End of studies internship : Integrating carpooling to public transit through GTFS data based simulations

Abstract Many existing carpooling applications and platforms rely on long distance carpooling that is organized between both drivers and riders days ahead. In this study, we encourage the integration of carpooling to daily commute through the study of various simulation scenarios involving GTFS data and off-the-shelf routing software Open Trip Planner. We showcase several metrics as well as considerations to CO2 emissions, we also implement a financial model and study its profitability with regards to several important variables.

Keywords Computing , Transportation , Carpooling

1 Introduction

A carpooling system is constituted by a set of drivers willing to take detours in order to meet riders and of course riders willing to move to driver locations. Trip scheduling is usually done days ahead with both actors, which makes its integration to daily commute challenging given that both riders and drivers need to have the same origin, destination and departure time.

With that in mind we present our work which will be grounded on the idea of integrating public transit (PT) to carpooling through scheduling detours to PT stops. Carpooling will serve as a feeder service to PT and improve the difficulty of matching between both riders and drivers by offering more carpooling options.

Our primary concern has been to offer a simulation demonstrating how such a system might work and provide statistics that can showcase how efficient it is compared to previously existing ways to travel. As such, we have used GTFS data which is a standardized database format used by both transportation agencies and developers in order to create the simulation through adding carpooling lines as ephemeral bus lines. This comes as a generalization of our previous work [1] which handles a toy case study of this system. GTFS data adds a layer of realism given that we now model and compute trajectories with a real and complex PT network.

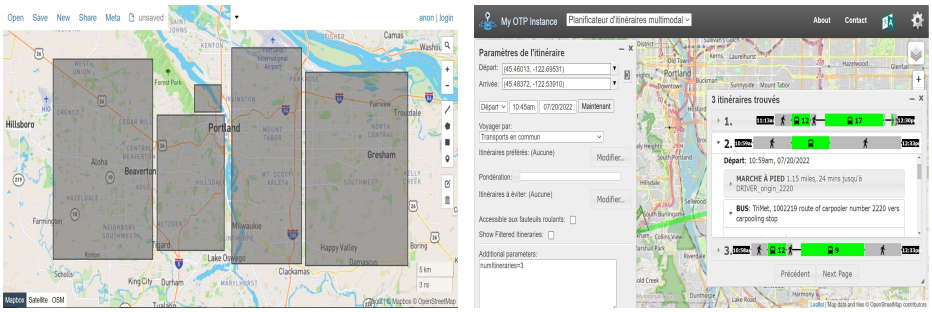
After presenting a simulation, our work then consisted of modeling payments and costs and creating incentives to users of our system and optimizing several parameters like detour rates with machine learning based techniques.

2 Related work

Previous works on integrating ride-sharing to transit with the use of intermediary stations can be found in [8] where they compute Pareto Optimal connections minimizing travel time and interchanges, our work differs from this by considering stations closest to the origin and destination and considering driver detour sizes as a parameter of the problem and without paying closer attention to connections.

GTFS databases [5] are used to share transportation information between both transportation agencies and developers, they have been used by the community for analyzing different networks with several KPIs developed through graph analysis as demonstrated in [9]. Our approach benefits both research topics, we will use GTFS data to provide insight upon our simulations and provide a heuristics based matching algorithm. This process allows to both gain network related

Regular Paper



(a) (b)

Fig.1. Maximum vehicle occupancy in (a) Current system. (b) Integrated system.

insight through the use of [9] and related research as well as provide flexibility in computing optimal routes and matching solutions following the example of [8].

A more in depth read of our heuristic approach as well as our most relevant parameters can be found in our previous work [1] where a toy case has been implemented involving the same algorithm for creating driver journeys. Our use of OTP and GTFS data discarded our previous matching algorithms as they are now replaced with multiple time-dependent shortest path calculations present in OTP that can be found in [2] and [10].

