UNIVERSIDAD DE CANTABRIA

ESCUELA DE DOCTORADO DE LA UNIVERSIDAD DE CANTABRIA
DOCTORADO EN INGENIERÍA CIVIL

TESIS DOCTORAL

ESTIMACIÓN DE LA DEMANDA DE TRANSPORTE PÚBLICO EN ÁREA METROPOLITANA A PARTIR DE LOS REGISTROS DE ENTRADA DE USUARIOS

\- - - - - -

PhD THESIS

IMPROVED ESTIMATION OF PUBLIC TRANSPORT DEMAND IN METROPOLITAN AREAS UTILIZING ACCESS-ONLY TICKETING RECORDS

Presentada por: **JUAN BENAVENTE PONCE**

Dirigida por: **Dr. JOSÉ LUIS MOURA BERODIA**
Dr. BORJA ALONSO OREÑA

Santander, 2021

Abstract

This thesis deals with some of the most common problems that arise from the use of the data from the operations support system of a public transport operator, especially when combining information from different subsystems. It presents a flexible methodology to improve the definitions of bus runs and boarding events, ameliorating the distortions that stem from the simultaneous use of data from the automated vehicle location, automated fare control, and scheduling subsystems. Then, for the quite common case of a system that only registers the access of the passengers to the buses, the trips they carried out and which ride did they choose to materialize them are inferred. To this end, a trip chaining model is defined, incorporating several enhancements from the available literature, and making use of the previously obtained enhanced accounts of passengers' validations and vehicles movements.

Firstly, the definition of each distinct vehicle run and boarding event carried out in the transport system is improved, integrating the information provided by stop-level events from its automated vehicle location and fare collection systems, and scheduling subsystem information at the initial stop of planned runs, if available. The data are structured; and then corrected and completed utilizing several criteria, which include identifying and combining all entries that are linked to the same call of a bus to a stop, and applying a probabilistic approach based on the distributions of travel and dwell times to both event filtering and the reconstruction of incompletely or wrongly registered runs, following a procedure akin to dead reckoning, utilizing as the initial fix the longest sequence of calls compatible with the configuration of the route the vehicle should be following.

In turn the origin and destination stops of the trips performed by transit users are deduced, as well as which runs offered by the public transport system they rode to do so. The trip chaining model implemented to do so benefits from the improved definitions of bus runs and boarding events and includes several contributions from the state of the art to better identify the end of each ride, find the last destination of the last ride of each day, and detect short activities that may be incorrectly be labeled as transfers.

Finally, the passenger trips thus obtained are aggregated and expanded to provide origin-destination matrices for different periods of the year and days of the week, considering the mobility patterns in the city.

A case study is discussed with one year of records from the automated vehicle location, fare collection, and scheduling subsystems in Santander City, Spain, employing captures from different interactive web visualization tools that has been developed for this work.

This document has been written using LaTeX. Its figures were created with Ti*k*Z, PGFPlots, Gimp, Inkscape, QGIS, and several tools developed for this work, based on Bokeh (a Python library).

Part of the work of this research has been submitted as the article '*Integration of automated vehicle location, fare control, and schedule data for improved public transport trip definition*' (Juan Benavente, Borja Alonso, Andrés Rodríguez and José Luis Moura) to a noted journal and is currently under review.

Acknowledgments

I would like to recognize the material resources and data access that have made this work possible; provided by the University of Cantabria, the SETA project (EU H2020 grant id: 688082) the Science and Innovation Ministry (TRA2017-85853-C2-1-R), and Santander City Council (MOBIDATA-SDR).

On a more personal level, I would like to start thanking Juan Pablo, who introduced me to the world of Transportation Engineering. Also Ángel, for his support and knowledge. I need to mention Jorge, who has helped me navigate the requirements to complete this thesis. Ding, Panchamy, and Tin from DiTTlab not only gladly shared their expertise, but also showed me how to be a kind and warm host. José Luis and Borja have been infinitely patient and provided essential guidance and aid. The conversations with Iñaki have yielded many fresh perspectives and useful insights. Andrés has been indispensable to obtain the data this research is built upon, and has also gone out of his way to help as much as he could.

My colleges at SUM$^+$LAB have been encouraging and supportive, sharing their experience from their doctoral studies, or simply trying to cheer me up. My parents have always been nothing but supportive and loving during all these years, putting my needs always first, and I owe them everything I may be able to accomplish. My brother Javier (I miss you and I hope we can see each other soon!) is also a cornerstone of my word, and sharing our experiences these years through the distance has really helped.

I also wish to thank you reader, for taking the time to examine this work. I hope you find it useful and interesting. Writing it surely was both.

Finally, I need to thank Ana, my companion and better half, for all her love and care, which shows through everything she says or does. Many times, she has not only picked up the slack while I concentrated in this thesis but there were other tasks that had to be done, but anticipated to my needs even before I realized them.

Contents

Abstract i

Acknowledgments iii

Contents v

List of Tables ix

List of Figures xi

Glossary xv

Acronyms xvii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Objectives . . . 4
- 1.3 Thesis contribution . . . 5
- 1.4 Thesis outline . . . 6

2 Literature review 9
- 2.1 Intelligent Public Transport Systems . . . 9
- 2.2 Transit destination estimation . . . 14
- 2.3 Research objective . . . 18

3 Methodology 19
- 3.1 Input data . . . 19
  - 3.1.1 Bus stops . . . 19
  - 3.1.2 Travel times lower bounds . . . 21
  - 3.1.3 AFC . . . 21
  - 3.1.4 AVL . . . 21
  - 3.1.5 Schedule . . . 21
  - 3.1.6 Route-level information . . . 22

3.2 Preprocessing . . . 22
3.2.1 AFC . . . 22
3.2.1.1 Create stop groups . . . 23
3.2.1.1.1 Partition by vehicle, rank over each set . . 23
3.2.1.1.2 Partition by vehicle, route, group, and bus stop; rank over each subset . . . 25
3.2.1.1.3 Create stop groups . . . 26
3.2.1.2 Split stop groups in boarding groups . . . 27
3.2.1.3 Output . . . 29
3.2.2 AVL . . . 31
3.2.2.1 Filter out duplicate rows . . . 31
3.2.2.2 Remove rows not linked to a real bus stop . . . 31
3.2.2.3 Identify *trajectories* . . . 32
3.2.2.4 Determine *visit groups* . . . 32
3.2.2.5 Merge entries of each *visit group* . . . 32
3.2.2.6 Identify and remove unfeasible or unrealistic trajectory legs . . . 34
3.2.2.7 Output . . . 34
3.2.3 Schedule . . . 34
3.3 Vehicle runs definition . . . 35
3.3.1 Analyze AVL trajectories as sequences and fragments of routes 36
3.3.1.1 Identify distinct AVL trajectory sequences . . . 36
3.3.1.2 Single out template sequences . . . 36
3.3.1.3 Break down sequences in template fragments . . . 37
3.3.2 Choose link travel times and dwell times distribution models 39
3.3.3 Assemble vehicle runs . . . 39
3.3.4 (Optional) detect and merge instances where a vehicle changed its id mid-run . . . 45
3.3.4.1 Identify pairs of vehicle runs that should be combined 45
3.3.4.2 Update vehicle runs characterization tables . . . 47
3.3.5 Ascribe vehicle runs to scheduled runs and update visit time spans . . . 48
3.3.6 Select vehicle runs backed by enough information . . . 49
3.4 Boarding groups imputation . . . 52
3.5 Passenger trips inference . . . 53
3.5.1 Trip chaining demand models . . . 53
3.5.2 A ‘base’ trip chaining model . . . 54
3.5.3 Trip chaining model improvements . . . 56
3.5.3.1 Alighting stop inference refinement . . . 58
3.5.3.2 Alighting inference for the last trip of the day and other incomplete trips . . . 58

3.5.3.2.1 The Density-based spatial clustering of application with noise algorithm 59
3.5.3.2.2 Last or otherwise incomplete destination deduction 60
3.5.3.3 Trip maximum duration 61
3.5.3.4 Improved activity detection 61
3.5.3.4.1 Detection of alightings close to stops that would have been reachable from one of the previous rides of the trip 61
3.5.3.4.2 Directness check 62
3.5.4 Final trip chaining model 63
3.5.5 Multiple passengers per tap-in consideration 66
3.5.6 Allow passenger rides to span two vehicle runs 66
3.5.7 Trip chaining output 67
3.6 Aggregate trip analysis 69

4 Case study **71**
4.1 Introduction 71
4.1.1 Santander IPTS dataset 71
4.1.2 Route 1 75
4.2 Implementation of the methodology 76
4.2.1 Input data 76
4.2.1.1 Bus stops and subroutes 76
4.2.1.2 AFC 76
4.2.1.3 AVL 76
4.2.1.4 Scheduling information 76
4.2.2 Preprocessing 77
4.2.2.1 AFC 77
4.2.2.2 AVL 77
4.2.3 Vehicle runs definition 78
4.2.3.1 Analyze AVL trajectories as sequences 78
4.2.3.2 Specify travel times and dwell times distribution models 78
4.2.3.3 Assemble vehicle runs 79
4.2.3.4 Merge instances where a vehicle changed its id mid-run 79
4.2.3.5 Ascribe vehicle runs to scheduled services and update visit time spans 79
4.2.3.6 Select vehicle runs backed by enough information . 79
4.2.4 Boarding groups imputation 79
4.2.5 Trip inference 80
4.2.5.1 Introduction 80

4.2.5.2 Model parameters 81
4.2.5.3 Overall results 82
4.2.6 Trip aggregation 86

5 Discussion **89**
5.1 Vehicle runs definition 89
5.1.1 Reconstruction of a vehicle run from fragmented and erroneous information 89
5.1.2 Vehicle id mid-run change 90
5.1.3 No AVL data and wrong vehicle id 92
5.1.4 Handling of alternating route configurations 92
5.2 Treatment of initial termini 93
5.3 Robustness against missing and wrong data 96
5.4 Trip chaining examples 98
5.4.1 Example 1 99
5.4.2 Example 2 99
5.4.3 Example 3 101
5.4.4 Example 4 102
5.4.5 Example 5 102
5.4.6 Example 6 102
5.4.7 Example 7 104
5.5 Desire lines 104
5.5.1 Trips attracted by zone 8 on a working day in school season 104
5.5.2 Trips attracted by zone 8 on a working day in school season 107
5.5.3 Trips attracted by zone 11 on a holiday in school season . . 107
5.5.4 Trips attracted by zone 108 on a working day in summer . . 107

6 Conclusions and future research **111**
6.1 Conclusions 111
6.2 Future research 113

Bibliography **115**

7 Appendices **125**
7.1 Daily OD matrices 126
7.2 Runs from other Santander routes 133

List of Tables

3.1 Methodology input data 20
3.2 Raw AFC preprocessing procedure. *Vehicle*, *route*, and *group UID* remain constant through this example. 24
3.3 AFC pre-processing output: *boarding_groups* 29
3.4 How entries of a trajectory linked to a single visit are identified . . 33
3.5 Treatment of multiple AVL entries from the same visit to a stop . . 34
3.6 AVL preprocessing output 35
3.7 raw schedule preprocessing output: *schedule* 36
3.8 Analysis of the trajectories of a route for one of its template sequences 37
3.9 AVL sequences analysis output 38
3.10 Vehicle runs definition 44
3.11 Trip chaining output: *rides* and *incomplete_trips* 68

4.1 Distribution of activity nodes though the year 83
4.2 Trip ending or splitting criteria occurrences 86
4.3 Destination estimation method utilized for last or incomplete trips . 86
4.4 Transformation from inferred trips to individual trips 86
4.5 Mean daily individual trips across the different periods of the year . 88

7.1 Daily origin-destination (OD) matrix of a working day (winter, spring, and autumn) 126
7.2 Daily OD matrix of a Saturday (winter, spring, and autumn) 127

7.3 Daily OD matrix of a holiday (winter, spring, and autumn) 128
7.4 Daily OD matrix of a working day (summer) 129
7.5 Daily OD matrix of a Saturday (summer) 130
7.6 Daily OD matrix of a holiday (summer) 131

List of Figures

1.1 Error sources when working with IPTS data 2

3.1 Methodology outline 21
3.2 Splitting stop groups in boarding groups 30
3.3 Individual ticketing events to encompassing *boarding groups* transition 31
3.4 Vehicle run inference process from AVL seeds 42
3.5 Vehicle runs inference process 43
3.6 Merging of 2 vehicle runs 47
3.7 Associating scheduled and inferred vehicle runs 50
3.8 Improvement of arrival and departure times of a run once its likely planned beginning is identified 51
3.9 Analysis of the IPTS data that sustain each of two consecutive potential runs of a vehicle and imputation of boarding groups 52
3.10 A 'Round-trip' (left) vs chained trips (right) 54
3.11 Trip chaining model example 57
3.12 DBSCAN algorithm example 59
3.13 Detection of a short intermediate activity thanks to an otherwise counterproductive transfer 62
3.14 Graphic interpretation of the circuity ratio 63
3.15 Trip chaining model flowchart 64

4.1 Santander city 72
4.2 Santander city bus stops 73

4.3 Distribution of raw AVL events 74
4.4 Distribution of SC validations 74
4.5 Distribution of cash payments 75
4.6 Bus stops of route 1 77
4.7 Frequent activities availability ratio along each type of day 83
4.8 Distribution of frequent activities on different day types in winter . 84
4.9 Distribution of frequent activities on different day types in summer 85
4.10 Location of the 113 transportation zones in Santander 87

5.1 Characterization of a vehicle run from fragmented and erroneous information 90
5.2 Characterization of a run when its vehicle id changes while it happens 91
5.3 Vehicle run characterization from AFC data only. Actual vehicle not the planned one 92
5.4 Runs of one vehicle covering Route 3 during a workday in school season 94
5.5 Runs of one vehicle covering Route 4 during a workday in school season 95
5.6 Distributions of abs. departure time errors per gap for scenarios (A) and (C) 96
5.7 Distr. of abs. departure time errors when there is no gap for scenarios (A) and (B) 97
5.8 Distribution of the deviation from the ground truth for different proportions of real and wrong data. Extremes at 1^{st} and 99^{th} percentiles 98
5.9 Trip chaining example 1 100
5.10 Trip chaining example 2 101
5.11 Trip chaining example 3 102
5.12 Trip chaining example 4 103
5.13 Trip chaining example 5 103
5.14 Trip chaining example 6 105

5.15 Trip chaining example 7 106
5.16 Trips attracted by zone 8 on a working day in school season 107
5.17 Trips attracted a Saturday during winter by zone 77 108
5.18 Trips attracted by zone 11 on a holiday in school season 108
5.19 Trips attracted by zone 108 on a working day in summer 109

7.1 Runs of one vehicle following the only configuration of route 2 . . . 133
7.2 Runs of one vehicle on different days that show the configurations of 5C1 134
7.3 Runs of one vehicle following route 11 135
7.4 Runs of one vehicle following route 13 136
7.5 Runs of one vehicle covering Route 17 during a workday in school season 137
7.6 Daily runs of different vehicles that show the configurations of Route 18 to Monte neighborhood 138
7.7 Runs of one vehicle following route 15. 139
7.8 Runs of a vehicle (not the initially planned one) following route 19 . 140
7.9 Runs of one vehicle following route 20 141
7.10 142
7.11 Runs of a vehicle of route 41 142
7.12 Runs of a vehicle of route 42 143
7.13 Runs of a vehicle of route 43 144
7.14 Runs of a vehicle of route 44 144
7.15 Runs of route 61 in a day 145
7.16 Runs of route 6C1 in a day 146
7.17 Runs of route 6C2 in a day 146
7.18 Runs of route 7C1 in a day 147
7.19 Runs of a vehicle of Route 7C2 in a working day in school season . 147
7.20 Runs of route N1 in a day 148
7.21 Runs of route N2 in a day 149

7.22 Runs of route N3 in a day 150

Glossary

Some of these words are described to establish their interpretation within this document.

activity Task or pursuit that motivates a person to move from one place to another. i, xii, 1, 4, 5, 7, 8, 14–16, 18, 53–55, 59–63, 65, 66, 76, 82, 97, 102, 110

day The threshold between one day and the next occurrs at the moment of least activity in the city, just before daily services such as transportation resume. For instance, in Santander it occurs around 06:00. 5, 7, 15, 16, 51, 55, 56, 60, 64, 68, 71, 78, 81–84, 90, 95, 97, 102, 146–148

journey A sequence of trips starting and ending at home [1]. 53, 54

PostgreSQL Popular open-source object-relational database system that uses an extended version of the structured query language (SQL) language [2]. 6, 76

ride Movement of a passenger on a single vehicle from boarding to alighting. i, 3–5, 8, 14–17, 53–56, 58, 60–68, 80–84, 96, 97, 99, 100, 102, 109, 110

run Movement of a transit vehicle through a predefined sequence of stopping points [3]. i, xi, xii, 2–8, 10, 12, 13, 15, 16, 18, 19, 21, 22, 27, 31, 34–37, 39–41, 43, 45–49, 51–53, 56, 58, 60, 66–68, 71, 73, 75–82, 87–91, 94, 95, 100, 109–111

subroute One of the sequences of stops that describe the runs offered in an IPTS. In the case of 'linear' routes, with two termini, there will also be two subroutes that their vehicles will alternately follow; while circle routes with a single terminus are implemented in pairs, running in opposite directions, presenting a single subroute each. 36

tap-in The SC transaction made when boarding a public transport vehicle [4]. 12, 14, 16, 23, 61, 66, 80

tap-out The SC transaction made when alighting a public transport vehicle [4]. 12, 14, 18

trajectory Consecutive AVL records that present the same values for the columns that may be used to tell different runs apart. 6, 22, 31–34, 36, 37, 39, 78, 89

trip Movement of a traveler through the public transport system under analysis. Its beginning and end are linked with the conclusion of an activity and the start of another, respectively. i, ii, xi, 4, 5, 7, 8, 11, 14–17, 19, 53–56, 58, 60–69, 80–85, 96, 97, 99, 100, 102, 106, 108–111

Acronyms

AFC "automated fare control". The collection of components that automate the ticketing system of a public transportation network. 3–7, 10–15, 17–19, 22, 23, 25, 29, 32, 36, 40, 41, 43, 48, 52, 53, 66, 69, 71, 73, 75, 76, 79, 80, 84, 85, 87, 88, 90, 91, 94, 109, 110

AVL "automated vehicle location". The collection of components that provide the position of a vehicle. 3–8, 11–14, 18, 19, 21–23, 27, 29, 31, 32, 34, 36, 37, 39–41, 43, 48, 49, 52, 71, 73, 75–77, 79, 87, 89–91, 94, 95, 109, 110

DBSCAN Density-based spatial clustering of application with noise 7, 17, 59–61, 68, 81, 82, 84, 110

GNSS global navigation satellite system 10, 11

GPS global positioning system 11, 14, 39, 75

GTFS general transit feed specification 2, 10, 13

IPTS intelligent public transportation system 1–4, 6, 7, 9, 12, 13, 16, 17, 19–21, 35, 36, 40, 45, 46, 48, 49, 51–53, 61, 73, 75, 80, 87, 88, 109, 111

NFC "near field communication". Short range communications standard based on magnetic induction. [5] One of the three working modes allows a device to work as a smart card (SC). xiii

OD origin-destination 3–5, 8, 12, 13, 18, 66, 69, 85, 100, 102, 108, 111, 124–129

SC "smart card". A card with an embedded integrated circuit chip. It can store data, run their own functions, and interact with smart card readers. Older technologies require physical contact, while newer ones work wirelessly through near field communication (NFC). [6]. xiii, 5, 8, 11–14, 17, 22, 54–56, 60, 63–66, 68, 73, 80, 82, 84, 99, 102, 110, 111

SQL "structured query language". Standard to manage and retrieve information from relational databases [7]. xi, 6

Chapter 1

Introduction

1.1 Motivation

Technological and social development provided the background that made possible and could make use of what is likely to be the first modern public transport system, in the 17th century in Paris. It offered five routes, fixed schedules, and variable fares according to traveled distance [8]. Even though this early example probably only lasted a few years, as history continued, the pressure from increasingly larger populations who lived in denser urban areas and the concentration of activities stimulated the creation of mass transit systems, progressively more complex and with greater capacity, making use of new technologies as they were developed. In turn, public transportation changes the accessibility between locations in the territory, with great social and economic repercussions.

One important milestone for transit systems occurred during the decade of the 1970s. The increasing travel demand could not be satisfied with just more and better installations and vehicles: the systems had become so complex it was extremely difficult to manage them efficiently. The answer was the creation in Japan of the first intelligent public transportation systems (IPTSs), making use of the computing and communication technologies available at the time to maintain a high quality of service.

These IPTSs coordinate and rely on multiple subsystems, that deal with different aspects of the service: inform users and workers according to their needs, ticketing, scheduling, keeping track of the vehicles, maintenance, customer satisfaction, communications, emergencies management, etc.

Even though IPTSs were created to solve an operational problem (the daily management of the service), the progress of sensing, communications and data science opened more possibilities. Increasingly larger volumes of valuable information

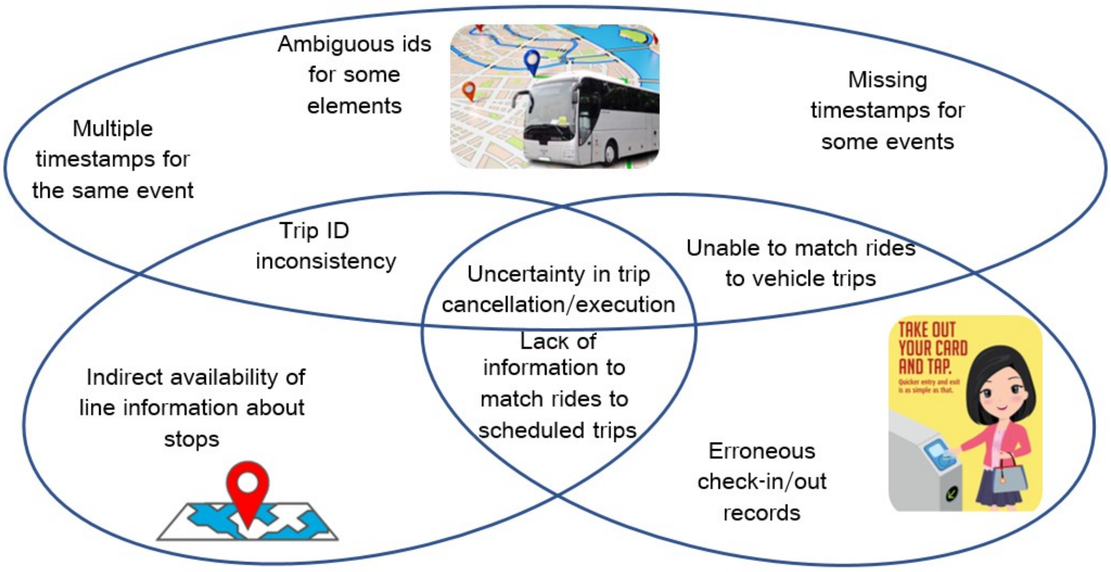


Figure 1.1: Error sources when working with IPTS data [3]

were being generated, which could be retrieved progressively closer to real-time; while data storage and processing became cheaper and more potent.

Thus, besides improved operational responses, these technologies allow to use, at very little extra cost, a by-product of already existing systems to generate useful output for tactical or strategic decisions.

However, several obstacles must be cleared to make use of IPTS data. Service planning and short-term vehicle locations may be available online (an example is the Open Data portal of Santander City [9]), but for longer periods the collaboration of the transit authority may be necessary. This cooperation becomes essential in the case of ticketing data, due to passenger privacy requirements and the reticence of businesses to share sensitive financial information.

Once the data is available, multiple opportunities for research and development arise, being the integration of data from different subsystems to create better models one of them [10, 11]. However, as fig. 1.1 shows, a series of common problems arise: some related to each separate source, others to the fusion process:

- Planning information (lower left):
  - If the schedule was obtained for a source that follows the general transit feed specification (GTFS), only individual runs will be directly available, and not the configuration of the route they are following.

- Automated fare control (AFC) (lower right):
  - Records with wrong or missing information: boarding location, number of validations, alighting time or place (if available), etc.
- Automated vehicle location (AVL) (upper center):
  - Multiple entries for the same event (for instance, the driver may open its doors more than once during a call to allow a last-moment passenger to get on the vehicle).
  - The id of some events or elements may be set improperly or change during a lapse when it should not (e.g., a route with different configurations, and the vehicle not reflecting the change between two of them at the proper time, or the value that should be usable to tell different runs apart not changing accordingly to the vehicle progress).
  - Some events may not be timestamped.
- Planning and AFC:
  - No information to match passenger rides to the scheduled vehicle runs that were boarded.
- Planning and AVL:
  - Inconsistencies or missing values that prevent from matching programmed and materialized rides.
- AFC and AVL:
  - Contradictory or absent records may prevent from matching validations or payments to the vehicle runs that were boarded.
- Planning, AFC, and AVL:
  - It not unusual for AVL data to show a different number of runs than the one that was planned. In this case, AFC information may be considered to identify runs that carried passengers and thus are extremely likely to have occurred.

The method described in this work originated from the necessity to create a representation of the public transportation supply and demand using boardings-only AFC and AVL stop-level data, and the planned (and sometimes recorded) starting times of the runs from an IPTSs where the aforementioned problems made the available dataset initially too fragmented, incomplete, and inaccurate for its intended use (obtaining public transport OD matrices).

1.2 Objectives

Two of the most relevant features for the analysis of a public transit network are the definition of the vehicle runs as they are actually carried out during day-to-day operations, and the rides passengers materialize as they travel between activity locations. The former can usually be defined by one of a previously established set of routes, and the arrival and departure times from each of the pre-defined stop locations; while the latter require knowing where and when users got on and alighted from a bus, and the runs that define the movements of that bus in-between.

The datasets which are useful to build these models may, in most cases, be obtained from the IPTSs. However, as discussed in page 2, several obstacles impede the full benefit that could be gained from these sources. This is the main motivator behind this work, which can be broken down in a series of goals:

- Analyze published research on these topics:
  - The use of IPTS data, looking for encountered obstacles and ways to solve them.
  - Models of call dwell time and travel time between stops of public buses as they perform their runs.
  - Trip chaining models, aiming to build one including several of the enhancements proposed by researchers.
- Create a methodology to enhance the definition of the runs provided by the AVL records of an IPTS, making use of AFC, planning information, and probabilistic models of vehicle travel and dwell times to produce a representation closer to reality, reducing the impact of the issues described in page 2.
- Building upon these improved runs, correct AFC entries that are linked to an incorrect stop or route.
- Implement a trip chaining model to processes the aforementioned runs and trips, applying several modifications from the state of the art to better its output.
- Aggregate and expand the results from the trip chaining model, calculating daily OD matrices during different periods of the year in the city.
- Analyze the case study of Santander (Spain, 173 957 inhabitants), with data that encompass the AVL and AFC events of one year, and the schedule of where and when each planned run should begin.

1.3 Thesis contribution

The main contribution of this research is the development of a methodology to obtain a better definition of the bus runs and passenger rides and trips; combining AVL, AFC, planning information (if available), and probability distributions of bus travel times between stops, and the dwell times during the subsequent calls. It aims to be suitable in situations with different information availability, completeness and reliability. Particularly, scheduled run beginnings may be known or not, optionally including which vehicle had been initially assigned to the task. Other noteworthy aspects of this work are:

- Vehicle runs that have taken place are identified, providing for each call arrival and departure times, as well as identifying the raw data sources that were utilized to make the determination (table 3.10).
- Each cluster of ticketing events is assigned to the corresponding visit of a vehicle, correcting the automatically logged stop and route if necessary (section 3.4).
- Distinction between runs that are part of the planned timetable, and those that respond to operational decisions (section 3.3.6).
- Detection and treatment of instances where the id of a vehicle changes during a run (section 3.3.4).
- As the analysis of the case study shows (page 96), this methodology is robust against missing and erroneous data.
- Specific treatment for particularly problematic termini (page 40).
- A trip chains model has been implemented, using several enhancements proposed in the state of the art to better infer the alighting stop, to solve the problem of the last ride of the last chain of the day, and to better discern when between an alighting and the next boarding an activity or a transfer took place (section 3.5.3). It also considers the case of multiple passengers boarding a vehicle tapping-in the same SC; and the possibility that users may stay inside the bus as it visits a terminus and begins a new run, sometimes with a different configuration of the route.
- Frequent activity nodes, within a region of space and a time window, have been identified. They are used to further infer the destinations of trips, that would not have been fully defined otherwise; and to analyze the mobility patterns in the city.
- The rides obtained from the trip chaining model have been aggregated to provide OD matrices for different periods of the year.

- Several software tools developed to implement the methodology, and to visualize its results, amounting to over 15 000 lines of code. The vast majority is written for the SQL version of PostgreSQL 12 and its procedural language PL/pgSQL, while the rest is Python 3.8.

1.4 Thesis outline

This introduction is followed in page 9 by a literature review, which covers the three main topics of this work: IPTSs, and the exploitation of their data; the modeling of the time public buses spend as they cover a route, divided in the travel time between consecutive stops and the dwell time during each call; and the trip chaining models.

After that, the methodology followed in this work will be presented (page 19). It begins stipulating the required input data (section 3.1).

In turn, it details how the AFC records are treated (section 3.2.1), making use in the first place of the available features of the table to group in 'boarding groups' those entries that should, in the absence of error, allow to classify then according to which run did the users board; and then refining this grouping applying other criteria (limit how long a call can be, and utilizing AVL data).

Then, the preprocessing of the AVL dataset is explained in section 3.2.2. Its initial part also consists in condensing in a single event all entries related to the same call; and it also identifies those rows that are part of unfeasible vehicle travel legs, evaluating if the incoherence can be solved assuming that the dwell time recorded by the AVL is incorrect, and filtering out the rows that define that leg otherwise. Remaining AVL entries are classified in trajectories (consecutive calls that share the same parameters that would allow, in the absence of error, to identify each run that has taken place).

After the preparation of the planning data in section 3.2.3, section 3.3.1, studies these trajectories as atemporal sequences of bus stops, and divided in fragments of the 'template' sequences of stops that full, perfectly defined runs of a route should follow.

Then, section 3.3.2 implements the distribution models for the travel times and dwell times, which are used to identify AVL fragments and AFC 'boarding groups' likely to be part of the same run.

In turn, section 3.3.3 describes how vehicle runs are put together, beginning with AVL fragments of the 'templates' that describe the routes that are offered by the IPTS, processed from longest (more likely to be correct) to shortest (more probable to originate from some of the issues described in page 2). This is accomplished through a process not unlike dead reckoning: using the earliest and latest known events of the incomplete run that is being assembled, prediction intervals are computed (making use of the distributions of the travel and dwell times distributions) for increasingly further away AVL and AFC events that should have been registered by the IPTS. If one is found, it is used as the new fix in the current direction; while unknown intermediate calls are approximated to their most likely arrival and departure times (as also are those in the 'head' or 'tail' of a run, if no reliable IPTS entry could be found in the corresponding terminus). The behavior of the methodology can be adapted to deal with, as it is not uncommon when working with IPTS data, termini where the AVL entries are particularly unreliable.

Next, in section 3.3.4, an optional step is provided to deal with the issue of vehicles that change their ids during a run.

Section 3.3.5 then shows how to link these inferred runs with the planning of the services, if available. This new information is utilized, to improve the definition of the runs found by the methodology.

The inferred runs obtained thus far are filtered in section 3.3.6. If more than one alternative is found at the same time for a vehicle, the one backed by the most evidence from the IPTS is kept; while in other cases a threshold is established to tell those runs that happened from those that just appeared in the databases due to some error.

To conclude the part of the methodology that improves the IPTS data, section 3.4 checks that the 'boarding groups' deducted from the AFC are compatible with the runs that have been defined. If not, the information provided by the IPTS is progressively disregarded (first, only the bus stop where the event happened, and if that is not enough also the route the traveler supposedly got on), until a match is found or the 'boarding group' is deemed a probable error.

Then, the following section of the methodology (3.5) details the trip chaining model that has been implemented, starting with an introduction and an example sections 3.5.1 and 3.5.2.

After that, the different improvements that have been added to it are described, such as the use of 'generalized time' [12] to better choose the most likely alighting stop, or establishing an upper limit on how long a trip may last (sections 3.5.3.1 and 3.5.3.3).

Other enhancement worth mentioning is the use in 3.5.3.2.1 of Density-based spatial clustering of application with noise (DBSCAN) to find likely destinations for those trips that could not be completely defined with the 'first origin of the same day' and 'first origin of the next day' approaches.

Two other improvements are defined, to deal with one of the issues of trip chaining models: how to tell apart short activities from transfers. To this end, two additional criteria have been formulated.

On one hand, if while building up a trip, it is detected that the user could have reached earlier an inferred alighting point (directly, or getting off at a close stop) by staying longer on a previous ride, the program concludes that an intermediate activity has occurred, and 'cuts' the trip downstream said ride, between the alighting and subsequent boarding that show the greater leeway, considering when they happen, and the time the traveler needs to walk from the former to the latter.

On the other, if after a trip has been completely defined it is deemed too circuitous (3.5.3.4.2, [13]), it is divided in two, following a reasoning similar to the previous point to decide where.

The methodology concludes in section 3.6, explaining how OD matrices are created, aggregating the trips obtained by the previous step, and applying an expansion factor to take into account SC entries for which a trip could not be inferred, and those users that paid with cash and thus cannot be tracked.

Then, the case study of Santander (chapter 4) is presented, describing the dataset and the peculiarities that have been encountered during the step-by-step application of the methodology, and the values adopted by its parameters.

Chapter 5 contains a discussion of the different results that have been obtained from the case study, verifying the behavior of the different programs. It is worth highlighting the study from section 5.2, that corroborates the design decisions pertaining who termini data is treated; and section 5.3, which shows that the methodology succeeds at improving the definition of the AVL runs, with an increasing resistance to bogus data as more good information is available.

Finally, chapter 6 shows the conclusions of this thesis, and lays out future lines or research.

Chapter 2

Literature review

2.1 Intelligent Public Transport Systems

IPTSs are composed by a series of hardware and software elements that function together to provide the means to identify, regulate, and manage the available elements of an urban public transit network in real time (infrastructure, vehicles, users, etc.) [14]. They were made possible by combining on-board-computers and the increasing capabilities of communication technologies during the decade of the 1970s, where the common answer at the time, which consisted in more investment in physical infrastructures and vehicles, could not cope with the growing travel demand. Initially, their purpose was solely to monitor and control operations, but as real-time communication and location technologies have become more available, the approach has been shifting from large periodicity, asynchronous acquisition methods to real-time systems, making IPTSs useful tools to attain a reliable mobility, sustainable both economically and environmentally [15, 16].

They are built upon various sub-systems that oversee different tasks such as providing information to travelers or workers, managing incidents and special events, locating the vehicles, scheduling, ticketing, passenger counting, geographic information, payrolls, maintenance, weather, customer satisfaction, or communications. If successfully implemented, they increase the quality of service, decrease operating costs, improve the decision-making process, and facilitate fleet management [17]. Four key aspects characterize the information they provide [10]: spatial and temporal detail, coverage (all events or only exceptional ones), representativeness (fleet penetration and data recovery rate), and quality.

Stop-level records, which can be stored in the IPTS at a fairly low incremental cost, have allowed to better estimate previously utilized performance indicators and usage metrics (e.g., travel times) [18], and also to assess previously nigh impossible to quantify attributes due to data scarcity, such as those related to service reliability [19]. However, they may require a significant effort to attain meaningful conclusions [20]. Also, adequate visualization tools are needed to be able to comprehend the vast amount of output that can be generated. [21].

Setting aside those cases where fixed timetables are not available (e.g., bus routes in Jinan, China, with high uncertainty in travel times, multiple agencies, and a departure schedule that changes according to on-site observed demand, where a study employed artificial neural networks for improved real-time bus arrival estimation based on AFC and historical vehicle location information [22]), the data that describe transit services (i.e. routes, their schedules, and where the stops are) are published in advance. The most widespread tool to do so is the static component of the GTFS [23].

However, even though an extension has been proposed [24], this format cannot yet represent some real-time changes, such as defining additional runs. Also, transportation agencies may not keep a compilation of these files through time, though in some cases they can be obtained from a third party (for example OVapi [25], Transitland [26], or OpenMobilityData [27]). And finally, GTFS data may not be accurate enough for some applications, such as accessibility measurement ([28]).

Some of the available vehicle location technologies are [29]:

- Global navigation satellite system (GNSS), which relies on a constellation of satellites providing signals from space that transmit positioning and timing data which can be used by receivers mounted on the vehicles to determine their location [30] through trilateration, typically within a 5 m radius under open sky. Factors such as satellite geometry, signal blockage, and atmospheric conditions [31] worsen its accuracy.

- Signpost tracking systems, that rely on short-range communications (optical, magnetic, radio, etc.) between the vehicle and series of known signposts to infer the location of the vehicle. They require fixed routes and continuous maintenance, providing the position of a vehicle with an error between 1 m and 20 m.

- Ground based radio systems, where the receiver on the vehicle utilizes the signals broadcast by one or more antennas to deduce its location, within a 30 m to 400 m radius, due to interference from other radio transmissions.

- Dead-reckoning, which is the process of calculating the location of a moving object combining an initial known position (the fix), and a registry of traveled distances and directions (e.g., from a wheel rotation monitor and an on-board compass). Its precision is strongly dependent on the contact between the road and the wheel, and decreases as the vehicle travels away from the fix.

GNSSs are widely the most employed location method in public transportation, being global positioning system (GPS), owned by the U.S. government, the prevalent alternative. Besides other applications such as identifying headway irregularities [32], implementing more intelligent vehicle priority strategies [33], and fleet management and operations [34]; GNSS location information is used to initially estimate and finally identify the arrival of the vehicle to each point of interest.

Arrival time predictions can be communicated to passengers and drivers, and employed to make better operational decisions. They require to link a vehicle's real-time state data with the schedule it is following and the historical records representative of the same situation (working day or holiday, season, time of the day...); and can be carried out following different approaches: artificial intelligence, Kalman filtering, support vector machines, regression analysis, time series modeling, etc. [35].

Since requiring to register a state where the bus is completely still next to a stop to assert that a visit is happening could require too frequent updates, with their associated network traffic; the actual arrival event is usually equated with the vehicle being detected inside a relatively small region that encloses the stop, while its velocity is lower than a threshold [36]. This event is stored in the AVL database, including besides its timestamp and bus stop identifier other possibly useful information obtained from on-board sensors (e.g., dwell time, route identification, door opening and close times, etc.).

AFC systems have as main purpose to improve the revenue collection process, but they also provide valuable data, especially when enriched with the user tracking and characterization possibilities of SC technology.

In essence, SCs consist in a micro-chip embedded in a plastic card. They rely on an external power source (utilizing metal connectors on the surface of the card or contactless technology) to update its storage and, if necessary, to locally execute small programs, while enforcing defined security constraints. [37].

As consequence of their widespread use, mobile phones are adopting the roles of SCs, including those related to public transport, through mobile applications or their own near-field-communication capabilities [38]. Still, this change confronts obstacles such as the need to attain a minimum return on previous investments, fraud risk, the requirement of being able to work while the phone is off or uncertainty due to future technological changes [39, 40].

This information can be useful at the operational, tactical, and strategic levels of public transport management, with multiple applications [41] such as passenger behavior modeling [42, 43, 44], event-based multi agent simulation [45], vehicle load profiles [46], quality of service assessment [47], constructing transit origin-destination matrices [48, 49], or estimation of passengers' excess trip time [50].

Obstacles to fully benefit from this source of information are the reluctance of farebox manufacturers to ease communications with other on-board devices to prevent fraud, the disinclination of the operator to share business-sensitive details [51], and that a validation may not be required to exit the system (tap-in only configuration). For instance, in a study with data from Guangzhou (China) [52] researchers decided to develop a methodology to extract bus boarding and alighting information from access-only raw SC data that does not identify the stop where it happened, combining the identification of runs direction, boarding cluster, boarding stop, and alighting stop (utilizing a series of criteria that build upon the trip chaining model).

Another example is the use of data from a tap-in, tap-out public transport network in Singapore [53] where researchers explore the reasons why AFC may provide incorrect information, and propose how to identify these erroneous entries and their source.

Scientists from Brisbane (Australia) and Hong Kong (China) have published a review in the field of transit OD estimation using SC data [54], where AFC data cleansing is identified as the first obstacle to solve, identifying sources and types of errors, and classifying boarding stop estimation problems based on which features are available in the SC data.

The subsystems that contribute to an IPTS often fail to properly capture information that would be useful for later analysis, because they usually have other goals: to support tactical planning and emergency response in the case of AVL, and manage concessions for AFC. Consequently, a series of issues commonly arise, related to internal problems of each dataset or inconsistencies between them. Those within the scope of this work are [3]:

- Erroneous AFC records, which can be caused by malfunctions, atypical traveler behavior, emergency route detours or mishandling of the equipment by drivers and operators [11].
- Wrong AVL entries due to system failures, incorrect driver operations or termini-specific issues.
- Multiple records for the same AVL event, possibly with different attributes (timestamp, vehicle, or route identification).
- Lost AVL or AFC events.
- Inconsistent or missing information for the same element along different tables. For instance, this hampers matching passenger rides with materialized and planned runs.
- Uncertainty regarding whether a programmed run actually took place.

In some cases, these problems can be so severe that researchers have developed methodologies that model public transport features indirectly, instead of using a more immediate, but error-prone alternative (e.g., utilizing AVL instead of AFC or automated passenger counter records to estimate public transport demand [55]).

There are many published examples of the combined application of multiple automated collection data systems on the different aspects of urban transit management and planning. Among those that utilize AVL and AFC data, some noteworthy examples are:

- Space-temporal load profiles of urban transit vehicles during a month in The Hague (Netherlands), fully integrating GTFS records as a third data source with AVL and AFC check-in and check-out information [3].
- Offline processing of automated train tracking and magnetic card-based fare collection systems in San Francisco Bay Area (USA) [56].
- Estimation of OD matrices and path choice models for rail passengers of the Chicago Transit Authority [19].
- Metro and bus OD matrices, speed profiles of vehicles and quality service indicators, etc. for the Transantiago public transport system in Santiago de Chile [57].
- "Driver assisted bus interviews": if SC records are correctly linked with AVL information, they can function as revealed preference surveys [58].
- Tracking SCs along metro and bus to identify transfer behavior in Shenzhen (China), making use of bus AFC records that only show card id and sweeping time [59].

Each run performed by a bus in a IPTS can be conceptualized as a path that starts at a first stop, continues as the bus calls at midway stops, and ends at a final one. From a spatiotemporal perspective, it can be regarded as a concatenation of sections [60], where each of them encompasses the time between arrivals at two (non-necessarily consecutive) stops; or as a series of calls at consecutive stops and traveling the links between them [61]. In the latter case, dwelling time at each stop depends on the number and characteristics (special needs and payment mode) of alighting and boarding passengers, how long door operations take, etc. [62]; while link travel times are affected by the available infrastructure, service management, traffic flows, driver behavior, weather, etc. [63].

Several probability distributions are proposed in the existing literature to characterize the variability of link travel times [61] such as shifted log-normal, log-normal, normal [63], gamma, Weibull, Burr Type XII [64], generalized extreme value [65], etc. Numerous real-life studies [66, 61, 67, 68] choose the former, which shows a probability density of zero when the value of the random variable falls below a threshold (which would be the free-flow link travel time) and can adequately fit asymmetric, positively skewed data; and that for many links is the function that most likely describes how travel times are distributed.

A study conducted on GPS data from taxis during the morning peaks of 5 weekdays in Wuhan (China) [69] found that link travel times may be best represented by log-normal, gamma or normal distributions (on 50 %, 30 %, and 20 % of the analyzed links, respectively) and opted, to avoid computationally intractable calculations, to assume that travel times along a path can be approximated by normal distributions.

Regarding dwell times, the majority of works suggest that, due to their non-negative nature and possible skewness, the log-normal distribution is likely to be the best alternative (e.g., a study of 18 months of data from a bus route in New Jersey, USA [70]; 6000 records from a one-day study in Changzhou, China; or an analysis of 1-month data from public buses in Jinan, China [71]).

Other possible distributions are normal, used by commercial traffic micro-simulation software such as Aimsun [72] or Vissim [73], and also chosen in some scientific work (e.g., to characterize 1-day data from a bus stop in Chennai City, India [74]); Wakeby, which outperformed the log-normal distribution in a study with 3 months of data from 4 stops in Auckland, New Zealand [75]; or Erlang, proposed in a study that analyzed 435 records from 12 bus stops in Shanghai, China [76].

2.2 Transit destination estimation

SC tap-in and tap-out events can provide the time and location of the boarding and subsequent alighting of a user from a public transport system, but the latter is not available in many cases. The most prevalent methods to infer these missing alighting events are based on deep learning, and trip chaining models.

Deep learning models, though complex, can take into account very comprehensive factors, and can infer individual passengers trips. However, they require large datasets and are more appropriate for entry-exit systems [77]. For instance, a work with 120 days of data from the Beijing Metro Airport Line (China) [78] manages to provide short-term predictions (15 min, 30 min, and 60 min) of passenger volumes.

On the other hand, trip chaining models also provide individual trips, and the core algorithm is comparatively simple. They can be applied utilizing only AFC data; though several improvements regarding alighting stop selection, and telling activities and transfers between rides apart, rely on AVL information.

A trip chaining model [79] has been chosen for this work and is explained in detail in section 3.5.1. Besides an estimation of where and when each ride ends, it also identifies which boardings and alightings respectively follow or precede an activity (signaling its end or beginning), and which just delimit different rides of the same trip.

Multiple studies have been carried out utilizing these models, being the aim of many of them to improve its performance or applicability [47]. For instance, a study with 2 months of data from Gatineau (Canada) transit authority [80] proposes as a possible destination for the last trip of each day the stop of the first boarding of that same day, or of the next one; and utilizes historical records in those remaining cases that could not be matched to a likely alighting stop, searching for similar regular rides (same initial stop, route, and around the same time).

In a study based on data from 5 days from 4 bus routes in Guangzhou (China) [52], scientists firstly had to preprocess the AFC data to deal with missing boarding stop and route direction information. Then, they managed to detect when and where the user got off for parts of the available data, inferring the alighting stop by applying their trip chaining model to the first applicable of three levels of selection criteria: later same day, use a transaction from the peak hour period (more likely to reflect a regular pattern), or the closest to the terminal station.

A thesis from the University of Queensland [4] utilizes data from Brisbane (Australia) to study the assumptions of these models and the sensibility of their parameters; model trip purpose in a multi-modal system; and propose utilizing as the destination of the last trip of the day, instead of directly the location of the first boarding of said day, the call of its last ride closest to that first boarding.

Research from the University of Chile with 2 weeks of data from Transantiago [12], the public transport system authority in Santiago (Chile), points out the unrealistic user behavior that stems from always choosing as the alighting point of the previous run the closest stop to the next boarding; and formulates an alternative criterion based on a minimizing a 'generalized time', which has the effect of favoring choices that tend to maximize the time the user has for non-travel related pursuits, but penalizing walking in favor of continuing on the bus to a parameter-controlled degree.

The analysis of 6 days of data from the Chicago Transit Authority [81] led scientists to suggest an improvement to the reasoning behind destination detection in multi-modal systems. If a user starts the day riding a train, followed by two different rides on the same bus route, and at last takes a train again; the 'mirrored' bus routes support utilizing the later train ride to infer the destination of the former.

It can be difficult for the trip chaining model to tell apart 'short activities' bound to a location from transfers during a trip, especially if fare policies reward users that hasten their next ride, and frequencies and routing alternatives provide them a greater opportunity to do so. These 'short activities' are not the same as incidental or opportunistic actions, which the user would indifferently perform during any transfer of a trip (for instance, buying a magazine at a station during the trip home would be an incidental action, while hopping to the city center to pick a parcel and quickly board another bus would be a 'short activity'). Thus, many studies propose ways to tackle this issue.

Some researchers opt to fine-tune the maximum span between tap-ins part of the same trip, which they conclude is linked to the size of the IPTS: 90 min in a case where the operator manages a fleet of over 1000 buses ([82], with 10 % of observed tap-ins during a transfer being 72 min or longer apart); 30 min for Bardford and Southport (United Kingdom, populations of 500 000 and 90 000, respectively).

In other cases, the distinction between transfer or activity is made considering the gap from the alighting to the next boarding. This is the case in a study with 2 weeks of data from New York City Transit (USA), that sets a threshold of 18 min; or in the aforementioned work in Santiago de Chile ([12], 30 min). The analysis of data from 1 week period from Seoul (South Korea) shows that more than 80 % of transfers occur in less than 10 min [83].

Some works use more complex criteria. For instance, in a paper that utilizes bus data from a 1-week period from Seoul (South Korea) [46], if users did not get on the earliest run they could have boarded of the route they finally chose for their next ride of the day (allowing for a leeway of 5 min), or if they took the same route as in their previous ride, the researchers conclude that an activity happened in-between.

Many other examples [84, 13, 85] share the idea of establishing an array of checks consecutive rides must pass in order to add up to a single trip, inferring the existence of an intermediate activity otherwise. These conditions range from requiring a minimum distance between the origin and destination of the tentative trip; not entailing a geometrically overly circuitous path or, in finer detail, in a series of rides that exceed in more than a given threshold the fastest travel time possible through the IPTS at that moment; or complying with an upper limit to the ratio between time spent transferring (from previous alighting to next boarding) to total travel time.

Another problem of this model that has induced multiple studies is how to infer a destination for the last ride of each day, or for those rides where there is a next one, but it is not deemed adequate to infer the alighting of the previous (for instance, it may be too far way). Two common answers for the former case are to try to use the first boarding of the day, or the first ride of the following day (both later refined as the closest stop to it of that latest untreated ride). For those rides still without an inferred alighting, research has focused on studying, with a gradually increasing level of aggregation, the mobility patterns of the users, to find the likely destinations of their travels.

