Attuning Transport Supply to Low Demand Conditions using Microscopic Models

Acknowledgements

Writing a thesis is not a one man show. A lot of people have contributed to this thesis directly and indirectly. First of all, I want to thank my international jury members Dr. Elenna Dugundji (VU Amsterdam - Faculty of Science, Mathematics) and prof. Dr. Johan Holmgren (Malmö universitet) for the time and effort they spent reading my thesis and for the valuable feedback they gave to improve my thesis.

Four years ago, IMOB made it possible for me to start a PhD. Therefore, I want to thank my promoter prof. Dr. Davy Janssens and co-promoter prof. Dr. ir. Tom Bellemans for giving me this unique opportunity. Furthermore, my special gratitude goes to my internal committee of which the members are prof. Dr. Davy Janssens, prof. Dr. ir. Tom Bellemans, prof. Dr. ir. Bruno Kochan, prof. Dr. Geert Wets and Dr. ir. Luk Knapen for their critic view on my research and for the valuable comments during our six monthly PhD meetings. Every PhD meeting they came up with new questions and triggered me to conduct more research.

I owe a lot to my supervisor Dr. Luk Knapen. Thank you for the brainstorm sessions we did together regarding the research I was conducting, for being curious when I implemented something new or when I had a bug, for reading all my work and giving feedback on it. Furthermore, I learned a lot of new things about Linux, although I think I taught you some new things about LaTeX as well. Luk, thank you for the very nice cooperation!

During my stay at IMOB, I had the chance to cooperate with some international researchers. I want to thank Dr. Michał Maciejewski (Poznan University of Technology) for the cooperation in the SmartPT project. Thanks to Dr. Irith Ben-Arroyo Hartman (University of Haifa) for the cooperation with the DCC optimisation software. A special thanks goes to prof. Dr. Stéphane Galland (UBFC/UTBM). During my research we cooperated multiple times and I used his agent-based system SARL a lot. At specific periods in my research he acted as a co-supervisor. Hence, thanks for the interesting work sessions and for teaching me a lot of new things!

When I came to IMOB, probably a lot of colleagues thought I was a little weird. I used Linux as an operating system, I did not use Word for my text editing and they even did not see me use Powerpoint. Despite my different background, my colleagues accepted me very well. Hence, thanks to all my colleagues for the interesting conversations and the interest in what I did. I want to thank the colleagues of the secretariat for their help of all kinds.

Thanks to An, Jan and Wim, the "party committee" with whom I arranged birthday cards, decorated rooms and organised team building activities, for the cheerful conversations and breaks in the afternoon.

Thanks to my good friends Jan, Dimitri, Roy, Philippe and Anniek for the game nights, dinners, quizzes and other activities.

I want to thank my parents who always encouraged me to study hard and well. Furthermore, they gave me the opportunity to be able to study at UHasselt. Bram, thank you as well for being a nice brother. Of course, all other members of my family, thanks for being interested in what I did.

Very special thanks goes to my father, who coordinates everything which has something to do with the construction of our home, administratively as well as practically. Mandy and I appreciate your dedication very much. Thank you very much for your help!

My new colleagues at InfoFarm, thank you as well for being nice colleagues!

And of course I want to thank all other people who did a contribution to this thesis but which I forgot to mention!

I want to thank my girlfriend Mandy. The last few years, we did not choose the easy way. We decided to build a house and to get a child. Some people may think this is not a good idea during a PhD. Nevertheless, you were always there to support me. You helped me read my papers as well as this thesis. I can imagine it was not easy for you when I complained once more when some experiment did not go as expected or when I got some hard reviews on a paper. Thank you for being there when I needed it. My daughter Ellis Q., thank you for bringing a lot of joy to our family. I love you both.

Glenn Cich
November 2018

Contents

List of Figures

List of Tables

List of Abbreviations

ABM	**A**gent-**B**ased **M**odelling
AI	**A**rtificial **I**ntelligence
API	**A**pplication **P**rogramming **I**nterface
ATS	**A**dapted **T**ransport **S**ervices
CFP	**C**onstraint **F**easible **P**ath
CVRPTW	**C**apacitated **V**ehicle **R**outing **P**roblem with **T**ime **W**indows
DAG	**D**irected **A**cyclic **G**raph
DCC	**D**ay **C**are **C**entre
DRC	**D**emand **R**esponsive **C**onnector
DRT	**D**emand **R**esponsive **T**ransport
DVRP	**D**ynamic **V**ehicle **R**outing **P**roblem
FEE	**F**ixed **E**nd **E**arly
FEL	**F**ixed **E**nd **L**ate
FSE	**F**ixed **S**tart **E**arly
FSL	**F**ixed **S**tart **L**ate
FSM	**F**inite **S**tate **M**achine
FTS	**F**lexible **T**ransport **S**ystem
GDPR	**G**eneral **D**ata **P**rotection **R**egulation
GIS	**G**eographic **I**nformation **S**ystem
GPS	**G**lobal **P**ositioning **S**ystem
GTFS	**G**eneral **T**ransit **F**eed **S**pecification
GUI	**G**raphical **U**ser **I**nterface
ITS	**I**ntelligent **T**ransportation **S**ystems
JSON	**J**avaScript **O**bject **N**otation
LMS	**L**ess **M**obile **S**ervices
LP	**L**ogical **P**rocess
LWMSE	**L**east **W**eighted **M**ean **S**quared **E**rror
MATSim	**M**ulti-**A**gent **T**ransport **Sim**ulation
MDT	**M**aximum **D**etour **T**ime
MSE	**M**ean **S**quared **E**rror

NTL	**N**on **T**ransfer **L**ocations
OD	**O**rigin-**D**estination
OR	**O**perational **R**esearch
OSM	**O**pen**S**treet**M**ap
OTM	**O**perational **T**ravel **M**odel
OTP	**O**pen**T**rip**P**lanner
OVG	**O**nderzoek **V**erplaatsings**G**edrag (Flemish household travel survey)
PDES	**P**arallel **D**iscrete-**E**vent **S**imulation
PDOP	**P**osition **D**ilution **O**f **P**recision
PLM	**P**ersons with **L**imited **M**obility
POI	**P**oint **O**f **I**nterest
PT	**P**ublic **T**ransport
RFP	**R**equest **F**or **P**roposals
SBRT	**S**chool **B**us **R**ou**T**ing
SD-CFP	**S**emi-**D**isjoint **C**onstraint **F**easible **P**ath
SL	**S**ingle-**L**inkage
TAZ	**T**raffic **A**nalysis **Z**one
TDF	**T**ime **D**eviation **F**unction
TFTM	**T**hin **F**lows **T**ravel **M**odel
TFTM$_{Dem}$	**T**hin **F**lows **T**ravel **Dem**and **M**odel
TFTM$_{Sup}$	**T**hin **F**lows **T**ravel **Sup**ply **M**odel
TL	**T**ransfer **L**ocations
TSC	**T**rip **S**equence **C**omposer
TTB	**T**ime **T**able **B**ased
TWW	**T**ime **W**indow **W**idth
VRP	**V**ehicle **R**outing **P**roblem
XML	e**X**tensible **M**arkup **L**anguage

Chapter 1

Introduction

1.1 About the Doctoral Thesis

Since technology is always evolving, more and more fields are combined to conduct research. Computers are becoming faster and more powerful every year and hence, can be used as a tool to conduct research. In this thesis the field of Computer Science is combined with Transportation Science. This thesis is a bundle of three software tools each handling specific situations. It includes tools for *simulation*, *optimisation* and *data collection*.

Social exclusion can be a big problem for the society. This thesis will highlight some possible problematic cases and investigate possible consequences and solutions. Special attention is devoted to specific cases in Flanders, Belgium which include (i) the evolution from *basic mobility* to *basic accessibility*, (ii) the introduction of the *compensation decree* and (iii) reductions in *subsidies*. Although the main focus is on a local situation, all the tools can be easily used for other regions as well, after collecting the appropriate data.

In transportation there is always a *transport demand* and a *transport supply*. Transport demand is the term used for people who need transport services, while the transport supply describes all entities which provide transport. In this thesis the main focus is on the latter. Nevertheless, demand will be simulated as well, but will get less attention. In this thesis, the customers of these transport services share some characteristics. They cannot drive a car themselves and using *Public Transport (PT) is not always evident for them.* Reasons can be very divergent. It may have to do with a mobility impairment, with age, or due to lack of PT stops in the neighbourhood.

The more general outline of the thesis is as follows (more details will be given later on): Part I of the thesis introduces an agent-based *simulation tool* which is able to simulate a diversity of transport providers. Customers attempt to fulfil a schedule by arranging transport between different locations. Customers as well as transport providers will interact to negotiate about appropriate transport solutions. As the thesis is focused on transport providers, the goal of this simulation is to investigate the viability of the different transport providers with a given fleet.

Part II of the thesis discusses an *optimisation tool* to reduce the use of vehicles in very specific cases. It discusses cases were a set of commuters who are unable to drive need to commute to a common end point. In some cases those people are brought by their relatives in other cases they are brought by an organised (chartered) bus. By looking at the different constraints of both the relatives and the commuters, the optimisation tool will attempt to organise a carpool in order to use car capacities as efficiently as possible and to minimise the number of chartered buses as well as their driven kilometres. The goal of this optimisation tool is to find out how much can be saved by using carpooling.

Part III of the thesis covers a recurrent problem in simulations, namely *data*. The results of simulations depend on (i) the quality of the model and (ii) the quality of the data. In this part a semi-automatic travel diary collection tool is presented which can be used to collect data in the future.

Part IV gives an overview of possible future research and some improvements. It ends with a conclusion about the thesis.

1.2 Mobility Landscape in Flanders, Belgium

1.2.1 Basic Mobility and Basic Accessibility

Recently, there is a lot going on with the mobility landscape in Flanders, Belgium. In 2001, the Flemish Government approved a decree about basic mobility. *Basic mobility* makes sure that every citizen of Flanders who lives in a residential region has access to mobility. Hence, if someone does not have access to a car, he or she can use PT. It is clearly “supply driven”, the bus drives whether or not there is a demand at the time specified in the time-table. Specific rules about the location of PT stops, the frequency, the amplitude (the time window in which PT is provided), etc. were designed and written down based on the location (degree of urbanisation) and time (peak vs. off-peak times). However, this kind of service seemed to be quite expensive for the Flemish Government. Some PT lines are successful and well occupied, however some lines are scarcely occupied, which results in very inefficient use of bus capacities. For these kind of lines, the concept of “belbus” was introduced. This is a kind of *Demand Responsive Transport (DRT)*, in which a passenger should book a trip a day in advance. This bus will drive if and only if at least on passenger has ordered this transport. The bus will drive a predefined route and will stop at the PT stops which were ordered in advance. Since the Flemish PT providers have to economise and the costs of the belbus were very high for each trip, the belbus concept is being reduced. The Flemish PT provider now attempts to give other alternatives using regular PT. In some cases, this results in much longer travel times. As a result a (small) portion of the population possibly has problems taking PT.

However, the concept of basic mobility was quite unique, it was (and still is) very expensive for the Flemish Government. Besides the high expenses, the bus capacities are not used well, there were not many possibilities to promote "multimodality", such as combining PT trips with other transport means and at the end not all transport disadvantages were solved.

Previously mentioned disadvantages triggered a new concept of transport organisation, namely *basic accessibility*. With this concept, the Flemish Government wants to integrate all the available transport into one connected network where passengers can switch easily between different transport modes (called combi-mobility). The main idea of basic accessibility consists of reaching important places and in comparison with basic mobility it is demand driven instead of supply driven. The Flemish Government will introduce a *layered transport network*, which consists of four layers: (i) train net, (ii) core net, (iii) additional net and (iv) customised transport. The *train net* is the main backbone of the transport network, national as well as international and is provided by trains. The *core net* connects main cities and other important locations. The transport is arranged by buses, trams and metros. Furthermore, the *additional net* of which the transport is also arranged by buses connects smaller cities to the core net and/or the train net. It acts as a feeder for the other nets and provides peak hour services for home-work and home-school commuting trips. Finally, *customised transport* provides local transport supply. It consists of all kinds of transport initiatives to support low demand regions. The main goal again is to feed the core net [Mobile Flanders, 2018].

The previously mentioned measures are supported by the "decree to compensate the public service obligation of the transport of persons with a disability or seriously limited mobility" [Flemish Parliament, 2012] (in Flanders known as the "compensation decree"). This decree is applicable since December 2012 and makes sure that there is a mobility system which offers subsidised adapted transport for the whole of Flanders. The compensation decree introduces the *Adapted Transport Services (ATS)* (Dutch: Openbaar Aangepast Vervoer (OAV), previously also known as Dienst voor Aangepast Vervoer (DAV)). The ATS are operational in 27 service areas. In every service area, only one service provider is compensated. Those service providers can have a variety of vehicles, but they need at least one adapted vehicle. How much compensation a service provider gets depends on the trip distance and whether the customer is wheelchair bound. Customers can use ATS if they meet certain characteristics. Besides the ATS, there are also Less Mobile Services (LMS) (Dutch: Minder Mobielen Centrales (MMC)) which have even more strict rules. Only people having an income lower than two times the current minimum living wage which is dependent on the family status between €595.13 and €1,190.27 can use the LMS service [Programmed Federal Public Service for Social Integration, 2018]. Volunteers using their own cars are the drivers of a LMS. Hence the characteristics of ATS are more situation dependent; ATS providers have to decide for every customer whether they could use the compensation tariffs, based on different criteria such as their personal situation or even weather conditions. LMS providers on the other hand use the objective criteria of the wages. Currently, in all transport regions, transport providers with a compensation scheme have been appointed.

1.2.2 Pilot Studies

As mentioned before, basic accessibility is not yet operational in Flanders but will be introduced in the coming year(s). However pilot studies are currently in progress in Westhoek, Aalst, Mechelen and Antwerp. In these regions, they attempt to introduce the new basic accessibility concept instead of following the strict rules of basic mobility. In all other regions in Flanders, the concept of basic accessibility will start at the end of 2019.

The outcome of such pilot studies can be unpredictable. Some policy changes can give completely different outcomes than expected, resulting in high expenses. A possible solution can be *simulations*. By using simulations, a large number of scenarios and policy changes can be simulated and the results can be compared with each other. At the end, the scenarios with the best results can be tested in a pilot study. This thesis provides a simulation project including adapted transport.

1.3 Simulation Scopes

Simulations can be performed in different scopes in different kind of fields. For this thesis the focus is on the transportation simulations [May, 1989; Bellemans et al., 2002]. The well known scopes of simulations are (i) macroscopic, (ii) microscopic and (iii) mesoscopic. *Macroscopic* simulations [Cristiani and Sahu, 2016; Ngoduy and Wilson, 2013; Boel and Mihaylova, 2006] use mathematics to describe flows instead of really modelling every individual vehicle or person. Due to this low level of detail larger regions can be simulated efficiently. Impedance matrices are used to define travel times and hence different situations such as peak and off-peak can be simulated. The model uses aggregate variables for densities, speeds, etc.

Microscopic simulations [Derbel et al., 2018; Caprani et al., 2016; Li et al., 2017] are the most detailed simulations. They model each individual separately, this includes movements, interactions, etc. This simulation and its input data is limited in scope, since modelling individuals costs a lot of computational power. The advantage of this approach is that changes in behaviour of one individual can have a huge impact on the other individuals as well, hence it is possible to see how the system reacts if some individual properties change. Agent-based simulation with interactions between the different agents can also be seen as a microscopic simulation.

Mesoscopic simulations [Osorio and Selvam, 2015] use ideas from macroscopic as well as microscopic simulations. However, it will be not as detailed as microscopic simulations and it will not be as computationally efficient as macroscopic simulations. MATSim can be used as an example for both microscopic as mesoscopic. MATSim models every individual (= microscopic) on an aggregate level. The actual position of each vehicle on the link is not modelled (= mesoscopic); vehicles only occupy space on a network link.

Since the target group, which is faced in this thesis is only a small portion of the complete population, it was decided to use microscopic simulations. The use of microscopic simulations allows us to simulate the real world very detailed. Small changes such as more vehicles per provider or price increases for specific providers can have a huge impact on the individuals. Small changes in macroscopic simulation might get lost due to the high level of aggregations. Furthermore, the relatively small cases that will be handled are perfectly suitable for microscopic simulation.

1.4 Thin Flows: Basic Definitions

In this thesis, the main focus is on people who are part of *thin flows*. Thin flows are characterised by following definitions.

Consider a *network* containing *nodes* and uni-directional *links*. The network corresponds to a digraph $G(V, E)$.

Definition 1.4.1 (Connection). A *connection* is a set of paths (in the graph theoretical sense) from a given source node N_S to a given target node N_T.

The connection does not necessarily contain all possible paths linking N_S to N_T neither is it required that the nodes correspond to locations where activities can be performed; as a consequence, it is not required that any of the paths corresponds to a trip.

Definition 1.4.2 (Period Specifier). A *period specifier* is an expression that selects an infinite sequence of days using well known periods used in calendars. The well known periods are *day, month, year, century, etc.* The smallest resolution unit is a day.

Examples are: (i) every 26th of July, (ii) every first Monday of November, (iii) everyday, (iv) every Monday in July and August, (v) every New Year's Eve (vi) every week starting at the first Monday following July 11.

Definition 1.4.3 (Time-of-day Interval). A *time-of-day interval* specified by $\langle t_b, t_e \rangle$ where $t_b, t_e \in \mathbb{R} | (t_b \in [0, 24]) \wedge (t_e \in [0, 48[\wedge (0 \leq t_e - t_b < 24))$

The duration of a *time-of-day interval* never exceeds 24 hours. A *time-of-day interval* can contain midnight. Hence it can overlap at most two consecutive days.

Definition 1.4.4 (Period Sequence, Period Sequence Duration). A *period sequence* is defined by a triple $\langle I, S, E \rangle$ where I is a time-of-day interval, S is a period specifier and E is an era specified by $\langle D_0, D_1 \rangle$ where D_0 is the initial date and D_1 is the final date. The sequence consists of interval starting at t_b on each day $d \in (S \cap E) = (S \cap [D_0, D_1])$. The *duration* dur(I) of a periodic sequence I is the duration of the daily interval times the number of selected days: $\text{dur}(I) = (t_e - t_b) \cdot \text{nDays}(S, E) = (t_e - t_b) \cdot |S \cap [D_0, D_1]|$.

A periodic sequence is a finite sequence that does not contain any day that precedes D_0 and not any day that succeeds D_1. For concrete situations the period delimited by D_0 and D_1 shall be sufficiently large because stochastic phenomena are studied and sufficiently short in order to avoid including unwanted trends (e.g. the bus system will probably no longer be similar to the current one on July 26 2051).

Definition 1.4.5 (Passage). A *passage* $p(C, T)$ where C is a connection and T is a trip, is a subtrip $\overline{T} \subseteq T$ such that the path $P(\overline{T})$ belongs to connection C i.e. $(P(\overline{T}) \subseteq P(T) \wedge (P(\overline{T}) \in C)$.

Let $\mathcal{T}$ denote the set of all trips.

Definition 1.4.6 (Flow). A *flow f* identified by a pair $\langle C, K \rangle$ where C is a connection and K is a periodic sequence, is the set of passages $\{p(C, T) \mid T \in \mathcal{T}\}$ for which the intersection of K with at least one time period of moving on $p(C, T)$, is non-empty.

Note that $\langle N_S, N_T \rangle$ passages are considered, not trips or travellers in order to define flows. A particular traveller can perform more than one of the considered passages, even during a particular day. By definition, a *flow* is an finite set. Note also that it is not required that the passage period is contained in a period belonging to K: simple overlap no matter how short is sufficient.

Definition 1.4.7 (Mode Set Specific Flow). A *mode set specific flow* identified by a triple $\langle C, K, M\rangle$ where C is a connection, K is a periodic sequence and M is a set of travel modes is a flow f such that all modes used in the trips containing the passages defining f are in M, i.e. $\forall p(C,T) \in f \; : \; M(T) \subseteq M$ where $M(T)$ denotes the set of modes used in trip T.

Note that all modes for the trip T are used in the definition, not only the modes used in the passage $p(C,T) \subseteq T$.

Definition 1.4.8 (Flow Size, Flow Rate). The *size* s of a flow f is its cardinality: $s = |f|$. The *rate* of a flow is its size divided by the length of the interval: $r = s/\mathrm{dur}(K) = |f|/\mathrm{dur}(K)$

Hence, flow rates are expressed as a number of travellers per time unit. A flow rate is the expected value of a stochastic computed over a *period sequence*. Please refer to Figure 1.1.

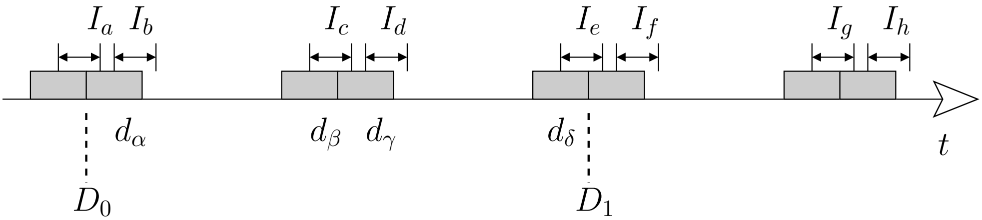


FIGURE 1.1: Periodic sequence $\langle I, S, E\rangle$: Shaded blocks represent days selected by the *period specifier* S. The $I_a, \ldots, I_h$ are instances of the *time-of-day interval* I. The days D_0 and D_1 are the earliest and latest days respectively in era E (that consists of days $d_\alpha, \ldots, d_\delta$). The *period sequence* consists of the intervals $I_b, \ldots, I_e$ i.e. the ones that start in a day selected by $\langle S, E\rangle$.

Definition 1.4.9 (Thin Flow). A *thin flow* f is a flow for which the flow rate does not exceed a given threshold.

Note that for a given era E, a given connection C and flows f_a and f_b defined by the respective period sequences K_a and K_b, none, either or both of f_a and f_b might be thin.

Examples of people whose trips contribute to particular thin flows are people with a disability, elderly people and people living in less dense areas.

The characteristics of thin flows clearly match the problems that were mentioned in Section 1.2. The reduction of subsidies and the economisation of PT providers in Flanders are causing problems for this target group of travellers, since they either have no choice for travelling any more, or the alternatives are very expensive and hence not affordable on the long term. Since basic accessibility is not yet implemented (only some pilot studies are currently conducted), simulations can be a possible intermediate step. Since simulation software is much cheaper and a lot of scenarios can be verified without any problem, they are very suitable to be considered as a pre-study.

1.5 DRT Concept Overview

Demand Responsive Transport (DRT) is a key concept within this thesis. It is a widely used concept and it can be found under other names as well, such as *demand responsive transit*, *flexible transport services*, *paratransit* or *dial-a-ride*.

1.5.1 Definition

Since it is such a broad concept, a lot of definitions are present in the literature [Davison et al., 2014; Bruni et al., 2014; Jain et al., 2017; Ryley et al., 2014; Chevrier et al., 2012]. However, a lot of similarities can be observed. DRT is (i) provided for everyone, however a lot of definitions state that the main target group are mobility impaired people and elderly people, (ii) operated with small vehicles, such as mini-buses or vans, (iii) not (necessarily) bound to a route and/or time table (it is demand driven), (iv) charged based on passengers and not per vehicle (as opposed to a shared taxis), (v) proposed as sustainable transport mode, (vi) a door-to-door solution and (vii) an intermediate form of public transport. A complete and comprehensive definition is given by Ellis [2009]: *DRT is handled by a provider, which provides transport by passenger cars, vans or small buses. Customers, e.g. (mobility impaired) people or other providers can call a DRT provider who then dispatches a vehicle to bring a customer from his origin to his destination. The characteristics of a DRT service is twofold: (i) vehicles do not serve a fixed route or a fixed time schedule, although this is possible on peak times and (ii) vehicles tend to pick up multiple customers at different locations (new requests can be handled in real time) before dropping them off at their destination.*

For operational and software purposes, it is important to keep the following aspects of transport in general in mind: (i) route, (ii) timing, (iii) passengers, (iv) drivers, (v) cost, (vi) vehicles and (vii) booking. For the route, there can be fixed routes, which means that a vehicle drives a predefined route even if there are no customers; an example is general PT. For the fixed routes, there can be some variants, such as little deviations of the routes or even completely flexible between predefined locations such as bus stops. Besides the fixed routes, there are also feeder services and door-to-door services. These kinds of services are used to pick-up customers either at a fixed position or at their home and bring them to transit hubs. From such transit hubs, they can continue their journey with public transport. Customers should agree on the pick-up time and arrival time [Lee and Savelsbergh, 2017; Li and Quadrifoglio, 2010]. Door-to-door services are organised in a way such that customers are picked-up and dropped of at their desired location. Drivers of such services offer help with boarding and alighting the vehicle. [Neven et al., 2014]. In a PT context, the timings are fixed by means of a time table. In other cases the time table becomes fixed from the first booking. An example of such service is the "De Belbus" in Flanders. A more flexible service is the one where bookings can be made and were the passenger gets a guaranteed departure and arrival window. An example of such service is a shared taxi where a customer is picked up while other customers are already in the vehicle. Finally, there are also the completely flexible services, such as taxis and door-to-door services. For the passengers, there are a lot of possibilities. In most of the cases everyone is allowed to use a service, such as a taxi or PT. For other services rules apply such as only people with a mobility impairment. In other cases there can be a closed group which means that only a specific group can use that service. An example of such service is a day care centre where the visitors are picked up every morning to go to the centre or a school bus transporting only children of a specific school. The driver can either be paid or he/she can be a volunteer. The price a customer has to pay can be (partly) subsidised or not. The drivers use vehicles which are adapted or not. Besides the adapted vehicles, drivers can help customers to board and alight the vehicle. Finally some services are bookable and others are not. For example taxis should be booked in advance, while PT is accessible without any booking.

In Part I, (DRT) transport services which do not have a fixed route, neither a fixed timing are investigated by means of a simulation. The simulation does not support multiple customers in the same vehicle. The customers can be, but not necessarily are mobility impaired. The services use both paid and volunteering drivers and some are subsidised and some are not. Since the software supports the transport of mobility impaired people, there are vehicles available that can transport people in a wheelchair for example. In the simulation, all the services should be booked in advance, except for PT such as buses and trains.

In part II, more attention is paid to the closed groups services. In this case a day care centre of mobility impaired people is investigated. An attempt is made to replace the chartered buses (paid transport without subsidising) by volunteering transport (nearly free transport).

1.5.2 Examples

There are a lot of initiatives based on DRT. However, only a few seem to be viable. A nice example can be found in Denmark. Currently, they have a DRT system called "FlexDanmark". FlexDanmark is a public company, owned by Danish public transport companies. It is responsible for the planning and coordination of DRT in Denmark, but it does not provide transport itself. The DRT concept in Denmark is organised as a *door-to-door* transport or *transit stop to door* service. There is no time table included. In order to make the coordination easier, Denmark is subdivided into regions and people can book DRT with an app. The main idea of this system is to use the available vehicles as efficiently as possible. This means that different people will be transported in the same vehicle at the same time. This means also that the trip will become cheaper if customers are more flexible in their needs [Mobile Flanders, 2016].

Another example is "My Bus" which is operating in Glasgow, Scotland. It is a bookable door-to-door service used in rural areas for transport such as social visits or shopping. The driver gives the customer assistance with boarding and alighting the vehicle. Trips should be booked at least two hours in advance, however it is advised to book a day in advance.

In Canada, there is a service called "Handi-Transit" which is a service of Winnipeg's public transit system. It offers door-to-door transport for people who are not able to use the traditional fixed PT in the city. Their mobility impairment is measured according to predefined criteria. Some rules apply for this kind of service, for example the origin and destination have to be within 500 metres of the fixed PT line. The vehicles are adapted for wheelchair and other mobility aid transportation. This service also includes assisting customer boarding and alighting the vehicle. Trips should be booked in advance and vehicles will be shared with other customers as well. Of course, the service will attempt to pick up and drop off every customer as close to their requested time intervals [Winnipeg Transit, 2018].

In Flanders, there is also a concept of DRT, namely "De Belbus". More information about this concept was already given in Section 1.2.1.

1.6 Motivations to Use Sampled Data

Since researchers always want larger and more realistic simulations, more and more data is needed. Furthermore, there is a well known saying "garbage in, is garbage out", which means nothing more than if bad data is used for simulations, bad results will be the output.

One particular example of mobility data are the so called "travel diaries" for which participants need to indicate their travel behaviour, such as start and end times of their activities and trips, the transport mode, etc. Before the smartphones era, participants had to provide the data using pencil and paper; researchers were supposed to manually convert this data. However, a lot of consistency problems were reported. Currently, more advanced techniques are developed.

In Part III a *prompted recall method* was developed. This method allows that participants only need to indicate what they did, but space and time information is collected by *Global Positioning System (GPS)* technology. The goal of this tool is to collect data in a reliable manner. This tool was not used to collect the travel diaries which were used in this thesis. However it was already successfully used for some projects such as ICOMFlex, in which change in travel behaviour when people could work at home or in satellite offices was investigated, iScape in which pollutant related feedback was given to participants and in a project about a new PT line in Tanzania. However, our experience is that it is not that easy to find participants for such projects. The main reason is the privacy issue, since people are tracked 24 hours a day, everything about their daily activities is revealed. The introduction of the *General Data Protection Regulation (GDPR)* in Europe makes it even more difficult to get permission to start such projects since quite sensitive personal data is dealt with.

Because of these privacy issues, it was decided to use sampled data for both Part I and Part II. But the reader should keep in mind that the proposed method in Part III could be used for both software tools. A lot of effort was spent to create the data samplers. The data samplers were fed with means, standard deviations extracted from surveys and synthetic populations and hence, are considered to be correct on an aggregate level. Samples of synthetic populations were extracted from the work of Neven et al. [2014]. More details about these samplers can be found in the corresponding parts.

1.7 Outline

This thesis consists of a bundling of several *peer-reviewed* papers both in proceedings as well as in journals. Note that the papers can slightly differ from the published ones: (i) terminology is made more consistent over the different papers, (ii) references to finished research have been added and (iii) language can be improved. As mentioned in Section 1.1, this thesis consists of four parts. A short overview is given, more details can be found in the corresponding chapters. Part I is the largest part and consists of five chapters. In Chapter 2, the preparation and the fusion of the *OpenStreetMap (OSM)* and the *General Transit Feed Specification (GTFS)* data sets needed for simulation are described. OSM is used for the network, while GTFS is used to simulate PT. The GTFS data set includes the location of the bus stops, the time tables of the trips, the routes etc.

In Chapter 3, an initial design for the agent-based software is discussed. Here the agent-based framework SARL which will be used for the microscopic simulations is introduced for the first time. A few important concepts will be highlighted as well.

In Chapter 4, an improvement in the agent-based framework SARL is discussed. Since SARL did not provide any synchronisation in simulated time, an implementation of a conservative synchronisation mechanism was proposed and implemented. This is needed because in the simulation, time needs to proceed minute per minute and communication between customers and providers needs to be chronologically correct.

Next in Chapter 5, a core feature of the simulation is discussed, namely the integration of any API. The possibility to integrate APIs reduces the work for the developer of the application. Complex algorithms such as a VRP solver or a PT router do not need to be implemented by the simulation developer, they only need to be connected with the application. As a consequence, it will reduce research and development time. In this chapter, the integration of the *Multi-Agent Transport Simulation (MATSim)* is discussed.

Chapter 6 combines the knowledge and techniques of this part to develop a working simulator and to run experiments. A sensitivity analysis will be conducted based on the providers' tariffs and fleet sizes. These different experiments will give insight in several measurements such as (i) the number of wanted trips a customer is unable to conduct, (ii) the total income and expenses of a provider and (iii) the total number of trip requests, rejections, acceptances and proposals in the interaction between customer and provider.

The final goal is to investigate whether transport providers in a DRT context are viable over a specific period. In order to fetch these results, customers and transport providers will discuss about requests and corresponding proposals. Providers will send a proposal when they are able to fulfil a request, but reject when it is not possible. As a consequence, in some cases customers will not be able to conduct a trip because no provider can fulfil that request.

Finally, in Chapter 7 a critical reflection about the presented simulation is given.

Part II consists of one chapter only. Chapter 8 describes the optimisation algorithms that are used to optimise transport organised by facilities using carpooling and DRT. Hence the transport includes (i) volunteers who are willing to pick up facility visitors and bring them to the facility or to common places and (ii) DRT providers. By using this carpool approach, possibly less facility visitors need to be picked up by DRT providers or the routes of the DRT providers are becoming more efficient, since facility visitors are brought together in a common location. This will result in less vehicles and hence less vehicle kilometres which results in a reduction of expenses. Given a set of facility visitors and drivers with corresponding constraints, the software will be able to find an optimal solution to cover their mobility needs. Facility visitors who are not part of a carpooling solution need to fall back on a "taxi-like" bus system.

Part III also consists of one chapter. Chapter 9 supports the collection of travel diaries of the thin flows population. Travel diaries are a summary of the activities and the corresponding trips of a person. They include the start and end times of the activity as well as the trip, the type of activity, the mode used for the trip etc. Since this is a lot of information to remember, a lot of mistakes are made by the respondents if data is recorded infrequently using pencil and paper. Therefore, a GPS tracking method is proposed. A stop detection algorithm is developed to automatically detect the locations where a respondent did an activity and the corresponding trip connecting the activities in time and space. The purpose is to collect the diary by *prompted recall* by allowing the traveller to annotate the detected stops. A sensitivity analysis is conducted and it is compared with an existing trip detection algorithm.

Part IV ends the thesis. Chapter 10 gives an overview of proposed future research. It also discusses some new material. Finally, in Chapter 11, a conclusion is drawn.

An overview of the contents and the relationships between the different parts can be seen in Figure 1.2. As already mentioned, the target group are people belonging

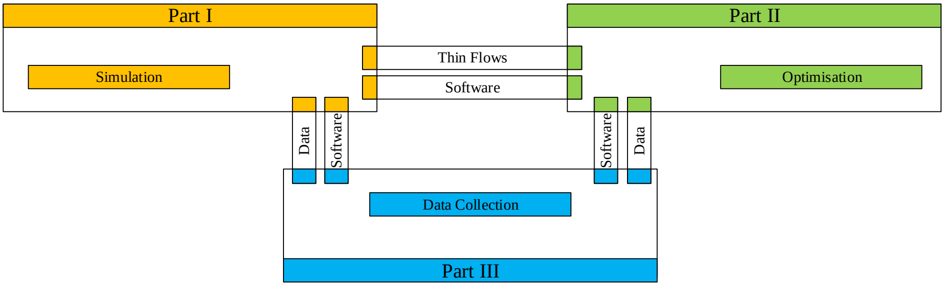


FIGURE 1.2: Overview of the different parts in the thesis together with the relationships between them.

to thin flows. For Part I this includes people with a disability and elderly people. For Part II, the participants can be children, but in our case study people with a mental or physical disability are dealt with.

Furthermore, Part I conducts simulations with a focus on the individual level, while Part II attempts to optimise the system as a whole. However both parts attempt to cut costs for expensive transportation means. In Part I, this is conducted with a partially subsidised service, while in Part II, the attempt to cut costs is conducted by means of volunteers serving a closed community.

Since detailed simulations and optimisations are used, detailed data is essential. The software in Part III can be used to collect this detailed data which might be used in Part I and Part II.

1.8 Research Questions and Contributions

1.8.1 Research Questions

In Part I, a microscopic simulator is developed to simulate travel services *demand* and *supply*. Since microscopic simulations need detailed and correct data, this data needs to be prepared. Three kinds of input data will be used: (i) OSM for the transportation network, (ii) GTFS for PT and (iii) a population for demand as well as supply. Since the population that will be modelled is very case specific, it is difficult to retrieve such data from questionnaires. OSM and GTFS are both open-source data sets. As a consequence, there is a large probability that the data is not as clean as required. During this research, all kinds of errors were found such as (i) missing values, (ii) erroneous values, (iii) missing links between data sets or (iv) even redundant data in geometries. In the literature [Haklay, 2010; Girres and Touya, 2010; Mooney and Corcoran, 2012; Barron et al., 2014; Haklay et al., 2010], most authors focus on (i) the reasons why there are a lot of issues with OSM and (ii) measuring the quality and completeness of OSM. Adeel [2017] describes a method in his Master's thesis to impute missing values in the OSM data set. The software package R and two different packages (i) Multivariate Imputation by Chained Equations (MICE) and (ii) Amelia were used. Both packages use a multiple imputation technique to impute missing values. The data set of London was used and different experiments were conducted. Results were promising, but more experiments regarding validation are needed. Issues related GTFS seem (to our knowledge) not to be present in the literature. This thesis is questioning, whether it is possible to

1. prepare this input data to be used in a microscopic simulator?

