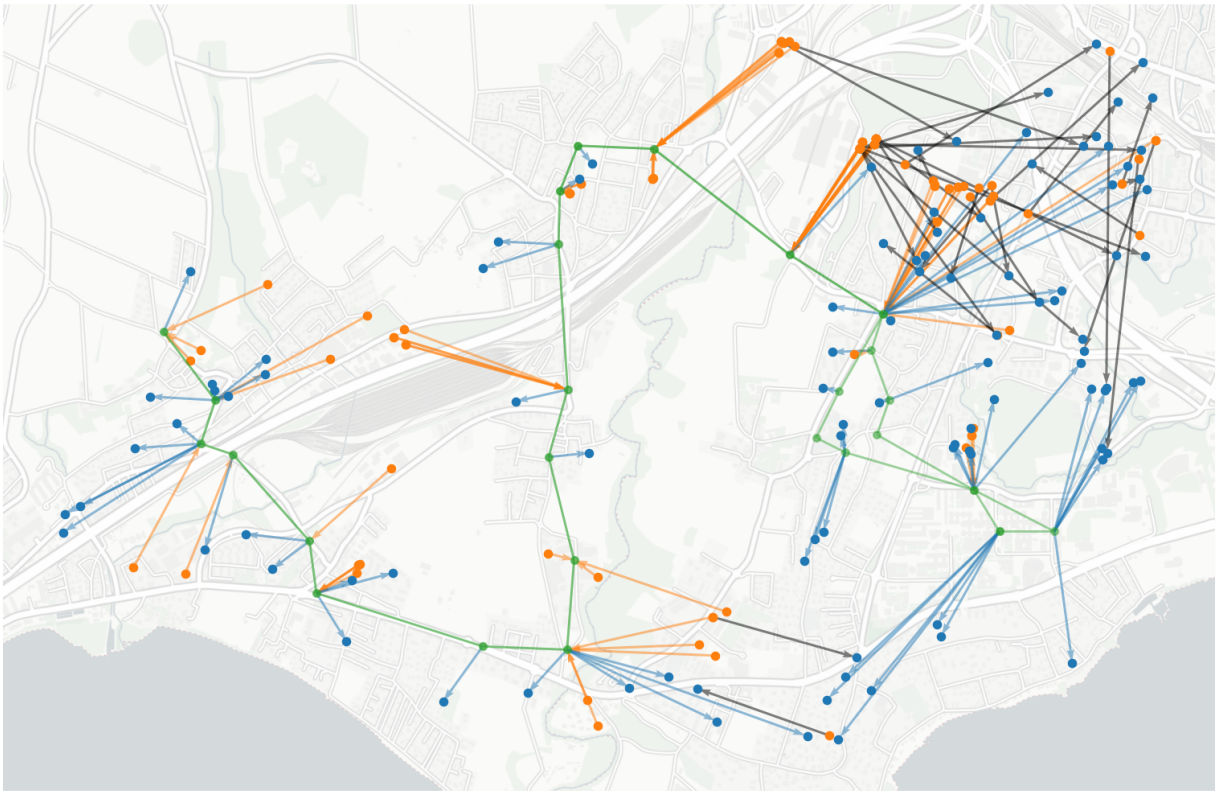


Figure 1: Bus line 705 (MBC), example result

Introduction

The increasing demand for efficient urban logistics has led to the exploration of innovative delivery solutions. One such approach is the exploration of drone-based deliveries system. One interesting idea to research in that field is the integration of public transport systems to improve the performance of drone-based systems. Indeed, in many cases, the delivery system would operate alongside regular public transport service, which it could use to reduce the distance drones have to travel. An example could be using a drone to transport the package from the pickup point to a bus station, put the package in a bus, then have another drone pick the package at another bus stop for the last mile delivery to the delivery point.

In researching the pertinence of such a system, one could use synthetic data like randomised tasks and geometric bus lines, an example of which is found in Figure 2. Although a good starting point, it is an interesting question to ask ourselves to what extend results obtained with this method correspond to the reality.