For example, access and usage patterns (walking distances, frequency and consistency of daily travels, customer behavior) have been calculated at residential area level with one month of data from the Chicago Transit Authority (USA) [86].

A model to predict which locations people visit has been created from the records of 626 public transport users during a period of 3 months in London (United Kingdom) [87]. Its main findings are that, from a spatial perspective, fixed probabilities can be assigned to the most frequent locations each user visits; and when visiting other places, they mostly select popular places in the city; while from a temporal point of view, the distributions of how long people stay at their most and second most visited destinations present in the first case two distinct peaks at 9 h and 14 h, and a mode of 9 h in the second, consistent with the activities 'home', and 'work', respectively.

In another example, using card type ('adult', 'elderly', or 'student') and the ability to access specific parts of the network without penalty ('regular', 'interzone', or 'express') to cluster in 5 categories the records from 7118 SCs during 9 months from Gatineau Transit Authority, scientists were able to create travel profiles for each group [88].

It is also worth mentioning the analysis of a set of 5 typical weekdays of SC data from Metro Transit in the Minneapolis/St. Paul metropolitan area (USA) [89]; in which researchers built the training set for a classification decision tree utilizing behavioral and heuristic rules, which can then be used to infer the purpose of other trips according to their class.

A study based on the data from the weekdays for one month provided by the Outaouais Transport Society [90] located the 'anchor points' of each individual using 'hot spot' analysis, identifying concentration of events through spatial, temporal, land utilization and IPTS features.

Other researchers opt for the widely utilized DBSCAN ([91], described in 3.5.3.2.1) to extract the spatial and temporal patterns from AFC data. For instance, analyzing 4 months of multi-modal data from South East Queensland (Australia) [42]; or 5 weekdays of SC records from the Beijing Transportation Research Center (China) [92].

Finally, a probabilistic approach, based on a three-dimensional latent Dirichlet allocation model, has been tested with SC data from 10 000 randomly selected passengers with at least 20 observations during a 3-month period from Guangzhou Metro (China) [93]. Besides capturing essential latent passenger behavior patterns, comparing the output of the model with the available tap-out information shows an increased accuracy in destination estimation.

2.3 Research objective

To the best of the author's knowledge, there is room for improvement in the methodologies to apply in situations where AFC, AVL, and schedule data are available, but they are particularly challenging to use to full advantage: information of varying reliability to differentiate runs within each of the 3 subsystems, but no direct way to identify entries of different subsystems that describe the same run; missing of wrong AVL entries, AFC with wrong state information, or failing to correctly identify the current stop; only the planned (and sometimes, the detected) starts of the runs available from the scheduling subsystem, which may be stipulated at a stop 'downstream' the initial terminus of the route; users not requiring to check-out when leaving a vehicle; or unplanned runs that respond to daily operational decisions and are not shown in the schedule of the system.

This work has been written hoping it will be useful to other researchers and transportation engineers during their activities; such as auditing, obtaining transit origin-destination matrices and travel patterns, user behavior modeling, or estimation of vehicle load profiles. Specifically, this thesis aims to use these improved definitions of vehicle runs and boarding events to, through the application of a trip chaining model, including several of the improvements found in the existing literature, analyze the evolution of the mobility in a city through the year, obtaining OD matrices for different day types, as well as frequent activity destination nodes, defined by a spatial location and the time window when they start.

Chapter 3

Methodology

This section begins specifying the sources and expected structure of the input data. Then (page 22), it details the preprocessing steps that are applied to AVL, AFC, and planning information; representing each visit of a bus to a stop as a single event from each source. This is followed in page 35 by the improved definition of the runs that the vehicles of the IPTS have carried out. After that (page 52), AFC events are assigned to bus calls. Subsequently (page 53), the trips performed by the passengers are inferred using the trip chaining model. Finally in page 69, travelers movements are aggregated to analyze the evolution of public transport demand and bus occupation. Figure 3.1 shows an overall summary of the whole process.

3.1 Input data

Table 3.1 contains a summary of the required bus stops, AFC, AVL, travel times lower bounds, schedule, and route-level data. It is worth noting that the ids of *bus stops*, *routes*, and *vehicles* need to be consistent throughout all the subsystems. The *group* columns in the AFC and AVL data should contain a unique identifier for each set of values from other columns present in their respective subsystems that can help to differentiate between runs of a vehicle.

Regarding the schedule, the methodology is designed to work even when it is incomplete, or to detect unplanned runs. This section assumes that the three columns with temporal information may be available in at least part of the dataset.

3.1.1 Bus stops

The location and name of the bus stops are needed. Ξ is defined as a set composed by tuples ξ_i, which represent each of these entries, differentiated by a

Table 3.1: Methodology input data

Column	*Type*	*Description*
id	integer	Bus stop id used throughout the IPTS.
location	(real, real)	Geographical coordinates.
name	text	Human-readable name.

(a) *bus stops*

Column	*Type*	*Description*
bus_stop_0	integer	Initial stop id.
bus_stop_1	integer	Final stop id.
free_flow_tt	time	Lower bound of the time a bus needs for this trip.

(b) *travel times lower bounds*

Column	*Type*	*Description*
bus_stop	integer	Bus stop id.
route	integer	Route id.
vehicle	integer	Vehicle id.
instant	timestamp	Validation time.
group	integer	UID for each set of values from other columns that differentiate distinct visits of the vehicle.

(c) *raw AFC*

Column	*Type*	*Description*
bus_stop	integer	Bus stop id.
route	integer	Route id.
vehicle	integer	Vehicle id.
stop_duration	time	How long the bus stayed at the stop.
instant	datetime	Arrival time to the station.
group	integer	UID for each set of values from other columns that differentiate distinct visits of the vehicle.

(d) *raw AVL*

Column	*Type*	*Description*
line	integer	Line id.
vehicle	integer	Vehicle id.
bus_stop	integer	Bus stop id.
planned_start	timestamp	Planned bus departure time.
recorded_arrival	timestamp	Detected bus arrival time.
recorded_start	timestamp	Detected bus departure time.

(e) *raw schedule*

Column	*Type*	*Description*
route	integer	Route id.
unreliabl_trm	boolean	True if undependable AFC and AVL at termini.
trp_strt_crrct	time	Mean span from AVL entries to trip start detection.
max_hdwy	time	Uppr. bound of the interval separating consec. trips.
max_trvl_lg	time	Travel time between consecutive stops uppr. bnd.
min_rnd_trp	time	Lower bound of how long a roundtrip takes.

(f) *route-level information*

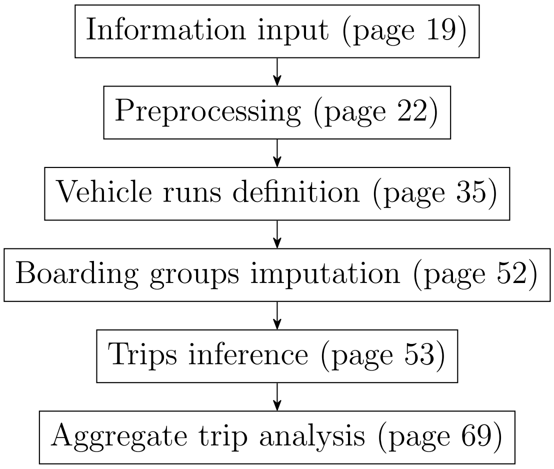


Figure 3.1: Methodology outline

unique *id* m_i (other variables not shown):

$$\begin{aligned} \varXi = \big\{ \xi_i &= (m_i, \ldots) \big\} \\ i : \text{unique row id} & \quad i \in \mathbb{Z} \\ m : \text{bus stop id (bus stops info)} & \quad m \in \mathbb{Z} \end{aligned} \tag{3.1}$$

3.1.2 Travel times lower bounds

AVL events that imply impossible vehicle movements are recognized and filtered out with a table of lower bounds for the travel times between stops.

3.1.3 AFC

The methodology makes use of the ticketing system records: when did each transaction take place; which vehicle was boarded at which stop; and, if available, other non-temporal columns which can help to tell entries from different visits apart.

3.1.4 AVL

A registry of the visits of the buses to the stops is needed, including fields that provide temporal information, and that help differentiate the different runs of each offered route.

3.1.5 Schedule

This methodology utilizes the planned beginnings of runs along each route, characterized by which vehicle was going to be used, and where and when they start. Two other timestamps may be recorded by the IPTS, corresponding to the detected arrival and departure of the bus to the first stop of the new run.

3.1.6 Route-level information

Several aspects that describe each route as a whole are also used (eq. 3.2):

- Whether the timestamps of AVL and AFC events at the termini, which are more prone to suffer the problems described in the literature (page 12), are particularly unreliable (y).
- If there is a systematic time deviation between the events recorded in the scheduling and AVL subsystems, the earlier can be corrected by the appropriate constant value z. This may happen for instance if the AVL stores when the doors of the buses close, while the scheduling subsystem registers the moment vehicles cross certain geofence.
- An upper bound of the headway between runs during normal operations, s.
- An upper bound e of how long a route leg connecting consecutive stops may last.
- A lower bound d of how much time a vehicle needs to come back to a stop after traveling the whole route.

$$\begin{array}{ll} y: \text{termini are unreliable} & \text{boolean} \\ z: \text{run start detection lag} & \text{time} \\ s: \text{headway upper bound} & \text{time} \\ e: \text{leg upper bound} & \text{time} \\ d: \text{round trip lower bound} & \text{time} \end{array} \tag{3.2}$$

3.2 Preprocessing

This section specifies how to process raw AFC (page 22) and AVL (page 31) datasets, synthesizing in a single entry the information that each provides regarding a bus call.

In the case of AVL, lower travel time bounds are used to filter out unreliable data, and remaining entries are classified in trajectories (page 32).

Finally, the analysis of the scheduling subsystem extracts the available arrival and departure time information at the first stops of the planned runs, and time buffers to match them to their corresponding detected runs (page 34).

3.2.1 AFC

It is assumed that there are no duplicate rows in the raw AFC information, since due to the monetary repercussions of the data, ticketing information is managed in a very careful way. SC and manual payment operations are atomic: they are either completed successfully or do not happen.

As is explained in detail in section 3.3.3, AFC information (which provides one data point per validation) is used to deal with the limitations of the AVL data (ideally, one datum per bus visit). Thus, the objective is to classify as a single boarding event all the validations that happen each time a bus calls at a stop. The first and last ticketing events of these "boarding groups" can be used as an approximation of when the bus arrived and left the stop. To create them, a three-part process is carried out:

- For each *vehicle*, *route*, and *group*; identify as a "stop group" each set of consecutive AFC events (section 3.2.1.1).
- Some *stop groups* may contain payments or validations from unrelated events (for example two tap-ins of the same *stop group* may happen too far apart from each other, or the AVL data could have registered a call at another stop in between them). Two criteria are used to identify these instances, splitting *stop groups* in *boarding groups* (section 3.2.1.2).
- Gather the results in the table *boarding_groups* (section 3.2.1.3).

The rest of this section details and exemplifies each of these steps.

3.2.1.1 Create stop groups

The AFC records pertaining each bus are analyzed, distinguishing *stop groups* of consecutive entries referring to the same *stop id*.

This procedure relies on the fact that, as is represented on table 3.2, for a set (in this case, the raw AFC entries linked to a single *vehicle*) where a relation ('*happened before*') can be used to establish rankings over the whole set (column 'rank over set') and also over the different subsets defined by a partition (entries with the same *vehicle*, *route*, *group*, and *bus stop id*; column 'rank over subset'), the difference between the rank over the whole set and over a particular subset (column 'classif. variable') provides a distinct value for the members of that subset that appear consecutively when ranking all elements of the whole set (the *stop group*, shown in column '*stop grp.*'). The meaning of the rest of the columns of table 3.2 and the coloring of the cells will be explained as they are mentioned thorough the rest of this description of the AFC preprocessing.

The process is explained as three consecutive tasks:

3.2.1.1.1 Partition by vehicle, rank over each set

AFC entries are grouped by *vehicle*, and then ranked chronologically, starting from the superset Ω that contains all entries from the *raw_afc* table:

$$\Omega = \left\{\omega_i = (t_i, v_i, a_i, \alpha_i, b_i, \ldots)\right\} \tag{3.3}$$

Table 3.2: Raw AFC preprocessing procedure. *Vehicle*, *route*, and *group UID* remain constant through this example.

stop id	*rank over set* −	*rank over subset* =	*classif. variable*	*stop grp.*	*validation instant*	*time gap*	*below upper limit*	*∄ intermed. AVL entry*	*board. grp. change*	*board. group id*
					...					
A	1	1	0	A	03-24 12:31:23	<null>			1	987
B	2	1	1	B	03-24 12:33:56	<null>			1	988
C	3	1	2	C	03-24 12:36:14	<null>			1	989
C	4	2	2		03-24 12:36:16	00:00:02	✓	✓*	0	
D	5	1	4	D	03-24 12:37:24	<null>			1	990
E	6	1	5	E	03-24 12:39:44	<null>			1	991
E	7	2	5		03-24 12:45:22	00:05:38	✓	✓*	0	
F	8	1	7	F	03-24 12:47:37	<null>			1	992
G	9	1	8	G	03-24 13:48:51	<null>			1	993
G	10	2	8		03-24 13:50:59	00:02:08	✓	✓*	0	
G	11	3	8		03-24 13:53:04	00:02:05	✓	✓*	0	
G	12	4	8		03-24 14:11:11	00:18:07	✓	✗**	1	994
G	13	5	8		03-24 14:11:13	00:00:02	✓	✓*	0	
					...					
B	45	2	43	B	03-24 15:49:28	<null>			1	1004
C	46	3	43	C	03-24 15:51:33	<null>			1	1005
D	47	2	45	D	03-24 15:54:02	<null>			1	1006
D	47	2	45		03-24 15:54:02	00:00:00	✓	✓*	0	
D	48	3	45		03-24 23:05:49	08:11:47	✗		1	1007
E	49	2	47	E	03-24 23:09:09	<null>			1	1008
					...					

*: No visit of this vehicle to a different bus stop was found in the AVL data between $\boldsymbol{t_i - \Delta t_i\,(\omega_i)}$ and $\boldsymbol{t_i}$

**: The AVL data reveals that this vehicle visited bus stop K at 14:01:51

Each element ω_i is a tuple that represents one row of *raw_afc*:

$$
\begin{aligned}
i &: \text{unique row id} && i \in \mathbb{Z} \\
t &: \text{validation instant} && \text{full date (time)} \\
v &: \text{vehicle id} && v \in \mathbb{Z} \\
a &: \text{route id} && a \in \mathbb{Z} \\
b &: \text{bus stop id} && b \in \mathbb{Z} \\
\alpha &: \text{AFC group UID} && \alpha \in \mathbb{Z}
\end{aligned}
\tag{3.4}
$$

Then, a partition Σ of Ω is established, where each subset X_{v_i} contains the entries from *raw_afc* of the bus v_i:

$$\Sigma = \left\{ X_{v_i} \subset \Omega \mid X_{v_i} = \left\{ \left(x_{v_i}\right)_j \right\} = \{\omega_k \mid v_k = v_i\} \right\} \tag{3.5}$$

Equation (3.6) defines a binary relation Θ_{v_i} (‘*happended after*’) over each X_{v_i} (also known as an endorelation):

$$\Theta_{v_i} = \left\{ \left(\left(x_{v_i}\right)_l, \left(x_{v_i}\right)_m \right) \mid t_l > t_m \right\} \tag{3.6}$$

Θ_{v_i} creates a total preorder over X_{v_i}, allowing to assign a rank $(\gamma_{v_i})_n$ to each of its elements (some may be tied with each other). For each X_{v_i} there is a set Γ_{v_i} of ranks, as many as distinct timestamps:

$$\Gamma_{v_i} = \left\{ (\gamma_{v_i})_j \right\} = \left[1 \ldots |\Gamma_{v_i}| \right] \tag{3.7}$$

Finally, eq. (3.8) establishes the functions β_{v_i} that link each element $(x_{v_i})_j$ of each subset X_{v_i} to its rank within it. Thus, column 'rank over set' of table 3.2 contains the rank $\beta_{v_i}(\omega_i)$ of each *raw_afc* entry within the subset of all the rows related to vehicle v_i.

$$\begin{gathered} \beta_{v_i} : X_{v_i} \to \Gamma_{v_i}; \quad \forall\, (x_{v_i})_j\,, (x_{v_i})_k : \\ \begin{cases} \beta_{v_i}\left((x_{v_i})_j \right) > \beta_{v_i}\left((x_{v_i})_k \right) \Leftrightarrow (x_{v_i})_j \;\Theta_{v_i}\; (x_{v_i})_k \\ \beta_{v_i}\left((x_{v_i})_j \right) < \beta_{v_i}\left((x_{v_i})_k \right) \Leftrightarrow (x_{v_i})_k \;\Theta_{v_i}\; (x_{v_i})_j \\ \beta_{v_i}\left((x_{v_i})_j \right) = \beta_{v_i}\left((x_{v_i})_k \right) \Leftrightarrow \qquad t_j \;=\; t_k \end{cases} \end{gathered} \tag{3.8}$$

3.2.1.1.2 Partition by vehicle, route, group, and bus stop; rank over each subset

AFC entries are classified by *vehicle*, *route*, *group*, and *bus stop*; and ranked chronologically. These four columns of the *raw_afc* table remain constant during all the validations of a particular *boarding event*. The process is analogous to what has already been described in 3.2.1.1.1. The family Φ partitions Ω in several $Y_{v_i,a_i,u_i,\alpha_i,b_i}$ subsets. Each one contains the entries from *raw_afc* that show in their columns *vehicle*, *route*, *group*, and *bus stop* the values defined by the tuple $(v_i, a_i, \alpha_i, b_i)$:

$$\Phi = \left\{ Y_{v_i,a_i,\alpha_i,b_i} \subset X_{v_i} \middle| \begin{aligned} Y_{v_i,a_i,\alpha_i,b_i} &= \left\{\, (y_{v_i,a_i,\alpha_i,b_i})_j \,\right\} \\ &= \left\{ (x_{v_i})_k \mid a_k = a_i \wedge \alpha_k = \alpha_i \;\wedge\; b_k = b_i \right\} \end{aligned} \right\} \tag{3.9}$$

The binary relation $\Lambda_{v_i,a_i,\alpha_i,b_i}$ (again, '*happened after*') is characterized over each Y_{v_i,a_i,α_i,b_i} subset in eq. (3.10):

$$\Lambda_{v_i,a_i,\alpha_i,b_i} = \left\{ \left((y_{v_i,a_i,\alpha_i,b_i})_l\,, (y_{v_i,a_i,\alpha_i,b_i})_m \right) \mid t_l > t_m \right\} \tag{3.10}$$

Each element of Y_{v_i,a_i,α_i,b_i} can be mapped to a rank value $(\delta_{v_i,a_i,\alpha_i,b_i})_n$ thanks to the total preorder established by $\Lambda_{v_i,a_i,\alpha_i,b_i}$ over it. The set $\Delta_{v_i,a_i,\alpha_i,b_i}$ of all ranks of the elements of subset Y_{v_i,a_i,α_i,b_i} within it is:

$$\Delta_{v_i,a_i,\alpha_i,b_i} = \left\{ (\delta_{v_i,a_i,\alpha_i,b_i})_j \right\} = \left[1 \ldots |\Delta_{v_i,a_i,\alpha_i,b_i}| \right] \tag{3.11}$$

Lastly, (3.12) shows how functions $\varepsilon_{v_i,a_i,\alpha_i,b_i}$ link each element $(y_{v_i,a_i,\alpha_i,b_i})_j$ of each subset Y_{v_i,a_i,α_i,b_i} to its rank within it. Thus, column 'rank over subset' of table 3.2 contains the rank $\varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i)$ of each *raw_afc* entry within the subset of all the rows that share its *vehicle*, *route*, *group*, and *bus stop* values.

$$\begin{aligned} &\varepsilon_{v_i,a_i,\alpha_i,b_i} : Y_{v_i,a_i,\alpha_i,b_i} \to \varDelta_{v_i,a_i,\alpha_i,b_i};\\ &\forall\left((y_{v_i,a_i,\alpha_i,b_i})_j, (y_{v_i,a_i,\alpha_i,b_i})_k\right) :\\ &\begin{cases} \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_j\right) > \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_k\right) \\ \quad \Leftrightarrow (y_{v_i,a_i,\alpha_i,b_i})_j \; \varLambda_{v_i,a_i,\alpha_i,b_i} \; (y_{v_i,a_i,\alpha_i,b_i})_k \\ \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_j\right) < \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_k\right) \\ \quad \Leftrightarrow (y_{v_i,a_i,\alpha_i,b_i})_k \; \varLambda_{v_i,a_i,\alpha_i,b_i} \; (y_{v_i,a_i,\alpha_i,b_i})_j \\ \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_j\right) = \varepsilon_{v_i,a_i,\alpha_i,b_i}\left((y_{v_i,a_i,\alpha_i,b_i})_k\right) \Leftrightarrow t_j = t_k \end{cases} \end{aligned} \tag{3.12}$$

3.2.1.1.3 Create stop groups

The difference between $\beta_{v_i}(\omega_i)$ and $\varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i)$ from eqs. (3.8) and (3.12) returns the 'classif. parameter' of element ω_i ($\zeta(\omega_i)$, shown in table 3.2). This value, if one ranks chronologically all entries of set X_{v_i} (the ones related to *bus* v_i), remains the same and is unique for each group of rows from its subset Y_{v_i,a_i,α_i,b_i} (those that report *bus*, *line*, *group*, and *bus stop* values of v_i, a_i, α_i, and b_i) that appear consecutively.

$$\begin{aligned} &\zeta(\omega_i) = \beta_{v_i}(\omega_i) - \varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i);\\ &\forall(\omega_i,\omega_j) :\\ &\begin{cases} v_i = v_j \;\wedge\; a_i = a_j \;\wedge\; \alpha_i = \alpha_j \;\wedge\; b_i = b_j \\ \quad \wedge\; \zeta(\omega_i) = \zeta(\omega_j) \Longleftrightarrow \omega_i \text{ and } \omega_j \text{ belong to the same stop group.} \\ \text{Otherwise} \qquad\qquad\quad \Longleftrightarrow \omega_i \text{ and } \omega_j \text{ belong to different stop groups.} \end{cases} \end{aligned} \tag{3.13}$$

The column '*stop grp.*' of table 3.2 shows the outcome of this first approximation to the objective of identifying the *boarding groups*; displaying a single letter for all consecutive *raw_avl* entries with the same v_i, a_i,α_i, b_i, and $\zeta(\omega_i)$ values. The coloring of columns 'stop id,' 'classif. variable,' and 'stop group' illustrates the classification process and its result. For instance, rows pertaining to calls at *bus stop* D are gathered in two *stop groups*, differentiated by $\zeta(\omega_i)$ values of 2 and 45 .

A way to verify how this first process has performed is to study the 'time gap' (table 3.2) between consecutive rows of the same *stop group*, Δt_i (eq. 3.14), noting that if ω_{i-1} does not belong to Y_{v_i,a_i,α_i,b_i}, $\varepsilon_{v_i,a_i,\alpha_i,b_i}$ is not defined, so neither is Δt_i:

$$\Delta t_i = t_i - t_{i-1} \quad \text{if} \quad \varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_{i-1}) = \varepsilon_{v_i,a_i,\alpha_i,b_i}(\omega_i) - 1 \tag{3.14}$$

As this gap increases, it is more likely that the latter entry took place during a different visit of the bus (with no intermediate entries due to no validations being recorded until the bus came back). The next section studies this situation.

3.2.1.2 Split stop groups in boarding groups

The question regarding excessive time gaps between some of the entries that are part of the same *stop group*, is addressed with the following assumptions:

- In some cities, it is not uncommon for the driver to allow passengers to wait for the start of a run inside the bus, especially if the weather is bad. However, if the separation between two consecutive entries of the same *stop group* is greater than the maximum headway s (eq. 3.2), their group is split between them. This happens in the next to last row of table 3.2: the time elapsed since the previous validation is extremely long (represented with the symbol '✗' in column 'below upper limit'), so one can be certain that this row is describing a different *boarding event* and the group is split, as has been portrayed with the change in color from orange to red. An adequate value for this parameter depends on the particularities of the case under analysis.
- For all other pairs of consecutive entries of the same *stop group*, if table *avl_coalesced* (defined in section 3.2.2.5 during the description of the pre-processing of the AVL data) shows that the bus visited another stop between their *timestamps*, they belong to different *boarding groups*. An example of this situation can be found in the row with 'rank over set' = 12 of table 3.2, where the 5-entries *stop group* (G, 8) it is part of is split in two boarding groups (993 and 994); because, as the symbol ✗ of column '∄ AVL entry' denotes, between its timestamp (14:11:11) and the one from the previous entry (13:53:04) a lookup through the raw AVL data (not represented) has concluded that the bus called at bus stop K at 14:01:51.

These premises are utilized to define η, (eq. (3.15), column 'board. grp. change' of table 3.2), a value that equals 1 if a row is the first of a boarding group, and 0 in other cases. σ_j represents an entry of table *avl_coalesced* (section 3.2.2.5), while h_j and p_j are its vehicle id and arrival time, respectively:

$$\eta\left(\omega_i\right) = \begin{cases} 0 & \text{if } \Delta t_i \leq \boldsymbol{s} \\ & \qquad \wedge \; \nexists \sigma_j \;\middle|\; h_j = v_i \\ & \qquad\qquad\qquad \wedge \; t_i - \Delta t_i \leq p_j \leq t_i \\ 1 & \text{otherwise} \end{cases} \tag{3.15}$$

An index θ is then defined over Ω, sorting its rows by *vehicle*, chronological *rank* within the entries of their vehicle (ascending), *bus stop* (ascending), and *boarding group change* (descending). The relative ordering of entries ω_i, ω_j with $v_i = v_j$,

$\beta_{v_i}(\omega_i) = \beta_{v_j}(\omega_j)$, $b_i = b_j$, and $\eta(\omega_i) = \eta(\omega_j) = 0$ is inconsequential. When ordered with this index, the elements of Ω appear consecutively if they are part of a *boarding group*, with a value of *group change* of 1 for the first entry and 0 for the others up until its end. In other words, *group change* equals one when a *stop group* commences (since a *boarding group* also begins) or when a *stop group* is split due to one of the two previous criteria (section 3.2.1.2).

$$\begin{aligned}
&\theta : \Omega \to \{1 \ldots |\Omega|\};\\
&\quad \theta(\omega_i) > \theta(\omega_j) \iff v_i > v_j\\
&\qquad \lor \quad v_i = v_j \ \land \ \beta_{v_i}(\omega_i) > \beta_{v_j}(\omega_j)\\
&\qquad \lor \quad v_i = v_j \ \land \ \beta_{v_i}(\omega_i) = \beta_{v_j}(\omega_j) \ \land \ b_i > b_j\\
&\qquad \lor \quad v_i = v_j \ \land \ \beta_{v_i}(\omega_i) = \beta_{v_j}(\omega_j) \ \land \ b_i = b_j\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land \ \eta(\omega_i) \leq \eta(\omega_j)
\end{aligned} \tag{3.16}$$

The *boarding group id*, $o(\omega_i)$, of each row ω_i is the sum total of all $\eta(\omega_j)$ values from rows ω_j such that $\theta(\omega_j) \leq \theta(\omega_i)$:

$$o(\omega_i) = \sum_{j \mid \theta(\omega_j) \leq \theta(\omega_i)} \eta(\omega_j) \tag{3.17}$$

Going back to Table 3.2, the content and colors of the cells of columns 'time gap,' 'below upper limit,' '∄ intermediate AVL entry,' 'boarding group change,' and 'boarding group id' have been chosen to describe how stop groups are split in boarding groups:

- If an entry is the first of its *stop group* ('time gap' = <null>), a new *boarding group* should also begin (rows with 'rank over set' $\in \{1, 2, 3, 5, 6, 8, 9, 45, 46, 47, 49\}$). 'group change' equals 1, and there is no need to check columns 'below upper limit' or '∄ intermediate AVL entry.' For each of these rows, the columns involved in the identification of their *stop group* and *boarding group* are filled with the same color, different from their respective predecessors.

- If the lapse between two successive validations of the same *stop group* is too long, they are the end and beginning of two different *boarding groups*. The latter row shows the symbol ✗ at 'below upper limit,' while its column '∄ intermediate AVL entry' is not needed, and 'group change' is 1. It also depicts its whole decision process, utilizing one color for '*stop id*,' 'grouping parameter,' and 'stop group'; and another for 'below upper limit' and 'boarding group id,' showing how each *stop group* is split in *boarding groups*.

- For the remaining pairs of consecutive rows that share the same *stop group*, the symbol of column '∄ intermediate AVL entry' indicates whether they belong to the same *boarding group*:

Table 3.3: AFC pre-processing output: *boarding_groups*

Column	***Type***	***Description***
id	integer	Id of the boarding group.
bus_stop	integer	Bus stop id.
vehicle	integer	Vehicle id.
route	integer	Route id.
group	integer	AFC group UID.
boarding_range	time range	[earliest validation, latest validat.]

✗: The vehicle related to both entries has moved to another stop (and eventually back) at a time between their timestamps, so they belong to different *boarding groups*. Again, ‘group change’ = 1, and colors illustrate the reasoning behind this decision: one color for ‘*stop group*’ and the first *boarding group,* and a different one for the second *boarding group* created by the split.

✓: There is no evidence that the vehicle has moved between the timestamps of both entries, so it is concluded that they belong to the same *boarding group*: ‘group change’ = 0. The columns of the latter row that decide its *stop group* and *boarding group* show the same colors as in the former.

Alternatively, fig. 3.2 illustrates how stop groups are separated in boarding groups with a flow diagram.

Finally, for each boarding group o_x, the instants of its first $\vartheta(o_x)$ and last $\iota(o_x)$ validations are computed, as well as how long it lasted $\kappa(o_x)$:

$$\begin{aligned} \vartheta(o_x) &= \min\left(\left\{t_i \mid o(\omega_i) = o_x\right\}\right) \\ \iota(o_x) &= \max\left(\left\{t_i \mid o(\omega_i) = o_x\right\}\right) \\ \kappa(o_x) &= \iota(o_x) - \vartheta(o_x) \end{aligned} \tag{3.18}$$

Boarding groups that last longer than the maximum headway for their route (s, eq. 3.2) are considered to originate from unreliable data, and are not utilized to infer missing visits to stops not recorded by the AVL.

3.2.1.3 Output

The results of the AFC pre-processing are gathered in the table *boarding_groups*, structured as shown in table 3.3, while fig. 3.3 depicts an example transition from 15 individual ticketing events to 4 encompassing *boarding groups*.

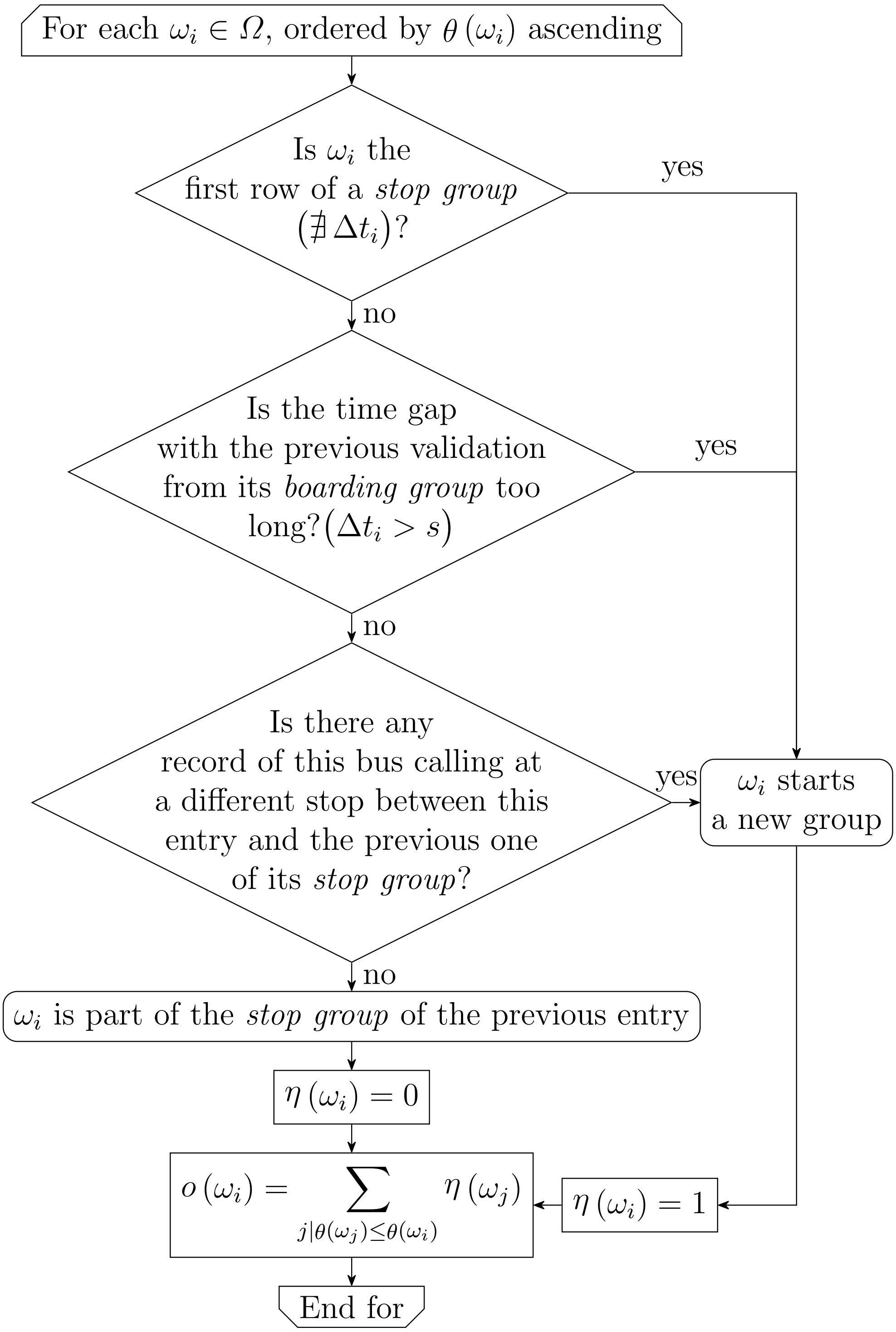
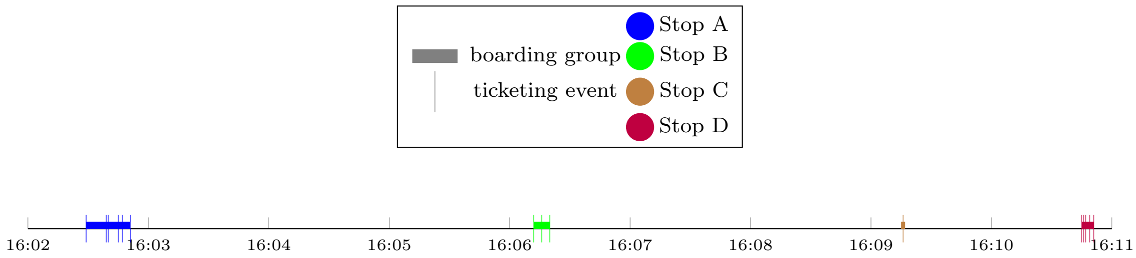


Figure 3.2: Splitting stop groups in boarding groups



AFC data sample from one vehicle

Figure 3.3: Transition from individual ticketing events to encompassing *boarding groups*

3.2.2 AVL

The procedures detailed in this section aim to characterize the movement of the vehicles with a single record for each stop of each run. Also, after unfeasible entries are identified and removed, the remaining ones are classified in trajectories.

3.2.2.1 Filter out duplicate rows

The first action is to identify and filter out duplicate entries, creating the multiset $\mathit{\Theta}$ of all *raw_avl* rows, defined over the set $\mathit{\Lambda}$ of distinct AVL records λ_i; and the multiplicity function ζ that returns how many times each λ_i appears in the raw AVL dataset:

$$\begin{aligned}
&\mathit{\Theta} = \langle \mathit{\Lambda}, \zeta \rangle = \left\{ \vartheta_i = {\lambda_i}^{\zeta(\lambda_i)} \right\} \\
&\mathit{\Lambda} = \left\{ \lambda_i = (t_i, v_i, a_i, b_i, \beta_i, \vartheta_i) \right\} \\
&\zeta \ : \ \mathit{\Lambda} \to \mathbb{Z}_{\geq 1}
\end{aligned} \tag{3.19}$$

Each λ_i is a tuple with the fields described in eq. (3.20):

$$\begin{array}{ll}
i : \text{unique row id} & i \in \mathbb{Z} \\
t : \text{instant} & \text{full date (time)} \\
v : \text{vehicle id} & v \in \mathbb{Z} \\
a : \text{route id} & a \in \mathbb{Z} \\
b : \text{bus stop id} & b \in \mathbb{Z} \\
\beta : \text{AVL group id} & \beta \in \mathbb{Z} \\
\vartheta : \text{stop duration} & \text{time}
\end{array} \tag{3.20}$$

3.2.2.2 Remove rows not linked to a real bus stop

It is assumed that entries with a *bus stop id* not found in set Ξ (defined in eq. 3.1) are caused by exceptional events that do not happen consistently as the buses travel their routes, and are not linked to a position change. Equation (3.21) establishes $\mathit{\Psi}$, a subset of $\mathit{\Lambda}$ after these have been filtered out:

$$\mathit{\Psi} = \{\psi_i = \lambda_j \mid \exists \xi_k : b_j = m_k\} \tag{3.21}$$

3.2.2.3 Identify *trajectories*

The next step is to utilize the columns from the AVL data to differentiate between the runs that constitute the public transportation supply. To this end, in this work a '*trajectory*' is defined as consecutive AVL records that share the same *vehicle*, *route*, and *group*. The relation R maps each element of $\mathit{\Psi}$ with distinct values of h, f, and β to a different *trajectory id* (r):

$$R(\psi_i) = r_i \in \mathbb{Z}; \quad \forall (\psi_i, \psi_j) :$$
$$\begin{cases} r_i = r_j \iff a_i = a_j \wedge v_i = v_j \wedge \beta_i = \beta_j \iff \psi_i \text{ and } \psi_j \text{ are part} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{of the same trajectory.} \\ r_i \neq r_j \iff a_i \neq a_j \vee v_i \neq v_j \vee \beta_i \neq \beta_j \iff \psi_i \text{ and } \psi_j \text{ are part} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{of different trajectories.} \end{cases} \tag{3.22}$$

3.2.2.4 Determine *visit groups*

Table 3.4 depicts how each trajectory is examined to tell apart those occasions when more than one row is added to the dataset for the same call at a stop (for example, when the doors are re-opened to let a late passenger in the bus). The procedure to identify these '*visit groups*' (calculate each entry's ranks over its trajectory, and among those records with the same trajectory and stop values; and then evaluate each element's classification variable as their subtraction) is similar to the one that has already been described and implemented in page 23 to find *stop groups* in the AFC data. Its result is a relation M (eq. (3.23)) which assigns the same *visit group* id (μ_i) to consecutive entries of a *trajectory* that happen in the same *bus stop*:

$$M(\psi_i) = \mu_i \in \mathbb{Z}; \quad \forall (\psi_i, \psi_j) :$$
$$\begin{cases} \mu_i = \mu_j \iff \psi_i \text{ and } \psi_j \text{ are part of the same visit group.} \\ \mu_i \neq \mu_j \iff \psi_i \text{ and } \psi_j \text{ are part of different visit groups.} \end{cases} \tag{3.23}$$

3.2.2.5 Merge entries of each *visit group*

The set $\mathit{\Sigma}$ summarizes the information pertaining the *visit groups*. Its elements contain the fields shown in eq. (3.24), and are stored in the table *avl_coalesced* (table 3.6b):

$$\mathit{\Sigma} = \left\{\sigma_{\mu_i} = (r_{\mu_i}, b_{\mu_i}, n_{\mu_i}, p_{\mu_i})\right\} \tag{3.24}$$

Table 3.4: How entries of a trajectory linked to a single visit are identified

Stop	***Chronological rank over trajectory***	**-**	***Chronological rank over (trajectory, stop)***	=	***Visit group number***	***Stop***
A	1	-	1	=	0	A
B	2	-	1	=	1	B
C	3	-	1	=	2	C
C	4	-	2	=	2	
D	5	-	1	=	4	D
E	6	-	1	=	5	E
C	7	-	3	=	4	C
F	8	-	1	=	7	F
...						

Each of its elements σ_{μ_i} is a tuple with the characteristics of the *visit group* μ_i (table 3.5 outlines this process):

- Its unique *visit group id* (μ_i).
- Its *trajectory* r_{μ_i}: The characteristics that define it (*route*, *vehicle*, and *group UID*) are referred to as r_{μ_i}, v_{μ_i}, and β_{μ_i}.
- Its *bus stop* b_{μ_i}.
- The moment n_{μ_i} the bus arrived at the stop, defined as the minimum instant (w) from all elements of Ψ that are part of this *visit group*:

$$n_{\mu_i} = \min\left(\{w_j \mid \psi_j : M(\psi_j) = \mu_i\}\right) \tag{3.25}$$

- The instant p_{μ_i} when the bus left the stop defined as the maximum of these two values:
  - $(p_{\mu_i})_1$: The maximum of the addition of the *instant* (w_j) and the duration (ϑ_j) for those elements ψ_j of the *visit group* where *stop duration* is defined.
  - $(p_{\mu_i})_2$: The maximum of the *instant* (w_j) for those elements ψ_k of the *visit group* that do not report a *stop duration* value ($p_k =$ <null>).

$$\begin{aligned} (p_{\mu_i})_1 &= \max\left(\{w_j + \vartheta_j \mid \psi_j : M(\psi_j) = \mu_i \wedge \exists p_j\}\right) \\ (p_{\mu_i})_2 &= \max\left(\{w_k \mid \psi_k : M(\psi_k) = \mu_i \wedge \nexists p_j\}\right) \\ p_{\mu_i} &= \max\left((p_{\mu_i})_1, (p_{\mu_i})_2\right) \end{aligned} \tag{3.26}$$

Table 3.5: Treatment of multiple AVL entries from the same visit to a stop

instant	*duration*	*stop*
...		
12:50:29	0	151
12:51:11	19	135
12:51:18	198	135
12:51:47	<null>	135
12:55:07	14	134
...		

(a) *avl*

arrival	*departure*	*stop*
...		
12:50:29	12:50:29	151
12:51:11	12:54:36	135
12:55:07	12:55:21	134
...		

(b) *avl_coalesced*

3.2.2.6 Identify and remove unfeasible or unrealistic trajectory legs

Regarding each trajectory as a series of legs between its *visit groups*, those shorter than the free flow time between the involved stops are not possible. Two situations arise:

- Moving backwards the departure time in the former stop, thus increasing the leg length, solves the issue. This amounts to assuming that the information regarding how long the bus stayed in the initial stop of the leg is not reliable.
- Not even setting the dwell time in the former stop to zero leaves enough time to travel to the latter. In this case, both *visit groups* are considered as unreliable and removed.

Also, those legs longer than the upper bound e (eq. 3.2) for their route are used to split their trajectories. Thus, AVL entries that present the same *vehicle*, *route*, and *group*, but separated by a leg too long to have occurred during a single run, are considered separately.

3.2.2.7 Output

Table 3.6 shows how the outcome of AVL preprocessing is stored in tables *trajectories* and *avl_coalesced*.

3.2.3 Schedule

Firstly, the events recorded in the scheduling subsystem should be corrected by the appropriate value z (eq. 3.2), if defined for the corresponding route.

Then, a time range n is created for each planned run, encompassing the arrival and departure times that can be deducted from the most specific available columns, as long as they provide coherent information (e.g., departures cannot happen before arrivals). Another time buffer q is also created around its planned *start time* t_p , with a semi-width equal to the maximum headway s (eq. 3.2). It is used in section 3.3.5 to match each entry of the schedule to the run that materializes it.

Table 3.6: AVL preprocessing output

Column	***Type***	***Description***
id	integer	Id of the visit.
route	integer	Route id.
vehicle	integer	Vehicle id.
group	integer	AVL group UID.
stops_sequence	integer	Stops seq. id (described later).
trajectory_range	time range	[traject. start, traject. end]

(a) *trajectories*

Column	***Type***	***Description***
id	integer	Id of the visit.
bus_stop	integer	Bus stop id.
ord_wthin_trj	integer	Chronological order within traject.
trajectory	integer	Trajectory id.
avl_range	time range	[arrival, departure]

(b) *avl_coalesced*

Equations (3.27) and (3.28) respectively enunciate the parameter and variables, and detail the conditions just described; while table 3.7 shows the structure of the planning information after preprocessing.

$$\begin{aligned}
t_p &: \text{planned departure time} && \text{full date (time)} \\
t_d &: \text{recorded departure time} && \text{full date (time)} \\
t_a &: \text{recorded arrival time} && \text{full date (time)} \\
n &: \text{visit range from scheduling subsystem} && \big[\text{arrival time}, \\
& && \quad \text{departure time}\big] \\
q &: \text{run search buffer} && \big[\text{lower time bound}, \\
& && \quad \text{upper time bound}\big]
\end{aligned} \tag{3.27}$$

$$\begin{aligned}
n &= \begin{cases} [t_a, t_d] & \text{if } t_a \leq t_d \\ [t_p, t_d] & \text{if } (t_a > t_d \lor \nexists t_a) \land t_p \leq t_d \\ [t_d, t_d] & \text{if } (t_a > t_d \lor \nexists t_a) \land t_p > t_d \\ \langle null\rangle & \text{otherwise} \end{cases} \\
q &= [t_d - s, t_d + s]
\end{aligned} \tag{3.28}$$

3.3 Vehicle runs definition

This section aims to improve the representation of the runs that occurred in the IPTS, which is hindered by the problems described in page 12. Besides missing,

Table 3.7: raw schedule preprocessing output: *schedule*

Column	***Type***	***Description***
id	integer	Planned start id.
bus_stop	integer	Stop id.
vehicle	integer	Vehicle id.
run_srch_buff	t. range	[t. min, t. max] to match to a run.
sched_range	t. range	[arrival, depart.] from the sched. subsystem.

duplicate, o erroneous entries, events referring to the same run may have been mistakenly labeled as belonging to different ones.

The outline of this process is: study trajectories according to their underlying sequences of bus stops, and these as fragments of the subroutes offered by the IPTS (page 36); choose travel and dwell times models for each route (page 39); build tables that describe when each bus run visits each stop, combining AVL and AFC entries (page 39); detect and treat instances where the id of a vehicle changes mid-run (page 45); link runs to planned starts, making use of the extra information to improve their definition (page 48); and filter out possible runs not supported by enough IPTS evidence (page 49).

3.3.1 Analyze AVL trajectories as sequences and fragments of routes

AVL trajectories are analyzed as just ordered *sequences* of stops, which are the building blocks to assemble the full runs that have occurred, defined by their "template sequences." Table 3.8 illustrates this process, and table 3.9 gathers the outputs of its three steps:

3.3.1.1 Identify distinct AVL trajectory sequences

An id is assigned to each unique stops sequence extracted from the trajectories of each route, as shown in tables 3.8a, 3.8b and 3.9a. The field *stops_sequence* of the trajectories table (page 35) signifies this relation.

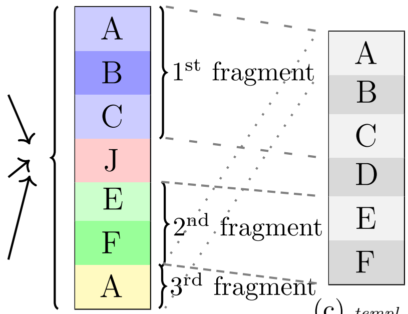
3.3.1.2 Single out template sequences

This methodology assumes that each route can be split in a series of "subroutes" that represent the runs that compose it (for instance, the runs back and forth between the termini of a linear route; or a single run in the case of circular routes). Each subroute is characterized by its "template sequence" of stops table 3.8c) that a typical, completely carried out, perfectly recorded run of that *subroute* must follow. They can be known beforehand, or ascertained through the examination of the sequences of stops found during their previous step, and their relative frequencies, since the templates are very likely to be among those found most often. They are stored as illustrated in table 3.9b.

Table 3.8: Analysis of the trajectories of a route for one of its template sequences

id	***stops***
...	
123	A B C J E F A
...	
4567	A B C J E F A
...	
18460	A B C J E F A
...	

(a) *trajectories*



(b) *sequence* 124

(c) *templ. seq. 16365*

Ord. within sequence	*Ord. within template (stop number)*	*Length*	*Sub-seq. id*	*Seq. id*	*Template*
...					
1	1	3	1	124	16365
5	4	2	2	124	16365
7	1	1	3	124	16365
...					

(d) *fragments*

The elements of each of the *template sequences* of each route can be identified by their ordinal (the "stop number").

3.3.1.3 Break down sequences in template fragments

As depicted in tables 3.8b to 3.8d, the sequences followed by the trajectories can be split in:

- Continuous *fragments* of their route's *template sequences* (i.e., no elements missing between their extremes), that represent parts of runs that the AVL system managed to record correctly. They allow to view each trajectory found in the AVL data as a series of *segments* that fit in its template. Table 3.9c how they are stored.

- Incompatible portions (caused by erroneous entries in the AVL subsystem; the vehicle carrying out other *subroute*; or incorrect operations (e.g., not updating the on-board computer to reflect that the bus is following a different route).

Table 3.9: AVL sequences analysis output

Column	***Type***	***Description***
id	integer	Id of the sequence.
route	integer	Line id.
stops_sequence	tuple of int	Sequence of stop ids.

