PoolLines: modeling carpooling as ephemeral lines in GTFS for effective integration with public transit

Youssef Chaabouni, Andrea Araldo, André de Palma, Souhila Arib

HAL Id: hal-03886819
https://hal.science/hal-03886819

Submitted on 15 Dec 2022

1

PoolLines: Modeling Carpooling as Ephemeral Lines in GTFS for effective integration with Public Transit

Youssef Chaabouni
youssefchaabouni98@gmail.com
Télécom SudParis, IP Paris
France

Andrea Araldo
araldo@imtbs-tsp.eu
Télécom SudParis, IP Paris
France

André de Palma
andre.de-palma@cyu.fr
CY Cergy Paris Université
France

Souhila Arib
souhila.arib@cyu.fr
CY Cergy Paris Université
France

ABSTRACT

In carpooling systems, a set of drivers owning a private car can accept a small detour to pick-up and drop-off other riders. However, carpooling is widely used for long-distance trips, where rider-driver matching can be done days ahead. Making carpooling a viable option for daily commute is more challenging, as trips are shorter and, proportionally, the detours tolerated by drivers are more tight. As a consequence, finding riders and drivers sharing close-enough origins, destinations and departure time is less likely, which limits potential matching.

In this paper we propose an *Integrated System*, where carpooling matching is synchronized with Public Transit (PT) schedules, so as to serve as a feeder service to PT in the first mile. Driver detours are proposed towards PT selected stations, which are used as consolidation points, thus increasing matching probability. We present a computationally efficient method to represent PT schedules and drivers trajectory in a single General Transit Feed Specification database, which allows to compute multimodal rider journeys using any off the shelf planners. We showcase our approach in the metropolitan area of Portland, Oregon, considering 8k randomly generated trips. We show the benefits of our Integrated System. We find that 10% more riders find a feasible matching with respect to the status quo, where carpooling and PT are operated separately. We release our code as open source.[1]

CCS CONCEPTS

• **Applied computing → Transportation**.

KEYWORDS

Transportation, Carpooling, Public Transit, GTFS, Open Data

[1]https://github.com/YoussefChaabouni/Carpooling

Unpublished working draft. Not for distribution.

1 INTRODUCTION

Mass Public Transit (PT) is irreplaceable to enable mobility in big urban conurbations. However, the level of service offered by PT is generally not satisfactory in the entire urban region [6]. Indeed, in suburban areas PT agencies cannot guarantee high frequency and high coverage service, as the lower demand density would determine an unbearable cost-per-passenger. Flexible modes, which adapt their routes on the trip request observed at a certain moment [4], are a valid complement to conventional fix PT in such areas. Demand-responsive buses have been mostly considered [7]. On the other hand, the benefits of integrating carpooling to conventional PT have been less studied. Some articles proposing such integration are [5, 8, 11]. The first assumes that riders do not necessarily choose the most convenient path but obey to what the system dictates and that carpooling can only occur in the first mile (from the origin to a PT stop) and not in the last mile (from a PT stop to a destination). The second can only match one rider per driver. The third can only handle one PT line only. None of the three has been applied to a real PT network.

In this work, we propose an open-data driven approach to model an *Integrated System*, where conventional fixed PT is complemented by carpooling. Our approach has GTFS data at the core, a standard specification of PT schedules, which allows us to apply it to real PT networks. Our simple key idea consists in representing a carpooling driver trip as an *ephemeral bus line*, which passes only once. We call all such lines *PoolLines*. The advantage of representing drivers as PoolLines into the GTFS database is that the computation of multimodal rider trips, composed of possibly multiple carpooling and PT transit segments, can be done using off the shelf routing software (we use OpenTripPlanner in our case [2, 10]).

We summarize our contribution as follows:

- We propose a modeling approach to describe PT schedules and driver journeys in a single GTFS database [1].
- We compute rider journeys on such a database, using available open-source software. Such journeys can include a chain of carpooling trips with different drivers as well as a combination of carpooling and transit.
- We showcase our approach in the metropolitan area of Portland, Oregon.
- We show that our Integrated System is able to serve more riders by requiring minimum overall detours to drivers, confirming the importance of using stations as consolidation points.

2 MODEL AND IMPLEMENTATION

Table 1 provides the key notations and values used henceforth.