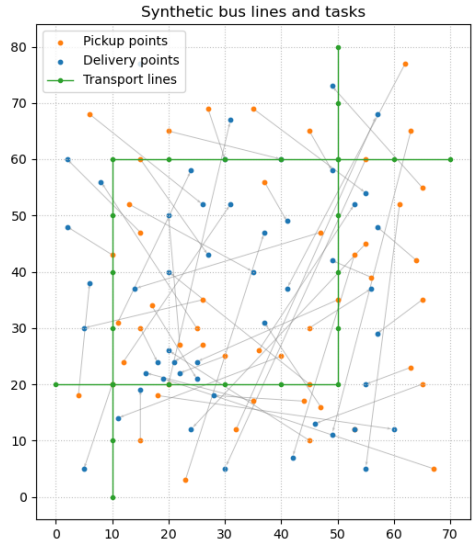


Figure 2: A synthetic bus network with synthetic drone tasks

The goal of this project is to provide the data that will allow to solve this question, by anchoring the experiment to the real world. Using availlable data about public transport in Switzerland, we export bus lines trajectories as well as timetables. Then, using statistical data about the population density and the location of shops, we generate tasks that match the surronding geography. This will enable a comparison with synthetic data and provide a validation framework for previously developed assumptions. Finally, we implement a basic metrix (total drone distance) to compare the impact a drone-and-bus delivery system could have (i) around a suburban bus route; (ii) in a full agglomeration and (iii) in the synthetic environment shown in Figure 2.

Contents

1 Methodology

1.1 Objectives

In this project, we aim to challenge and refine prior assumptions about drone and bus environments, by introducing real-world data in two different aspects. With real bus data, it will be possible to take into account at first the location of bus stations, their mutual distance and the routes of the different bus lines. One could also imagine taking into account real travel times, variations in travel time, delays and unpredictibility. By generating tasks based on statistical data, models could take into account the relationship between the location of customers and shops with the location of bus stops, as well as their councentration.

Additionally, we also had personnal goals. The first one was to develop a deeper personnal understanding of transport and geographical data availlable in Switzerland, and the second one to develop a reusable and user-friendly tool that can generate bus and task data on demand, for a given area and a given date. The methodology used was designed to stay scalable and adaptable so that it can be applied across different regions rather than being limited to a single bus line or geographical area.

1.2 Data sources

To ensure scalability of its results, the project relies on publicly availlable national data. The primary source for public transport data is the Open data platform mobility Switzerland managed by SBB CFF FFS, which provides comprehensive real-time and historical records of public transport movements, including bus, train, and metro schedules. In particular, we will use the Actual data dataset for bus information and the Services points dataset for stop informations. These comprehensive datasets include details such as stop locations, bus schedules, as well as actual departure and arrival time, from which actual bus routes will be reconstructed.

In addition to transport data, the study incorporates demographic and commercial activity data from the Swiss Federal Office of Statistics, a dataset known as GEOSTAT. Specifically, the STATPOP data provides population density metrics, household sizes, and demographic distributions, which help to determine where potential delivery demand is highest. The STATENT data offers insight into business distributions, particularly retail locations, enabling the identification of key shops from where packages would have to be taken. By combining transport, demographic, and commercial data, we provide insight into real-world demand patterns.

1.3 Task generation

Task generation is a crucial component of this study, as it determines the origin and destination points for package deliveries. As previously stated, we used the GEOSTAT dataset for this part. This dataset uses grid cells of 100m x 100m to aggregate the data, however, to gain in realsticness we introduced random offsets for each customer and shop. The method however

varies for customers (deliveries destination) and shops (deliveries origins).

For deliveries destinations (customers), we used the STATPOP data that provides, in particular, total population within each cell. For each task generated, we will firstly select a cell within the study area, using the total population in each cell as weights. Then, we add random offsets from -50 m to 50 m to the center of the cell and consider this point to be the delivery point of the task. By using population data to generate customer points, we ensure that task distribution reflects actual residential density. Random offsets within the grid cells add variability, preventing clustering in a single point and better simulating realistic delivery demand.

For deliveries origins (shops), the STATENT dataset provides business density metrics, which are filtered by sector. As such, we selected the retail sector. Here, we use two variables per cell: the number of retail locations and the number of retail jobs in jobs equivalent. The methodology is a little different than to the customer points. We firstly will generate, inside each cell, one random point for each shop and assign him a weight equal to the average number of jobs per shop in that cell, that is the number of jobs in the cell divided by the number of shops in the cell. Then, for each task generated, we will pick a random point from this set using those weights. This point will be the pickup point of the task. This method allows to represent the reality that individual shops are likely to be the pickup point of many drones deliveries, instead of having many different pickup points appearing.