(a) *stops_sequences*

Column	***Type***	***Description***
id	integer	Sequence id.
n_stops	integer	Number of stops (materialized for convenience).
name	text	Human-friendly name.

(b) *template_sequences*

Column	***Type***	***Description***
ord_within_seq	int	Event ordinal within its sequence
stop_number	int	Event ordinal within the template.
fragment	int	fragment id within its sequence.
sequence	int	sequence id.
template	int	template sequence id.

(c) *fragments*

3.3.2 Choose link travel times and dwell times distribution models

These models are utilized as part of the criteria for identifying AVL fragments or boarding groups that are part of the same run, to infer missing stop information, and to filter out erroneous recorded run starting times. For each route, link travel times between consecutive stops, and dwell times for all of them but the last one, are needed.

They should consider known factors that modify travel and dwell times in the system under study, such as the time, whether it is a working day or not, or seasonal mobility changes.

3.3.3 Assemble vehicle runs

Runs are constructed starting from a "seed" that is completed backwards and forward in time, looking for AVL segments and *boarding groups* events part of the same subroute and with the same *vehicle id* as the seed that, according to the instant of the furthermost known data point in the current growth direction and the probability distributions of the duration of unknown intermediate legs and calls at stops, fall within their **minimum-amplitude prediction interval of probability $\boldsymbol{g}$**.

'$\boldsymbol{g}$' is a parameter of this methodology (eq. 3.29). The closer it is to one, the wider and more computer-intensive the search needs to be, and the risk of considering invalid or unrelated events as part of the current run increases. If set too low however, events that really were part of the run that is being characterized may be ignored.

$$\boldsymbol{g}: \text{probability of the prediction intervals} \quad \boldsymbol{g} \in [0, 1] \tag{3.29}$$

For each subroute and direction (backwards or forward in time), the seeds are selected following two consecutive iterative process. Firstly, by looping over the AVL fragments with a length of at least $\boldsymbol{c}$, from longer to shorter. '$\boldsymbol{c}$' is the parameter '***minimum AVL seed length***' (eq. 3.30). This decision stems from the hypothesis that **longer AVL trajectory fragments are more likely to be reliable**, while shorter ones may be caused by clock, GPS, or operation errors. After that, those *boarding groups* not filtered out are also used as seeds. The algorithm skips those seeds contained in the tables of events to be ignored (explained in page 40).

$$\boldsymbol{cz}: \text{min. AVL seed length} \quad \boldsymbol{c} \in \mathbb{N} - \{0\} \tag{3.30}$$

Once a seed has been established, it "grows" both back and forward in time, following a procedure that bears similarity to dead reckoning: starting from the furthest known point in a direction (the initial fix), minimum-amplitude prediction intervals of probability $\boldsymbol{g}$ for the departures or arrivals (if traversing backwards or forward, respectively) of the calls at consecutively farther away stops are computed as the sum of the involved travel and dwell times from intermediate stops, until one of following conditions is reached (checked in this order) and a new fix is selected:

- The prediction interval intersects the *avl_range* of at least a record from table *avl_coalesced*. In this case, the closest to the most likely arrival and departure times range is chosen, and a portion of its encompassing fragment is identified and added to growing new run, from said record up to what comes first between:
  - The next-to-last or second stop of the route, while growing forward or backwards, respectively.
  - The end of its fragment in the current growth direction.

  This distinction aims to on one hand to save computer time, by adding in a single step several calls of the vehicle; and it also makes sure that a feasibility range is always calculated at the termini. Besides being used as part of the current process to filter out unrealistic IPTS entries at those stops; they are also employed to decide the best way to include the information available from the schedule.
- The prediction interval intersects the *boarding_range* of at least a compatible *boarding group*. The closest to the most likely arrival and departure times range is chosen.
- If the stop under scrutiny is a terminus, the most likely arrival (or departure, if growing the run backwards) and dwell time are chosen.

In the first or second conditions, “compatible” means that it refers to the same route and vehicle as the seed; and is not in the table of events to be ignored (explained in page 40). Also, if more than one possibility appears, the most likely one according to the utilized link travel time and dwell time distributions is selected.

In all three instances, once the new fix has been selected, the set of most likely values for link travel times and dwell times is used to infer the arrival and departure times at missing intermediate stops.

After reaching a terminus, growth in the current direction ends. For those routes where data at the termini have been deemed particularly unreliable ($y =$ true, eq. 3.2), if the call at the closest stop is backed by AVL or AFC data, arrival and departure times are always inferred.

Once a seed has grown to encompass a full run, as described by its *template*; a buffer encompassing it is created, extending backwards and forwards in time from each call's respective arrival and departure, adding the roundtrip time lower bound for the corresponding route (d, eq. 3.2). AVL segments and *boarding groups* that overlap it are added to the tables of elements to ignore during the reminder of the run assembly process. This procedure serves two purposes: to enforce that no event is utilized as part of more than one run; and that vehicles follow feasible itineraries (enough time passes before they return to the same stop, as part of another run).

Figure 3.4 displays a flowchart of the first part of this process, which utilizes segments of AVL data as seeds. The second part is completely analogous, but for the fact that only the remaining AFC information is utilized.

Figure 3.5 shows a complete example. Its main steps are:

1 The initial seed is an AVL segment that goes from the arrival at :20:13 at AB, to the departure at :22:41 from AE.

2 It grows backwards, utilizing the search range [:18:51, :19:53] at the terminus AA. It has been defined setting the arrival at AB as a fixed point, and calculating the prediction interval of probability $\boldsymbol{g}$ for the presence of the vehicle at AA.

- A single compatible overlapping AVL event is found (3a), with arrival and departure times :18:31 and :18:55, respectively:
  - If the readings at the termini for this route have been deemed as reliable as in other stops ($y =$ false), 3a is accepted as the call of the bus at the initial terminus.
  - Otherwise, since the fix for the search is in the stop next to the terminus (3c), the inferred visit 3b, from :19:15 to :19:46, is preferred.

  Since this is one of the route's terminus, the growth backwards ends.

4 Growing forward, the search range to be used at AF is computed, utilizing as a fixed reference the departure time from AE (:22:41). The result is the prediction interval of probability $\boldsymbol{g}$ of the presence of the bus at AF: [:23:04, :24:05], which intersects no compatible entry from the AVL or AFC subsystems.

5 The script keeps searching forward. At AG, another prediction interval of probability $\boldsymbol{g}$ is created for the arrival of the bus. This time, the sum of the individual distributions of travel times from AE to AF, and from AF to AG; and of the dwell time at AF will be needed. The ensuing range ([:23:17, :24:57]) overlaps with a *boarding group* ([:23:55, :23:59]). Its earliest and latest ticketing events are used as an approximation of the arrival and departure at AG.

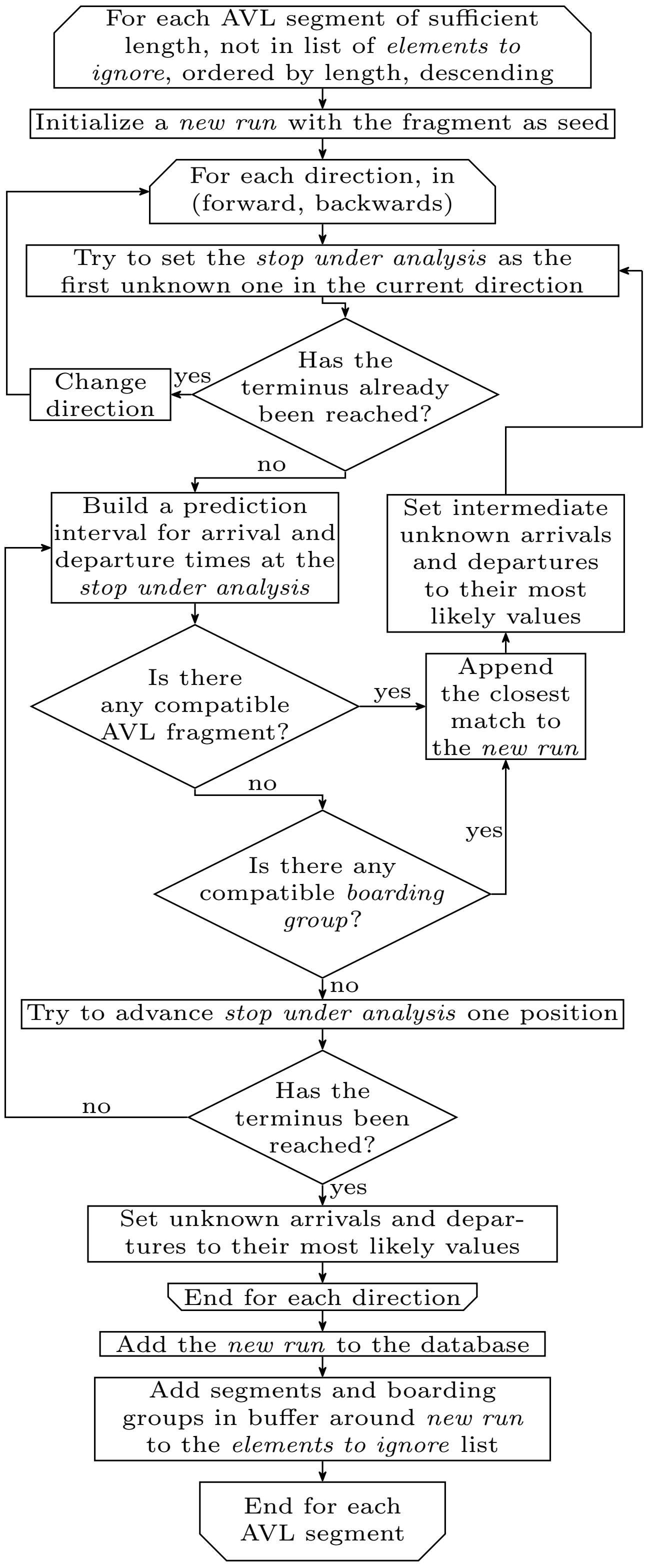


Figure 3.4: Vehicle run inference process from AVL seeds

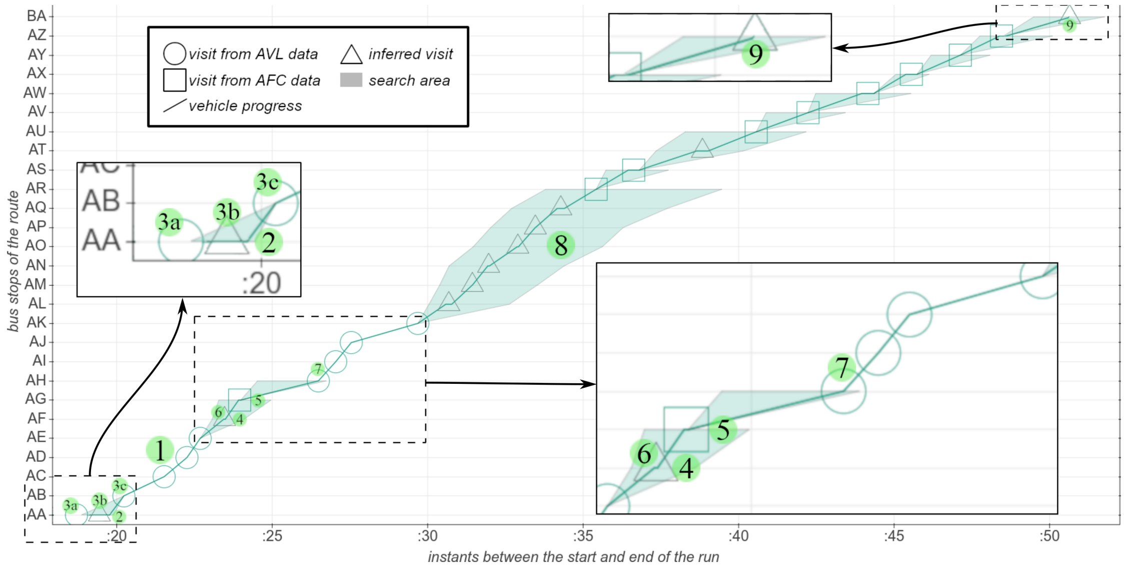


Figure 3.5: Vehicle runs inference process

6 Considering now the gap of 1m14s between the bus leaving AE at :22:41, and arriving at AG at :23:55; the most likely combination of the travel times from AE to AF and from AF to AG; and of dwell time at AF that add up to it is, according to their respective probability distributions, 45s, 26s, and 3s; respectively. Thus, arrival and departure times at AF are set to [:23:26, :23:29].

7 Again, the search takes place at stop H, finding a compatible AVL entry. This one, and other three from the same fragment are added to the run.

8 Several intermediate stops had to be inferred between the departure from AK and the arrival and AR. Missing arrival and departure times are set to their most probable values, according to the 7 travel time and 6 dwell time distributions involved.

9 Finally, the other terminus of the route is reached. Since no compatible AVL or AFC is found, the arrival at this stop; as well as arrivals and departures at others downstream the last known departure, if any; are set to their mean values.

Table 3.10 shows the output generated once all AVL and AFC data have been processed, materialized as 3 tables:

- *runs*, which synthesizes each of the vehicle runs that have been detected by this methodology.

Table 3.10: Vehicle runs definition

Column	***Type***	***Description***
vehicle	int	Vehicle id.
id	int	Run id.
merged_run	int	Encompassing run id (if appl.).
template_sequence	int	Template sequence id.
scheduled_beginning	int	Scheduled beginning id (if appl.).
run_range	time range	[1st stop depart., last stop arriv.]
merged_runs	tuple of int	Encompassed runs ids (if appl.)

(a) *runs*

Column	***Type***	***Description***
stop_number	int	Ordinal of stop in template.
id	int	Id of the visit.
run	int	Run the visit is part of.
avl_coalesced_id	int	*avl_coalesced* source (if appl.).
boarding_group_id	int	*boarding_group_id* source (if appl.).
visit_range	time range	[arrival, departure]

(b) *visits*_to_stops

Column	***Type***	***Description***
origin	int	Ordinal of the last stop with IPTS data.
stop_number	int	Ordinal of the stop under scrutiny.
run	int	Run the visit is part of.
search_range	time range	Bounds of the prediction interval.

(c) *search_ranges*

- *visits_to_stops*, that characterizes each run.
- *search_ranges*, where the prediction intervals utilized during the creation of the runs are saved.

3.3.4 (Optional) detect and merge instances where a vehicle changed its id mid-run

Due to the way operations are handled by the IPTS, some vehicles may change their *id* mid-run, as it happens in the case study analyzed in this paper. They can be detected in this methodology as two extremely close in time "former" and "latter" runs, where a single vehicle could have provided all non-inferred *visits_to_stops*. This section follows the nomenclature described in (3.31).

$$
\begin{aligned}
&\phi, \lambda : \text{formr, lattr } \textit{run ids} && \phi, \lambda \in \mathbb{Z} \\
&C : \text{“should be merged” relation} && C = \big\{(\phi, \lambda) \mid \phi,\, \lambda \text{ are the same run}\big\} \\
&T : \text{instants} && \text{possible full dates (time)} \\
&R : \text{time ranges} && R = \big\{(\rho[0], \rho[1]) \in T^2 \mid \rho[0] \leq \rho[1]\big\} \\
&\&\& : \text{“overlap” relation} && \&\& = \big\{\, (\rho_i, \rho_j) \mid \rho_j[0] \leq \rho_i[1] \leq \rho_j[1] \\
& && \qquad \vee\ \rho_j[0] \leq \rho_i[0] \leq \rho_j[1] \\
& && \qquad \vee\ \rho_i[0] < \rho_j[0] \wedge\ \rho_i[1] > \rho_j[1]\big\} \\
&\sigma_i : \text{run } i \text{ range} && \sigma = (\text{start}, \text{end}) \in R \\
&\upsilon_{i,j} : \text{visit range for run } i \text{ at stop } j && \upsilon = (\text{arrivl}, \text{depart.}) \in R \\
&\epsilon_{i,j} : \text{search range for run } i \text{ at stop } j && \epsilon = (\text{ lower bound, upper bound}) \in R \\
&\tau : \text{stop number} && \tau \in \mathbb{N} \\
&\mu_i : \text{lower bound of the travel time from stop } i \text{ to } i+1 && \text{time}
\end{aligned}
\tag{3.31}
$$

To correct this problem, the authors propose the following procedure to be carried out for each *template sequence*:

3.3.4.1 Identify pairs of vehicle runs that should be combined

- To decrease process time, only those runs with overlapping *run_ranges* are considered:

$$\phi\ C\ \lambda \implies \sigma_\phi\ \&\&\ \sigma_\lambda \tag{3.32}$$

- For each call of each run at a stop, a time buffer is created, as the smallest one that includes its *visit_range* and, if exists, its *search_range*. Two runs happen closely enough to be candidates when their time buffers overlap.

$$\begin{aligned}\phi\, C\, \lambda \implies \exists \tau \mid \upsilon_{\phi,\tau} \;\&\&\; \upsilon_{\lambda,\tau} \vee \epsilon_{\phi,\tau} \;\&\&\; \upsilon_{\lambda,\tau} \\ \vee\; \upsilon_{\phi,\tau} \;\&\&\; \epsilon_{\lambda,\tau} \vee \epsilon_{\phi,\tau} \;\&\&\; \epsilon_{\lambda,\tau}\end{aligned} \tag{3.33}$$

- Finally it must be possible, considering the lower bounds of the travel times between stops, for a single bus to perform all *visits_to_stops* entries from both runs that stem from the IPTS data. How this condition is met depends on the highest *stop_number* for which the "former" run presents a non-inferred *visits_to_stops* entry (τ_ϕ); and, correspondingly, on the lowest one from the "latter" (τ_λ):
  - If $\tau_\phi = \tau_\lambda = \tau$, they both represent the same stop, where the IPTS has records with both the old and the new *vehicle ids*. The following time ranges are computed at said stop:
    * A "feasibility range" $\zeta_{\phi,\lambda}$ that delimits the time span in which it is possible for the bus to have arrived after departing from the $(\tau-1)^{\text{th}}$ stop, as described in the "former" run ϕ, and still make it to the $(\tau+1)^{\text{th}}$ from the "latter" λ, taking into account the minimum bounds of the durations of the involved route legs:

$$\begin{aligned}\zeta_{\phi,\lambda} \in R;\quad \zeta_{\phi,\lambda} = \big(\upsilon_{\phi,(\tau-1)}[1] + \mu_{(\tau-1)}, \\ \upsilon_{\lambda,(\tau+1)}[0] - \mu_{(\tau)}\big)\end{aligned} \tag{3.34}$$

    * A "bus presence range" $\eta_{\phi,\lambda}$, which is the minimum-span range that encompasses those of both the "former" and "latter" runs:

$$\begin{aligned}\eta_{\phi,\lambda} \in R;\ \eta_{\phi,\lambda} = \big(min\,(\upsilon_{\phi,\tau}[0], \upsilon_{\lambda,\tau}[0])\,, \\ max\,(\upsilon_{\phi,\tau}[1], \upsilon_{\lambda,\tau}[1])\big)\end{aligned} \tag{3.35}$$

    The condition is met if these two ranges overlap:

$$\phi\, C\, \lambda \wedge \tau_\phi = \tau_\lambda \implies \zeta_{\phi,\lambda} \;\&\&\; \eta_{\phi,\lambda} \tag{3.36}$$

  - If $\tau_\phi < \tau_\lambda$, the time span between the recorded departure of the former run from stop τ_ϕ and the registered arrival of the latter at τ_λ should be greater of equal than the lower bound of the total travel time between them.
  - If $\tau_\phi > \tau_\lambda$, the two candidate runs do not originate from a single one that changed its id once.

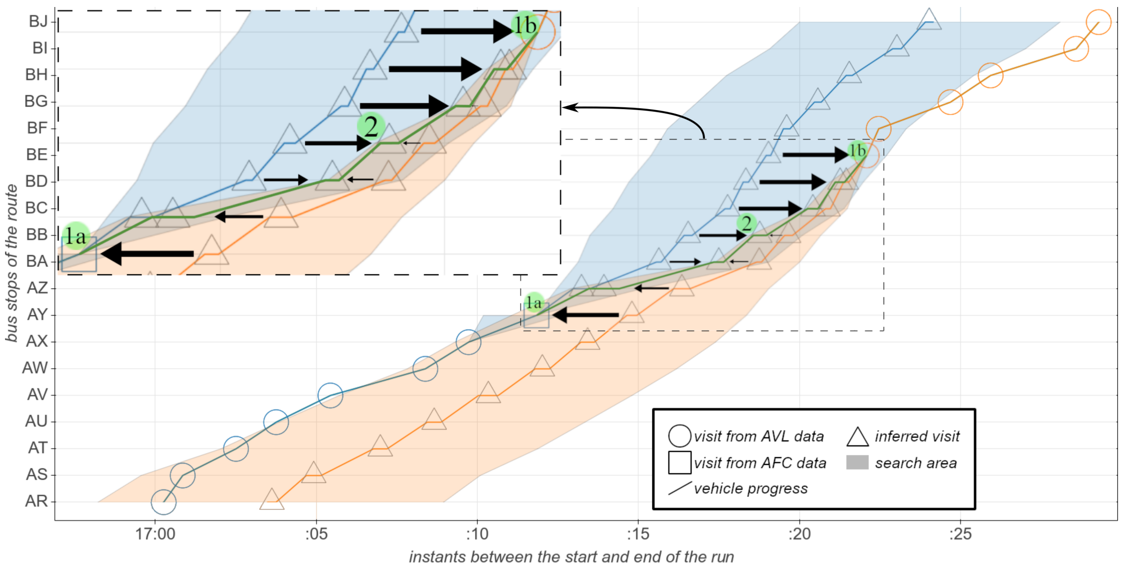


Figure 3.6: Merging of 2 runs (former blue, latter orange, modified visits green line)

Equation (3.37) summarizes these criteria:

$$\phi \; C \; \lambda \iff \begin{cases} \zeta_{\phi,\lambda} \;\&\&\; \eta_{\phi,\lambda} & \text{if} \quad \tau_\phi = \tau_\lambda \\ \upsilon_{\lambda,\tau_\lambda}[0] - \upsilon_{\phi,\tau_\phi}[1] \leq \sum_{\tau=\tau_\lambda}^{\tau_\phi - 1} \mu_i & \text{if} \quad \tau_\phi < \tau_\lambda \end{cases} \tag{3.37}$$

3.3.4.2 Update vehicle runs characterization tables

The runs that comply with eqs. (3.32), (3.33) and (3.37) are merged in a new one. Arrival and departure times at any stop between τ_ϕ and τ_λ are chosen as the most likely ones, according to dwell and travel time distributions; and the information to link them is stored in the columns *merged_run* and *merged_runs*. Figure 3.6 illustrates an example, where the methodology detects that entries that were on a first approach used to assert that two different runs of a route between stops AR and BJ took place (blue and orange) are actually part of a single one, and then re-evaluate unknown calls where this fact may be used to improve arrival and departure estimations:

1 The two runs proposed by section 3.3.3 of this methodology comply with the conditions that identify them as a single one, with an intermediate *vehicle id* change:

- Their time buffers overlap in at least a stop, as can be seen observing the parts colored blue and orange.

  - Considering only calls backed by IPTS entries, the latest from one of the runs (visit of blue at AY, 17:11:50, 1a) happens in a stop prior to the earliest from the other (visit of orange at BE, 17:22:05, 1b). The span between the departure from the former and the arrival at the latter is 10m15s, while the sum of the lower bounds of the route legs involved is 2m23s, which means that a single vehicle could be responsible for both.

2 Intermediate arrival and departure times between AY and BE are re-calculated. Instead of their mean values according to their respective distributions and the departure from AY or the arrival at BE; they adopt the most likely combination of values that satisfy both conditions at the same time.

3.3.5 Ascribe vehicle runs to scheduled runs and update visit time spans

This part of the methodology has several goals: firstly, to differentiate between planned runs that were materialized or not; to identify non-scheduled, extra runs; and to remove inferred *visits to stops* that did not actually take place, for those runs that are successfully identified as starting downstream the initial terminus stop.

After a run has been linked to its scheduled beginning, the additional information from the *schedule* table may be used to further refine arrival and departure times. These are the proposed steps, also shown as a flowchart in fig. 3.7:

- A loop is performed over all (*scheduled beginning*, *run*) pairs where the latter's *departure* from the planned stop falls within the former's buffer q (eq. 3.27), considering those that share the same *vehicle id* first, and then ordered by the absolute value of the time span between the run's departure and the scheduled start, ascending. Unless either of them has already been linked, they become so with each other.

- For each pairing that is found, starting at the initial terminus of the whole route, inferred *visits to stops* are consecutively removed from the run, until one that it backed by AFC or AVL records is reached.

- If the planning subsystem registered the start of the run, the plausibility of its corresponding time range n (eq. 3.27) is evaluated, utilizing the appropriate feasibility range stored in the *search_ranges* table (if not available, one is computed utilizing the closest downstream data-supported call of the run). If n is judged credible, two situations may occur:

  - If the initial call of the run was previously deduced from other IPTS data, the available information is combined to obtain the earlier and latest presence of the bus at that stop.

- Otherwise, n is used as the [arrival, departure] range at the beginning of the run.

- Downstream inferred visits, up to the first one sustained by IPTS data, are improved to their new most likely values, considering the total travel time between the scheduled run start and that first known data point, and travel and dwell time distributions.

Figure 3.8 shows the first stops from an example run:

1 Its initial estimation has been linked to a planned start at stop AF, with a gap between their inferred and planned departure times of 51s.

2 The stops upstream the planned start are not backed by any IPTS records and are erased.

3 In this case, the arrival and departure were logged by the scheduling subsystem at 07:25:39 and 07:26:01, respectively. Since these times falls within the *search range* for that run at stop AF ([07:23:39, 07:26:13]), they are accepted as what really happened.

4 Visits to AG, AH, AI, and AJ are also recalculated, considering the new information.

3.3.6 Select vehicle runs backed by enough information

The last task is to establish and apply criteria to accept or reject each of the possible runs that have been identified by this methodology. It is suggested to set boundaries that consider these features (eq. 3.38):

- Whether or not a planned departure was mapped to the run (w). In the latter case, also consider if the *id* of the vehicle is the same in both databases (p), and whether the scheduling subsystem registered a compatible starting time (ν).

- The total number h of *boarding groups* attributed to the run, as described in section 3.4.

- How many *visits* of that run stem from AVL data (f).

- The number of stops between the earliest and latest visits supported by IPTS data (l).

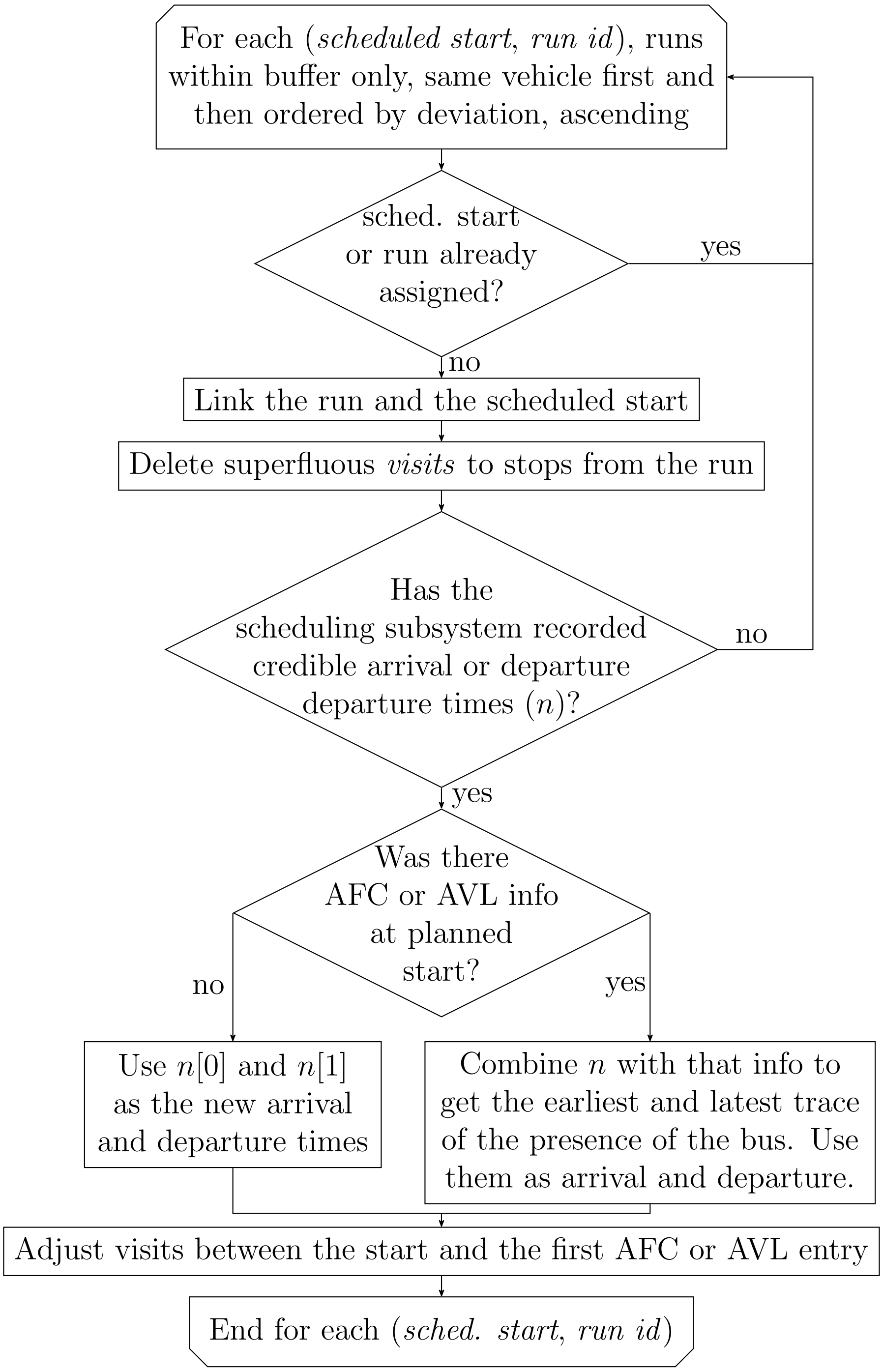


Figure 3.7: Associating scheduled and inferred vehicle runs

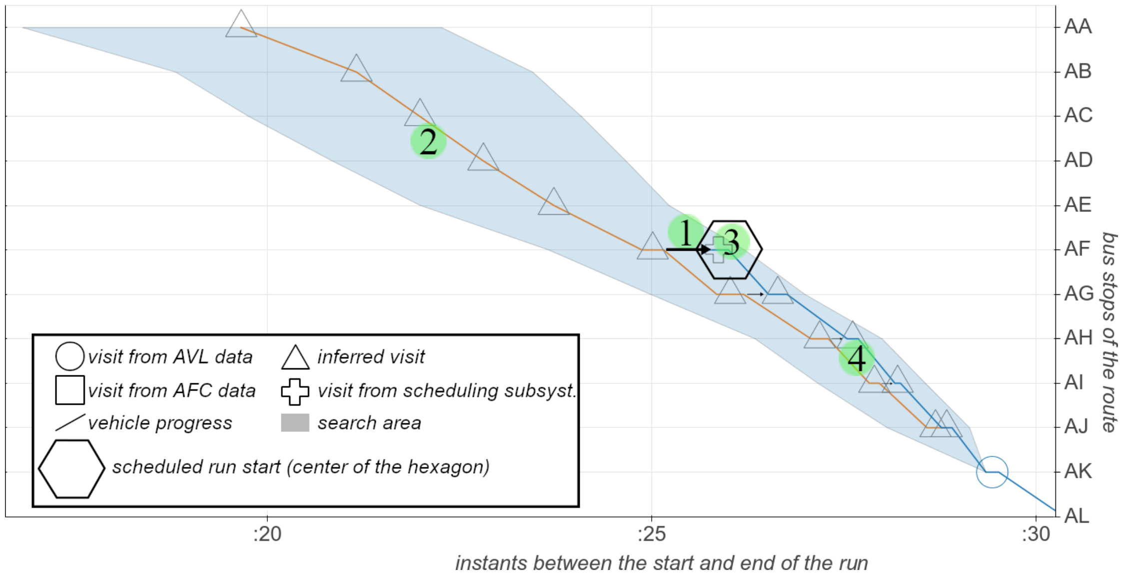


Figure 3.8: Improvement of arrival and departure times of a run once its likely planned beginning is identified. Final characterization blue , removed or modified entries orange .

$$
\begin{array}{rll}
w: & \text{run was planned} & \text{boolean} \\
p: & \text{the planned vehicle was utilized} & \text{boolean} \\
\nu: & \text{the scheduling subsystem registered a valid starting time} & \text{boolean} \\
h: & \text{boarding groups count} & h \in \mathbb{N} \\
f: & \text{visits backed by AVL data} & f \in \mathbb{N} \\
l: & \text{longest range between stops backed by IPTS data} & l \in \mathbb{N}
\end{array}
\tag{3.38}
$$

As a first step, candidate runs of a vehicle following alternative configurations of a line which overlap in time are compared, choosing the one with greater backing from the previously enumerated features. This happens for instance when a line changes part of its path at certain times during the day: some IPTS records may end up pointing at the wrong alternative.

The features of the remaining possible runs are evaluated to filter out those that in all likelihood did not happen. Figure 3.9 provides an example, analyzing two consecutive possible runs of a vehicle, covering complementary subroutes between AI and BF termini, both composed of 23 route legs. Only 2 consecutive entries from the *avl_coalesced* table hint at the existence of the earlier (1); while the latter is supported by a planned run of that vehicle for which the scheduling subsystem recorded the first call (2), 4 *boarding groups* (shown in 3a and 3b),

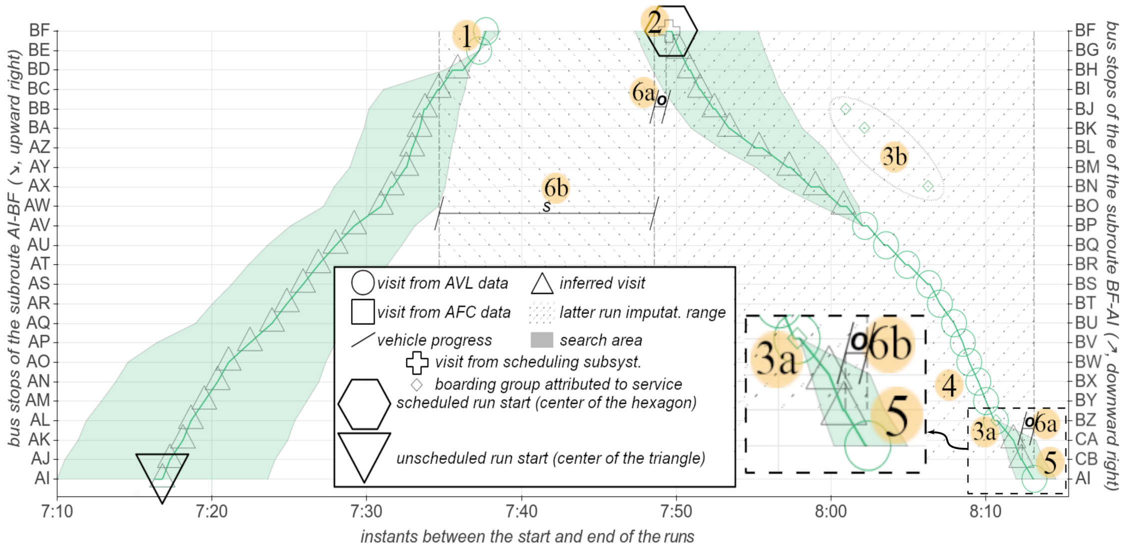


Figure 3.9: Analysis of the IPTS data that sustain each of two consecutive potential runs of a vehicle and imputation of boarding groups

12 *avl_coalesced* rows (4), and by the fact that the span between its earliest (2) and latest (5) calls obtained from recorded IPTS observations covers the whole route. The former almost certainly did not happen, while the latter most likely did.

3.4 Boarding groups imputation

Once the calls of all possible runs have been defined and refined, boarding groups are firstly mapped to a run, and then to the stop where they took place.

The first task requires to build two imputation ranges for each run:

- A 'default' one, which covers from the moment when the vehicle arrived at the initial stop to the moment it left the next-to-last one (as no AFC events should be assigned to the last stop of a run), extended forward and backwards in time the parameter $\boldsymbol{o}$ (identified as 6a for the latter case of fig. 3.9), which allows for some **leeway between AVL and AFC events**, to cover cases such as validations after the vehicle leaves the stop or minor clock desynchronizations. It has been marked with this pattern: .

$$\boldsymbol{o} : \text{AFC leeway} \quad \text{time} \tag{3.39}$$

- An 'extension', stretching backwards from the previous 'default' range, covering an interval equal to the upper bound for the headway of the route the route (s from eq. 3.2, marked as 6b). It is used to consider the possibility of passengers boarding the vehicle well before the detected start of a run. This may happen for instance, if they enter a vehicle as it arrives at a terminus, and wait inside for its next departure). It has been marked with this pattern: .

Then, each *boarding group* is treated in a three-step process:

- Firstly, it is assigned to the run whose 'default' time range it overlaps and that refers to the same *vehicle* and *route*. If no run is found, the route id sameness requirement is dropped, to treat those cases where the ticketing subsystem state did not reflect the route the vehicle was really following.
- If no run was found, the procedure described in the previous point is carried out again, utilizing the 'extension' range of each run instead.
- Any *boarding group* left is not linked to a run.

Finally, the proper stop within the run is identified, considering all its calls but the last one:

- If the gap between the *boarding_range* (eq. 3.3) and the *visit_range* (eq. 3.10) at the stop specified by the *boarding group* is less or equal to the maximum leeway $\boldsymbol{o}$ (eq. 3.39), that stop is accepted as the one where the travelers got on the bus (e.g., 3a in fig. 3.9).
- Otherwise, it is assumed that the AFC did not properly identify the *id* of the stop. The one from the closest call of the vehicle is chosen instead (e.g., 3b, where the 3 *boarding groups* that were recorded as happening at stops BJ, BK, and BN are respectively assigned to BO, BP, and BS).

3.5 Passenger trips inference

With the aim to apply this work to IPTSs that only require validation when getting on a vehicle, in this section the trip chaining model is applied to vehicle runs and passenger boardings to infer the unknown alighting stops of the rides that have taken place. After an introduction and an exposition of the core reasoning behind the algorithm which stems from it, the different improvements that are utilized to improve the results are described.

3.5.1 Trip chaining demand models

While traditional travel demand models assume that the choices made for each trip are independent of those made for other trips in the same journey, those based on trip or activity chains can reflect the increasingly complex way humans behave, particularly in urban areas. The former assumption is reasonable for 'round-trip' journeys, consisting of two symmetric trips with both activity centers in common; while the latter allows for trips to influence each other in in complex ways (for

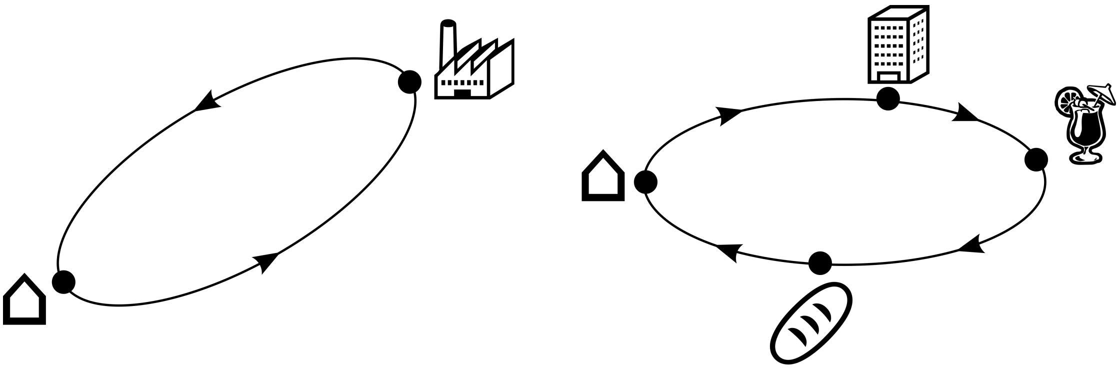


Figure 3.10: A 'Round-trip' (left) vs chained trips (right)

instance, commuters that leave their cars at home are very unlikely to be able to drive them until they return) [1].

Figure 3.10 illustrates the difference between a 'round-trip' journey on the left, and another composed by a series of chained trips on the right. In the former case, the person just goes to work (maybe driving a car), and when the shift ends, returns; while in the latter, the commuter leaves home (maybe using some combination of public transport modes), performs the 'primary activity' of the journey (working), and after that, before returning home, decides to perform 2 other consecutive trips to carry out activities at different chosen locations before it is time to go back home.

3.5.2 A 'base' trip chaining model

To facilitate its exposition, this 'base model' already incorporates some subsequent ideas that deal with the last ride of the day ([80, 4]).

A 'trip' is the movement of a traveler through a public transport system. Its beginning and end are linked with the conclusion of an activity and the start of another, respectively. The i^{th} trip that has been carried out by the holder of the SC c during the day d will be referred as $\mathrm{tr}_{c,d,i}$. Each of the rides that happen during it are named $\mathrm{ri}_{c,d,i,j}$, with the index j indicating their temporal succession. The different variables associated with each ride can thus be identified utilizing the same sub-indexes. E.g.: 'boarding time' ($\mathrm{bt}_{c,d,i,j}$), 'boarding stop id' ($\mathrm{bs}_{c,d,i,j}$), 'alighting time' ($\mathrm{at}_{c,d,i,j}$), 'alighting stop id' ($\mathrm{as}_{c,d,i,j}$), etc.

$$
\begin{array}{rll}
c: & \text{card id} & \text{text} \\
d: & \text{day} & \text{date} \\
\mathrm{tr}_{c,d,i}: & i^{\text{th}} \text{ trip for card } c \text{ during day } d & \mathrm{tr}_{c,d,i} \in \mathbb{N} - \{0\} \\
\mathrm{or}_{c,d,i}: & \text{Origin stop of trip } \mathrm{tr}_{c,d,i} & \mathrm{or}_{c,d,i} \in \mathbb{N} \\
\mathrm{de}_{c,d,i}: & \text{Destination stop of trip } \mathrm{tr}_{c,d,i} & \mathrm{de}_{c,d,i} \in \mathbb{N} \\
\mathrm{ri}_{c,d,i,j}: & j^{\text{th}} \text{ ride for trip } \mathrm{tr}_{c,d,i} & \mathrm{ri}_{c,d,i,j} \in \mathbb{N} - \{0\} \\
\mathrm{bt}_{c,d,i,j}: & \text{boarding time of ride } \mathrm{ri}_{c,d,i,j} & \text{full date (time)} \\
\mathrm{bs}_{c,d,i,j}: & \text{boarding stop id of ride } \mathrm{ri}_{c,d,i,j} & \mathrm{bs}_{c,d,i,j} \in \mathbb{N} - \{0\} \\
\mathrm{at}_{c,d,i,j}: & \text{alighting time of ride } \mathrm{ri}_{c,d,i,j} & \text{full date (time)} \\
\mathrm{as}_{c,d,i,j}: & \text{alighting stop id of ride } \mathrm{ri}_{c,d,i,j} & \mathrm{as}_{c,d,i,j} \in \mathbb{N} - \{0\} \\
\mathrm{pa}_{c,d,i,j}: & \text{passengers that boarded the vehicle} & \in \mathbb{N} - \{0\} \\
 & \text{using the same SC (page 66)} & \\
\mathrm{tr}'_{c,d,i}: & \text{trip for which an alighting could not be} & \mathrm{tr}_{c,d,i} \in \mathbb{N} - \{0\} \\
 & \text{found. `}'\text{' will be used to signify the rest of} & \\
 & \text{variables in the same way. E.g.: } \nexists \mathrm{de}'_{c,d,i} &
\end{array}
\tag{3.40}
$$

The basic reasoning behind this model is that people who use the public bus for their daily movements will most likely leave a vehicle at the stop closest to the one utilized to get on the next bus of the same day, as long as they are no more than $\boldsymbol{M}$ apart.

$$
\boldsymbol{M}: \text{max. dist. from alighting to next boarding} \quad \boldsymbol{M} \in \mathbb{R} \tag{3.41}
$$

To find out if the rides that correspond to these 2 boarding events are part of the same trip or are the end and beginning of 2 different ones with an intermediate activity, the methodology focuses on the inferred alighting of the former ride, and the known boarding on the latter: whether the distance between them is greater than $\boldsymbol{mdp}$, and if the time that passed between them exceeded $\boldsymbol{mtt}$.

$$
\begin{array}{rll}
\boldsymbol{mdp}: & \text{max. dist. between stops during a transfer} & \boldsymbol{mdp} \in \mathbb{R} \\
\boldsymbol{mtt}: & \text{max. t. from alighting to boarding during a transfer} & \text{time}
\end{array}
\tag{3.42}
$$

$$
\forall \left(\mathrm{ri}_{c,d,i,j}, \mathrm{ri}_{c,d,k,l}\right) \Big| \nexists \mathrm{ri}_{c,d,m,n} \mid \mathrm{bt}_{c,d,i,j} < \mathrm{bt}_{c,d,m,n} < \mathrm{bt}_{c,d,k,l} :
$$
$$
\begin{cases}
\mathrm{bt}_{c,d,i,j+1} - \mathrm{at}_{c,d,i,j} \leq \boldsymbol{mtt} \wedge \mathit{dist}\left(\mathrm{as}_{c,d,i,j}, \mathrm{bs}_{c,d,i,j+1}\right) \leq \boldsymbol{mdp} \\
\qquad \iff k = i; l = j + 1 \\
\qquad\qquad \mathrm{ri}_{c,d,i,j} \text{ and } \mathrm{ri}_{c,d,i,j+1} \text{ are part of the same trip.} \\
\text{Otherwise} \iff k = i + 1; l = 1 \\
\qquad\qquad \mathrm{ri}_{c,d,i,j} \text{ and } \mathrm{ri}_{c,d,i,j+1} \text{ belong to different trips.}
\end{cases}
\tag{3.43}
$$

Figure 3.11 shows an, where a user has performed 4 rides in a single day: 2 on the red line, at 7:00 and 17:30; and another pair on the green one, at 7:20 and 17:00. The trip chaining model tells us that the most likely alighting point of each ride along its corresponding run (¿↓?) is the closest to the stop where the next one is boarded; while accepting that the first and last rides of the day are respectively the first and last legs of trips from and to home, the alighting stop for the latest ride can be assumed to be the closest to the earliest boarding. Supposing that the traveler has enough time to walk from one stop to the next during transfers, the by far most likely interpretation is that 2 trips have happened:

- One in the morning, probably from home to work, composed by 2 rides:
  - The first on the red line, from 7:00 to ¿↓1?, where the alighting time would be obtained from definition of the vehicle run.
  - The second on the green line, from 7:20 to ¿↓2?.
- And the trip back home in the evening, also materialized in 2 rides:
  - Firstly on the green line, from 17:00 to ¿↓3?.
  - And finally on the red line, from 17:30 to ¿↓4?.

Since for each SC on each day the boarding to each ride is utilized to guess the alighting from the previous; the last one requires particular consideration. Two alternatives are:

1. Use the first boarding of the day, as already explained on the example of (fig. 3.11). Most transit users depart from home on the first trip of the day, and return to it on the last.
2. For those days with only one trip, the strategy describe above does not work. In these cases, if the next day registered any validations, the first one is evaluated as the possible destination of the last trip of the previous day. This may happen for instance if the user went to work in the morning using the public bus, but returned home by other means.

After applying this reasoning to all entries from all SCs, the output is a series of trips, composed by one or more rides each, plus a series of boarding events for which no alighting could be inferred.

3.5.3 Trip chaining model improvements

Several improvements from the existing literature have been implemented over the 'base' methodology to better define the rides and their encompassing trips, aiming to solve the different problems encountered during its application.

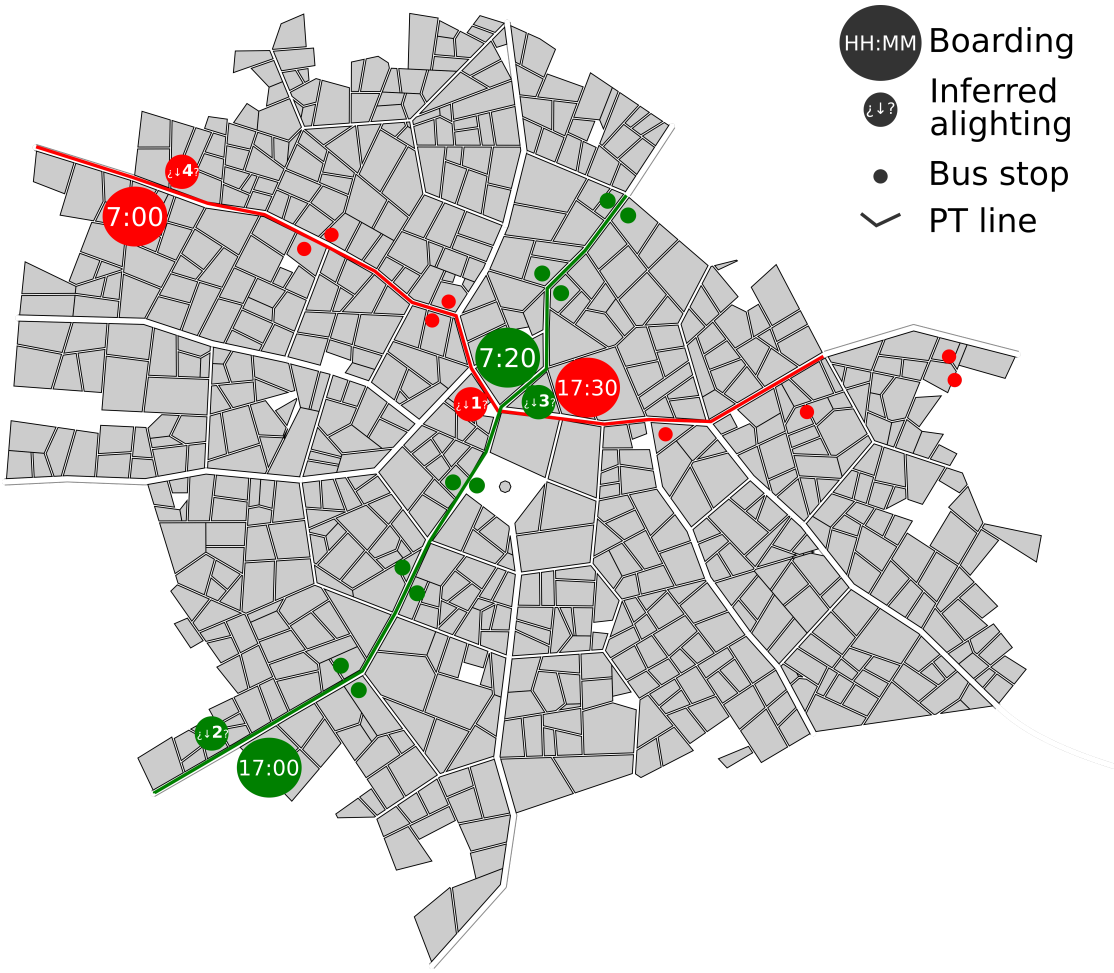


Figure 3.11: Trip chaining model example [94]

Each of the following subsections describes an issue and the steps taken to tackle it.