   (a) find anomalies, correct and compress data in OSM? (Chapter 2)
   (b) find anomalies, correct and compress data in GTFS? (Chapter 2)
   (c) sample thin flows customers and providers? (Chapter 6)

Martens [2018] discusses the need of an evolution of the current transport systems. The ageing population in many countries introduces an increase in the number of people with a mobility related impairment. About 6 % to 10 % of the population experiences a mobility related impairment. The need of an inclusive transport system is suggested. This inclusive transport system shares characteristics with basic accessibility as discussed in Section 1.2.1. Instead of implementing such policy changes in real cases, microscopic simulations can be a step in between [Ronald et al., 2015]. The main benefit is that a lot of scenarios can be modelled costless. The simulation in this thesis will include interactions between customers (demand) and transport providers (supply) because they need to negotiate about trip requests and proposals. Microscopic simulations and more specifically agent-based modelling could fit these requirements perfectly. Hence, is it possible to

2. use micro-simulations for simulating thin flows?
   (a) use agent-based modelling for simulating thin flows? (Chapters 3 and 6)

There are several agent-based frameworks available. For this thesis, it was decided to use the SARL framework [Galland et al., 2017]. It is an easy to use general-purpose agent-based framework. There is one drawback, namely it does not provide any synchronisation mechanism for simulated time. There is a need for modelling simulated time since multiple customers will request transport at different or the same transport provider(s). Hence, it is important that messages will be received in the correct order since transport should be requested, accepted/rejected and committed. Furthermore, functionality needs to be implemented to connect this agent-based framework with other external APIs such as OpenTripPlanner (to simulate PT). Hence, is it possible to

3. use this non-synchronised SARL agent-technology to simulate thin flows?
   (a) implement a (conservative) synchronisation approach in SARL? (Chapter 4)
   (b) integrate co-simulation in SARL? (Chapter 5)

When introducing a new policy, an important benchmark is the viability of the system [Ronald et al., 2015]. Neven et al. [2014] reported already viability issues with DRT in Flanders. Hence, the goal in this thesis is to model different situations to be able to answer the main research question: is it possible to

4. be viable (under specific circumstances) for transport provider within the compensation decree? (Chapter 6)

In Part II, another case study regarding thin flows is discussed. In this case study, facility visitors who are going to the same facility on a regular base attempt to carpool, but they cannot drive themselves. Such alternatives are needed since the cost of picking up all the facility visitors at their homes by a chartered bus is becoming too expensive. In the proposed solution, facility visitors are either brought directly to the facility, or to a common location, from which they are picked up by a chartered bus. By collecting visitors at a common location, the route of a charted bus might become more (cost) efficient. Visitors as well as drivers have very specific constraints regarding detour time, arrival and departure time window etc. Such data is again difficult to collect and in this case there are also privacy issues. Hence, is it possible to

5. create a synthetic population of thin flows facility visitors? (Chapter 8)

The idea of the presented algorithm is to organise carpooling among the different visitors; the cars used for carpooling are driven by volunteers (most of the time relatives of the visitors). The solution space of such algorithm can be (dependent on the number of visitors and drivers) extremely large. An exhaustive search method will work perfectly for a specific upper bound for the number of visitors and drivers. If the number of visitors and drivers is becoming too large, the exhaustive algorithm will probably not end (in a decent time span). For such situations, heuristics should be developed. Hence, is it possible to

6. develop an algorithm that solves this problem?
    (a) solve this problem by an exhaustive search method? (Chapter 8)
    (b) develop heuristics in order to be able to handle large cases? (Chapter 8)

The main goal is to reduce the cost, which is the total number of driven kilometres of the chartered buses. Hence, is it possible to

7. organise carpooling for (constrained) visitors going to the same facility and which are unable to drive? (Chapter 8)

8. reduce the number of chartered buses? (Chapter 8)

In part III the development and experimentation process of a TRIP/STOP detector is discussed. Travel diaries are a very useful source for transportation research. To reduce the burden of respondents and to increase the quality of the data [Raza et al., 2015] a (semi-)automatic tool to detect trips and stops from raw GPS traces is developed. In the literature, there is a distinction between different kinds of detectors. The first class uses car traces [Schönfelder et al., 2002; Du and Aultman-Hall, 2007; Guidotti et al., 2015]. These methods are based on the car engine to distinguish between trips and stops. A second class uses points of interests (POI) to find mainly stops [Alvares et al., 2007; Spinsanti et al., 2010]. The last category focuses on person traces which is also the focus of this thesis [Yan and Spaccapietra, 2009; Schüssler and Axhausen, 2009; Rasmussen et al., 2015; Tsui and Shalaby, 2006]. The first question is: is it possible to

9. extract annotated activity-travel data based on (person) GPS traces?
    (a) create a stop detector which is able to distinguish between trips and stops? (Chapter 9)

An always returning problem with (new) techniques is the possibility to validate results. For a TRIP/STOP detector this is a serious problem, since a fully annotated data set is the only way to validate the results. Another possibility is to create an algorithm which mimics a GPS trace and consequently the stop and trip times are known. Hence, is it possible to

10. validate the proposed algorithm(s)
    (a) by real data? (Chapter 9)
    (b) by sampled data? (Chapter 9)

If it is possible to find a technique to validate the results, it is also possible to use this technique to tune the input variables of the algorithm(s). By using different input variables and comparing the results it could be possible to get the best setting. Is it possible to

11. estimate the best settings for the proposed algorithm(s)? (Chapter 9)

1.8.2 Contributions

The contributions by this thesis consist of the methods and software tools mentioned in this section.

1. **Tool for OSM and GTFS data cleanup**

Microscopic simulations need detailed data and anomalies in data can be a huge problem. OSM contains both geometry information together with attribute values, but also landuse data. The cleaning conducted in this research focuses on the geometry and the attribute values. Microscopic simulations need detailed information about the speed, number of lanes etc. Exactly this information is in majority of the cases missing. The rule based algorithm in Chapter 2 is able (i) to first detect anomalies, (ii) to correct the data if possible and (iii) finally, if correction is impossible, to enrich the data by means of a rule based system and hence at the end solve the issue. OSM is a large data set, every point or link that can be removed without consequences is beneficial. In Chapter 2 several algorithms are developed to reduce the OSM data set by removing links with zero length and by removing points which do not have any geometric value. All those OSM features are bundled in a very user friendly toolkit together with other functionalities such as reading and writing OSM, extracting bounding boxes etc. (research question 1a).

The features developed for GTFS were mainly to support to work of Vuurstaek et al. [2018]. Functionality to clean the data and to remove objects that do not have any reference to other files were developed. Furthermore, there is functionality to remove stops from routes which seems to be wrong, such as bus stops that seemed to be used in both directions or bus stops that are used multiple times directly after each other. These functionalities are also explained in Chapter 2 and similar to OSM, this is bundled in a toolkit (research question 1b).

2. **Thin flows agent-based model**

The research question about the sampling of the population (research question 1c) is solved by using the data prepared and used by Neven et al. [2014] together with the Flemish household travel survey (OVG) of Flanders [Declercq et al., 2016] and the publicly available Flemish addresses database.

Since the outcome of policy changes are not easy to predict, simulation tools can be the solution. In Flanders, Belgium the official public bus operator is currently in a transition from the *basic mobility* to the *basic accessibility* concept. In Part I, the complete development process is explained to create a powerful and robust microscopic simulator to model demand and supply. The main advantage is that it can be used for all kind of scenarios integrating multiple kinds of demand and supply. The microscopic simulator is successfully developed in the SARL framework (research question 2a). The design is explained in Chapter 3, while the actual experiments are conducted in Chapter 6. Furthermore, a proof-of-concept version to integrate MATSim is presented in Chapter 5. This solves the research question about co-simulation (research question 3b).

3. **Conservative synchronisation mechanism for SARL**

The problem about a missing synchronisation mechanism is solved in Chapter 4 in which an extension is proposed that is able to provide a conservative synchronisation mechanism (research question 3a). This method is tested and verified and hence, ready for integration into the SARL framework.

4. **Viability testing of transport services within the compensation decree**

The question about the viability of a DRT system within the compensation decree context (research question 4) is answered by modelling different scenarios with the developed microscopic simulation. The simulation showed that a DRT system within the compensation decree could be viable under certain circumstances. This is explained in Chapter 6 and suggests a cooperation between normal DRT/taxis and adapted transport to increase the viability.

5. **Day care centre visitor transport optimiser**

In order to be able to test and validate the facility visitors problem, detailed data is needed. Since such data is difficult to get, again a sampling of the data is proposed (research question 5). Therefore, only mean and standard deviation for the home-to-facility distances were asked. This data together with the Flemish household travel survey (OVG) [Declercq et al., 2016] was used to create a population consisting of visitors and drivers. All other constraints regarding detour and time window width were given different values and modelled as a what-if analysis.

To solve the facility visitors problem, first an exhaustive search algorithm is developed. However, it uses some smart pruning steps in order to reduce the size of the search space. When the number of visitors and/or drivers becomes too large, the execution time of the exhaustive search algorithm becomes too long and hence two heuristics were developed (research questions 6a and 6b). Simulations showed that the heuristics came close to the optimal solution of the exhaustive search algorithm. Both algorithms used the power of graph theory. Hussain [2017] presented a carpool solution with the use of graph theory as well. He presents an advisory system for carpooling to the participating people. Although both problems seems to be very similar, they are not. Hussain [2017] attempts to give advice to people who want to carpool. Hence, his system attempts to find a *user optimum* for every individual. The core of the algorithm is finding *cliques* of small size (= vehicle capacity, hence most of the times three to four). Enumerating all those cliques is computationally feasible. Every clique constitutes a suggested carpool about which the clique members need to negotiate and agree. Every individual gets a number of best cliques as a solution. Organising those solutions has to be done by the individuals by phone or email. The system presented in this thesis is a *system optimum* since the goal is to reduce the driven kilometres as a whole. Individuals cannot choose from a set of solutions, since a choice has an influence on the complete solution (a driver is not available any more, detours are becoming to large etc.). This makes both problems completely different in both the used algorithms and time complexities.

The algorithms showed that organising a carpool with volunteers (even with a lot of constraints) is possible (research question 7). The solutions make it possible to reduce the cost with about 20 % to 30 % relative to the current pure chartered bus transport solution. (research question 8).

6. **Stop detection algorithm for prompted recall**

A collection of diaries with information about activity type, activity duration, transport mode and trip duration is a widely used data set in transportation. It can be used in all kinds of simulations. The stop detector which is presented in Chapter 9 has proven to be useful for projects (research question 9a).

7. **Sensitivity analysis and validation algorithm for stop detection algorithms**

In many cases it is difficult to compare the quality of different algorithms. The tool presented in Chapter 9 is able to give a particular score to a solution and hence makes it easier to compare the quality of different algorithms. Furthermore, the same technique can be used to tune input variables (research question 11). This validation technique can be used with real data (research question 10a); it was also applied to synthetic GPS traces having known distributions for the measurement error with the purpose of sensitivity analysis (research question 10b). Those synthetic trace generator could be also used to validate or test new algorithms.

Bibliography

A. Adeel. Imputation of Missing Values in OSM Networks. Master's thesis, UHasselt - Hasselt University, 2017.

L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macedo, B. Moelans, and A. Vaisman. A Model for Enriching Trajectories with Semantic Geographical Information. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '07, pages 22:1–22:8, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-914-2. doi: 10.1145/1341012.1341041.

C. Barron, P. Neis, and A. Zipf. A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, 18(6):877–895, 2014. doi: 10.1111/tgis.12073.

T. Bellemans, B. De Schutter, and B. De Moor. Models for Traffic Control. *Journal A*, 43(3-4):13–22, 2002.

R. Boel and L. Mihaylova. A Compositional Stochastic Model for Real Time Freeway Traffic Simulation. *Transportation Research Part B: Methodological*, 40(4):319–334, 2006. ISSN 0191-2615. doi: 10.1016/j.trb.2005.05.001.

M.E. Bruni, F. Guerriero, and P. Beraldi. Designing robust routes for demand-responsive transport systems. *Transportation Research Part E: Logistics and Transportation Review*, 70:1–16, 2014. ISSN 1366-5545. doi: 10.1016/j.tre.2014.06.002.

C. C. Caprani, E. J. OBrien, and A. Lipari. Long-Span Bridge Traffic Loading Based on Multi-Lane Traffic Micro-Simulation. *Engineering Structures*, 115:207–219, 2016. ISSN 0141-0296. doi: 10.1016/j.engstruct.2016.01.045.

R. Chevrier, A. Liefooghe, L. Jourdan, and C. Dhaenens. Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport. *Applied Soft Computing*, 12(4):1247–1258, 2012. ISSN 1568-4946. doi: 10.1016/j.asoc.2011.12.014.

E. Cristiani and S. Sahu. On the Micro-to-Macro Limit for First-Order Traffic Flow Models on Networks. *Networks & Heterogeneous Media*, 11:395–413, 2016. ISSN 1556-1801. doi: 10.3934/nhm.2016002.

L. Davison, M. Enoch, T. Ryley, M. Quddus, and C. Wang. A survey of demand responsive transport in great britain. *Transport Policy*, 31:47–54, 2014. ISSN 0967-070X. doi: 10.1016/j.tranpol.2013.11.004.

K. Declercq, D. Janssens, and G. Wets. Flemish Household Travel Survey 5.1 (2015–2016) Table Report. Technical report, UHasselt - Hasselt University, IMOB, Hasselt, Belgium, December 2016. (in Dutch).

O. Derbel, T. Peter, B. Mourllion, and M. Basset. Generalized Velocity-Density Model Based On Microscopic Traffic Simulation. *Transport*, 33(2):489–501, 2018.

J. Du and L. Aultman-Hall. Increasing the Accuracy of Trip Rate Information from Passive Multi-Day GPS Travel Datasets: Automatic Trip End Identification Issues. *Transportation Research Part A: Policy and Practice*, 41(3):220–232, 2007. ISSN 0965-8564. doi: 10.1016/j.tra.2006.05.001.

E. Ellis. *TCRP Report 136: Guidebook for Rural Demand-Response Transportation: Measuring, Assessing and Improving Performance*. Transportation Research Board, 2009. ISBN 9780309118071.

Flemish Parliament. Design of Decree to Compensate the Public Service Obligation of the Transport of Persons with a Disability or Seriously Limited Mobility. `http://docs.vlaamsparlement.be/pfile?id=1039564`, Dec 2012. Accessed: 2018-02-20 (in Dutch).

S. Galland, S. Rodriguez, and N. Gaud. Run-Time Environment for the SARL Agent-Programming Language: The Example of the Janus Platform. *International Journal on Future Generation Computer Systems*, October 2017. ISSN 0167-739X. doi: 10.1016/j.future.2017.10.020.

J.-F. Girres and G. Touya. Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS*, 14(4):435–459, 2010. doi: 10.1111/j.1467-9671.2010.01203.x.

R. Guidotti, R. Trasarti, and M. Nanni. TOSCA: Two-Steps Clustering Algorithm for Personal Locations Detection. In *23rd ACM SIGSPATIAL*, Seattle, Washington, November 2015. ACM. doi: 10.1145/2820783.2820818.

M. Haklay. How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets. *Environment and Planning B: Planning and Design*, 37(4):682–703, Aug 2010. ISSN 0265-8135. doi: 10.1068/b35097.

M. Haklay, S. Basiouka, V. Antoniou, and A. Ather. How Many Volunteers Does it Take to Map an Area Well? The Validity of Linus' Law to Volunteered Geographic Information. *The Cartographic Journal*, 47(4):315–322, Nov 2010.

I. Hussain. *Agent-based Simulation Model and Matching Support Framework for Carpooling*. PhD thesis, UHasselt - Hasselt University, 2017.

S. Jain, N. Ronald, R. Thompson, and S. Winter. Predicting susceptibility to use demand responsive transport using demographic and trip characteristics of the population. *Travel Behaviour and Society*, 6:44–56, 2017. ISSN 2214-367X. doi: https://doi.org/10.1016/j.tbs.2016.06.001. URL `http://www.sciencedirect.com/science/article/pii/S2214367X16300059`.

A. Lee and M. Savelsbergh. An extended demand responsive connector. *EURO Journal on Transportation and Logistics*, 6(1):25–50, Mar 2017. ISSN 2192-4384. doi: 10.1007/s13676-014-0060-6.

L. Li, B. Persaud, and A. Shalaby. Using Micro-Simulation to Investigate the Safety Impacts of Transit Design Alternatives at Signalized Intersections. *Accident Analysis & Prevention*, 100:123–132, 2017. ISSN 0001-4575. doi: 10.1016/j.aap.2016.12.019.

X. Li and L. Quadrifoglio. Feeder transit services: Choosing between fixed and demand responsive policy. *Transportation Research Part C: Emerging Technologies*, 18 (5):770 – 780, 2010. ISSN 0968-090X. doi: 10.1016/j.trc.2009.05.015. Applications of Advanced Technologies in Transportation: Selected papers from the 10th AATT Conference.

K. Martens. Ageing, Impairments and Travel: Priority Setting for an Inclusive Transport System. *Transport Policy*, 63:122–130, 2018. ISSN 0967-070X. doi: 10.1016/j.tranpol.2017.12.001.

A. D. May. *Traffic Flow Fundamentals*. Pearson, Upper Saddle River, NJ, 1 edition, 1989. ISBN 0-13-926072-2.

Mobile Flanders. Inspiration for demand responsive transport from Denmark. `https://www.mobielvlaanderen.be/overheden/artikel.php?nav=10&mbnr=174&id=1972`, Dec 2016. Accessed: 2018-10-10 (in Dutch).

Mobile Flanders. How does it work? `https://www.mobielvlaanderen.be/basisbereikbaarheid/hoe.php`, 2018. Accessed: 2018-06-18 (in Dutch).

P. Mooney and P. Corcoran. The Annotation Process in OpenStreetMap. *Transactions in GIS*, 16(4):561–579, 2012. doi: 10.1111/j.1467-9671.2012.01306.x.

A. Neven, K. Braekers, K. Declerq, T. Bellemans, D. Janssens, and G. Wets. Methodology to Optimize Resource Requirements of a Demand Responsive Transport System for Persons with Disabilities: A Case Study of Flanders. *TRB 93rd Annual Meeting Compendium of Papers*, 2014.

D. Ngoduy and R. E. Wilson. Multianticipative Nonlocal Macroscopic Traffic Model. *Computer-Aided Civil and Infrastructure Engineering*, 29(4):248–263, 2013. doi: 10.1111/mice.12035.

C. Osorio and K. K. Selvam. Solving Large-scale Urban Transportation Problems by Combining the Use of Multiple Traffic Simulation Models. *Transportation Research Procedia*, 6:272–284, 2015. ISSN 2352-1465. doi: 10.1016/j.trpro.2015.03.021. 4th International Symposium of Transport Simulation (ISTS'14) Selected Proceedings, Ajaccio, France, 1-4 June 2014.

Programmed Federal Public Service for Social Integration. (Equivalent) living wages. `https://www.mi-is.be/nl/equivalent-leefloon`, 2018. Accessed: 2018-06-11 (in Dutch).

T. K. Rasmussen, J. B. Ingvardson, K. Halldórsdóttir, and O. A. Nielsen. Improved Methods to Deduct Trip Legs and Mode from Travel Surveys using Wearable GPS Devices: A Case Study from the Greater Copenhagen Area. *Computers, Environment and Urban Systems*, 54:301–313, 2015. ISSN 0198-9715. doi: 10.1016/j.compenvurbsys.2015.04.001.

A. Raza, L. Knapen, K. Declercq, T. Bellemans, D. Janssens, and G. Wets. Diary Survey Quality Assessment using GPS Traces. *Procedia Computer Science*, 52:600–605, 2015. ISSN 1877-0509. doi: 10.1016/j.procs.2015.05.045. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015).

N. Ronald, R. Thompson, and S. Winter. Simulating Demand-responsive Transportation: A Review of Agent-based Approaches. *Transport Reviews*, 35(4):404–421, 2015. doi: 10.1080/01441647.2015.1017749.

T. J. Ryley, P. A. Stanley, M. P. Enoch, A. M. Zanni, and M. A. Quddus. Investigating the contribution of demand responsive transport to a sustainable local public transport system. *Research in Transportation Economics*, 48:364–372, 2014. ISSN 0739-8859. doi: 10.1016/j.retrec.2014.09.064.

S. Schönfelder, K. Axhausen, N. Antille, and M. Bierlaire. Exploring the Potentials of Automatically Colected GPS Data for Travel Behaviour Analysis: A Swedish Data Source. Technical report, Ecole Polytechnique Fédérale de Lausanne, 2002.

N. Schüssler and K. Axhausen. Processing Raw Data from Global Positioning Systems without Additional Information. *Transportation Research Record: Journal of the Transportation Research Board*, 2105:28–36, Dec 2009. ISSN 0361-1981. doi: 10.3141/2105-04.

L. Spinsanti, F. Celli, and C. Renso. Where You Stop Is Who You Are: Visited, 2010.

S. Tsui and A. Shalaby. Enhanced System for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1972(1):38–45, Jan 2006. ISSN 0361-1981. doi: 10.3141/1972-07.

J. Vuurstaek, G. Cich, L. Knapen, W. Ectors, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. GTFS bus stop mapping to the OSM network. *Future Generation Computer Systems*, 2018. ISSN 0167-739X. doi: 10.1016/j.future.2018.02.020.

Winnipeg Transit. Handi-Transit. `https://winnipegtransit.com/en/handi-transit/handi-transit/`, 2018. Accessed: 2018-10-11.

Z. Yan and S. Spaccapietra. Towards Semantic Trajectory Data Analysis: A Conceptual and Computational Approach. In *VLDB2009 (PhD Workshop)*, 2009.

Part I

A Simulation Tool for Thin Flows

Chapter 2

Data Preparation to Simulate Public Transport in Micro-Simulations Using OSM and GTFS

The peer-reviewed work presented in this chapter was published on May 12, 2016 in the Procedia Computer Science [Cich et al., 2016]. The work was presented at The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) in Madrid, Spain held on May 23-26, 2016. This paper was (besides the general use for micro-simulations) used as an initial step for the GTFS bus stop matching software in Vuurstaek et al. [2018]. The cleaned OSM as well as the GTFS bus stop matching is used in this thesis.

Keywords: OpenStreetMap; General Transit Feed Specification; Micro-Simulation; Data Cleaning; Public Transport

Abstract

Research on demand responsive collective transportation facilities that can act as feeder services to time table based public transportation (TTB PT) requires detailed and accurate information about the PT infrastructure, including the attachment of bus stops to the appropriate network link. Due to the size of the infrastructure, the data integration shall be automated. This paper describes the effort to prepare data from publicly available *OpenStreetMap* (OSM) and *General Transit Feed Specification* (GTFS) sources. Procedures are proposed (i) to build a network derived from OSM

suitable for simulations in transportation, (ii) to extract bus stops from GTFS and remove anomalies and (iii) to find candidate network links to attach them.

2.1 Problem Context

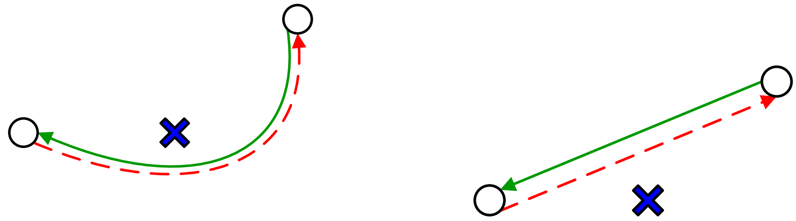
A *flow* in a transportation network is the set of passages from a source S to a target T using one of the available paths connecting S to T and for which the travel period overlaps a given set of periods (e.g. every Tuesday morning between 07:30h and 09:30h). *Thin flows* consist of small amounts of passages. Serving them by time table based public transportation (TTB PT) can be costly or lead to underused services. One of the research objectives of the Smart-PT project is to determine the viability of providers offering mini-bus based collective transportation on-demand as a replacement for TTB PT. Such services are expected to partly act as feeder services to TTB PT. The stochastic nature of the demand and the small capacity of the provided services require daily optimisation of the driven routes. The result is highly sensitive to small variations in the demand and in the characteristics of the local situation. Hence, an accurate high resolution representation of the TTB PT is required. Furthermore, the use of open and recent data is preferred. Therefore, an automated tool to integrate *OpenStreetMap* (*OSM*) derived transportation networks and *General Transit Feed Specification* (*GTFS*) is required. This paper discusses the first stage of an integration effort.

Figure 2.1 shows that geometrically complete and accurate data is required to attach a bus stop to the correct side of the road. On the other hand, the resulting network shall be as simple as possible because the algorithm that assigns bus stops to network links is based on combinatorial optimisation.

This paper is organised in the following way: in Section 2.2, an overview of the related work is given. Sections 2.3 and 2.4 are organised in the same way: first terminology is discussed, then the used algorithms followed by the results. Section 2.3 discusses OSM and Section 2.4 discusses GTFS. In Section 2.5, an algorithm is described that is able to find candidate locations for GTFS bus stops. Finally, a conclusion is drawn and future work is briefly discussed.

2.2 Related Work

Zilske et al. [2011] describe a process in order to use OSM in the *micro-simulator* MATSim. They mention issues related to getting high quality input data (in this case maps) to use in MATSim. Maps are in different (non-standard) formats, difficult to



(A) Situation with geometry: bus stop is connected to the (correct) solid green link.

(B) Situation where the geometry is replaced by a straight line segment: bus stop is connected to the (wrong) dashed red link.

FIGURE 2.1: Overview of possible problems due to geometry in OSM.

get and in most cases not referenced to each other. The authors converted an OSM data set to a MATSim compatible input format and attempted to integrate OSM and GTFS in order to simulate public transport.

In [Haklay, 2010; Girres and Touya, 2010], the authors describe methods in order to assess the quality of the OSM network. Eight quality indicators are discussed: (i) geometric/positional accuracy, (ii) attribute accuracy, (iii) completeness, (iv) logical consistency, (v) semantic accuracy, (vi) temporal accuracy, (vii) lineage and (viii) usage. The quality indicator which is of interest for this paper is the attribute accuracy. It describes the accuracy/correctness of the attributes in OSM. Haklay [2010] conducts an analysis of positional accuracy and the completeness of the OSM data set. In order to analyse this, the Ordnance Survey for the region of London, UK is used. Girres and Touya [2010] assess the attribute accuracy by studying the matching between lake names in the region of l'Alpes d'Huez. They noticed that only 55 % of the lake names are as informed as their base truth. However, when OSM describes a lake name, there is a nearly identical matching. Note that for such methods a base truth is required.

Mooney and Corcoran [2012] describe the annotation process in OSM. The main issue in the annotation process is the lack of discipline and automatic checking with respect to defining attribute names and with respect to assigning attributes to objects. Contributors can specify an unlimited number of *tag* elements and there are no context restrictions regarding the attribute values of those *tags*. The authors studied (i) the assignment of attribute values to *tag* elements, (ii) the type of contribution by the contributors and (iii) the use of the OSM Map Features page.

In [Barron et al., 2014], an attempt to assess the quality of OSM without the use of any base truth is conducted. They use the OSM-Full-History-Dump in order to assess the quality. They propose a framework consisting of a set of tools to assess the quality of several OSM characteristics. For example, they attempt to assess completeness of the road network by comparing the evolution in link lengths. When the length of the links stabilises, they assume that the links of that area are finished. In contrast, when the link lengths change a lot, it means that the links are not close to completion.

Our paper focuses on data cleaning as a prerequisite in the process of OSM-GTFS integration.

2.3 OpenStreetMap

2.3.1 Terminology

In order to conduct the cleaning and preparation steps, the OSM data (which is described in XML) is read into Java classes. It is assumed that the reader is familiar with the OSM terminology. The created Java classes are (i) *Point*, (ii) *Link*, (iii) *TransportInfrastructure* and (iv) *Road*.

Definition 2.3.1. A *Point* represents a node in the OSM data set. It merely defines the shape of a Link.

Definition 2.3.2. A *Link* represents a link which has no one-on-one mapping with the OSM data set. A Link consists of n Points where $n \geq 2$, because every Link needs at least a start and end Point. Links can only meet each other in the start and/or end Points, which means that a Link does not have any junctions in the intermediate Points that serve to define the road geometry.

Definition 2.3.3. A *TransportInfrastructure* is the base class for a Road. It holds a hash map of attributes (key-value pairs) and a list of Links. A TransportInfrastructure consists of n Links where $n \geq 1$. The hash map of attributes represents the attribute values of tags in the OSM data set.

Definition 2.3.4. A *Road* is a class, extended of TransportInfrastructure, which represents a way with the tag “highway” in the OSM data set. While a TransportInfrastructure has general functionality, a Road has specific functionality, such as information about the number of lanes, the speed limits, direction, etc.

A representation of the definitions can be seen in Figure 2.2. Note that it is possible to extend this software for other transportation infrastructures such as railways.

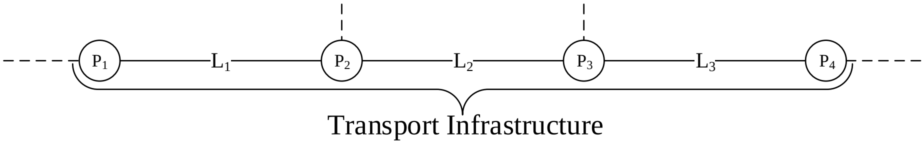


FIGURE 2.2: A representation of the terminology, with P_* the Points and L_* the Links of the presented TransportInfrastructure.

2.3.2 Algorithms

In order to combine the OSM data with a GTFS data set, the OSM data will be *cleaned, reduced* and *prepared*.

Definition 2.3.5 (cleaning, reduction, preparation). *Cleaning* is the process in which OSM objects are *enriched, corrected* and *completed*. *Reduction* is the process in which OSM objects that are not useful for any purpose will be removed. *Preparation* is the process in which application specific actions are performed.

Cleaning

The first cleaning step is resetting inconsistent individual and group values to predefined values. An invalid *individual* value is inconsistent with its range specification. Straightforward rules are developed such as “total number of lanes > 0”. Detecting inconsistent *group* values is done by comparing values; these rules are more complex such as “total number of lanes = number of forward lanes + number of backward lanes”.

A second cleaning step is *auto completing* OSM tags using predefined rules. This is done based on existing OSM tags. For example when “total number of lanes = 4”, “number of backward lanes = 2” and “number of forward lanes = UNDEFINED”, one can assume that the “number of forward lanes = 2”.

A third cleaning step (also enriching OSM tags) is adding and completing existing tags using a set of rules. Note that these rules are country dependent. In Belgium, the maximum speed of a “primary” road is 120 km/h. When there is a road with type “primary” and “max speed = UNDEFINED”, 120 is assigned to the max speed. Since the rules are country specific, it is not interesting to exhaustively list the set of rules used in the first three steps.

A fourth cleaning step is removing Links with a length of zero while maintaining the topology of the network.

A fifth cleaning step first merges TransportInfrastructures and then Links that are separated by *useless* splits. Such useless splits can be generated by previous cleaning steps or by mistakes during data entry.

Definition 2.3.6. A *useless split of a TransportInfrastructure* is a Point which has exactly two Links connected to it, where these two Links are in two separate TransportInfrastructures that have the same attributes, but a different list of Links.

The condition stating *exactly two Links* is sufficient because all Links in a TransportInfrastructure need to have identical attribute values, including the direction.

Definition 2.3.7. A *useless split of a Link* is a Point that has exactly two Links (belonging to the same TransportInfrastructure) connected to it.

In fact there is no reason for two TransportInfrastructures having identical attributes to be separated by a split. Note that it is very important that TransportInfrastructures are merged before the Links. Otherwise, new useless splits of Links might be introduced while merging TransportInfrastructures.

Reduction

The first reduction step is removing road types which are not wanted in the output data set. The cleaning tool allows the user to specify (i) a list of types to keep in the data set and (ii) a list of types to drop from the data set.

It is impossible to specify every road type that exists in the OSM data set due to the lack of attribute key and value validation as mentioned in Section 2.2. That is why a feature is implemented that will convert every type that is not contained in one of both lists to the "unclassified" type. This is done in order to avoid dropping road types that might be useful.

Another reduction is done by removing *sinks*, *sources*, *black holes*, *white holes* and *islands* in order to ensure that the transportation network constitutes a *strongly connected graph*.

Definition 2.3.8 (sink, source)**.** A node is a *sink* if it has no outgoing edges, i.e. you can enter it, but cannot leave it. A node is a *source* if it has no incoming edges, i.e. you can leave it, but cannot enter it.

Definition 2.3.9 (black hole, white hole)**.** A sub network is called a *black hole* if you can enter this sub network, but cannot leave it. A sub network is called a *white hole* if you can leave this sub network, but cannot enter it.

Definition 2.3.10. A sub network is called an *island* if you cannot enter and leave this sub network.

It was observed by interactive visual inspection that the part of the network constituting sources, sinks, black holes, white holes and islands represents a small number of road segments consisting mainly of walking roads. Therefore, it was judged that dropping those parts is a justified solution.

Preparation

As discussed in Section 2.1, the main goal (besides micro-simulations) of this data preparation is connecting GTFS stops to the OSM network. Note that the side of the road should be taken into account. Therefor, every TransportInfrastructure with direction BOTH is split into two identical TransportInfrastructures, one with direction FORWARD and one with direction BACKWARD. By doing this, candidate locations for GTFS stops can be assigned separately on both sides of the road.

2.3.3 Results

For the experiments the part of the OSM network delimited by the minimal bounding box that contains all the bus stops of "De Lijn" (PT provider for buses and trams of Flanders) which includes the northern part of Belgium and the southern part of the Netherlands is used. The following pipeline of steps is used: (i) remove Roads which are not needed, (ii) change type of Roads which do not occur in one of the two lists (iii) reset incorrect individual values, (iv) auto complete Roads, (v) reset incorrect individual values, (vi) reset incorrect group values, (vii) enrich Roads with rules, (viii) remove Links with zero length, (ix) merge TransportInfrastructures, (x) merge Links, (xi) remove Links/TransportInfrastructures not belonging to the strongly connected graph and (xii) convert TransportInfrastructures with direction BOTH into a FORWARD and BACKWARD TransportInfrastructure. Note that incorrect values are reset twice because new mistakes might be introduced when the data is auto completed. Suppose that "total number of lanes = 2", "total number of forward lanes = 4" and "total number of backward lanes = UNDEFINED"; in this case the individual values are correct. However, if the backward lanes are auto completed, it will result in -2, which is an incorrect individual value. In Table 2.1 an overview of the different steps is given. The main reduction in number of objects happens in Step (i) in which TransportInfrastructures are deleted which are not needed. Other decent reductions happen in Step (ix) and (x) in which TransportInfrastructures and Links are merged. At the end of Step (xi) the number of TransportInfrastructures, Links and Points are reduced by respectively 25.28 %, 32.60 % and 28.54 %. Finally, in Step (xii) there is a large increase in both TransportInfrastructures and Links due to the conversion of TransportInfrastructures with direction BOTH to FORWARD and BACKWARD.

TABLE 2.1: Results of the OSM data preparation pipeline. The symbol “#” represents the number of remaining objects and the symbol “δ” represents the number of modified objects. TransportInfrastructures is abbreviated as TI, Links as L and Points as P.

Description	**Step**	**#TI**	**#L**	**#P**	**δTI**
Initial	/	776,483	1,336,260	969,907	/
Remove not needed Roads	(i)	612,402	1,043,840	837,735	0
Change type of “unknown” Roads	(ii)	612,402	1,043,840	837,735	67,989
Reset incorrect individual values	(iii)	612,402	1,043,840	837,735	0
Auto complete Roads	(iv)	612,402	1,043,840	837,735	136,071
Reset incorrect individual values	(v)	612,402	1,043,840	837,735	0
Reset incorrect group values	(vi)	612,402	1,043,840	837,735	571,310
Enrich Roads with rules	(vii)	612,402	1,043,840	837,735	612,402
Remove Links with zero length	(viii)	612,402	1,043,815	837,731	0
Merge TransportInfrastructures	(ix)	580,211	1,043,815	837,731	0
Merge Links	(x)	580,211	905,397	699,313	0
Remove objects not belonging to the strongly connected graph	(xi)	580,211	900,702	693,068	0
Convert BOTH to FORWARD and BACKWARD	(xii)	1,132,382	1,764,648	693,068	0

2.4 General Transit Feed Specification

2.4.1 Terminology

The *General Transit Feed Specification (GTFS)* data set will not be explained in this paper. For more information the reader is referred to `https://developers.google.com/transit/gtfs/`.

2.4.2 Algorithms

Unresolved References Removal - Simplification

The GTFS data set uses files which are connected to each other by identifiers. Hence, there can be missing links when an identifier occurs in one file but not in the other files. Data records containing *unresolved references* are deleted from the data set.

The data preparation covered by this Section is used in order to be able to connect GTFS stops to the OSM network. Hence, duplicate trips (= trips which serve the same stops in exactly the same order) are useless because they need to be processed multiple times. It was decided to delete these duplicate trips because the time dimension is not interesting in this case (only in the sequence of stops). This reduction will decrease the complexity of the OSM-GTFS integration.

While analysing the GTFS data, situations where the same GTFS stop is served multiple times separated in the majority of the cases by a short interval (in most of the cases one minute) were encountered. It was decided to delete those stops because this will only make the OSM-GTFS integration more complex, without having an influence on the actual results of the assignments.

Trips Containing Stop Visits in Inconsistent Orders

Each stop has a unique identifier *id* and a *name* (in most but not all cases a combination of municipality and street name). If two stops share the same name, they are on the same bidirectional network Link (road segment) at opposite sides of the street (and hence, they correspond to opposite Link travel directions). As a consequence, each stop is uniquely identified by an *id*, but also by a pair $\langle name, dir \rangle$ where $dir \in \{FORWARD, BACKWARD\}$ and name identifies a network Link. People can board and alight a vehicle at only one side (in continental Europe the right hand side).

Assume two stops S_1 and S_2 so that both $\langle S_1, S_2 \rangle$ and $\langle S_2, S_1 \rangle$ are subsequences of some, not necessarily different, trips (i.e. the stops are used consecutively and in both orders). Cases where S_1 and S_2 are used in both sequence orders but not as consecutive stops in a trip are not considered. The Links corresponding to the respective stops need to be crossed in the same direction in all cases (because of the boarding and alighting side restriction).

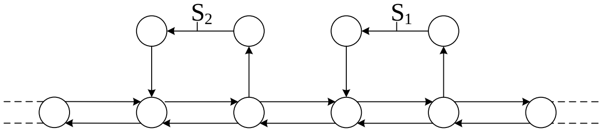
In case S_1 and S_2 share the same name, they correspond to a single Link and to opposite directions (because they are different stops). Both $\langle S_1, S_2 \rangle$ and $\langle S_2, S_1 \rangle$ imply a U-turn in the trip (which is possible).

In case S_1 and S_2 have different names and none of these names are shared names, it is sure that in all cases exactly the same stops are used (since none of the stops have a counterpart at the opposite side of the street). This case is shown in Figure 2.3a. This case was found in reality and necessarily induces a cycle in the route.