1.4 Bus data processing

Processing the raw bus data in a format that correspond to our study proved more challenging. Indeed, the original data is a table with a new row for every time a public transport reached a stop in switzerland on the given day. The first step therefore involves filtering the dataset to keep only the rows concerning the bus lines we are studying. The next step consists of reconstructing bus routes based on the records. One key challenge here is handling bus lines that have multiple routes. Some bus lines follow different routes depending on the direction, others have different variants, some will desserve some stops depending of the time (for example, one could stop at a school when classes start and finish only, or some only desserve their full route on peak hour). For each individual bus, we must determine which bus stops he stops at and the order in which he goes through them. Then, we group buses that follow the same route, and name those routes. We then select the most used route and order, and use it to base our order of stops. We then add the stops that are not in this order by interpolation on the next routes. This process allows us to define a fictional **distance** variable that place all stops of a given line on a single axis, making our final results much more readable. The third step is cleaning the data. Indeed, we noticed two irregularities in the data that we wanted to correct : (i) Some bus lines had strange routes that share only one or two stops with the 'main' route. We automatically filter those routes out and remove them from our results (by applying a default threshold of 5, that is we remove routes that share less than 5 stops with the route most used), altough the threshold can be adjusted by the user. (ii) Some buses seemed to have

skipped one stop and then come back to it later. Assuming a technical error, we corrected the time data automatically as well, by keeping the maximum value registered (that is, if a bus goes backwards, we change the time of the next stops and make it equal to the previous time). Finally, we export the results. To do so, we produce 4 types of data for each bus line:

1. Timetable data, that countains, for each bus, its scheduled and real arrival and departure time at each stop. This data is then further separated into scheduled times and real times, for easier readability. The output is both human- and machine-readable.

2. Stops data, that countains for each stop its name and id, its position, its **distance** value and whether it is on each route.

3. Route data, that countains for each route, whether each stop is a part of it.

4. Journey data, that countains for each individual bus journey, its ID, its route, number of stops and direction, and its start and end stops as well as the time at which it started and ended.

A python object that countains this data is also returned, for further use.

These steps ensures that the returned dataset accurately represents real-world transit operations, capturing the real routes but also route differences, delays and variabilities accross time and space, while staying sufficiently clear that it can easily be incorporated into an existing or future simulation.

1.5 Evaluation

In order to summarly evaluate the impact of real world data into a drone-and-bus delivery simulation, we decided to focus on the total drone distance, that is the sum, for every task, of the distance from the pickup point and the delivery point to their respective nearest bus stop, comparing it with what would have been the direct distance without using the bus system. In the cases with different bus lines, we assumed that each package could only use one bus line and therefore used the line which would give him the smallest distance.

2 Application

A comparative study was conducted to analyze the impact of using real data on delivery simulation metrics across three different cases, while demonstrating the possibilities of our solution. The first case, a baseline scenario, represents synthetic bus lines and tasks. The second case represents a suburban environment, that is the area surronding bus line 705 that radiate from EPFL to the small villages in the direction of Morges, while the third case represents a city area, namely the Lausanne agglomeration.

2.1 Baseline case : synthetic bus lines and tasks

For this case, we consider a rectangle of 70 x 80 space units. Two bus lines are set up, each with 12 bus stations equally spaced, their central parts making a square around the city center, while tasks are taken from the PDPTW dataset. 50 tasks are considered for this case, altough 2 tasks have the same pickup and delivery point. Figure 3 represents this case.