Our intuition at first considered the matching problem as a complex one involving a large number of states and the presence of intelligent users that can be thought of with a utility maximization approach. This perspective called for Distributed Deep Reinforcement Learning considerations as present in [4]. The scalability as well as the optimality reached in this work is encouraging, but our work offers an ease of use to several different cases with the availability of GTFS data, In depth analysis of transportation networks as a whole and the flexibility to develop even AI based optimizations as part of future work.

3 Problem statement and existing applications

Mobility As A Service which is the use of ridesharing in daily life has seen several applications. Two distinctions need to be made, The long distance trips and short distance commute. Long distance travel is thought of as a compromise between riders and drivers and can be planned days ahead, such is the case of Blablacar. As a daily commute, it is more instantaneous as in the case of ridesharing applications such as Uber, Bolt and Lyft. A main challenge in this field exists in reliability, people will still use private cars if their trips are found in a zero risk, zero delays situation. Convenience and a proper compensation model are also challenges that these applications need to address as mentioned in [11].

Where long distance scheduled ridesharing is a great initiative to reduce emissions and congestion, the rarity of matchings in short commutes makes its founding principles unusable in that case. Also, the current short travel solutions seem to increase congestion as mentioned in a survey in [13] for the case of Uber and Lyft. Which isnt encouraging given that congestion is an existing problem in several cities of the world and it usually means more cars have been added on the road meaning more CO2 emissions. This is intuitive given that Uber drivers are similar to taxi drivers in the fact that they can have an idle presence on the road.

Our proposed solution tries to adapt the long distance matching principles to short distance daily commute through feeding to PT and integration. This enables to address the rarity of matches in long distance travel with alternative routes and detours to drivers and help reduce congestion by involving drivers with specific destinations which will reduce idle cars on the road.

Our contributions can be summarized as follows :

- We propose a model to describe both PT and drivers journeys in a single GTFS database
- We simulate riders journeys using an open source software with OTP
- Analyze several transportation scenarios and offer descriptive statistics
- Elaborate an incentive model for both riders and drivers and analyze its profitability

4 Method

Our approach will consist of four main stages, we first collect a raw GTFS database and OpenStreetMap data on a location of our choosing, in this case Portland, Oregon. We proceed to generating drivers and integrating their stop times to the GTFS data. We then proceed to generate riders with the same technique and use OTP requests to generate their journeys for different transportation scenarios. The third component of our work is a descriptive analysis. As a last step, we move on to computing the compensation model and analyzing it using the generated data.

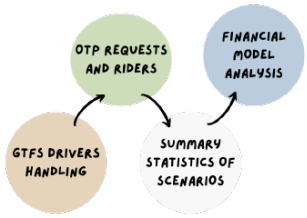


Fig.2. Main Workflow Chart

5 GTFS based simulation

5.1 Generating drivers' journeys

We discuss during this section how we generate drivers. Both riders and drivers are characterized by an origin, a destination, a departure time and a journey. A journey can be thought of as a set of location, time couples which are referred to as stop times. We took for our simulation the city of Portland Oregon as a starting location given the availability of GTFS data. We started by creating a geojson file with several rectangles with and choosing uniformly random points from a random rectangle. Its important to note here that as a current work in progress, we have also developed a script that allows us to compute population density of each rectangle with raster images and simulate a probability function that gives the rectangle with a higher average density a higher probability to be chosen.

We start by generating drivers in a notebook where we generate their origins, destinations and departure times between 10:30am and 11:30am. We then move on to generating their trajectories in a list conveniently named trajectories_list, that contains for each driver the different detours he might take. This trajectory is set by an algorithm that will randomly select either the closest station to the origin and destination, compute if

the added distance is less than 15 percent of the initial trip, then add it, it will then move on to adding the remaining station under the same condition.