Symbol	Description	Value
τ	Maximum detour ratio	15%
dw	Dwell time (§1)	1 minute
$d \in \mathcal{D}$	Drivers	
$r \in \mathcal{R}$	Riders	
$J(d) = \{b_d^0 = b_d^{org}, b_d^1, b_d^2 \ldots, b_d^{K_d} = b_d^{dst}\}$	Journey	

Table 1: Notation and numerical values.

We consider a set of $r \in \mathcal{R}$ riders, each departing at time t_r from origin org_r and willing to arrive at destination dst_r as soon as possible. Similarly, we consider a set of drivers $d \in \mathcal{D}$, each characterized by t_d, org_d, dst_d. We denote with $\mathcal{PT}$ public transit schedule, with physical stops $\mathcal{S}$. A scenario is determined by $(\mathcal{R}, \mathcal{D}, \mathcal{PT})$.

Given a certain scenario, our Integrated System first computes the journeys of drivers $\mathcal{J}^d = \{J(d)|d \in \mathcal{D}\}$. A journey is a sequence of *stoptimes*

$$J(d) = \{b_d^0 = b_d^{org}, b_d^1, b_d^2 \ldots, b_d^{K_d} = b_d^{dst}\}. \qquad (1)$$

Each stoptime b_d^i is a pair (p, t), where p is a physical location and t is a time-instant. Each journey must have at least two stop times. The first stoptime $b_d^{org} = (p, t)$ must always be located in the origin $p = org_d$ and happen at the departure time $t = t_d$ of the driver. The last stoptime $b_d^{dst} = (p', t')$ must be located at the destination of the driver, $p' = dst_d$. If a drive journey has only two stoptimes, it means the driver does not execute any detour. Otherwise, the intermediate stoptimes describe the entire detour.

After computing $\mathcal{J}^d$, the system computes the journeys of riders $\mathcal{J}^r$ exploiting connections via carpooling with drivers or PT. Multiple riders can share a carpooling trip with the same driver. The same rider is also allowed to execute a sequence of carpooling trips with different drivers. §2.2 explains how $\mathcal{J}^d$ and $\mathcal{J}^r$ are computed.

2.1 GTFS model

We now briefly describe the GTFS specification, on which $\mathcal{PT}$ is modeled [1, 9]. PT schedules are specified as a set of files. The available bus, subway, train or tramway lines (possibly with different service pattern each) are called *routes*. A route is a sequence of *stops*. The same route is served multiple times, each representing a *trip* departing at a specific time. The event of a trip arriving and departing at a specific stop is called *stoptime*. A trip is thus a sequence of stoptimes.

The basic idea of this work simply consists in adding driver journeys as new lines into GTFS data, which we call *PoolLines*. For a driver d, we add a new route named `route of carpooler number [id_of_carpooler]`, a single trip named `1162238700[id_of_carpooler]`. We also add two stops: `DRIVER_origin_[id_of_carpooler]` corresponding to origin org_d and `DRIVER_destination_[id_of_carpooler]` corresponding to destination dst_d. For each stop time $b_d^i = (p^i, t^i)$ in the journey (see (1)), we add a corresponding stoptime with location p^i and time t^i. The trip is thus a sequence of stoptimes corresponding to (1). In other words, we represent a driver journey as a bus that passes only once.

2.2 Computation of driver journeys

We select a subset $\mathcal{M} \subseteq \mathcal{S}$ of physical stops as potential *meeting points*: in such stops, drivers can pick-up or drop-off riders. Such

selection is a design parameter. It is advisable to select stops that are well connected to the rest of PT network. Indeed, by dropping-off riders there, they can reach many destinations from there. On the contrary, selecting a stop from which only one PT line passes with low frequency would be of little use. In our numerical result we select all subway stops as potential meeting points. We will investigate efficient criteria to select meeting points in our future work.

Over the time, new drivers declare their trip to the system, specifying origin, destination and departure time. We assume such declaration occurs some time before the departure time. For any new driver d, we initialize her journey as d to $J(d) = \{b_d^{org}, b_d^{dst}\}$. Let us denote with $l(J(d))$ the length of journey $J(d)$ in Km and with l_d^0 the length of her initial journey $J(d) = \{b_d^{org}, b_d^{dst}\}$, without any detour. The modification of the journey to include detours is similar to our previous work [5]. With 50% probability we first try to add a detour close to org_d, otherwise we first try close to dst_d. When trying to add a detour close to org_d, we take the meeting point $m \in \mathcal{M}$ closest to org_d and we compute the time t at which the driver can be there (the details of driver vehicle movement are described in §3) including a fix dwell time dw, i.e., the time to left riders alight or board. We build a stoptime $b = (m, t)$ and we check whether a new journey $J'(d) = \{b_d^{org}, b_d^{dst}\}$ would be acceptable, which depends on whether such a new journey imposes a too long detour. If $\frac{l(J'(d)) - l_d^0}{l_d^0} \leq \tau$, where *tau* is a designed parameter, the new journey is acceptable and we set $J(d) := J'(d)$. Otherwise $J(d)$ remains unchanged. We then try to add a detour close to the destination of the driver, in a similar way, checking that we do not exceed the original length more than τ times. With the other 50% probability we would instead first attempt to add a detour close to driver's destination and then close to her origin.