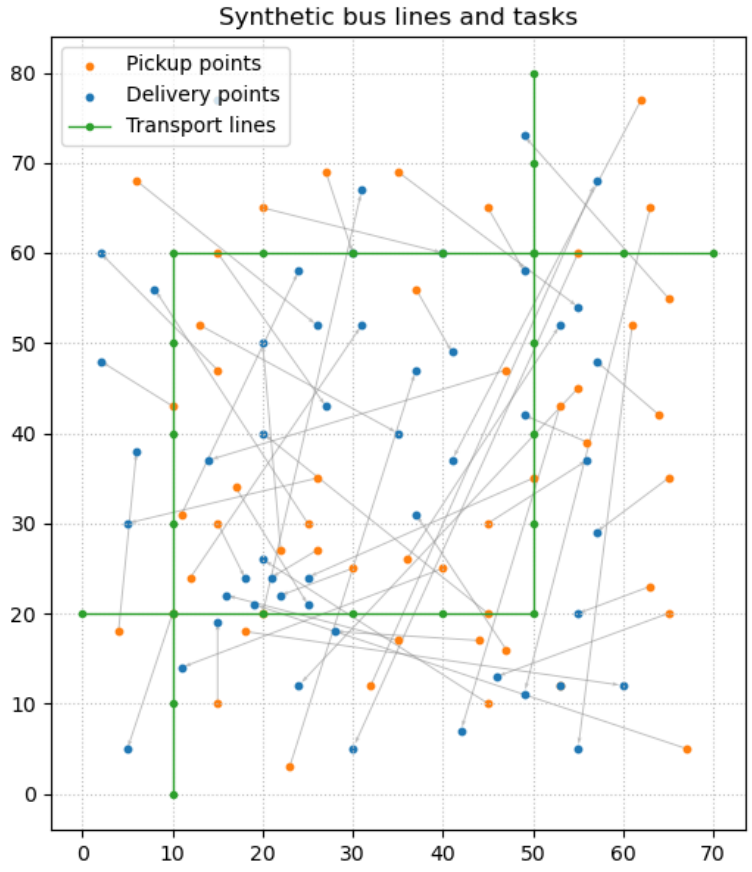
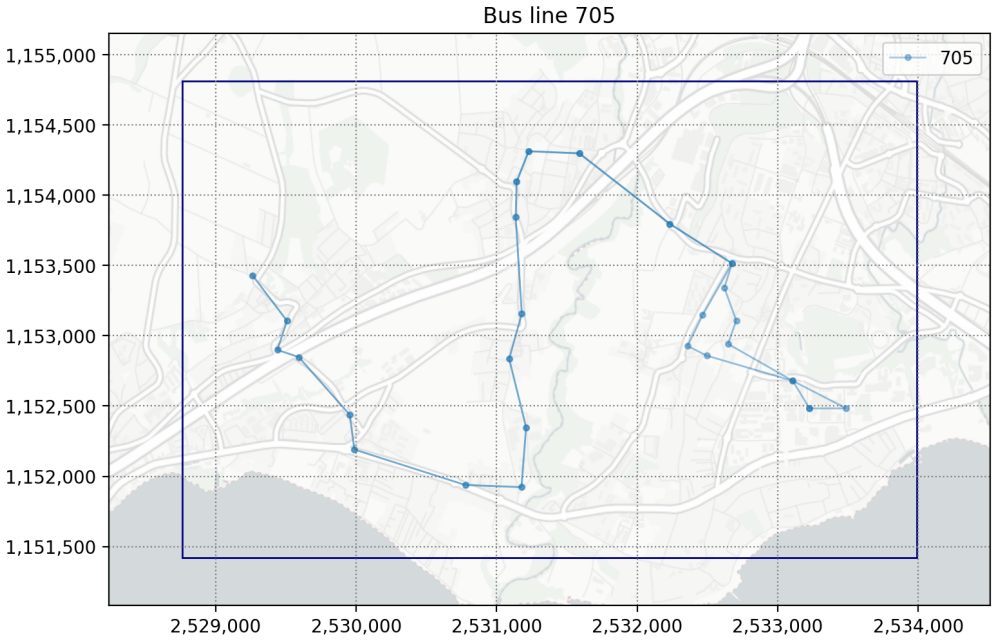


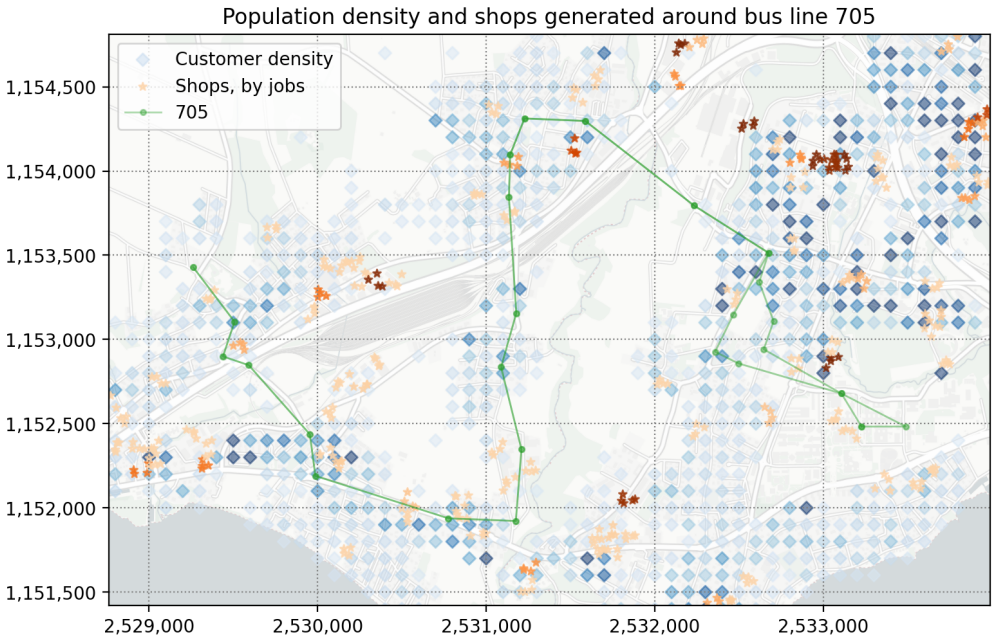
Figure 3: Our baseline case

2.2 Suburban case : Bus line 705

Here, we took bus line 705 from EPFL to represent a suburban environment. This bus line links EPFL to the Lonay village, passing through the villages of Ecublens, Echandens, Denges and Préverenge. We defined the study area as a rectangle fitting every stop of the bus line with a margin of 500m on each side. As can be seen on Figure 4, the line makes a Z-shape in order to desserve most villages in the area. Altough this reduces the interest of the line for users, it could make it very usefull in our case, by desserving most customers and shops.