3.5.3.1 Alighting stop inference refinement

Always choosing the closest stop from the previous ride as the alighting point can be unrealistic. For instance, if the previous route passes almost as close both on its way forwards and backwards to the boarding to the next one, the passenger will almost certainly leave the bus at the first opportunity, not spending a long extra time on the vehicle to avoid walking a few meters. A solution to this problem has already been formulated [12], and consists in choosing, among all candidate alightings of the previous run, one that:

1. Occurs at a time compatible with the next boarding (i.e., the traveler must have time to walk to the next stop).
2. Minimizes the 'generalized time' (gt), which is the alighting time from the previous run, plus the estimated walking time to the next boarding stop multiplied by a '***walking time penalization factor***' ($\boldsymbol{f_w}$). The interpretation of this condition is that users tend to choose the alighting stop that provides the greater time margin until their next boarding, but they also prefer, to a degree controlled by the parameter $\boldsymbol{f_w}$, riding the bus to walking. To check it, the maximum speed at which users are supposed to walk, $\boldsymbol{ws}$, must be chosen.

$$\begin{aligned} \boldsymbol{ws} &: \text{maximum walking speed} && \text{speed} \\ \boldsymbol{f_w} &: \text{walk penalization factor} && \boldsymbol{f_w} \in \mathbb{R} \\ gt &: \text{generalized time} && \textit{time} \end{aligned} \tag{3.44}$$

$$gt = t_{\text{alighting}} + \frac{dist\,(\text{possible previous alight., next board.})}{\boldsymbol{ws}} \cdot \boldsymbol{f_w} \tag{3.45}$$

3.5.3.2 Alighting inference for the last trip of the day and other incomplete trips

The 'base' model does not provide a destination for the last trip of the day if it is also the first, and the next day the card was not utilized, nor does it if the chain of trips is broken because an alighting and the consecutive boarding are too distant (eq. 3.41). Moreover, if a possible destination pd is too close to the start of the trip (less than $\boldsymbol{md}$ apart), it is deemed as not realistic and also rejected:

$$\begin{aligned} \boldsymbol{md} &: \text{last trip min. dist. from board. to alight.} && \boldsymbol{md} \in \mathbb{R} \\ pa &: \text{Stop where the last alighting of the day may have occurred} && pa \in \mathbb{N} - \{0\} \end{aligned} \tag{3.46}$$

$$\begin{aligned} &\text{If } \mathrm{ri}_{c,d,i,j} \text{ is the last ride of its day for its card:} \\ &\qquad dist\,(\mathrm{bs}_{c,d,i,j}, pa) > \boldsymbol{md} \implies \mathrm{as}_{c,d,i,j} \neq pa \end{aligned} \tag{3.47}$$

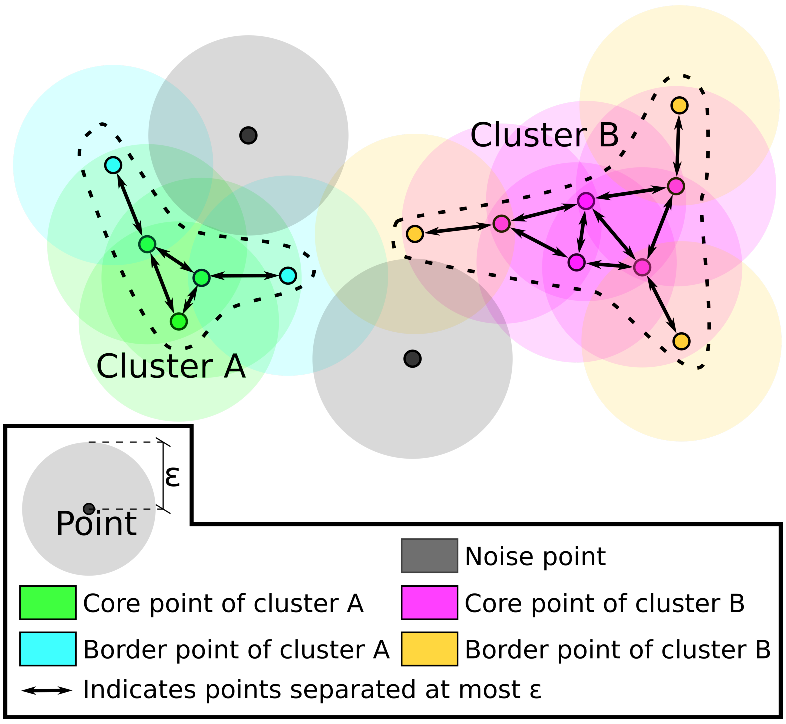


Figure 3.12: DBSCAN algorithm example (minPts=3 points)

For all these cases, another approach already utilized in the literature to identify activity locations is applied, based on the DBSCAN algorithm. This subsection firstly describes it, and then elaborates on how it is applied to this situation.

3.5.3.2.1 The DBSCAN algorithm

It is a method that, given a set of points, classifies them as either belonging to a cluster (a 'dense region'), or being noise. Moreover, points part of a cluster can be part of its 'core' or its 'border'. It requires a distance function and two parameters: ϵ, the 'maximum density reach distance'; and minPts, the minimum number of points within reach for a point to be considered as 'core'. The detailed rules of this algorithm to classify a point are:

- It is a 'core point' if, including itself, there are at least minPts points within a distance ϵ of it.
- It is a 'border point' if there are less than minPts points within a distance ϵ of it, being at least one of these a 'core point'.
- Otherwise, it is considered 'noise'.

Figure 3.12 shows an example of its application.

3.5.3.2.2 Last or otherwise incomplete destination deduction

Those trips for which none of the first two approaches (page 56) yield a valid result (eq. 3.47), are held back until all the entries available from that card are processed; as also are those that cannot be completed because the closest possible alighting to the next boarding is too far away from it (eq. 3.41). Then, DBSCAN, explained in 3.5.3.2.1, is applied on the last alightings of fully defined trips, to detect activity centers which regularly are, during certain periods of time during a day, the destination of the trips of the user.

In this work, for two alightings to be within the neighborhood of each other, they need to be situated on the map less than a distance $\boldsymbol{\epsilon_d}$ apart, and no more than a time $\boldsymbol{\epsilon_t}$ should pass from the earlier to the latest. This is equivalent to evaluating the proximity of the alightings of the last rides of two trips of the same SC on the same day ,$\mathrm{ri}_{c,d,i,j}$ and $\mathrm{ri}_{c,d,k,l}$, utilizing a distance function $dist'$, related to the Euclidean distance $dist$ as shown in eq. (3.49). In this equation, the * signifies that only the time of the day of the alighting is being considered, not the full date.

$$\begin{aligned}\boldsymbol{\epsilon_d} &: \text{maximum distance between alightings} && \boldsymbol{\epsilon_d} \in \mathbb{R} \\ \boldsymbol{\epsilon_t} &: \text{maximum span between alightings} && \textit{time} \\ \boldsymbol{minPts} &: \text{min. neighbors to be a core point} && \boldsymbol{minPts} \in \mathbb{N} - \{0\}\end{aligned} \tag{3.48}$$

$$\begin{gathered}\forall \left(\mathrm{ri}_{c,d,i,j}, \mathrm{ri}_{c,d,k,l}\right) \Big| \nexists \mathrm{ri}_{c,d,i,m} \mid m > j \ \wedge \ \nexists \mathrm{ri}_{c,d,k,n} \mid n > l : \\ \begin{cases} |\mathrm{at}^*_{c,d,i,j} - \mathrm{at}^*_{c,d,k,l}| \leq \boldsymbol{\epsilon_t} \\ \qquad\qquad \iff dist'\left(\mathrm{as}_{c,d,i,j}, \mathrm{as}_{c,d,k,l}\right) = dist\left(\mathrm{as}_{c,d,i,j}, \mathrm{as}_{c,d,k,l}\right) \\ \text{Otherwise} \iff dist'\left(\mathrm{as}_{c,d,i,j}, \mathrm{as}_{c,d,k,l}\right) = +\infty \end{cases}\end{gathered} \tag{3.49}$$

These spatiotemporal clusters provide a series of zones in the city where users regularly arrive within a certain time window (*destinations*). $\boldsymbol{\epsilon_d}$, $\boldsymbol{\epsilon_t}$ and $\boldsymbol{minPts}$ decide in each case what is a ‘regular trip’.

Rides and runs datasets may be partitioned following known features that influence the mobility patterns in the city (for instance, day of the week, or whether schools are open or not), establishing different values for $\boldsymbol{\epsilon_d}$, $\boldsymbol{\epsilon_t}$ and $\boldsymbol{minPts}$ for each partition.

The spatial location of these frequent activity centers is set in the center of mass of the stops of each cluster; with an associated user arrival window as the narrowest of two possible ones:

- From the earliest to the latest times of the day, considering all elements of the cluster.
- A fixed minimum amplitude $\boldsymbol{mtw}$.

$$\boldsymbol{mtw} : \text{Minimum time window} \quad \textit{time} \tag{3.50}$$

Finally, each incomplete trip that was held back is evaluated against the frequent *destinations* suggested by the DBSCAN. A valid *destination* should comply with eq. (3.47), and be reachable within its time window, supposing that the user walks directly to it after alighting. If there are several possibilities, the one that minimizes its $\boldsymbol{gt}$ (eq. 3.45) is chosen.

3.5.3.3 Trip maximum duration

Knowledge of the IPTS allows to set a threshold duration $\boldsymbol{mtr}$ of the trips in the city. Once a trip has surpassed it, no more rides are added to it.

$$\boldsymbol{mtr}: \text{Trip threshold duration} \quad \text{time} \tag{3.51}$$

3.5.3.4 Improved activity detection

In certain IPTS it is especially difficult, just using the $\boldsymbol{mtt}$ and $\boldsymbol{mdp}$ (eq. 3.42), to tell if between an inferred alighting and the next boarding users are transferring between the legs of a single trip, or if they are performing an activity amidst trips; because sometimes (but not always) the time dedicated to an activity is quite short before starting another trip to the next one. This behavior can be fomented by pricing schemes that offer free rides during a period after the first tap-in, and by a transportation offer with high spatial density and frequencies.

To ameliorate this problem, this work utilizes the two strategies described in the following subsections.

3.5.3.4.1 Detection of alightings close to stops that would have been reachable from one of the previous rides of the trip

While building a trip, if the inferred alighting stop of the latest ride under consideration could have been reached sooner if the user had stayed longer on a previous ride (directly, or getting off the vehicle at a stop closer than $\boldsymbol{vcd}$ to it), it is assumed that the user chose to get off earlier to reach, at that stop or further along the current chain of rides under analysis, a destination where to perform some short intermediate activity. Thus, the chain of rides under analysis is split between those that leave the greatest time margin for an in-between activity. The former part is considered a whole trip, and the latter is used as part of a new one to be completed.

$$\boldsymbol{vcd}: \text{Bus stop buffer} \quad \boldsymbol{vcd} \in \mathbb{R} \tag{3.52}$$

An example of the kind of situation solved by this enhancement is shown in fig. 3.13 where, to make it clearer, routes B and C share the same stops in the segment of interest, even though they could be separated at most $\boldsymbol{vcd}$ and still be detected. As can be seen, the user boards a bus of the route A (1); changes to route B at 2 with 3 min to spare; switches again at 3 this time to route C, with a margin of 19 min minutes; and arrives at 09:47 for an activity at 4.

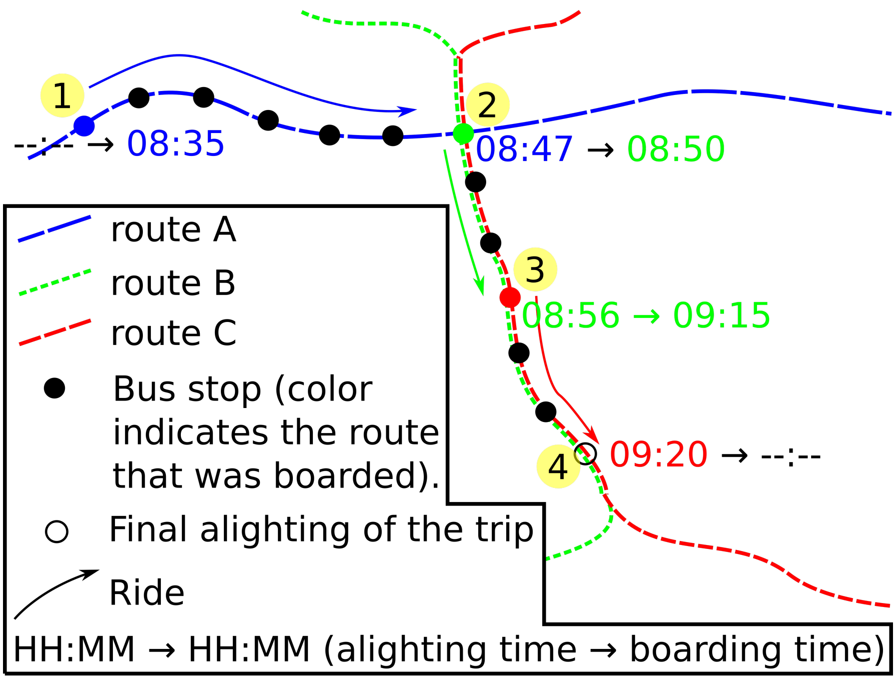


Figure 3.13: Detection of a short intermediate activity thanks to an otherwise counterproductive transfer

If $\boldsymbol{mtt} > 19\,\mathrm{min}$, the basic criterion of the trip chaining model would state that was happened at 3 was a transfer; but that really does not make sense, since if all the user wanted was to arrive to 4, there was no need to change buses at 3. Thus, we conclude that at 3 or further down the chain of rides, but before arriving to 4, the user carried out some short-length activity. In this case the only choice is 3, but if there were multiple possibilities, the one that offered the most leeway to perform an activity would be chosen.

3.5.3.4.2 Directness check

Once a possible trip has been fully defined from origin to destination, if the sum of the distances covered by the rides that compose it is greater than the direct distance from origin to destination, multiplied by an '***circuity ratio***' ($\boldsymbol{cr}$, [13]), the candidate trip is, as in 3.5.3.4.1, split between the rides that leave the greatest time margin for an in-between activity. The former part is considered a whole trip, and the latter is used as part of a new one to be completed.

$$\boldsymbol{cr} : \text{circuity ratio} \tag{3.53}$$

Figure 3.12 shows a graphic interpretation of this parameter, linking it to the sum of the legs of the isosceles triangle OFD, which has as its base the segment between the initial boarding and final alighting of the trip ($\overline{\mathrm{OD}}$), and a vertex

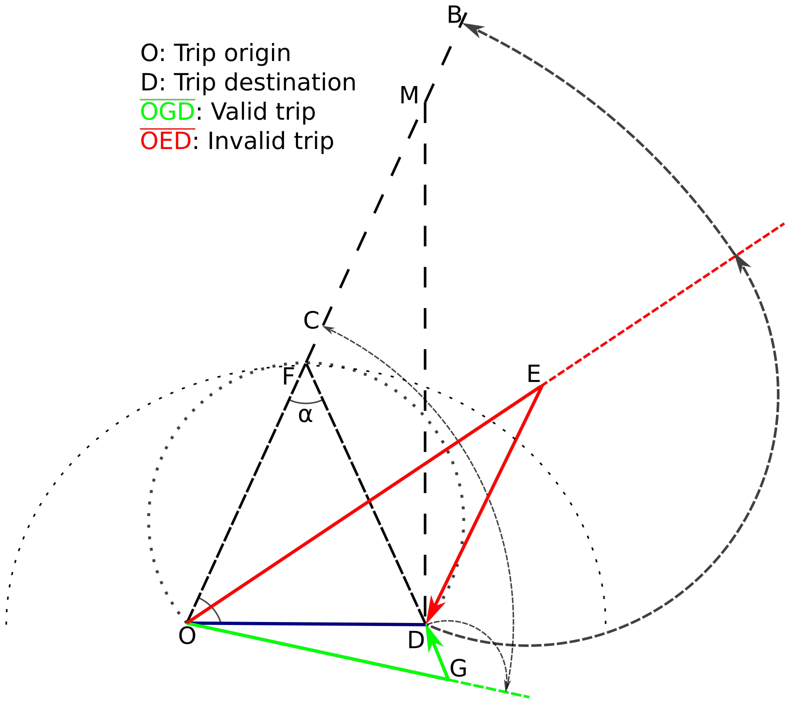


Figure 3.14: Graphic interpretation of the circuity ratio

angle α. The segment $\overline{\text{OM}}$ is used as the upper limit of the total traveled distance of a valid trip between O and D, which means that $\overline{\text{OM}} = \overline{\text{OD}} \cdot \boldsymbol{cr}$. Also, it can be seen that $\overline{\text{OM}} = \overline{\text{OF}} + \overline{\text{DF}} = \frac{\overline{\text{OD}}}{\sin\frac{\alpha}{2}}$; thus $\boldsymbol{cr} = \frac{1}{\sin\frac{\alpha}{2}}$, and $\boldsymbol{cr} \in [1, +\infty]$: the flatter α is, the less a trip is allowed to deviate from a straight line. Of the two possible trips, $\overline{\text{OGD}}$ is shorter than $\overline{\text{OM}}$, ($\overline{\text{OC}} < \overline{\text{OM}}$), so it is accepted as a single trip; while $\overline{\text{OED}}$ is longer than $\overline{\text{OM}}$, ($\overline{\text{OB}} > \overline{\text{OM}}$), and is split, which means assuming that the traveler performed some activity at E. The area defined by $\boldsymbol{cr}$ is an ellipse with the origin and destination of the trip as foci.

3.5.4 Final trip chaining model

The flowchart in fig. 3.15 shows the overall process once all enhancements previously described have been implemented to the ‘base’ algorithm. It relies on 3 tables among which each ride moves as it is processed.

- ‘*rua*’ (‘*rides under analysis*’), for those rides that are defined as a SC is processed, but it is not clear yet how they should be grouped in trips.
- ‘*roc*’ (‘*rides of one card*’), that stores the rides of the trips of the SC being analyzed that have been completely defined.
- ‘*rides*’, which stores the output of the process. Its columns allow to identify the SC, trip, and order within it of each ride; and contain the information of

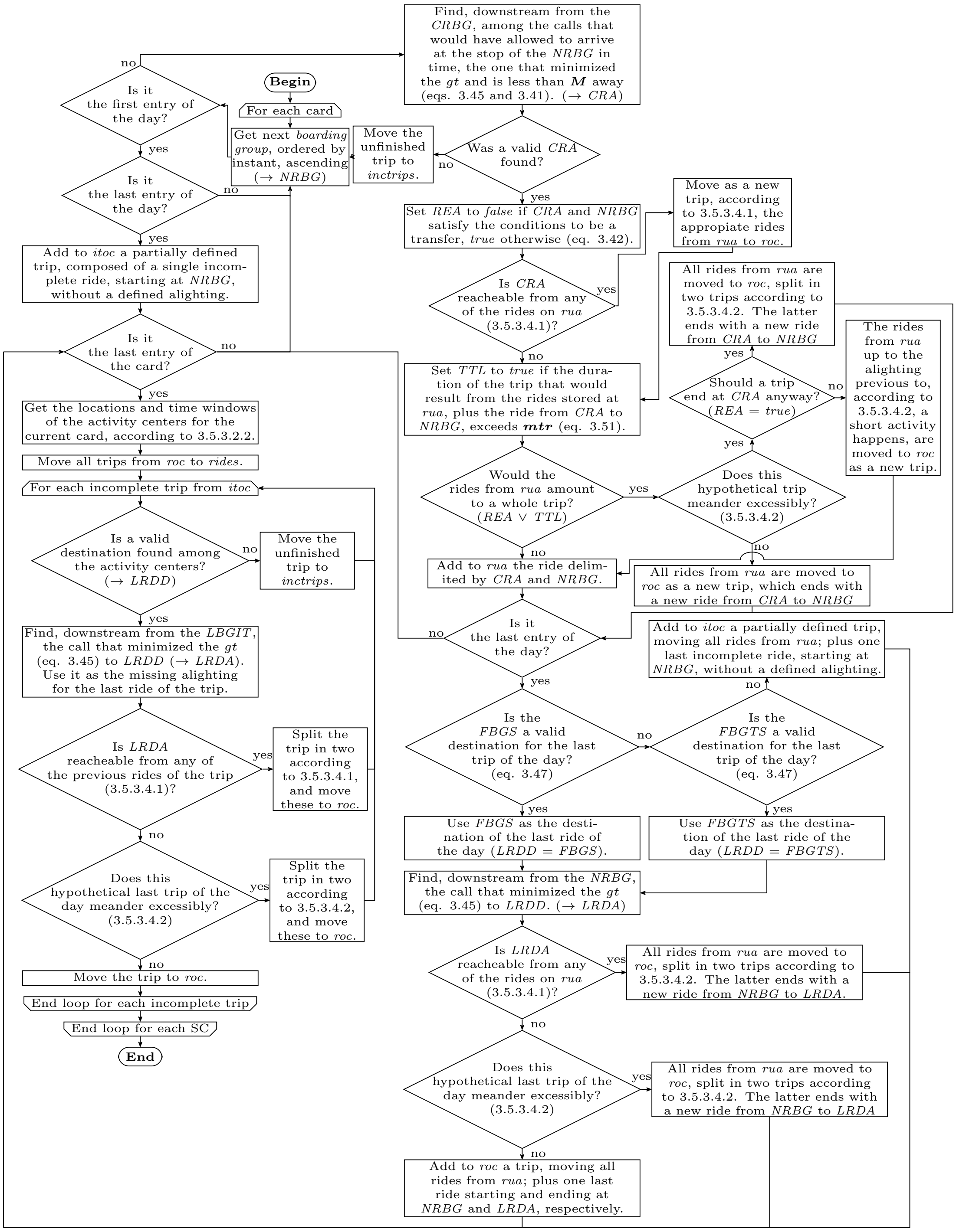


Figure 3.15: Trip chaining model flowchart

interest regarding their boarding and alighting events.

Two other tables store boarding events without an inferred alighting:

- '*itoc*' ('*incomplete trips of one card*'): as the program advances through the boarding events of one SC, those trips which are not immediately assigned a final alighting are stored here, to try to find one after all entries of that SC have been processed utilizing the enhancement described in 3.5.3.2.2.
- Those trips that the methodology has finally failed to link to a destination are stored in the table '*inctrips*' ('*incomplete trips*').

The algorithm makes decisions based on the entries determined by its progress as it iterates over the different *boarding groups* of each SC:

- '*NRGB*' ('*next ride boarding group*'): the boarding group most recently retrieved.
- '*CRBG*' ('*current ride boarding group*'): the boarding group retrieved just before *NRGB*.
- '*CRA*' ('*current ride alighting*'): the event where the ride that started with the *CRBG* ends. It is inferred thanks to the *NRGB* position and instant.
- '*OBG*' ('*origin boarding group*'): the boarding group of the initial ride of the trip that is currently being assembled by the program.
- '*FBGS*' ('*first boarding group*'): the bus stop of the boarding group of the first ride of the day.
- '*FBGTS*' ('*first boarding group of tomorrow*'): the bus stop of the boarding group of the first ride of the next day.
- '*REA*' ('*ride ends with activity*'): *true* if the *CRA* and *NRGB* are too far away or happen too apart to be part of a transfer, *false* otherwise (according to eq. 3.42).
- '*TTL*' ('*trip too long*'): *true* if the instants of the boarding and alighting from respectively the first and last rides stored in *rua* are separated more than ***mtr*** (eq. 3.51), *false* otherwise.
- '*LRDD*' ('*last ride of the day destination*'): the location that is used as goal for the last ride of a day, established as specified in page 60.
- '*LRDA*' ('*last ride of the day alighting*'): the event where the last ride of the day ends, found following the rules from page 60.
- '*LBGIT*' ('*last boarding group of the incomplete trip*'): the latest boarding group of an incompletely defined ride page 60.

Data structures and flows are defined aiming to minimize processing time: besides the outer loop over all AFC entries, ordered by SC and instant, no indexing is needed, using the different tables to keep data that needs to be processed together in one place.

3.5.5 Multiple passengers per tap-in consideration

To deal with users that tap-in a single SC to allow several passengers to board a bus, a 'public transport trip', as it is detected by the trip chaining model, consists in as many 'individual trips' (*it*) as the minimum value of the *passengers* column (table 3.11) for all the rides it encompasses; while excess validations *ev* are taken into account for the expansion of the OD matrices.

$$\begin{aligned} \mathrm{it}_{c,d,i} &: \text{number of travellers of trip } \mathrm{tr}_{c,d,i} \text{ (eq. 3.40)} \quad && \mathrm{it}_{c,d,i} \in \mathbb{N} - \{0\} \\ \mathrm{ev}_{c,d,i} &: \text{number of tap-ins from the rides of } \mathrm{tr}_{c,d,i} && \mathrm{ev}_{c,d,i} \in \mathbb{N} - \{0\} \\ &\quad \text{that are not part of the overall trip} \end{aligned} \tag{3.54}$$

$$\forall \mathrm{tr}_{c,d,i} \;,\;\; k = \max j \mid \exists \mathrm{ri}_{c,d,i,j} : \quad \begin{cases} \mathrm{it}_{c,d,i} = \min\left(\mathrm{pa}_{c,d,i,j}\right) \\ \mathrm{ev}_{c,d,i} = \sum\limits_{\alpha=1}^{k} \mathrm{pa}_{c,d,i,\alpha} - k \cdot \mathrm{it}_{c,d,i} \end{cases} \tag{3.55}$$

This formulation means assuming that, for each trip, *it* passengers will be the same people, moving between activity locations quite similar spatial and temporally; while *ev* validations won't be matched to a trip nor a ride.

3.5.6 Allow passenger rides to span two vehicle runs

The call where a passenger wishes to alight may be beyond a terminus of the route, and thus be reached while the vehicle is carrying out a different run than the one the user boarded. If it is not required to tap-in again at the terminus, the alighting stop may part of one of these two rides. The methodology considers this situation through two actions:

- Before the trip chaining model is applied, each run is evaluated, searching for another that:
  - Has the same vehicle id (those runs where the id changes, as explained in section 3.3.4, have both), and is following the same route (even if its configuration changes).
  - Departs from the same terminus where the former ends, after no longer than $\boldsymbol{mrs}$.

    $$\boldsymbol{mrs} : \text{max. time gap between subsequent rides} \quad \text{time} \tag{3.56}$$

  If a run meets these conditions, it is labeled as 'subsequent' of the one under analysis.

- While the trip chaining model is searching for the alighting of a ride, it considers the remaining calls of the run that corresponds to the boarding event, plus some of those from the 'subsequent' one. To avoid analyzing illogical rides (staying on a bus for much longer than necessary, due to deciding to board the wrong direction of a 'linear' route, or the one of two complementary circular routes that goes in the opposite of the needed direction), only the first 'nssr' first stops from the 'subsequent' run are allowed, based on the parameter $\boldsymbol{mcr}$:

$$\begin{aligned} \boldsymbol{mcr}: &\ \text{maximum circuity of a ride} && \boldsymbol{mcr} \in [0..1] \\ \mathrm{tnsr}_i: &\ \text{Total number of stops of route } i && \mathrm{nsrt} \in \mathbb{N} - \{0\} \\ \mathrm{nsfr}: &\ \text{Number of possible alighting stops} && \mathrm{nssr} \in \mathbb{N} \\ &\ \text{from the first ride} && \\ \mathrm{nssr}: &\ \text{Number of stops of the second ride} && \mathrm{nssr} \in \mathbb{N} - \{0\} \\ &\ \text{to consider as possible alighting points} && \end{aligned} \tag{3.57}$$

$$\mathrm{nssr} = \max\left(0, round\left(\mathrm{tnsr}_i \cdot \boldsymbol{mcr} - \mathrm{nsfr}\right)\right) \tag{3.58}$$

This restriction is founded on a principle similar to 3.5.3.4.2, but applied to a single circular path along which the vehicle is supposed to be moving, and its aim is to avoid needless computations in main trip chaining loop, since all alighting candidates filtered out by this criterion should result in rides that would be rejected anyway. As an example, if a route has 10 stops, numbered from 1 to 10, and $\boldsymbol{mcr} = 0.6$, it means that the program will try to consider at least $10 \cdot 0.6 = 6$ possible alighting points:

  - For users that board at stops [1..4], taking into account the calls from the vehicle run they get on is enough.
  - However, if they board at stops [5..9], the appropriate number of visits from the subsequent run will also be considered. For instance, boarding the earlier run at stop 8, would make the script regard as candidate alightings those at stops [9, 10] from that run, and those at stops [2..5] from the subsequent one (if exists).

3.5.7 Trip chaining output

The output of this section is shown in table 3.11. It includes the rides that have been delimited by the trip chaining model (their encompassing trips completely defined by the boarding and the alighting of its earliest and latest rides, respectively), and a list of incomplete trips without an alighting for the last ride. *Run* id is registered for both the boarding and the alighting of the trip, reflecting that the vehicle run may change once mid-ride (quite common in 'circular routes', but it may also happen in 'linear' ones where the paths in both directions do not completely mirror one another).

Table 3.11: Trip chaining output: *rides* and *incomplete_trips*

Column	***Type***	***Description***
gap_too_lng	boolean	These columns are TRUE if the corresponding criterion from section 3.5 to end a trip is met as the ride is being processed.
too_far_away	boolean	
too_circuit	boolean	
trp_too_lng	boolean	
alght_reachbl	boolean	
inc_ride_sol	enum	Shows which method is applied for the last ride of those trips that could only be completed using the first boarding of the same or the next day (page 56, only applicable for the last ride of the day), or using a destination found with DBSCAN (section 3.5.3.2).
run_id_ini	int	Id of the run the vehicle is carrying out when the user gets on the bus (table 3.10a).
run_id_end	int	Idem, when the user leaves the bus.
passengers	int	How many people board the vehicle.
visit_id_ini	int	Id of the call where the bus was boarded (table 3.10b).
visit_id_end	int	Idem, where users leave the bus.
trip	int	Id of the overarching trip that includes this ride.
n_in_trp	int	Ordinal of the trip within its ride.
card	int	SC id.
bg	int	*Boarding group* linked to this ride (table 3.3).

3.6 Aggregate trip analysis

Initial OD matrices are created aggregating the trips inferred by the chip chaining model for different periods of the year, according to the observed mobility patterns in the city.

These matrices are expanded utilizing a linear factor that considers as origins of extra trips the AFC events that could not be translated to fully defined trips. Three extra sources of individual trips are considered:

- Those trips $\mathrm{tr}'_{c,d,i}$ from the trip chaining model where only the origin $\mathrm{or}'_{c,d,i}$ could be identified.
- 'Excess validations' (as defined in eq. 3.54).
- Cash payments.

The effect of these extra individual trips result in an expansion of the OD matrix cells for the row corresponding to the known origin, as shown in eq. (3.60)

$$\begin{aligned}
\mathrm{od}'_{a,b} &: \text{Individual trips between stops a and b from} & \mathrm{od}'_{a,b} \in \mathbb{N} \\
& \text{successfully defined public transport trips} & \\
\mathrm{od}_{a,b} &: \text{Expanded individual trips between stops a and b} & \mathrm{od}_{a,b} \in \mathbb{N} \\
\mathrm{et}_{a} &: \text{Total number of extra individual trips with} & \\
& \text{a as origin stop.} &
\end{aligned} \tag{3.59}$$

$$\begin{aligned}
\mathrm{od}'_{a,b} &= \sum_{c,d,i \mid \mathrm{or}_{c,d,i}=a \,\wedge\, \mathrm{de}_{c,d,i}=b} \mathrm{it}_{c,d,i} \\
\mathrm{od}_{a,b} &= round\left(\mathrm{od}'_{a,b} \cdot \left(1 + \frac{\mathrm{et}_a}{\sum_b \left|\mathrm{od}'_{a,b}\right|}\right)\right)
\end{aligned} \tag{3.60}$$

Chapter 4

Case study

4.1 Introduction

This methodology has been applied to the AVL and AFC events, and scheduled run beginnings from the vehicles that, for 1 year, run in Santander, a city on the northern coast of Spain (fig. 4.1).

Since most detailed examples of the vehicle runs definition and boarding groups imputation (sections 3.3 and 3.4) are shown for route 1, it will be described in greater detail; while the exposition of trips inference and the aggregate trip analysis (sections 3.5 and 3.6) refer to the whole dataset.

4.1.1 Santander IPTS dataset

Santander has approximately 460 bus stops, as shown in fig. 4.2. Its public transport offer is articulated through 33 routes:

- 19 'linear' routes, with 2 subroutes each that travel the opposite directions between their 2 termini: 1, 2, 3, 4, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23; and the nightly N1, N2, and N3.
- 3 circle routes, where each of the directions they can be traveled from their single termini can be itself considered a route: 5C1 and 5C2, 6C1 and 6C2, 7C1 and 7C2.
- Other routes, offered less frequently, seasonally, or just during special events: 30, 41, 42, 43, 44, 45, and 50.

These routes can show different configurations. The logic behind the planning decision of which one is active varies in each case: period of the year, day of the week, hour of the day, etc. As an extreme case, consecutive runs of routes 17 and 18 switch back and forth between two different itineraries during all day. Each of

Figure 4.1: Santander city [95] [96]

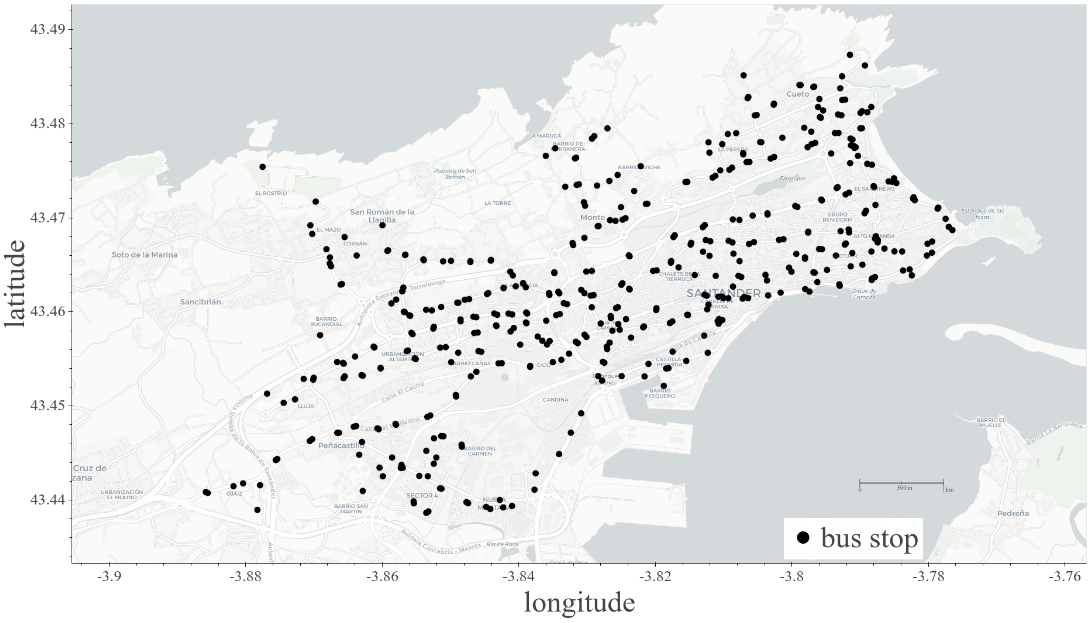


Figure 4.2: Santander city bus stops

these configurations are considered as different route, as defined in section 3.3.1, and thus have their own *template sequences*.

During the year, the AVL subsystem registered 12 400 000 events, distributed as shown in fig. 4.3. It is worth pointing at the densely serviced corridor from approximately mid-city eastward, which turns northwest after reaching the Península de la Magdalena: many of the most important lines of the city travel it, calling at the same stops.

Regarding AFC data, figs. 4.4 and 4.5 show the distributions of the 16 100 000 SC validations and 1 370 000 cash payments, respectively.

While the IPTS is extremely helpful during day-to-day operations, the exploitation of its data must overcome several issues:

- Low AVL and AFC reliability at most route termini (y = true, eq. 3.2), due to how on-board computers are sometimes operated and to the fact that when a bus is empty as it approaches the end of the route, drivers often find more convenient to wait until their next run in a stop upstream from the final one.
- Daily, each run sometimes cannot be reliably identified with an id within the AVL and AFC datasets: this field may show several values within a single run, or the same value may be used for consecutive runs. Also, this id is not

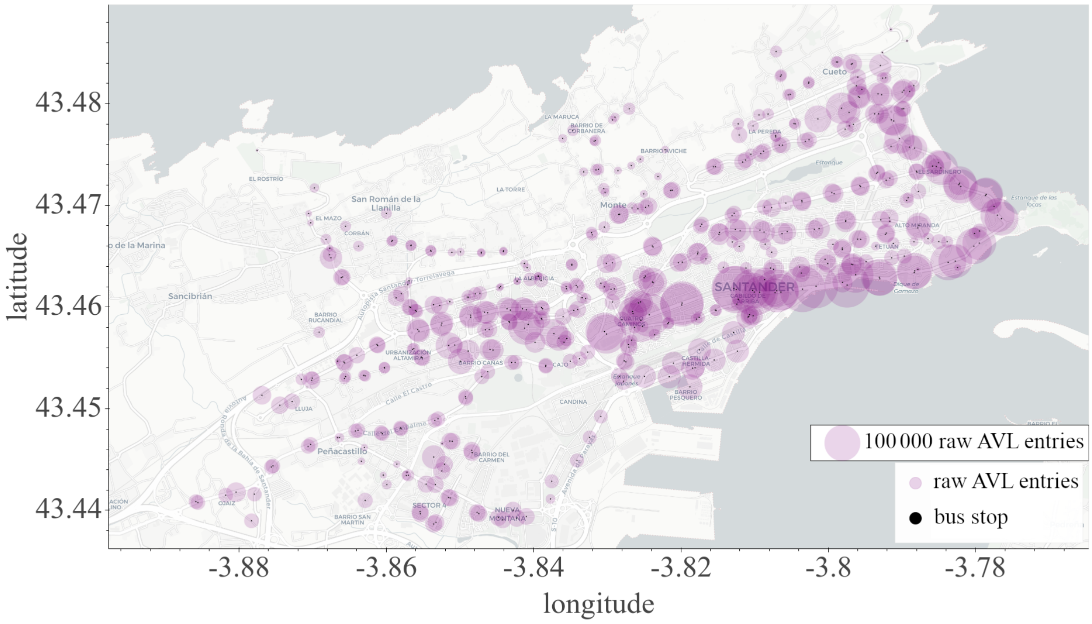


Figure 4.3: Distribution of raw AVL events

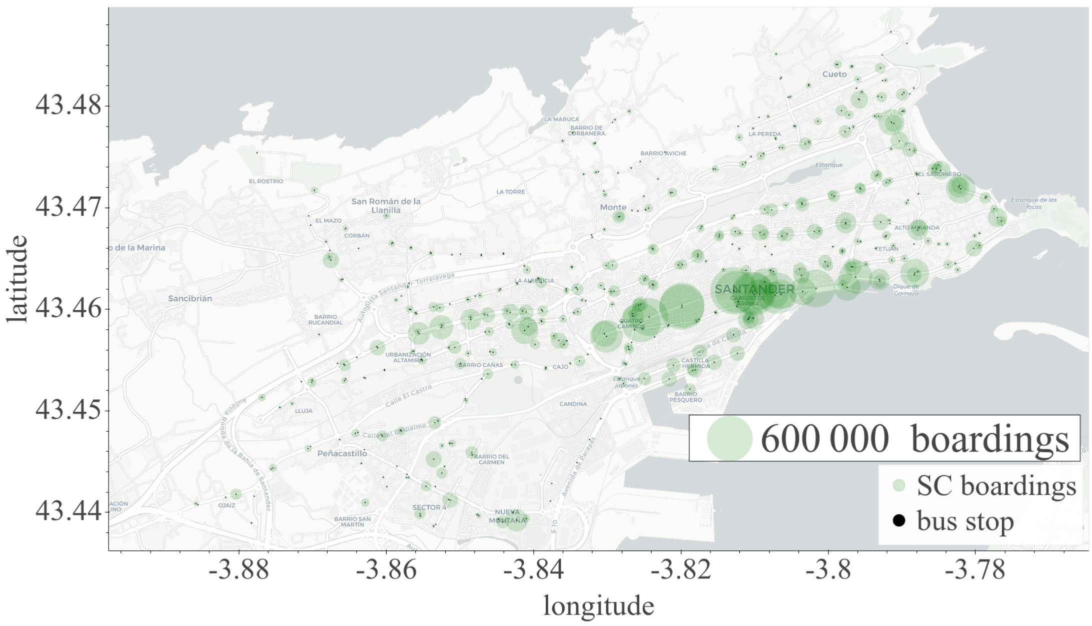


Figure 4.4: Distribution of SC validations

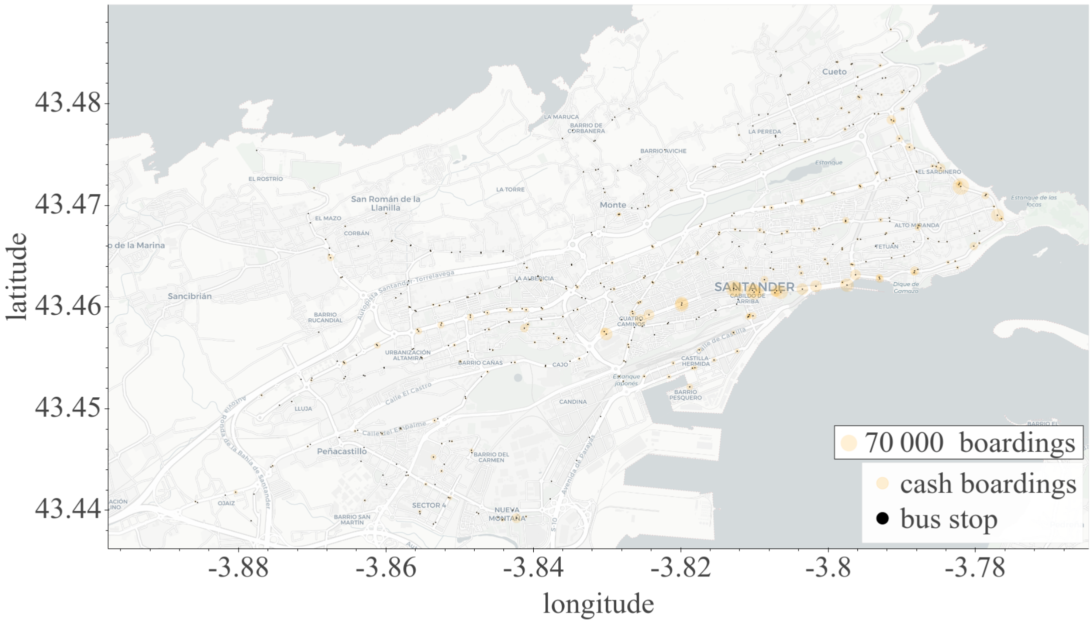


Figure 4.5: Distribution of cash payments

consistent between the AVL, AFC, and planning information.

- Sometimes the AVL or AFC subsystems do not properly reflect which configuration of the current route is being carried out.
- Missing AVL entries.
- Wrong AVL and AFC events that stem from the limitations of the IPTS, such as GPS signal loss, communication failures, or on-board computer errors; or from atypical or incorrect operations (e.g., setting vehicle state parameters that mistakenly identify the task being performed).
- The information regarding whether a planned run finally happened and when did it start is most of the times accurate, but sometimes a normally performed run fails to register, or it does with highly inaccurate timestamps.
- Occasionally, the id of a vehicle changes in the middle of a run, presenting 2 different values.

4.1.2 Route 1

Route 1 operates from approximately 07:00 to 23:00, with headways of at most $s = 20\,\text{min}$ (eq. 3.2). In approximately half of the occasions, the scheduling subsystem records, with a deviation of around $z = 20\,\text{s}$, the arrival and departure of the vehicle from the first stop of the run. A complete roundtrip requires at least $d = 1\,\text{h}$, while a single route leg, even in the most unfavorable circumstances, should not take more than $e = 15\,\text{min}$.

4.2 Implementation of the methodology

This implementation utilizes the procedural language PL/pgSQL within a PostgreSQL 13.2 database for its core tasks; and Python 3.8 and Bokeh 2.2 to show an interactive representation of the results. As previously mentioned, the explanation is initially focused on the case of line 1; while from section 4.2.5 on, it broadens to the whole dataset.

4.2.1 Input data

4.2.1.1 Bus stops and subroutes

The location of the 75 ones that shape route 1, which is divided in two subroutes with one intermediate stop ('Consuelo Berges 16') and both termini in common, is shown in fig. 4.6. These subroutes provide the templates which are used to break down the *stop sequences* found during the treatment of the AVL data in 4.2.3.1.

This itinerary begins at the Pctcan science park in the west, and traverses the city eastward through main arteries, passing by many of its commercial, residential, touristic, and administrative centers until it reaches La Magdalena Park (one of its foremost leisure locations). Then, it turns north-westward, and follows the coastline, providing access to Santander's most popular beaches. Finally, it ends in Valdenoja, a dormitory suburb with some limited commercial use.

During non-business days the activity at Pctcan greatly diminishes, so buses do not visit the 3 easternmost stops. This fact is reflected on the dataset as a different configuration for route 1, which in turn is reflected by this methodology with two other *template sequences* to build vehicle runs. Also, especially during working days, several planned but not announced reinforcement runs begin downstream the first stop, to make use of short free slots drivers have between other assignments.

4.2.1.2 AFC

The dataset includes 2 586 600 raw AFC events for this route. Almost all (99.99 %) correspond to real stops within the city, while the rest have *ids* that do not refer to a physical stop.

4.2.1.3 AVL

There are correspondingly 1 569 417 raw AVL events. All represent calls at real stops of the city.

4.2.1.4 Scheduling information

While the daily timetable that travelers consider when planning their runs on route 1 specifies, depending on whether it is a business day or not, around 100 or

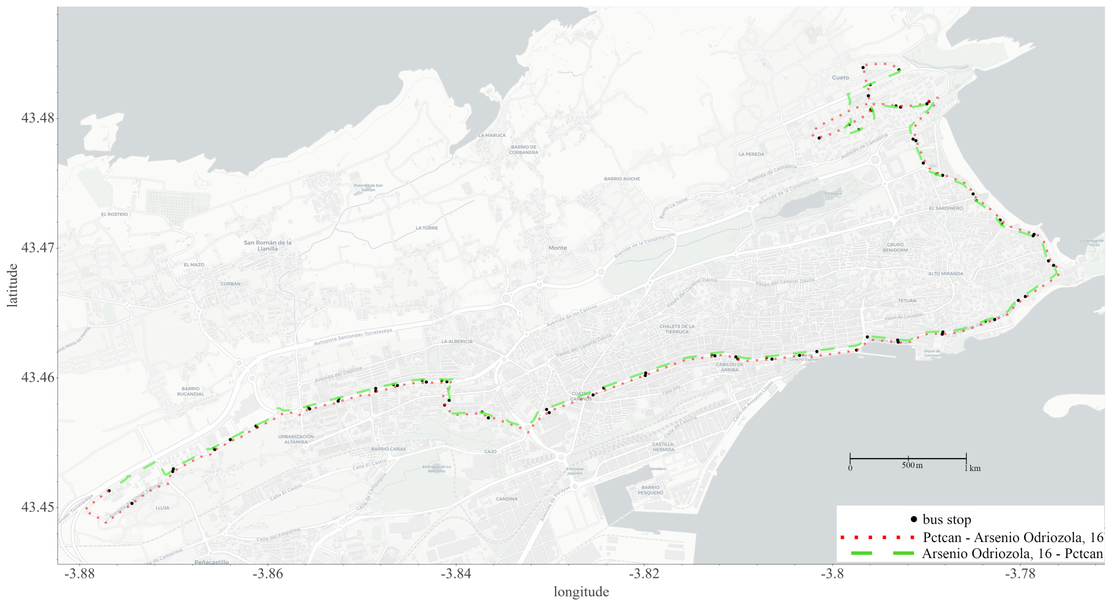


Figure 4.6: Bus stops of route 1

80 places and times where a run begins, the transport authority plans some extra actual vehicle runs, offering less-known additional runs of the route, such as several starting at Valdecilla hospital for staff that just ended their shifts, or reinforcing the offer during known peak demand periods when the distribution of available resources allows to do so. In approximately 95 % of occasions a detected run start time was logged. Extra vehicle runs not present in the scheduling information may occur due to tactical decisions during day-to-day operations.