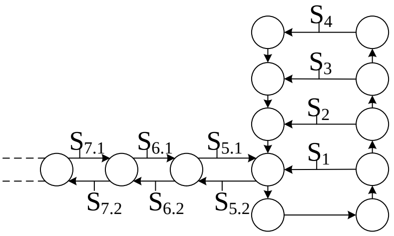
Finally, if the stops have different names and at least one of the names is a shared name, then it is assumed that an error was made while creating the GTFS database and a *voting* concept is used to select one of the cases to be kept. In this case the number of trips for both occurrences $\langle S_1, S_2 \rangle$ and $\langle S_2, S_1 \rangle$ are counted. The case that is found in the majority of trips is assumed to be the right one. The other trips are corrected by swapping the "wrong" pair. The reduction step will be explained using the synthetic example shown in Figure 2.3b. For that case the *shared stop name* assumption means that exactly one name is associated with each of the pairs $\{S_{5.1}, S_{5.2}\}$, $\{S_{6.1}, S_{6.2}\}$ and $\{S_{7.1}, S_{7.2}\}$. Suppose the following four trips are present: $\{S_4 - S_{5.2} - S_{6.2} - S_{7.2}, S_3 - S_{5.2} - S_{6.2} - S_{7.2}, S_2 - S_{5.2} - S_{6.2} - S_{7.2}, S_1 - S_{6.2} - S_{5.2} - S_{7.2}\}$. The first step consists of finding pairs of *consecutive* stops used in both orders. In this case a list of ordered pairs is found $\langle S_{5.2}, S_{6.2} \rangle$ and $\langle S_{6.2}, S_{5.2} \rangle$. The next step is counting the number of trips in which the pairs occur in every trip of the data set. In our case, $\{S_{5.2}, S_{6.2}\}$ and $\{S_{6.2}, S_{5.2}\}$ are compared with every trip in the data set. The pair $\{S_{5.2}, S_{6.2}\}$ occurs three times, while $\{S_{6.2}, S_{5.2}\}$ occurs only one time. It is assumed that the case having the highest occurrence frequency is the correct one and hence, the last trip will be corrected. Suppose that there is not a majority, but the occurrences of both pairs are equal; in this case none of the trips are corrected. Note that these cleaning steps were only tested for buses and trams (GTFS for "*De Lijn*"). The single side boarding/alighting assumption does not hold for trains (at least not in Belgium).

2.4.3 Results

For the experiments, the following pipeline of reduction steps were used: (i) find unique trips, (ii) remove duplicate consecutive stops and (iii) correct stops which are used in both directions. (iv) remove data which is not connected,



(A) A real case where stops S_1 and S_2 are visited in both orders. The trip moving from the left to the right contains two loops.



(B) The example explained in the text.

FIGURE 2.3: Two parts of a network: the circles represent road junctions, the arrows represent lanes and the symbols S_i and $S_{i,j}$ identify stops; the location on the Link is denoted by the small stroke perpendicular to the Link.

Step (iv) is executed in a loop, since removals of inconsistent data can trigger other data to be inconsistent as well. In Table 2.2, an overview is given of the percentage of deleted objects in every file. For the case study "De Lijn", the following results where found: the algorithm detected 19.12% of trips (i.e. the unique ones) in which duplicate consecutive stops occur and 0.57% of the trips were corrected due to the inconsistent orders.

TABLE 2.2: Overview of deleted objects in percentage per file for a set of GTFS files. In the header CD stands for Calendar Dates and ST for Stop Times.

GTFS File	Agencies	CD	Routes	Stops	ST	Trips
De Lijn	0.00	42.62	15.06	19.13	97.45	97.50
TEC	0.00	22.22	0.00	33.33	93.11	93.20
MIVB	0.00	0.00	32.56	39.12	99.84	99.79
Connexxion	0.00	27.55	0.00	22.06	97.20	96.85
EBS	54.17	79.44	5.41	37.12	81.44	84.04

2.5 Finding Candidate Locations for GTFS Stops

2.5.1 Algorithm

The goal of this algorithm is finding a set of candidate locations, called *projected stops*, for every GTFS bus stop. One of these projected stops will be chosen as the representative of the GTFS stop in the assignment algorithm which is presented in [Vuurstaek et al., 2018]. These projected stops were determined by the use of a PostGIS database. In this database, all the cleaned OSM data was imported (as described in Section 2.3.2).

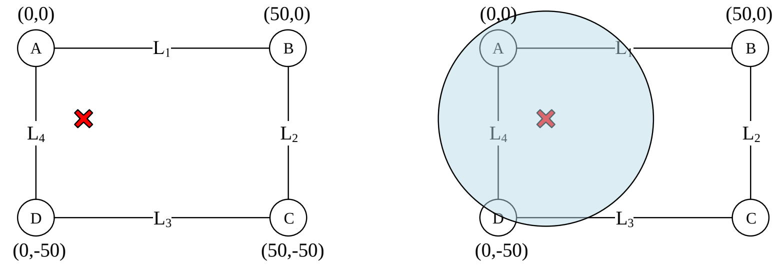
To find projected stops, the radius in which needs to be searched and the desired maximum number of projected stops are determined. For this step the *geographic information system (GIS)* functionality of the PostGIS database is used.

For every GTFS stop, projected stops are calculated. This is achieved by finding every Link in a radius R of the GTFS stop. This query will return between zero and maximum X nearest Links as a result. Both R and X can be configured in the software. When no Links are found in the radius R, the algorithm will increase the radius R and will attempt to find Links again. This process will continue until at least one Link is found. In order to find a candidate location, the nearest point on the link geometry on which the GTFS stop is projected is needed. This is done by another query which can find the projection of a Point (coordinates of the GTFS stop) on a Link (found by the previous query).

In Figure 2.4, an example of these steps is given. For this example, a radius of 50 metres and a maximum number of projected stops of three was chosen. In Figure 2.4a, the starting situation with four Links and a GTFS stop (cross) can be observed. In Figure 2.4b, the radius which is specified can be seen and it is clear which Links are found. The final step is finding the exact location of the projection, this can be seen in Figure 2.4c. Note that it is also possible that two or more projections on the same link with the same distance are found. In such cases a random point out of these closest points will be chosen. This can be seen in Figure 2.4d.

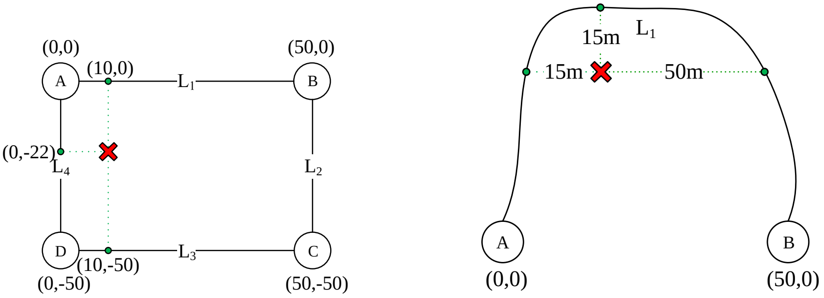
2.5.2 Results

In [Haklay et al., 2010], the positional accuracy of OpenStreetMap roads in the Greater London area was investigated. In *complete areas* the average error is 9.57 m with a standard deviation of 6.51 m. In incomplete areas the average error is 11.72 m and the standard deviation is 7.73 m. In [Haklay et al., 2010], completeness is defined as "*a measure of the lack of data*" and examined for specific areas by visual inspection of maps and by comparing (by means of GIS) the total road length found in OSM and in



(A) Four Links and a GTFS stop.

(B) Three Links fall in the radius.



(C) Projections on the Links are calculated.

(D) Three projections on the same link of which two projections share the shortest distance. A random point from the two shortest will be chosen.

FIGURE 2.4: Overview of finding projected stops.

reference maps respectively. From several non-authoritative website sources it was found that the accuracy threshold $\overline{d}$ at 95 % for GPS devices (used to locate the bus stops) can be assumed to be 20 m. Based on both error limits, it was decided to use 30 m as the radius to find matching Links for a bus-stop. For the experiments a maximum of ten projected stops per GTFS stop was chosen. In Table 2.3, an overview of the number of GTFS stops having a specific number of projected stops is given. For instance, there are 16,196 GTFS stops which have two projected stops.

In total there are 127,066 projected stops found for a total of 30,654 GTFS stops, which is on average 4.15 projected stops per GTFS stop; this means the number of projected stops found per GTFS stop was not underestimated. In none of the cases an increase of the radius was needed, which means that 30 m was well chosen. One can observe that the number of even occurrences is significantly higher than the amount of odd occurrences. This is due to the fact that TransportInfrastructures with a direction of BOTH (which is the majority) were converted to a FORWARD

TABLE 2.3: Number of GTFS stops having a number of projected stops.

Number of Projected Stops found per GTFS stop	Occurrence Frequency
1	161
2	16,196
3	255
4	3,276
5	161
6	5,450
7	95
8	2,030
9	66
10	2,964

and a BACKWARD TransportInfrastructure.

2.6 Conclusion and Future Work

Accurate detailed data is required for micro-simulation aimed at the evaluation of collective transportation facilities. The data preparation consists of (i) the creation of an automatic tool to import OSM and GTFS data and (ii) the development of an algorithm to automatically assign about 30k bus stops to the OSM network (about 500k links). This paper describes the first stage of the OSM-GTFS integration, i.e. the algorithms to clean, reduce and prepare the OSM and GTFS data. Due to the size and the update frequency of the OSM and GTFS data, integrating them interactively is not an option. The bus stop matching algorithm can be found in [Vuurstaek et al., 2018].

Acknowledgements

The research reported was partially funded by the IWT 135026 Smart-PT: Smart Adaptive Public Transport (ERA-NET Transport III Flagship Call 2013 “Future Travelling”).

Bibliography

C. Barron, P. Neis, and A. Zipf. A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, 18(6):877–895, 2014. doi: 10.1111/tgis.12073.

G. Cich, J. Vuurstaek, L. Knapen, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Data Preparation to Simulate Public Transport in Micro-Simulations using OSM and GTFS. *Procedia Computer Science*, 83:50–57, 2016. ISSN 1877-0509. doi: 10.1016/j.procs.2016.04.098. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.

J.-F. Girres and G. Touya. Quality Assessment of the French OpenStreetMap Dataset. *Transactions in GIS*, 14(4):435–459, 2010. doi: 10.1111/j.1467-9671.2010.01203.x.

M. Haklay. How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets. *Environment and Planning B: Planning and Design*, 37(4):682–703, Aug 2010. ISSN 0265-8135. doi: 10.1068/b35097.

M. Haklay, S. Basiouka, V. Antoniou, and A. Ather. How Many Volunteers Does it Take to Map an Area Well? The Validity of Linus' Law to Volunteered Geographic Information. *The Cartographic Journal*, 47(4):315–322, Nov 2010.

P. Mooney and P. Corcoran. The Annotation Process in OpenStreetMap. *Transactions in GIS*, 16(4):561–579, 2012. doi: 10.1111/j.1467-9671.2012.01306.x.

J. Vuurstaek, G. Cich, L. Knapen, W. Ectors, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. GTFS bus stop mapping to the OSM network. *Future Generation Computer Systems*, 2018. ISSN 0167-739X. doi: 10.1016/j.future.2018.02.020.

M. Zilske, A. Neumann, and K. Nagel. OpenStreetMap for Traffic Simulation. In *Proceedings of the 1st European State of the Map - OpenStreetMap Conference,*, pages 126–134, Vienna, Austria, 2011.

Chapter 3

Towards an Agent-Based Model for Demand Responsive Transport Serving Thin Flows

The peer-reviewed work presented in this chapter was published on May 12, 2016 in the Procedia Computer Science [Cich et al., 2016]. The work was presented at The 5th International Workshop on Agent-based Mobility, Traffic and Transportation Models, Methodologies and Applications (ABMTRANS 2016) in Madrid, Spain held on May 23-26, 2016.

Keywords: Demand Responsive Transport; Thin Flows; Micro-Simulation; Agent-Based modelling; Organisational modelling

Abstract

Low volume traveller flows cause problems for public transportation (PT) providers. The Smart-PT project aims to find out how such flows can be combined to increase the service provider viability. The capability to conceive multi-modal trips is fundamental in that context and is modelled by the Trip Sequence Composer (TSC) concept. A TSC is an essential component of the traveller's brain, of the customer support operated by collective transport providers, of trip advisers in websites, etc. A simulation model design to evaluate the effect of cooperating TSCs on the viability of demand responsive collective transport providers is presented. While obeying specific regulations, specialised services targeting mobility impaired people can also serve regular requests in order to save fleet and personnel costs. All stakeholders are assumed to optimise their private objectives and none of them has global perfect knowledge.

3.1 Problem Context and Objectives

Informally, a *flow* is a set of movements (partial trips, legs) that uses a path in a given *connection* during a period that overlaps with a given *period sequence*, for instance the movements from a given bus stop to the central station on Sunday morning between 09:00h and 12:00h in 2015. In a *thin flow* the number of movements is below a given threshold. *Thin flows* originate from low demand which in turn is caused by lack of zonal attraction or low customer density. The latter can follow from low population density and from special customer requirements (e.g. wheelchair bound customers). *Thin flows* are defined using *period sequences* and *connections*. A *period sequence* is defined by a pair $\langle I, S, E \rangle$ where I is a time-of-day interval, S is a period specifier and E is an era specified by $\langle D_0, D_1 \rangle$ where D_0 is the initial date and D_1 is the final date. For example, every first Wednesday of the month from 13:30h till 16:30h in September-December 2016. A *connection* between a source S and a target T is a set of paths connecting S to T.

Due to budgetary constraints time table based public transportation (PT) on thin flows is reduced and services on demand are discontinued. The Flemish government decided to investigate the viability of providers that serve demand from customers both with and without special requirements. This requires accurate modelling (i) of providers collecting (customer class specific) fares and spending money to run the operations, (ii) of customers who are sensitive to cost, trip duration and timely service and (iii) of interactions between providers and customers.

An agent-based model (ABM) is a class of computational models to simulate the actions and interactions of autonomous agents with a view to assess their effects on the system as a whole. ABM is now widely used for modelling increasingly complex systems. Application of ABM is not only limited to the computer science domain. Currently, many research areas such as transportation behaviour modelling need to analyse and understand the complex phenomenon of interactions between different entities. While traditional modelling tools cannot catch the complexity, ABM can do it through modelling the interaction of autonomous agents and deducing the rules for such a system. Nevertheless, for a successful application and deployment of an agent-based system, a large number of the agent methodologies recognises (to varying extents) the idea that a multi-agent system can be conceived in terms of an organised society of individuals in which each agent plays specific roles and interacts with other agents [Zambonelli et al., 2003]. As pointed out by Ferber et al. [2004], an approach based on organisations and roles offers a number of advantages and can contribute to agent-oriented software development in the following points: heterogeneity of languages, modularity, multiple possible architectures and security of applications. In other words, an organisational approach may break down the design complexity of an agent-based system. Therefore, in this paper, **an organisational and agent-based model is proposed to evaluate travel demand and supply in thin flows. The model is aimed to simulate thin flow travel over a two year period in order to determine the conditions of viability for the competing and optimising transport providing providers, none of which has full knowledge about the universe. This paper constitutes the first step towards this goal by providing a simulation model.** Thin flows are expected to be highly variable in time because the demand for each individual can show trips separated by stochastic periods of time and consist of trips covering several locations. Furthermore, the capacity of the vehicles is typically small. Hence, providers need to solve vehicle routing problems frequently in order to deliver high quality service while keeping the vehicle occupancy at a decent level. The resulting solutions of the combinatorial optimisation are expected to differ greatly from day to day. Under such conditions, economic viability can be assessed only by integration over a long period.

The remainder of the paper is organised as follows: in Section 6.2, an overview of important related work is given. Section 3.3 introduces the main concepts used in the simulation. Section 3.4 describes the agent-based framework which will be used to develop the simulation software. Finally, in Section 3.5, a conclusion is presented and future work is discussed. This paper reports on the ongoing design of *Demand Responsive Transport* (*DRT*) simulation.

3.2 Related Work

Zografos et al. [2008] attempt to find a methodological framework for developing and assessing Flexible Transport System (FTS) business models. They discuss three dimensions, namely the *FTS context*, *business strategy and functions* and *service offerings*. The FTS context describes elements such as the site location, potential market and market opportunities. The business strategy handles elements such as business vision and the economic structure. Finally, the service provided is characterised by elements such as *service topology*, *target market* and *types of vehicles* used. The authors state that it is sufficient to specify these three dimensions in order to define a new business. The developed framework to help decision makers consists of a development phase in which alternative FTS businesses that are compatible in the local market are identified, a screening phase in which the economic feasibility is analysed and a prioritisation phase in which the remaining models of the screen phase are ranked and assessed. They tested their framework in a case study located in Helsinki, Finland.

Neven et al. [2015] assess the impact of different policy decisions on resource requirements of Demand Responsive Transport (DRT) services in Flanders, Belgium. Mainly, there are two types of DRT: (i) services offered in low demand rural areas and (ii) door-to-door services for mobility impaired people. The authors focus on (ii). A synthetic population of mobility impaired people was created, and their corresponding transportation requests with specific travel-characteristics (the specific travel demand) were generated, based on survey data and official data about disability. This is assigned to the transportation network and time dependent inter-zonal travel times are computed. Vehicle Routing Problems (VRP) are solved under several budgetary constraints resulting in a what-if analysis to support policy decisions by the Flemish government. Spatial and temporal effects are taken into account. The results show that the change of the modal split, better accessible public transport and improved flexibility regarding customers are the key elements to minimise resource requirements in a DRT system for mobility impaired people. Similar to many other papers, the solution is Operational Research (OR)-based and assumes full knowledge by a central optimiser.

The model described in this paper provides several refinements. A multi-month period is simulated to average the effects of stochastic demand. Thereto, a multi-day agenda (having a one or two week period) needs to be generated for each participant in a *thin flow*. Trips requested by customers with and without special requirements will be served by the same provider and can be combined. Multi-modal trips are supported because a thin flow can feed a thick flow. Customer behaviour is modelled explicitly because service quality perception feeds back to trip request behaviour. Customer satisfaction determines recurrent use of services and depends on the travel cost (monetary, time loss) and on the ability to match the preferred time windows. Finally, modelling thin flows requires the use of street addresses. The spatial granularity of a zone based system is insufficient to achieve accurate results.

3.3 Concepts

3.3.1 Demand Responsive Transport

According to Ellis [2009], *Demand Responsive Transport* (*DRT*) is handled by a provider, which provides transport by passenger cars, vans or small buses. Customers, e.g. (mobility impaired) people or other providers can call a DRT provider who then dispatches a vehicle to bring a customer from his origin to his destination. The characteristics of a DRT service is twofold: (i) vehicles do not serve a fixed route or a fixed time schedule, however this is possible on peak times and (ii) vehicles tend to pick up multiple customers at different locations (new requests can be handled in real time) before dropping them off at their destination.

The following measurement quantities proposed by Ellis [2009] could be used to evaluate DRT provider viability in a multi-year simulation: (i) *passenger trips* which measures the number of served customer requests, (ii) *on-time trips* which measures the number of trips in which customers are picked-up in the predefined time window, (iii) *no-shows* which is the failure or not willingness of a customer to show up for a reserved trip at the scheduled time and location, (iv) *late cancellation* which is a cancellation done by the customer of a reserved trip shortly before the vehicle is scheduled to arrive, (v) *missed trips* which are trips in which a DRT provider fails to pick up a scheduled customer and (vi) *trip denials* which are trips that could not be fulfilled by a DRT provider (e.g. no vehicles available). Some of these variables will be used to estimate the revenues and costs associated with the services, in order to assess the viability of the DRT providers.

3.3.2 Booking TripSequences

Planning and booking trips are essential mechanisms. A *trip* corresponds to a movement between two locations. It consists of *tripComponent*s (legs using different modes or providers). A *tripSequence* consists of trips that need to be handled atomically; the requester is interested in receiving a proposal for either *all* or *none* of them (e.g. round trips). A request for booking can apply to multiple tripSequences at once.

Both customers and providers are characterised by means of *labels* defining specific requirements and provisions. Four categories of *labels* are distinguished: (i) *physical* labels which describe the mobility impairment, (ii) *personal* labels which affect the fares (e.g. family status, age and employment status), (iii) *preference* labels which describe person specific preferences such as transportation mode or maximum travel time and (iv) *financial* labels which constrain the mode choice (e.g. income category and car ownership).

Operational, legal and infrastructure labels define the capabilities and properties for the transport providers (e.g. target customer population segment, wheelchair support, etc.).

Label matching is used to decide which customer categories can be served based on physical requirements (wheelchair, visual support, etc.) and tariff rules (based on impairment, age, income, etc.) and which fares apply in each particular case.

To conclude, several kinds of constraints need to be taken into account: customer and provider properties encoded by labels as well as time windows applying to both the booking procedure (time between request issuing and required trip start) and the requested trips (pick-up, drop-off).

When a trip request is sent to a provider, eventually the customer will get a reply, either positive or negative. If the reply is positive, a *trip proposal* is sent to the customer. Analogous to the trip request, this consists of *trip proposal components*. First of all, every trip proposal component refers to a trip request component in order to identify which components belong together. Note that a trip request proposal can be subdivided in different legs (= multi-modal trip). A trip proposal component has (i) an origin-destination pair $\langle (X_{from}, Y_{from}), (X_{to}, Y_{to}) \rangle$, (ii) a departure time window [DTW_s (hh_s:mm_s), DTW_e(hh_e:mm_e)], (iii) an arrival time window [ATW_s (hh_s:mm_s), ATW_e(hh_e:mm_e)], (iv) a time until the proposal expires, (v) a commitment of the travel, (vi) a price in the desired monetary unit per customer and (vii) a transport mode. Important elements of this trip proposal component are the commitment of the travel which means whether or not a provider ensures that the customer can travel, e.g. if a customer travels with public transport there is a (small) chance that there is no place left in the vehicle. Another important aspect is the time until the proposal expires. When a proposal is sent out to a customer, the provider keeps track of the reservation until the proposal expiration time. Hence, a customer can lose the trip proposal if he fails to approve it before it expires.

3.3.3 Trip Sequence Composer

The *trip sequence composer (TSC)* is an important entity in this design and every autonomous agent contains a TSC instance. The TSC can be seen as the knowledge of an agent of whatever kind to compose (multi-modal) trips. Examples of such TSCs in reality are (i) the brain of a traveller, (ii) help desk support of a provider, (iii) a website of a public transport provider (route planner), (iv) a navigation app of a smartphone and (v) the personal coach of a mentally disabled person. An important fact is that every agent's TSC has a limited knowledge about the "world". A TSC might have some knowledge about his own capabilities (e.g. schedules of public transport or driving a car) and he might know some other TSCs such as a taxi provider or his own navigation app. An agent is for instance able to compose a trip to his work by using public transport (because he knows the schedules). In contrast, if he wants to travel to the airport he might want to rely on the knowledge of a taxi provider. Hence, a TSC is able to arrange trips (by using his own knowledge) from point *A* to point *B* (i) by only using his own capabilities or (ii) by subdividing the trip request component and delegate some of the simplified requests to other TSCs.

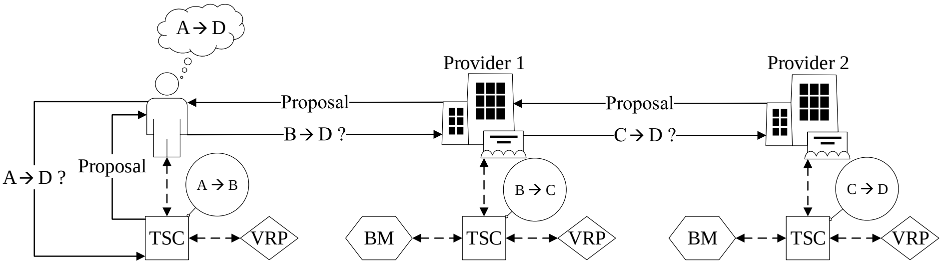
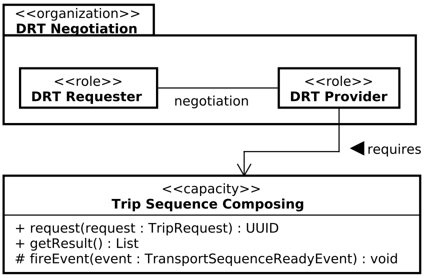


FIGURE 3.1: The main communication flows between agents and TSCs to request a trip. TSC stands for trip sequence composer, VRP stands for vehicle routing problem solver and BM stands for business manager. Solid arrows indicate direct communication between autonomous agents, while dashed arrows indicate function calls to entities.

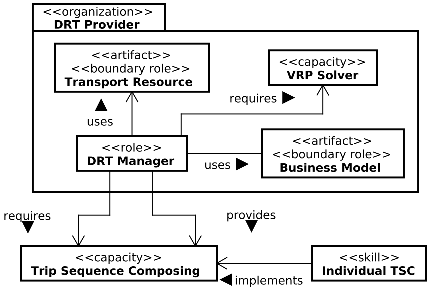
In Figure 3.1 the conceptual design of booking a trip in combination with the different agents is shown. A specific customer wants to travel from A to D, hence a trip request from A to D is sent to his own TSC. His own TSC is not able to fulfil the complete request, hence the TSC will split it in $A \rightarrow B$ (which can be done by himself) and $B \rightarrow D$ (which another TSC needs to solve). The TSC of the customer is aware of some providers providing transport services, hence provider 1 is contacted. The TSC of provider 1 is able to fulfil the request partially. It is able to bring the customer from $B \rightarrow C$. The remaining part of the trip ($C \rightarrow D$) is propagated to provider 2 (which is known by provider 1). The TSC of provider 2 is able to fulfil this request. A proposal is sent to provider 1, which combines this proposal with his own proposal and sends it back to the customer. Eventually, his proposal is combined with the proposal of provider 1 and hence, a full proposal from $A \rightarrow D$ is made. Note that the TSC invoke other entities such as the *business manager* and the *vehicle routing problem solver* as well. The business manager keeps track of the number of customers served, the revenues, the time spent, the distance driven, etc.; while the vehicle routing problem solver attempts to schedule the trip.

3.4 Organisational and Agent-Based Models

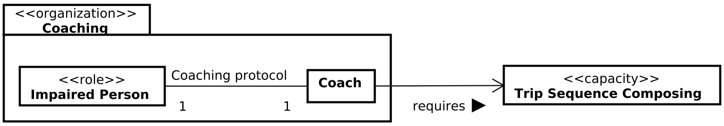
In this Section, an organisational model for the *thin flows* application and its mapping to the corresponding ABM based on the meta-model defined by Cossentino et al. [2010] is proposed. The concepts presented in the previous Sections are mapped to three organisations and their respective roles. These organisations are then mapped to agents that are playing the roles in the different organisations for fulfilling the system's requirements.



(A) Organisation for DRT Negotiation. The roles *DRT Requester* and *DRT Provider* are interacting for negotiating a trip sequence. *DRT Provider* uses its know-how for building the sequence. The organisation is a possible solution for representing this know-how.



(B) Organisation for DRT provider. The *DRT manager* has the role to compute and provides trip sequence according to the provider's constraints.



(C) Organisation for coaching impaired people. Some impaired people need to be coached for obtaining a trip sequence. They must interact with their coach, who is able to build the sequence.

FIGURE 3.2: Organisations in the DRT system.

The DRT system that is considered in this paper is decomposed into three different organisations (Figure 3.2). Each organisation is a subsystem that fulfils one or more of the requirements of the system. The central organisation is related to the *DRT Negotiation* (Figure 3.2a). It enables an agent to request and negotiate a trip sequence. The requester and the trip sequence provider are defined as the two roles in this subsystem. The latter role requires a specific know-how from the playing agent for building and composing the trip sequence. This know-how is represented by the concept of *capacity* in the organisational meta-model.

The second organisation defines the transport providers as a specific subsystem (Figure 3.2b). The DRT managers in the providers are in charge of building and composing trip sequences. If the agent playing this role owns a dedicated skill, i.e. an concrete realisation of the *Trip Sequence Composing* capacity, then it uses it for building the sequence. Otherwise, the role's behaviour will contact other transport providers by joining a new instance of the *DRT Negotiation* organisation, in which it will play the role *DRT Requester*. According to the organisational meta-model, a contribution link between the two organisations exists since the *DRT Manager* and *DRT Requester* roles require and provide respectively the same *Trip Sequence Composing* capacity. In other words, the *DRT Negotiation* organisation contributes to the *DRT Manager* behaviour by providing the part that is corresponding to the capacity.

Finally, several impaired persons will need a coach for negotiating their trip sequences. The *Coaching* organisation (Figure 3.2c) defines the specific interaction in this context. If the coach has not the personal capacity to build the trip sequence, she/he could participate to an instance of the *DRT Negotiation* organisation for obtaining a valid sequence from the transport providers.

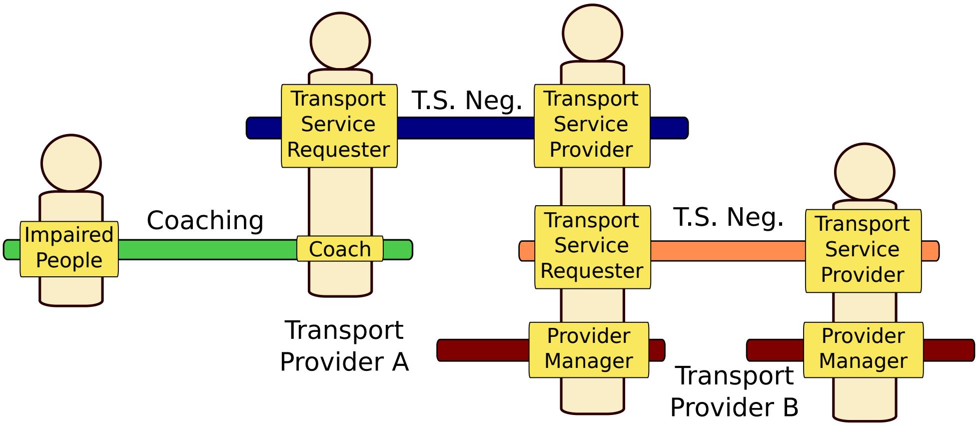


FIGURE 3.3: Example of the agents that are playing roles in the defined organisations.

At the end of this organisational design phase, the organisation structure is mapped into a society of agents in charge of realising the expected behaviours. Each of the previously identified organisations is instantiated in form of groups. Corresponding roles are then associated to agents. The agents are assumed to run the behaviour of each role they are playing. Consequently, a part of their personal behaviours is related to the decisions of joining or exiting the groups, and selecting the scheduling policy of their different played roles. All of these elements are finally merged to obtain the complete set of agents involved in the solution. Figure 3.3 presents an example of four agents (one impaired person, one coach and two transport providers) that are participating in five different groups.

3.5 Conclusion and Future Work

The model proposed in this paper is currently being implemented with the SARL agent-oriented language. It provides concepts and statements that could be directly mapped to the organisational concepts used for building our model. In order to provide a proof-of-concept and to validate the added value of our proposal, the model will be applied during the next couple of months to the region of Leuven, Belgium for which historic DRT bus occupation data as well as historic data about trips generated by a day care centre are available to generate the demand.

Acknowledgements

The research reported was partially funded by the IWT 135026 Smart-PT: Smart Adaptive Public Transport (ERA-NET Transport III Flagship Call 2013 “Future Travelling”).

Bibliography

G. Cich, L. Knapen, S. Galland, J. Vuurstaek, A. Neven, and T. Bellemans. Towards an Agent-based Model for Demand-Responsive Transport Serving Thin Flows. *Procedia Computer Science*, 83:952–957, 2016. ISSN 1877-0509. doi: 10.1016/j.procs.2016.04.191. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.

M. Cossentino, N. Gaud, V. Hilaire, S. Galland, and A. Koukam. ASPECS: an Agent-Oriented Software Process for Engineering Complex Systems - How to Design Agent Societies Under a Holonic Perspective. *Autonomous Agents and Multi-Agent Systems*, 2(2):260–304, Mar 2010. doi: 10.1007/s10458-009-9099-4.

E. Ellis. *TCRP Report 136: Guidebook for Rural Demand-Response Transportation: Measuring, Assessing and Improving Performance*. Transportation Research Board, 2009. ISBN 9780309118071.

J. Ferber, O. Gutknecht, and F. Michel. From Agents to Organizations: an Organizational View of Multi-Agent Systems. In *Agent-Oriented Software Engineering IV*, volume 2935 of *LNCS*, pages 214–230. Springer Verlag, Mar 2004.

A. Neven, K. Braekers, K. Declercq, G. Wets, D. Janssens, and T. Bellemans. Assessing the Impact of Different Policy Decisions on the Resource Requirements of a Demand Responsive Transport System for Persons with Disabilities. *Transport Policy*, 44:48–57, 2015. ISSN 0967-070X. doi: http://dx.doi.org/10.1016/j.tranpol.2015.06.011.

F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing Multiagent Systems: The Gaia Methodology. *ACM Transactions on Software Engineering and Methodology*, 12(3):317–370, July 2003. ISSN 1049-331X. doi: 10.1145/958961.958963.

K. G. Zografos, K. N. Androutsopoulos, and T. Sihvola. A Methodological Approach for Developing and Assessing Business Models for Flexible Transport Systems. *Transportation*, 35:777–795, 2008. ISSN 00494488. doi: 10.1007/s11116-008-9176-6.

Chapter 4

Addressing the Challenges of Conservative Event Synchronisation for the SARL Agent-Programming Language

The peer-reviewed work presented in this chapter was published on May 12, 2016 in the Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection [Cich et al., 2017]. The work was presented at 15th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS17) in Porto, Portugal held on June 21-23, 2017.

Keywords: Multi-agent simulation; Conservative event synchronisation; SARL agent programming language; Janus platform

Abstract

Synchronisation mechanism is a key component of an agent-based simulation model and platform. Conservative and optimistic models were proposed in the domain of distributed and parallel simulations. However, the SARL agent-programming language is not equipped with specific simulation features, including synchronisation mechanisms. The goal of this paper is to propose a conservative synchronisation model for the SARL language and its runtime platform Janus.

4.1 Introduction

In agent-based simulations, the entire simulation task is divided into a set of smaller sub tasks each executed by a different agent. These agents may run in parallel, and communicate with each other by exchanging timestamped events or messages. In this paper, an event refers to an update to the simulation system's state at a specific simulation time instant. Throughout the simulation, events arrive at destination agents, and depending on the delivery ordering system of the simulation, they are processed differently. The two commonly used orderings are (i) event reception order and (ii) event timestamp order (the timestamp is assigned to the event by the emitting agent). With the first type, events are delivered to the destination processes when they arrive at the destination. On the other hand, with the timestamp order, events are delivered in non-decreasing order of their timestamp, requiring runtime checks and buffering to ensure such ordering.

The key question is how to create a synchronisation model, and its implementation, based on the SARL agent-programming language, and assuming that the execution platform is fully distributed. In this paper, the SARL agent-programming language is equipped with an event synchronisation model with the following characteristics: (i) the model follows a conservative synchronisation approach, (ii) the synchronisation process is hidden to the agents by using the *capacity* and *skill* concepts and (iii) the agent environment is integrated into the synchronisation process.

The remainder of the paper is organised as follows: in Section 4.2, an overview of the current state of the art is given. Section 4.3 introduces the agent-based framework SARL that is used in the model described in this paper. Section 4.4 describes the method used to synchronise events in SARL. The evaluation of this method is described in Section 4.5. Finally, in Section 4.6, the paper is concluded.

4.2 Related Work

Parallel Discrete-Event Simulation (PDES) has received increasing interest as simulations become more time consuming and geographically distributed. A PDES consists of Logical Processes (LPs) acting as the simulation entities, which do not share any state variables (similar to agents) [Fujimoto, 2000].

A PDES that exclusively supports interaction by exchanging timestamped messages obeys the local causality constraint if and only if each LP processes events in non-decreasing timestamp order [Fujimoto, 1990]. To satisfy the local causality constraint, different synchronisation techniques have been proposed for distributed systems which generally fall into two major classes of synchronisation: *conservative* *or pessimistic*, which strictly avoids causality violations; and *optimistic*, which allows violations and recovers from them.

Conservative synchronisation algorithms strictly avoid any occurrence of causality errors. To do so, the LP is blocked from further processing of events until it can make sure that the next event in its local future event list has a timestamp smaller than the arrival time of any event that might be arriving at the LP in the future. The main issue of any conservative parallel simulator is determining if it is safe for a processor to execute the next event. To deal with this issue, several techniques have been proposed which are further classified into four categories: (i) methods with dead–lock avoidance, (ii) deadlock detection and recovery, (iii) synchronous operation and (iv) conservative time windows.

Optimistic synchronisation algorithms do not try to stop the LP's execution to synchronise them. It allows causality errors to occur and to be detected by the arrival of an event with a timestamp that is less than the local time of the receiving LP. Optimistic algorithms recover from the causality error by undoing the effects caused by those events processed speculatively during the previous computation. This recovery operation is known as *rollback*, during which the state of the LP is restored to the one that was saved just prior to the timestamp of the violating event. The main issue of any optimistic parallel simulator is related to the necessary storage space that is needed for recovery, and the positive ratio of the time spent for performing the recovery on the time spent for executing the behaviour of the system.

In the past three decades, numerous approaches have been proposed by different researchers in this field. A number of surveys can be found in the literature which summarise both conservative and optimistic techniques [Fujimoto, 1990, 2000; Jafer et al., 2013; Perumalla, 2006; Perumalla and Fujimoto, 2001; Tropper, 2002].

In multi-agent systems, several models of synchronisation were proposed. In this domain, agents are assimilated to LPs. Weyns and Holvoet [2003] describe a conservative synchronisation module in the multi-agent system that is based on the composition of the $synchronisation_a$ modules for each agent a. The approach of the model is to let synchronisation be the natural consequence of situatedness of agents and not be part of the agents decision mechanism. This is reflected in the fact that the composition of a set of synchronised agents only depends on the actual perception of the agents. Such synchronisation is based on the exchange of a structured set of synchronisation messages.

Braubach et al. [2004] propose a centralised service that has the role to manage the time evolution, and to notify the agent when the time is evolving. In this model, each agent notifies the time management service when it has finished its task for a given time period. This model is one of the most simple pessimistic synchronisation algorithms. Its major drawback is related to the introduction of the centralised service that makes it harder and less efficient to distribute the agents over a computer network.

Xu et al. [2015] propose an asynchronous conservative synchronisation strategy for parallel agent-based traffic simulations. The authors propose to replace the global synchronisation barrier in the multi-agent system by a local synchronisation strategy that enables agents to communicate individually and providing each of the agents with a heuristic for increasing the time window look ahead in order to predict the next safe events.

4.3 SARL: An Agent-Oriented Programming Language

SARL[1] is a general-purpose agent-oriented programming language [Rodriguez et al., 2014]. Such language should thus provide a reduced set of key concepts that focuses solely on the principles considered as essential to implement a multi-agent system. In this paper, four elements of the metamodel of SARL are used: Agent, Space, Capacity and Skill. These four concepts are explained below.

Agent An *Agent* is an autonomous entity having a set of skills to realise the capacities it exhibits. An agent has a set of built-in capacities considered essential to provide the commonly accepted competences of agents, such as autonomy, reactivity, proactivity and social capacities. The various behaviours of an agent communicate using an event-driven approach.