(a) Area considered



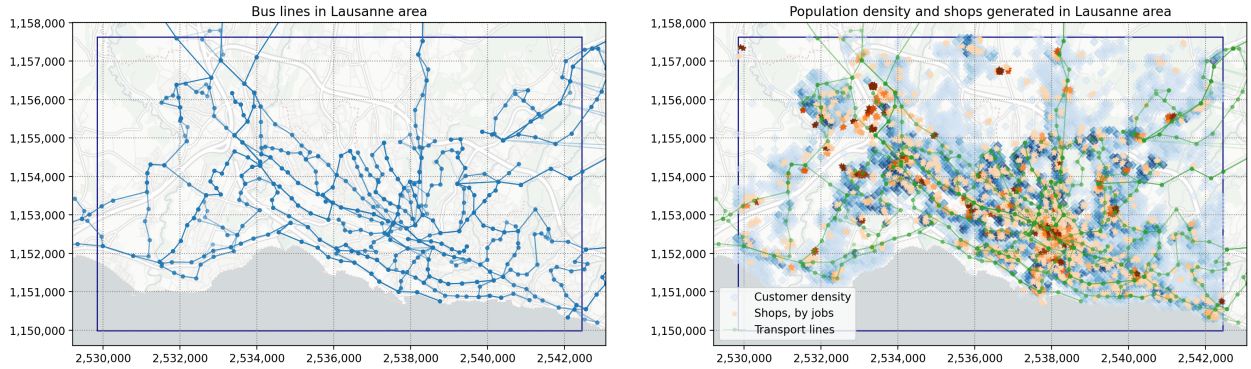
(b) Population densities and generated shops

Figure 4: Suburuban case : surroundings of bus line 705

Looking at the population and shops (Figure 4b), we can see the bus line desserves pretty well the population, with the exception of the St Sulpice village in the south and of the Ecublens/Chavannes-près-Renens centers at the north-east, which countain a high density of population and shops.

2.3 City case : Lausanne agglomeration

For the third case, we wanted to explore a more complete environment, at a larger scale. We choose for that the Lausanne agglomeration, as depicted in Figure 5, as it involves the whole city of Lausanne and surroundings municipalities. In particular, as we can see in the population densities, this area includes the dense city centre, the greater agglomeration with Renens as a second pole, and then outspots, such as the Romanel, Crissier, St Sulpice municipalities. Following population, transport lines radiate from the city and desserve most of the population, altough we could identify a significant zone not well desserved (around coordinates $(2,536,000\ ,\ 1,157,000)$), which could be explained in that it is desserved by the LEB train and therefore ignored by the bus network. This is an interesting case of some places where the drones would not much benefit from the bus network but the customers could still expect to be served by the delivery system as they are a part of the agglomeration. To account for the greater population, 1000 tasks will be generated.



(a) Area and bus lines considered

(b) Population densities and generated shops

Figure 5: City case : Lausanne agglomeration

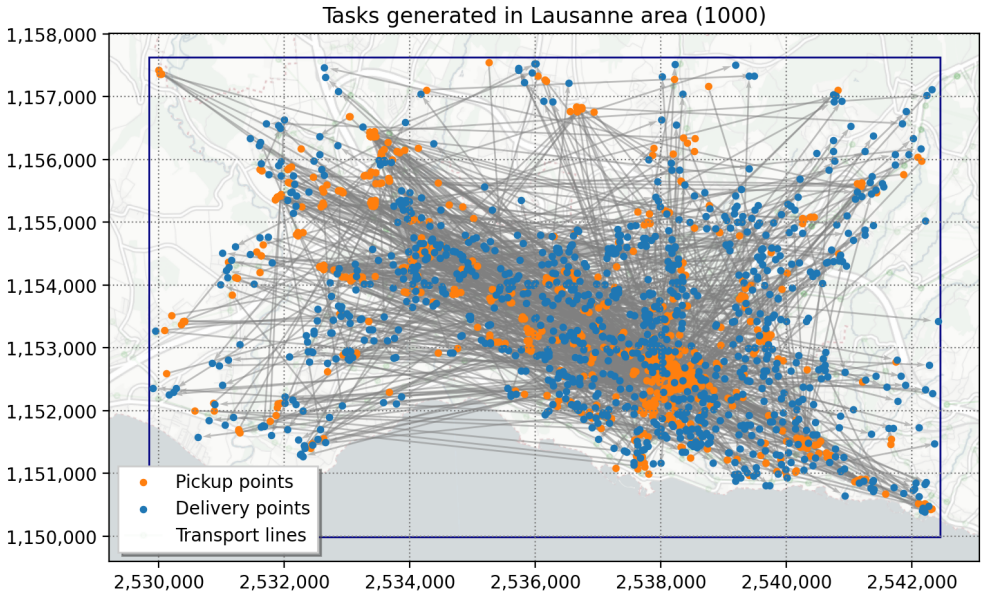
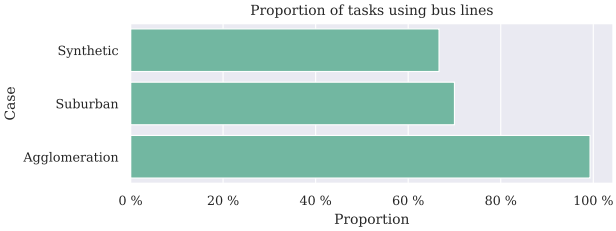