We assume a centralized system controller receiving all drivers' declarations and computing their journeys. For simplicity, we assume drivers always obey to the journeys computed by the controller. Appropriate economic compensation, in order for such an assumption to hold, is outside the scope of this paper and is object of our future work. Once a driver declaration arrive, we assume the controller's computation of her journey is instantaneous. In the time window between the arrival time at_d of driver d declaration and her departure t_d, the controller has the opportunity to match riders to d. At time t_d, just before starting the actual movement, driver d checks what are the stop times that are "useful", i.e., the ones in which some rider is picked-up or dropped-off. The driver skips all other stoptimes.

We add the resulting drivers' journeys $\mathcal{J}^d$ into the GTFS files, as described in §2.1, in the form of *PoolLines*.

2.3 Computation of riders' journeys

We give the modified GTFS files as input to OpenTripPlanner [2], an open source software which computes multiple shortest paths from an origin, to a destination at a given departure time. OTP is able to ingest PT schedules in GTFS format and to compute time-dependent paths (the path returned changes depending on the departure time, based on the PT schedule), combining walk and one or more PT lines in a single path. Since some of the GTFS lines
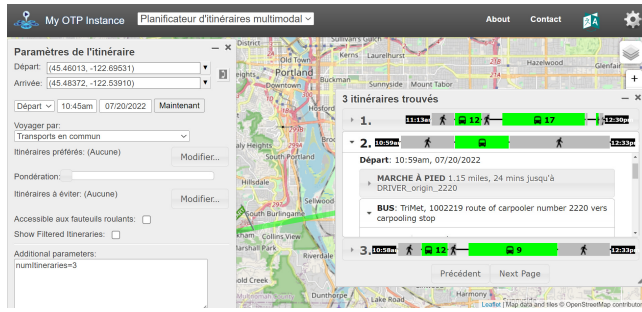
Figure 1: Example of multimodal path as a result of a rider query.

Figure 2: Rectangles used for generating riders and drivers' origins and destinations

are our PoolLines, OpenTripPlanner is able to return paths that include one or multiple carpooling segments.

When a rider r requests a trip from origin org_r, destination dst_r and departure time t_r (such request may arrive directly at time t_r or in advance), the following HTTP query is sent to the OTP server:

```
http://localhost:8803/otp/routers/current/plan?
?fromPlace=      "+str(origin[0])+"%2C"+str(origin[1])
&toPlace=        "+str(destination[0])+"%2C"+str(destination[1])+"
&time=      "+str(hours)+"%3A"+str(minutes)+str(am_or_pm)+
&date=   07-20-2022
&numItineraries=           10
&mode=   TRANSIT%2CWALK&maxWalkDistance=          2500.032
```

. Here, origin[0] and origin[1] correspond to the longitude and latitude coordinates of the generated rider. We also have the same case for destination[0] and destination[1]. Hours, minutes and am_or_pm are variables that express his departure time. We have chosen 10 itineraries so that a single rider can have different options.

2.4 Implementation

For our numerical results, we wrote a script in Python that generates a scenario $(\mathcal{R}, \mathcal{D}, \mathcal{PT})$. First, GTFS data are modified adding drivers' journeys $\mathcal{J}^d$ as explained in §2.1 and §2.2. An OTP server is run, taking as input the modified GTFS. The script than generates for each rider r an HTTP request via the API of OTP, as illustrated in §2.3. Obviously, for scalability reason, no graphical interface is used at this stage. The paths proposed by OTP are stored in Python variables for further analysis. Although OTP can return multiple alternative paths, we consider only the shortest one for each user. If a driver vehicle reaches its seat capacity, which is a quite rare event, both this driver and corresponding riders will be discarded from the results. Although a better method to directly discard the routes directly from the OTP request is a work in progress.