Space A *Space* is the abstraction to define an interaction space between agents or between agents and their environment, which may be the real world or a simulated environment. The simulated environment subsystem could be modelled with a multi-agent system by itself. In the SARL toolkit, a concrete default space, which propagates events, called `EventSpace` is proposed.

Capacity *A Capacity is the specification of a collection of functions* that support the agent's capabilities, which are represented by the Capacity concept. This specification makes no assumptions about its implementation. It could be used to specify what an agent can do, i.e. what a behaviour requires for its execution.

Skill A *Skill* is a possible implementation of a capacity fulfilling all the constraints of this specification.

---

[1] `http://www.sarl.io`

The Janus platform[2] was redesigned and reimplemented in order to serve as the software execution environment of the SARL programs. Janus is designed in order to be a fully distributed platform over threads and a computer network. The execution unit in Janus is the event handler: the part of the SARL agent that is executed when a specific event is received. Each of these units are executed in parallel to the other units, even in the same agent.

The design of the Janus platform may cause issues for creating agent-based simulation applications. Indeed, several notions of time must be considered: user time (the real time, machine time) and simulated time. According to Lamport [1978], simulated time is a logical clock that induces a partial ordering of events; it has been refined in distributed context as logical virtual time by Jefferson [1985]. The Janus platform does not make any assumption on the ordering of the events that are exchanged by the agents. As a consequence it is impossible to use the Janus platform for agent-based simulation involving a time concept without providing the platform with a specific synchronisation mechanism. A model of such a mechanism is described in Section 4.4. Agents timestamp the event notifications they emit using their current perception of simulated time (the logical clock). They perceive each other behaviour as a sequence of events ordered by the logical clock. Agents can only emit events that comply to the Lamport partial order for logical time induced by the causality rule. In case agent A_0 uses information about agent A_1 notified by an event E_0 timestamped by $t(E_0)$, it can no longer notify any event E_1 that precedes E_0 in the partial order. This requires agents to synchronise their perception of the common logical clock.

Additionally, agent-based systems often include an *agent environment*, which is the software layer between the external world and the agents. This environment contains objects and resources, a.k.a. artefacts, that are not agents, but could be used by them. All the actions on the artefacts must be also synchronised in order to preserve the integrity of the agent environment state.

4.4 Event Synchronisation Model for SARL

In this Section, an event synchronisation model for the SARL programming language and its Janus execution environment is presented.

A *time period* is delimited by two discrete moments in (real or simulated) time. Each *moment in time* can be thought to bear a label which is the *timestamp*. In the

---

[2] `http://www.janusproject.io`

remaining part of the paper the terms *timestamp* and time period will be used interchangeably. Hence, the term *timestamp* is also used to identify the time period starting at the *moment in time* it is associated with.

According to the SARL metamodel, interaction among the simulation agents on one hand, and between the simulation agents and the agent environment on the other hand is supported by events. Each event e in the set $\mathbb{E}$ of events that are not already fired in the simulation agents is defined by a time stamped t_e and a content c_e. The timestamp t_e is the simulation time for which the event is fired. It is always greater or equal to the current simulation time t: $\forall e \in \mathbb{E}, e = \langle t_e, c_e \rangle \implies t_e \geq t$.

4.4.1 General Architecture

The proposed event synchronisation model is designed by considering the following three major assumptions and constraints (in bold face).

The **synchronisation process is hidden to the agents** by using the capacity and skill concepts. Indeed, the synchronisation process is related to the simulation and not to the simulation agent architectures and models. For example, the simulation agent models should be the same if they are instantiated during simulation or deployed on embedded computers. In order to enforce this characteristic, a proposal is made to provide skills that are implementing the standard interaction agent capability, which is provided by the SARL metamodel, with the proposed synchronisation mechanisms. This approach enables a clear distinction between the application-dependent models in the agents, and the simulator-dependent modules. It increases the level of abstraction that the framework will provide to application developers.

The **agent environment is part of the simulated system**. The agent environment as a key component of the system must be considered in the event synchronisation model. In this work, it is assumed that the agent environment is modelled with a complex hierarchy of agents, as proposed by Galland and Gaud [2015]. The root agent in this hierarchy represents the entire environment for the application logic layer (even if the environment is distributed over multiple environmental agents). In the context of this paper, and for simplicity reasons, two kinds of agents are used: (i) an *environment agent*, and (ii) a *simulation agent*, which represents the application logic's agent.

The **event synchronisation model follows a conservative synchronisation approach**. As explained in Section 4.2, two major approaches of synchronisation can be considered: conservative and optimistic. In order to select the best approach, the two types of interaction between the simulation agents and the environment are considered: (i) the simulation agents perceive the state of the environment; and (ii) the simulation agent acts in order to change the state of the environment. First, consider the data representing the perception of an agent at simulation time *t*: this needs to be extracted from the same state of the environment for all agents in order to ensure the consistency of the agents' behaviours for time *t*. Second, the simulation agents are supposed to act in the environment simultaneously and autonomously. Solving the joint actions of the agents requires to avoid them to directly change the environment's state. Agents are sending desires of actions, named *influences*, that are gathered and used by the agent environment in order to compute its next state. This approach is known as the influence-reaction model [Michel, 2007]. Because the agent environment may be modelled by means of a hierarchy of agents, according to Galland and Gaud [2015], the influence-reaction model may be locally applied if each subagent inside the environment is supporting a specific spatial zone. The influence-reaction model implies the introduction of at least one rendez-vous point during the simulation process: the agent environment is waiting for all the simulation agents to provide their influences. Besides the types of interaction, one can take into account the possible drawbacks of an optimistic approach. The optimistic approach will need a lot of space in order to store the different states of the simulation. This indicates as well that this approach will be application dependent because the used data structures are application specific which might induce burden to the designer/programmer. The type of applications for which this development is done needs a strict synchronisation between a lot of agents. The possibility of rollbacks will be very high and hence very time consuming. The bottle neck introduced with our conservative mechanism will probably cause less time loss in these cases. These considerations lead us to select a conservative approach in designing our event synchronisation model.

Synchronisation-Unaware Simulation Agent Architecture.

The general architecture for the simulation agents can be described by Algorithm 4.1. The simulation agents are able to react to `PerceptionEvent` events, which are fired by the agent environment to notify the simulation agent that its perception has changed. When the simulation agent has executed its reaction behaviour, it sends its list of desired actions to the agent environment by calling the `influence` function. This function is provided by the `EnvironmentInteractionCapacity` capacity (Algorithm 4.1), which represents the capacity of an agent to interact with its environment. The `EnvironmentInteractionCapacity` implementation will pack the influences into an occurrence of the `AgentIsReadyEvent` event, and send the latter to the agent environment. The simulation agent is also able to react to events that were fired by other simulation agents.

```
agent SimulationAgent {
  uses EnvironmentInteractionCapacity
  on PerceptionEvent {
    /* React on the perception receiving from the environment */
    [...]
    /* Send AgentIsReadyEvent to the environment */
    influence([...])
  }
  on Event {
    /* React on events from simulation agents */
    [...]
  }
}
capacity EnvironmentInteractionCapacity {
  def influence(desiredActions : Object*)
}
```

LISTING 4.1: General algorithm for the simulation agents and definition of the `EnvironmentInteractionCapacity` capacity.

All the agents in the SARL specification are provided with built-in capacities for which the execution platform provides the implementation. The first built-in capacity that is relevant to our synchronisation model is `Time`. It provides the functions for accessing the value of the current simulation time t. The second built-in capacity is `Behaviours`. It provides the `asEventListener` function, which replies the entry point for all events that are received by the agent. This capacity also provides the `wake` function to emit events inside the context of the agent itself. Specific implementation of these two capacities will be provided in Section 4.4.2 in order to integrate our synchronisation model in a way that is transparent to the simulation agent.

Synchronisation-Unaware Environment Architecture.

The general architecture for the simulation agents can be described by Algorithm 4.2. In this paper, it is considered that the agent environment can be modelled using a dedicated (holonic) agent according to the model proposed by Galland and Gaud [2015], in which the proposed agent represents the agent environment and is managing time evolution.

```
agent AgentEnvironment {
  uses TimeManager
  var expectedNumberOfInfluences : Integer
  var influences : List
  on StartSimulationStep {
    sendPerceptionsToAgents
  }
  on AgentIsReadyEvent {
```

```
    influences += occurrence
    if (influences.size == expectedNumberOfInfluences) {
appliesInfluencesToEnvironmentState
readyForTimeEvolution
    }
  }
}
```

LISTING 4.2: General algorithm for the agent environment.

The agent environment is waiting for the `StartSimulationStep` event that is fired by the platform's time manager[3]. When the event is received, the agent environment computes the agents' perception from the environment's state and sends `PerceptionEvent` events to the simulation agents. The implementation of the `sendPerceptionsToAgents` is application specific; it is not detailed in this paper. When receiving the `PerceptionEvent` occurrence, each simulation agent updates its knowledge with the timestamp of the event.

After sending the perception to the simulation agents, the agent environment is waiting for the agents' influences, according to the influence-reaction model [Michel, 2007]. When all the expected influences are received, the agent environment updates its state, and notifies the time manager that the simulation time t can evolve. Indeed, inside a simulation process including an environment as a whole entity, the simulation agent at time t can evolve according to the state of the environment [Michel, 2007; Weyns et al., 2007]. Basically, time evolution might be modelled by $t' := t + \Delta t$, where t is the current simulation time, Δt is a constant time evolution amount, and t' is the new simulation time.

Additionally, it is considered to dynamically determine the time increment using $t' := \min\{t_e | \forall e \in \mathbb{E}, t_e > t\}$. This approach is still vulnerable to deadlocks of simulation agents when they enter a deadlock or unexpectedly crash. In this case, the agent environment will wait infinitely for their response and hence the simulation will end in a deadlock as well. However this issue can be solved by using the machine time in order to detect a deadlock. The environment agent could keep track of the expected execution time per agent. When an agent exceeds this time, the environment agent could assume there is something wrong; the environment agent can proceed and ask the agent in deadlock to leave the simulation. In case a simulation agent wants to leave the simulation (deliberately stop, or crash) the environment agent is aware of that by listening the specific events fired by the execution platform and can update its list of agents to monitor.

---

[3] The time manager is a platform module or another agent that is storing and managing the time t over the simulation.

4.4.2 Conservative Event Synchronisation Mechanism

The event buffering is needed to ensure a pessimistic approach. The main idea is that simulation agents can send events to each other, but the events are not directly fired to the appropriate simulation agent. Hence, if a simulation agent decides to send an event to another simulation agent, this event is saved somewhere within the agent. This is done for every event that is sent for a given time period.

In the SARL specification, events may be received by an agent from another agent or from itself. In the first case, the `Behaviours` built-in capacity provides the agent's event listener that could be used for receiving the events. For supporting the second case, the `Behaviours` capacity provides the `wake` function for firing an event inside the context of the agent. For every simulation agent, the events that are received from other agents, or from itself are intercepted by a specific skill implementation of the `Behaviours` capacity. The intercepted events are kept in a bucket until a specific event of type `TimeStepEvent` that is representing a time step in the simulation is received. The `PerceptionEvent` event described in the previous Section is a subtype of `TimeStepEvent`. Consequently, when a simulation agent receives its perception from the agent environment, all the buffered events for the current simulation time are fired in the agent context as well as the `PerceptionEvent` whose occurrence advances the agent's time to the next timestamp. Algorithm 4.3 provides a SARL implementation of the specific skill. The internal class `InternalBuffer` is defined to represent the event buffer (defined as a multiple-value map).

If the received event e is not of type `TimeStepEvent`, e is buffered. Each event is mapped to a time interval with the *filter* function $(t_e \mapsto [t_i, t_{i+1}[)$ where $t_e \in [t_i, t_{i+1}[$ is the event timestamp and t_i and t_{i+1} are consecutive values of discrete time in the simulation. Hence for a given time t, the agent has to process a list of events $\{e | e \in \mathbb{E}, \mathit{filter}(t) = \mathit{filter}(t_e)\}$. If the event e is not explicitly timestamped, then the default timestamp is assumed to be equal to the time of the next simulation step (computed by the `nextTimeStep` function in Algorithm 4.3).

In a simulation agent, if the received event e is of type `TimeStepEvent`, the current simulation *time* is updated with the timestamp of e. Due to our conservative synchronisation approach, this timestamp is equal to the global time simulation. Additionally, the buffered events are consumed and fired into the current simulation agent by using the default event listener (provided by the execution platform). Finally, the `SynchronisationAwareSkill` implements the two functions of the `Behaviours` capacity that correspond to the two methods for receiving events: the `asEventListener` and `wake` functions.

```
skill SynchronisationAwareSkill implements Behaviours, Time {
  val defaultSkill : Behaviours
```

```
  val eventBuffer : EventListener
  var time : Integer
  new (platformSkill : Behaviors) {
    defaultSkill = platformSkill; eventBuffer = new InternalBuffer
  }
  def asEventListener : EventListener { return eventBuffer }
  def wake(e : Event) { eventBuffer.receiveEvent(e) }
  def getTime : Integer { return time }
  def nextTimeStep : Integer { return time + 1 }
  class InternalBuffer implements EventListener {
    val buffer : Map<Integer, Collection<Event>> = new MultiMap
    def receiveEvent(e : Event) {
if (e instanceof TimeStepEvent) {
  time = e.timestamp
  var events = buffer.remove(time)
  for (be : events) {
    defaultSkill.asEventListener.receiveEvent(be)
  }
  defaultSkill.asEventListener.receiveEvent(e)
} else {
  var timestamp = if (e instanceof TimestampedEvent) e.timestamp
  else nextTimeStep
  if (timestamp > time) {
    buffer.put(timestamp, e)
  }
}
    }
  }
}
```

LISTING 4.3: Skill implementation of behaviours and time capacities.

The second built-in capacity that must be overridden to enable event synchronisation is the `Time` capacity. This capacity provides the `getTime` function that is returning the current simulation time. In Algorithm 4.3, the local attribute `time` is defined, which is the local simulation time from the agent point of view. According to our conservative approach, this local time is updated with the global simulation time that is the timestamp of a received `TimeStepEvent` occurrence.

In order to use the previously defined `SynchronisationAwareSkill` skill, it must be given to the simulation agent as the skill to be used when the functions of `Behaviours` and `Time` are invoked. In order to ensure that the synchronisation process is hidden to the agents, the definition of the simulation agents cannot be changed. Algorithm 4.4 describes this discarded approach, which is based on the explicit creation of an instance of the `SynchronisationAwareSkill` skill, with the `Behaviours` skill from the platform as argument. This skill instance is mapped to the two capacities `Behaviours` and `Time`. From this point the agent is automatically synchronised with the rest of the system.

```
agent SimulationAgent {
```

```
    on Initialize {
      var syncSkill = new SynchronisationAwareSkill(getSkill(Behaviours))
      setSkill(syncSkill, Behaviours, Time)
    }
    [...]
  }
```

LISTING 4.4: Bad practice: explicit set of the synchronisation skill in the simulation agents.

A better approach is to install the `SynchronisationAwareSkill` skill when a another simulation-based skill is installed into the agent. The `EnvironmentInteractionCapacity` capacity is defined in Section 4.4.1. The corresponding skill may be defined in order to install the synchronisation skill when it is installed, as illustrated by Algorithm 4.5.

```
  skill SimulationEnvironmentInteractionSkill implements EnvironmentInteractionCapacity
    {
    def install {
      var syncSkill = new SynchronisationAwareSkill(getSkill(Behaviours))
      setSkill(syncSkill, Behaviours, Time)
    }
    [...]
  }
```

LISTING 4.5: Good practice: installing the synchronisation skill from another simulation skill.

4.5 Performance

In order to be able to measure the performance without being biased by application related calculations, a very simple ping-pong application is created.

In the time period starting at T_0, every agent has 20 % probability to emit a *ping* message to X other agents where $X \sim Uniform(1 : 100)$. The message needs to be delivered in the time period $T_d \sim Uniform(T_1 : T_e)$ where T_e denotes the end of simulated time. The measured time is illustrated in Figure 4.1, where T_0 and T_1 denote the start and the end of the interval; T_a denotes the end of the reception of the events sent by the environment to the simulation agents; T_b the end of "application level payload work" done by the agents, and finally T_c the end of delivering the `AgentIsReadyEvent` to the environment. For every time period, the number of emitted messages is computed together with the total amount of time needed to execute this time period. Experiments are realised for 200 agents on a Linux Ubuntu 14.04LTS laptop with 8 GB memory and a Intel Core i5-4210M CPU 2.60 GHz × 4. The number of time periods that are simulated is 2,500.

Experimental results are illustrated in the graph represented in Figure 4.2.

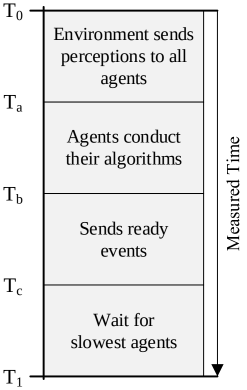


FIGURE 4.1: An overview of the time measurement in the experiments.

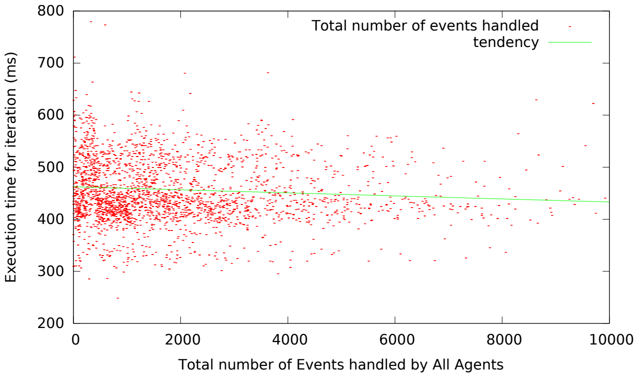


FIGURE 4.2: Graph that represents the total number of events handled in a specific iteration (x-axis) against the total execution time for that iteration in ms (y-axis) for the case of 200 agents and 2,500 iterations.

In the experiments, all the agents have the same actions to do. Consequently, they have approximately the same execution time. It is clear to see that the execution time follows a constant tendency, and hence seems to be independent of the number of processed events over the full range of observations. The execution time for a single period between two consecutive increments of simulated time includes: perception of the environment, application specific *payload* work and end-of-period notification. The duration required for the payload work in the experiment is negligible. The large variance of the execution time masks the expected dependency on the number of events.

4.6 Conclusion and Perspectives

A proof of concept is given for the support of the event synchronisation using the SARL language and its Janus execution platform, without changing neither the specification of SARL nor the code of the Janus platform. Similar to Weyns and Holvoet [2003], the plan is to refine this model by including regional synchronisation. Another perspective is to provide an optimistic synchronisation model. From a technological point-of-view, this synchronisation mechanism will be included into the Janus execution platform.

Acknowledgements

The research reported was partially funded by the IWT 135026 Smart-PT: Smart Adaptive Public Transport (ERA-NET Transport III Flagship Call 2013 "Future Travelling").

Bibliography

L. Braubach, A. Pokahr, W. Lamersdorf, K.-H. Krempels, and P.-O. Woelk. A Generic Simulation Service for Distributed Multi-Agent Systems. In *From Agent Theory to Agent Implementation (AT2AI'04*, pages 576–581, 2004.

G. Cich, S. Galland, L. Knapen, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Addressing the Challenges of Conservative Event Synchronization for the SARL Agent-Programming Language. In Yves Demazeau, Paul Davidsson, Javier Bajo, and Zita Vale, editors, *Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection*, pages 31–42, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59930-4.

R. M. Fujimoto. Parallel Discrete Event Simulation. *Communications of the ACM*, 33 (10):30–53, 1990.

R. M. Fujimoto. *Parallel and Distributed Simulation Systems*. Wiley, New York, 2000.

S. Galland and N. Gaud. Organizational and Holonic Modelling of a Simulated and Synthetic Spatial Environment. In D. Weyns and F. Michel, editors, *Agent Environments for Multi-Agent Systems IV*, pages 147–169, Cham, 2015. Springer International Publishing. ISBN 978-3-319-23850-0. doi: 10.1007/978-3-319-23850-0.

S. Jafer, Q. Lui, and G. Wainer. Synchronization Methods in Parallel and Distributed Discrete-Event Simulation. *Simulation Modelling Practice and Theory*, 30:54–73, 2013.

D. R. Jefferson. Virtual Time. *ACM Trans. Program. Lang. Syst.*, 7(3):404–425, July 1985. ISSN 0164-0925. doi: 10.1145/3916.3988.

L. Lamport. TI Clocks, and The Ordering of Events in a Distributed System. *Communications of the ACM*, 21:558–565, 1978.

F. Michel. The IRM4S Model: The Influence/Reaction Principle for Multiagent Based Simulation. In *Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS07)*, Honolulu, Hawaii, USA, 2007. ACM. ISBN 978-81-904262-7-5. doi: 10.1145/1329125.1329289.

K. S. Perumalla. Parallel and Distributed Simulation: Traditional Techniques and Recent Advances. In *Proceedings of the 2006 Winter Simulation Conference*, pages 84–95, Monterey, CA, 2006.

K. S. Perumalla and R. M. Fujimoto. Virtual Time Synchronization over Unreliable Network Transport. In *Proceedings of the 15th International Workshop on Parallel and Distributed Simulation*, pages 129–136, Lake Arrowhead, CA, 2001.

S. Rodriguez, N. Gaud, and S. Galland. SARL: A General-Purpose Agent-Oriented Programming Language. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 103–110, Aug 2014. doi: 10.1109/WI-IAT.2014.156.

C. Tropper. Parallel Discrete-event Simulation Applications. *Journal of Parallel and Distributed Computing*, 62(2):327–335, 2002.

D. Weyns and T. Holvoet. Model for Situated Multi-Agent Systems with Regional Synchronization. In *Concurrent Engineering, Agents and Multi-agent Systems*, pages 177–188, Madeira, Portugal, 2003.

D. Weyns, A. Omicini, and J. Odell. Environment as a first-class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, 2007. ISSN 1387-2532.

Y. Xu, W. Cai, H. Aydt, M. Lees, and D. Zehe. An Asynchronous Synchronization Strategy for Parallel Large-scale Agent-based Traffic Simulations. In *Proceedings of the 3rd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM PADS '15, London, United Kingdom, June 2015.

Chapter 5

Modelling Demand Responsive Transport Using SARL and MATSim

The peer-reviewed work presented in this chapter was published on June 12, 2017 in the Procedia Computer Science [Cich et al., 2017]. The work was presented at the International Workshop on Agent-based Modelling and Applications with SARL (SARL 2017) in Madeira, Portugal held on May 16-19, 2017.

Keywords: Co-Simulation; SARL; MATSim; Agent-Based Modelling; Demand Responsive Transport; Thin Flows; Dynamic Vehicle Routing

Abstract

Demand responsive collective transportation might be a solution to serve *thin flows* that occur when the average demand per time unit for travel between particular locations is small. Small capacity and/or specially equipped vehicles are deployed to serve low population density areas and mobility impaired passengers. The variability of the demand and the low vehicle capacity require daily optimal planning of routes. The cost for daily tours heavily depends on the temporal and spatial distribution of the demand. This paper proposes a co-simulation model to evaluate the profitability of thin flow service providers over multiple years of operation under the specific condition of public compensation (subsidising).

5.1 Introduction

Public transport (PT) is a widely deployed transportation mode over the world. In Flanders (the northern part of Belgium) PT by bus is based on the concept of "*basic mobility*". This concept distinguishes between different kinds of areas, such as metropolitan, urban, small towns and outskirts. For every area, rules specify the amplitude and frequency of PT as well as the maximum distance between homes and bus stops. "Basic mobility" aims to ensure that every citizen of Flanders is offered a minimum PT service level. This goal was generous, but turns out to be very expensive and hence, not achievable to cover the whole of Flanders especially for so called *thin flows* where the demand is low due to spatial dispersion (areas of low population density) or to special requirements (mobility impaired people needing support). Some *thin flow* customers have special needs that are not served by regular PT. An elaborated definition of *thin flows* is given by Cich et al. [2016] (Chapter 3).

Therefore, the Flemish government is currently developing a new legislation called "*basic accessibility*". The main idea is that PT cannot provide every transport for every traveller. PT should still be one of the main transportation modes, but it should be complemented with other transport means.

In this paper the focus is on Demand Responsive Transport (DRT) that serves both regular and mobility impaired (possibly wheelchair bound) people in thin flows in order to increase efficiency while aiming to fulfil the "basic accessibility" requirements. For mobility impaired travellers part of the cost for the driven distance can be compensated (subsidised). Two main questions arise: (i) Can those DRT services substitute particular PT services? (ii) Under which subsidy conditions are DRT services for thin flows viable over a long term period?

In order to be able to answer those questions, a *micro-simulator* is developed to evaluate proposed solutions. In this paper, it will be discussed how the evaluation scenarios using the agent-based framework SARL and the multi-agent transport simulator MATSim in an integrated software package is modelled. While writing this paper, development is still going on, hence no results can be presented yet.

The remainder of the paper is organised as follows: in Section 5.2 an overview of relevant literature regarding co-simulation is given. Section 5.3 describes the *demand* and *supply* models. Section 5.4 discusses design aspects related to co-simulation. Finally, a conclusion is drawn and future work is presented in Section 5.5.

5.2 Related Work

In order to avoid "reinventing the wheel", building complex microscopic DRT models consists in combining different smaller models, each responsible for modelling one or more components of the overall system. Depending on a particular case, these components can be a transport network, traffic, fleet operations, customer service, land use, electricity grid, etc. This co-simulation approach has been frequently applied to create tools aimed at solving different transport problems. A selection of related research is presented in this section.

Nicolai and Nagel [2015] describe the process to create an interaction between MATSim and UrbanSim. UrbanSim is a microscopic urban simulation model that models long term land-use evolution. It will call MATSim in regular time intervals in order to update different kinds of indicators. When MATSim is done with the calculations, the control is given back to UrbanSim which will update their data sets. This will proceed until the desired period is simulated. The paper describes the data requirements for both simulators as well as the conversions of the data between MATSim and UrbanSim. Co-simulation was applied to Brussels and Zürich.

Waraich et al. [2013] conduct a similar approach. They use MATSim together with PEV Management and Power System Simulation (PMPSS) in order to compare the efficiency of different PHEV schemes. They add this PMPSS step into the MATSim loop; when MATSim reaches an equilibrium, the PMPSS is conducted and several checks are done regarding the electric grid. If there are violations, a new MATSim run is started.

Literature reports on efforts to combine *microscopic* and *mesoscopic* simulations, frequently referred as *hybrid model*s (Burghout and Wahlstedt [2007]; Casas et al. [2011]). The main challenges in those simulations are the boundary regions between the microscopic and the mesoscopic simulation. When a car leaves the mesoscopic simulation, it shall be transformed to the microscopic simulation. Therefore, constraints need to be checked such as whether there is room for that specific car in the microscopic simulation. In order to make the models reliable, one shall take care of several requirements including consistency in network representation, transparent communication and data exchanges. Burghout and Wahlstedt [2007] propose a hybrid model that combines the mesoscopic model Mezzo and the microscopic model VisSum. In their framework, Mezzo controls the synchronisation. Their case study was Stockholm for which they simulated Stockholm mesoscopically and a small portion on the southeast border of central Stockholm microscopically.

None of the research mentioned above models negotiation between actors (which is an essential part in our model). Chun and Wong [2003] describe a sound model for direct negotiation between actors of different kinds.

5.3 Modelling Demand Responsive Transportation

Evaluating viability of DRT providers requires micro-simulation; methods based on aggregation are inappropriate because averaging demand ignores the effects of distribution in the temporal and/or spatial dimensions. Spatial and temporal variability heavily affect the outcome of the optimisation carried out by the DRT providers. This in turn induces negotiation about service timing between customers and providers. Furthermore, rules about subsidies depend on customer properties, location and the availability of alternative solutions (accessible PT). Those terms will be categorised by *labels*. Because of the need to model detailed interactions between classes of actors, the SARL framework is chosen.

Our model simulates two main concepts, namely (i) *demand* and (ii) *supply* for a time period of several months by using an *activity based model* to predict a weekly schedule for each agent and assuming that history is periodic. The application is divided in two parts: (i) *Thin Flows Travel Model* (TFTM) including *negotiation* between agents (demand and supply) to agree about trip details (which implies the need for schedule adaptation) and (ii) *Operational Travel Model* (OTM) including efficient scheduling of trips to be served which is typically done by a Vehicle Routing Problem (VRP) algorithm. An overview of the concepts is given in Figure 5.1.

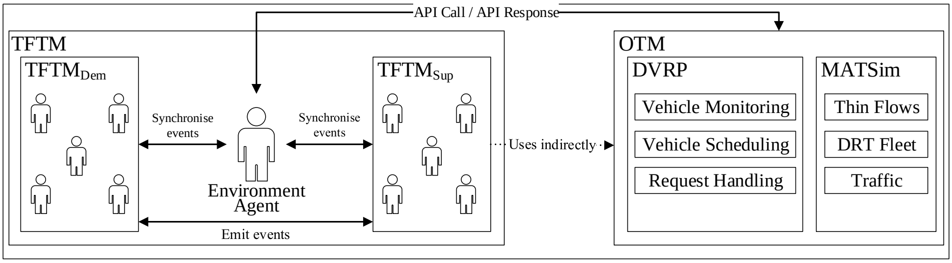


FIGURE 5.1: An overview of the software. The Environment Agent is responsible for managing time and for communication with OTM. Agents in TFTM$_{Dem}$ and TFTM$_{Sup}$ can communicate with each other. Note the dashed line between TFTM$_{Sup}$ and OTM, indicating that TFTM$_{Sup}$ makes use of OTM (not directly, but through the Environment Agent).

5.3.1 Thin Flows Travel Demand Model (TFTM$_{Dem}$)

The demand model TFTM$_{Dem}$ represents *customers* belonging to the *thin flows* who want to execute schedules. In order to do so, they need to arrange transport. Transport can be done by own means, such as by foot, by bike, by car, etc. or it can be requested from the *supply* side.

TFTM$_{Dem}$ models customer behaviour and negotiations leading to *thin flow travel generation* and is implemented using the general-purpose agent-oriented programming language SARL described in Rodriguez et al. [2014].

Customers can (i) ask for $N \geq 1$ different proposals for multi-leg (possibly multi-modal) trips, (ii) wait for $M \in [1, N]$ proposals before deciding which option to choose, (iii) refuse some proposals (mostly based to timing or cost constraints) and (iv) require sequences of chronologically non-contiguous trips (both the back and forth trips) and consider those as an atomic request. This implies the need for mutual commitment. Because a provider can subcontract trips, multi-level commits are required. Such mechanism implies the need for several minimal and maximal delays that apply to both customers and suppliers respectively (see Section 5.3.6).

5.3.2 Thin Flows Travel Supply Model (TFTM$_{Sup}$)

The supply side consists of providers that provide some kind of transport such as taxis, public transport, etc. Such transport services can be microscopically simulated with OTM. The negotiation model for the transport providers and preprocessing of requests are implemented in SARL. However, OTM is provided by an external simulator. MATSim and in particular its *Dynamic Vehicle Routing Problem (DVRP)* module is used to model daily operations for taxi like DRT providers. The requests from TFTM$_{Dem}$ to TFTM$_{Sup}$ are preprocessed by TFTM$_{Sup}$ in order to filter and (if necessary) to reject the request before sending it to the DVRP module of MATSim. This approach is twofold: (i) reduce the processing time of the time consuming OTM call and (ii) from the customer's perspective, all the knowledge about constraints etc. resides in the TFTM model, whereas OTM is aware of constraints related to fleet operations. The constraints on the customer's side include the time windows described in Section 5.3.6 and the labels. Every agent is qualified by a set of *labels*. Customer labels specify home address, age, income level and *mobility support* requirements (e.g. wheelchair, visual impairment). Supplier labels specify available mobility support. Personal data are used to determine the subsidising level and hence, the fare for a trip (using rules set by law). The preprocessing will deal with this *label matching*, e.g. if a customer is wheelchair bound, he needs a provider that owns at least one free vehicle that can transport a wheelchair.

5.3.3 Operational Travel Model

The main goals of OTM are twofold: (i) microscopic simulation of all components of the analysed transport system, including thin flows, DRT fleet, traffic, etc. (provided by MATSim [Horni et al., 2016]), and (ii) dynamic vehicle routing that enables suppliers to monitor and schedule vehicles, and to handle incoming requests (provided by MATSim's DVRP extension [Maciejewski, 2016]). DVRP adds to MATSim the concept of DynAgents whose plans can be changed at any moment, which is crucial for simulation of on-demand transport services, where both supply and demand are dynamic and stochastic. DVRP's routing algorithm listens to events emitted by MATSim and reacts to changes in the transport system. It also communicates with TFTM via a co-simulation protocol described in Section 5.4.

In the current version of the SARL-MATSim integration, the DRT service is offered by fleets of non-shared taxis, and each fleet belongs to a different provider. Providers monitor, schedule and route their taxis in real time. Request submission events coming from TFTM to OTM are translated into taxi requests, and answered according to the protocol. On the other hand, events generated on the OTM side, e.g. taxi pick-up or drop-off, are transmitted back to TFTM. Since the taxi service is used as a mean of DRT and therefore taxis may serve parts of longer multi-modal trip chains, there are several certain constraints imposed on the taxi dispatching algorithm, such as provision of the drop-off location on request submission, support of advance requests, or request rejection (by a provider) and cancellation (by a customer).

5.3.4 Trip Sequence Composer

The intelligence for both the customers and the suppliers is modelled by a *Trip Sequence Composer* (TSC). The TSC cooperate in a multi-level demand-supply chain while requiring alternative proposals. Therefore, tree structures describing requests and proposals respectively are passed back and forth between TSCs. More information can be found in Cich et al. [2016] (Chapter 3)

5.3.5 Negotiation

Negotiation between two or more parties applies to a pair (or vector) $\mathbf{V}$ of variables for which a value needs to be agreed upon. Successful negotiation requires the existence of a *value pair* $\mathbf{Q_V}$ with $\mathbf{Q_V}[i] = value(v_i)$ that is acceptable to each participant. Each variable can be continuous or discrete. For a continuous variable $v_i \in \mathbf{V}$ the range of values R_{v_i} acceptable to a given individual j can consist of disjoint intervals.

Each value pair $\mathbf{Q_V}$ corresponds to a (scalar) utility $U_j(\mathbf{Q_V})$ for individual j. Chun and Wong [2003] focus on negotiation to establish a schedule for cooperation and propose that each individual j acts as a utility maximiser; its utility function U_j is unknown to each individual $k \neq j$. However, each participant computes and shares a normalised *preference level*. Each individual estimates the preference level of the other party and emits proposals that maximise its own utility and the preference level of the other party (in order to get the proposal accepted). In this context both customers and suppliers act as *negotiating individuals*. The dependence $U_j(\mathbf{Q_V})$ is defined by the respective behavioural model.

Negotiation is coded in the behaviour of a customer. If a customer does not agree with a proposal, he will deny this proposal and send a new request with adapted parameters based on the previous proposal. This process will continue until a desired proposal is found or until the customer decides to stop trying.

5.3.6 Time windows

The time windows that apply to the negotiation between customers and suppliers are shown in Figure 5.2. The messages that are actually exchanged during this process can be found in Figure 5.3. Note that a single trip can consist of multiple legs served by different providers each having their own time windows.

TW 1 : Maximum period between reception of a request by a company for a *trip* and the earliest departure time specified in the request: specifies *earliest request emission time*.

TW 2 : Minimum period between reception of a request by a company for a *trip* and the earliest departure time specified in the request: specifies *latest request emission time*.

TW 3 : Maximum period between reception of a request for a *trip* by a company and notification of acceptance or denial (rejection): specifies *latest proposal/denial emission time*.

TW 4 : Maximum period to leave a proposal unanswered: specifies *latest acceptance/rejection time*.

TW 5 : Maximum period to leave the acceptance of a proposal open before sending an *commitment*: specifies *latest commitment/denial time*.

TW 6 : Minimum period between sending the commitment or cancellation to the company and the earliest departure time of the trip specified in the request: specifies *latest commitment/cancellation time*.

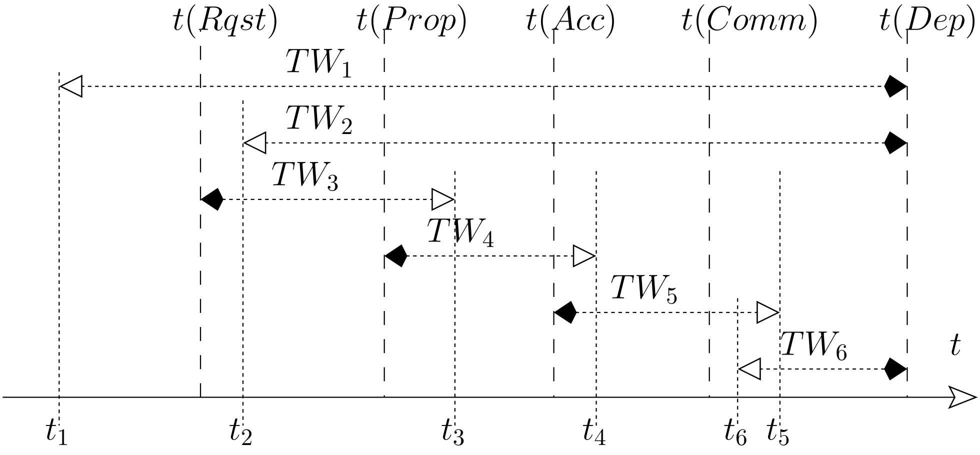


FIGURE 5.2: Overview of events and time intervals. The black symbol on an arrow identifies the reference time. No time-out does occur in this situation. Actual message transmission is assumed to be instantaneous (emission and reception of a message coincide in the diagram). Labels at the top denote events. Label t_i at the bottom denotes the deadline induced by TW_i.