After these steps, we move to the core idea of our paper, which is considering drivers as ephemeral bus lines. By that we mean that we will alter a GTFS database in order to add the driver routes. We do that by adding for each driver a single route named route of carpooler number n to the routes.txt file, a single trip in trips.txt and as many stop times as there are items in his corresponding trajectory in stop_times.txt file. This will ensure the creation of a GTFS database that will take into account PT and carpooling which we will feed to our routing software which is in this case OpenTripPlanner (OTP)

5.2 OTP and riders' journeys

In this part, we will discuss how we use OTP in order to generate riders' journeys. After obtaining the GTFS database discussed in the previous section, we use it in order to start an OTP server. We initialize riders by generating their origins, destinations, departure times and an OTP request. The latter will be used in order to generate responses for each rider that we will store. These responses will contain the different itineraries riders have taken, a python script will compute for each rider the fastest route and store information related to it such as the total duration, the modes of transportation used and the total distance. This script will also save information about drivers such as the riders that interacted with them as well as their boarding and alighting times.

Such a process is executed for three different scenarios, the first being our proposed integrated system, which takes into accounts multimodal paths. The current system that separates carpooling and transit and the no carpooling system which allows only travel by foot or transit.

Fig.3. Code Architecture Diagram

6 Publication results

After collecting all the necessary data from both OTP and python, we move to developing statistics and analysis tools for several aspects of the project. One of our main statistics has been to compare the three different systems to each other.

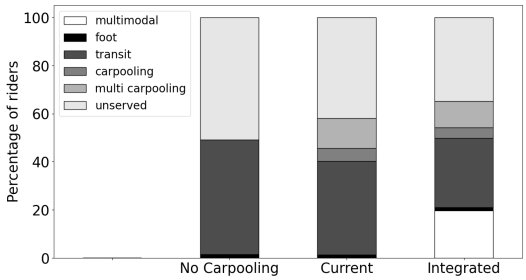


Fig.4. Outcome of riders in the different scenarios

we can see in *figure 4* that the integrated system has served more riders than the current one with an increase of roughly 8 percent. But more importantly, we can see that from the riders in the integrated system, 20 percent have chosen a multimodal option as it was the fastest one available to them.

figure 5 showcases the detours taken by drivers in our integrated system simulation, the decrease in effec-

tive drivers count indicates the existence of a maximum threshold of maximum detours where drivers become inefficient, later analysis of such this metric is necessary as the shorter the detour, the more inclined drivers are to making them, but then less detours will be made. Compromise can be made between both these assertions as well as profitability and costs, which will be the subject of our next section.

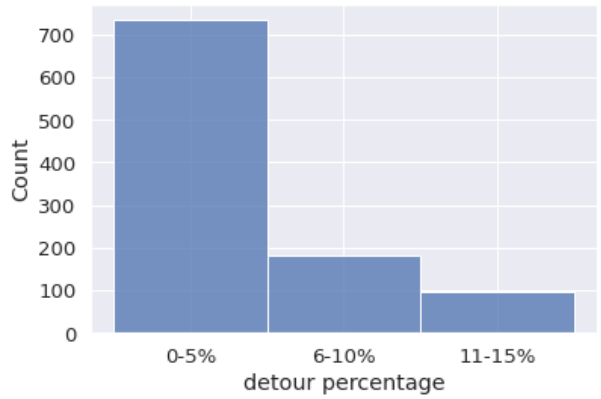


Fig.5. Detours taken by drivers in integrated

We also managed to compute the number of riders for each driver in both the current and integrated system as shown in *figure 6*. We call this metric maximum vehicle occupancy and it is one of the most important metrics of our study. The future direction of this project needs to move to optimizing this metric as more people in less cars is a fundamental environmental argument towards a real implementation of such a system.