3 NUMERICAL RESULTS

We now show the performance of the proposed Integrated System in a large scale metropolitan area. Results confirm the trends we observed in a previous work [5], where however the lack of a proper unifying model for PT and carpooling was limiting the scenarios to have one PT line only.
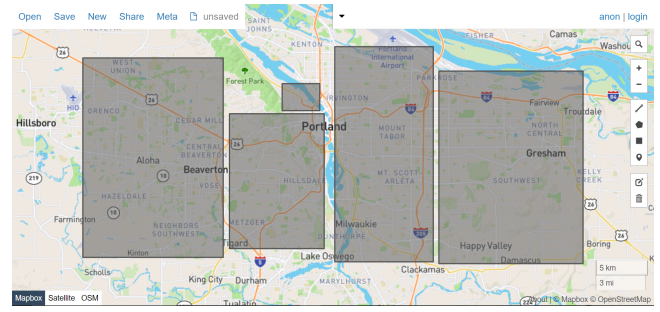
3.1 Scenario

For our experiments, we generate drivers and riders origins and destinations by defining several rectangles in the city's map and then choosing one rectangle at random in which we will generate our point in a uniformly random manner.

We generate with this process a density of 4.8 drivers/km²/hour and 8.3 riders/km²/hour. Given that the generation area is 662.47km², we end up with 2848 drivers and 5498 riders. We then impose on drivers a maximum percentage of detour. The simulation takes place from 10:30 AM to 11:30 AM and we build the results statistics from riders that started their journey from 10:45 AM to 11:15 AM. Other numerical values are reported in Table 1. Values that are not specified here are the same as in [5, Table 1].

For simplicity, we assume all drivers declare their trip at the beginning of the day, before all riders' declaration arrive.

We compare three systems:

- *No Carpooling System*, where carpooling is not available, as it is still common in several urban conurbations.
- *Current System*, i.e., the status quo, where carpooling exists but it is not integrated with PT: either a user journey is entirely done by carpooling or entirely done within PT.[2]
- *Integrated System*, where PT and carpooling are part of the same network: we have conventional lines (buses, subways, tramways, etc.) operating on regular routes and schedules and PoolLines, representing the trips of carpooling drivers. In such system, a user trip request may be satisfied by a multimodal route, combining fixed PT segments and PoolLines.

3.2 Benefits of the Integrated System

Fig. 3 shows the modal split of the three systems.

Unserved riders' trips are the ones for which no feasible route was found. A route is feasible if (i) the waiting time (or the sum of all waiting times in case of multiple segment) is less than 45 minutes, (ii) the total walking distance is less than 2.5 Km and (iii) the total journey time does not exceed the time the rider would take by walking from origin to destination.

---

[2]In theory, also in this system, a user may autonomously request trips from a certain location to a certain PT stop, then choose to which other stop to go and, from them, reserve another carpooling trip. We ignore such possibility since (i) in practice it is very unlikely that a user does this quite difficult computation and (ii) we are interested in the system aspects, instead of user behavior.
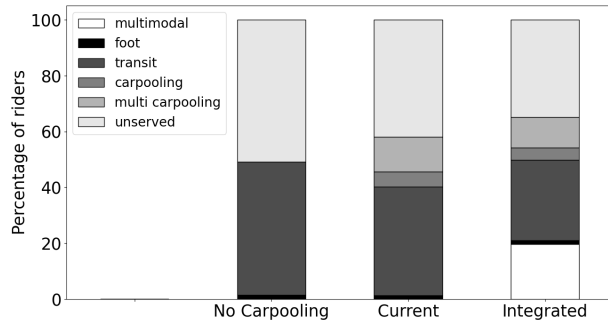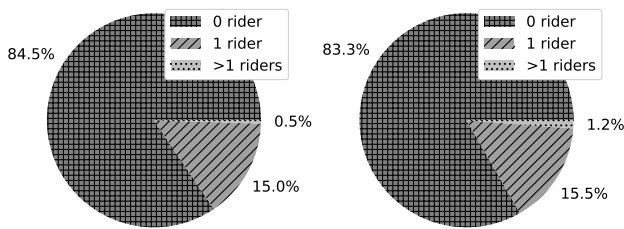
Figure 3: Riders' modal split.

Figure 4: Vehicle occupancy of Current (left) and Integrated (right) systems.