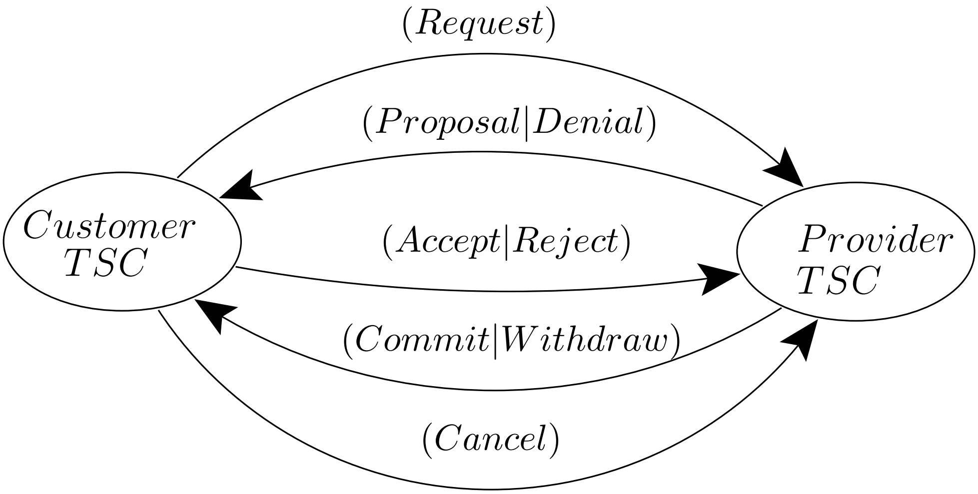


FIGURE 5.3: Overview of messages exchanged (events).

5.4 Co-Simulation Protocol

5.4.1 Simulated Time Management

The SARL and the underlying Janus framework do not have a notion of simulated time. TFTM will be the coordinator of the simulation. A dedicated single *environment agent* in TFTM is responsible for managing progress of simulated time and for the communication with OTM. As soon as no agent needs to perform any more action, simulated time is advanced to the first moment at which at least one agent will perform an action. Hence, simulated time progresses in a non-monotonic way between discrete values. Let t_i and t_{i+1} denote consecutive simulated time values. All messages emitted by agents in period $p_i = [t_i, t_{i+1}[$ will be received only in p_{i+1}.

In particular, while simulating a specific time period, agents will generate trip requests, i.e. a request to go from A to B with information about time windows to get picked up, to be dropped off, etc. These trip requests will be collected. Hence, at the end of a time period, there is a list of queries for OTM (MATSim) available. These queries will be converted to a single JSON[1] object. This JSON object is sent to OTM which will process the individual queries by simulating MATSim to the same time period as TFTM. A closed loop approach is used which means that for every time period p_i, first TFTM will be executed. When TFTM is finished, OTM will execute the same time period and return its output back to TFTM. Important to note here is the fact that TFTM and OTM run in two separate Java Virtual Machines. The communication is done by *sockets*.

5.4.2 Message Exchange

TFTM$_{Sup}$ uses OTM (in this case the DVRP module in MATSim) to simulate the transport of each individual provider. The main advantage of this method is twofold: (i) customers will be assigned to vehicles as efficient as possible and requests will be denied if the provider cannot find an efficient/profitable solution at the operational level and (ii) vehicles will encounter congestion when they pick up and drop off customers. Trips can be requested and the corresponding proposals can be rejected or accepted and finally committed. Message exchange details are out of scope of this paper.

---

[1] `http://www.json.org/`

5.5 Conclusion and Future Work

In this paper, a framework to simulate supply and demand especially for *thin flow* demand is described. A cooperation between the multi-agent model TFTM implemented using the SARL framework and the agent-based micro-simulator MATSim is proposed. TFTM is the main simulator that is responsible for increasing time and provide negotiations between agents while MATSim (OTM) is responsible for scheduling trips, i.e. assigning customers to vehicles as efficient as possible. OTM is a general concept and hence, it can be extended by other "smart" software. In future research, the plan is to integrate PT as well. OTM will be extended by "OpenTripPlanner" [2]; this software is able to give information about multi-modal PT alternatives.

Acknowledgements

The research reported was partially co-funded within ERA-NET Transport III Flagship Call 2013 "Future Travelling", project: "Smart-PT - Smart Adaptive Public Transport", under grant IWT 135026 (Belgium) and NCBR ERA-NET-TRANSPORT-III/2 /2014 (Poland).

Bibliography

W. Burghout and J. Wahlstedt. Hybrid Traffic Simulation with Adaptive Signal Control. *Transportation Research Record: Journal of the Transportation Research Board*, 1999: 191–197, 2007. ISSN 0361-1981. doi: 10.3141/1999-20.

J. Casas, J. Perarnau, and A. Torday. The Need to Combine Different Traffic Modelling Levels for Effectively Tackling Large-Scale Projects Adding a Hybrid Meso/-Micro Approach. *Procedia - Social and Behavioral Sciences*, 20:251–262, 2011. ISSN 1877-0428. doi: 10.1016/j.sbspro.2011.08.031.

H. W. Chun and R. Y. M. Wong. N* - An Agent-Based Negotiation Algorithm for Dynamic Scheduling and Rescheduling. *Advanced Engineering Informatics*, 17(1): 1–22, 2003. ISSN 1474-0346. doi: 10.1016/S1474-0346(03)00019-3.

G. Cich, L. Knapen, S. Galland, J. Vuurstaek, A. Neven, and T. Bellemans. Towards an Agent-based Model for Demand-Responsive Transport Serving Thin Flows. *Procedia Computer Science*, 83:952–957, 2016. ISSN 1877-0509. doi: 10.1016/j.procs.

---

[2] `http://www.opentripplanner.org/`

2016.04.191. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.

G. Cich, L. Knapen, M. Maciejewski, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Modeling Demand Responsive Transport using SARL and MATSim. *Procedia Computer Science*, 109:1074–1079, 2017. ISSN 1877-0509. doi: 10.1016/j.procs.2017.05.387. 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017) / Affiliated Workshops.

A. Horni, K. Nagel, and K.W. Axhausen, editors. *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press, London, 2016. doi: 10.5334/baw. License: CC-BY 4.0.

M. Maciejewski. Dynamic Transport Services. In Horni et al. [2016], chapter 23. doi: 10.5334/baw. License: CC-BY 4.0.

T. W. Nicolai and K. Nagel. Integration of Agent-Based Transport and Land Use Models. In M. Bierlaire, A. de Palma, R. Hurtubia, and P. Waddell, editors, *Integrated Transport and Land Use Modeling for Sustainable Cities*. EPFL Press, 2015. ISBN 978-2-940222-72-8.

S. Rodriguez, N. Gaud, and S. Galland. SARL: A General-Purpose Agent-Oriented Programming Language. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 103–110, Aug 2014. doi: 10.1109/WI-IAT.2014.156.

R. A. Waraich, M. D. Galus, C. Dobler, M. Balmer, G. Andersson, and K. W. Axhausen. Plug-In Hybrid Electric Vehicles and Smart Grids: Investigations Based on a Microsimulation. *Transportation Research Part C: Emerging Technologies*, 28:74–86, 2013. ISSN 0968-090X. doi: 10.1016/j.trc.2012.10.011. Euro Transportation: selected paper from the EWGT Meeting, Padova, September 2009.

Chapter 6

Viability Evaluation of Transport Supply for Thin Flows with Agent-Based Simulation

The work presented in this chapter was submitted on August 22, 2018 to the European Journal of Transport and Infrastructure Research [Cich et al., 2018].

Keywords: Thin Flows, Agent-Based Modelling, Transport Supply, Mobility Impaired People, Demand Responsive Transport, Public Transport

Abstract

in regions where there is a low transport demand, time-table based public transportation is a problem. This is characterised by *thin flows*. Currently in Flanders, efforts are done to solve that problem. Combi-mobility is a key concept where public transport should be combined with other transport services such as *Demand Responsive Transport*. This paper aims to simulate this situation in an agent-based microsimulator. The goal is to determine the economic viability of transport providers in the context of thin flows. Results show that profits of the transport providers highly depend on the fleet size and fare prices. However fleet sizes also have a huge impact on the possibility to travel for the customers. Results showed that providers could be viable in some circumstances.

6.1 Introduction

The research covered by this paper aims to determine the economic viability of providers in a *Demand Responsive Transportation (DRT)* context. In particular, transport requirements by mobility impaired people may be difficult to serve in an economically profitable (hence, sustainable) way. Agent-based simulation is used to answer the research questions.

The operation of time-table based public transportation is problematic in cases where insufficient customers are attracted. Those cases are characterised by *thin flows*. Intuitively, *thin flows* are small sets of travellers moving from an origin to a destination during a particular sequence of recurring time intervals (e.g. every Monday morning). A formal definition is given in Section 6.4.2. Thin flows constitute a problem for the organisation of collective and public transportation from the point of view of economic efficiency (services profitability). One of the reasons is the difficulty to predict the demand (the size of the thin flow, the number of travellers in a particular period of time). The comfort of large aggregated stochastic values having relatively low variance is missing. The optimal solution for nearly every trip of a transportation provider heavily depends on the decisions made by each individual customer. Thin flows not only occur in regions with low population density or scattered housing. They also do occur where customers have particular requirements due to mobility impairment: such requirements may cause the customers to be unable to use regular public transportation. In Flanders (Belgium), mobility impaired people can be granted partial subsidising of travel cost in case they make use of accredited special services. Details about the rules specified in the so-called *compensation decree* are described in Section 6.3.2. The decree specifies the tariffs (fares), the fraction of the cost that qualifies for subsidy and the rules to become an *accredited transport provider*. The economic viability of a service is evaluated by micro-simulation. This paper introduces the development of a synthetic environment (software tool) consisting of interacting customers (travellers) and transport providers. Each of them is characterised by a behavioural model that is aimed to mimic reality as close as possible. Simulations will be executed for a period of simulated time to estimate the costs and benefits of the transport providers under the conditions specified in the *compensation decree* and for a synthetic population that is statistically similar to the Flemish one w.r.t. the marginal distributions for several socio-economic properties.

Social exclusion due to transport problems is already widely discussed in the literature [Cass et al., 2005; Lucas, 2012; Preston and Rajé, 2007; Mackett and Thoreau, 2015; Delbosc and Currie, 2011]. Authors use or give different definitions, but the main thought needs to be taken into account in this paper is that mobility impaired people cannot perform the activities they want to do due to lack of transport. The physical inability to use public transport, the absence of public transport stops or the inability to pay for a taxi are reasons of lack of transport. However being wealthy does not mean you are socially included. In this paper the main focus is on a subset of *thin flows*, namely elderly and mobility impaired people.

For the focus group in this paper, using public transport is not always easy. Depending on the mobility impairment, people can suffer from difficulties such as (i) alighting or boarding public transport, (ii) planning a public transport trip including intermediate connections or multiple modes, (iii) reading public transport time tables or (iv) knowing at which stop the bus is currently stopping.

Alternatives such as taxis or DRT are in most of the cases too expensive for this focus group. The Flemish Government did already some effort in an attempt to solve this issue. An overview of the current situation in Flanders can be found in Section 6.3.

The remaining of the paper is organised as follows: in Section 6.2, a broad overview of related work is discussed. Next, in Section 6.3, the current situation regarding important mobility related legislation in Flanders, Belgium is highlighted. In Section 6.4, important principles of the simulator that is used in the paper are discussed. In Section 6.5, agent-based modelling and the reasons why it is used are explained. Simulation details are presented in Section 6.6, followed by an explanation about the used data in Section 8.5. The actual experiment is handled in Section 6.8. The limitations and future work are discussed in Section 6.9. The paper ends with a conclusion and discussion in Section 6.10.

6.2 Related Work

The following subsections focus on important aspects of respectively *demand* and *supply* related to thin flows.

6.2.1 Demand: Mobility Impairment

Despite the paper will focus on elderly and mobility impaired people, literature shows that not all elderly people are mobility impaired. Research of Haustein [2012] investigated a 60 plus population in Germany. Based on different scales, the participants where classified in four categories: (i) captive car users, (ii) affluent mobiles, (iii) self-determined mobiles and (iv) captive public transport users. These categories can be used as starting point for intervention strategies since people in the same category have a matching profile. Eventually the goal is to shift more people from car to Public Transport (PT). Other research from Shrestha et al. [2017] investigated public transport needs of elderly people. They used the five GOAL profiles (i) fit as a fiddle, (ii) hole in the heart, (iii) happily connected (iv) oldie but a goody and (v) care-full. Together with the important PT requirements: (i) affordability, (ii) availability, (iii) accessibility and (iv) acceptability. It is clear that a lot of older people are capable of driving and hence do not need public transport. However, at some point elderly people will switch from car to public transport. As a consequence, PT should be adapted to the previously mentioned requirements. Both researches show that elderly can be categorised according to their abilities.

There is a large variety of mobility impairments. Research about vision impairment was conducted by Gallagher et al. [2011] and Fürst and Vogelauer [2012]. While hearing impairment was also investigated in the research of Fürst and Vogelauer [2012]. People with a vision impairment suffer from a lot of problems while travelling by public transport as well as by taxi. They encounter difficulties with lack of information about time tables, spoken announcements etc. For hearing impaired people, unclear spoken announcements were a problem as well. Furthermore, guidance dogs are not always welcome.

6.2.2 Supply: Paratransit and Demand Responsive Transport

Sammer et al. [2012] conducted a survey in Austria. Since their survey was about mobility impairment, the target audience were so called "hard-to-reach persons". As a consequence, only 541 persons participated. They individually interviewed every participant (45 to 90 minutes). The survey used 15 different kinds of mobility impairments. Questions were about encountered problems while travelling, use of mobility aids, additional burdens etc. One of their main findings was that far more people felt impaired in their mobility than those who are traditionally considered as impaired. According to Sammer et al. [2012], it was even more than one third of the population.

Research about DRT (whether or not used as a feeder service for public transport) is already widely conducted in other researches; either with questionnaires and literature review or with simulations. Nelson et al. [2010] give an overview of Flexible Transport Systems (FTS) in Europe and North America. At different locations such as Sweden, Finland, Belgium, Italy, United Kingdom etc. initiatives were taken regarding FTS and DRT. In general those initiatives were dedicated for special groups such as mobility impaired people and in most of the cases in parallel with already existing public transport. This vision is outdated and is currently evolving in services for the whole community and as feeder services for public transport.

Nguyen-Hoang and Yeung [2010] conducted a benefit-cost analysis for paratransit in the United States by estimating demand and cost models. Their research demonstrated that the overall level of service has an impact on the demand for paratransit. There was also an extremely price-inelasticity. It demonstrates that the paratransit users need this type of transport, because they have very few alternatives.

Jain et al. [2017] did an extensive literature review about DRT. In their research they wanted to predict the susceptibility to use DRT in specific regions in the Greater Melbourne region without doing a questionnaire. They used demographic and trip characteristics together with information about existing DRT services around the world. Based on this information they extracted eleven parameters which could affect the usability of DRT. The following characteristics were found: (i) age groups of 15-24 and 55 or more, (ii) trips of females, (iii) not in workforce, (iv) no driving license, (v) low household vehicle ownership (0 or 1), (vi) low household income, (vii) single person household, (viii) lack of train station proximity (ix) shopping and social trips (x) higher trip waiting time and (xi) higher trip walking time. These eleven characteristics were used to give each region a score based on an *extreme value* approach. Regions with a high score were more susceptible for use of DRT than others. Validating the results of DRT was not possible since there was no data available. However they validated their approach by conducting the same technique, but this time for public transport.

Lee and Savelsbergh [2017] implemented an algorithm for a Demand Responsive Connector (DRC) service. In a DRC service, customers are picked up from their addresses and transported to transit hubs from which they can travel to their final destination with public transport. The authors studied a more flexible, extended version of DRC, where customers can be transported to alternative hubs instead of the traditional regional DRC where customer are brought to predefined hubs. An integer programming and a heuristic method are proposed. They concluded that the extended version could be beneficial, but it is depended on (i) passenger density, (ii) number of stations and (iii) size of the flex window. Their idea was minimising operating costs instead of minimising vehicles, however the differences in number of vehicles will in most of the cases be the same. Promising results are presented.

Shen et al. [2017] developed a two-stage algorithm for a Demand Responsive Connector service with on demand stations. The algorithm aims at minimising the system cost (both customers as providers). The algorithm was tested in Nanjing city, China. Results show that the proposed algorithm could be used for trips generated around the transfer points.

Ronald et al. [2015] attempt to explore the viability of DRT services before the actual implementation. They used demand from a real-world scheme. The existing fixed-time schema and a new ad hoc scheme were modelled in MATSim. Optimisation algorithms for passengers as well as for operators were used. Both optimisation algorithms had different outputs, however less difference was observed in areas of low demand. For an ad-hoc system, high vehicle kilometres travelled were observed when the demand was high. Although the study gave promising results, some limitations were reported; user preference and mode choice were ignored.

6.3 Current Situation in Flanders

6.3.1 Basic Mobility versus Basic Accessibility

In Flanders, public transport follows the principle of *basic mobility* [De Lijn, 2002; Flemish Government, 2002]. It is based on the "Decree concerning the organisation of passenger transport by road"Flemish Government [2001] and it makes sure that there is a basic supply off public transport. Three criteria decide about basic mobility: (i) the maximum distance from homes to the nearest public transport stop, (ii) the number of trips per hour and the maximum wait time and (iii) the amplitude. Those three criteria are filled in based on the type of region (metropolitan, urban, small-town and rural) and time (peak and off-peak hours).

The concept of basic mobility was generous, but turned out to be not viable. Therefore, the Flemish Government decided to adapt the idea of basic mobility to the concept of *basic accessibility* [Flemish Parliament, 2016]. The idea here is that public transport is not the only mode. Public transport should be supplemented with other modes such as bike, taxis or DRT. This will be achieved by introducing four *transport layers*: (i) train net, (ii) core net, (iii) additional net and (iv) customised transport. The *train net* is the main backbone of the transport network, national as well as international and is provided by trains. The *core net* connects main cities and other important locations. The transport is arranged by buses, trams and metros. Furthermore, the *additional net* connects smaller cities to the core net and/or the train net. It acts as a feeder for the other nets and provides peak hour services for home-work and home-school commuting trips. Finally, *customised transport* provides local transport demand. It consists of all kinds of transport initiatives to support low demand regions. The main goal is again to feed the core net [Mobile Flanders, 2018].

6.3.2 Adapted Transport

Currently in Flanders, Belgium the "Decree to compensate the public service obligation of the transport of persons with a disability or seriously limited mobility" [Flemish Parliament, 2012] (in Flanders known as the "compensation decree") applies since December 2012. This decree makes sure that there is a mobility system which offers subsidised adapted transport for the whole of Flanders.

The compensation decree introduces the *Adapted Transport Services (ATS)* (Dutch: Openbaar Aangepast Vervoer (OAV), previously also known as Dienst voor Aangepast Vervoer (DAV)). The ATS are operational in 27 service areas shown in Figure 6.1. In every service area, only one service provider is compensated. Those service providers can have a variety of vehicles, but they need at least one adapted vehicle. How much compensation a service provider gets depends on the trip distance and whether the customer is wheelchair bound. Customers can use ATS if they meet certain characteristics. Besides the ATS, there are also Less Mobile Services (LMS) (Dutch: Minder Mobielen Centrales (MMC)) which have even more strict rules. Only people having an income lower than two times the current minimum living wage which is dependent on the family status between € 595.13 and € 1,190.27 can use the LMS service [Programmed Federal Public Service for Social Integration, 2018]. Volunteers using their own cars are the drivers of a LMS. Hence the characteristics of ATS are more situation dependent; ATS providers have to decide for every customer whether they could use the compensation tariffs, based on different criteria such as their personal situation or even weather conditions. LMS providers on the other hand use the objective criteria of the wages. Currently, in all transport regions, transport providers with a compensation scheme have been appointed.

However, Neven et al. [2014] investigated the situation as it is and concluded that such a system is in fact not affordable for the society. The authors showed by means of simulations that a large number of resources are needed to cover the demand. Furthermore, the transport should be provided by volunteers and by a limited number of providers accredited to operate according to the conditions specified by the *compensation decree*, which seems not to be feasible.

6.4 Simulator Principles

The application consists of two types of entities, namely customers (= people who want to travel) and providers (= entities that provide travel services). Customers need to fulfil a schedule of one or more activities, each one at a given location.

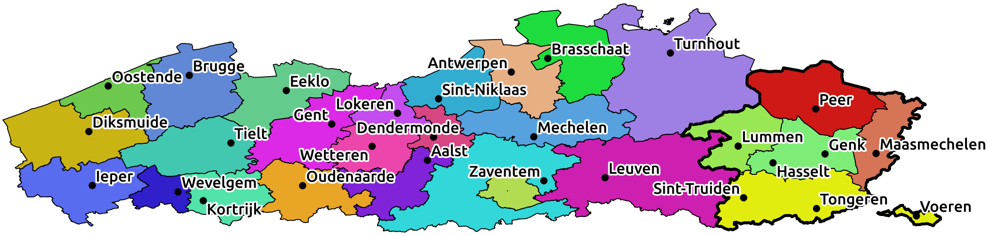


FIGURE 6.1: An overview of the transport regions in Flanders. Limburg (eastern province) is highlighted with a thicker black outline.

6.4.1 Application Concepts

In order to execute their daily activities, individuals need to arrange the transport. Routes, legs and trips will define such arrangement.

Definition 6.4.1 (Route, Leg)**.** A *route* is an alternating sequence of nodes and connecting links in the transportation network. A *leg* is a route that makes use of exactly one transportation mode.

In this paper all routes are considered to be multi-modal routes. A *multi-modal route* is a sequence of zero or more legs so that the destination of leg L is the origin of the successor of L (if any). Two consecutive legs in a multi-modal route do have different modes. An *empty multi-modal route* has zero length. It may occur when different activities are executed at the same location. Note that a route corresponds to a walk in the graph that represents the transportation network. In most cases a route corresponds to a *path* but *trails* and *walks* are allowed.

Definition 6.4.2 (Trip)**.** A *trip* is a triple $\langle r, t_0, t_1 \rangle$ where r is a (*multi-modal) route*, t_0 is the start time of the trip and t_1 is the end time of the trip.

Definition 6.4.3 (Activity, ActivityType)**.** An activity is a triple $\langle at, t_0, t_1 \rangle$ where at is an activityType, t_0 is the start time of the activity and t_1 is the end time of the activity. The purpose of an activity is called the *activityType*.

An *activity* is the act of spending a non-zero time interval to a particular purpose. Examples purposes are: working, shopping or visiting-the-physician.

Definition 6.4.4 (Episode)**.** An *episode* is an ordered pair consisting of a possibly empty *trip* and an *activity*.

The activity is performed at the target location of the *trip* (which coincides with the source location in the case of an empty trip). The schedule for an individual for a particular period consists of an ordered sequence of episodes. In simulations, the history of an individual is assumed to be periodic, i.e. to consist of an infinite repetition of a single schedule.

Definition 6.4.5 (Schedule)**.** A schedule for a particular individual and for a given period of time P (e.g. day, week, month, year) is a sequence of episodes that do not overlap in time and that completely cover the period P.

Customers can arrange *tripSequences* for a particular schedule as a whole (=atomic) or not at all.

Definition 6.4.6 (TripSequence)**.** A *tripSequence* is a sequence of one or more trips (to be executed in order to perform parts of a schedule) within which consecutive trips are separated by $N \geq 1$ activities (and hence by $N - 1$ trips that do not belong to the tripSequence.

Example: the schedule mentioned below requires two tripSequences. Trips (1),(4) on one hand and (2), (3) on the other hand constitute two separate tripSequences.

(1) Hasselt $\rightarrow$ Antwerp : bus + train (multi-modal trip)
(2) Antwerp $\rightarrow$ Deurne : car as passenger (single-mode trip)
(3) Deurne $\rightarrow$ Antwerp : car as passenger (single-mode trip)
(4) Antwerp $\rightarrow$ Hasselt : bus + train (multi-modal trip)

Definition 6.4.7 (Tour)**.** A *tour* is an ordered set of trips (of which some may belong to tripSequences where the start location of the first one coincides with the end location of the last one.

Note that a tour can be embedded in another tour (the former then is called a sub-tour).

Definition 6.4.8 (AtomicRequest)**.** An *atomicRequest* is used to book a tripSequence. An atomicRequest consists of one or more *tripRequests* (see Definition 6.4.9). TripRequests belonging to the same atomicRequest apply to the same tripSequence. In order to create a *proposalsForRequest* (see Definition 6.4.10) for an atomicRequest, every request belonging to that tripSequence need to be fulfilled.

Definition 6.4.9 (TripRequest)**.** A *tripRequest* specifies the trip for a specific part of the atomicRequest. Important components are: (i) an origin-destination pair: $\langle (X_{from}, Y_{from}), (X_{to}, Y_{to}) \rangle$, (ii) a departure and/or arrival time window [TW_s (hh_s:mm_s), TW_e(hh_e:mm_e)], (iii) a set of persons that will join and (iv) possible transport modes.

A solution for an atomicRequest consists of a *proposalsForRequest* set.

Definition 6.4.10 (ProposalsForRequest)**.** A *proposalsForRequest* is a response to an atomicRequest. A proposalsForRequest consists of one or more *proposalAlternative*(s) (see Definition 6.4.11). All the proposalAlternatives are possible solutions for an atomicRequest.

Definition 6.4.11 (ProposalAlternative)**.** A *proposalAlternative* consists of a list of proposals (see Definition 6.4.12).

Definition 6.4.12 (Proposal)**.** A proposal is a solution for travelling that meets the set of restrictions and requirements of a specific leg to solve a request: (i) an origin-destination pair $\langle (X_{from}, Y_{from}), (X_{to}, Y_{to}) \rangle$, (ii) a departure and arrival time window [TW$_s$ (hh$_s$:mm$_s$), TW$_e$(hh$_e$:mm$_e$)], (iii) commitment of the travel (i.e. a public transit provider can not guarantee exact timings (frequent delays) or that there is a free space in the transit (not frequent for non-disabled customers, but a realistic chance that there is no space left for a wheelchair)), (iv) price of the proposal in the desired monetary unit (€, US$ ...) and (v) transport mode.

Definition 6.4.13 (Trip Sequence Composer)**.** A *Trip Sequence Composer (TSC)* is an entity that can belong to an autonomous actor. A TSC is able to arrange trips (by using his/her knowledge) from point A to point B (i) by only using his own capabilities or (ii) by splitting the tripSequence request and/or one or more of the constituting tripRequests and delegate some of the simplified requests to other TSCs. The output of a TSC is always a proposalsForRequest which can constitute a multi-modal trip.

6.4.2 Thin Flows

In our simulation framework the focus will be on *thin flows*. The concept of thin flows was already mentioned by Cich et al. [2016a, 2017b] (Chapters 3 and 5): a formal definition is given below.

Consider a *network* containing *nodes* and uni-directional *links*. The network corresponds to a digraph $G(V, E)$.

Definition 6.4.14 (Connection)**.** A *connection* is a set of paths (in the graph theoretical sense) from a given source node N_S to a given target node N_T.

The connection does not necessarily contain all possible paths linking N_S to N_T neither is it required that the nodes correspond to locations where activities can be performed; as a consequence, it is not required that any of the paths corresponds to a trip.

Definition 6.4.15 (Period Specifier)**.** A *period specifier* is an expression that selects an infinite sequence of days using well known periods used in calendars. The well known periods are *day, month, year, century, etc.* The smallest resolution unit is a day.

Examples are: (i) every 26th of July, (ii) every first Monday of November, (iii) everyday, (iv) every Monday in July and August, (v) every New Year's Eve (vi) every week starting at the first Monday following July 11.

Definition 6.4.16 (Time-of-day Interval)**.** A *time-of-day interval* specified by $\langle t_b, t_e \rangle$ where $t_b, t_e \in \mathbb{R} | (t_b \in [0, 24]) \wedge (t_e \in [0, 48[\wedge (0 \leq t_e - t_b < 24))$

The duration of a *time-of-day interval* never exceeds 24 hours. A *time-of-day interval* can contain midnight. Hence it can overlap at most two consecutive days.

Definition 6.4.17 (Period Sequence, Period Sequence Duration)**.** A *period sequence* is defined by a triple $\langle I, S, E \rangle$ where I is a time-of-day interval, S is a period specifier and E is an era specified by $\langle D_0, D_1 \rangle$ where D_0 is the initial date and D_1 is the final date. The sequence consists of interval starting at t_b on each day $d \in (S \cap E) = (S \cap [D_0, D_1])$. The *duration* dur(I) of a periodic sequence I is the duration of the daily interval times the number of selected days: $\mathrm{dur}(I) = (t_e - t_b) \cdot \mathrm{nDays}(S, E) = (t_e - t_b) \cdot |S \cap [D_0, D_1]|$.

A periodic sequence is a finite sequence that does not contain any day that precedes D_0 and not any day that succeeds D_1. For concrete situations the period delimited by D_0 and D_1 shall be sufficiently large because stochastic phenomena are studied and sufficiently short in order to avoid including unwanted trends (e.g. the bus system will probably no longer be similar to the current one on July 26 2118).

Definition 6.4.18 (Passage)**.** A *passage* $p(C, T)$ where C is a connection and T is a trip, is a subtrip $\overline{T} \subseteq T$ such that the path $P(\overline{T})$ belongs to connection C i.e. $(P(\overline{T}) \subseteq P(T) \wedge (P(\overline{T}) \in C)$.

Let $\mathcal{T}$ denote the set of all trips.

Definition 6.4.19 (Flow)**.** A *flow* f identified by a pair $\langle C, K \rangle$ where C is a connection and K is a periodic sequence, is the set of passages $\{p(C, T) \mid T \in \mathcal{T}\}$ for which the intersection of K with at least one time period of moving on $p(C, T)$, is non-empty.

Note that $\langle N_S, N_T \rangle$ passages are considered, not trips or travellers in order to define flows. A particular traveller can perform more than one of the considered passages, even during a particular day. By definition, a *flow* is an finite set. Note also that it is not required that the passage period is contained in a period belonging to K: simple overlap no matter how short is sufficient.

Definition 6.4.20 (Mode set specific flow)**.** A *mode set specific flow* identified by a triple $\langle C, K, M \rangle$ where C is a connection, K is a periodic sequence and M is a set of travel modes, is a flow f such that all modes used in the trips containing the passages defining f are in M, i.e. $\forall p(C,T) \in f \ : \ M(T) \subseteq M$ where $M(T)$ denotes the set of modes used in trip T.

Note that all modes for the trip T are used in the definition, not only the modes used in the passage $p(C,T) \subseteq T$.

Definition 6.4.21 (Flow Size, Flow Rate)**.** The *size* s of a flow f is its cardinality: $s = |f|$. The *rate* of a flow is its size divided by the length of the interval: $r = s/\mathrm{dur}(K) = |f|/\mathrm{dur}(K)$

Hence, flow rates are expressed a number of travellers per time unit. A flow rate is the expected value of a stochastic computed over a *period sequence*. Please refer to Figure 6.2.

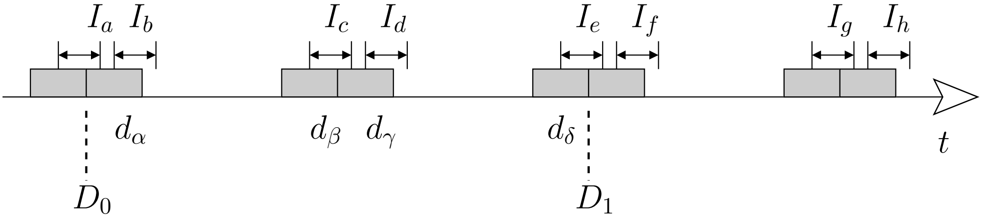


FIGURE 6.2: Periodic Sequence $\langle I,S,E\rangle$: Shaded blocks represent days selected by the *period specifier* S. The $I_a, \ldots I_h$ are instances of the *time-of-day interval* I. The days D_0 and D_1 are the earliest and latest days respectively in era E (that consists of days $d_\alpha, \ldots, d_\delta$). The *period sequence* consists of the intervals $I_b, \ldots, I_e$ i.e. the ones that start in a day selected by $\langle S,E\rangle$.

Definition 6.4.22 (Thin Flow)**.** A *thin flow* f is a flow for which the flow rate does not exceed a given threshold.

Note that for a given era E, a given connection C and flows f_a and f_b defined by the respective period sequences K_a and K_b, none, either or both of f_a and f_b might be thin.

6.4.3 Service Areas

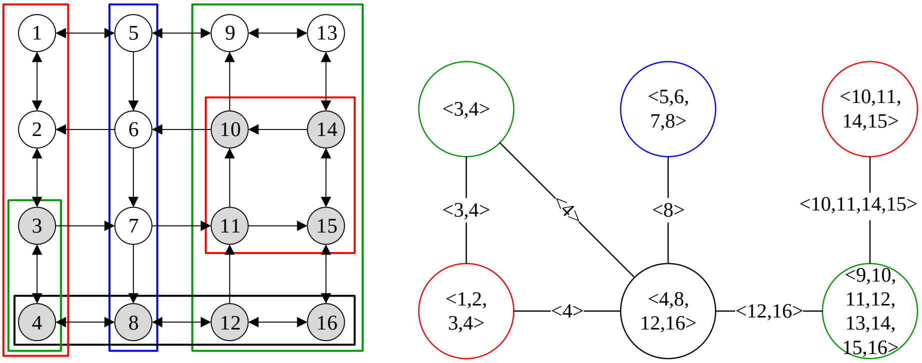
Individuals are assumed to move between locations using networks (street network, network of bus stops, etc). The concept of a service area reflects the reachability of locations using a particular service. Each service corresponds to one particular travel mode (e.g. bus, taxi) but multiple services may operate the same mode.

Definition 6.4.23 (Service Area)**.** A *service area is a set of locations that are directly connected by a particular service.*

Locations can be viewed as vertices in a reachability graph G_S^R for service S. Two vertices in G_S^R are connected by an edge if and only if the corresponding locations are directly connected by a particular service, e.g. two consecutive bus stops in a route specified in the time-table, two locations that are sufficiently near to each other so that walking between them is feasible.

There is a distinction between *provider service areas* and *customer (or personal) service areas*. The former in general are static and defined by legal rules. The latter are bound to an individual and depend on the actual position of the owner (and hence are dynamic).

Providers operate in *service area*s. In these areas, they are allowed to pick up and drop off customers. The service areas can overlap and the shared (common) locations are called *transfer point*s. Transfer points are meant to be used by a customer to transfer from one provider to another provider, i.e. changing from a physical transport mode. From all those service areas, one can create an *intersection graph*, which describes the possible transfers that are possible when arranging a trip. An example of this situation can be seen in Figure 6.3. The nodes of an intersection graph have the complete list of locations in the service area as attribute, while the edges have the overlapping locations as edge label.



(A) An example of original service areas.

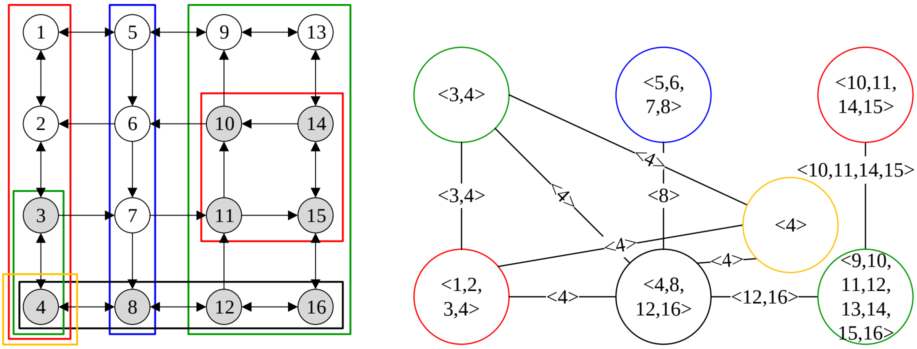
(B) Intersection graph of the service areas represented in Figure 6.3a.

FIGURE 6.3: An example of a set of service areas together with the generated intersection graph.

These service areas are used to propagate atomicRequests. If a provider is unable to fulfil a request, it will attempt to bring the customer to a (random) transfer point from which the customer can continue his travel with another provider.

Personal service locations are bound to individuals. The personal service location derived from the customer's current position determines which locations can be reached. It is used to initiate a trip requesting process. Suppose that the home location of a customer is located on node 4 in Figure 6.4a and that the personal service area consists of the home location only. One can easily create a new intersection graph such as the one in Figure 6.4b. This customer has three alternatives for starting booking his trip because the *personal service area* shares a location with three *provider service areas*.

The service areas of providers are fixed during the simulation, while the service area of a customer can change for every request.



(A) An example of the an extension made to the service areas represented in Figure 6.3a. The orange service area indicates the the own capabilities of the requester.

(B) Intersection graph of the service areas represented in Figure 6.4a.

FIGURE 6.4: An example of an extension of the original service areas together with the generated intersection graph.

6.5 Agent-Based Modelling of Thin Flows

Our DRT model simulates two main concepts, namely (i) *demand* and (ii) *supply* for a specific period of time by using an *activity based model* to predict a weekly schedule for each actor and assuming that history is periodic. It is assumed that each actor within the system will be modelled by a software agent, according to the definitions of multi-agent systems and agents that are provided by Wooldridge and Jennings [1995]. The following terminology was taken from Cich et al. [2017b] (Chapter 5). The application is divided into two parts: (i) *Thin Flows Travel Model* (TFTM) including *negotiation* between agents (demand and supply) to agree about trip details (which implies the need for schedule adaptation). TFTM is again subdivided into

TFTM$_{Dem}$ which represents the demand side (customers) and TFTM$_{Sup}$ which represents the supply side (providers). The other part is the (ii) *Operational Travel Model* (OTM) including efficient scheduling by providers of trips to be served; this is typically done by a Vehicle Routing Problem (VRP) algorithm or based on Public Transport time tables. The module can be an integral part of the agent application itself or can be provided using an API. More information about this module will be discussed in Section 6.6.5

6.5.1 Why Agent-Based Modelling?

The research question is to determine the *viability of DRT providers on the Flemish market* which is characterised as follows.

Transport providers receive requests from customers having time dependent periodic transport needs. Requests are characterised by spatio-temporal data (origin, destination, specified start and/or end time) and a limit to the timing flexibility.

DRT service providers typically operate a fleet of vehicles having a small capacity driven by people who are not available at any time of the day and possibly on a limited set of days only.

Supply and demand affect each other . The effective purchase of a trip by a customer C to provider P depends on the choice made by C from the set of proposals offered by multiple providers. This choice depends on the customer's complete activity plan for the day and not just on the attributes of a single trip.

Each proposal a provider P emits to a customer C depends on the set of requests S received from the complete customer population and on the provisional and committed proposals already emitted by S i.e. on the expected and already confirmed workload for S.