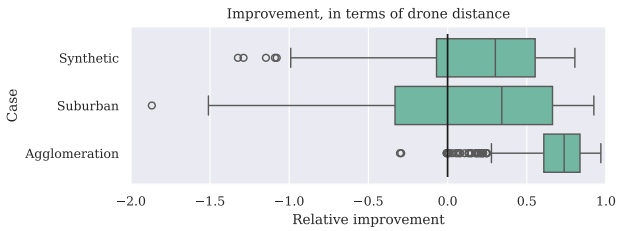
Figure 6: City case : Tasks generated

2.4 Results

The drone trajectories with and without using the bus lines are depicted for each case in Figure 8. For visualisation purposes, we represented on the maps the packages for which the improvement would be negative as not using the bus network and instead going directly to the delivery point, with a black arrow. Quantitatively, we will use two metrics : the percentage of packages for which using the bus lines reduces the distance drones have to travel, and the distribution of the relative improvement in distance. They are visualised in Figure 7 :



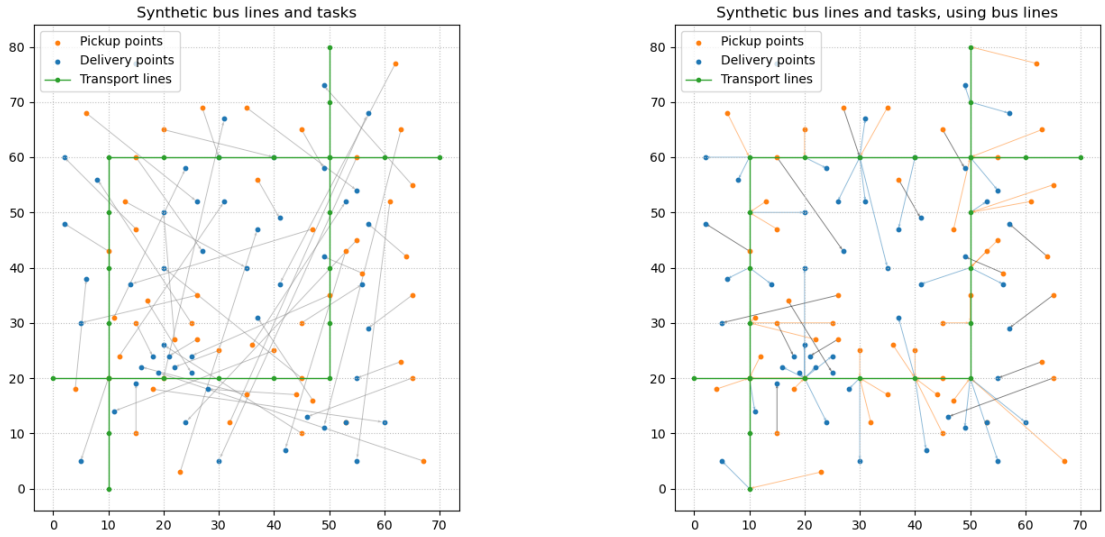
(a) Proportion of tasks using bus lines



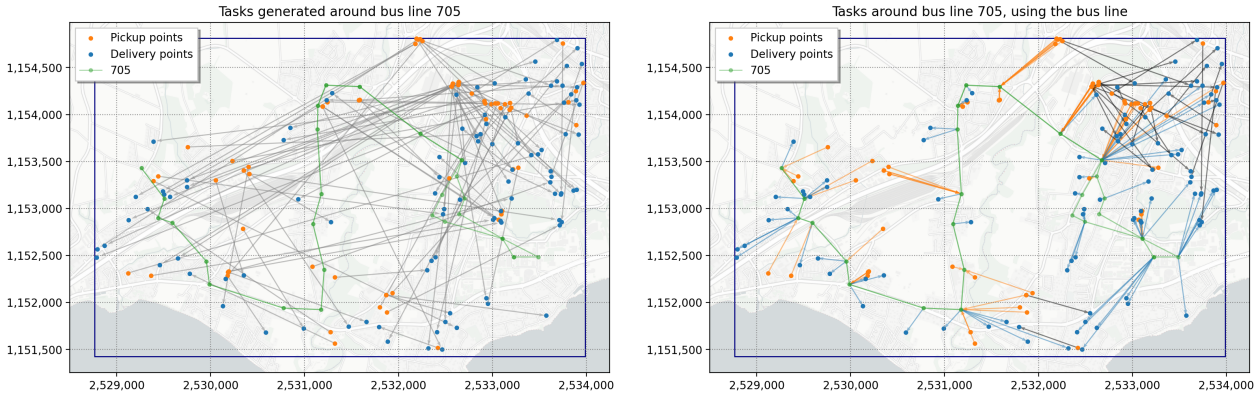
(b) Improvement (normalised to the direct distance)

Figure 7: Results

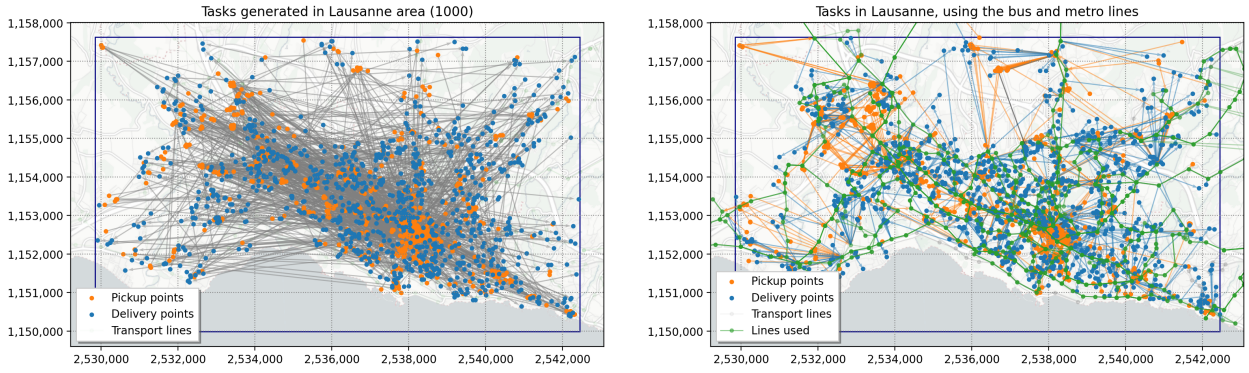
We can see that the proportion of tasks using bus lines increase slightly between the synthetic case and the suburban case, and increase much more using the agglomeration case. In the relative improvement, we can see that when the suburban case does not serve well the packages, the negative improvement can be very big, however, in such a case, the drone would go directly to the delivery point, without using the bus line.



(a) Baseline case



(b) Suburban case



(c) City agglomeration case

Figure 8: Comparison of drone trajectories without using the bus lines (left) and with using the bus lines (right)

2.5 Extra analysis : number of bus lines necessary

The very good results of the agglomeration case can be explained easily by the dense network of public transport lines availlable. Interesting related questions what is the impact of the number of bus lines, and what is the minimum of bus lines that should be included in the scheme to get an already good result. In order to do so, we will order the bus line by their effectiveness and then, while adding them progressively to our scenario, measure the evolution of both the improvement in total drone travel distance and the number of packages using the bus line.

In order to order them by their effectiveness, we used two rankings : the number of packages using the line, and the total reduction in drone travel distance they are responsible for. The results can be visualised in Figure 9 :