7 Costs analysis

In this part, we will discuss how we implemented a financial model into our work and how we analyzed it with various metrics. The main logic behind our financial model is the implementation of two rider costs, the private cost, which is the cost if a rider chooses to drive his own car, and a system cost, which computes the case where the rider uses our system.

$$C_r^{priv} = d(org, dst).\alpha + VoT.\frac{d(org, dst)}{v_{car}} + K_{park} \quad (1)$$

$$C_r^{pool} = \beta_r + VoT.T^{pool}(org, dst) \quad (2)$$

Private costs involve distance and a vehicle use constant which represents insurance, fuel and maintenance costs. We also express the value of time for the rider which is a constant and a parking constant. The pooling costs involve the price to pay and the value of time. When setting these two costs as equal, we can formulate the price to pay as such.

$$\beta_r = C_r^{priv} - VoT.T^{pool}(org, dst) \quad (3)$$

This constant can be positive or negative, it is positive when the system is beneficial to the rider and reduces his expenses and negative in the other case. To have a better idea on the profit such a system can create, we must calculate driver costs, which will be the amount of money we compensate the driver within our system. We formulate driver costs as such :

$$C_d^{add} = \alpha.detour + VoT.detour \quad (4)$$

We pay the driver for the detour he took by considering both the value of time and the vehicle use constant. This will allow us to compute the general profit made from our system which is intuitively subtracting money paid to drivers from money received from riders or given in a case of a negative beta. The formula will look as follows :

$$P = \sum_r \beta_r - \sum_d C_d^{add} \quad (5)$$

Now several refinements have been considered after an implementation of these formulas. The fundamental one has been to add co2 emissions to the profit as we can compute the co2 emissions our system saves compared to the current system. We have found that this calculation is negligible to the whole profit as it saved around 32 euros per hour of simulation. But this is still

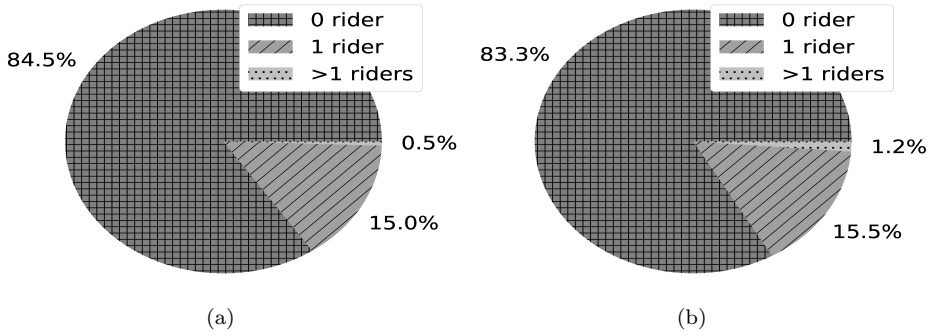


Fig.6. Maximum vehicle occupancy in (a) Current system. (b) Integrated system.

an important item to add to the computation, as with a larger system this profit can become significant, further study in future work is necessary as co2 emissions reduction is always a solid argument towards carpooling systems.

In our analysis of profit, we defined an efficiency metric, which is the total distance traveled divided by the price to pay and engaged in several analyses regarding this metric as well as the price to pay. The first strategy was to consider only profitable riders, by that we exclude both riders that have a negative beta and drivers that took them. The profit has lowered by 50 percent through this experiment, a further analysis of this deflation is necessary through the study of different cases as this only provides an insight towards deflation.

We then moved to considering efficiency thresholds as well as price to pay thresholds. We notice a gradual decrease in profit as we eliminate carpooling jobs with lower efficiency. We can conclude that such an approach cant optimize profit, but rather a more in depth look at the inner matching algorithm might be a better approach. Which will lead us to the current work in progress for our project that consists of the development of a machine learning based tool that will learn which detours to propose to drivers provided several features that we will discuss in the last section.

8 Ongoing work in detour optimization

The first step has been the selection of the most relevant features. Considering the geographical nature of our project, considering geographical and geometric properties felt essential. The simplest features have then been distances between origin and stations, destination and stations as well as the full distance between origin and destination. Then we needed features related to population density with the idea that stations with greater population density are more prone to be visited. This has been implemented through the parsing of raster data which is satellite imaging that contains population density.