The origin and destination of the requested trip as well as the vehicle location immediately before trip execution determine the *empty* (non-occupied) and *loaded* (passengers on board) distances driven. Small changes in the locations and timing of the requested trips (as well as the order of arrival of the requests) may cause large effects on the optimal route for a given vehicle on a given day. Trips emerge from a large set of independent stochastic processes (one for each individual eventually needing transport); predictability of the daily tour(s) of a vehicle is low due to the large variance.

Customer behaviour and transport needs are assumed to be periodic in time but the period length may depend on the individual and daily schedules and transport needs are not necessarily synchronised among individuals (although some patterns may exist). As a consequence the aggregated demand is not periodic but emerges from a set of stochastic processes and shows a large variance. The low degree of predictability of the demand, the high sensitivity of the DRT profitability to the demand caused by the small vehicle capacity and the complex rules governing the decision to choose a particular proposal cause aggregated methods to be unsuitable. Micro-simulation modelling both the customer and provider behaviour is required because:

1. stochastic demand having large variance for each customer (and even for regionally or daily aggregated demand)
2. high sensitivity of providers profitability to the time series of requests due to limited vehicle capacity
3. personal preferences and/or requirements by customers w.r.t. fares, travel mode and time loss
4. travel cost compensation (by public services) for both customers and providers based on eligibility rules that depend on customers socio-demographic properties, provider accreditation and spatial constraints

The viability of the providers is determined by aggregating the daily profit over a long period (ideally at least two years). The expected value and variance can be estimated by computing a large number of (bi-)weekly profit values.

The previous needs for micro-simulation modelling could be directly supported by an agent-model, because of the autonomy property and individual behaviour centric modelling that are inherent to agent modelling [Wooldridge and Jennings, 1995]. Moreover, the need for agent-based micro-simulation is also induced: (i) by interdependence of customers using (time-)shared vehicles with a limited capacity and (ii) by the customer-provider relationship involving direct communication between these actors. In the following section, an introduction to the SARL framework is provided. This framework is used for creating and implementing the agent model of the thin flow application.

6.5.2 Introduction to SARL

SARL[1] is a general-purpose agent-oriented programming language described by Rodriguez et al. [2014] and Galland et al. [2017]. It is a general-purpose language, which means that it provides a set of key features and statements that are needed to implement a multi-agent system whatever its application domain. In order to be

---

[1] `www.sarl.io`

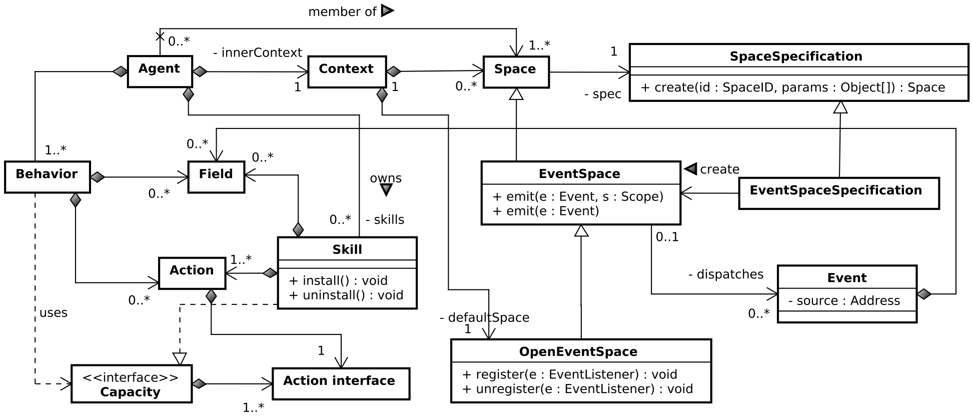


FIGURE 6.5: The class diagram of the major concepts that are supported by the statements of the SARL agent-programming language [Galland et al., 2017].

able to understand the remaining part of this paper, the most important SARL concepts will be briefly explained below, and illustrated by the class diagram on Figure 6.5.

Agent: *agent*s are *autonomous* entities that are the main actors in a SARL application. Agents can provide *capacities* and play *behaviours*. In SARL, agents provide by default some essential built-in capacities, but it is possible to provide custom capacities as well.

Behaviour: A *behaviour* is a set of functions and/or set of actions an agent will perform in specific situations.

Capacity: A *capacity* is a collection of function specifications to indicate the capabilities of an agent. Hence, it explains *what* an agent can do, but not *how* it does it. It is comparable with a Java interface.

Skill: A *skill* is a possible implementation of a capacity. Hence, while capacities describe what can be done, the skill describes how it will be done.

Space: A *space* is the support of the interaction between the agents, respecting the rules defined in an associated space specification. SARL defines natively a particular type of space, namely the *event space* to provide a support to event-driven interactions. Within an event space, agents communicate using events.

Event: An *event* is the specification of anything that happens in an event space, and may potentially trigger effects by a behaviour.

SARL was selected because it provides: (i) the key concepts for designing and implementing the agent behaviours and interactions, and (ii) an easy-to-use programming language for implementing agent models. The main problem encountered by SARL in transportation simulations is that it does not provide specific facilities to synchronise agents in simulated time. Checkpoints need to be introduced to make sure that agents do behave consistently in simulated time. Cich et al. [2017a] (Chapter 4) implemented a conservative synchronisation technique. By using this technique, it is now possible to timestamp events. Events will be received by the agents on the appropriate simulated time and hence agents will be synchronised in simulated time.

6.6 Simulation Details

The simulation model includes the behaviours of on the one hand customers and on the other hand providers. The goal of customers is, given a schedule, booking the transport to and from the activities in their schedules. Customers can request transport at providers. Providers will attempt to fulfil this request, but will ask other providers for help if they are unable to fulfil the complete request. A customer has then to transfer between two providers. Customers can ask the same trip to different providers. As a consequence, multiple proposals will be received from which the customer can choose. The decision of which proposal to choose is based on a scoring function which takes into account money as well as time.

6.6.1 Trip Requesting and Proposal Generation

In an agent-based framework, agents are supposed to "live" autonomously [Wooldridge and Ciancarini, 2001]. This means that they can only control their own attributes and own states. Agents are aware of the attributes and states from the other agents only if the latter publish them within a space where the agents can perceive information. Consequently, requests and proposals are exchanged between different agents using events. The challenging problem here is to know to which agent(s) a proposal or request needs to be sent.

Two kinds of tree data structures are used. The first one is the *trip sequence composing tree* which is responsible of keeping track of the request path, i.e. to which providers the requests are sent. The second one is the *proposal propagating tree* which keeps track of the built proposal(s). The proposal propagating tree is sent between agents during the simulation, while the trip sequence composing tree stays within the initiating agent.

6.6.2 Agents and their Behaviours

The way agents act in the simulation is coded in a SARL *behaviour*. For this application, two types of behaviours are used, namely (i) a *customer behaviour* and (ii) a *provider behaviour* that might be played by (i) a *customer agent* and/or (ii) a *provider agent*.

Customer agents can only play a customer behaviour, because they do not provide any kind of transport service to other agents. In contrast, a provider can play both behaviours. Their initial behaviour is the provider behaviour, because they provide transport services, but from the moment they ask additional trips to other providers, the customer behaviour is activated.

Agents, as well as their behaviours are implemented as *Finite State Machine*s (FSM). Two types of FSM are used: (i) a *life cycle FSM* and (ii) a *communication FSM*.

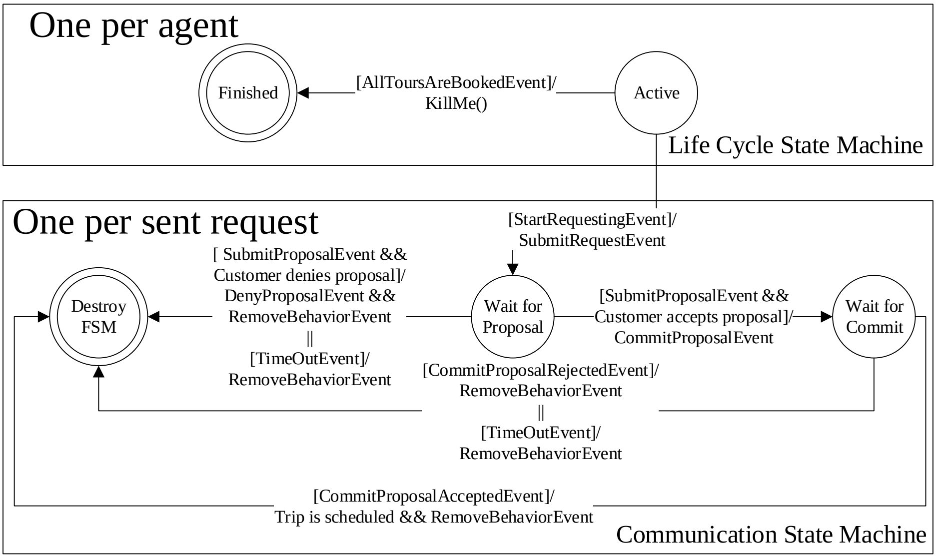
The life cycle FSM is used to simulate the more general states of an agent, while the communication FSM describes the negotiation process. Every agent has exact one life cycle state machine, while it can have a large number of communication FSMs. In Figure 9.5, the customer and provider FSM can be seen. Both FSMs, have the following states in common:

Active: Means that the agent is active in the simulation.

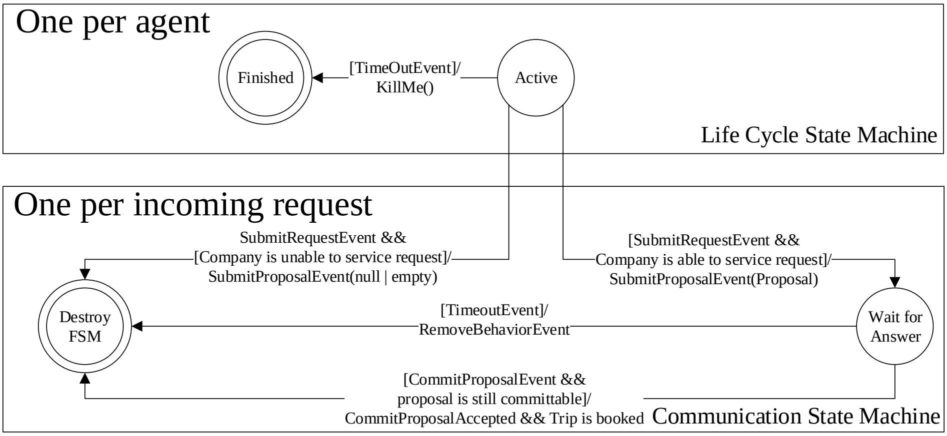
Finished: Means that the agent has conducted everything he had to do in the simulation. When the agent reaches this state, he will be removed from the simulation.

Destroy FSM: This state is in particular for the communication FSM. When reaching this state, this state machine will be destroyed. This can have several reasons.

In Figure 6.6a, the FSM of the customer agent can be seen, while the FSM of the provider agent can be seen in Figure 6.6b. Hence, in the beginning of the simulation all the agents are launched and a life cycle FSM is created for every individual agent. At the start of the simulation, they are all in the “Active” state. Provider agents will never initiate an interaction. The interaction starts when a customer agents decides to book a trip. For every request he sends (even the same request to different provider agents), a customer communication FSM is created. When the request is sent, the customer agent will move to the “Wait for Proposal” state. When a provider agent receives a request, a provider communication FSM is created. If he is not able to fulfil this request, he sends an empty proposal back and immediately reaches the “Destroy FSM” state. In the other case, he sends a valid proposal and reaches the “Wait for Answer” state. The customer agent will receive this proposal and if the



(A) An overview of the Customer FSM.



(B) An overview of the Provider FSM

FIGURE 6.6: A representation of the agents' FSMs.

proposal is empty or if the proposal is not good enough, he will reach the “Destroy FSM” state. In the other case, the customer agent will accept the proposal and sends a commitment to the provider agent. The provider agent will respond with an accept message and finally reach the “Destroy FSM” state. At the customer agent, this accept message will be received and he will also end in the “Destroy FSM” state.

Note that there are also *time out events* introduced. They are mainly created to avoid deadlocks, in which customers and/or providers are waiting for a message which will never be received.

6.6.3 Communication with External APIs

Taxis, ATP, LMS and Public Transport

TFTM_{Dem} and TFTM_{Sup} communicate by means of atomicRequests and corresponding proposalsForRequests. When the trip proposing is integrated in the OTM module, no external communication is needed. In the other case, when OTM uses an external API, Java socket communication by means of JSON messages is needed. Currently five different messages can be exchanged and if needed this can of course be extended. The messages are (i) *initialising message* is used to synchronise the providers between TFTM and OTM (only conducted once in the beginning of the simulation), (ii) *submit request message* indicates that TFTM wants a solution for a request (iii) *submit proposal message* indicates that OTM has generated a proposal for a request (the request is empty if he is unable to fulfil this request), (iv) *commit proposal message* indicates that TFTM wants to commit a proposal, (v) *commit proposal accepted message* indicates that OTM accepts a commitment.

More details can be found in Figure 6.7. Note that the corresponding classes have functionality to export to JSON and the import from JSON. Those JSON messages are actually exchanged between TFTM and OTM.

Cich et al. [2017b] (Chapter 5) discussed how MATSim can be used as VRP solver (OTM) in conjunction with the SARL model. However, the protocol for communication between OTM and TFTM is not discussed. In this paper MATSim is not used, but by using the correct protocol any API can be connected. OpenTripPlanner (OTP) is used for public transport trip planning. Requests are sent to OTP using a REST API and results are returned as JSON messages from which the information that is necessary for the simulation is extracted.

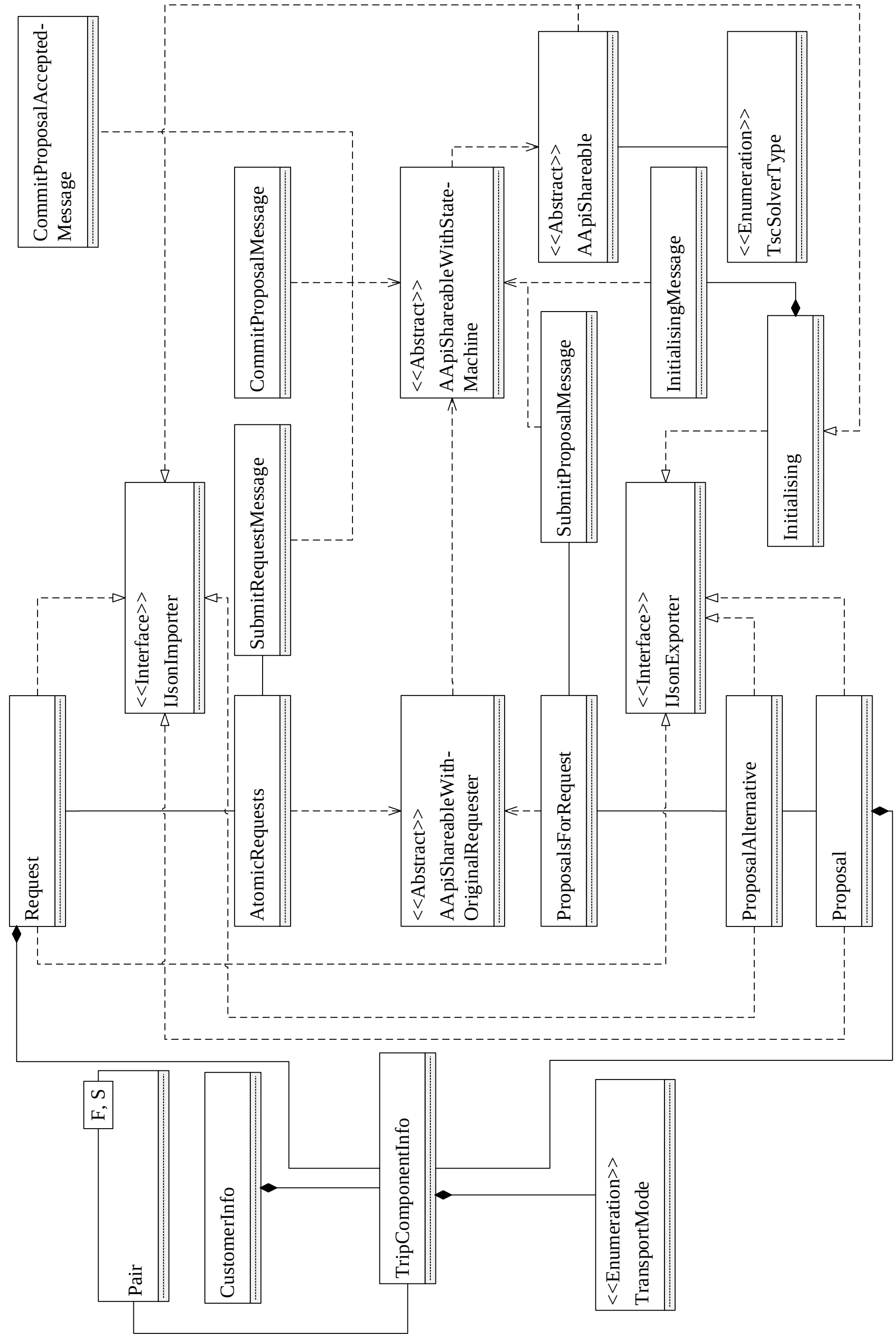


FIGURE 6.7: UML of important classes related to internal (within SARL) and external (APIs) communication.

6.6.4 Agents and their Schedules

The goal of the customers is to book trips in order to fulfil the schedule they receive in the beginning of the simulation. As will be explained in Section 6.7.2, the schedules are star based which means that the tours will always start and end at the home location. Customers will try to book tours in an atomic way, which means that they only will commit their travel if and only if they can reach their destination, but also able to return to their origin. As a consequence, when a customer is unable to book a trip in the tour (because no provider can provide the desired trip), the customer will remove the complete tour from their schedule and this will be logged. The tours will be booked in a chronological order, but customers will not start booking their trips at the same moment. At every timestamp, a customer will decide whether he will start a new tour requesting process. This is done if and only if the following criteria are met: (i) he is not busy requesting another tour, (ii) he is not in a commit phase of another tour and (iii) $t \geq \tau$ where $\tau \sim U(t_{early}, t_{late})$. When all the trips of a customer are handled, he will leave the simulation and will write an output report which can be analysed.

Customers will distribute their trip needs to different providers. Hence a customer will receive different alternatives for their trips; they need to score them somehow in order to decide which alternative is the best. An adjusted version of the Charypar-Nagel utility function [Nagel et al., 2016] is used. The scoring function scores activities as well as trips to those activities. The activity score was not adapted. However, a little adjustments were conducted regarding the travel score. In MATSim, the score of a travel is based on a specific travel cost for specific travel modes. Hence to make it simple, if someone drives with a car for 40 kilometres, the cost per kilometre is multiplied by this distance. However in our model, it is known how much a travel will cost, because the service provider has calculated this cost.

The last step is adapting the current schedule based on the chosen travel modes. Schedule adaptation is done based on the method presented in Knapen et al. [2018] and is mainly based on re-timing and rearranging activities.

6.6.5 Simplified Vehicle Routing Problem

This VRP solver uses a first in first out (FIFO) technique, which means that there is no buffering functionality. In other words, a provider is not able to collect a number of requests in order to schedule the trips more efficiently. In the current model, customers will distribute their requests to providers from which they know they are capable of handling their tripRequests (based on service areas and possible mode). A random free vehicle of that specific provider will be assigned if any is available. The current version does not support multiple persons in a car either. It was decided to not spend a lot of time to this issue since this was not the main focus of this research. However, the authors are aware of these limitations, but it opens a lot of perspectives for future research. As already discussed, it is possible to integrate all kind of APIs and hence a lot of possibilities are still open. A conceptual view of the simplified VRP solver can be seen in Figure 6.8. One can notice the difference between loaded and empty distances as well.

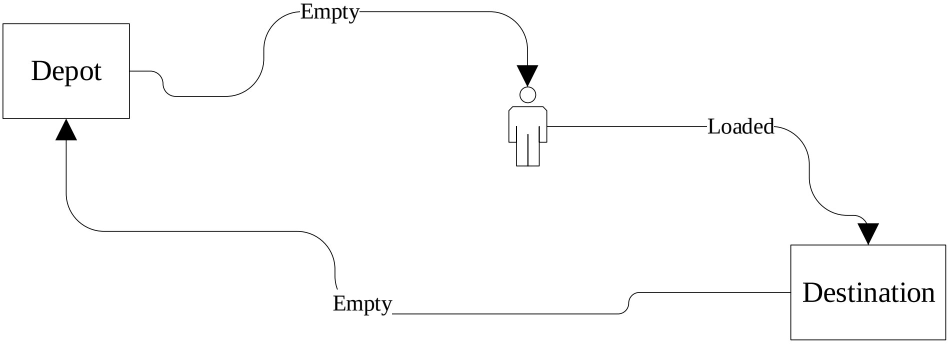


FIGURE 6.8: Conceptual view of the VRP implementation.

6.6.6 AtomicRequest Denials

Customers emit atomicRequests to providers. Depending on the type of provider, tripRequests within atomicRequests will be handled in different ways. PT providers always attempt to fulfil the tripRequest. If there is no PT for the requested arrival or departure time, this provider will attempt to give an alternative as close as possible to the requested ones. In some cases it is possible that the tripRequest will be denied. This is the case when the walk distance of the customer is not large enough. Hence, when a customer cannot reach the necessary PT stops, the tripRequest cannot be fulfilled.

For all the other providers (taxi, ATS and LMS), the algorithm explained in Section 6.6.5 is used. This means that for every vehicle in the fleet of that provider, the requested departure and/or arrival time is checked with the occupancy time windows of that vehicle. A vehicle is assigned if and only if he is able to drive from the depot to the customer, bring the customer to his destination and return to the depot without violating other time windows. If no vehicle of a specific provider meets previous constraints, the provider will deny the tripRequest.

6.7 Data

6.7.1 Network

In order to construct a transportation network, *OpenStreetMap (OSM)* was used. The software developed in Cich et al. [2016b] (Chapter 2) was used to clean the OSM network and to select only the bounding box of Limburg. Furthermore, train stations were map matched and GTFS bus stops were assigned to the road by means of the software presented in Vuurstaek et al. [2018]. An overview of the network data can be found in Table 6.1.

Type	**Amount**
Links	167,797
Nodes	321,863
Train Stops	85
Bus Stops	5,090

TABLE 6.1: An overview of the network characteristics.

6.7.2 Synthetic Population - Initial Predicted Daily Activity Schedules

The synthetic population of persons with limited mobility (PLM) in Flanders (Belgium) described in Neven et al. [2014] is used. For each individual, socio-demographic attributes (age, travel support tool (e.g. wheelchair) requirement) and a daily schedule predicted by an activity-based model are given. In the predicted schedules, each activity location is specified in a coarse way as a traffic analysis zone (TAZ) of approximately $5\,km^2$.

The properties of *individuals* were refined as follows for use in the simulation.

1. Each individual is assigned a set of feasible travel modes by random sampling using the probabilities specified in Table 6.3
2. An individual is eligible for travel cost compensation (see Section 6.3.2) if and only if a travel support tool is required.
3. The maximum daily total autonomous movement (walk, wheelchair) distance (to move from/to PT stops) is determined by uniform sampling using the parameters shown in Table 6.2.

Category	**Lower Limit in m**	**Upper Limit in m**
wheelchair	250	250
compensated	250	2,000
others	250	5,000

TABLE 6.2: Lower and upper limit for uniform sampling of maximum daily total distance that can be moved autonomously (walk, wheelchair).

The predicted daily *schedules* have a spatial star pattern; i.e. each *out-of-home* activity in a schedule is followed by a *return-to-home* trip. An example is shown in Figure 6.9. A schedule specifies the timing and location for each activity. In order to feed the micro-simulator, TAZ based location specifiers were disaggregated by randomly sampling a street address within the respective TAZ making use of the publicly available Flemish addresses database.

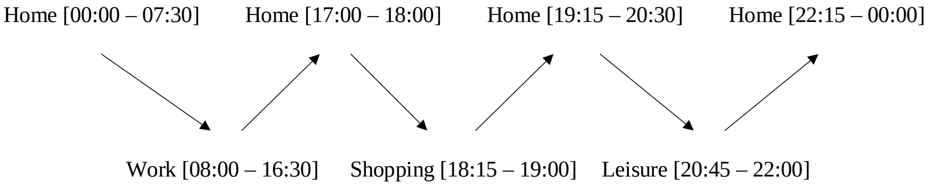


FIGURE 6.9: An example of a star based schedule. The schedule contains three tours.

Mode	**Probability**
walk	100 %
taxi	100 %
ATS	if individual eligible for compensation : 100 % else : 0 %
LMS	if individual eligible for compensation : 40 % else : 0 %
bus	75 %
train	75 %

TABLE 6.3: Probability for a travel mode to be feasible for an individual.

6.7.3 Multi-day Schedule Generation

In order to evaluate the viability of a service provider, its operations need to be simulated for a sufficiently long period in order to cope with temporal and spatial variability of the travel demand.

A multi-day schedule for each individual is derived from the single-day schedule predicted by the activity-based model. The single-day schedule is used as a *base pattern* to generate a schedule for every day in the repeated period by dropping activities with a probability that depends on the activity type. The respective *retaining* probabilities are shown in Table 6.4 (column Prob.).

6.7.4 Schedule Adaptation

During the schedule execution simulation, times in the predicted schedules need to be adjusted in order to comply with the proposals offered by demand responsive travel service providers. Schedule adaptation is based on the technique described in Knapen et al. [2018] that takes as input (i) a duty list (the list of activities to be performed along with the limits for the duration), (ii) a feasible time period for each activity and (iii) a partial order relation among the activities (restricting the feasible chronological sequences) and derived from both timing and logical constraints. Non-linear *time deviation functions (TDF)* are used to express relative and absolute constraints involving activity *start* and *end* times. Relative constraints are used (i) to avoid overlap of trip and activity periods and (ii) to ensure minimum and maximum durations. Absolute constraints ensure that activities are performed during feasible time-of-day periods (e.g. shop opening times, appointment in the hospital). Feasible activity execution periods expressed as time-of-day are shown in Table 6.4.

FSE stands for *fixed start early* (activity cannot start before this time), *FSL* for *fixed start late* (activity cannot start after this time), *FEE* for *fixed end early* (activity cannot end before this time) and *FEL* for *fixed end late* (activity cannot end after this time). Time-of-day values are expressed in minutes. Subscripted X and Y symbols refer to time-of-day values predicted by the activity-based model. The value -1 means that no constraint does apply.

The set of constraints leads to a set of simultaneous non-linear equations for which a solution is found using a relaxation method. Every activity order that complies with the given partial order is evaluated. The method returns the schedule having a minimum total *time deviation* as the optimal one.

In the thin flows simulator, individuals are assumed not to change the activity order predicted by the activity-based model. Hence, the TDF-based rescheduler is used only to adjust the activity timing conditional on the proposals delivered by

TABLE 6.4: Overview of the timing ranges of the different activities. In the headers, FSE stands for fixed start early, FSL for fixed start late, FEE for fixed end early and FEL for fixed end late. If a number is -1, this means that it is completely flexible. The X and Y are used to define relative constraints; they indicate that those timings were taken from the actual state description of the predicted schedule during the simulation. *Prob.* indicates the probability to keep an activity of the given type when generating a *multi-day* schedule from a predicted *single-day* schedule that is used as a pattern from which activities are dropped.

Activity	FSE	FSL	FEE	FEL	Prob.(%)
Other	1	-1	1,339	-1	100
Shopping	480	-1	1,140	-1	70
Culture	840	-1	1,320	-1	10
Daytime Activities	480	-1	1,080	-1	100
Doctor	$X_D - 10$	X_D	Y_D	$Y_D + 10$	40
Recreation	420	-1	1,339	-1	100
Revalidation	$X_R - 10$	X_R	Y_R	$Y_R + 10$	40
School	510	-1	990	-1	100
Social	$X_S - 60$	X_S	Y_S	$Y_S + 60$	40
Sport	1	-1	1,139	-1	60
Work	480	-1	1,080	-1	100
Hospital	$X_H - 10$	X_H	Y_H	$Y_H + 10$	20

the travel service providers. The resulting schedules are evaluated (compared) as described in Section 6.6.4.

6.7.5 Providers

In this section, an overview of the different provider characteristics will be provided. However experiment specific preferences will be discussed in Section 6.8.1

Taxis

According to the legislation in Flanders, in every municipality only one taxi can be permitted for every 1,000 inhabitants [Association of Flemish Cities and Municipalities, 2012]. However based on Declercq et al. [2016], it appeared that less than 1 % of the Flemish trips is conducted by taxi. It is assumed that the thin flow population is a large portion of this 1 %. Hence, it was decided to not take into account this real legislation and use some fixed number of taxis for each municipality.

Furthermore, there is 20 % probability that a vehicle is equipped for mobility impaired passengers. Taxi fares can differ for every municipality. To give the reader an idea of magnitude, the fares of the city of Hasselt are shown in Table 6.5. For the experiments, different fare prices are used; more information can be found in Section 6.8.1

TABLE 6.5: The rates for a taxi service. The rates are simplified and based on the situation of the city of Hasselt [City of Hasselt, 2018].

	Wheelchair Customer	**Non-Wheelchair Customer**
Board Rate	2.40 €	2.40 €
$>$ **0 km**	2.18 €/km	2.18 €/km

Public Transport

Public Transport (PT) is another important factor in the simulation since in the new basic accessibility model of Flanders, public transport will be used more efficiently. *OpenTripPlanner (OTP)* [OpenTripPlanner, 2017] is used to model PT. *General Transit Feed Specification (GTFS)* and OSM were used as an input source for OTP. GTFS of “De Lijn” (bus provider) as well as of “NMBS” (train provider) were used. OTP is able to plan a complete multi modal route from origin to destination. The most common tariffs for PT in Flanders were used and can be found in Table 6.6.

TABLE 6.6: The fares of public transport. The bus [De Lijn, 2018] and train [NMBS, 2018] rates are based on most price-friendly alternatives for adults.

	Wheelchair Customer	**Non-Wheelchair Customer**
De Lijn (bus)	2.15 €/hour	2.15 €/hour
NMBS (train)	7.70 €/trip	7.70 €/trip

Adaptive Transport Services

Limburg is divided into five transport regions (see Figure 6.1). In every region, one transport provider is compensated and hence should transport customers who qualify for the compensation rules. The scheduling of the vehicles is conducted with the simplified VRP as described in Section 6.6.5. Special tariffs hold for this kind of transport; they are shown in Table 6.7.

Less Mobile Services

For the less mobile services a real case data set was used. This data set contained the locations of all the LMSs in Limburg. The tariffs can be seen in Table 6.8

TABLE 6.7: The rates for adaptive transport Services (ATS). The provider is not allowed to charge *empty kilometres* (e.g. the distance from the depot to the pick-up location and the distance from the drop-off location to the depot or to the next customer [Inter (Private Foundation Accessible Flanders), 2018a]

	Wheelchair Customer	**Non-Wheelchair Customer**
Board Rate	2.00 €	2.00 €
< **25 km**	1.00 €/km	1.50 €/km
26-50 km	1.75 €/km	1.75 €/km
> **51 km**	2.00 €/km	2.00 €/km

TABLE 6.8: The rates for Less Mobile Services (LMS). Remark: the distances driven from depot to pick-up location and from drop-off location back to depot are charged at the specified rates [Inter (Private Foundation Accessible Flanders), 2018b].

	Wheelchair Customer	**Non-Wheelchair Customer**
> **0 km**	0.30 €/km	0.30 €/km

6.8 Experiments

6.8.1 Cases and Stochasticity

For the experiments two variables will be adjusted, namely the fleet size of all the providers (taxi, LMS and ATS) as well as the the taxi fares. The reason for those two variable is because the viability of providers has to be investigated. Besides the number of customers, the fares determine the income of a provider, while the fleet size and the drivers determine the costs of a provider. Different values for both variables were chosen and they will be combined to create different cases. By investigating these cases, the viability can be checked under different circumstances.

The different types of cases together with the corresponding case label can be found in Table 6.9. The different levels of taxi fares can be seen in Table 6.10.

TABLE 6.9: The 12 different types of experiments (cases) with the corresponding case label.

	Fleet Sizes			
Taxi Fares Level	1	5	10	25
Cheap	C-Chp-1	C-Chp-5	C-Chp-10	C-Chp-25
Normal	C-Nrm-1	C-Nrm-5	C-Nrm-10	C-Nrm-25
Expensive	C-Exp-1	C-Exp-5	C-Exp-10	C-Exp-25

TABLE 6.10: The 3 different levels of taxi fares.

Taxi Fare Level	Wheelchair Customer		Non-Wheelchair Customer	
	Board Rate	**> 0 km**	**Board Rate**	**> 0 km**
Cheap	1.20 €	1.14 €/km	1.20 €	1.14 €/km
Normal	2.40 €	2.18 €/km	2.40 €	2.18 €/km
Expensive	3.60 €	3.32 €/km	3.60 €	3.32 €/km

Due to the high level of stochasticity, those experiments were conducted for five days of a week (without the weekend).

6.8.2 Customers

For the experiments 1,250 customers were sampled based on the information given in Section 6.7.2. This is a 5 % sample of our data set. In Table 6.11, the actual allowed mode distribution can be seen. In Table 6.12, the characteristics of the customers can be found. Hence around 4.16 % of the customer is wheelchair bound and 15.36 % may use compensated providers.

TABLE 6.11: The percentage of individuals who are able/allowed to use each particular mode.

Taxi	Bus	Train	ATS	LMS	Walk
100 %	75.93 %	76.40 %	15.36 %	5.92 %	100 %

TABLE 6.12: Customer characteristics. Note that *Wheelchair* implies *Compensated*.

Wheelchair	Compensated
4.16 %	15.36 %

6.8.3 Costs of Providers

The cost of a provider includes the cost for the vehicles as well as the cost for the drivers. Both were estimated for one day. For the vehicles the following costs were taken into account: (i) the purchase of the vehicle together with the expected usage and selling price, (ii) the expected number of kilometres driven each year ($\approx$ 75,000 km), (iii) the expected fuel usage and price, (iv) the taxes, (v) the insurances, (vi) lifespan and price of tires and (vii) maintenance. If it is assumed that the car is used six days in a week, the expected cost of a vehicle for one day is € 63.60. For these calculations, common values for the variables were used. For the driver only the salary is used. If an average salary is taken for a driver, a cost of €96.12 is assumed if the driver works less than four hours (half a day) and €192.24 if the driver works more than four hours (full day).

These costs will be used to simulate the actual profits of the different providers.

6.8.4 Results

Ability to Perform All Intended Tours

One important measurement regarding customer satisfaction is the ability to perform all the activities in a day. A customer cannot perform all intended activities in a day if and only if he has at least one tour for which he did not find a valid proposal.

The results can be found in Table 6.13. For the three different price categories, the same trend can be seen. For one vehicle per provider, around 62 % of the customers have at least one tour for which they cannot find a feasible solution. For five vehicles, a large decrease can be seen to 33 %. For 10 vehicles it is around 22 % and finally for 25 vehicles, still 13 % of the customers have at least on tour which they cannot conduct.

Note that the percentage(s) are quite high. The reason is that customers ask taxi trips for a specific time range. If the provider cannot fulfil this request for this time range, the proposal will be denied. Hence nor providers, nor customers will try to adapt the time range in order to fit each other.

TABLE 6.13: An overview of the percentage of the population which are not able to fulfil all their tours.

Case	**Unable to book $\geq$ 1 (in %)**
C-Chp-1	61.63
C-Chp-5	32.35
C-Chp-10	20.10
C-Chp-25	12.03
C-Nrm-1	62.80
C-Nrm-5	31.99
C-Nrm-10	21.78
C-Nrm-25	12.89
C-Exp-1	61.44
C-Exp-5	35.10
C-Exp-10	23.23
C-Exp-25	13.88

TABLE 6.14: An overview of the relative mode share distribution based on the travelled distance.

Case	Taxi	Bus	Train	ATS	LMS	Walk
C-Chp-1	45.43	47.75	0.00	2.34	0.59	3.89
C-Chp-5	72.50	23.08	0.00	2.38	0.26	1.78
C-Chp-10	80.56	16.00	0.00	1.91	0.26	1.27
C-Chp-25	85.35	11.68	0.00	1.74	0.28	0.95
C-Nrm-1	41.96	51.49	0.00	1.92	0.57	4.06
C-Nrm-5	66.15	29.38	0.00	2.08	0.27	2.12
C-Nrm-10	72.37	23.47	0.00	2.12	0.29	1.75
C-Nrm-25	75.54	20.75	0.00	1.90	0.26	1.55
C-Exp-1	41.13	52.18	0.00	1.98	0.57	4.14
C-Exp-5	63.56	31.58	0.00	2.23	0.29	2.34
C-Exp-10	68.32	27.57	0.00	1.85	0.26	2.00
C-Exp-25	69.80	26.24	0.00	1.81	0.27	1.88

6.8.5 Mode Share

Taxi fares affect the mode choice. An important thing to keep in mind is that customers will always choose one of the alternatives, even if it is a very expensive taxi trip. In Table 6.14, the mode share based on the travelled distance can be found. In all the three price alternatives, the same trend can be seen: the more vehicles that are allowed, the more the taxi is used. The reason is that customers want to do as many activities from their schedule as possible irrespective of the travel cost. If the available fleets are larger, more customers can make use of taxis and hence more activities can be performed. However the increase is the largest with the cheap taxis and the lowest for the expensive taxis. This can be clearly seen in Figure 6.10. The sum of the taxi and PT shares is similar in all cases. The sum of the mode shares of ATS and LMS is in all the cases more or less the same. Since only a small part of the customers was allowed to use these services, more or less everyone of the qualifying customers will make use of these services. Finally, the walk mode share is directly correlated with the PT use, since walking is used in combination with PT usage, i.e. walking to and from PT stops. Apparently, trains are not used in the simulation. This is not surprising, since trains in Flanders are used to travel longer distances such as going to Antwerp or Brussels. However, there are some popular train routes within Limburg, but this is not clear in the simulation. A possible reason is that the combination of trips to be able to use a train is too difficult in the majority of the cases. The probability that a trip destination is close to a bus stop is much larger than for train station.