4.2.2 Preprocessing

4.2.2.1 AFC

Following the methodology outlined in section 3.2.1, 719 971 *stop groups* have been found. Using a value of 20 min for the parameter s, the maximum headway for this route, leads to splitting them in 724 550 *boarding groups* (0.6 % more events). Of these, 108 (0.01 %) last more than $\boldsymbol{s}$ and are not considered. There is, on average, 1 *boarding group* per 4 passenger boardings. Moreover, they provide a first fallback estimation of arrival and departure times at the stops, which is utilized if no AVL records are available.

4.2.2.2 AVL

As explained in section 3.2.2.3, consecutive AVL events that represent the same visit to a stop are merged, leaving 1 532 299 entries (2 % less). Of these, 78 520 (5 %) are deemed unreliable because they are part of impossibly short travel legs. The remaining 1 453 779 entries, gathered in the table avl_coalesced, are classified in 45 840 trajectories.

4.2.3 Vehicle runs definition

4.2.3.1 Analyze AVL trajectories as sequences

The 45 840 trajectories present 5800 different sequences of stops. The two most frequent ones match the already known itineraries of the subroutes under study (fig. 4.6), accounting for around 30 % of the trajectories. Others contain in most cases one or several fragments compatible with one of the subroutes (as described in table 3.8d), though sometimes (2 % of trajectories) the state of a vehicle did not change between subroutes, so a single trajectory contains information regarding more than one run.

4.2.3.2 Specify travel times and dwell times distribution models

Due to its computational advantages, two families of Normal distributions (eq. 4.1) have been chosen to model leg travel and stop dwell times. Considering the mobility cycles of the city, each of these families provides a different function for each subroute, stop, type of day (working, Saturdays, or Sundays and holidays), period of year (summer or not), and time bin (with a span of 30 min, and approximately 16 daily hours of service, there are 32 possible time buckets: 07:00 to 07:30, 07:30 to 08:30, and so on).

$$
\begin{aligned}
p_{a,\tau,\gamma,\delta,\zeta,\eta} &: \text{run leg travel time} && t \in T; \\
& && t \sim \mathcal{N}\left((\mu_p)_{a,\tau,\gamma,\delta,\zeta}\,, \left((\sigma_p)_{a,\tau,\gamma,\delta,\zeta} \right)^2 \right) \\
u_{a,\tau,\gamma,\delta,\zeta,\eta} &: \text{dwell time} && u \in T; \\
& && u \sim \mathcal{N}\left((\mu_u)_{a,\tau,\gamma,\delta,\zeta}\,, \left((\sigma_u)_{a,\tau,\gamma,\delta,\zeta} \right)^2 \right) \\
a &: \text{route id} && \text{From the methodology (eq. 3.4)} \\
\tau &: \text{stop number} && \text{From the methodology (eq. 3.3.1.2).} \\
& && \qquad \text{For run legs, their first stop.} \\
\gamma &: \text{period of year} && \gamma \in \{\text{'summer'}, \text{'rest of the year'}\} \\
\delta &: \text{type of day} && \delta \in \{\text{'working'}, \text{'saturday'}, \\
& && \qquad \text{'sunday or holiday'}\} \\
\zeta &: \text{time of day bin} && \zeta \in \{1 \dots \eta\} \\
\eta &: \text{time bins in a day} && \eta \in \mathbb{N}
\end{aligned}
\tag{4.1}
$$

Route 1's leg travel times and dwell times have been characterized at each stop by roughly $2\,\text{periods} \cdot 3\,\frac{\text{day types}}{\text{period}} \cdot 32\,\frac{\text{distributions}}{\text{day type}} = 192$ distributions each. Their means and standard deviations have been calculated utilizing the pertinent entries from table *avl_coalesced*.

4.2.3.3 Assemble vehicle runs

After applying the process described in section 3.3.3, setting its parameters to to $\boldsymbol{g} = 0.998$ and $\boldsymbol{c} = 2$ stops, 42 319 possible runs were found.

4.2.3.4 Merge instances where a vehicle changed its id mid-run

This refinement leads to the detection of around 2 daily occurrences of this issue, reducing the number of candidate runs to 41 641.

4.2.3.5 Ascribe vehicle runs to scheduled services and update visit time spans

40 352 runs have been mapped to a scheduled run beginning (111 utilizing a vehicle different from the planned one); while the other 1289 were not. 86 % of logged run start times were utilized to characterize the first call of their runs.

4.2.3.6 Select vehicle runs backed by enough information

After considering the results from sections 4.2.3.5 and 4.2.4, the following acceptance criteria have been chosen (utilizing the nomenclature from eq. 3.38):

- For runs mapped to a scheduled beginning ($w =$ True):
  - Always accept if the planned vehicle was utilized ($p =$ True).
  - If a bus other than the scheduled one was used ($p =$ False), require at least 3 boarding groups linked to the run ($h \geq 3$).
- Unscheduled runs are required to offer stronger evidence: at least three ticketing events and no less than 12 total entries (one third of the number of stops of a subroute) endorsing its existence ($h \geq 3 \ \land \ h + f \geq 12$)

Applying these thresholds, the methodology reports on average 120 and 97 daily runs, depending on weather analyzing a business day or not. In the former case, the 96.5 % of runs had previously been planned, and were materialized with the intended vehicle; while 3 % were planned, but executed with a different vehicle; and 0.5 % were unplanned runs. During non-business days, the corresponding ratios are 99.2 %, 0.5 %, and 0.3 %; which is consistent with weekends and holidays being usually less demanding for the public transport of the city, resulting in less deviations from the schedule to react to the evolution of the traffic system.

Of all raw AVL data available, 92 % was finally used to provide information to re-create a call of a run. The source utilized to discern bus calls was AVL, statistical inference, a run beginning logged by the scheduling subsystem, and AFC in 91 %, 7 %, 2 %, and 1 % of occasions.

4.2.4 Boarding groups imputation

Applying the criteria described in section 3.4, using an AFC leeway of $\boldsymbol{o} = 1$ min, provides the following results:

- 94.7 % of all boarding groups have been deemed to be correctly reporting their route and bus stop.
- 5 % have been assigned to another stop than the automatically logged one.
- 0.3 % were linked to a run of a different route.

4.2.5 Trip inference

4.2.5.1 Introduction

Passengers in Santander may access the vehicles validating a SC, or paying with money to the driver. The 'standard' card charges for a single trip roughly half the fare asked to those customers using cash. It may be used to cover the costs for several users, swapping it multiple times, or asking the driver may perform this operation in a single transaction. Moreover, it allows owners to travel for free during one hour after a tap-in, if they keep boarding different routes. This policy encourages users to expedite their activities and board a bus of another route as soon as possible, to enjoy a free ride; which is especially easy to do in the corridor mentioned in page 73.

Other types of SCs allow for unlimited free rides, and are available to people part of certain groups, for personal use only: 'large' or single-parent families, seniors, affected by disabilities (completely free or renewable monthly, depending on their severity), unemployed, or teens (renewable trimonthly by paying a reduced fee). These users are more likely to board a bus for quite short trips, in situations where others who would have to pay will probably walk instead.

The clear benefits of SCs make them the preferred choice in roughly 90 % of rides recorded by the AFC.

The capability of 'standard' SCs to pay for multiple users is reflected in the column '*passengers*' of the table *boarding groups*, summarizing how many people gained access to a bus during a call thanks to a single SC. As described in section 3.5.5, this information is preserved through the trip chains model, linked to the appropriate inferred rides and trips.

Another aspect of the IPTS that has been considered is that users may stay inside the vehicle between runs if the wait before the next one is not too long; and in many cases they need to do so to arrive at their destination. Not only in the case of 'circular' routes; some 'linear' ones have paths that greatly diverge near the termini (e.g., route 13, shown in fig. 7.4, which deviates from the straighter path when traveling towards 'Lluja - Cementerio' to visit Rucandial neighborhood; and on the opposite direction before reaching 'Residencia Mayores de Cueto' to the stop 'Hermanos Tonetti 16'). The trip chaining model has been programmed to allow for this behavior, having identified beforehand which runs are consecutive (even in those cases where the configuration changes): users may board one run of a route, and get off the bus while it is already following another run of the same route. In the case of 'circular' routes this means that the bus called at the single terminus in-between; while for linear routes, it called at one of the two available termini and started going towards the other.

To decrease the computation complexity of identifying frequent activity destinations through DBSCAN (section 3.5.3.2), their geographical location on the map will be approximated to the closest stop which is part of its cluster.

4.2.5.2 Model parameters

After some testing, and taking into consideration the peculiarities of the city, the following parameters were adopted for the trip chaining model:

$$
\begin{array}{lllll}
\boldsymbol{M}{=}2\,\mathrm{km} & \boldsymbol{mdp}{=}400\,\mathrm{m} & \boldsymbol{mtt}{=}40\,\mathrm{min} & \boldsymbol{ws}{=}4.8\,\frac{\mathrm{km}}{\mathrm{h}} & \boldsymbol{mrs}{=}15\,\mathrm{min} \\
\boldsymbol{f_w}{=}1.3 & \boldsymbol{md}{=}400\,\mathrm{m} & \boldsymbol{mtr}{=}1\,\mathrm{h} & \boldsymbol{vcd}{=}160\,\mathrm{m} & \boldsymbol{mcr}{=}0.6 \\
\boldsymbol{\epsilon_d}{=}160\,\mathrm{m} & \boldsymbol{\epsilon_t}{=}15\,\mathrm{min} & \boldsymbol{mtw}{=}30\,\mathrm{min} & \boldsymbol{cr}{=}2.92 &
\end{array}
$$
$$
\boldsymbol{minPts} = \left\{ \begin{array}{lccc} & \text{workdays} & \text{Saturdays} & \text{holidays} \\ \text{school season} & 20 & 4 & 5 \\ \text{summer} & 20 & 4 & 5 \end{array} \right\} \tag{4.2}
$$

Or, to put down in words:

- An alighting is only inferred if the subsequent boarding takes place no more than 2 km away. Maybe the user utilized a different mode, and even if that were not the case, the odds of choosing the wrong stop are great.
- If an inferred alighting takes place more than 400 m away of the next boarding, or if it happens more than 40 min later, it is accepted that an activity has taken place in-between. Otherwise, the user was just transferring from one route to another as part of a single trip.
- Travelers are considered to walk in straight lines, at a maximum speed of $4.8\,\frac{\mathrm{km}}{\mathrm{h}}$.
- Transit users tend to choose to leave vehicles in the stops that leave them the most leeway, considering how long it would take them to walk to the next known boarding. However, they prefer riding a bus to walking, perceiving the same disutility from 1 min and 28 s of the former as from 1 min of the latter.
- When trying the different solutions implemented to find the alighting for the last trip of the day, a minimum distance of 400 m is required between origin and destination.
- After a trip surpasses the 1 h threshold, no more rides are added to it.

- The year is partitioned in the 6 categories that is known influence mobility in Santander (summer and rest of the year in one hand; and workdays, Saturdays, and holidays in the other). Within each, DBSCAN is applied to search for the location and time window of frequent activity destinations. To be neighboring events, two alightings must occur no more than 160 m apart, within a time gap of 15 min. Then number of neighbors an alighting needs to be a core point is different and computed separately for each partition, to reflect how users may change their behavior in different occasions. For instance, routinely going to work on workdays, and visiting family or friends on Saturdays, or going to the beach in summer.

- The time window during which a traveler should arrive to a frequent activity found through DBSCAN for it to be accepted as the destination of the trip under analysis will be no narrower than 30 min.

- If the sum of the distances between boarding and alighting stops of the rides of a possible trip is greater than 2.92 times the direct distance between its origin and destination (equivalent to $\alpha = 40°$ in fig. 3.14), it is judged too circuitous and is split according to 3.5.3.4.2.

- Passengers can stay on the vehicle as part of the same ride between two runs of the same route, when the second one begins at the same terminus that the first ends, and no more than 15 min later. Calls from the latter run are added to the list of possible alightings (which also includes those from the former that happen after the boarding event) until there are at least as many candidates as 60 % of the total number of stops of the route.

4.2.5.3 Overall results

15 524 405 SC boarding events have been analyzed. 75 % successfully: alighting stop of the ride, and overarching trip. In 1 % of occasions, at least one ride of the trip could be defined, but the overall destination could not. For the other 24 % percent, the methodology could not find the alighting stop of the ride (nor the destination of the trip).

68 337 frequent activity nodes have been found, under the parameters of eq. (4.2). Table 4.1 shows their distribution among the different types of days that have been considered, while fig. 4.7 shows the relative temporal density of the availability of frequent activities in the city. For instance, on workdays, in school season, there is a morning peak at 08:12, with a frequency slightly greater than 10 %: that is the portion of all frequent activity destinations that were detected during that type of day which are available at that time. Workdays present two main routine activity peaks at early morning and early afternoon, being quite sharper in school season, probably due to all the students that regularly go to their educational centers and back. Routine activities during Saturdays and holidays tend to happen more

Table 4.1: Distribution of activity nodes though the year

	workdays	***Saturdays***	***holidays***
school season	34 037	14 507	9080
summer	5503	2925	2285

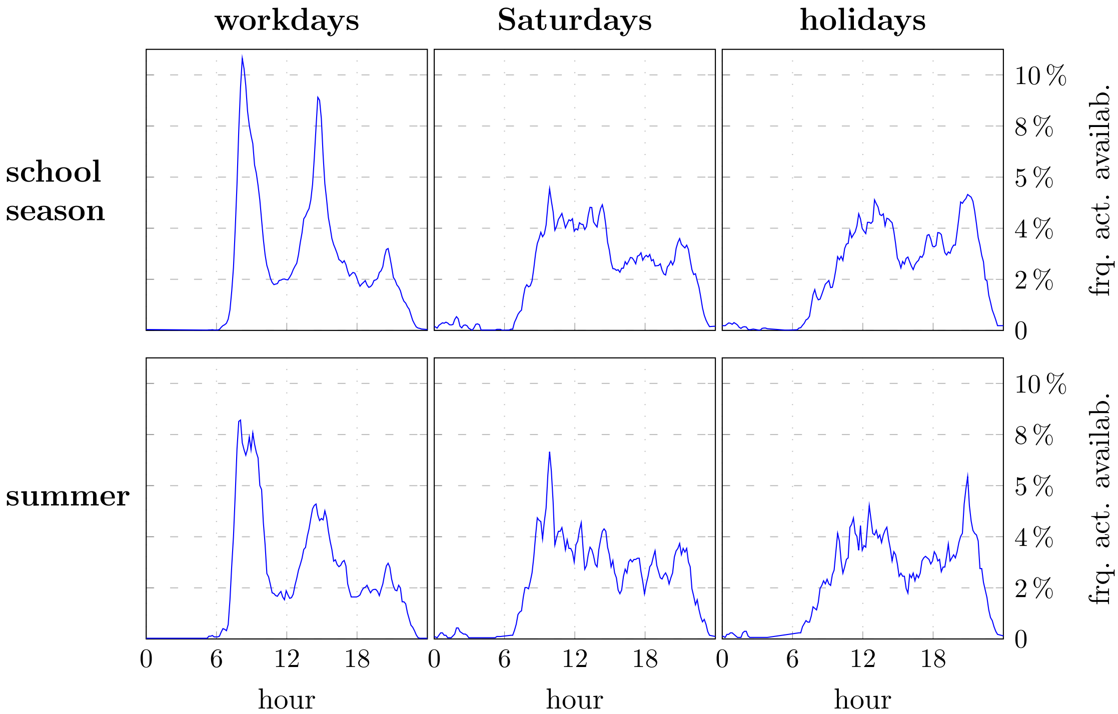


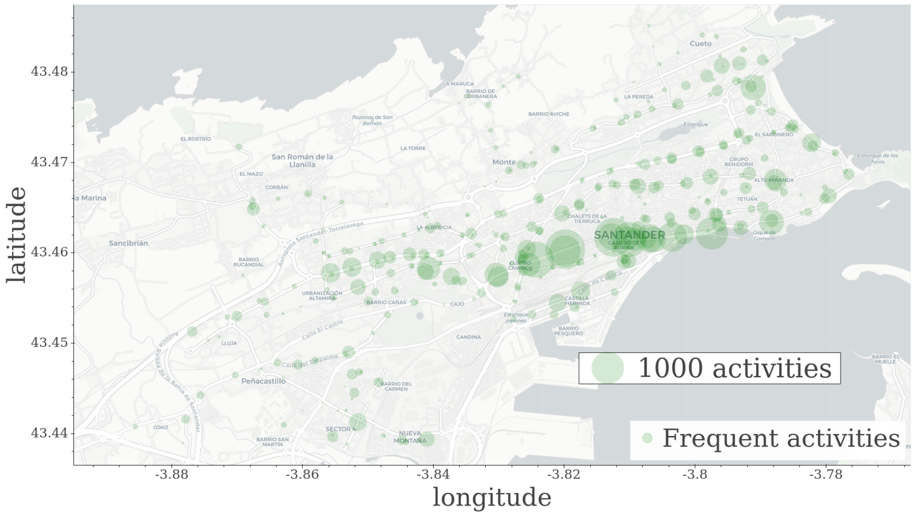
Figure 4.7: Frequent activities availability ratio along each type of day

uniformly, and later.

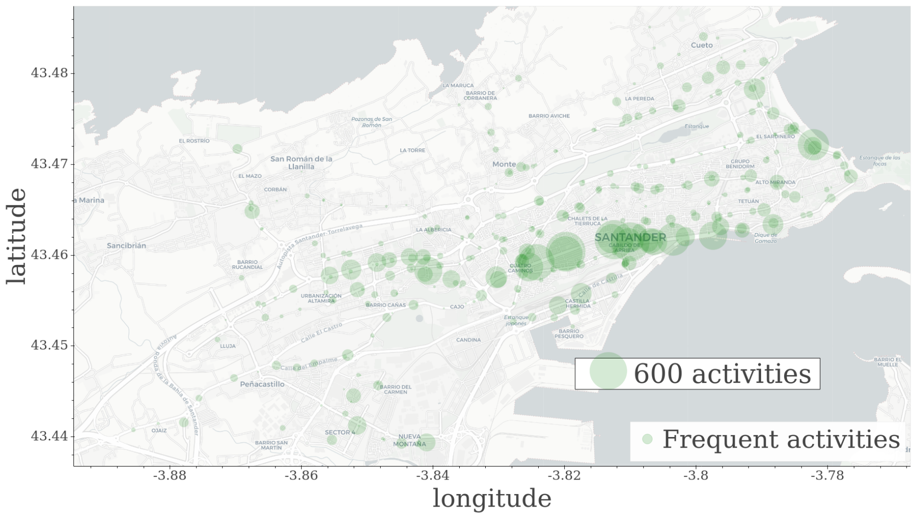
In turn, figs. 4.8 and 4.9 display the spatial distribution of frequent activities on a working day in winter, and a summer holiday. They show how citizens gravitate towards the coast on their spare time.

Table 4.2 shows the frequency with which each trip ending or splitting criteria were met while processing the last ride of each trip (They may happen simultaneously, and thus they do not add up to 100 %). It is worth noting that many trips are the last of their days, which makes the three approaches applicable for this situation (page 56, section 3.5.3.2) even more important. The distribution of which one provided an estimation of the final alighting stop of the last trip of each day (or, of the final alighting stop of trips that were unrelated to the next boarding of the same day, as explained in page 55) is shown in table 4.3.

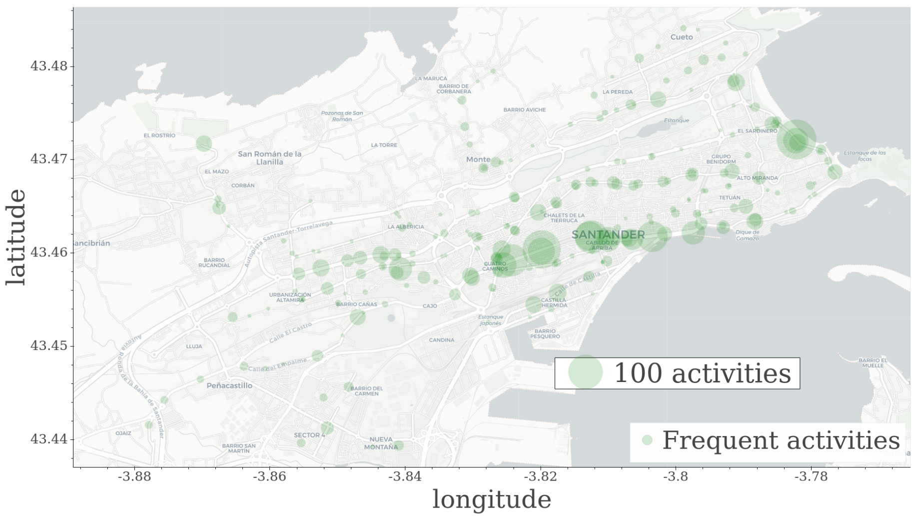
Figure 4.8: Distribution of frequent activities on different day types in winter



(a) Working day

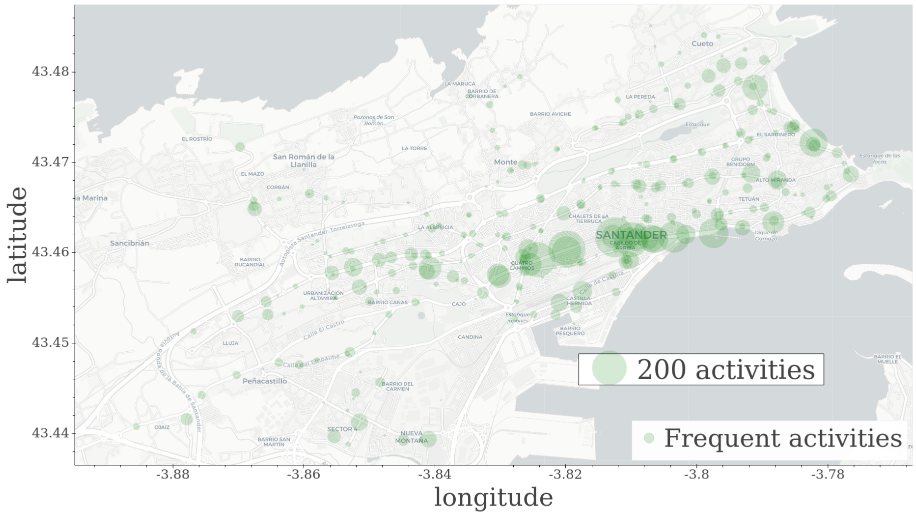


(b) Saturday

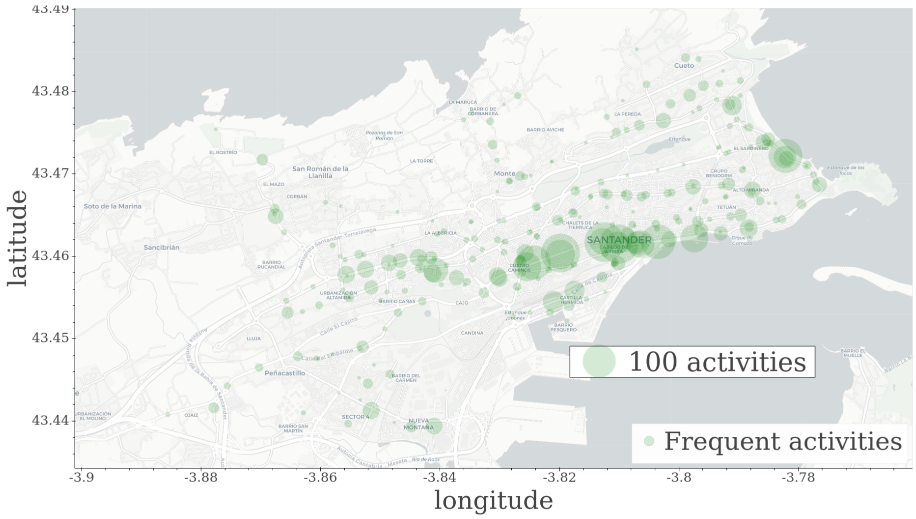


(c) Holiday

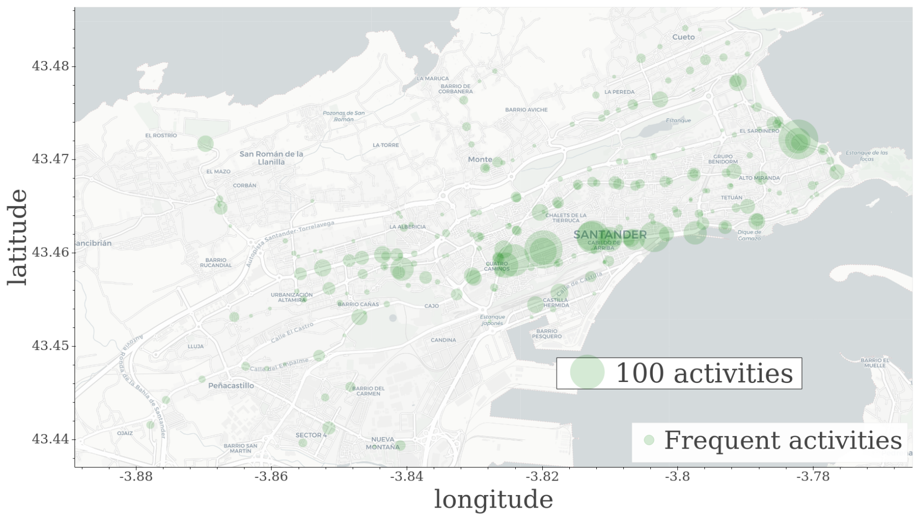
Figure 4.9: Distribution of frequent activities on different day types in summer



(a) Working day



(b) Saturday



(c) Holiday

Table 4.2: Trip ending or splitting criteria occurrences

Criterion	***Frequency***
gap too long	48 %
stops to far away	8 %
too circuitous	1 %
trip too long	46 %
alighting reachable sooner from former ride	7 %
last ride of the day	45 %

Table 4.3: Destination estimation method utilized for last or incomplete trips

Method	***Frequency***
first stop of the day	82 %
first stop of the next day	15 %
destination from DBSCAN	3 %

4.2.6 Trip aggregation

10 996 165 trips have been defined by the trip chaining model. However, considering that in some of these inferred trips multiple users got on the bus (as explained in section 3.5.5), the number of individual trips is 11 869 841, showing table 4.4 their distribution.

An expansion factor is needed to consider the rest of the AFC events that are not part of the inferred trips. The great economic benefit of using a SC supports the assumption that only sporadic, non-transferring users use cash. In turn, incomplete trips provided by the trip chaining model still manage to identify a likely trip origin.

Thus, as explained in eq. (3.60), an expansion factor is calculated for each cell,

Table 4.4: Transformation from inferred trips to individual trips

Simultaneous users	***Trips inferred***	***Individual trips***
1	10 247 155	10 247 155
2	658 901	1 317 802
3	69 434	208 302
4	14 811	59 244
5	3413	17 065
6	1251	7506
	...	
45	1	45
Total		11 869 841

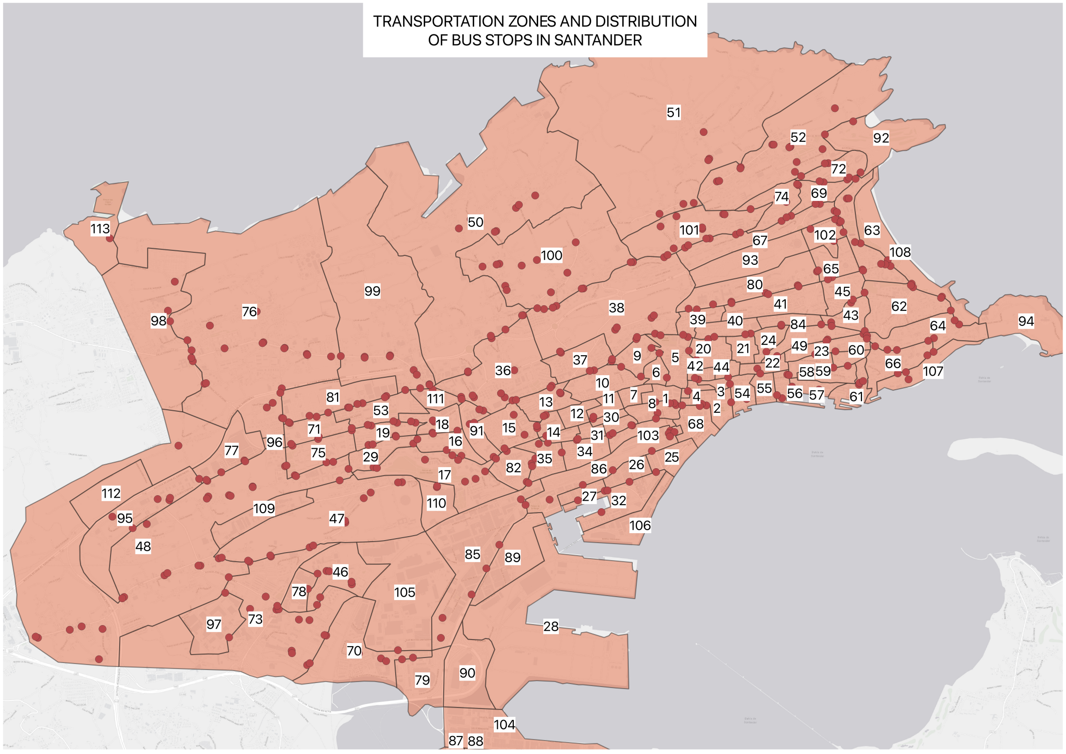


Figure 4.10: Location of the 113 transportation zones in Santander

to add these extra individual trips of known origin but unknown destination. They fall under 3 categories, which add up to a total of 5 844 323 individual trips:

- 4 197 756 individual trips, from isolated alightings or partially defined trips (some intermediate transfer was found, but not the final alighting).
- 1 633 763 cash AFC events.
- 12 804 from excess validations (eq. 3.54) from trips completely defined by the trip chaining model.

This process ends with 6 457x457 OD matrices, which are in turn aggregated to the 113 transportation zones usually defined in studies in Santander City, which are shown in fig. 4.10 These matrices have been included in the first appendix.

Table 4.5 shows the different periods of time of the year from the point of view of mobility patters, and how many individual trips happen on average in each. As can be seen, summer is the peak season in Santander from the point of view of public transport use. It is noticeable the increase of roughly one third of individual trips on Saturdays and holidays from winter to summer. They probably reflect that both tourists and locals take advantage of the usually good weather.

Table 4.5: Mean daily individual trips across the different periods of the year

	workdays	***Saturdays***	***holidays***	*weighted average*
school season	55 566	29 956	20 272	45 443
summer	58 746	38 804	32 648	51 821
weighted average	56 396	31 902	32 270	47 051

Chapter 5

Discussion

5.1 Vehicle runs definition

This section gathers several examples to illustrate how this methodology has successfully improved the characterization of runs that were registered in the IPTS in a way that impeded their consideration.

5.1.1 Reconstruction of a vehicle run from fragmented and erroneous information

Figure 5.1 shows the case chosen for this analysis. The temporal horizontal axis has been broken in three regions with a shift between them for easier visualization:

- The central one, where the actual run detected by the methodology and the planned departure (5) are depicted. Its temporal axis has been placed in the lower part of the plot.
- The leftmost area, with its temporal axis located in the upper part of the figure. It includes the relevant raw AVL and AFC data, with a −20 min shift:
  - 4 AVL sequences:
    - 1: From 'Arsenio Odriozola 16' to 'San Fernando 66,' with a gap of almost 1 h between 'Plaza de Italia' and 'Luis Martínez'.
    - 2: From 'San Martín' to 'Pctcan,' overlapping with 1 along its first 9 stops, and missing data at 'Avenida de Valdecilla' and 'Torres Quevedo 22.'
    - 3: A single event, at 'Plaza de Italia'.
    - 4: A single event, at 'Pctcan', the last stop of the run. It happens around 30 s before 2 ends.

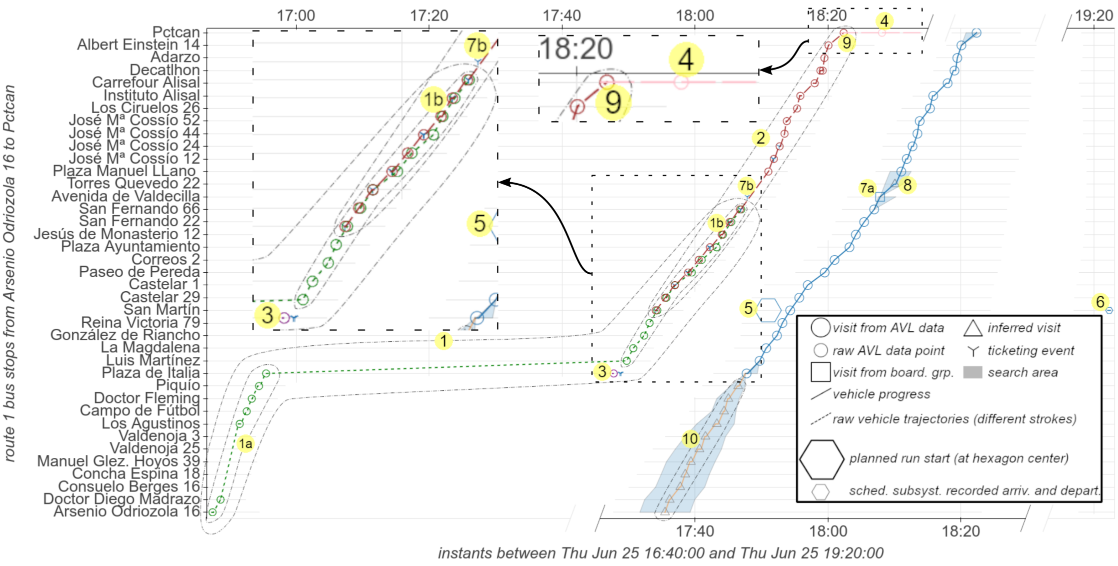


Figure 5.1: Characterization of a vehicle run from fragmented and erroneous information

- 19 AFC events, occurring between 'Plaza de Italia' and 'José Mª Cossío 24.'

- The rightmost zone only contains the clearly unrelated arrival and departure times logged by the planning subsystem (6), with a −40 min shift.

The run has been defined making use of the available information. The first part of sequence 1 was considered as 2 different fragments, discarding the earlier (1a, which was probably caused by an incorrect vehicle state) and utilizing the latter (1b). After the last entry from 1, the call at 'Avenida Valdecilla' (7a) is approximated from a ticketing event (7b); and the one at 'Torres Quevedo 22' (8) is inferred considering departure and arrival times from the previous and next stop, respectively. Of the two possible arrivals at the final terminus (9), the one from sequence 4, which happens 30 s earlier, is more likely according to the departure time from 'Albert Einstein 14' and the travel time distribution between these stops during the time period [17:30-18:00] on a workday.

It is worth noting that even though the run was planned to start at 'San Martín,' the methodology has detected that it actually began a few stops upstream (at 'Plaza de Italia,' from run 3). The search for previous events (10) did not return any match, so that is the stop where the run began.

5.1.2 Vehicle id mid-run change

Figure 5.2 shows how the information regarding a run of the subroute from 'Pctcan' to 'Arsenio Odriozola' appears in the IPTS, and its characterization by

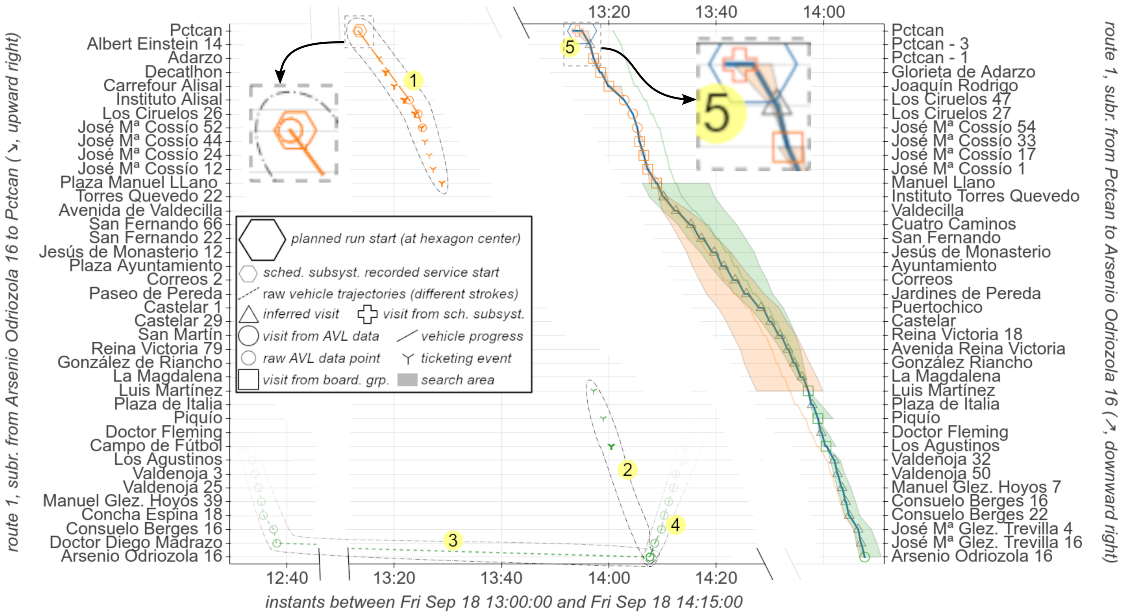


Figure 5.2: Characterization of a run when its vehicle id changes while it happens

this methodology. Again, the horizontal temporal axis has been divided in three zones:

- The rightmost area, which contains, with the temporal axis on top, the two runs initially found, how they have been combined, and the planned start linked to them.
- The middle and leftmost regions show, with shifts of -40 min and -20 min and their temporal axes at the bottom, the pertinent raw records.

Initially, step 3.3.3 had found two runs:

- One for vehicle 14 (orange), backed by a 4-stops trajectory, and several ticketing events (1), being the latest one at 'Manuel Llano.'
- Another for vehicle 224 (green), inferred from 4 ticketing events at 3 stops (2 , the earliest at 'Luis Martínez'), and any of the two raw AVL events with the same timestamp at 'Arsenio Odriozola 16' terminus, which are part of opposite trajectories which end (3) or begin (4) there.

These have been detected, as described in section 3.3.4, to be part of a single run (displayed with a thicker blue line). Its corresponding scheduling subsystem entry (5) only detected the departure of the vehicle, a bit later than the available AVL data at that stop. Since it falls within the feasibility range from 'Pctcan - 1,'

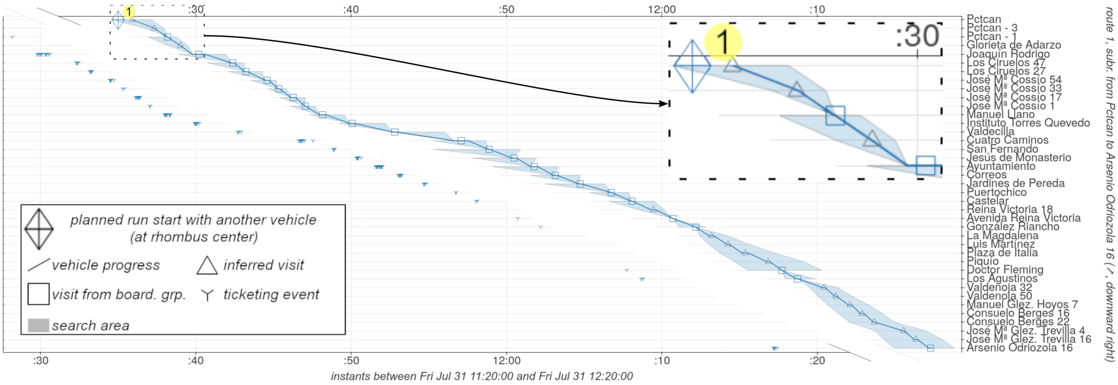


Figure 5.3: Vehicle run characterization from AFC data only. Actual vehicle not the planned one

it is accepted and used to update the departure time at 'Pctcan,' and to improve the inferred call at the intermediate stop 'Pctcan - 3.'

5.1.3 No AVL data and wrong vehicle id

Figure 5.3 shows a case that illustrates two situations that happen in the case study: the AVL subsystem not recording any entry, and a vehicle different from the planned one carrying out the run.

There is a shift of 10 min between where the run and the scheduled departure are drawn (rightmost part, temporal axis on top), and where the raw AFC data can be found (on the left, temporal axis at the bottom). It can be seen (1) that, since the scheduling subsystem did not register the beginning of the trip, the calls and 'Pctcan' and 'Pctcan - 3' had to be inferred using the arrival at 'Pctcan - 1' as the fix.

5.1.4 Handling of alternating route configurations

Several routes of Santander change their configuration through the day, to better address the evolution of the demand; or to cover a wider area of the city, alternatively visiting one zone or the other, at the expense of longer frequency in those areas. The runs of a bus before and after a configuration change are particularly prone to be mislabeled or unregistered altogether: the vehicle location and ticketing systems may improperly be set for a configuration of the route, while it is in fact following another. This all may be compounded by the rest of issues already discussed in page 73.

For instance, route 3 connects with Peñacastillo municipality (a residential area on the west border, roughly 6 km away from the center of Santander). Its 'main' configuration joins the main transportation corridor, described at page 73, and ends at ‘Paseo de Pereda 35’ (around 2 thirds along its west-east arm). However, to better satisfy travel demand, when classes start or end, to go to and from the educational centers alongside the Los Castros Avenue, one of the alternative configurations of route 3 extends the ‘main’ one, moving north through the Tetuán Tunnel, and then traverses Los Castros westwards, ending at The University of Cantabria Rectorate.

Figure 5.4 shows an example of this situation, for one bus on route 3, on a Wednesday in school season. On three occasions (morning, around midday and evening) the route changes accordingly to when most students begin or end their daily classes. The movement of the bus can be coherently followed, occasionally switching between figs. 5.4a and 5.4b at termini. Focusing on the third period when the vehicle follows the ‘main’ configuration, from approximately 16:00 to 20:00, it can be appreciated that the AVL did not properly register the position of the vehicle, but the methodology has successfully identified which configuration of route 3 was materialized, and inferred each call of each run combining AFC, scheduled service beginnings, and the pertinent distributions of travel and dwell times for route 3, on a workday in school season.

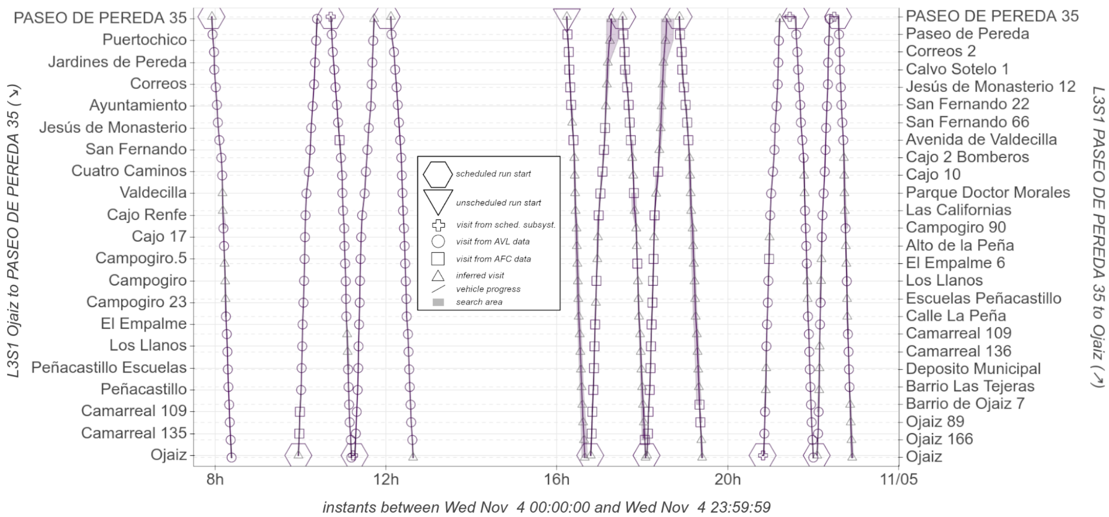
A similar situation in route 4 (fig. 5.5), that connects a neighborhood in the southwest of Santander, with a mix of residential use, traditional fishing, and restaurants. A significant number of calls of several runs had to be approximated from AFC or the travel and dwell times distributions.

5.2 Treatment of initial termini

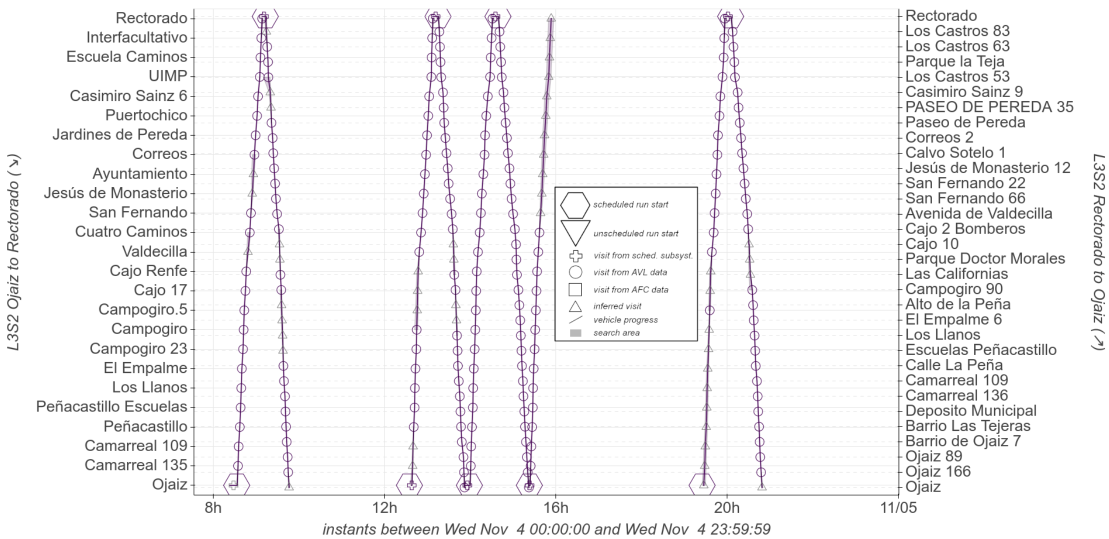
The objective of this section is to study the benefit of how this methodology handles the data available at especially problematic termini, as it happens in this route. To this end, the 25 466 runs which present recorded starting times from the planning subsystem that, as described in section 3.3.5, have been accepted for their characterization, are used as the ground truth to be compared with the results obtained in three scenarios where that information is not considered:

- A Follow the default methodology behavior for a route when the scheduling subsystem did not record the start of a run (page 40).
- B If the data at the first stop is deemed feasible, utilize it in the same way as any other stop.
- C if the planned start of a run falls within its corresponding feasibility range (already stored in the *search_ranges* table, or computed utilizing the closest downstream data-supported call of the run), it is used as departure, if it happens later than any available AFC or AVL entries. This means assuming that schedule adherence is high enough to trust the planned departure times, unless they are impossible or very unlikely.