This approach has created two issues, the first being that population density around stations is a constant which makes feeding it to a neural network counterintuitive, the second is that our simulations didnt account for density data yet. In response to the second problem, we assigned a probability function for each rectangle of our location generator that depended on the average population density of each rectangle. This can be done by simulating a discrete probability law through the cumulative distribution. To solve the first issue, further insight on assigning constants to NN is necessary, another perspective might be the creation of a feature for stations that finds compromise between distance and density and limit the possible detours to the stations

that have the best scores in this metric.

The second step of this work is the creation of a convenient dataset, given that we want to optimize the detours through machine learning. We wanted to generate for a given driver several detour outcomes, the reasoning behind this is to fabricate an objective function that will optimize the detours with information about profitability and other factors in each case.

9 Conclusion and future work

To summarize our achievements during this project, we have successfully created a tool to simulate several scenarios in transportation. Starting from a toy case in our previous study and now generalized to a real network in Portland, Oregon using GTFS data and an off-the-shelf routing software (OTP). The code behind this work is interesting as it allows for many other applications to transportation such as testing new means of transportation or adding new bus lines/ train lines and seeing how it affects mobility.

More than a simulation tool, we provided several analytics and ways to get a deeper grasp of ridesharing problems. From an environmental perspective to the financial one, our project has moved fast towards providing strong arguments towards its real case implementation. The results we have now are in favor of the integrated system, from it being the fastest amongst the status quo with relevant quantities, to it being an eco-friendly initiative that saves CO2 emissions without any deep or structural engineering effort but rather an increased efficiency in using existing platforms and means of transportation.

Such research work is very important as the environmental issues are growing, transportation being the second most polluting sector with 22% of overall air pollution within which 40% are due to automobiles. Simple solutions such as the one we proposed become a necessity. Presenting accurate simulations with precise and strong arguments towards finding concrete solutions.

As future work, congestion as well as other factors should be considered, as this will add an extra layer of realism to our simulations. The payment model should be studied with more attention to the current state of the art. It can even be incorporated into a utility-maximization model for both riders and drivers which will open up the project to a multiagent system perspective.

Learning based detour optimization should be completed, work towards a mobile application worthy architecture that considers realtime GTFS data and automatic Machine Learning. Some aspects of this project such as the formalization of states and the large number of users suggest a deep reinforcement learning perspective. The latter perspective would complement the innovative aspect of this work with artificial intelligence technology that is reliable and the many decisions of which can be grasped through AI explainability.

References

[1] A. Araldo et al. Pooling for first and last mile: Integrating carpooling and transit. In hEART Conference, 2022.

[2] Yishu Wang et al. .Time Dependent graphs: Definitions, Applications, and Algorithms.

[3] Hamzehi, S., Franeck, P., Kaltenhuser, B., Bogenberger, K. (2021). Approximate Collaborative Fleet Routing with a Pointer Generation Neural Network Approach. IFACPapersOnLine, 54(2), 195-202.

[4] Abubakr Alabbasi, DeepPool: Distributed Model-free Algorithm for Ride-sharing using Deep Reinforcement Learning

[5] General Transit Feed Specification. https://gtfs.org/.

[6] Open Trip Planner. https://www.opentripplanner.org/.

[7] volution du Taux Moyen dmission de CO2 en France. Technical report, ADEME, 2022.

[8] S. Fahnenschreiber et al. A Multi-modal Routing Approach Combining Dynamic Ride-sharing and Public Transport. Transp. Res. Procedia, 13:176183, 2016.

[9] P. Fortin et al. Innovative gtfs data application for transit network analysis using a graph-oriented method. J. of Pub. Tr., 19:1837, 12 2016.

[10] Bast, H., Delling, D., Goldberg, A., Mller-Hannemann, M., Pajor, T., Sanders, P., ... Werneck, R. F. (2016). Route planning in transportation networks. In Algorithm engineering (pp. 19-80). Springer, Cham.

[11] https://mobility-as-a-service.blog/carpooling-lessons-learned/

[12] https://www.uber.com/gb/en/about/sustainability/

[13] https://www.vox.com/the-goods/2019/8/6/20757593/uber-lyft-traffic-congestion-pricing