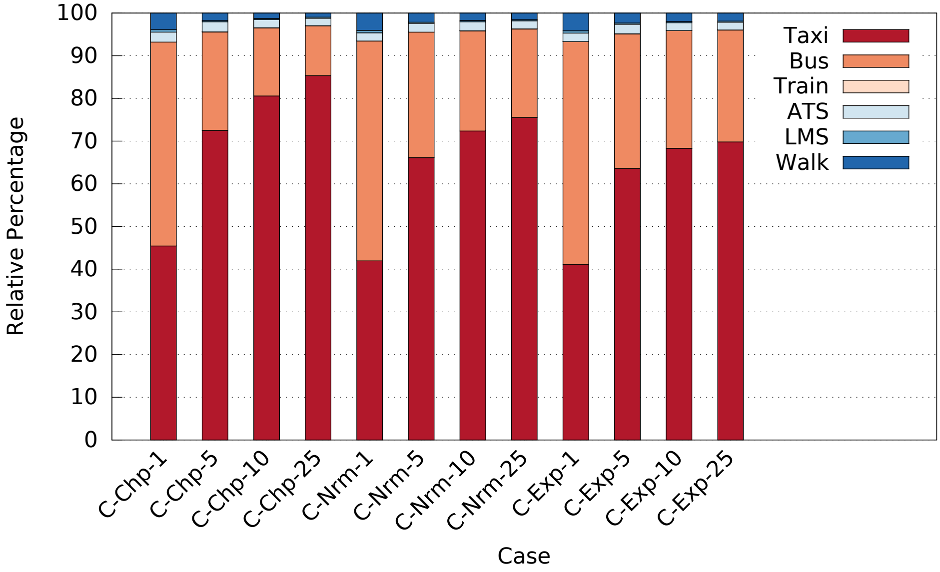


FIGURE 6.10: A visual representation of Table 6.14.

6.8.6 Provider Profits

In Table 6.15 the average profits of providers are shown per case. First of all, a LMS is a non-profit organisation. It is operated by volunteers and the main goal is to break even. That is why, in contrast with the other providers, only a cost per kilometre is taken into account for MMC drivers. As can be seen, in every case there is more or less a break even situation. When the ATS providers are examined, very large losses are seen. The reason is twofold. Firstly, since these providers were only allowed to transport compensated customers, the demand was quite low. Secondly, the compensation rule in Flanders is developed to stimulate high occupancy rates in vehicles. The current simulator does not yet provide optimisation by combining trips (trip sharing) and hence does not evaluate the possible beneficial effect of such mode of operation. For taxis the demand was high since all the customers were allowed to make use of taxis. This can clearly be seen in the profits.

Fleet sizes highly influence the profits of providers. For almost all the cases, fleets of size 25 are hard to handle. Here a trade-off should be taken into account. On the one hand, large fleet sizes provide more transport options to the customers, on the other hand they increase the cost for providers.

The results shown suggest that the ATS service should definitely be combined within taxi provider services. Hence mixed customers should be allowed, e.g. a customer in a wheelchair could be transported together with a normal taxi user.

In this case, the normal user pays the normal taxi tariff, while the wheelchair user pays the special compensation tariff and the provider will get compensation for this wheelchair user.

Table 6.15 is visualised and can be seen in Figure 6.11.

TABLE 6.15: An overview of the average profits of providers.

Case	ATS profit (in €)	LMS profit (in €)	Taxi profit (in €)
C-Chp-1	-72.28	2.81	222.28
C-Chp-5	-752.93	3.76	311.93
C-Chp-10	-1,977.33	5.80	-529.78
C-Chp-25	-5,720.69	5.54	-4,080.49
C-Nrm-1	-91.68	2.51	582.61
C-Nrm-5	-783.67	4.12	1,467.83
C-Nrm-10	-1,970.08	5.87	884.37
C-Nrm-25	-5,652.91	6.41	-2,411.62
C-Exp-1	-92.88	2.93	1,026.15
C-Exp-5	-788.49	4.26	2,633.97
C-Exp-10	-1,991.25	5.86	2,447.64
C-Exp-25	-5,684.61	6.29	-810.50

6.9 Limitations and Future Work

Despite the fact that the current simulation is already quite powerful, a number of limitations can be observed.

A more serious drawback is the limited version of the VRP solver. More realistic results could be achieved if different customers could drive together in the same vehicle. However it is expected that execution times of the simulation will increase drastically, since a VRP(-like) algorithm should be integrated.

In this paper a very specific type of schedules was investigated. Future research could extend the schedules into non-star based schedules. By doing this, customers can plan their day in a more flexible way.

6.10 Conclusion and Discussion

This paper presented a powerful and robust framework to simulate demand and supply. The software could be used to simulate other scenarios and all other kind of data can be retrieved from it; e.g. bus profiles for a city could be extracted. Because of the availability of an API, the travel providers using different operating modes (vehicle scheduling techniques) can be integrated. In the experiments 12 cases were

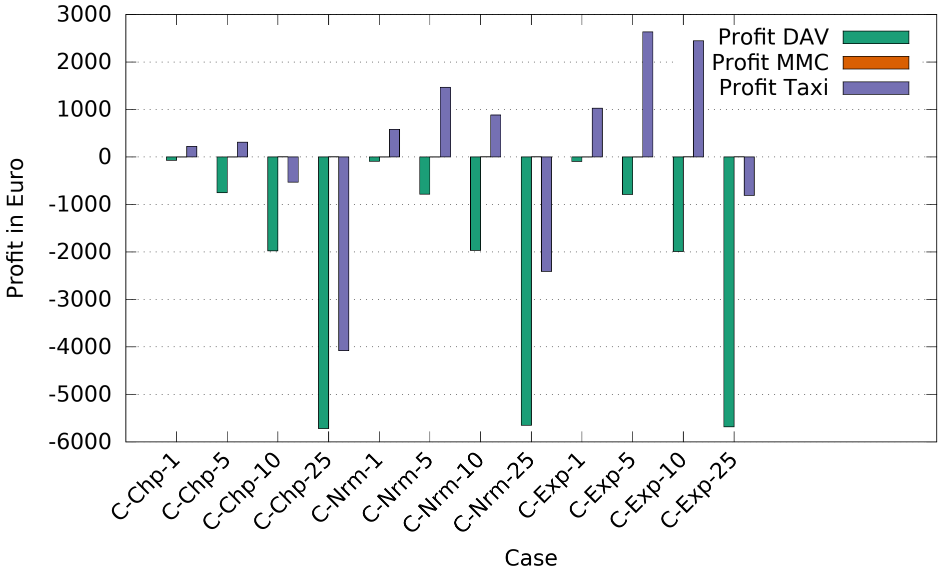


FIGURE 6.11: A visual representation of Table 6.15.

investigated were prices as well as fleet size were used as a variable. The different cases showed an expected shift in mode share and also in profit. According to the results, it is hard for transport providers to collect sufficient fares to be viable.

In Table 6.16, an overview is given of the incoming, rejected, proposed and booked requests per case. It is clear that providers receive large number of requests. On average around 63 % of the incoming requests are rejected. For the other 37 % of the incoming request, a proposal is created. From this 37 %, only 17 % is actually booked.

In order to make the simulation more accurate following concepts need to be integrated: (i) trip/taxi sharing so that the trips of multiple customers can be combined and (ii) a behavioural model for schedule adaptation so that customers become flexible about the activity timing and can negotiate about the pick-up and drop-off times.

More accurate results could be obtained if this feature could be integrated. It is expected that taxi providers will need less vehicles to serve the same number of customers. This requires the integration of a request buffering mechanism and a fast vehicle router.

The scoring function could be adapted in order to deny proposals if the price is too high as well. This is a very hard problem to solve, since it is a matter of value of time and willingness to pay.

A suggestion is definitely to combine ATS and taxi services. ATS services made a lot of losses in the simulation due to the limited demand. If those services will be combined with a taxi provider, everything will be more efficient and the losses of transporting compensated customers will probably disappear.

TABLE 6.16: An overview of the number of incoming requests (# Incoming Requests), the number of rejected requests (# Rejected Requests), the number of proposed requests (# Proposed Requests) and the number of booked requests (# Booked Requests) per case.

	# Incoming Requests		**# Rejected Requests**		**# Proposed Requests**		**# Booked Requests**	
Case	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
C-Chp-1	456.08	1,044.97	343.28	448.16	112.80	656.88	19.72	91.05
C-Chp-5	612.59	1,759.87	381.68	655.12	230.91	1,139.13	36.55	85.61
C-Chp-10	666.28	1,989.37	390.91	704.47	275.37	1,312.98	41.63	69.46
C-Chp-25	692.94	2,122.31	390.56	736.75	302.39	1,409.13	45.00	57.08
C-Nrm-1	472.98	1,071.40	354.71	459.03	118.27	670.46	21.27	104.96
C-Nrm-5	621.23	1,784.50	386.39	669.09	234.85	1,147.59	38.17	114.96
C-Nrm-10	657.39	1,974.74	389.11	728.99	268.28	1,270.83	42.72	103.12
C-Nrm-25	677.90	2,083.87	387.99	774.54	289.92	1,328.57	47.23	98.16
C-Exp-1	454.88	1,036.05	342.69	454.80	112.19	635.06	20.50	100.37
C-Exp-5	576.27	1,657.32	363.37	682.18	212.90	998.24	37.69	118.59
C-Exp-10	629.48	1,888.34	375.81	743.42	253.67	1,166.34	44.80	127.20
C-Exp-25	693.59	2,109.21	397.12	777.49	296.47	1,351.56	50.91	136.06

Acknowledgements

The research reported was partially funded by the IWT, Belgium 135026 Smart-PT: Smart Adaptive Public Transport (ERA-NET Transport III Flagship Call 2013 "Future Travelling").

The cooperation with Stéphane Galland was partially funded by UHasselt - Hasselt University with BOF-project R-6529: Large-scale agent-based modelling and simulation for transport and mobility.

Bibliography

Association of Flemish Cities and Municipalities. Taxi Services & Services for Renting Vehicles with Drivers. `http://www.vvsg.be/Omgeving/Mobiliteit/Documents/Taxidiensten%20-%20tekst%20losbladig%20werk%20lokale%20economie.pdf`, 2012. Accessed: 2018-03-22 (in Dutch).

N. Cass, E. Shove, and J. Urry. Social Exclusion, Mobility and Access. *The Sociological Review*, 53(3):539–555, 2005. ISSN 1467-954X. doi: 10.1111/j.1467-954X.2005.00565.x.

G. Cich, L. Knapen, S. Galland, J. Vuurstaek, A. Neven, and T. Bellemans. Towards an Agent-based Model for Demand-Responsive Transport Serving Thin Flows. *Procedia Computer Science*, 83:952–957, 2016a. ISSN 1877-0509. doi: 10.1016/j.procs.2016.04.191. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.

G. Cich, J. Vuurstaek, L. Knapen, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Data Preparation to Simulate Public Transport in Micro-Simulations using OSM and GTFS. *Procedia Computer Science*, 83:50–57, 2016b. ISSN 1877-0509. doi: 10.1016/j.procs.2016.04.098. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.

G. Cich, S. Galland, L. Knapen, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Addressing the Challenges of Conservative Event Synchronization for the SARL Agent-Programming Language. In Yves Demazeau, Paul Davidsson, Javier Bajo, and Zita Vale, editors, *Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection*, pages 31–42, Cham, 2017a. Springer International Publishing. ISBN 978-3-319-59930-4.

G. Cich, L. Knapen, M. Maciejewski, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Modeling Demand Responsive Transport using SARL and MATSim. *Procedia Computer Science*, 109:1074–1079, 2017b. ISSN 1877-0509. doi: 10.1016/j.procs.2017.05.387. 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017) / Affiliated Workshops.

G. Cich, L. Knapen, S. Galland, and D. Janssens. Viability Evaluation of Transport Supply for Thin Flows with Agent-Based Simulation. *European Journal of Transport and Infrastructure Research*, 2018. Submitted.

City of Hasselt. With the Taxi. `https://www.hasselt.be/nl/met-de-taxi`, 2018. Accessed: 2018-02-22 (in Dutch).

De Lijn. Basic Mobility: A Stop at Less Than 750 Metres. `https://codex.vlaanderen.be/Portals/Codex/documenten/1008191.html`, 2002. Accessed: 2018-02-20 (in Dutch).

De Lijn. Pay with your Phone. `https://www.delijn.be/nl/vervoerbewijzen/ticket-op-gsm/`, 2018. Accessed: 2018-02-22 (in Dutch).

K. Declercq, D. Janssens, and G. Wets. Flemish Household Travel Survey 5.1 (2015–2016) Table Report. Technical report, UHasselt - Hasselt University, IMOB, Hasselt, Belgium, December 2016. (in Dutch).

A. Delbosc and G. Currie. The Spatial Context of Transport Disadvantage, Social Exclusion and Well-Being. *Journal of Transport Geography*, 19(6):1130–1137, 2011. ISSN 0966-6923. doi: 10.1016/j.jtrangeo.2011.04.005. Special section on Alternative Travel futures.

Flemish Government. Decree Concerning The Organization of Passenger Transport by Road. `https://codex.vlaanderen.be/Portals/Codex/documenten/1008191.html`, 2001. Accessed: 2018-02-20 (in Dutch).

Flemish Government. Decision of the Flemish Government Concerning Basic Mobility in the Flemish Region. `https://codex.vlaanderen.be/Zoeken/Document.aspx?DID=1010398&param=informatie`, 2002. Accessed: 2018-02-20 (in Dutch).

Flemish Parliament. Design of Decree to Compensate the Public Service Obligation of the Transport of Persons with a Disability or Seriously Limited Mobility. `http://docs.vlaamsparlement.be/pfile?id=1039564`, Dec 2012. Accessed: 2018-02-20 (in Dutch).

Flemish Parliament. Concept Note. With Basic Accessibility to an Efficient and Attractive Transport Model in Flanders that Optimally Meets the Global and Local Transport Demand. `https://docs.vlaamsparlement.be/docs/stukken/2015-2016/g614-1.pdf`, Jan 2016. Accessed: 2018-02-20 (in Dutch).

E. W. M. Fürst and C. Vogelauer. Mobility of the Sight and Hearing Impaired: Barriers and Solutions Identified. *Qualitative Market Research: An International Journal*, 15(4):369–384, 2012. doi: 10.1108/13522751211257060.

B. A. M. Gallagher, P. M. Hart, C. O'Brien, M. R. Stevenson, and A. J. Jackson. Mobility and Access to Transport Issues as Experienced by People with Vision Impairment Living in Urban and Rural Ireland. *Disability & Rehabilitation*, 33(12):979–988, 2011. doi: 10.3109/09638288.2010.516786.

S. Galland, S. Rodriguez, and N. Gaud. Run-Time Environment for the SARL Agent-Programming Language: The Example of the Janus Platform. *International Journal on Future Generation Computer Systems*, October 2017. ISSN 0167-739X. doi: 10.1016/j.future.2017.10.020.

S. Haustein. Mobility Behavior of the Elderly: An Attitude-Based Segmentation Approach for a Heterogeneous Target Group. *Transportation*, 39(6):1079–1103, Nov 2012. ISSN 1572-9435. doi: 10.1007/s11116-011-9380-7.

Inter (Private Foundation Accessible Flanders). Adaptive Transport. `http://www.meermobiel.be/aangepast-vervoer`, 2018a. Accessed: 2018-02-22 (in Dutch).

Inter (Private Foundation Accessible Flanders). Volunteer Transport and Less Mobiles Central (LMC). `http://www.meermobiel.be/info-vrijwilligersvervoer-mmc`, 2018b. Accessed: 2018-02-22 (in Dutch).

S. Jain, N. Ronald, R. G. Thompson, and S. Winter. Predicting Susceptibility to Use Demand Responsive Transport using Demographic and Trip Characteristics of the Population. *Travel Behaviour and Society*, 6(Supplement C):44–56, 2017. ISSN 2214-367X. doi: 10.1016/j.tbs.2016.06.001.

L. Knapen, G. Cich, and D. Janssens. A Task Scheduling Method for Agent/Activity-Based Models. *Procedia Computer Science*, 130:761–766, 2018. ISSN 1877-0509. doi: 10.1016/j.procs.2018.04.132. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / Affiliated Workshops.

A. Lee and M. Savelsbergh. An Extended Demand Responsive Connector. *EURO Journal on Transportation and Logistics*, 6(1):25–50, Mar 2017. ISSN 2192-4384. doi: 10.1007/s13676-014-0060-6.

K. Lucas. Transport and Social Exclusion: Where Are We Now? *Transport Policy*, 20: 105–113, 2012. ISSN 0967-070X. doi: 10.1016/j.tranpol.2012.01.013. Urban Transport Initiatives.

R. L. Mackett and R. Thoreau. Transport, Social Exclusion and Health. *Journal of Transport & Health*, 2(4):610–617, 2015. ISSN 2214-1405. doi: 10.1016/j.jth.2015.07.006.

Mobile Flanders. How does it work? `https://www.mobielvlaanderen.be/basisbereikbaarheid/hoe.php`, 2018. Accessed: 2018-06-18 (in Dutch).

K. Nagel, B. Kickhöfer, A. Horni, and D. Charypar. A Closer Look at Scoring. In A. Horni, K. Nagel, and K.W. Axhausen, editors, *The Multi-Agent Transport Simulation MATSim*, chapter 3, pages 23–34. Ubiquity Press, London, 2016. doi: 10.5334/baw. License: CC-BY 4.0.

J. D. Nelson, S. Wright, B. Masson, G. Ambrosino, and A. Naniopoulos. Recent Developments in Flexible Transport Services. *Research in Transportation Economics*, 29(1):243–248, 2010. ISSN 0739-8859. doi: 10.1016/j.retrec.2010.07.030. Reforming Public Transport throughout the World.

A. Neven, K. Braekers, K. Declerq, T. Bellemans, D. Janssens, and G. Wets. Methodology to Optimize Resource Requirements of a Demand Responsive Transport System for Persons with Disabilities: A Case Study of Flanders. *TRB 93rd Annual Meeting Compendium of Papers*, 2014.

P. Nguyen-Hoang and R. Yeung. What Is Paratransit Worth? *Transportation Research Part A: Policy and Practice*, 44(10):841–853, 2010. ISSN 0965-8564. doi: 10.1016/j.tra.2010.08.006.

NMBS. Rail Pass. `https://www.belgianrail.be/nl/vervoersbewijzen/passes-kaarten/rail-pass.aspx`, 2018. Accessed: 2018-02-22 (in Dutch).

OpenTripPlanner. Multimodal trip planning & analysis. `www.opentripplanner.org`, 2017. Accessed: 2018-03-21.

J. Preston and F. Rajé. Accessibility, Mobility and Transport-Related Social Exclusion. *Journal of Transport Geography*, 15(3):151–160, 2007. ISSN 0966-6923. doi: 10.1016/j.jtrangeo.2006.05.002.

Programmed Federal Public Service for Social Integration. (Equivalent) living wages. `https://www.mi-is.be/nl/equivalent-leefloon`, 2018. Accessed: 2018-06-11 (in Dutch).

S. Rodriguez, N. Gaud, and S. Galland. SARL: A General-Purpose Agent-Oriented Programming Language. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 103–110, Aug 2014. doi: 10.1109/WI-IAT.2014.156.

N. Ronald, R. G. Thompson, and S. Winter. Comparison of Constrained and Ad Hoc Demand-Responsive Transportation Systems. *Transportation Research Record*, 2536 (2536):44–51, 2015. doi: 10.3141/2536-06.

G. Sammer, T. Uhlmann, W. Unbehaun, A. Millonig, B. Mandl, J. Dangschat, and R. Mayr. Identification of Mobility-Impaired Persons and Analysis of Their Travel Behavior and Needs. *Transportation Research Record*, 2320(2320):46–54, 2012. doi: 10.3141/2320-06.

J. Shen, S. Yang, X. Gao, and F. Qiu. Vehicle Routing and Scheduling of Demand-Responsive Connector with On-Demand Stations. *Advances in Mechanical Engineering*, 9(6), 2017. doi: 10.1177/1687814017706433.

B. P. Shrestha, A. Millonig, N. B. Hounsell, and M. McDonald. Review of Public Transport Needs of Older People in European Context. *Journal of Population Ageing*, 10(4):343–361, Dec 2017. ISSN 1874-7876. doi: 10.1007/s12062-016-9168-9.

J. Vuurstaek, G. Cich, L. Knapen, W. Ectors, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. GTFS bus stop mapping to the OSM network. *Future Generation Computer Systems*, 2018. ISSN 0167-739X. doi: 10.1016/j.future.2018.02.020.

M. Wooldridge and P. Ciancarini. Agent-Oriented Software Engineering: The State of the Art. In *Agent-Oriented Software Engineering: First International Workshop (AOSE 2000)*, volume 1957 of *Lecture Notes in Computer Science*, pages 1–28. Springer-Verlag, 2001.

M. Wooldridge and N.R. Jennings. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2):115–158, 1995.

Chapter 7

Postscript: Tool Applicability

7.1 Generalisability

In this part, a powerful simulation tool for DRT was presented. The goal in this tool was to simulate customers who were able to plan their tours in advance. Specific tours were booked as a sequence of trips which were dependent on each other, e.g. someone goes to the shop and wants to be sure that he/she is able to come back. Parts of those trips can be, but not necessary are multi-modal. Essential characteristics of the tool are the fact that customers get a commitment from all transport providers with whom they will travel (except for PT, since this is time table based) and customers book complete (multi-modal) tours.

In this part, DRT was used as a door-to-door service for customers who were not able to walk long distances or for people with a mobility impairment. However, in some cases DRT was also used as a feeder/connector service. It is possible that customers are picked up at their home and brought to a public transport stop from which they continued their trip.

The question of how generalisable this tool is could be raised. In this part a robust software tool was presented. The rules and assumptions that were integrated in the software make the tool applicable for all kinds of DRT services for which a booking in advance is required. In the presented software a customer who wants to book a trip must comply with some rules. In our case, they had to specify a departure and/or arrival interval and they had to book their trip at least x minutes before the departure time. Every other DRT or transport service that has similar rules could be simulated by this tool.

The tool is (for the moment) less applicable for situations were an ad-hoc request happens. An example of an ad-hoc request in this context is that a customer calls a specific transport service to be picked up as soon as possible. This is a problem when the new request has a timing conflict with trips already committed to other customers. To give the reader a small reminder: a customer always books a back and forth travel by means of a tour. Tours consist of trips where the origin and destination are locations to perform an activity. To make a trip multi-modal, tripComponents were introduced. Those components distinguish different modes of a trip. A customer attempts to book a tour by dispatching (in this case) two trips to different transport providers. Transport providers could at their turn dispatch parts of the trip to other transport providers in order to be able to fulfil a complete trip. To be clear, every transport provider in this chain provides a tripComponent of a trip. An important implementation choice here is that a tour only could be committed if and only if a solution was found for all the trips within the tour. This means that a trip is committable if and only if there is a solution for every tripComponent in the trip. The main consequence is that while distributing this chain of tripComponents over different transport providers, a lot of time windows of transport providers were blocked for a given time. This means that if another customer would like to travel at that moment, it was impossible for that customer (dependent on the number of vehicles of course). Integrating only a small ad-hoc trip and to be more specific a tripComponent is hard due to this commitment procedure, but also due to the tour concept. The main consequence of this implementation choice is that there are more booking failures and that the income of the providers probably is slightly underestimated.

Part II

Transport Optimisation Organised by Facilities

Chapter 8

A Simulation Study of Commuting Alternatives for Day Care Centres

The work presented in this chapter was published on May 12, 2018 in the Journal of Future Generation Computer Systems [Cich et al., 2018].

Keywords: Mobility Impaired People; Optimisation; Carpooling; Commuting; Cost Savings; Algorithms

Abstract

In Flanders (Belgium), mobility impaired people need to travel frequently from their homes to a *Day Care Centre (DCC)*. Currently this is done by *subsidised bus services* but recently a decision was made to cancel these subsidies. The fare the DCC guests will have to pay for transport by bus is too high for most of them.

This paper investigates a solution where voluntary drivers bring as many DCC guests as possible to the DCC by carpooling. These drivers can pick up and drop off DCC guests along the way to their work location or any other destination.

In general it turns out to be impossible to drive all the DCC guests to the DCC by carpooling. The remaining DCC guests will be picked up by dedicated buses. The goal is to keep the bus travel cost as low as possible. The solution is constrained by car capacities, time windows for both drivers and DCC guests, upper bounds for detours and the availability of intermediate transfer locations. The main challenge is the involvement of multiple transportation service providers. Some of these are not under the control of the consultant in charge of finding an efficient solution for

the DCC and hence, their operation and cost cannot be included in the objective function. Solving the problem requires consideration of several cases each leading to a heavy combinatorial computation.

Although it seems to be impossible to find a carpool solution in which all the passengers reach the DCC, the results are promising. In several cases four or more chartered buses can be saved on. However, average results show a saving around one to two chartered buses which represents a cost reduction between 20 % and 30 %.

8.1 Introduction

Nowadays, carpooling and shared mobility receive increasing attention in the literature because of the high potential to reduce vehicle expenses and congestion on the roads. However, these alternatives are mainly aimed at healthy people. Alternatives for mobility impaired people and children going to school are scarce. For carpooling to school in Flanders, a few examples can be found such as the Schoolpool introduced by Taxistop [Taxistop, 2017]. For mobility impaired people, carpooling is less obvious. Alternatives such as using public transport are often difficult and sometimes even impossible. In many cases the only option for them is (expensive) demand responsive transport.

In this paper, a general carpool-like problem is tackled. This problem consists of people (*facility visitors*) who need to go to a common location (*facility*) where they perform their (daytime) activities and need to be brought by other people (*drivers*). Examples of facility visitors are (i) children going to school, (ii) mentally impaired people going to a day care centre or (iii) children going to the same sports club. They can be transported by their parents, neighbours, public transport or taxi-like approaches. Note that those problems are similar, but also slightly different. Commuting by private car is in most of the cases convenient, but results in unnecessary many vehicles on the road and inefficient use of vehicles. Public transport is sustainable, but not always possible for the target population. Taxi-like approaches are also possible, but expensive.

An obvious solution to this problem is making more efficient use of private cars. Drivers can pick up other facility visitors which are commuting to the same destination. However, this is not easy due to facility visitors' and drivers' constraints. This paper presents, given a set of facility visitors, a set of drivers and a set of constraints, a solution to commute from their homes to a common destination. Note that facility visitors not present in a solution, should still rely on other transport possibilities. Furthermore, some locations can be used as transfer locations, where facility visitors can transfer from one driver to another.

The remaining part of the paper is organised as follows: Section 8.2 gives an overview of related work. Next, in Section 8.3, the case study which is investigated in this paper is explained. In Section 8.4 the concepts that are used in the paper are described, followed by the details about the data collection in Section 8.5. In Section 8.6, limitations and assumptions that are used for the algorithms are described. Section 8.7 defines the discussed problem as a graph theoretical problem. The used algorithms are explained in Section 8.8. Results can be found in Section 8.9. Section 8.10 discusses and compares the results of the proposed methods. Finally, Section 8.11 concludes the paper.

8.2 Related Work

The problem that is tackled in this paper is related to *carpooling* which is a special case of *ride-sharing* and is even more related to the *school bus routing problem*.

Authors extensively investigated reasons why people do or do not carpool/share rides. Li et al. [2007] conducted a survey in Houston and Dallas to investigate why people do or do not carpool. A main reason why they carpool, which is of interest in this research, is sharing vehicle expenses. Important reasons why they do not carpool are location and schedule limitations and flexibility. Results of this research will show similar findings.

Buliung et al. [2009] investigate the factors for a successful carpool with data of a web service Carpool Zone, provided by the Smart Commute in the Greater Toronto and Hamilton area. They found that the main factors where spatial accessibility to carpool matches, car ownership and socio demographics. Factors such as carpool infrastructure and personal attitudes were less important.

The Flemish Government and Traject [Flemish Government and Traject, 2006] investigated a variety of measures to promote sustainable commuting trips. They investigated objective and subjective obstacles for the employee as well as for the employer. For employers, flexible working times are one of the main obstacles for not carpooling. Various activities after work and the lack of knowing other employees were other reasons why people do not carpool. The subjective obstacles were in line with the objective ones, namely flexibility and efficiency. Carpooling is also considered as an incursion on one's privacy. People consider their car and the trip from home to work or work to home as a moment to relax. Reasons to carpool are related to financial, social and environmental benefits.

The Belgian Federal Government [Federal Government, 2008] investigated the effectiveness of different measures for different types of transport for commuting purposes. For carpooling they only investigated objective measures such as infrastructure, cooperation with the government and other companies, information about carpooling, etc. Measures which seem to increase carpooling were information sessions about carpooling, a guaranteed ride back home, organising carpools, access to a database and parking spots especially for carpools.

Another research in Belgium by Vanoutrive et al. [2012] investigated three characteristics: location, organisational factors and promotion. They found a higher number of carpooling in less accessible regions. Activity sectors such as construction, manufacturing and transport were popular carpool sectors as well. Based on the literature, it seems to be clear that a decreased flexibility always returns as a main reason for not using carpooling. This will be investigated later on in this paper.

Note however that many of the subjective factors mentioned in the research covering *carpooling for commuters* may not hold in the case of *carpooling for facility visitors* because in many cases the drivers are their parents.

As mentioned before, ride-sharing shares characteristics with the facility visitors problem in this paper. Agatz et al. [2012] give an overview of ride-sharing. The objectives of ride-sharing include (i) minimising system-wide vehicle-miles, (ii) minimising system-wide travel time and (iii) maximising the number of participants. These objectives are also applicable to the problem addressed in this paper. Ride-sharing takes into account constraints as well. The most important one is the time window. In most of the ride-sharing applications, participants specify an earliest possible departure time and a latest possible arrival time. In other applications it is also possible to specify a maximum excess travel time. Other constraints relate to personal preferences. In the presented paper, even more constraints will be taken into account which are explained in Section 8.4. In Agatz et al. [2012], a distinction is made between the *static* and the *dynamic* ride-share applications. Static means that the set of drivers and riders are initially known by the algorithm, while in dynamic applications these sets can change and hence, the algorithm needs to work in real time. In this paper, the focus will be on the static method. If every driver picks up at most one passenger, a *maximum-weight bipartite matching algorithm* can be used, however the problem which is addressed in this paper is more similar to the single driver, multiple rider arrangements and these problems are harder to solve. Our problem can even be classified as a single rider, multiple driver arrangement because facility visitors are allowed to transfer between drivers.

Park and Kim [2010] give an extensive overview of the school bus routing problem. Clearly there are several similarities with the problem addressed in this paper. In the school bus routing problem, there are mainly four types of data: (i) students, (ii) schools, (iii) vehicles and (iv) distance matrix. The students can be compared with the facility visitors in our problem, the schools with the facilities, the vehicles with the drivers and the distance matrix is calculated in our problem as well. The different steps to solve a school bus routing problem can be read in Park and Kim [2010]. There is a difference between the *single school problem* where students have to be dropped off at one school, whereas *multiple schools* indicates that students can be dropped off at different schools. Clearly, our problem matches the single school problem as well as the subdivision into the *morning* and *afternoon* problems. The school bus problem shares a number of characteristics and constraints with the facility visitors problem, such as *vehicle capacity*, *maximum riding time*, *school time window* (= facility opening times) and earliest pick-up time for facility visitors.

Bögl et al. [2015] solve a school bus routing problem (SBRT) that allows for transfers (because pupils for multiple schools are served by a single set of buses). The proposed solution solves the (i) bus stop selection, (ii) pupil to stop assignment, (iii) bus routing and (iv) bus scheduling subproblems in an integrated solver. Subproblems are handled in an iteration of hierarchically organised algorithms that allows for feedback from the bus *routing* and *scheduling* stages to the bus *stop selection* and *pupil assignment*.

Research on multi-modality for people transport focuses on route advisers for individual trips (also referred to as *Traveller Information Services*). In most cases the advice combines car-sharing and public transportation trips. The *SocialCar* EU-project [Rizzoli et al., 2016] and the *RTA Trip Planner* (Chicago Regional Transport Authority) [Regional Transportation Authority Chicago, 2017] are typical examples. The report by Van Audenhove et al. [2014] is a typical report that recognises the need for multi-modality and the opportunities created by the availability of Intelligent Transportation Systems (ITS) and smartphones. It states: '*Urban mobility is one of the toughest system-level challenges facing actors of the mobility ecosystems. In the future, innovative mobility services will be driven less by improvements in single transport modes than by integration. What is needed is system-level collaboration between all stakeholders of the mobility ecosystem to come up with innovative and integrated business models.*' It scores the travel infrastructure for a large list of cities worldwide and investigates future business models. Several similar reports can be found.

However, research that addresses the *operational problem of multi-modal recurrent trips* seems to be absent.

Solutions for *school bus* problems on the one hand and *carpooling* problems on the other hand have extensively been discussed in literature. This paper investigates the feasibility of a multi-modal solution which is required for budgetary and/or sustainability reasons but seems to be more complicated than both unimodal solutions.

Last but not least the financial situation addressed in this paper is of all times.

In older research, Oxley and Richards [1995] attempt to measure additional transport cost for people with a disability. They used Office of Population Censuses and Surveys (OPCS) of Disability, Family Expenditures Survey and Department of Social Security Research Report to investigate this matter. They concluded that the analysis of the extra expenditures of people with a disability was complex and did not show clear evidence to prove that disability introduces extra expenditures for transport. However, they showed a lower income level of impaired people. They showed that when income rises, there is an increased expenditure on transport as well. The expenditures rise substantially faster than the income rate. For impaired people this rate is even faster than for able-bodied people.

Research carried out a few years later by Roberts and Lawton [1999] investigated the need for financial assistance for transport costs for families with disabled children. This research in the United Kingdom stated that the Government is aware of the importance of transport for disabled people. However, only one third of the families with disabled people received financial support for transport related costs of the Family Fund Trust grants. This research suggests that the support is insufficient for many families.

Although abundant literature is found on the *carpooling* and *school bus routing* sub-problems, none could be found that covers the combined problem discussed in this paper.

8.3 Case Study: Day Care Centre Visitors Commuting

8.3.1 Problem Definition

As a case study, the problem is investigated where a set of people, suffering from mobility and/or intellectual impairment and living in different locations have to be mobilised every morning from their homes to a given *day care centre (DCC)*, and back to their homes in the afternoon. The dedicated buses that used to drive them will no longer be subsidised by the government. The fares the DCC guests need to pay without subsidies is too high for most of them. Neven et al. [2015] mentioned high costs for demand responsive transport for mobility impaired people as well. An alternative solution proposed to cut costs is the use of volunteers who drive the DCC guests and if necessary, a few vehicles can be hired by the DCC. The problem has many constraints, both with regard to the drivers, as well as with regard to DCC guests.

Some (but not all) of the volunteers can drive some DCC guests directly to the DCC and back, picking up additional DCC guests along the way. Other drivers can only pick up and drop off DCC guests at certain locations on the way to their work or other errands, provided the detour they make is not too big, and provided the number of DCC guests in their car exceeds neither their car's seating capacity nor their personal capacity of handling several DCC guests at once. Each driver has a *morning time window* within which he/she has to leave home, and arrive at work, and similarly, an *evening time window*. Some locations (homes of DCC guests, or other locations such as a community centre or a sheltered bus stop) serve as *transfer locations*. These are locations where a few DCC guests can be gathered and wait until they are picked up to go to the DCC or dropped off on the way back home. The transfer locations also have a time window within which DCC guests have to be dropped off and picked up, as well as a *capacity constraint* - the maximum number of DCC guests that can stay there at any given time. Some DCC guests can reach the transfer locations independently, by walking or cycling. The DCC guests should not be commuting in either direction for more than T minutes ($T = 90$), and they should not have to wait in more than one transfer location for reasons of convenience.

Finally, dedicated bus services (called *chartered buses* hereafter) are used to transport DCC guests who are not being served by a volunteer driver. The chartered buses have different capacities and costs, proportional to their capacity. The cost of using a chartered bus covers the cost of using the vehicle and hiring the driver. It is the sum of a fixed start fee and a time- (or distance-)dependent fee. This paper aims to find routes so that all the DCC guests reach the DCC in the morning, and return home in the evening in such a way that all constraints are met and the cost of the chartered vehicles is kept as low as possible.

As mentioned in Section 8.2, the SBRT solved by Bögl et al. [2015] shares the following properties with the DCC problem which means that one problem can be reduced to the other one by selecting appropriate parameter settings:

1. a subset of the pupils can move autonomously to the pick-up locations,
2. pupils can change vehicle at particular transfer locations and
3. departures and arrivals are subject to time window constraints.

Essential differences are:

1. (electric) wheelchair and person incompatibility (e.g. different intellectual disabilities) constraints in the DCC problem,
2. no detour constraints on individual bus trips in the SBRT as opposed to carpool driver specific upper limits for detours in the DCC problem,

3. the number of buses is minimised by the SBRT and the number of carpool drivers is given in the DCC problem and

4. the cost per unit distance is the same for each partial trip in the SBRT. In the DCC problem the bus can bring DCC guests to the DCC only. Since DCC guests can transfer at most once, the pre-transfer partial trip is by carpooling. The cost per unit distance for the pre-transfer part is much lower than for the chartered bus part because carpool drivers act as volunteers. For this reason, the results obtained by the SBRT and the DCC problem cannot be easily compared. However, the SBRT results show the importance of transfers. In the situation where sufficient carpool drivers are available and no chartered bus is required, the costs for all trip parts are equal and the beneficial effect of transfers shown by Bögl et al. [2015] will apply.

The SBRT solves the complete set of subproblems required for optimisation. In our case, carpooling trips are used as feeders for transfer locations where DCC guests are picked up by one or more chartered buses that cannot be part of the optimisation for the reasons mentioned in Section 8.3.2.

8.3.2 Cooperation between Stakeholders

Three parties are involved: the transport *provider* (the bus operator), the transport *organiser* (the institution, including the volunteering car drivers) and the *passenger* (the DCC guest). Solving the daily commuting problem requires the formulation of an objective function involving elements of all three parties. However, an additional complication arises: cooperating transport providers and transport organisers each provide a part of the solution but act *independently* and hence do not share all available information. Each party tries to optimise its own operations. In the specific Flemish context where the solution will be applied, the size and spatial distribution of *day care centres* and schools may allow/require bus operators to serve passengers for multiple institutions during a particular trip. In order to operate efficiently, the transport provider (bus operator) needs to solve instances of the capacitated vehicle routing problem with time windows (CVRPTW). However, the pick-up locations to be served by the bus depend on the solution found for the carpooling-based problem proposed by each transport organiser (day care centre, school). The set of pick-up locations constitutes the interface between the respective transport *provider* and transport *organiser* problems.