Foot indicates trips done by walking from the origin to the destination of the rider. When a rider trip is done by *Carpooling*, she walks to the meeting point indicated by the system controller, she is picked-up by a driver, she is dropped off at another meeting point and, from there, she walks to her destination. *Multi-Carpooling* indicates the path includes multiple carpooling path, with multiple drivers. *Transit* paths include walk and transit segments. *Multimodal* paths include walk, carpooling and transit segments.

Unserved users would be left without travel alternatives and would be obliged to use their private cars. We observe that Carpooling alone reduces unserved users, but such gains are amplified when integrating carpooling and PT. Our proposed Integrated System thus show potential in reducing car-dependency of areas not sufficiently served by conventional PT. Moreover, by considering the set of riders that are only served in the integrated system, meaning riders that would be unserved in the current system. We can showcase by subtracting the distance they would drive in their private cars by the detours taken by drivers that took them on-board in the integrated system that the integrated system saves *6392* kilometers of vehicle travel and by using [3], we can compute the amount of CO2 saved on average which results in *1240* kg of CO2 saved per hour of simulation.

Fig. 4 shows that our proposed Integrated System enables higher sharing of driver trips. In particular, while trips shared with more than one riders were almost nonexistent in the Current system, they reach 1.2% in the Integrated system. This can be explained with the low probability of having good matching between origin,
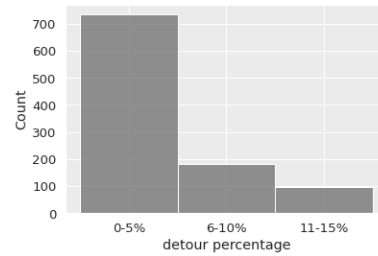
Figure 5: Distribution of drivers' detour (% in the x-axis).

destination and departure times of drivers and riders. Having good matching up to (or from to) a PT stop, instead, is much more likely, since stop aggregate demand, thus acting as consolidation points. Higher sharing of driver trips is important from an economic point of view, since it accumulates payments from multiple riders to compensate the same driver trip. All the above gains are obtained by demanding small detours to drivers as shown in Fig. 5.

4 CONCLUSION AND FUTURE WORK

We have presented an Integrated System, where conventional fixed PT lines are seamlessly combined with carpooling trips. The underlying approach consists in modeling the entire system in a unified model, where everything is a line: subways, buses, tramways lines are operated following a pre-defined routes; trips of carpooling drivers are also represented by lines (which we call *PoolLines*). Such unified model is stored in a GTFS database, which is then used by off the shelf routing software (OpenTripPlanner in our case) to compute multimodal routes. The benefits of such an Integrated System with respect to the Current system, where carpooling and PT are operated separately, are lower levels of unserved riders and higher sharing of drivers' trips.

Some heavy assumption make our work preliminary, as the fact that rider demand is inelastic, the omission of explicit model of private car as available alternative to riders, the lack of congestion models and the lack of payment models.

AKNOWLEDGEMENT

REFERENCES

[1] General Transit Feed Specification. https://gtfs.org/.
[2] Open Trip Planner. https://www.opentripplanner.org/.
[3] Évolution du Taux Moyen d'Émission de CO2 en France. Technical report, ADEME, 2022.
[4] A. Araldo, A. Di Maria, A. Di Stefano, and G. Morana. On the Importance of demand Consolidation in Mobility on Demand. In *IEEE/ACM DS-RT*, 2019.
[5] A. Araldo et al. Pooling for first and last mile: Integrating carpooling and transit. In *hEART Conference*, 2022.
[6] A. Badeanlou et al. Assessing transportation accessibility equity via open data. In *hEART Conference*, 2022.
[7] G. Calabrò, A. Araldo, et al. Integrating Fixed and Demand Responsive Transportation for Flexible Transit Network Design. In *TRB*, 2021.
[8] S. Fahnenschreiber et al. A Multi-modal Routing Approach Combining Dynamic Ride-sharing and Public Transport. *Transp. Res. Procedia*, 13:176–183, 2016.
[9] P. Fortin et al. Innovative gtfs data application for transit network analysis using a graph-oriented method. *J. of Pub. Tr.*, 19:18–37, 12 2016.
[10] OpenTripPlanner. Routing Bibliography. docs.opentripplanner.org/en/latest/Bibliography/.
[11] M. Stiglic et al. The benefits of meeting points in ride-sharing systems. *Transp. Res. Part B*, 82:36–53, 2015.