Figure 5.4: Runs of one vehicle covering Route 3 during a workday in school season

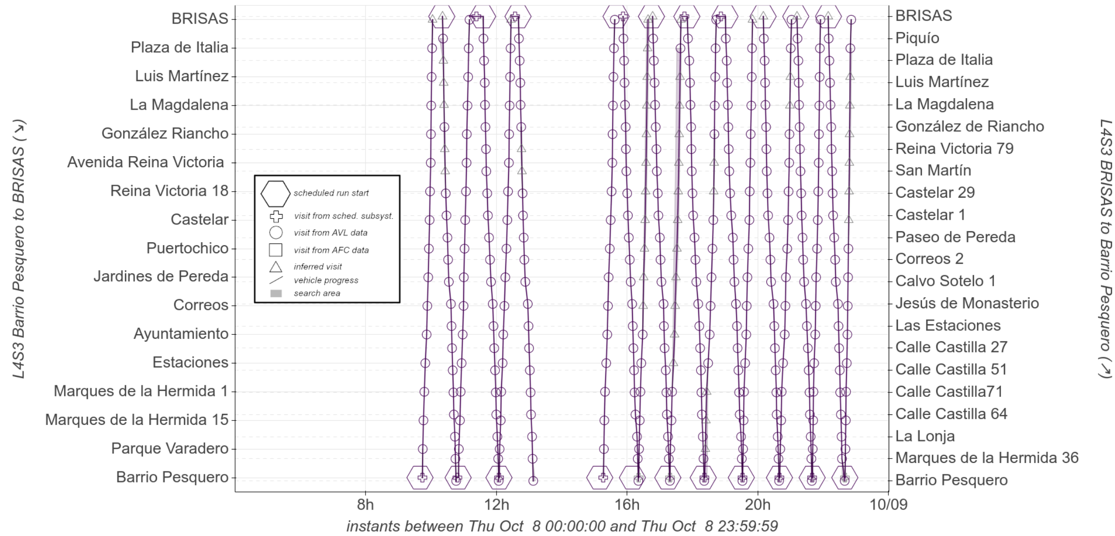


(a) 'Main' shorter configuration

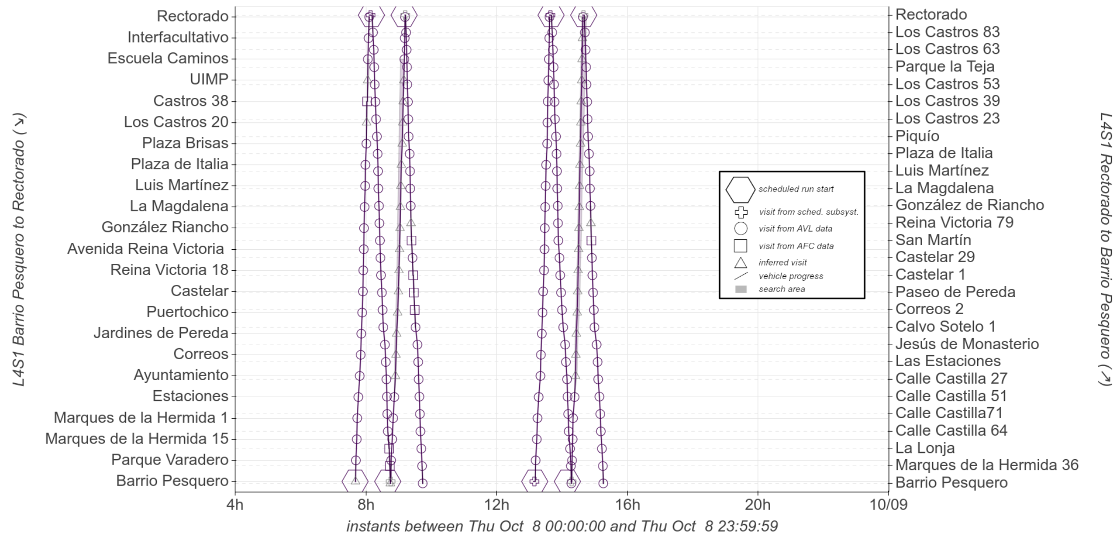


(b) Alternative configuration, adding extra stops to the educational hub in Los Castros

Figure 5.5: Runs of one vehicle covering Route 4 during a workday in school season



(a) 'Main' shorter configuration



(b) Alternative configuration, adding extra stops to the educational hub in Los Castros

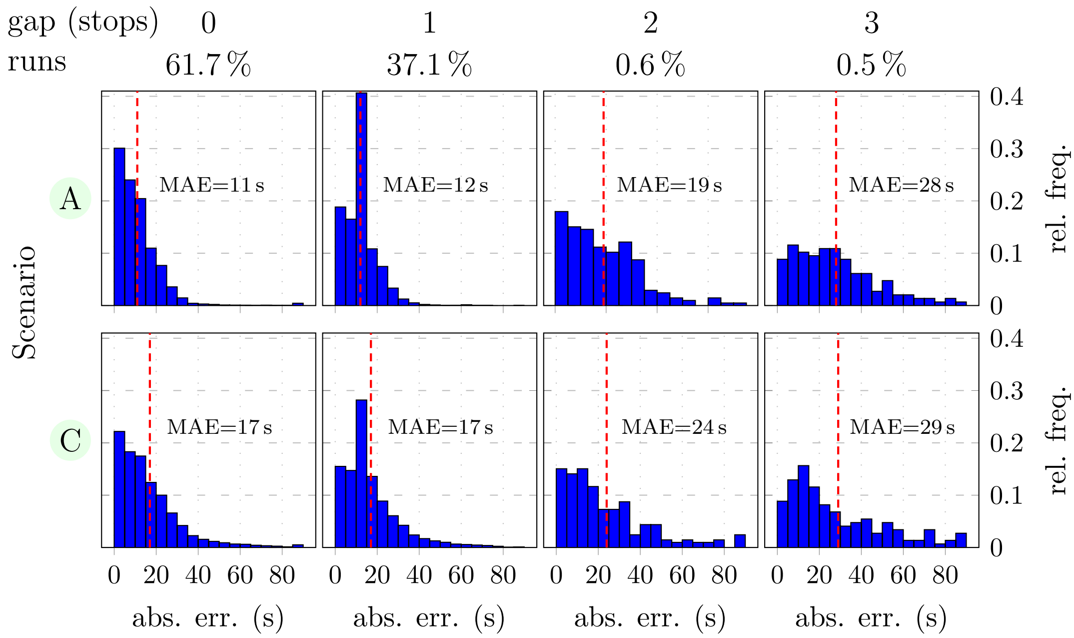


Figure 5.6: Distributions of abs. departure time errors per gap for scenarios A and C

Figure 5.6 shows the distributions of the absolute error of the run departure time reported in scenarios A and C. Runs have been classified according to their "gap": how far away (measured in route legs) are their earlier visits based on AVL or AFC data from their scheduled beginnings. As can be seen, the decision of relying on the inferred start time rather than the planned one provides approximations with less dispersion (standard deviations of 13 s and 17 s, respectively) and a smaller mean absolute error (MAE), though as the uncertainty increases (more unknown calls between the start of the run and the first data point) this benefit lessens.

Scenarios A and B only differ for those runs where compatible AVL or AFC data at the scheduled first stop can be found (zero gap). Figure 5.7 shows their distributions of absolute errors in this case. Again, scenario A infers the missing data with less dispersion (std. devs. of 15 s and 17 s, respectively) and MAE (11 s versus 13 s).

5.3 Robustness against missing and wrong data

This section analyses how the methodology is affected by missing and erroneous AVL information and run start detection (ticketing events are fully available in all scenarios). The 16 863 runs where all calls were fully recorded by the scheduling and AVL subsystems (49 % of all) is used as the ground truth; and compared with

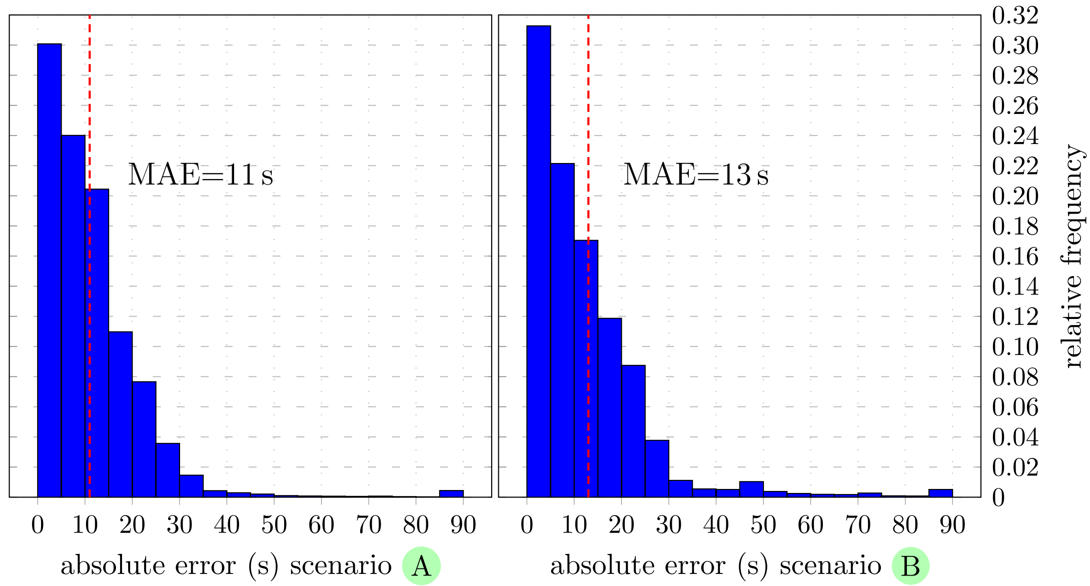


Figure 5.7:)
Distr. of abs. departure time errors when there is no gap for scenarios A and B

the results of running this methodology utilizing only part of the recorded raw AVL data and scheduling subsystem detections, chosen through Bernoulli sampling; also adding different amounts of synthetic AVL erroneous readings, which have been randomly generated following these rules:

- *bus_stop*, *vehicle*, and *group* are chosen between all their distinct values.
- *instant* happens between 07:00 and 23:00 of any day.
- Sampling from the distribution of *durations* is simulated utilizing its percentiles and the Uniform Distribution.

In fig. 5.8, the percentages are relative to the raw AVL entries and planned runs available in the dataset. For instance, a scenario with 25 % of real data and 100 % of simulated errors only reads the arrival and departure of the vehicles at the initial stop recorded by the scheduling subsystem in 25 % of the scheduled runs; while its raw AVL input is created combining a Bernoulli sample of the real information with a probability 25 % and 4 times as many bogus entries.

As more real data are available in a scenario, the more accurately runs are characterized. For instance, with a relatively small sample (25 %), while the 99$^{\text{th}}$ percentile does not significantly differ from not using AVL or detected run starts

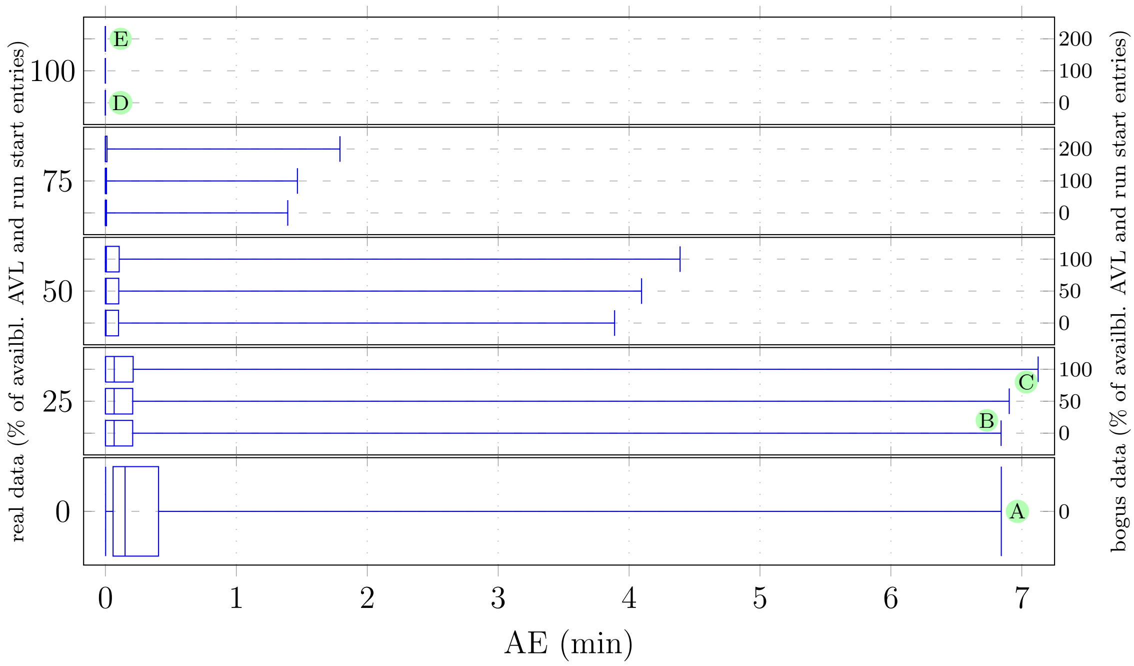


Figure 5.8: Distribution of the deviation from the ground truth for different proportions of real and wrong data. Extremes at 1st and 99th percentiles

at all (slightly less than 7 min), it can already be appreciated that absolute error (AE) is quite more likely to be smaller: lower quartile, median, and upper quartile reduced from 4 s, 9 s, and 24 s to 0 s, 4 s, and 13 s, respectively (A & B).

It is also noticeable the strength of the methodology against artificial incorrect entries, which grows as more true readings are available in the scenario. Two examples are:

- With just 25 % of real information, the effect of adding four times as many wrong entries is a relatively small increase of the 99th percentile, from 6m51s to 7m07s (B & C).
- If all real data is available, the methodology successfully identifies the correct values as seeds (section 3.3.3), and is able to completely ignore many false events (D & E).

5.4 Trip chaining examples

This section will show several examples from the *rides* table that contains all trips found by the methodology (table 3.11).

5.4.1 Example 1

Figure 5.9 shows the 4 trips carried out by one user during a working day in summer, in two pictures:

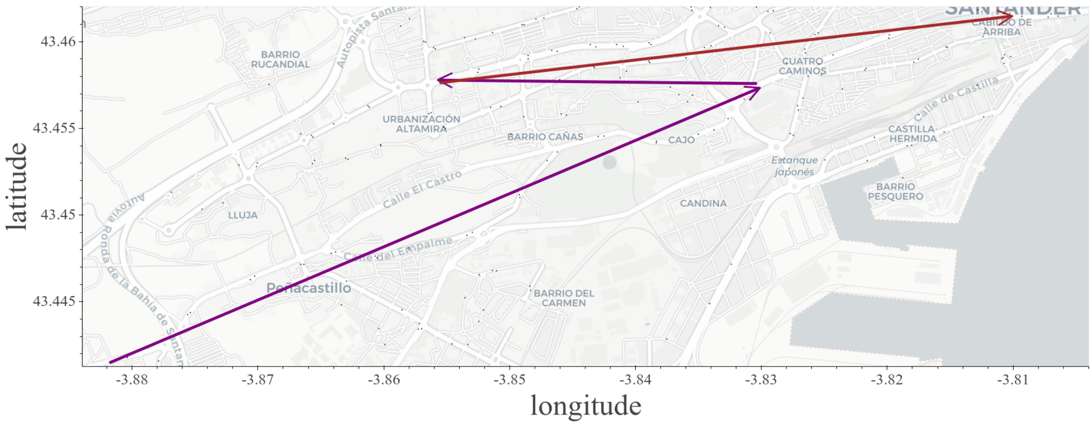
- In fig. 5.9a the 2 first trips can be seen:
  - The first trip (purple), composed by 2 rides:
    - On route 3, from 'Barrio de Ojaiz 7' (08:11:28) to 'Valdecilla' (08:25:46).
    - The traveler crossed the street, to board a bus of route 1, from 'Avenida de Valdecilla 7' (08:33:59) to 'Instituto Alisal' (08:41:27).
  - The second trip (brown), from 'Los Ciruelos 47' (09:10:44) to 'Ayuntamiento' (09:27:55), on route 1 again. Even though less than $\boldsymbol{mtt} =$ 40 min pass from the very close alighting stop, if the user had wanted to go to 'Ayuntamiento' he could have stayed on the previous bus, and thus the program correctly identifies this as a new trip
- Figure 5.9b shows the other 2 trips:
  - The third trip (pink), from 'Plaza Ayuntamiento' (09:40:05) to 'San Fernando 66' (09:44:14), on route 7C1. Again, according to the time gap from the previous alighting this could have been a transfer, but if the user destination had been 'San Fernando 66' all along, he would have alighted from the second trip at the very close stop of 'San Fernando'.
  - Finally, the traveler performs the last trip of the day (green) to probably return home from 'San Fernando 66' (10:24:37) to 'Barrio de Ojaiz' (10:41:02).

5.4.2 Example 2

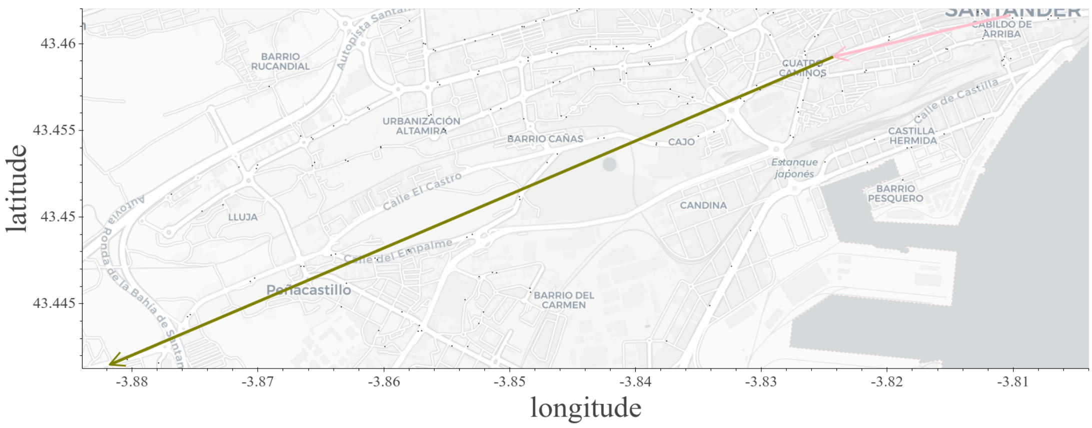
In this case from a working day in school period, depicted in fig. 5.10, the chain of trips is as follows:

- The first trip (red), composed by 2 rides:
  - On route 12, from 'Francisco Tomas y Valiente 11' (12:28:11) to 'Cuatro Caminos' (12:37:57).
  - The traveler crossed the street, to board a bus of route 2, from 'San Fernando 66' (12:56:51) to 'Instituto Alisal' (13:06:26).
- The second trip (purple) occurs after a 1-hour activity. It also is composed by 2 rides:
  - On route 1, from 'Los Ciruelos 47' (13:57:32) to 'Jesús de Monasterio' (14:10:59).

Figure 5.9: Trip chaining example 1



(a) First two trips



(b) Last two trips

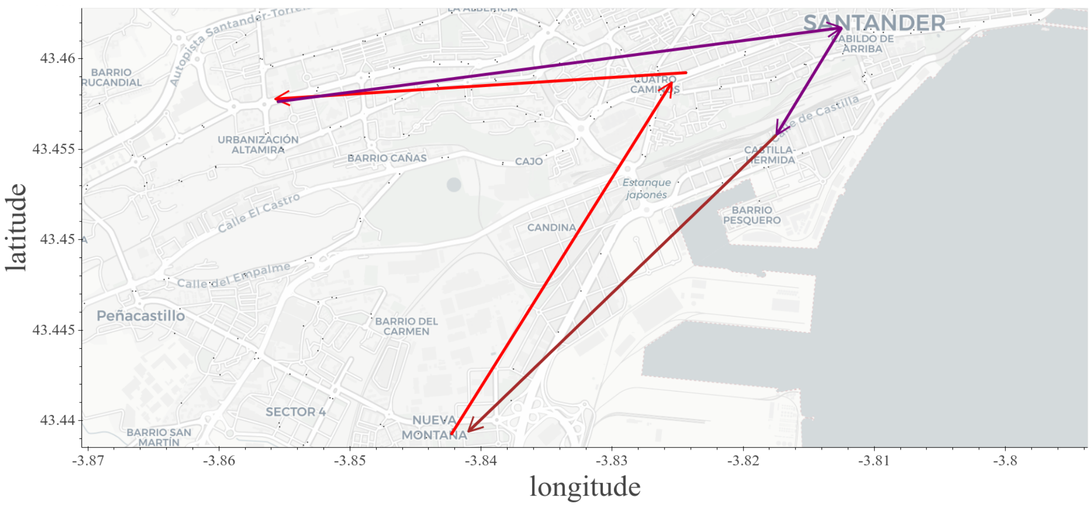


Figure 5.10: Trip chaining example 2

  - On route 4, from ‘Jesús de Monasterio’ (14:17:41) to ‘Calle Castilla 51’ (14:23:59).

- The last trip of the day (brown), to probably return home from ‘Calle Castilla 51’ (14:35:28) to ‘Francisco Tomas y Valiente 7’ (14:40:52), riding route 19. Even though the rest of the criteria allowed for this last ride to be part of the previous trip, the total path would be too circuitous (more than 3 times the direct distance between ‘Los Ciruelos 47’ and ‘Francisco Tomas y Valiente 11’), and thus a break was introduced just before this ride, where more leeway for some short activity is available (11 min and 29 s).

5.4.3 Example 3

Figure 5.11 illustrates a chain of trips for a working day in school period:

- It starts with a trip on route 17 (green) from ‘San Fernando 66 (10:53:25) to ‘Corbán’ (11:06:43).

- The second trip (red) (after roughly 4 and a half hours) goes from ‘Corbán’ (15:43:11) to ‘B^{o} La Sierra’ (15:45:18).

- The next ride begins too far away and too long after the previous ends to be part of the same trip, so it starts a new one (purple), which was carried out by two passengers, from ‘Joaquín Rodrigo’ (16:54:11) to ‘Puertochico’ (17:16:01) on route 1.

- The next and last ride of the day, for the same reasons as before a trip on its own (brown), goes from ‘General Dávila 58’ (22:22:09) to ‘Camilo Alonso

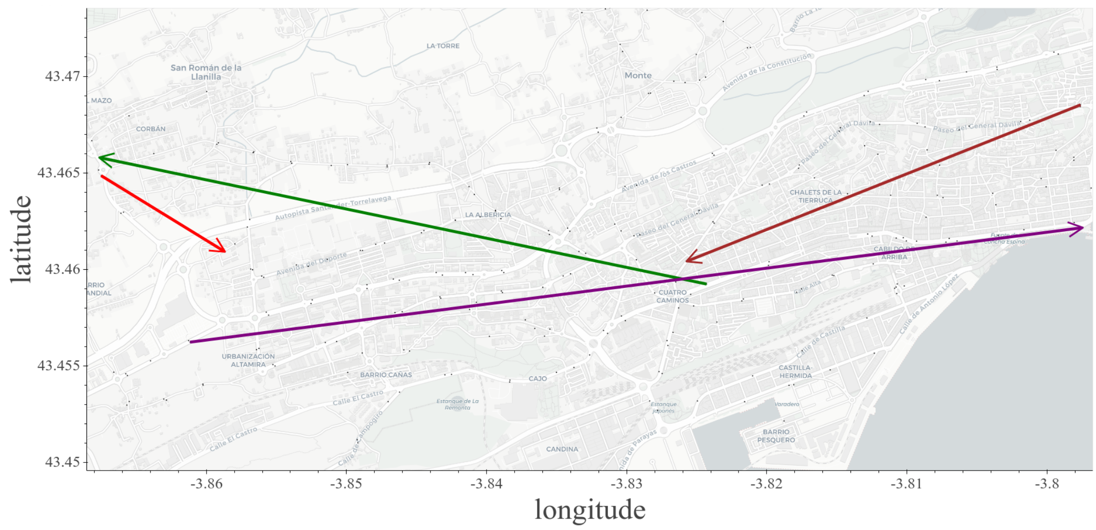


Figure 5.11: Trip chaining example 3

Vega 19’ (22:29:36).

5.4.4 Example 4

This case from a summer working day is an example of multiple users sharing the same SC. On each occasion, three people boarded the vehicle:

- The first trip (brown) goes from ‘Camarreal 135’ (17:25:40) to ‘Jesús de Monasterio’ (17:42:51), on route 3.
- Roughly 4 and a half hours later, they return in another run of the same route (blue), from ‘Jesús de Monasterio 12’ (22:21:04) to ‘Camarreal 136’

5.4.5 Example 5

This trip chain happens between a Saturday and a Sunday in summer, and is captured thanks to the decision to specify that, from the point of view of the mobility in the city, the transition between one day and the next happens at 06:00. Thus, for later aggregation of individual trips in OD matrices, these would be considered as happening during a summer Saturday:

- One ride on route 1 (green) from ‘Correos’ (17:11:56) to ‘Luis Martínez’ (17:20:05), a zone with several popular spots for young people to go out.
- 9 h later, come back home (red) on route N1, from ‘Luis Martínez’ (02:17:52) to ‘Correos 2’ (02:29:03).

5.4.6 Example 6

Two consecutive Saturday and Sunday summer days are shown in fig. 5.14:

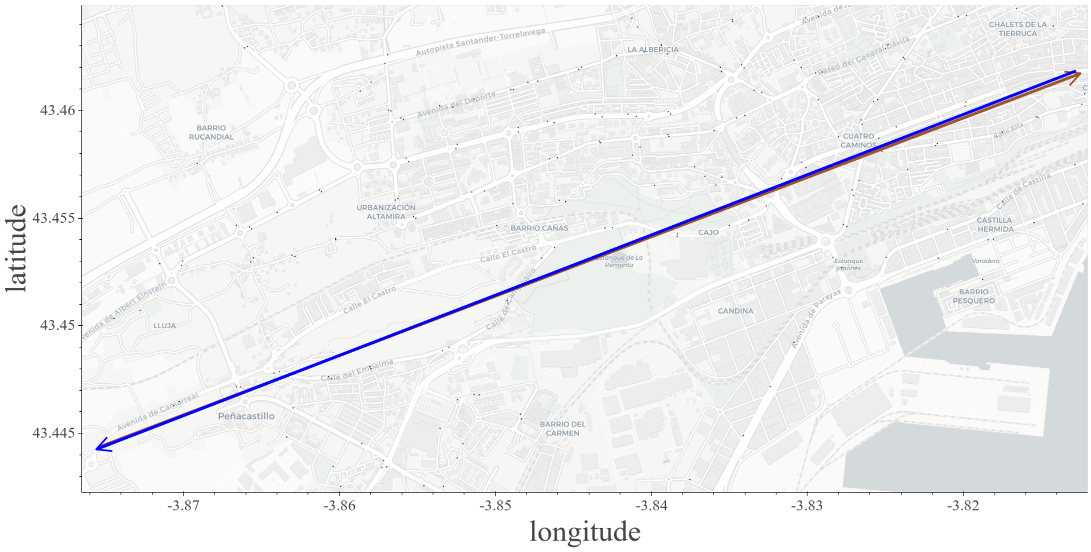


Figure 5.12: Trip chaining example 4

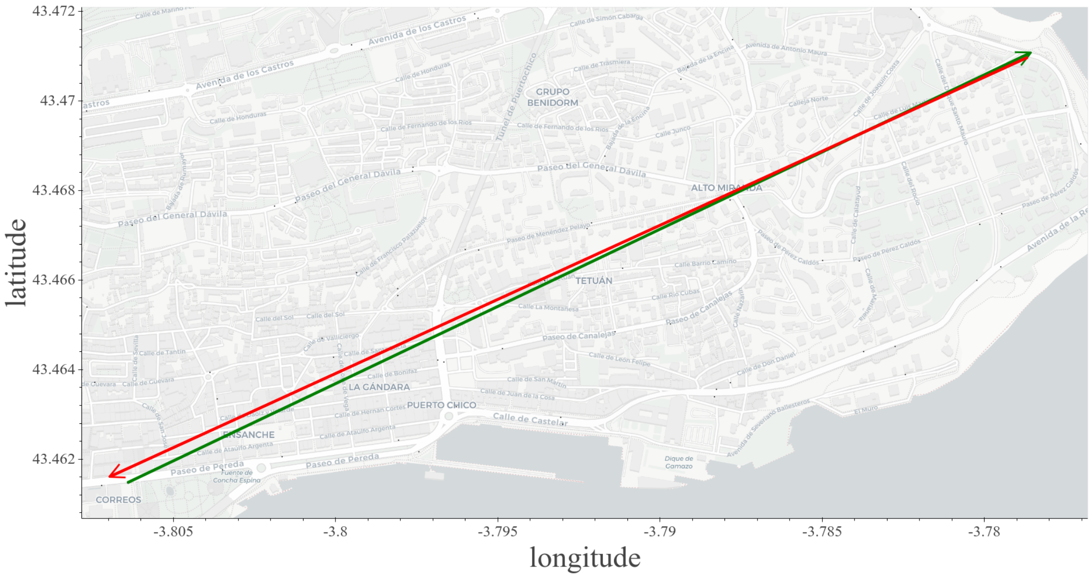


Figure 5.13: Trip chaining example 5

- On Saturday, the user only makes a single-ride trip (red) from 'Catedral' (14:09:51) on route 5C2. To infer the most likely alighting bus stop, since the card is utilized again next day, the earliest of those Sunday validations is assumed to be the most likely destination ('Menendez Pelayo 61'). Thus, the program supposes that this Saturday trip ends at 'Menéndez Pelayo 46' (14:18:23).
- On Sunday, two trips take place:
  - From 'Menendez Pelayo 61' (08:12:57) to 'Camilo Alonso Vega 10' (08:24:04), on route 5C1 (purple)
  - From 'Valdecilla' (10:13:08) to 'Puertochico' (10:26:11), on route 2 (brown). Again, 'Menendez Pelayo 61' is assumed to be the destination, this time because it is where the first boarding event of the day under analysis.

5.4.7 Example 7

It is possible to infer the destinations of 3 trips on a sunday in school season thanks to the detection of a frequent activity around 'San Fernando 66' stop on that day type, where the traveler usually arrives between 13:48:10 and 14:18:10. These 3 trips all start past 13:30 at different stops, being materialized as rides of routes 1 or 2, arriving at 'San Fernando 66' around 14:00. Since they all are the only ride of their respective days and the first boarding of the next day does not exist (the SC was not used) or is too close (eq. 3.47) or too far away (eq 3.41), the 'first boarding of the same day' and 'first boarding of the following day' (section 3.5.2) criteria are not applicable.

Several of the trip chains that have as one of their destinations a node part of this frequent activity cluster, can be seen in Figure 5.15b.

5.5 Desire lines

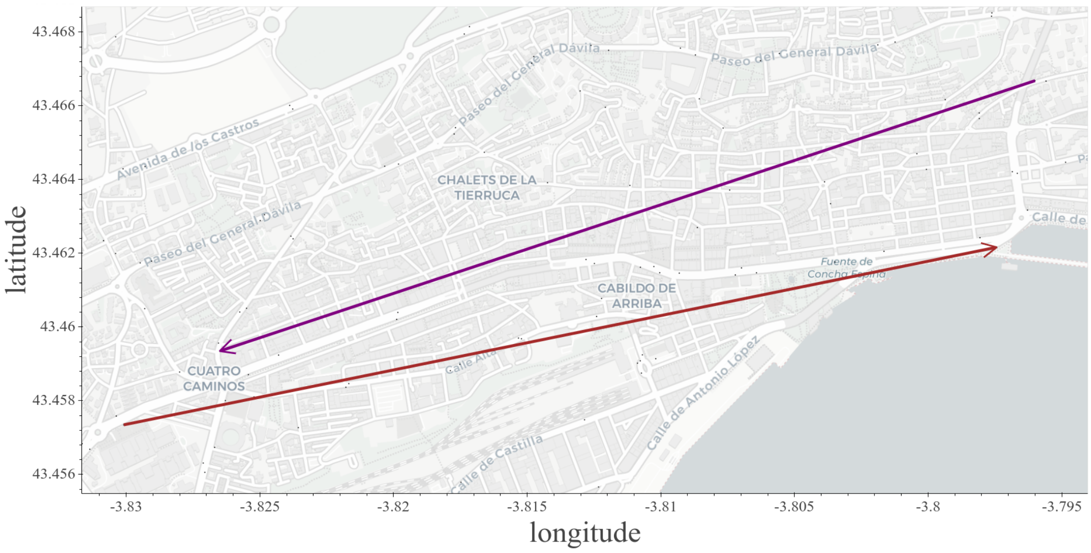
Several cases corresponding to highly demanded destinations for different parts of the year have been built with the information contained in the OD matrices generated in section 4.2.6.

5.5.1 Trips attracted by zone 8 on a working day in school season

Zone 8 is where the city council plaza is, one of the most prominent locations of the commercial and services sector, as well as government buildings (treasury and motor vehicles departments, city council, traffic, etc.). Figure 5.16 shows those OD pairs with it as a destination and more than 30 individual trips. As can be seen, few originate from the east. Maybe those citizens find some kind of activity closer to home, and do not need to travel to zone 8; while those from the west do.

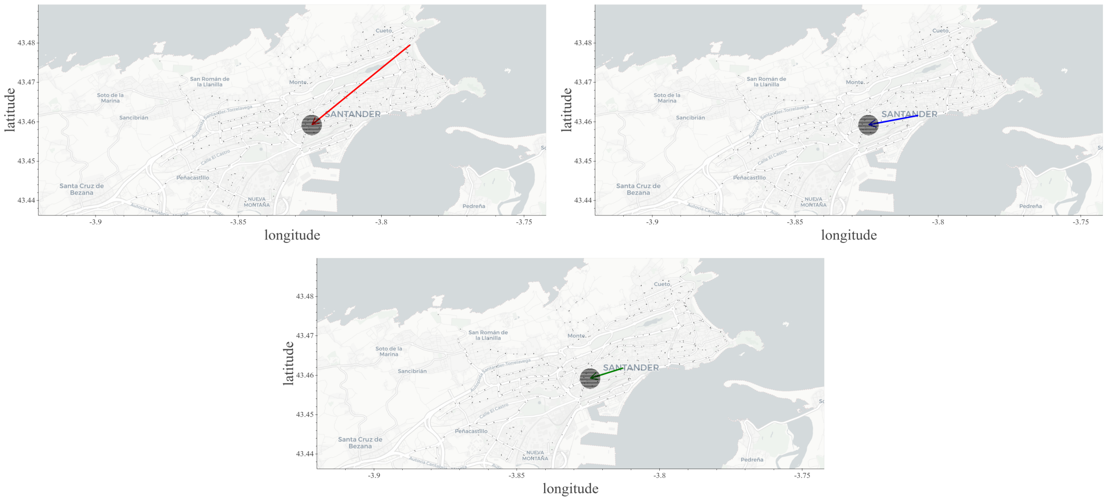
Figure 5.14: Trip chaining example 6

(a) Trip on the former day

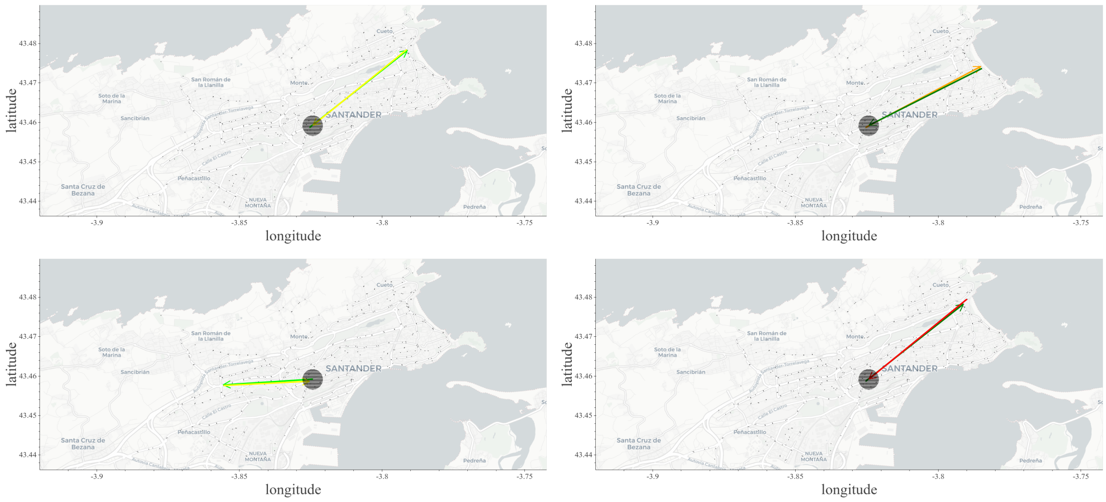


(b) Trips on the latter day

Figure 5.15: Trip chaining example 7



(a) Inferred trips



(b) Days with trips that contributed to the cluster

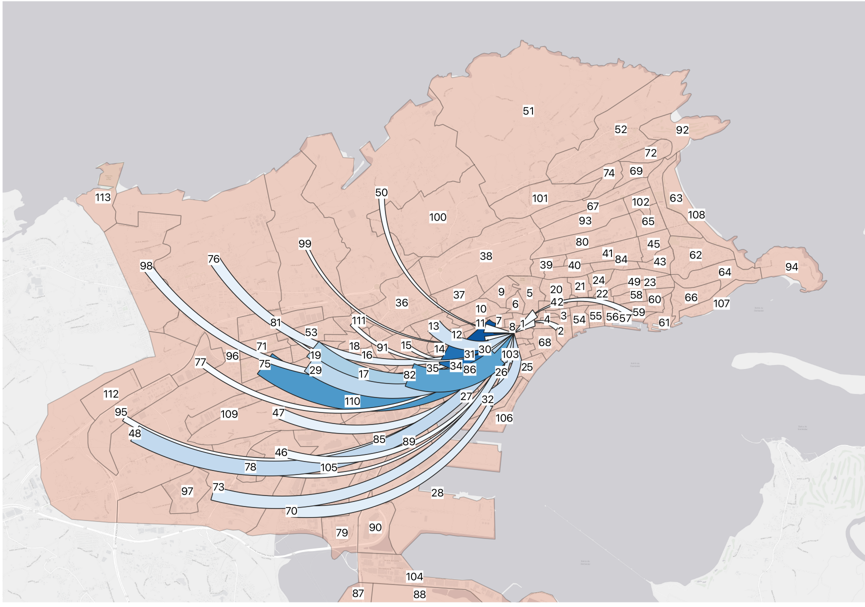


Figure 5.16: Trips attracted by zone 8 on a working day in school season

5.5.2 Trips attracted by zone 8 on a working day in school season

Zone 77 accommodates a shopping center and several large store chains. As can be seen in fig. 5.17 (showing desire lines with more than 10 individual trips), many citizens go there on Saturdays to shop in a closed environment.

5.5.3 Trips attracted by zone 11 on a holiday in school season

Zone 11 is a popular destination for recreational activities during Sundays (restaurants, bars, pubs, etc.) Figure 5.18 shows its desire lines for more than 20 individual trips.

On the contrary to what was observed for working days in the very close zone 8 (section 5.5.1), this time citizens from all over Santander are attracted.

5.5.4 Trips attracted by zone 108 on a working day in summer

Finally, fig. 5.19 illustrates the high demand of travelling to Santander beaches during summer. Desire lines only drawn for OD pairs with 20 or more individual trips.

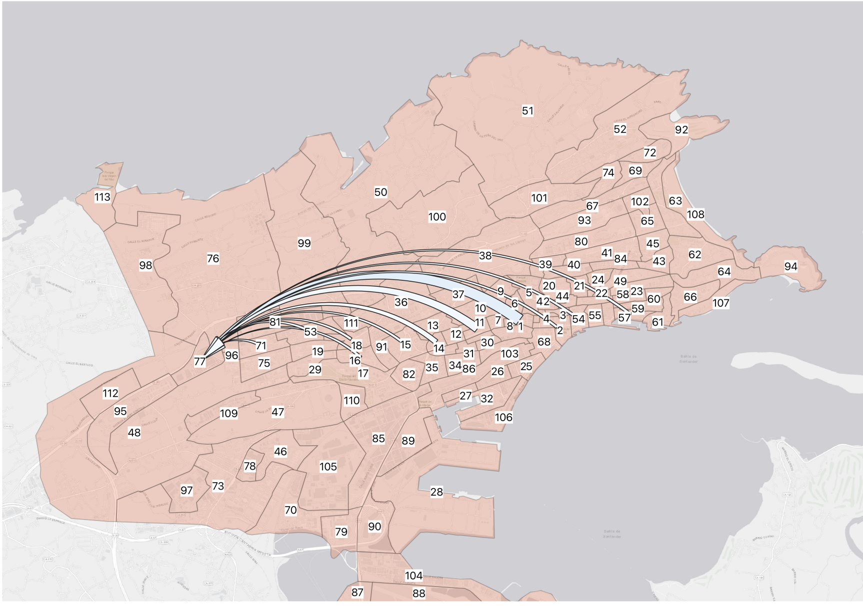


Figure 5.17: Trips attracted a Saturday during winter by zone 77

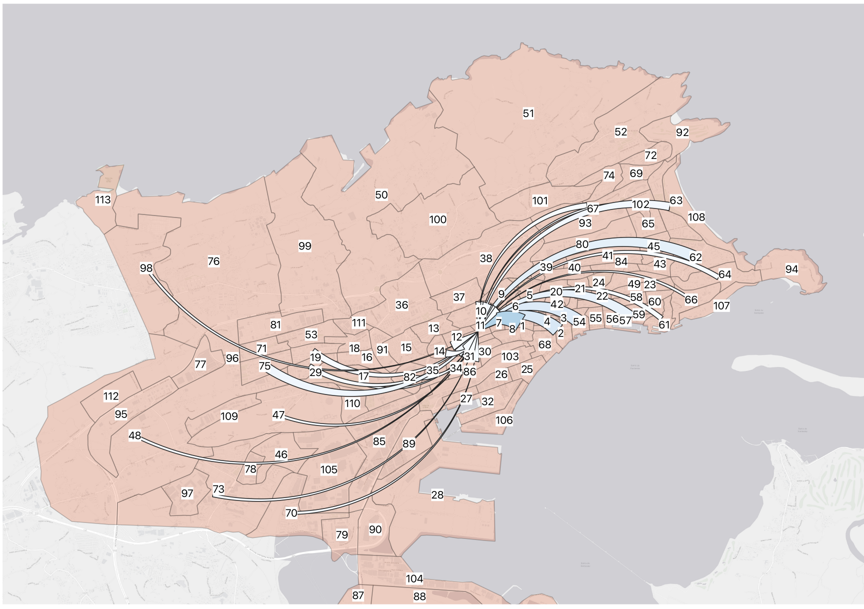


Figure 5.18: Trips attracted by zone 11 on a holiday in school season

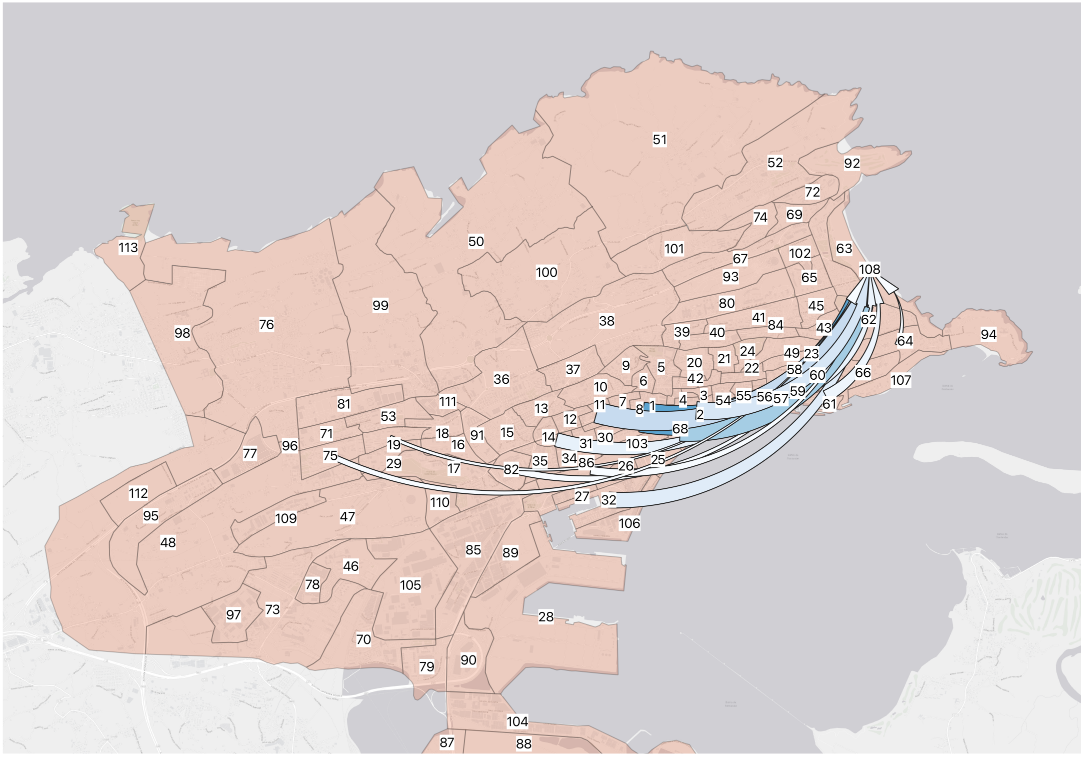


Figure 5.19: Trips attracted by zone 108 on a working day in summer

Chapter 6

Conclusions and future research

6.1 Conclusions

The methodology described in this thesis begins combining AFC, AVL, scheduling subsystem information, and probability distributions of travel and dwell times, providing a better definition of the runs of the routes offered in a public transport system and the trips (composed by one or more rides) that transit users carry out within it; ameliorating the problems that commonly occur when working with IPTS data: ambiguous ids for some elements of the system; missing or multiple entries related to the same AVL event; inconsistent run ids between the different subsystems, which impedes identifying their respective records related to the same run; AFC events with wrong information; and uncertainty regarding whether a programmed run actually took place.

Events whose attributes wrongly classify them as part of different runs are singled out and treated, as also are those unlikely to have really happened. Calls at each stop of each run are delimited considering the multiple sources of data available in that instance, providing the most likely arrival and departure times instead if there is none. A way to detect and handle those cases where the vehicle changes its id mid-run, which among other problems would lead to the misrepresentation of their load profiles, is formulated.

A run and a stop are assigned to each ticketing record, distinguishing those cases where the AFC state information and timestamp are coherent with the corresponding run call, and those where their timestamp and vehicle id are utilized to infer the ticketing action that really took place.

The discussion of the case study shows several instances where initially incompletely or wrongly recorded runs are properly defined. To show the flexibility of the methodology, it is shown how if no AVL data is available at all, the vehicle run is reconstructed approximating arrivals and departures from each stop to the earliest and latest AFC events of the corresponding boarding groups. Moreover, the program finds the event in the scheduling subsystem that likely corresponds to this rebuilt run even though the id of the planned vehicle and that of the AFC records do not match.

To evaluate how it fares characterizing calls at the termini, those starting times logged by the planning subsystem deemed to be correct have been used as empirical evidence; to compare with the outputs (without using that data) of the chosen strategy of preferring to infer the initial call in unreliable termini if the stop next to them is backed by real data, and two alternatives that were considered while writing this thesis: treat termini as any other stop, and consider planned run start times if feasible. It can be seen how the former consistently provides a better approximation of when runs have begun. This improvement may be particularly useful to better audit how closely a system adheres to its timetable.

Also, to assess the impact of bad AVL records, as well as missing AVL and detected run beginning information, those runs perfectly recorded in the original dataset (start logged by the scheduling subsystem, and all other calls derived from AVL) have been used as the ground truth, studying how their characterization deviates with different amounts of real and bogus simulated entries. The results show significant improvements: with as little as 25 % of real AVL and run start detection data, even when adding 4 times as many wrong entries the results are significantly better than those from applying the methodology using only AFC records. The more real data is available, the closer the characterization is to the run that took place, and the more resistant it is to incorrect values: for instance, if 100 % of the real information is utilized, the methodology can completely filter out twice as many erroneous data.

This work makes use of the improved definitions of runs and boarding events with a trip chaining model, which successfully captures the mobility patterns of users in the city, thanks to several enhancements that, as illustrated again in the discussion of the case study, improve the likelihood of the inferred trips and runs being correct. The case where a single SC may be used to grant access to several passengers is considered, as also is the possibility that a single ride may overlap one or two vehicle runs, which may correspond to different configurations of a route.

One of these enhancements is an analysis of frequent activity destinations, specifying the area where they are located within the city, and the time window when the user arrives through DBSCAN. They have provided another way to find the last alighting of some inferred trips that would otherwise remain undefined, and a representation of the spatial and temporal mobility patterns in the city, for 6 day types.

Individual trips have been aggregated and expanded to account for cash payments and for those SCs validations for which a destination could not be inferred, producing OD matrices for different part of the year.

This methodology is applicable to situations with different scheduling information: none, planned beginnings only, or scheduled and detected (but not necessarily correct) start times; while the id of the originally intended vehicle may be known or not. These improvements help provide more accurate depictions of user rides and vehicle runs.

6.2 Future research

As his next objective, the author is currently studying the extension of this methodology to provide more accurate vehicle load profiles.

Also, other interesting line of work is the integration within the ecosystem of a running IPTS, to have directly available the improved representation of what has occurred in the system. The way to approach this task would be to start with an application which would provide daily updated definitions, and to build up from that point forward (hourly, and finally 'near-real-time' information).

Another possible line of investigation is the utilization of other probability distributions to model dwell and travel times, or use more sophisticated models to calculate them.

It is likely that weather and comfort influence the attractiveness of taking a bus versus walking. Thus, it would be interesting to modulate the walk penalization factor f_w of the trip chaining model according to weather and bus occupancy.

Another variable that would be interesting to factor in the selection of the trip chaining model parameters is the type of SC card utilized in each transaction, since they correspond to different demographic groups. For instance, senior citizens and people with disabilities are more likely to get on the bus for short trips if the street is too steep, as are owners of SCs which provide a flat rate.

Also, it may be worth exploring utilizing existing research to infer the purpose of trips to improve the definition of the location and time window of frequent activity nodes, since they may show different spatial and temporal variabilities according to their nature. For instance, 'work' or 'go to school' should usually be quite rigid temporally, while the time when people return home can be more flexible. On the other hand, 'leisure' activities may be less precise spatially than other types.

And finally, even though this work has focused on urban public bus transport, it might be worth exploring its applicability to other ambits which similar characteristics, such as intercity bus services.

Bibliography

[1] Ennio Cascetta. *Transportation systems analysis: models and applications.* New York, 2009. DOI: 10.1007/978-0-387-75857-2.

[2] *PostgreSQL: About.* URL: https://www.postgresql.org/about/ (visited on 10/28/2019).

[3] Ding Luo et al. "Constructing Spatiotemporal Load Profiles of Transit Vehicles with Multiple Data Sources". In: *Transportation Research Record* 2672.8 (2018), pp. 175–186. ISSN: 21694052. DOI: 10.1177/0361198118781166.

[4] Azalden A M Alsger and B Eng. *Estimation of transit origin destination matrices using smart card fare data School of Civil Engineering*. Tech. rep. 2016.

[5] *What Is NFC? - NFC Forum — NFC Forum.* URL: https://nfc-forum.org/what-is-nfc/ (visited on 10/14/2019).

[6] *About Smart Cards : Introduction : Primer – Secure Technology Alliance.* URL: https://www.securetechalliance.org/smart-cards-intro-primer/ (visited on 10/14/2019).

[7] *SQL Introduction.* URL: https://www.w3schools.com/sql/sql{_}intro.asp (visited on 10/13/2019).

[8] J Robert. *Les tramways parisiens*. G. Fuseau, 1959. URL: https://books.google.es/books?id=6rihR0KTNRsC.