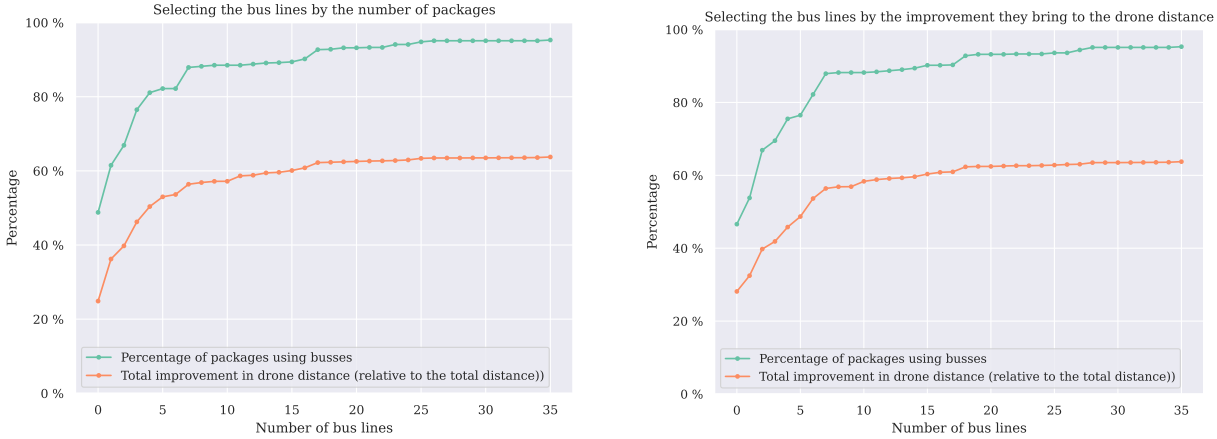


Figure 9: Influence of the number of bus lines availlable

To answer the question of how many lines are necessary for good enough result, we normalise both metrics by their maximum value (Figure 10). We notice that ranking the bus lines by the number of package bring good results faster (80% reached in 4-5 bus lines, compared to 5-6), and that only 10 bus lines are necessary to get 90% of the result.

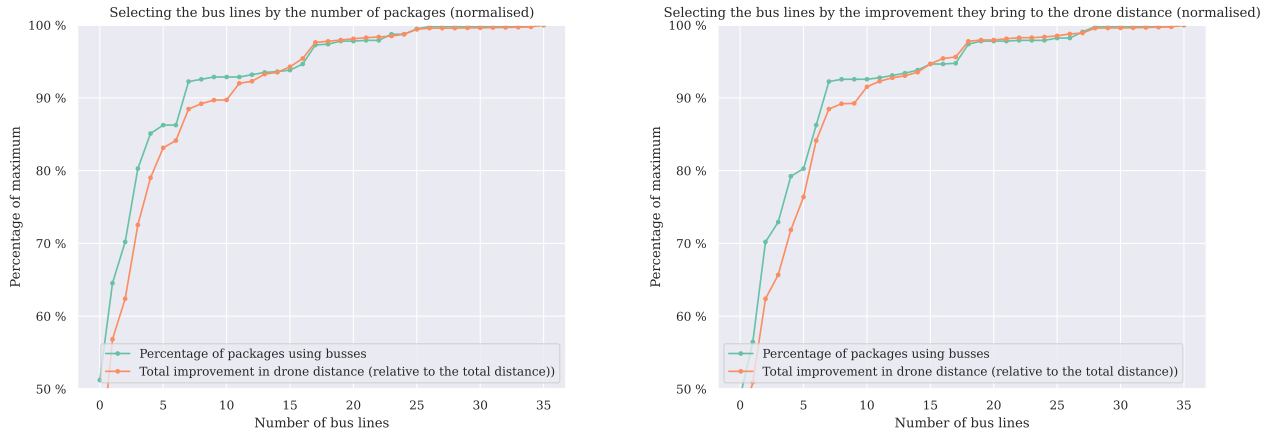


Figure 10: Influence of the number of bus lines availlable (normalised)

3 Conclusion

In this study, we have implemented multiple python tools to import, process and use general data about bus systems and socio-economical geographical variables into processed data for use in a drone-and-bus delivery simulation scenario, allowing to place these simulations in a more realistic setting. In addition, we have demonstrated that in realistic scenarios, key metrics like the possible improvement in total drone distance, are likely to improve compared to a synthetic scenario.

The work done here could be completed by adding other methods of tasks pairing, for example with a minimal and maximum distance or by pairing to the nearest shop. Including time data both in bus data and in tasks - adding time constraints to the deliveries - could also bring more realistic results. For tasks, new data sources would help anchor those time constraints to the reality, for example home utilities usage to generate times for the delivery, and shops opening hours for the pickup. For bus data, we already process the time data and output a timetable. This could allow a simulation to run more realistically, for example determining bus frequencies and mean travel times from these outputs, or even by including the variability in travel times and frequencies (for example, at rush hour) and monitoring their impact on the simulation. The data also includes scheduled and real departure and arrival times, which could allow for a simulation that includes the unpredictability of the bus real arrival time.

The processing of bus data could be made more robust to work also on other public transport in Switzerland. The outputs could also be used for other researchs purposes than a drone-and-bus delivery system, given their general aspect.

Code The code is availlable on github : `https://github.com/merlebleue/DRONES-AND-BUS`

Acknowledgements Many thanks to Minru Wang and Professor Geroliminis for their supervision. The code uses the smopy package for mapping, with backgrounds from CARTO in the visualisation.