[9] Ayuntamiento de Santander. *Datos Abiertos Santander — Dataset — programacion-tus.* URL: http://datos.santander.es/dataset/?id=programacion-tus (visited on 06/27/2021).

[10] Peter G Furth et al. "TCRP Web Document 23 (Project H-28): Contractor's Final Report Uses of Archived AVL-APC Data to Improve Transit Performance and Management: Review and Potential". In: 23.June 2003 (2003).

[11] Martin Trépanier, Nicolas Tranchant, and Robert Chapleau. "Individual trip destination estimation in a transit smart card automated fare collection system". In: *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 11.1 (2007), pp. 1–14. ISSN: 15472450. DOI: 10.1080/15472450601122256.

[12] Marcela A. Munizaga and Carolina Palma. "Estimation of a disaggregate multimodal public transport Origin-Destination matrix from passive smart-card data from Santiago, Chile". In: *Transportation Research Part C: Emerging Technologies* 24 (2012), pp. 9–18. ISSN: 0968090X. DOI: 10.1016/j.trc.2012.01.007. URL: http://dx.doi.org/10.1016/j.trc.2012.01.007.

[13] Jason B. Gordon et al. "Automated inference of linked transit journeys in london using fare-transaction and vehicle location data". In: *Transportation Research Record* 2343 (2013), pp. 17–24. ISSN: 03611981. DOI: 10.3141/2343-03.

[14] Sabeur Elkosantini and Saber Darmoul. "Intelligent public transportation systems: A review of architectures and enabling technologies". In: *2013 International Conference on Advanced Logistics and Transport, ICALT 2013* (2013), pp. 233–238. DOI: 10.1109/ICAdLT.2013.6568465.

[15] Laura Cecilia Cham. "Understanding Bus Service Reliability :" in: (2006).

[16] Luis Moreira-Matias et al. "Improving Mass Transit Operations by Using AVL-Based Systems: A Survey". In: *IEEE Transactions on Intelligent Transportation Systems* 16.4 (2015), pp. 1636–1653. ISSN: 15249050. DOI: 10.1109/TITS.2014.2376772.

[17] Carmen de Pablos Heredero, Luis J. Perez Bermejo, and Jose Luis Montes Botella. "Impacto de los sistemas de apoyo a la explotación (SAE) en la mejora de los servicios de transporte público urbano". In: *Cuadernos de Economía y Dirección de la Empresa* 15.1 (2012), pp. 12–24. ISSN: 11385758. DOI: 10.1016/j.cede.2011.07.001. URL: http://dx.doi.org/10.1016/j.cede.2011.07.001.

[18] Martin Trépanier, Catherine Morency, and Bruno Agard. "Calculation of Transit Performance Measures Using Smartcard Data". In: *Journal of Public Transportation* 12.1 (2009), pp. 79–96. ISSN: 1077-291X. DOI: 10.5038/2375-0901.12.1.5.

[19] Nigel H.M. Wilson, Jinhua Zhao, and Adam Rahbee. "The potential impact of automated data collection systems on urban public transport planning". In: *Operations Research/ Computer Science Interfaces Series* 46.1 (2009), pp. 75–99. ISSN: 1387666X. DOI: 10.1007/978-0-387-84812-9_5.

[20] Xiaolei Ma et al. "Development of a data-driven platform for transit performance measures using smart card and GPS data". In: *Journal of Transportation Engineering* 140.12 (2014), pp. 1–13. ISSN: 0733947X. DOI: 10.1061/(ASCE)TE.1943-5436.0000714.

[21] Robert L. Bertini and Ahmed El-Geneidy. "Generating transit performance measures with archived data". In: *Transportation Research Record* 1841 (2003), pp. 109–119. ISSN: 03611981. DOI: 10.3141/1841-12.

[22] Yongjie Lin et al. "Real-time bus arrival time prediction: Case study for Jinan, China". In: *Journal of Transportation Engineering* 139.11 (2013), pp. 1133–1140. ISSN: 0733947X. DOI: 10.1061/(ASCE)TE.1943-5436.0000589.

[23] Google. *GTFS Static Overview — Static Transit — Google Developers*. URL: https://developers.google.com/transit/gtfs (visited on 11/08/2020).

[24] MobilityData. *General Transit Feed Specification*. URL: https://gtfs.org/changes/ (visited on 11/08/2020).

[25] OVapi B.V. *OVapi*. URL: http://ovapi.nl/ (visited on 11/08/2020).

[26] Mapzen Foundation and Interline Technologies LLC. *Welcome • Transitland*. URL: https://www.transit.land/ (visited on 11/08/2020).

[27] MobilityData IO. *OpenMobilityData - Public transit feeds from around the world*. URL: http://transitfeeds.com/ (visited on 11/08/2020).

[28] Nate Wessel and Steven Farber. "On the accuracy of schedule-based GTFS for measuring accessibility". In: *Journal of Transport and Land Use* 12.1 (2019), pp. 475–500. ISSN: 19387849. DOI: 10.5198/jtlu.2019.1502.

[29] M Sandim et al. "Using GPS-based AVL data to calculate and predict traffic network performance metrics: A systematic review". In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. 2016, pp. 1692–1699.

[30] European Global Navigation Satellite Systems Agency. *What is GNSS? — European Global Navigation Satellite Systems Agency*. URL: https://www.gsa.europa.eu/european-gnss/what-gnss (visited on 10/12/2020).

[31] Per van Diggelen, Frank, Enge. "The World's first GPS MOOC and Worldwide Laboratory using Smartphones". In: *28th International Technical Meeting of the Satellite Division of The Institute of Navigation*. Tampa, Florida, 2015, pp. 361–369.

[32] Ilgin Gokasar and Yigit Cetinel. "A New strategy for the diagnosis of the bus headways using avl data". In: *MT-ITS 2019 - 6th International Conference on Models and Technologies for Intelligent Transportation Systems* (2019). DOI: 10.1109/MTITS.2019.8883314.

[33] Nick Hounsell and Birendra Shrestha. "A new approach for co-operative bus priority at traffic signals". In: *IEEE Transactions on Intelligent Transportation Systems* 13.1 (2012), pp. 6–14. ISSN: 15249050. DOI: 10.1109/TITS.2011.2172869.

[34] Hamed Saghaei. "Design and Implementation of a Fleet Management System Using Novel GPS/GLONASS Tracker and Web-Based Software-Automatic Vehicle Locator (AVL), Fleet Management System (FMS), Global Positioning System (GPS), Global Navigation Satellite System (GLONASS), Ge". In: (2016).

[35] Haitao Xu and Jing Ying. "Bus arrival time prediction with real-time and historic data". In: *Cluster Computing* 20.4 (2017), pp. 3099–3106. ISSN: 15737543. DOI: 10.1007/s10586-017-1006-1.

[36] Dejan Rančić, Bratislav Predić, and Vladan Mihajlović. "Online and post-processing of AVL data in public bus transportation system". In: *WSEAS Transactions on Information Science and Applications* 5.3 (2008), pp. 229–236. ISSN: 17900832.

[37] Philip T. Blythe. "Improving public transport ticketing through smart cards". In: *Proceedings of the Institution of Civil Engineers: Municipal Engineer* 157.1 (2004), pp. 47–54. ISSN: 09650903. DOI: 10.1680/muen.2004.157.1.47.

[38] Bekir Bartin, Kaan Ozbay, and Hong Yang. "Evaluation framework for mobile ticketing applications in public transit: A case study". In: *IET Intelligent Transport Systems* 12.9 (2018), pp. 1166–1173. ISSN: 1751956X. DOI: 10.1049/iet-its.2018.5248.

[39] ITS España Foro de Nuevas Tecnologías en el Transporte. *White paper on the application on NFC technology in the public transport*. December. 2013, p. 94. ISBN: 9788461647149.

[40] Francisco Liébana-Cabanillas, Sebastian Molinillo, and Miguel Ruiz-Montañez. "To use or not to use, that is the question: Analysis of the determining factors for using NFC mobile payment systems in public transportation". In: *Technological Forecasting and Social Change* 139.August 2018 (2019), pp. 266–276. ISSN: 00401625. DOI: 10.1016/j.techfore.2018.11.012. URL: https://doi.org/10.1016/j.techfore.2018.11.012.

[41] Marie Pier Pelletier, Martin Trépanier, and Catherine Morency. "Smart card data use in public transit: A literature review". In: *Transportation Research Part C: Emerging Technologies* 19.4 (2011), pp. 557–568. ISSN: 0968090X. DOI: 10.1016/j.trc.2010.12.003.

[42] Le Minh Kieu, Ashish Bhaskar, and Edward Chung. “Passenger segmentation using smart card data”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.3 (2015), pp. 1537–1548. ISSN: 15249050. DOI: 10.1109/TITS.2014.2368998.

[43] Takahiko Kusakabe and Yasuo Asakura. “Behavioural data mining of transit smart card data: A data fusion approach”. In: *Transportation Research Part C: Emerging Technologies* 46 (2014), pp. 179–191. ISSN: 0968090X. DOI: 10.1016/j.trc.2014.05.012. URL: http://dx.doi.org/10.1016/j.trc.2014.05.012.

[44] Boris Chidlovskii. “Mining smart card data for travellers' mini activities”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.11 (2018), pp. 3676–3685. ISSN: 15249050. DOI: 10.1109/TITS.2018.2852493. arXiv: 1712.06935.

[45] Le Minh Kieu et al. “An investigation of timed transfer coordination using event-based multi agent simulation”. In: *Transportation Research Part C: Emerging Technologies* 81 (2017), pp. 363–378. ISSN: 0968090X. DOI: 10.1016/j.trc.2017.02.018. URL: http://dx.doi.org/10.1016/j.trc.2017.02.018.

[46] Ka Kee Alfred Chu and Robert Chapleau. “Enriching archived smart card transaction data for transit demand modeling”. In: *Transportation Research Record* 2063 (2008), pp. 63–72. ISSN: 03611981. DOI: 10.3141/2063-08.

[47] Fumitaka Kurauchi and Jan Dirk Schmöcker. *Public transport planning with smart card data*. English. 2017, pp. 1–264. ISBN: 9781498726597. DOI: 10.1201/9781315370408.

[48] Ding Luo, Oded Cats, and Hans van Lint. “Constructing transit origin–destination matrices with spatial clustering”. In: *Transportation Research Record* 2652.1 (2017), pp. 39–49. ISSN: 21694052. DOI: 10.3141/2652-05.

[49] Jason B. Gordon, Haris N. Koutsopoulos, and Nigel H.M. Wilson. “Estimation of population origin–interchange–destination flows on multimodal transit networks”. In: *Transportation Research Part C: Emerging Technologies* 90.January (2018), pp. 350–365. ISSN: 0968090X. DOI: 10.1016/j.trc.2018.03.007. URL: https://doi.org/10.1016/j.trc.2018.03.007.

[50] Jinhua Zhao et al. “Unified estimator for excess journey time under heterogeneous passenger incidence behavior using smartcard data”. In: *Transportation Research Part C: Emerging Technologies* 34 (2013), pp. 70–88. ISSN: 0968090X. DOI: 10.1016/j.trc.2013.05.009. URL: http://dx.doi.org/10.1016/j.trc.2013.05.009.

[51] Peter G. Furth et al. *Using Archived AVL-APC Data to Improve Transit Performance and Management*. Washington, DC: The National Academies Press, 2006. ISBN: 9780309098618. DOI: 10.17226/13907. URL: https://www.nap.edu/catalog/13907/using-archived-avl-apc-data-to-improve-transit-performance-and-management.

[52] Zhen Chen and Wei Fan. "Extracting bus transit boarding and alighting information using smart card transaction data". In: *Journal of Public Transportation* 22.1 (2020), pp. 1–23. ISSN: 23750901. DOI: 10.5038/2375-0901.22.1.3.

[53] Steve Robinson et al. "Methods for pre-processing smartcard data to improve data quality". In: *Transportation Research Part C: Emerging Technologies* 49 (2014), pp. 43–58. ISSN: 0968090X. DOI: 10.1016/j.trc.2014.10.006. URL: http://dx.doi.org/10.1016/j.trc.2014.10.006.

[54] Etikaf Hussain, Ashish Bhaskar, and Edward Chung. "Transit OD matrix estimation using smartcard data: Recent developments and future research challenges". In: *Transportation Research Part C: Emerging Technologies* 125.August 2020 (2021), p. 103044. ISSN: 0968090X. DOI: 10.1016/j.trc.2021.103044. URL: https://doi.org/10.1016/j.trc.2021.103044.

[55] Luis Moreira-matias. "Towards an AVL-based Demand Estimation Model". In: (2016).

[56] Kelvin Buneman. "Automated and Passenger-Based Transit Performance Measures." In: *Transportation Research Record* (1984), pp. 23–28. ISSN: 03611981.

[57] Antonio Gschwender, Marcela Munizaga, and Carolina Simonetti. "Using smart card and GPS data for policy and planning: The case of Transantiago". In: *Research in Transportation Economics* 59 (2016), pp. 242–249. ISSN: 07398859. DOI: 10.1016/j.retrec.2016.05.004.

[58] Ka Kee Alfred Chu, Robert Chapleau, and Martin Trépanier. "Driver-assisted bus interview passive transit travel survey with smart card automatic fare collection system and applications". In: *Transportation Research Record* 2105 (2009), pp. 1–10. ISSN: 03611981. DOI: 10.3141/2105-01.

[59] Zilin Huang et al. "Citywide Metro-to-Bus Transfer Behavior Identification Based on Combined Data from Smart Cards and GPS". In: *applied sciences* (2019).

[60] Guojun Chen et al. "Bus-Arrival-Time Prediction Models: Link-Based and Section-Based". In: *Journal of Transportation Engineering* 138.1 (2011), pp. 60–66. ISSN: 0733947X. DOI: 10.1061/(ASCE)TE.1943-5436.0000312.

[61] Zhuang Dai, Xiaolei Ma, and Xi Chen. “Bus travel time modelling using GPS probe and smart card data: A probabilistic approach considering link travel time and station dwell time”. In: *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 23.2 (2019), pp. 175–190. ISSN: 15472442. DOI: 10.1080/15472450.2018.1470932. URL: https://doi.org/10.1080/15472450.2018.1470932.

[62] H Levinson. “ANALYZING TRANSIT TRAVEL TIME PERFORMANCE”. In: *Transportation Research Record* (1983).

[63] Xiaobo Qu et al. “Bus travel time reliability analysis: A case study”. In: *Proceedings of the Institution of Civil Engineers: Transport* 167.3 (2014), pp. 178–184. ISSN: 17517710. DOI: 10.1680/tran.13.00009.

[64] Michael.A.P. Taylor and Susilawati. “Modelling Travel Time Reliability with the Burr Distribution”. In: *Procedia - Social and Behavioral Sciences* 54.November 2014 (2012), pp. 75–83. ISSN: 18770428. DOI: 10.1016/j.sbspro.2012.09.727.

[65] Akhilesh Chepuri et al. “Examining travel time reliability-based performance indicators for bus routes using GPS-based bus trajectory data in India”. In: *Journal of Transportation Engineering Part A: Systems* 144.5 (2018). ISSN: 24732893. DOI: 10.1061/JTEPBS.0000109.

[66] M.M. Harsha, Raviraj H Mulangi, and H.D. Dinesh Kumar. “Analysis of Bus Travel Time Variability using Automatic Vehicle Location Data”. In: *Transportation Research Procedia* 48.2018 (2020), pp. 3283–3298. ISSN: 23521465. DOI: 10.1016/j.trpro.2020.08.123. URL: https://doi.org/10.1016/j.trpro.2020.08.123.

[67] Mingxin Li, Xuesong Zhou, and Nagui M. Rouphail. “Quantifying travel time variability at a single bottleneck based on stochastic capacity and demand distributions”. In: *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 21.2 (2017), pp. 79–93. ISSN: 15472442. DOI: 10.1080/15472450.2016.1163639.

[68] Karthik K. Srinivasan, A. A. Prakash, and Ravi Seshadri. “Finding most reliable paths on networks with correlated and shifted log-normal travel times”. In: *Transportation Research Part B: Methodological* 66 (2014), pp. 110–128. ISSN: 01912615. DOI: 10.1016/j.trb.2013.10.011. URL: http://dx.doi.org/10.1016/j.trb.2013.10.011.

[69] Bi Yu Chen et al. “Most reliable path-finding algorithm for maximizing on-time arrival probability”. In: *Transportmetrica B* 5.3 (2017), pp. 253–269. ISSN: 21680582. DOI: 10.1080/21680566.2016.1169953.

[70] Rajat Rajbhandari, Steven I. Chien, and Janice R. Daniel. “Estimation of bus dwell times with automatic passenger counter information”. In: *Transportation Research Record* 1841 (2003), pp. 120–127. ISSN: 03611981. DOI: 10.3141/1841-13.

[71] H Zhang, H Cui, and B Shi. “A Data-Driven Analysis for Operational Vehicle Performance of Public Transport Network”. In: *IEEE Access* 7 (2019), pp. 96404–96413. DOI: 10.1109/ACCESS.2019.2930279.

[72] Aimsun. “Public Transport Lines”. In: *Aimsun Next 8.4 Users' Manual.* Aimsun, 2020.

[73] PTV Group. “1.17.6.4 Using time distributions”. In: *PTV Vissim 2021 User Manual.* 2020, p. 240.

[74] Reebu Zachariah Koshy and V. Thamizh Arasan. “Influence of bus stops on flow characteristics of mixed traffic”. In: *Journal of Transportation Engineering* 131.8 (2005), pp. 640–643. ISSN: 0733947X. DOI: 10.1061/(ASCE)0733-947X(2005)131:8(640).

[75] Soroush RASHIDI and Prakash RANJITKAR. “Approximation and Short-Term Prediction of Bus Dwell Time using AVL Data”. In: *Journal of the Eastern Asia Society for Transportation Studies* 10 (2013), pp. 1281–1291.

[76] Xinkai Jiang and Xiaoguang Yang. “Regression-based models for bus dwell time”. In: *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014* (2014), pp. 2858–2863. DOI: 10.1109/ITSC.2014.6958148.

[77] Tian Li et al. “Smart Card Data Mining of Public Transport Destination : A Literature Review”. In: (2018), pp. 28–30. DOI: 10.3390/info9010018.

[78] Xin Yang et al. “Short-term prediction of passenger volume for urban rail systems: A deep learning approach based on smart-card data”. In: *International Journal of Production Economics* 231.September 2020 (2021), p. 107920. ISSN: 09255273. DOI: 10.1016/j.ijpe.2020.107920. URL: https://doi.org/10.1016/j.ijpe.2020.107920.

[79] James J. Barry et al. “Origin and destination estimation in New York City with automated fare system data”. In: *Transportation Research Record* 1817 (2002), pp. 183–187. ISSN: 03611981. DOI: 10.3141/1817-24.

[80] Martin Trépanier, Nicolas Tranchant, and Robert Chapleau. “Individual trip destination estimation in a transit smart card automated fare collection system”. In: *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations* 11.1 (2007), pp. 1–14. ISSN: 15472450. DOI: 10.1080/15472450601122256.

[81] Jinhua Zhao, Adam Rahbee, and Nigel H.M. Wilson. "Estimating a rail passenger trip origin-destination matrix using automatic data collection systems". In: *Computer-Aided Civil and Infrastructure Engineering* 22.5 (2007), pp. 376–387. ISSN: 10939687. DOI: 10.1111/j.1467-8667.2007.00494.x.

[82] Markus Hofmann and Margaret O'Mahony. "Transfer journey identification and analyses from electronic fare collection data". In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* 2005.October (2005), pp. 34–39. DOI: 10.1109/ITSC.2005.1520156.

[83] Wonjae Jang. "Travel time and transfer analysis using transit smart card data". In: *Transportation Research Record* 2144 (2010), pp. 142–149. ISSN: 03611981. DOI: 10.3141/2144-16.

[84] Flavio Devillaine, Marcela Munizaga, and Martin Trépanier. "Detection of activities of public transport users by analyzing smart card data". In: *Transportation Research Record* 2276 (2012), pp. 48–55. ISSN: 03611981. DOI: 10.3141/2276-06.

[85] Neema Nassir, Mark Hickman, and Zhen Liang Ma. "Activity detection and transfer identification for public transit fare card data". In: *Transportation* 42.4 (2015), pp. 683–705. ISSN: 15729435. DOI: 10.1007/s11116-015-9601-6. URL: http://dx.doi.org/10.1007/s11116-015-9601-6.

[86] Mariko Utsunomiya, John Attanucci, and Nigel Wilson. "Potential uses of transit smart card registration and transaction data to improve transit planning". In: *Transportation Research Record* 1971 (2006), pp. 119–126. ISSN: 03611981. DOI: 10.3141/1971-16.

[87] Samiul Hasan et al. "Spatiotemporal Patterns of Urban Human Mobility". In: *Journal of Statistical Physics* 151.1-2 (2013), pp. 304–318. ISSN: 00224715. DOI: 10.1007/s10955-012-0645-0.

[88] Catherine Morency, Martin Trépanier, and Bruno Agard. "Measuring transit use variability with smart-card data". In: *Transport Policy* 14.3 (2007), pp. 193–203. ISSN: 0967070X. DOI: 10.1016/j.tranpol.2007.01.001.

[89] Sang Gu Lee and Mark Hickman. "Trip purpose inference using automated fare collection data". In: *Public Transport* 6.1-2 (2014), pp. 1–20. ISSN: 16137159. DOI: 10.1007/s12469-013-0077-5.

[90] Ka Kee Alfred Chu and Robert Chapleau. "Augmenting transit trip characterization and travel behavior comprehension: Multiday location-stamped smart card transactions". In: *Transportation Research Record* 2183 (2010), pp. 29–40. ISSN: 03611981. DOI: 10.3141/2183-04.

[91] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (1996), pp. 226–231.

[92] Xiaolei Ma et al. "Mining smart card data for transit riders' travel patterns". In: *Transportation Research Part C: Emerging Technologies* 36 (2013), pp. 1–12. ISSN: 0968090X. DOI: 10.1016/j.trc.2013.07.010. URL: http://dx.doi.org/10.1016/j.trc.2013.07.010.

[93] Zhanhong Cheng, Martin Trépanier, and Lijun Sun. "Probabilistic model for destination inference and travel pattern mining from smart card data". In: *Transportation* 0123456789 (2020). ISSN: 15729435. DOI: 10.1007/s11116-020-10120-0. URL: https://doi.org/10.1007/s11116-020-10120-0.

[94] Watabou. *watabou - Random City Generator*. URL: https://watabou.itch.io/ (visited on 06/13/2021).

[95] *File:Spain map modern.png - Wikimedia Commons*. URL: https://commons.wikimedia.org/wiki/File:Spain{_}map{_}modern.png (visited on 11/16/2019).

[96] *Santander: A Digital Smart City Prototype in Spain - SPIEGEL ONLINE*. URL: https://www.spiegel.de/international/world/santander-a-digital-smart-city-prototype-in-spain-a-888480.html (visited on 11/16/2019).

Chapter 7

Appendices

7.1 Daily OD matrices

	1	2	3	4	5	6	8	9	11	13	14	15	16	17	18	19	20	21	22	23	24	29	30	31	32	33	34	35	36	37	38
1	158	1	0		30		1	17	638	15	602	325	330	28	74	44	19	26	0	17	50	127	0	0	0		0	0	52	3	20
2	325	63	16	0	0		45	2	338	15	385	149	134	22	29	25	0	4	8	26	31	46	0	0	0	0	0	1	19	1	13
3				0									0		0	0			85	55		0									
4	43	229			0	0	0	27	53	3	76	0	0	0	0	0	72	0	0	21	25	0	0	0	0				0	1	39
5	0	0		2	8		0		0	0	0	0	0				28	13	3	25	34	0								0	
6					15			66		0	0	0	0		0		5	6	6	0	27									0	
8	0	184	27	0	0		221	0	0	0	2	10	5	0	6	0	0	0	88	80	0	8	62	31	1			17	9	0	1
9	0	0		1	79	10	0	43	0	0	0	0	0		0		41	34	9	37	55	0			0				0	0	
11	0	202	16	0	76	0	568	36	0	10	352	107	177	14	42	26	38	39	55	56	56	65	0	0	0		0	0	24	8	9
13	0	35	6	0	21	0	174	10	32	45	0	26	0	0	1	2	8	10	10	10	11	0	0	0	23	8	0		2	11	
14		3	0		155	0	2	125	0	50	67	116	126	14	38	25	76	76	0	45	114	45	0		0				18	40	11
15	0	7	3	0	0		35	0	4	7	0	2	104	10	13	8	0	0	1	0	0	24	0		15	2	1	0	4	0	4
16		3	0	0	0		12	0	4	42		2	9	0	14	3	0	0	0	0	1	4			14	1		0	8	0	21
17		37	0	0	3		174	2	94	0	2	0	7	22	0	0	0	0	0	0	1	0	1		0	66	0	0	2	0	6
18		2	0	0	0		13	0	5	16	0	1		0	0	1	0	0	0	0	0	3	0		3	1			13	0	22
19		44	1	0	3		255	3	121	19	3	0	2	7	0	21	2	1	1	4	3	19	0	0	0	99	0	0	0	0	0
20	0	0		6		2	0	0	0	0	0	0	0		0		4	18	1	22	31	0	0	0				0	0	0	2
21	0	0		9			0		0	0	0	0	0		0				1	13	11	0	0	0				0			0
22	17	7		36			0	0	41	2	35	2	0	0	0	0	0		37	40		0	0	0	0		0	0	0	0	0
23	56	36	0	84	0		1	0	62	1	37	3	3	0	1	0	0		28	14		1	0	0	0			0	0	0	0
24	2	0		51			1		1	0	1	0	0	0	0	0			7	9	5	0	0					0	0		
29		38	1	0	2		220	2	110	1	2	0		28	0	0	1	1	1	1	1	2	0	0	0	83	1		0	0	0
30		3	0	0	0		26	0	0	0	0	9	1	0	0	0	0	0	0	0	0	0	4	4	0			2	0	0	0
31		0	0									0	1	0	0	0		0	0			0						1			
32	0	68	9		0		177	1	34	0	8	0	0	0	0	0	0	0	7	3	4	1	0	0	96			0	0	0	0
33	0	139	9	0	0		508	0	152	0	0	0	1	0	0	0	0	0	29	18	1	1	0	0	0			0	0	0	0
34		1	0	0	0		17	0	0			0	0				0	0	0	0	0	0	3					0			
35		6	0	0	0		35	0	9	1	0	5	6	0	1	0	0	1	0	0	0	0	5		0	6	3	12	0	0	0
36	0	34	0	0	0		28	0	11	14	0	0	0	0	17	12	1	0	0	0	0	14	0	0	0	2			32	0	97
37		4	4	0	17	0	20	18	0	40		25	4	0	0	0	15	14	3	14	17	0	0	0	0		0		0	37	0
38	7	72	0	0	0	8	8	2	13	35	4	25	5	0	10	0	0	0	0	0	1	15	0	0	0	1	0	0	16	68	115
39	0	0				9	0	3	0	62	0	39	3	0	1	0		0	0	0	1	1	0	0	0		1		0	81	73
40		0		0		0	0	0	0	73		59	6	0	0	0				0		0	0		0		0		1	119	22
42	2	1		33		0	0	2	11	1	16	1	0	0	0	0	11	0		2	1	0	0	0	0		0		0	0	3
43				0		0	0		0	34		24	1	0	0	0				23		0	0		0	0	0		3	57	19
44	12	5		26	0		0	0	36	1	40	1	0	0	0	0	0			1	0	0	0	0	0		0	0	0	1	
46	0	11	0	0	7		61	7	30	12	0	0	5	4	0	0	4	2	0	0	2	0	4	0	41	27	2	16	0	4	0
47		17	0		1		98	1	55	0	2	0	2	5	0	0	1	0	0	1	2	0	0		0	51	0	0	0	0	0
48	0	35	0	0	2	0	210	2	117	42	0	0	18	35	1	12	2	0	0	0	1	11	0	0	1	83	0	0	8	0	16
49	15	7		30			0	0	28	0	19	0	0	0	0	0	0		7			0	0	0	0		0	0	0	0	
50		11	0		0		42	0	1	15	0	1	0	0	0	0	0	0	0	0	0	0	0		0	5			18	0	65
51	3	2			0				2		3	1	0			0	0				0								0		
52	68	31	0	0	0			0	29	0	16	11	5	0	0	1	0	0	0	0	1	0	0	0	0			0	0	0	0
53	0	2			0		8	0	3	4		0		0		0	0	0	0	0	0	0	0			0	0		4	0	1
54	139	176	0		0		18	1	216	3	173	91	68	7	16	10	0	0	0	2	0	27	3	1	0		0	0	14	0	7
56	27	46	0	0	0		0	0	39	2	45	19	1	2	6	2	0		0	6		2	0	0	0	0	0	0	17	0	12
57	324	287	0	0	1	0	46	1	229	1	138	80	52	0	10	6	0	0	1	0	0	25	2	0	0	0	0	0	3	0	4
58	12	4		0				0	6	0	4	0	0	0	0	0			0	0			0	0	0				0	0	0
59	16	8	0	0	0		0	0	8	0	6	0	0	0	0	0	0	0	0	4		0	0	0	0			0	0	0	0
61	198	170	0	0	0	0	30	1	94	0	64	37	18	0	6	2	0	0	0	0	0	10	1	0	0		0	0	8	0	6
62	156	163	0	0	0	0	23	0	88	0	44	21	12	1	2	2	1	0	2	0	1	4	1	0	0		0	1	0	0	0
63	115	111	0	0	0		12	0	57	0	26	18	9	0	8	0	0	0	0	0	0	9	0	0	0			0	14	0	15
64	45	53	0	0	0		8	0	28	0	16	7	6	0	5	0	0	0	0	0	0	6	0	0	0		0	0	4	0	7
65	12	10	0				2	0	9	0	4	3	0	0	3	0	0	0	0	0		10	0	0	0			0	8	0	14

Table 7.1: Daily OD matrix of a working day (winter, spring, and autumn)

Table 7.2: Daily OD matrix of a Saturday (winter, spring, and autumn)

Table 7.3: Daily OD matrix of a holiday (winter, spring, and autumn)

Table 7.4: Daily OD matrix of a working day (summer)

Table 7.5: Daily OD matrix of a Saturday (summer)

Table 7.6: Daily OD matrix of a holiday (summer)

7.2 Runs from other Santander routes

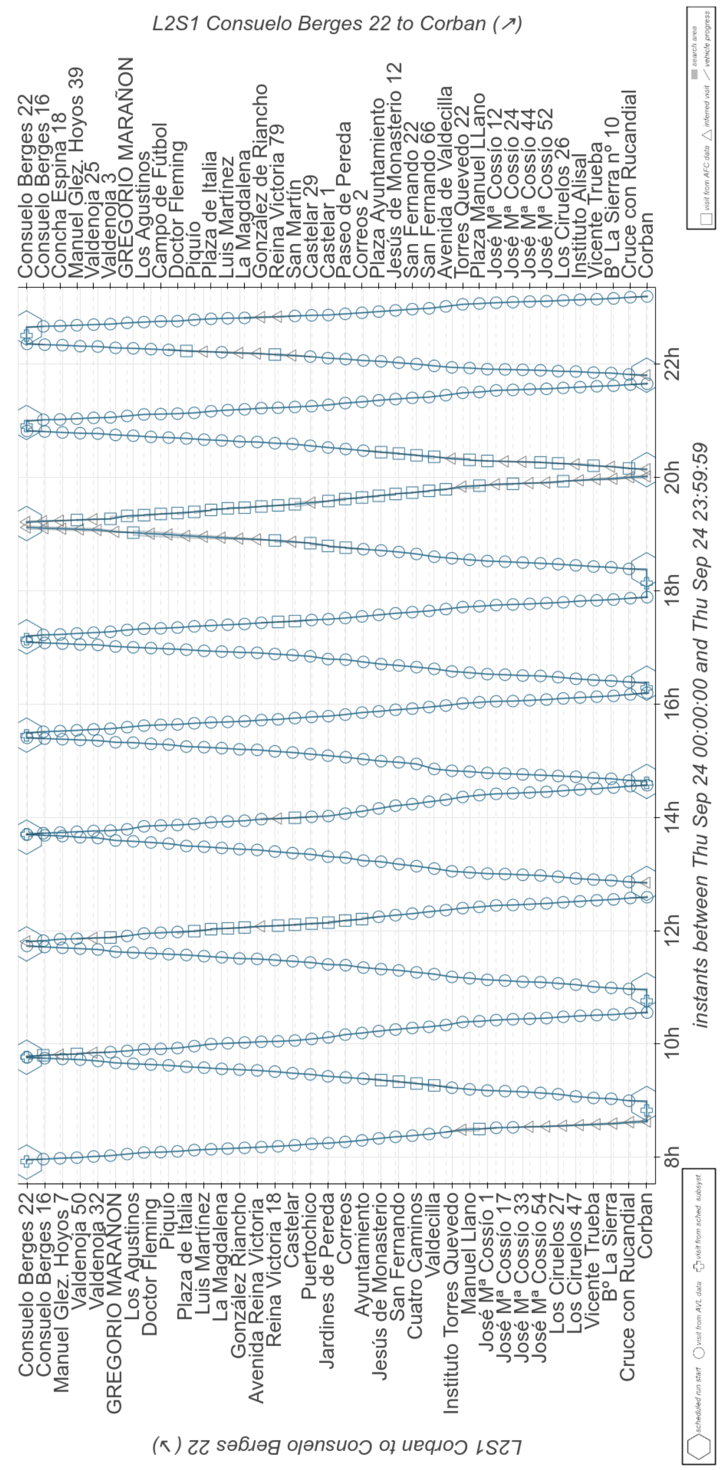
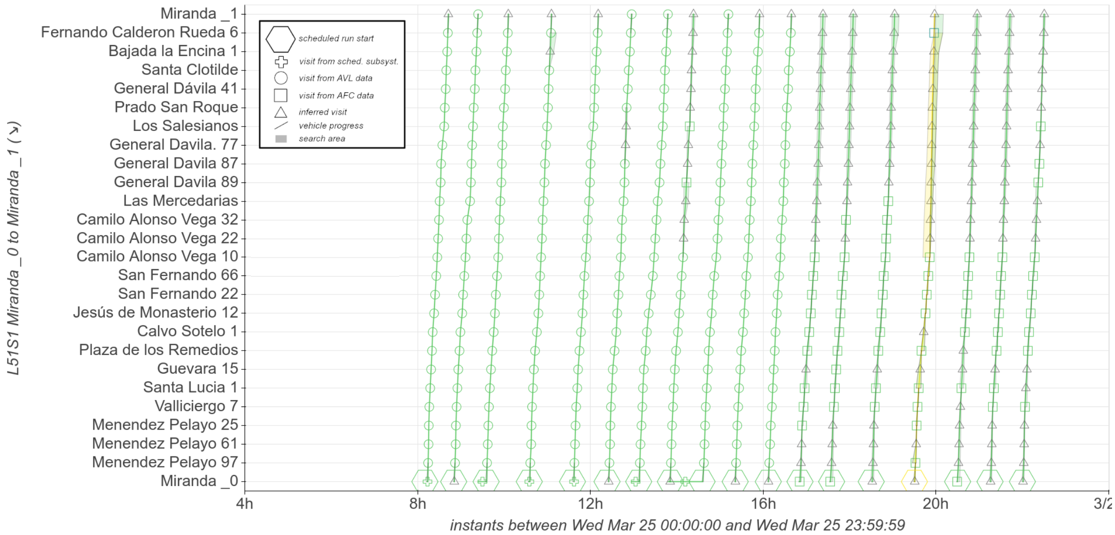
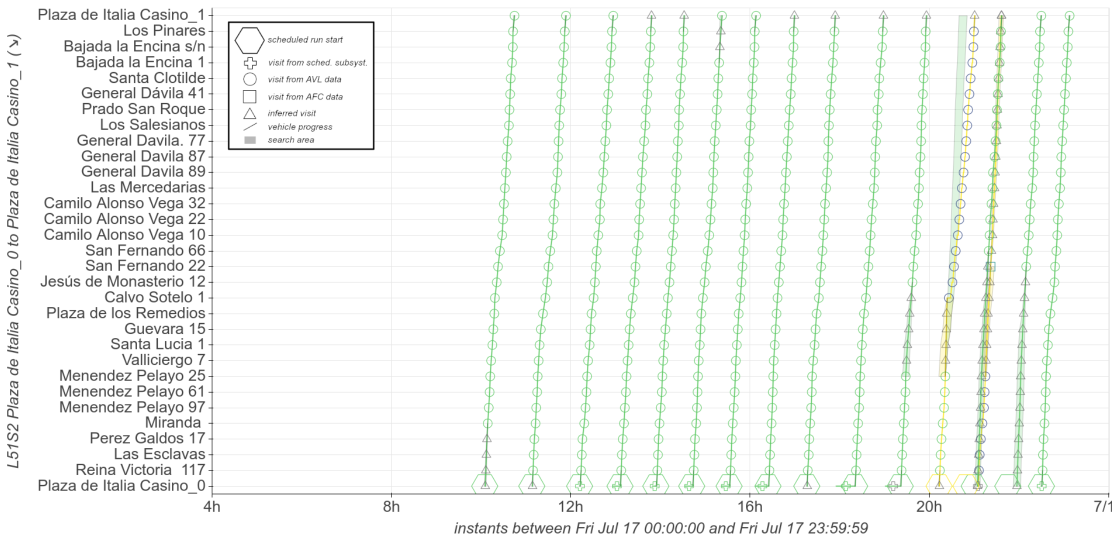


Figure 7.1: Runs of one vehicle following the only configuration of route 2

Figure 7.2: Runs of one vehicle on different days that show the configurations of 5C1. The changes from green to yellow correspond to vehicle id variations that were detected and treated.



(a) Configuration on workdays during school season



(b) Longer configuration for the rest of day types, extending up to Sardinero beach

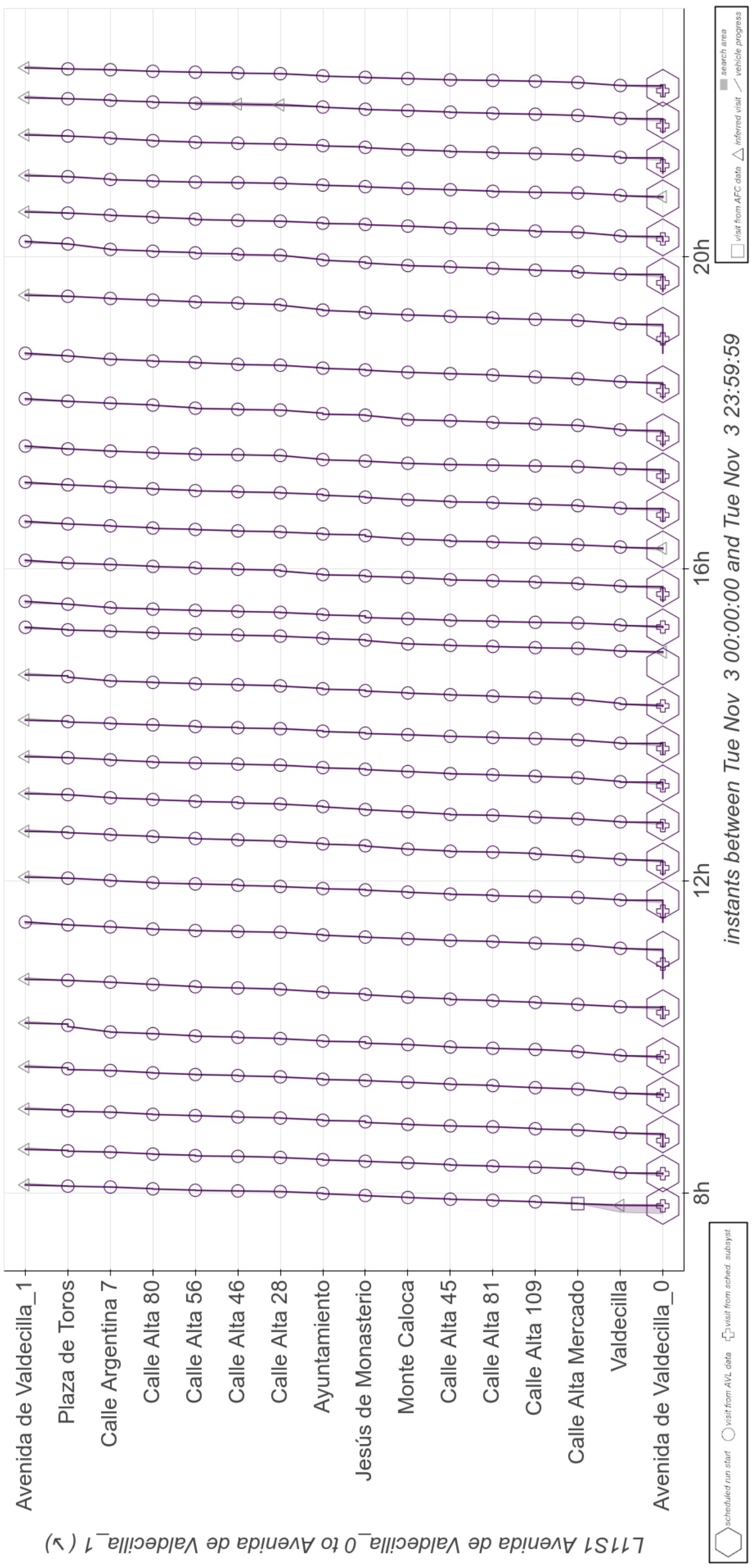


Figure 7.3: Runs of one vehicle following route 11

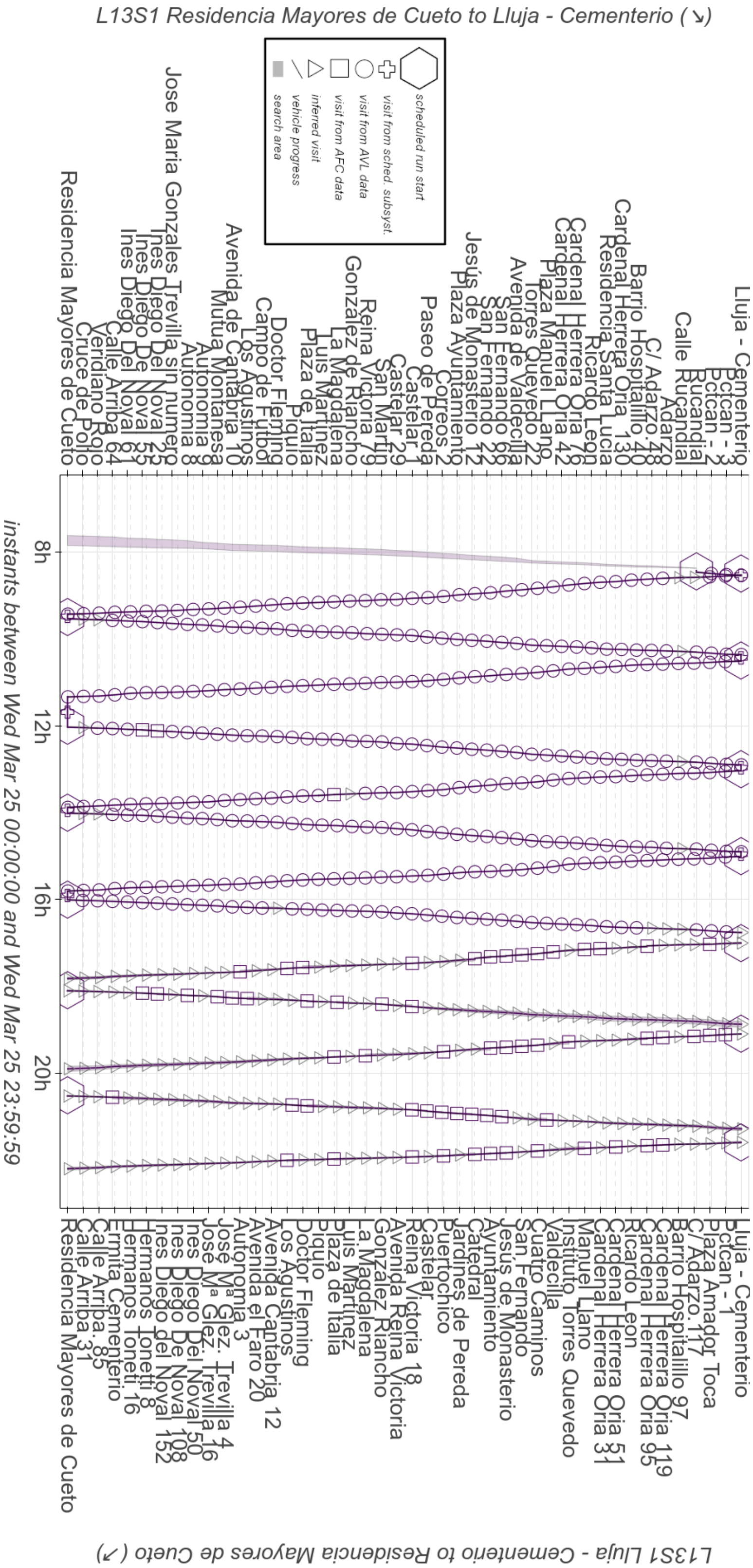
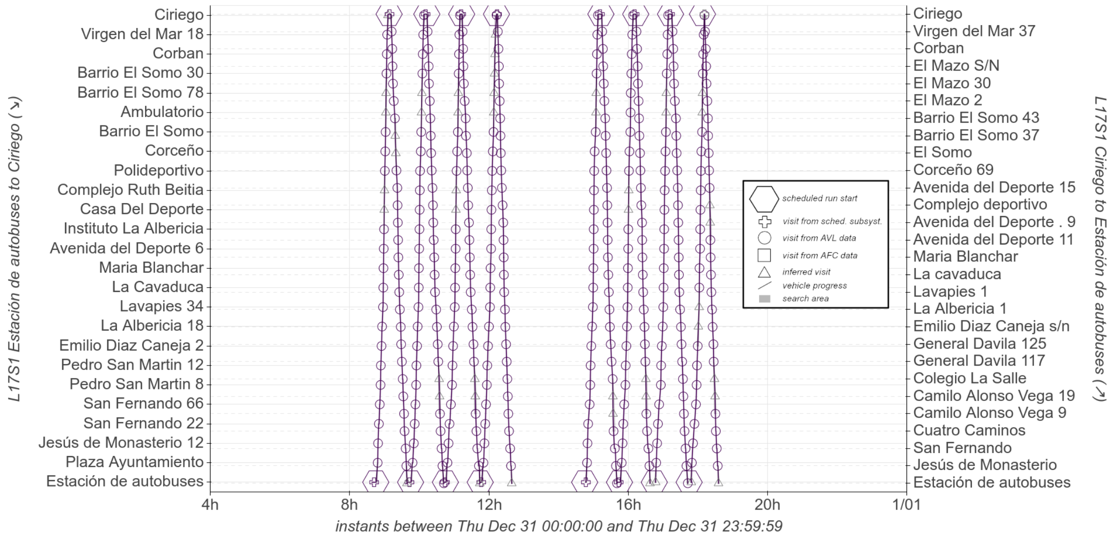
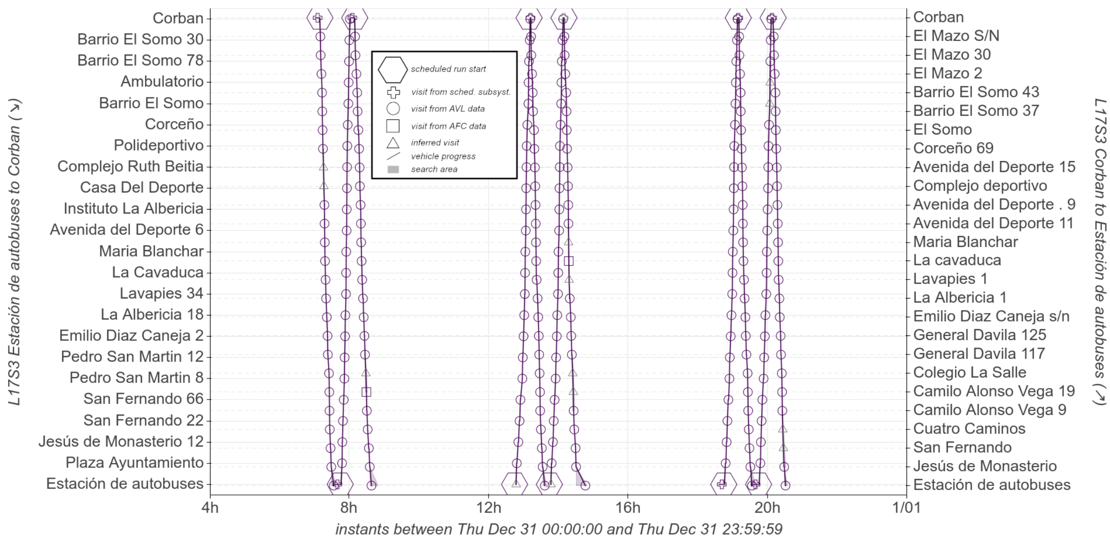


Figure 7.4: Runs of one vehicle following route 13

Figure 7.5: Runs of one vehicle covering Route 17 during a workday in school season

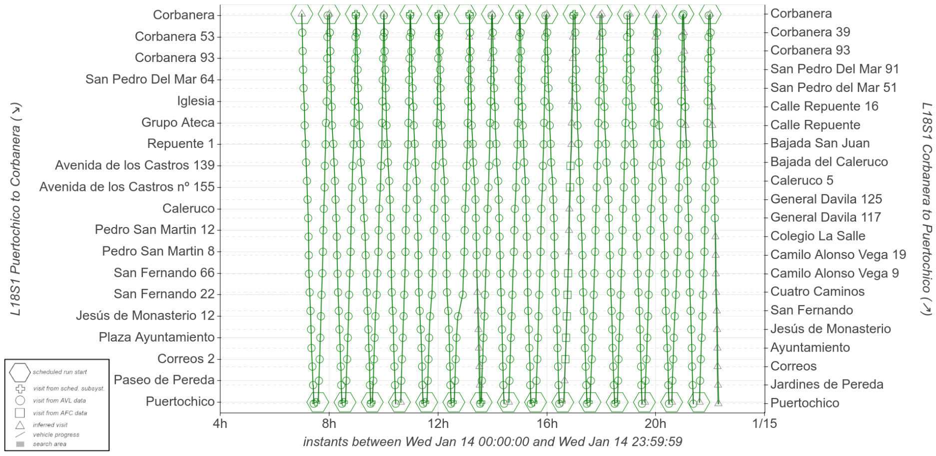


(a) Longer configuration, up to Ciriego

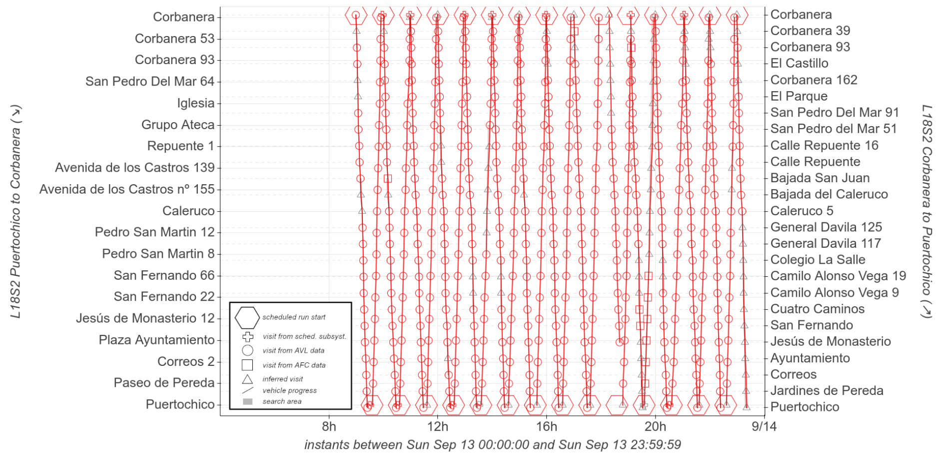


(b) Shorter configuration to Corbán

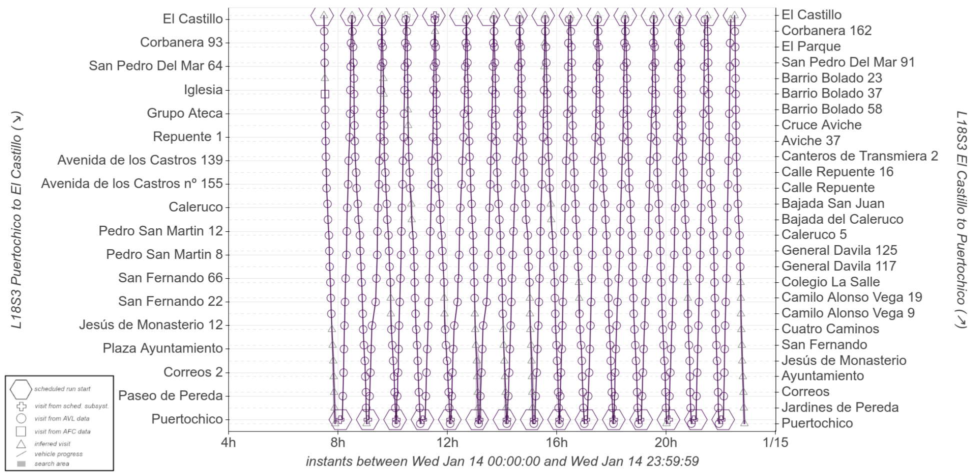
Figure 7.6: Daily runs of different vehicles that show the configurations of Route 18 to Monte neighborhood



(a) Configuration during a workday in school season, up to Corbanera



(b) Configuration during a workday in school season, up to El Castillo



(c) Summer configuration with some extra stops on the way to Monte

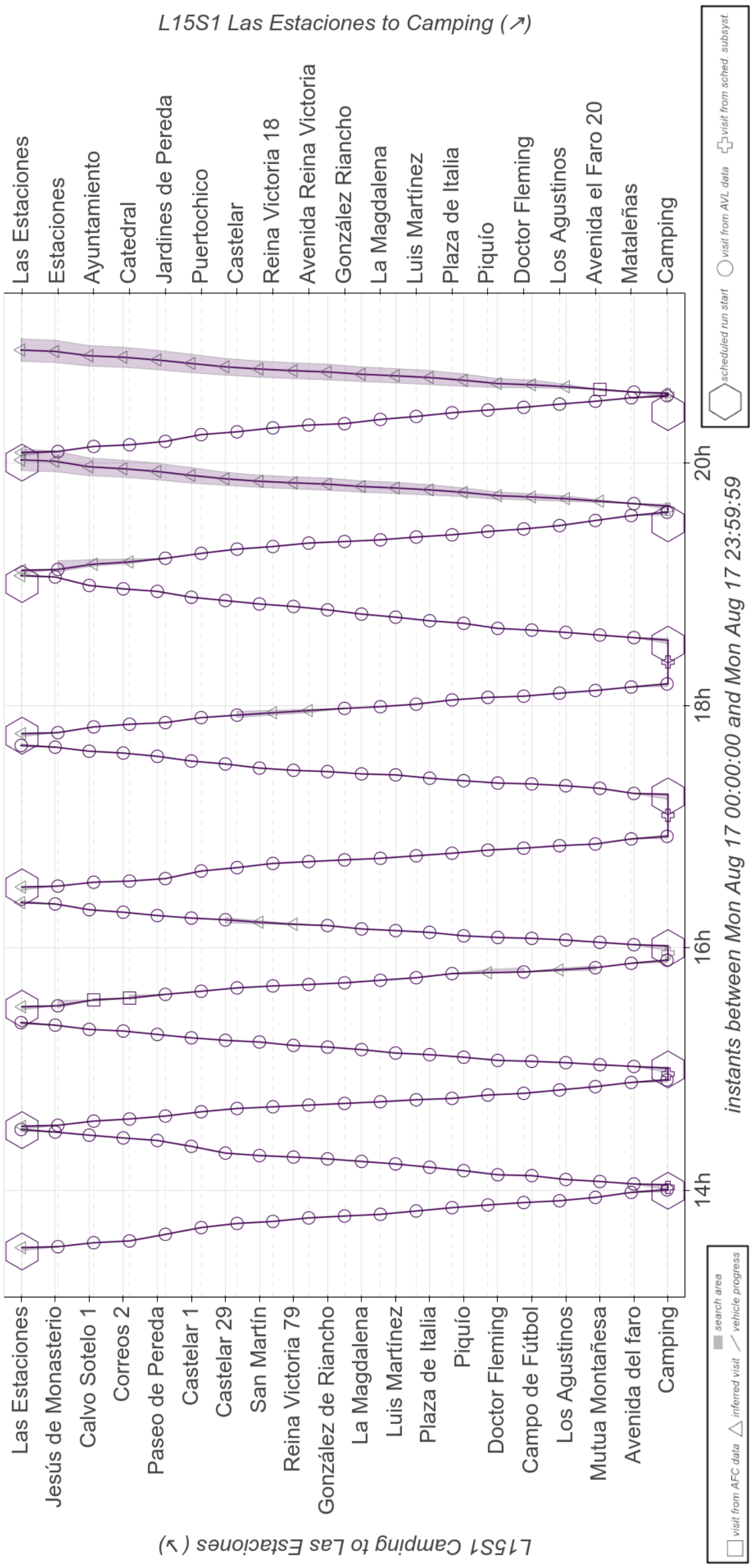


Figure 7.7: Runs of one vehicle following route 15.

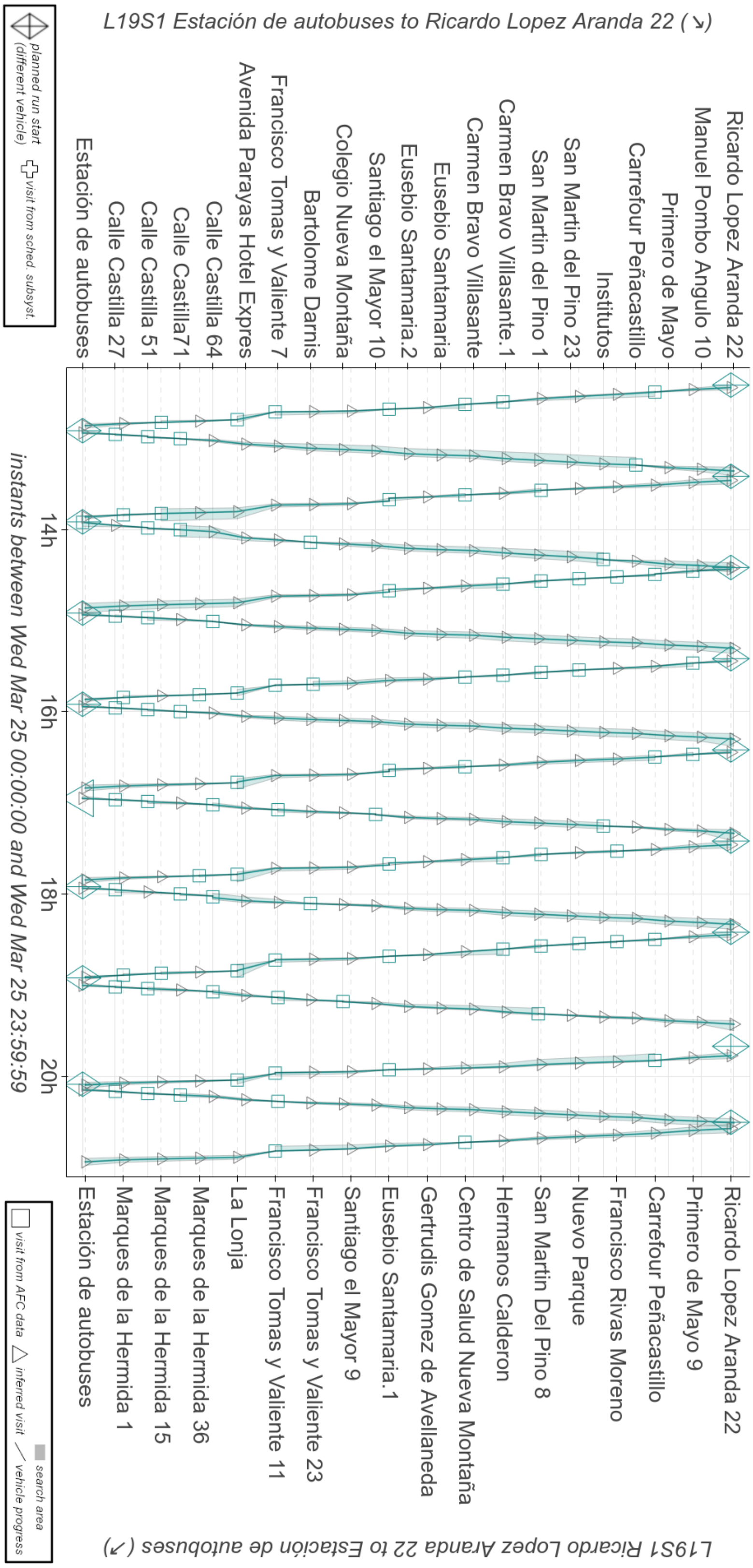


Figure 7.8: Runs of a vehicle (not the initially planned one) following route 19

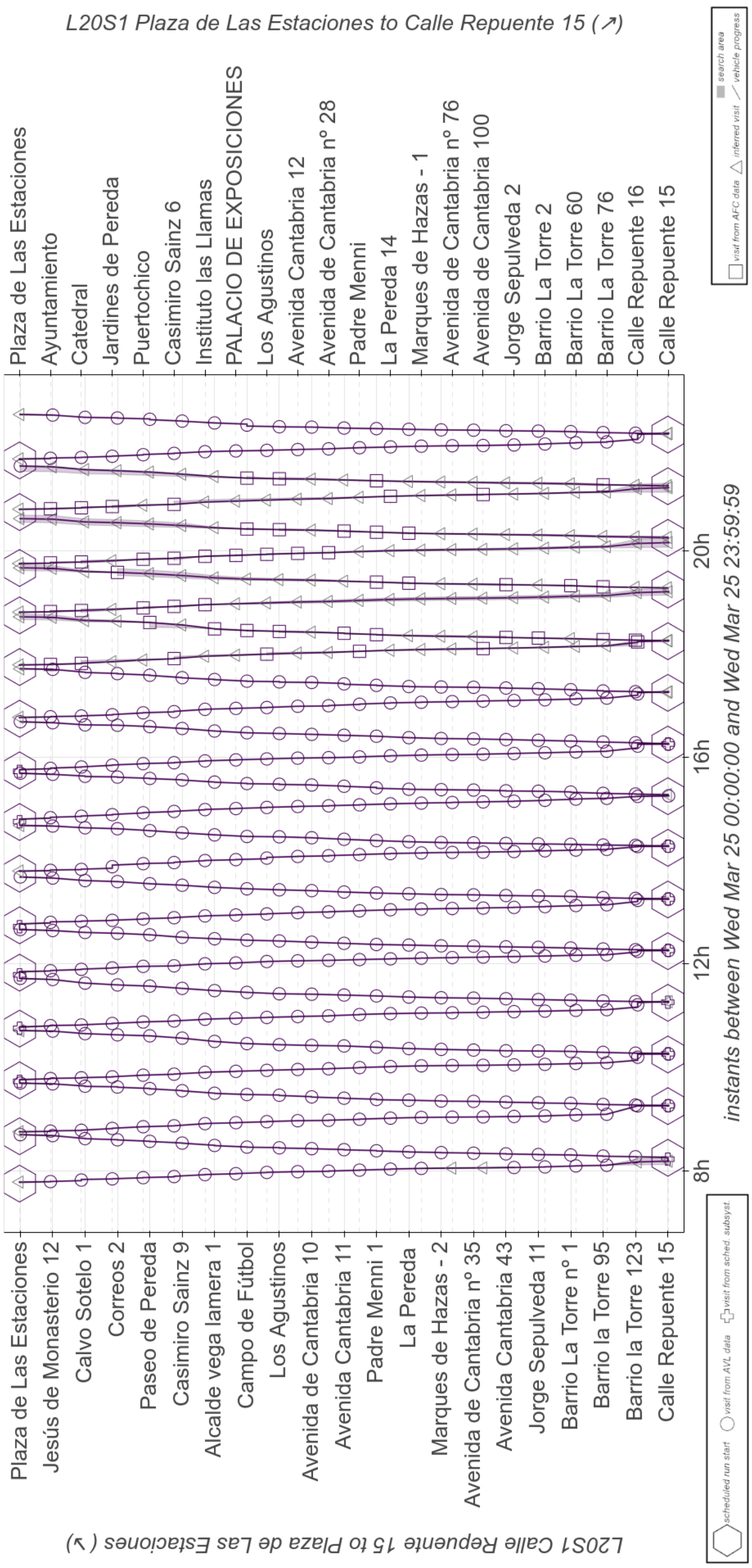


Figure 7.9: Runs of one vehicle following route 20

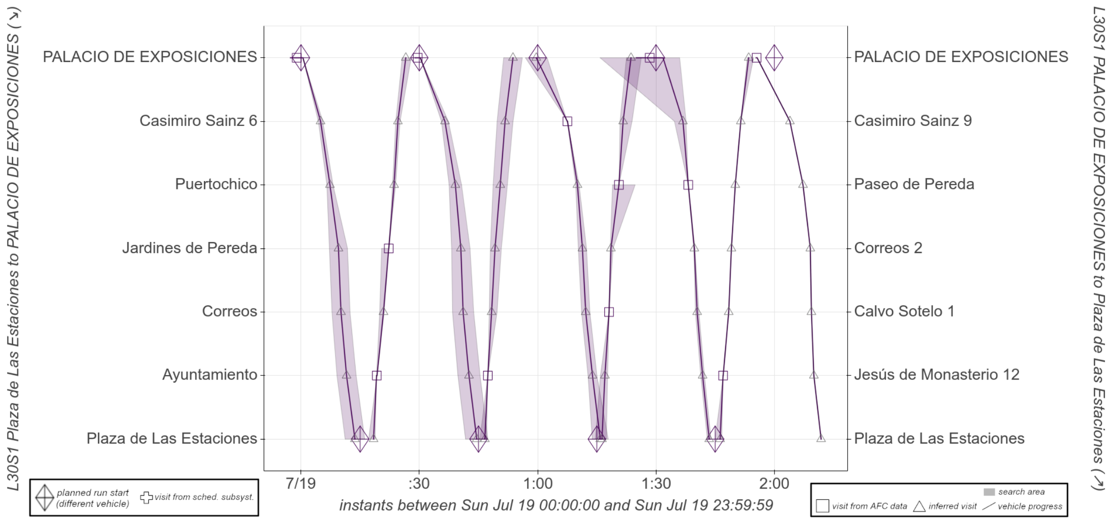


Figure 7.10: Runs of a vehicle (not the initially planned one) following route 30

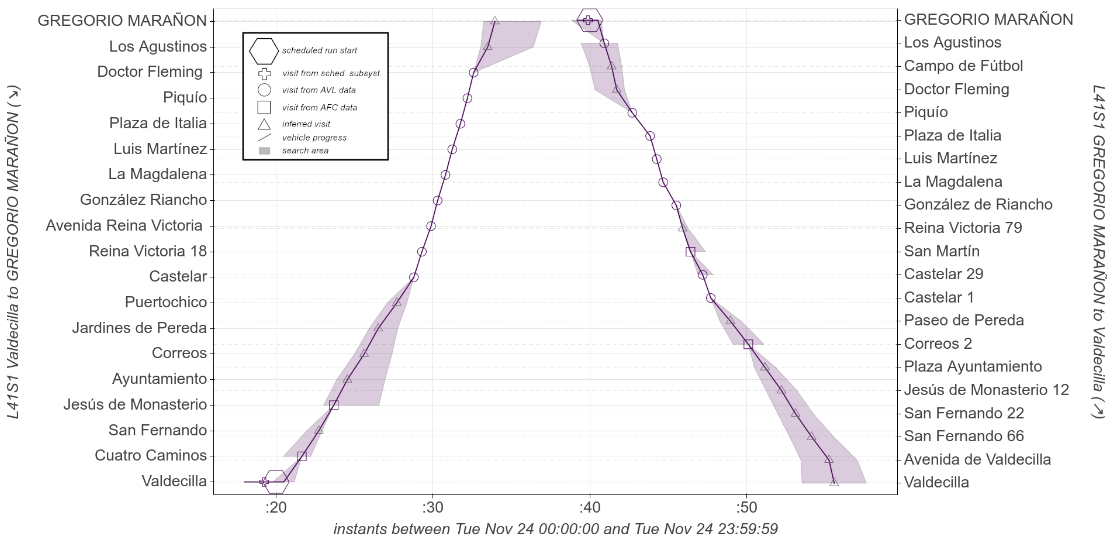


Figure 7.11: Runs of a vehicle of route 41 (special daily route to Valdecilla Hospital)

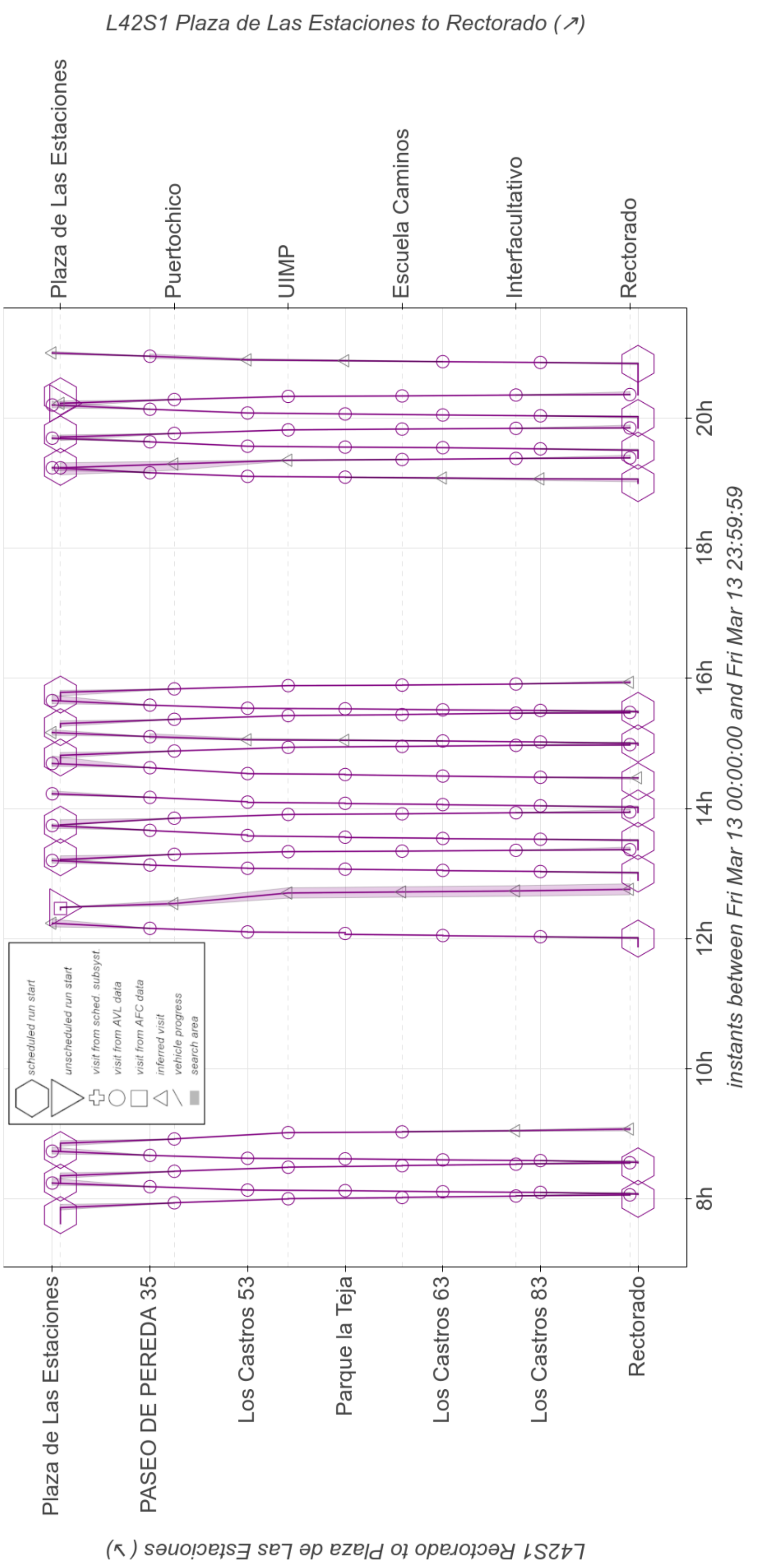


Figure 7.12: Runs of a vehicle of route 42 (designed to facilitate intermodality with regional transport)

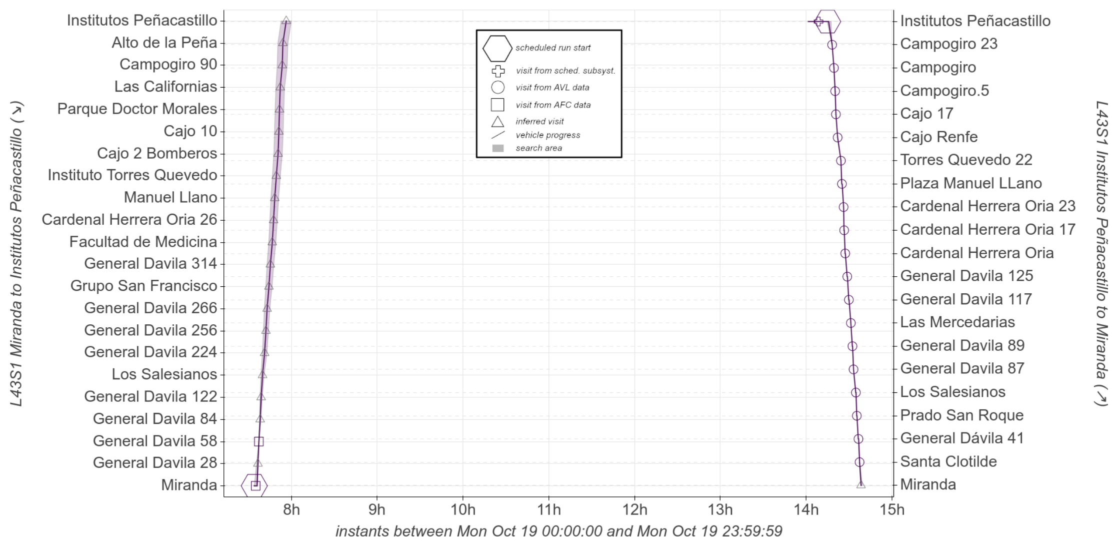


Figure 7.13: Runs of a vehicle of route 43 (special route to Peñacastillo secondary education center)

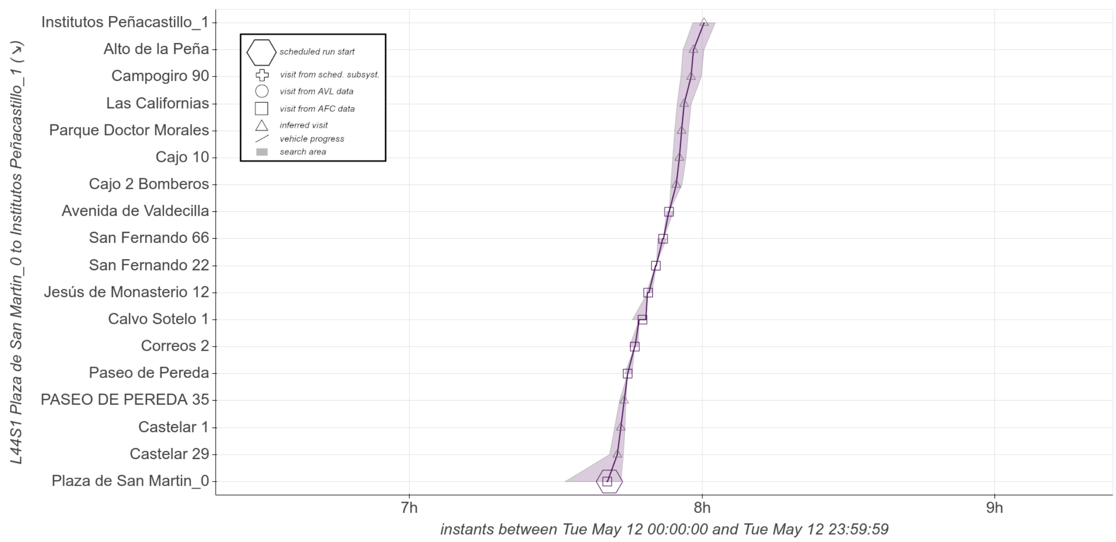


Figure 7.14: Runs of a vehicle of route 44 (another daily route to Peñacastillo)

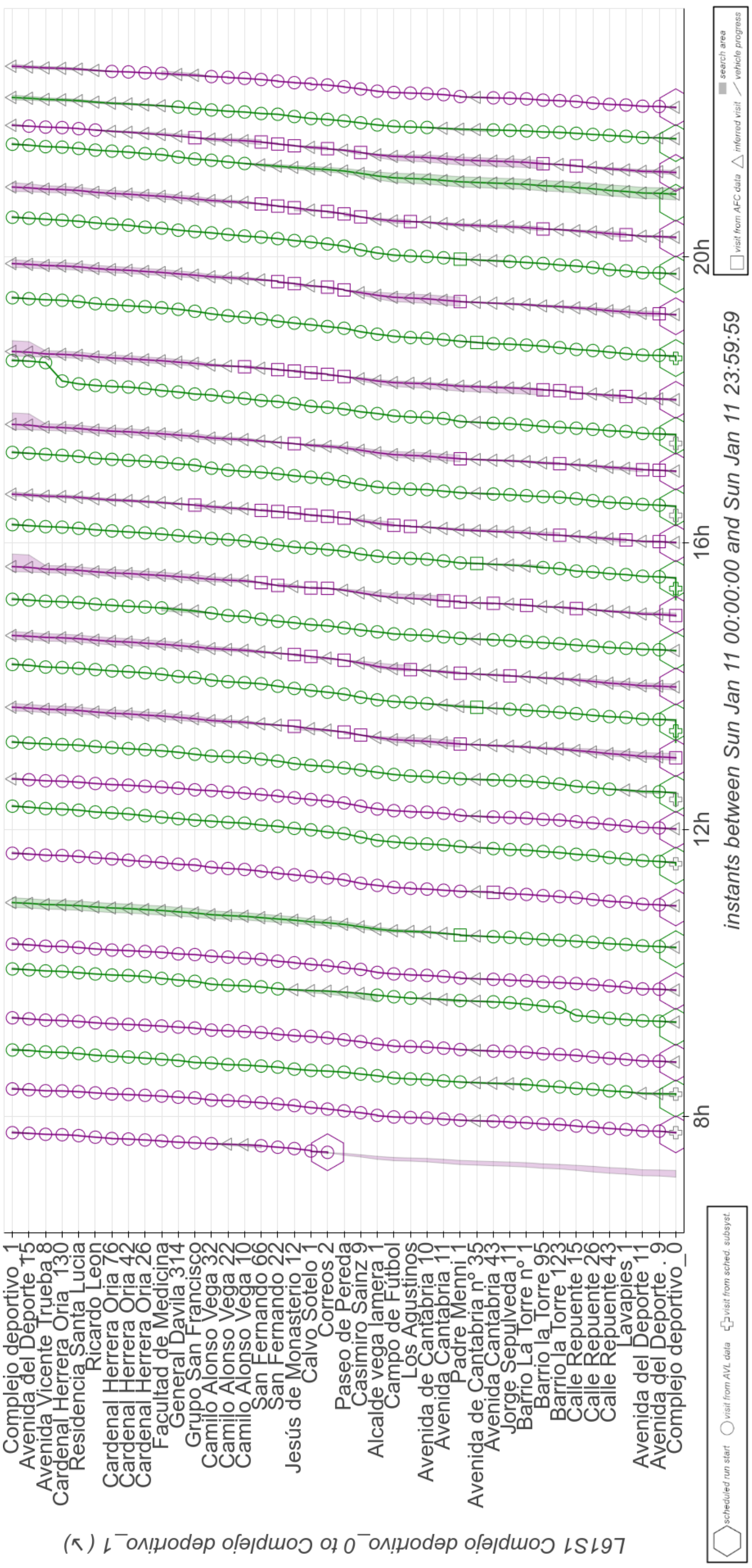


Figure 7.15: All runs of route 61 in a day (2 vehicles)

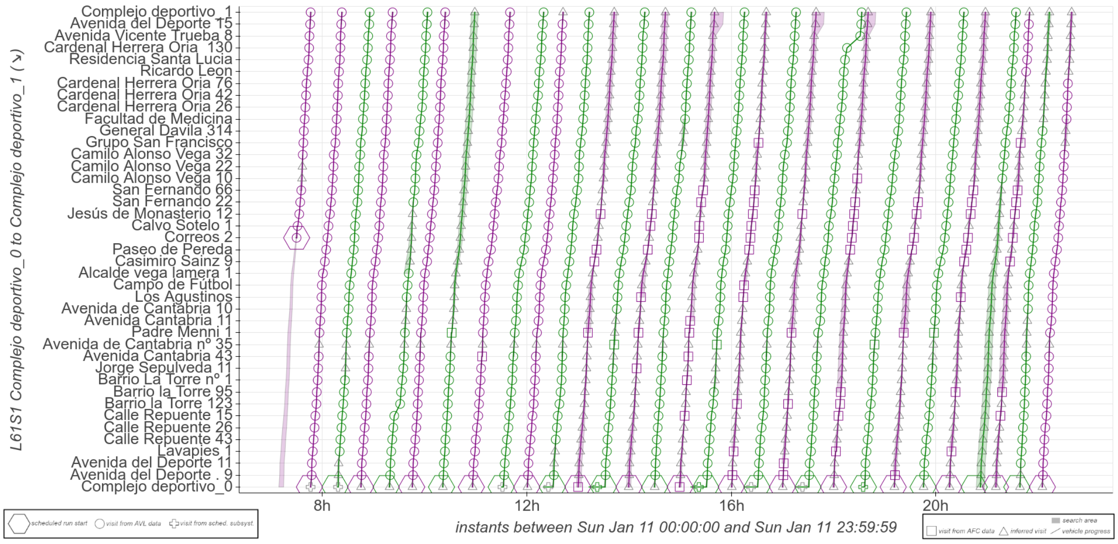


Figure 7.16: All runs of route 6C1 in a day (2 vehicles)

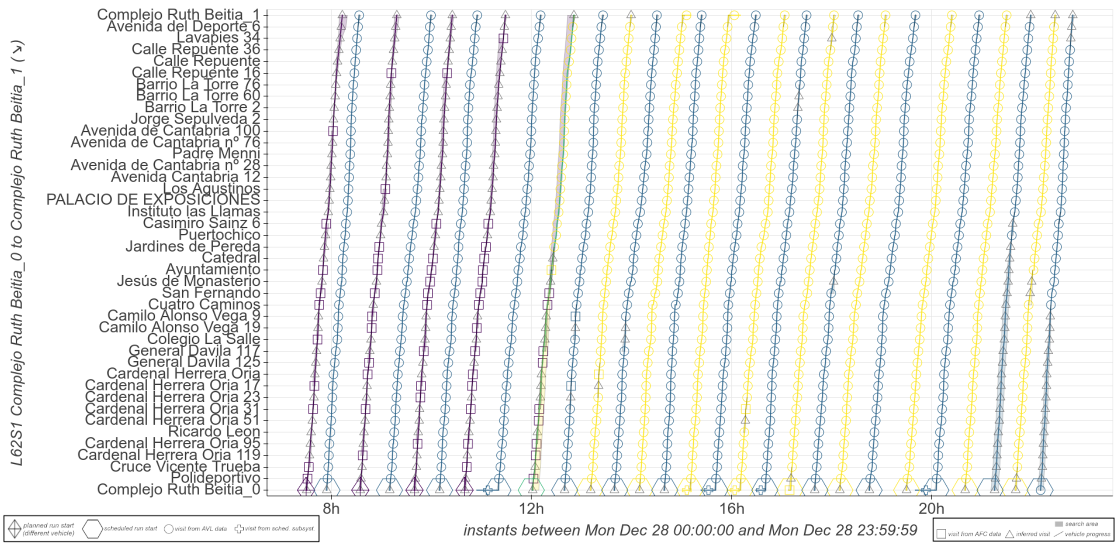


Figure 7.17: All runs of route 6C2 in a day (2 vehicles, one changes its id during the run which starts at 12:00)

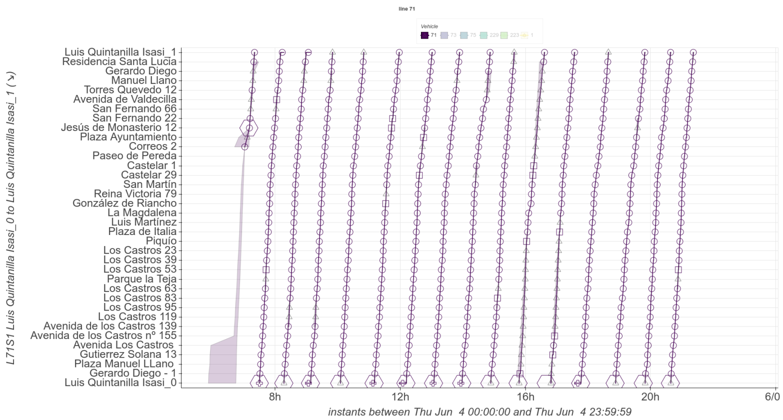
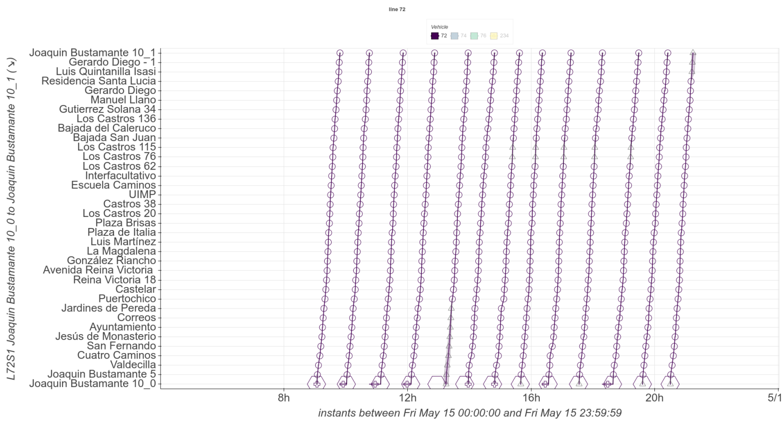
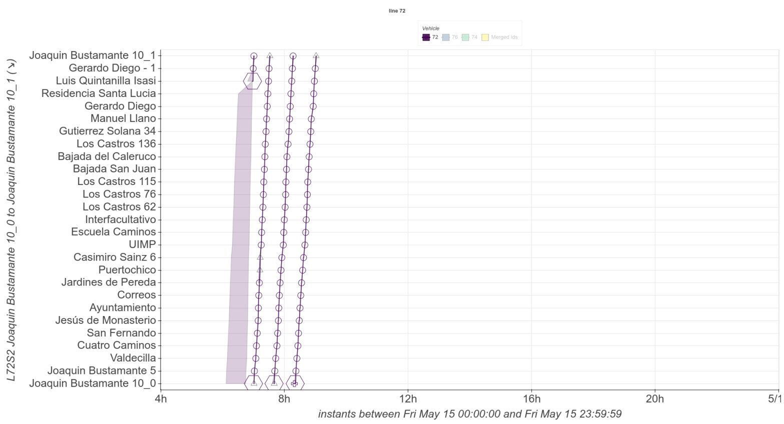


Figure 7.18: Runs of a vehicle of route 7C1



(a) 'Main' longer configuration



(b) Shorter early morning configuration. It skips some stops to arrive sooner to Los Castros educational hub.

Figure 7.19: Runs of a vehicle of Route 7C2 in a working day in school season

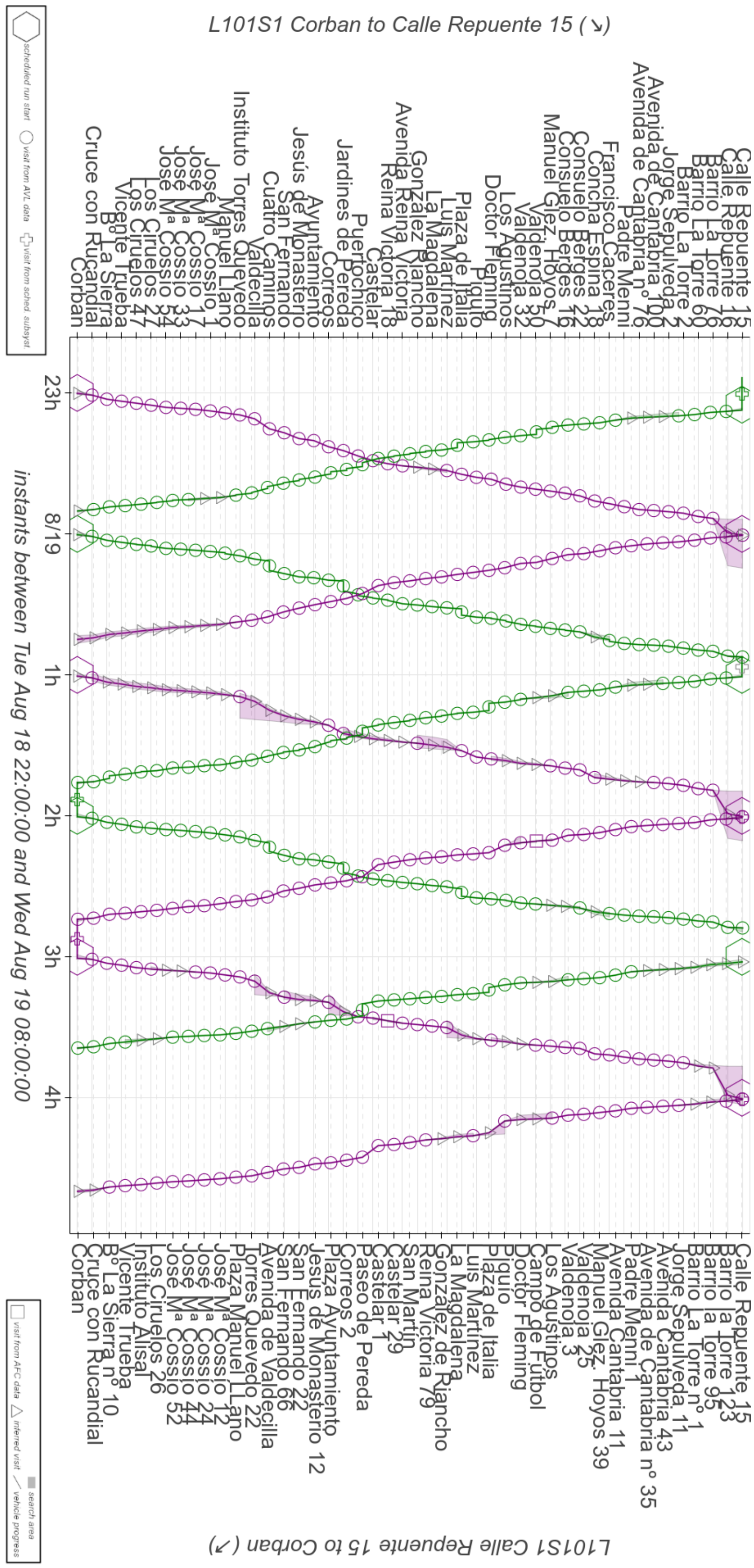


Figure 7.20: All runs of nightly route N1 in a day (2 vehicles)

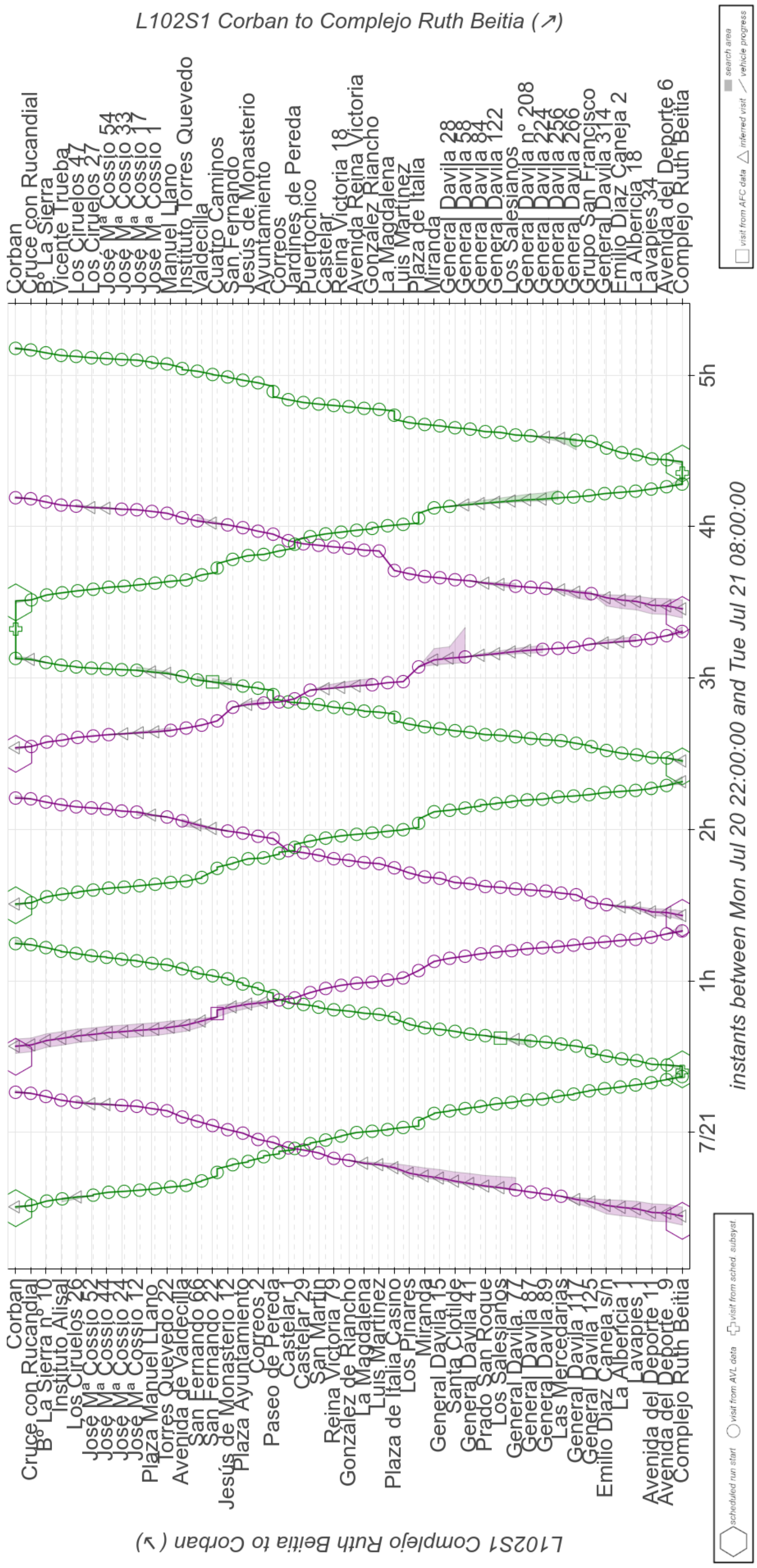


Figure 7.21: All runs of nightly route N2 in a day (2 vehicles)

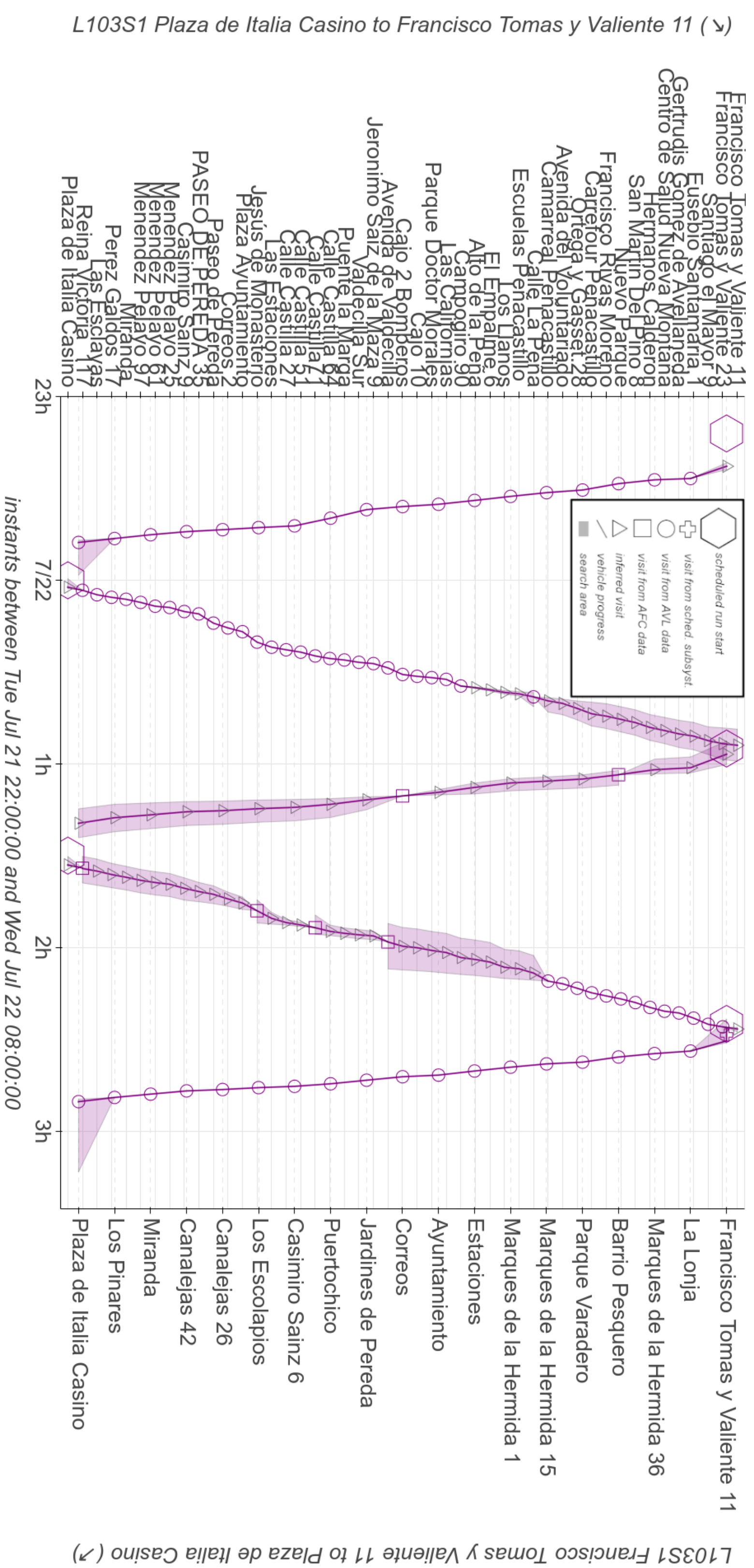


Figure 7.22: All runs of nightly route N3 in a day (1 vehicle)