An overall solution is conceived as follows. Each transport organiser proposes several different *sufficiently good* solutions for its own sub-problem. The transport provider determines its own optimal solution to all of these cases. Finally the optimal case is the one for which the combined cost of the transport provider and transport organisers is minimal.

8.3.3 Solution

This paper focuses on the transport *organiser* problem. Hence, the goal is also to investigate some objective functions in order to be able to find an economically feasible solution. Examples are minimising the number of pick-up locations for the chartered bus or minimising an approximation (based on partial information) for the cost of a chartered bus, which will not necessarily lead to the same results.

The locations to be served by the buses are the DCC guests' *homes* and the *transfer locations* where at least one DCC guest is not served by carpooling (i.e. either not picked up in the morning flow or not dropped off in the evening flow). These DCC guests are considered as being *stuck*.

Solving this combinatorial problem requires heuristics that are efficient with respect to the quality of the solution as well as with respect to the computational effort (time). This paper describes the exploration of the solution space that was carried out to support the design of heuristic solutions. It is interesting to know the level of potential savings in concrete cases in order to find out the suitability of the solutions because implementing them both at practical organisational level and at the level of required computer time is non-trivial.

To give the reader an idea of how a solution can look, an example is shown in Figure 8.1. White circles represent non-transfer locations, light grey circles represent transfer locations and the dark grey circle is the DCC. Locations are connected by a directed edge. A continuous line edge indicates the route of a carpool driver, while dashed edges indicate the route of a chartered bus; routes in different colours represent different drivers/chartered buses. For clarity, a filled black dot indicates that a voluntary driver is available in an origin location. All concepts mentioned in the remainder of the text are based on this case study.

8.3.4 Costs Distribution

Note that the distribution of the cost for the combined carpooling-bus solution over all DCC guests also presents difficult issues to solve. It is to be combined with the refunding of the costs of the carpool drivers. The chartered buses considered in this paper are taxi mini-buses. The fare for a taxi (excluding start cost) is approximately

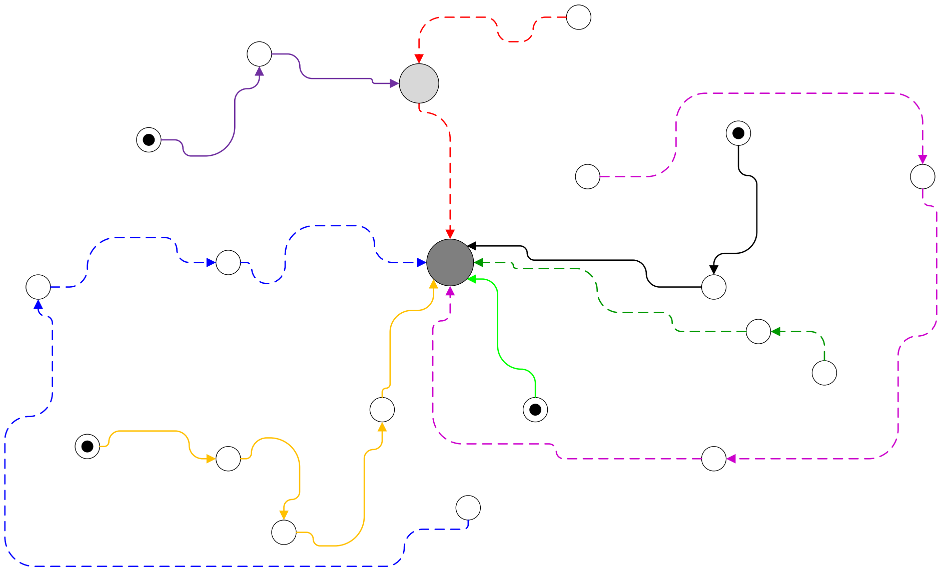


FIGURE 8.1: An example of a possible solution for the DCC problem.

2 Euro/km[1,2]. Travel cost refunding to a volunteer for a trip by private car is at most 0.3460 Euro/km [Flemish Centre for Volunteering, 2017]. Carpooling costs less per unit distance (by at least a factor 5) than the chartered bus travel but on the other hand consumes time offered voluntarily by the drivers. Research to solve the payment and refunding problem is *not* covered in this paper.

8.4 Concepts

The concepts described in the following subsections play an important role in this application.

8.4.1 Flows

The *morning flow* in a day describes the problem of reaching the DCC from the DCC guests' homes, while the *evening flow* describes the trips from the DCC to the DCC guests' homes. The focus in this paper is on the morning flow.

---

[1] `http://www.taxisverts.be/nl/tarieven-taxi-brussel`
[2] `http://www.antwerp-tax.be/nl/diensten-prijzen`

8.4.2 Participants

Participants is a collective name for *DCC guests* and *drivers*. *DCC Guests* are persons who need to reach the day care centre on a regular basis. Drivers are persons who are willing to pick up DCC guests and bring them to the day care centre or to locations near to the day care centre. Typically those drivers are the parents, family members or even other volunteers such as neighbours.

8.4.3 Locations

Four different types of *locations* can be identified: the day care centre (DCC), non-transfer location (NTL), transfer location (TL) and driver destination location. The *day care centre (DCC)* is the location where DCC guests stay during the day; typically those DCC guests travel to the DCC in the morning and leave the DCC in the evening. A *transfer location* is a location where a passenger can transfer from one vehicle to another. A transfer location is subject to capacity (number of people) and time window constraints. Note that a transfer location can coincide with a home location. Transfer locations are the only locations where partial trips are concatenated. They may be served (for both drop-off and pick-up) by a volunteer driver or by a chartered bus. A DCC guest may be dropped off at a transfer location and be picked up by another vehicle. The number of transfers in a passenger trip is limited to one. In the morning (evening) flow, each partial trip by bus ends (starts) in the DCC.

Non-transfer locations (NTL) are homes where the resident DCC guest can be picked up but no transfer is possible. Finally, *driver destination locations* are the end points of the drivers' trips. Typically, destinations are the work locations of the drivers in the morning flow, while in the evening it will be the home locations; however any location could be a destination location (shopping, running errands, leisure, etc.).

A location becomes a *stuck location* if and only if it hosts at least one DCC guest who is not part of a carpool solution and hence, needs to be picked up by a chartered bus.

8.4.4 Periodic Schemes

Each participant specifies one or more *periodic schemes*: such a scheme is a table which specifies the required trips and their spatio-temporal constraints for a sequence of consecutive days. The number of days involved defines the length of the period. Such a scheme is specified to start at a given calendar date and is then repeated until it is replaced by another one. Each individual specifies a *default* scheme and some special ones. Many people use weekly schemes that differ between working and holiday periods of the year.

For a driver, the periodic scheme specifies the driver's availability, the trip origin and destination, the departure and arrival time windows, the maximal detour (relative to the solo-driven trip) and the capacity of the available car for both autonomous and wheelchair-bound DCC guests.

DCC guests specify trip origin, destination and the associated time windows and add specific requirements: (i) the maximum number of transfers to reach the DCC, (ii) the requirement for a wheelchair, (iii) the incompatibility with specific drivers or DCC guests and (iv) the maximum allowed travel time (currently fixed to 90 minutes).

The maximum travel time is based on the current legislation in Flanders with respect to school children transport. Children in special schools are picked up by chartered buses and are allowed to travel at most 220 minutes a day [Committee for Education and Equal Rights and Committee for Mobility and Public Works, 2013-2014; Dillen, 2014].

Availability and time windows for transfer locations are also specified in periodic schemes. Note that in actual practice most of the transfer locations coincide with DCC-guests' homes; household members are prepared to take care of some DCC guests arriving at their location until they are picked up. Hence, the properties for most transfer locations are specified by *personal* periodic schemes. If the transfer location is a public bus stop or community centre (= dedicated transfer location), it has "unlimited" capacity.

Requirements and constraints for a *given calendar day and flow* are derived from the set of periodic schemes that apply for that date. The willingness to extend the time window width and the detour time may have a huge impact on the quality of the solution since at each relevant location the time windows for arrival and departure are *intersections* of time windows specified in the periodic schemes.

8.4.5 Chartered Buses

DCC Guests who cannot reach the DCC by carpool will be picked up by a *chartered bus* (a bus that is hosted by a commercial provider) either at home or at a transfer location. Buses are operated by one or more providers. Providers can serve DCC guests for multiple clients (day care centres, institutions, etc.) during a single trip. Hence, no function is available to map ordered sets of pick-up locations to a client specific cost value.

8.4.6 Organisation of Transport Using Independent Complementary Providers

Solutions to the *school bus routing problem* presented in operations research literature assume (i) long term stable demand, (ii) the use of a single transport mode and (iii) a single operator. On the other hand, the DCC problem is characterised by

- **variable demand**: Demand varies from day to day and if it turns out to be periodic, the period length in the most simple case is the *smallest common multiple* of the length of the default personal periodic scheme lengths which may be very large (people may have two, three and five week periods). In practice the period may be much longer due to the individual specific holiday periods. Consequently, the set of pick-up locations is not constant. Note that this is only a practical but not a fundamental problem.
- **multi-modal travel**: Except for the first part of the trip that may be executed autonomously by DCC guests, trips are either pure bus trips, pure carpool trips or mixed carpool-bus multi-modal trips.
- **multiple operators**: In the DCC problem two types of transport providers cooperate: (i) *carpool drivers* who are assumed to agree on a single globally applicable unit-distance fare (to refund the travel cost) (ii) and one or more *bus operators* offering proposals for transport services. The involvement of independent operators poses a fundamental problem.

For the morning flow, the main goal is to map the set of DCC guest departure (home) locations to a possibly empty set of transfer locations in order to minimise travel costs (this is because DCC guests can have at most one transfer). This coincides with the *pupil-stop assignment* stage mentioned in [Bögl et al., 2015] but in the DCC problem, the support of carpool drivers is required.

The cost to serve all locations of the *mapped* set by buses is unknown to the consultant in charge of finding a solution for the carpooling subproblem. This is because the bus operators are independent and can organise their operations autonomously (for the reason mentioned in Section 8.3.2).

In practice a request for proposals (RFP) needs to be issued and a tender needs to be organised. For practical reasons a medium to a long term contract is preferred. The contract is based on agreed unit-distance fares: the current practice is characterised by (i) single mode, (ii) single provider and (iii) daily changing demand. The agreement specifies the service level and unit-distance prices.

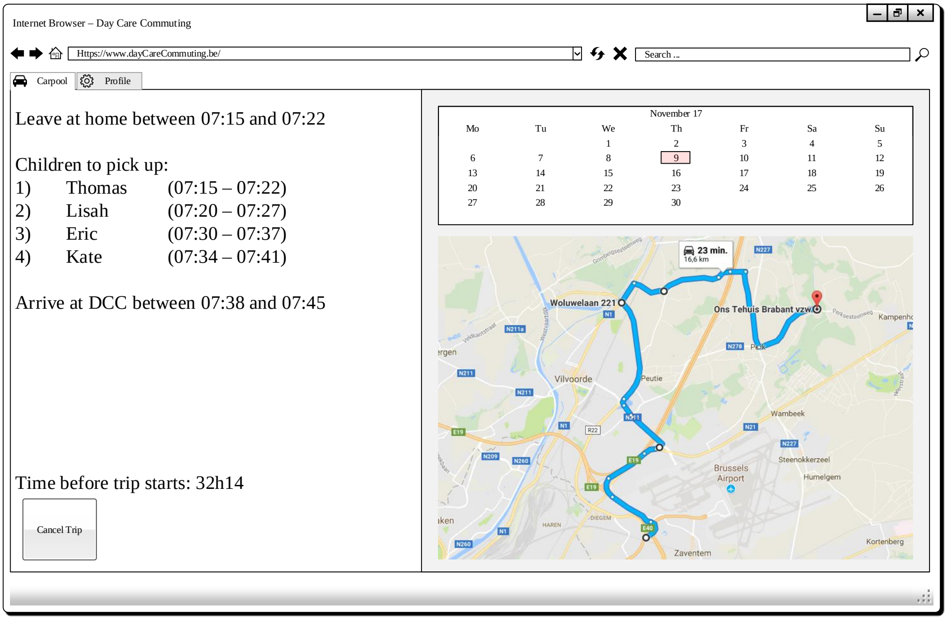
One way to overcome the problem of variance in the data is to use a large set of typical cases as a basis to set up a tender. A period covering the duration of the contract to be negotiated is to be considered. For each day in that period the DCC guest demand and driver availability are derived from the periodic schemes and the set of pick-up locations is determined using several different criteria (e.g. smallest transfer location set, etc.). The N most frequently occurring sets are passed to providers in the RFP.

Finding transfer locations that remain stable in time needs to be done anyway, in particular in the interest of the DCC guests.

8.5 Data

8.5.1 Real Situation: Collection by a Web Application

Many challenges arise when developing an application such as this DCC problem. Users of this application should be well informed, and updating information about planned trips and cancellation should be conducted in a very easy and straightforward way. In Figure 8.2, a possible example (mock-up) of a web application for a trip by a driver can be seen. However, GUI development is not in the scope of this paper.



Leave at home between 07:15 and 07:22

Children to pick up:
1) Thomas (07:15 – 07:22)
2) Lisah (07:20 – 07:27)
3) Eric (07:30 – 07:37)
4) Kate (07:34 – 07:41)

Arrive at DCC between 07:38 and 07:45

Time before trip starts: 32h14

November 17

Mo	Tu	We	Th	Fr	Sa	Su
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

FIGURE 8.2: An example of a web application to support (in this case) a driver. The map was taken from Google Maps [Google, 2017].

The idea is to create a platform where participants can provide all the necessary information as was explained in Section 8.4.

With this information, the application can compute a schedule for every driver and guest. A driver needs to receive information about when she has to leave home, which locations she should visit and in which order she has to do that, while guests need to receive information about who will pick them up (possibly with information about the car or a specific coloured sign to make it easier for (mentally) disabled passengers) and at what time. All this information will be provided according to the participants' preferences, e.g. by email, by text message, etc.

The authors are aware of the fact that, especially for mentally disabled guests, it would be difficult if the daily routing would change a lot. Fundamental research needs to be done before a substantiated reasoning about this issue can be conducted. However, the authors believe that there will be some kind of regularity, not necessarily on a daily base, but on a weekly base. Real challenges will appear when a driver who committed to drive, will cancel this commitment for any reason (e.g. illness). Current technology can be very helpful in this respect such as smartphone apps. Smartphone apps could provide information about the pick-up vehicle such as current location, delay, type of the vehicle, colour, arrival time, etc.

8.5.2 Sensitivity Analysis: Data Sampling

At the moment of writing this paper, there was only data available of origin locations of DCC guests of a specific DCC in Flanders. No data is currently available of individual preferences. Questionnaires are being developed to collect more data in order to be able to compute some realistic cases.

However, data is needed to test the proposed algorithms and to attempt to investigate the feasibility of the solutions. To solve this issue, a *data sampler* was developed. The sampler creates periodic schemes and determines for every participant and location different constraints for every day of the week. A flow chart of the following explanation can be found in Figure 8.3.

The software samples 30 DCC guests. For every DCC guest a home location is sampled; there is a 40 % probability this will become a transfer location. For every week day (excluding Saturday and Sunday), there is 80 % probability that a DCC guest needs to go to the DCC. For every DCC guest, there is a 25 % probability that a home driver is present. For every week day (excluding Saturday and Sunday), this driver has an 80 % probability of being able to drive. In 40 % of the cases the driver's destination is the DCC; in all other cases, the destination is randomly sampled using

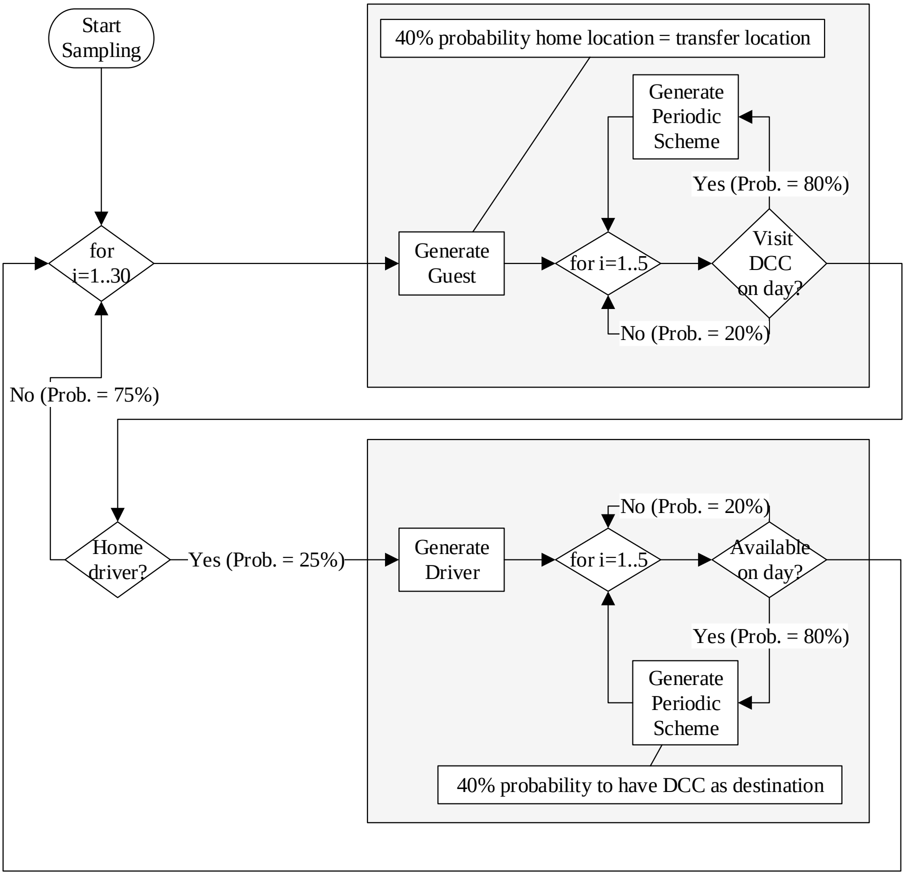


FIGURE 8.3: Flow chart describing the process of generating sample data.

the distribution for *home-work* distances and are uniformly distributed over space (within the borders of Flanders).

Home-work distances are determined by means of *inverse transform sampling* using the cumulative distribution $D(d)$ for the home-work travel distance found by the Flemish household travel survey (OVG) [Declercq et al., 2016]. Thereto, the following equations are used:

$$p \sim U(0,1) \tag{8.1}$$

$$p \in [p_i, p_{i+1}[\Rightarrow d \in [d_i, d_{i+1}] \tag{8.2}$$

$$p \in [p_i, p_{i+1}[\Rightarrow d \sim U(d_i, d_{i+1}) \tag{8.3}$$

where $U(a,b)$ denotes the uniform distribution with boundaries a and b.

Equation (8.1) samples a uniformly distributed value that represents a probability. Equation (8.2) evaluates the inverse $d = D^{-1}(p)$ of the distribution function $p = D(d)$ and delivers a distance range $[d_i, d_{i+1}]$. Finally, Equation (8.3) samples a distance value from that range.

The values for p_i and D_i are given in Table 8.1 which is the numerical specification of Figure 8.4.

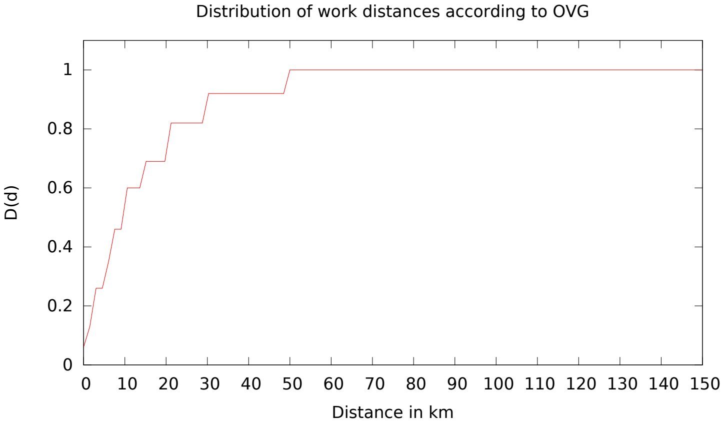


FIGURE 8.4: Cumulative relative frequency distribution $D(d)$ for home-work distance d acquired from the OVG survey. The numerical specification is given in Table 8.1.

Home-DCC distance is estimated using data for a particular Flemish DCC for which all 25 home locations were given. Bird's eye distances from these home locations to the DCC were calculated. Results showed a mean of 5 870.25 metres with a standard deviation of 2 582.05 metres.

The variables *maximum detour time* (MDT) and *time window width* (TWW) are configurable and used for sensitivity analysis. Their impact on the possible solutions will be examined during the experiments in Section 8.9.

First, MDT is used as follows: arrival times of the participants are sampled in the interval [07:00, 10:00]. Departure times are computed based on the travel times from

TABLE 8.1: Numerical specification for the cumulative distribution for the home-work distance in kilometres according to the OVG survey.

d[km]	D(d)
0	0.00
1	0.06
2.5	0.13
5	0.26
7.5	0.35
10	0.46
15	0.60
20	0.69
30	0.82
50	0.92
150	1.00

origin to destination, taking into account the preferred arrival time. In order to get the earliest required departure time, the start time of the departure time window is decreased with the maximum detour time.

Second, in order to ease results interpretation, all applicable time windows have the same length TWW. For participants, time windows are used for (i) trip departure time and (ii) trip arrival time. The same setting is used for the allowed waiting time in home-based TLs. In this case the time window is bound to the preferred departure time of the participant living at that location. Consider an event for which the moment in time having a preferred value t_0 (e.g. trip start time); then, the range for its effective value is given by $[t_0 - TWW/2, t_0 + TWW/2]$.

8.6 Limitations and Assumptions

The proposed algorithms in Section 8.8 are based on the following assumptions:

(i) If a driver visits locations on his way to his destination, all the DCC guests currently present at that location are picked up by that driver, and (ii) when a trip ends for a driver, all the DCC guests are dropped off at once either at the DCC or at a transfer location.

Obviously, paths in a solution respect all the constraints (as discussed in Section 8.4) which are related to locations as well as participants.

Due to lack of real surveyed data, the following assumptions are made while generating test cases: (i) the constraints related to wheelchair usage and incompatibilities between DCC guests are ignored (although supported by the model), (ii) at most one DCC guest is living in every location, (iii) home locations which are also TLs have a capacity of two, (iv) car capacity is always four (excluding the driver) and (v) the arrival time window of the DCC is [07:00, 10:00]. This paper only takes care of the morning flow. The evening flow should be analogously solvable.

8.7 Graph Theoretical Formulation of the Carpooling Subproblem

In this Section the DCC problem in graph theoretical tools is presented. This formulation supports in defining our objective function and optimisation problem. Let $G = (V \cup \{0\}, E)$ be a directed complete graph with 0 as the DCC. The vertex set $V = \{V^t \cup V^{nt} \cup V^w\}$ is composed of a subset V^t of *transfer locations (TL)*, a subset V^{nt} of *non-transfer locations*, and a subset V^w of other locations, which will be referred to as driver destination locations. The transfer locations are of two types. $V^t = \{V^t_h \cup V^t_{other}\}$ (Most transfer locations are the *homes* of the DCC guests, denoted by V^t_h but few are not, denoted by V^t_{other}). The set $V^t_h \cup V^{nt}$ represents the *homes* of the DCC guests, and each home contains (in this paper) one DCC guest. Some of the homes have cars (available to be used), but not all of them. By $car(v_i)$ the car associated with vertex v_i is denoted. If location v_i does not have a car then $car(v_i) = \emptyset$. Let $l : E \to \mathbb{R}^+$ be the length function that associates a length $l(e)$ to each $e = (v_i, v_j) \in E$, which is the time it takes to travel from v_i to v_j. Each vertex v_i has a *time window* interval $[e_i, l_i]$, meaning a vehicle can reach v_i no earlier than e_i and leave no later than l_i. In addition, each vertex $v^t_i \in V^t$ has a *passenger capacity* $c^t_i \geq 2$ which is the maximum number of DCC guests that can simultaneously be present at the transfer location during its time window. Each vehicle i has a *capacity* c^c_i which does not include the driver. It is assumed here that all cars have capacity $c^c_i = cc = 4$, except for chartered buses which have a larger capacity $c_B \in \{8, 12, 30\}$. In addition, each car has a *detour constraint*, which is the maximum extra time the driver of the car is willing to be on the road in order to pick up or drop DCC guests. The detour constraint of the car which belongs to home v_i is denoted by $\delta(i)$.

The focus is on formulating and solving the problem of planning the routes in the morning, where the DCC guests need to be driven from their homes to the DCC. It will be referred to as the *morning problem*. The *evening problem*, of driving the DCC guests from the DCC back to their homes is not covered by this paper, but can be solved by symmetry.

Every vehicle chooses a *feasible route* which begins at some home vertex in $V^t \cup V^{nt}$, picks up and drops some DCC guests in some vertices on its route, and ends up either in the DCC, or in a transfer or work location, without violating the capacity, time window, and detour constraints.

A *feasible passenger route* is a path from the passenger's home to the DCC that respects all time windows, does not involve waiting in more than one TL, and such that its total length, including waiting times, is less than T. An *optimal solution* to the problem is a planning of feasible routes for the vehicles and DCC guests such that the total cost of chartered buses is kept to a minimum. Two different objective functions are investigated: (i) minimising the number of pick-up locations for the chartered buses and (ii) minimising chartered buses and driven kilometres by using a simple version of a Vehicle Routing Problem solver (See Section 8.8.4).

Each car and its driver is associated with some route which corresponds to a directed path in G. Below are some standard definitions related to paths, and a collection of paths, which are called a *path family*.

Definition 8.7.1 (Path, initial, terminal, cardinality, length, distance)**.** Let $G = (V, E)$ be a directed weighted graph, with weight function $l : E \to \mathbb{R}^+$. A *path* P in G is a sequence of distinct vertices $(v_1, v_2, \ldots, v_l)$ such that $(v_i, v_{i+1}) \in E$, for $i = 1, 2, \ldots, l-1$. (Note that P is a directed path in G). The set of vertices and edges of a path is denoted by $V(P)$ and $E(P)$, respectively. The initial and terminal vertices of P, v_1 and v_l, are denoted by $ini(P)$ and $ter(P)$, respectively. Other vertices e_i, where $i = 2, \ldots, l-1$ are called *intermediate* vertices of the path. The *cardinality* of a path is defined by $|P| := |V(P)| = l$, and the *length* of the path is defined by $l(P) = \sum\limits_{e \in E(P)} l(e)$. For vertices u and v, the *distance* from u to v is the length of the shortest path from u to v, and is denoted by $dist(u, v)$. Note that $dist(u, v)$ may be different from $dist(v, u)$ since the graph is directed and not symmetric. If P and Q are paths where the terminal vertex of P equals the initial vertex of Q, then the concatenation of P and Q is denoted by $P * Q$.

Definition 8.7.2 (Path family)**.** A collection of paths, not necessarily disjoint, is denoted by a script letter $\mathcal{P}$ or $\mathcal{S}$ and is called a *path family*. It is written as $V[\mathcal{P}] := \bigcup\{V(P) : P \in \mathcal{P}\}$, $E[\mathcal{P}] := \bigcup\{E(P) : P \in \mathcal{P}\}$, i.e. $V[\mathcal{P}]$ and $E[\mathcal{P}]$ are the sets of vertices and edges covered by the paths in $\mathcal{P}$, respectively. Denote by $ter[\mathcal{P}] := \bigcup\{ter(P) : P \in \mathcal{P}\}$ the set of terminal vertices of paths in $\mathcal{P}$ and similarly $ini[\mathcal{P}] := \bigcup\{ini(P) : P \in \mathcal{P}\}$. The *out-degree* of a vertex v in a path family $\mathcal{P}$ is denoted by $deg^+_{\mathcal{P}}(v)$ and it is defined as the out degree of v in the graph induced by $E[\mathcal{P}]$. The term $deg^-_{\mathcal{P}}(v)$ is similarly defined.

In a feasible solution to our problem, each car starts from a location that contains a car, moves along some path in the graph picking up at least one DCC guest at every vertex on the path and ends up either in a TL or the DCC, dropping all the DCC guests in the car there, such that the car capacity constraints, the time window constraints of all the vertices of the path, and the detour constraints are satisfied. The cars that end up in a TL may continue to their driver destination (work or other errands), but as far as the solution is concerned, those trips from a TL vertex to a driver destination vertex are ignored . This motivates the following definitions:

Definition 8.7.3 (Constraint Feasible Path (CFP)). A *constraint feasible path (CFP)* in G is a path $P = (v_1, v_2, \ldots, v_l)$ where $v_1 \in V^t \cup V^{nt}$, v_1 contains a car, $v_l \in V^t \cup \{0\}$, $l \leq c+1$, satisfying the time window constraints of all the vertices in the path, as well as the detour constraints. The car associated with location $ini(P)$ is denoted by $car(P)$.

Since no DCC guest can wait in more than one TL, and once all DCC guests are picked up from a non-terminal location there is no other driver who should stop at his/her home, a family $\mathcal{P}$ of CFPs should have the property that every pair of paths in $\mathcal{P}$ is either disjoint, or have a unique vertex v_t in common - their common endpoint which is the DCC or a TL (where the drivers drop their DCC guests), or v_t is an endpoint for one of the paths, and the other path contains v_t and ends in the DCC - implying that the second car picks up all the DCC guests in the TL and drives them to the DCC (see Figure 8.5). This motivates the following definition:

Definition 8.7.4 (Semi-Disjoint Constraint Feasible Paths (SD-CFP)). A path family $\mathcal{P}$ consisting of CFPs is *semi-disjoint* if any pair of paths $P_1, P_2 \in \mathcal{P}$ satisfies one of the following three conditions:

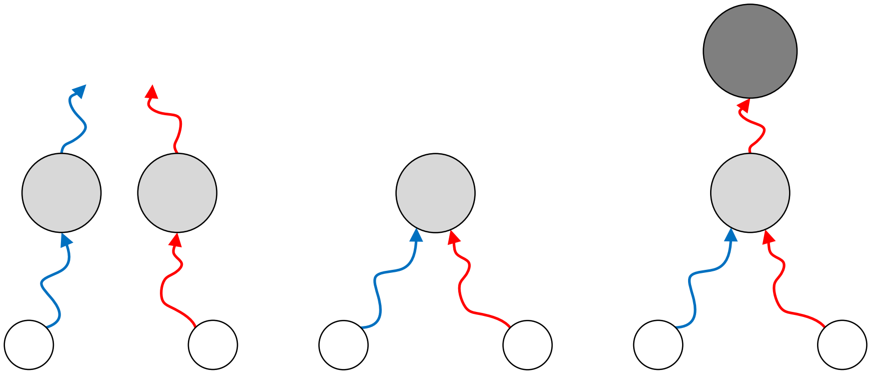
(a) $V(P_1) \cap V(P_2) = \emptyset$ (Figure 8.5a)

(b) $V(P_1) \cap V(P_2) = \{v_j\}$ for some $v_j \in V^t \cup \{0\}$ and $v_j = ter(P_1) = ter(P_2)$ (Figure 8.5b)

(c) $V(P_1) \cap V(P_2) = \{v_j\}$ for some $v_j \in V^t$ where $v_j = ter(P_1)$ and $ter(P_2) = \{0\}$ (Figure 8.5c)

In addition, for any such vertex v_j above, the passenger capacity is not violated, in other words the number of paths in $\mathcal{P}$ terminating at v_j is at most c_j^t, and the amount of time car(P_2) needs to wait for car(P_1) to reach v_j (in case car(P_1) reaches v_j after car(P_2)) is feasible with respect to $\delta(car(P_2))$, the detour constraint for car(P_2).

The problem that is addressed is finding a family $\mathcal{P}$ of semi-disjoint constraint feasible paths (SD-CFP) which covers as many vertices in $V^t \cup V^{nt}$ as possible. This will guarantee that as many children as possible are picked up by carpooling, and as few as possible need to be picked up by a paid vehicle. This is an NP-hard optimisation problem. The search space of possible solutions is exponentially large.



(A) Two drivers bring guests to distinct transfer locations.

(B) Two drivers bring guests to a joint transfer location. Both guests are dropped at this location.

(C) Two drivers bring guests to a transfer location. One of the drivers picks up these DCC guests and possibly other guests who are already at this location.

FIGURE 8.5: Possible relations between two CFPs. White circles represent non-transfer locations, light grey circles represent transfer locations and the dark grey circle is the DCC. Locations are connected by a directed edge which represents the route of a driver; routes in different colours represent different drivers.

8.7.1 Finding Family of Constraint Feasible Paths

An important step in the algorithm is finding possible paths of drivers. It means finding all CFPs in G, which is called *CFP family*, and denoted by $\mathcal{S}$. A collection of all CFPs is found in two stages:

In the first stage, a pre-process step is conducted; a compatibility digraph $G^c = (V \cup \{0\}, E^c)$ is constructed. For every ordered pair (u, v) of vertices in G where u contains a vehicle, an edge $e = (u, v) \in E^c$ is defined if and only if the vehicle in u can drive to location v considering the time $l(e)$, and if all of the following constraints are met: the time windows in u and v, the detour constraint $\delta(u)$, and the personal incompatibility between DCC guests. (If at least one of these constraints is not met, then (u, v) is not an edge in G^c).

The second stage includes finding all constraint feasible paths $\mathcal{S}$; using G^c the set of all CFPs in G of cardinality at most $cc + 1$ (where cc denotes the maximum car capacity is found, and is usually 4) in the following way: for each vertex v with a car, let $N = N^+_{G^c}(v)$ be the set of out neighbours of v in the graph G^c. Every ordered subset $s \subseteq N$ is checked with a terminal vertex in $V^t \cup \{0\}$, and of maximum size four. (There are at most $k(k-1)\dots(k-3)$, such subsets, where $k = |N|$). If the combined path $v \star s$ is a CFP, it is added to the set $\mathcal{S}$.

8.8 Algorithmic Solutions

As described in Section 8.7.1, the CFP family is computed. This CFP family represents all the possible paths drivers can use respecting all driver related constraints. For a solution to the DCC problem, an algorithm should choose at most one path for every driver. It is important to know that when a path is chosen, other paths in the CFP family may be *disabled*, since they do not obey the conditions of SD-CFP - see Definition 8.7.4. If a path is disabled, it cannot be chosen any more for the current solution. Hence, if path $P \in \mathcal{S}$ is chosen, then disable every $Q \in \mathcal{S}$ for which P and Q do not constitute SD-CFP as described in Definition 8.7.4. This also means that every driver appears at most once in the solution and non-transfer locations are only visited once, since it is irrelevant to visit a location without a pick-up action.

8.8.1 Exhaustive Search Method

First an exhaustive search method is used to find all families of SD-CFPs and choose a family which delivers the best score for the chosen goal. Goals and scoring functions are detailed in Section 8.8.4.

The input for the algorithm is a family of CFPs, as described in Section 8.7.1. Given this path family, a graph is built where a vertex represents a driver and vertices are connected by an edge if and only if they share at least one non-DCC location in one of their paths. This graph is partitioned into *connected components*. Two drivers belonging to different components can be handled independently. Clearly, each component constitutes an independent smaller problem. The more components are found, the better for the (costly) exhaustive search method. Every component can be handled separately by considering all the possible combinations of driver paths.

The enumeration of the solutions goes as follows: The algorithm starts by choosing a path for the first driver, next a path for the second driver and so on. When a path is chosen, the current combination of paths is checked on validity. When this combination is not valid, every possible combination with paths of the remaining drivers can be ignored, and the algorithm backtracks to the next CFP of the driver. As soon as a (possibly empty) path is successfully added for every driver, the combination is registered as a solution and scored; the search then continues to find the next one. A driver can appear at most once in each solution and the non-transfer locations will be handled by at most one driver. This means that during the enumeration (in some cases) a large number of paths of other drivers will be cancelled out.

Path enumeration order affects the required computational effort. This is a subject for further research. Two possible orders of search space scanning are represented schematically in Figure 8.6. $P_i D_j$ stands for path i of driver j. A path from the root to a green leaf represents a feasible solution, while paths ending in a red leaf are infeasible. An arrow represents the feasibility check of the driver path it points to. A red line through an arrow indicates that this check is not necessary any more since the previous feasibility check failed. Figure 8.6a and Figure 8.6b use the same input data, but use a different order of combining the paths of the drivers. The order of the chosen paths is not relevant for the solution since (i) every non-empty subset of the drivers and for each of the subset members exactly one of the feasible paths is considered, (ii) in a feasible solution, paths (irrespective of the driver) need to be semi-disjoint and (iii) feasible time windows are predefined (i.e. do not depend on e.g. the first arrival time of a passenger at a transfer location).

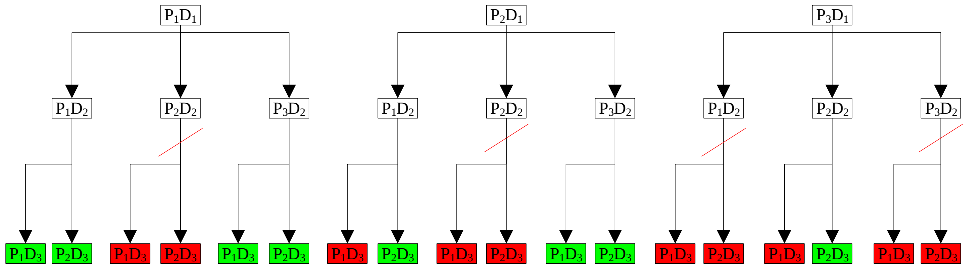
In total there are 18 possible combinations, which results in 27 feasibility checks from which 19 are actually conducted for Figure 8.6a and 24 feasibility checks for Figure 8.6b from which 21 are actually conducted.

The number of cases to evaluate can be (extremely) large. Given a set of drivers $D = \{d_1, d_2, \ldots, d_n\}$ and the set of paths for a driver d_i, $\mathcal{P}(d_i) = \{P_1^{d_i}, P_2^{d_i}, \ldots, P_m^{d_i}\}$, the total number of cases to evaluate equals

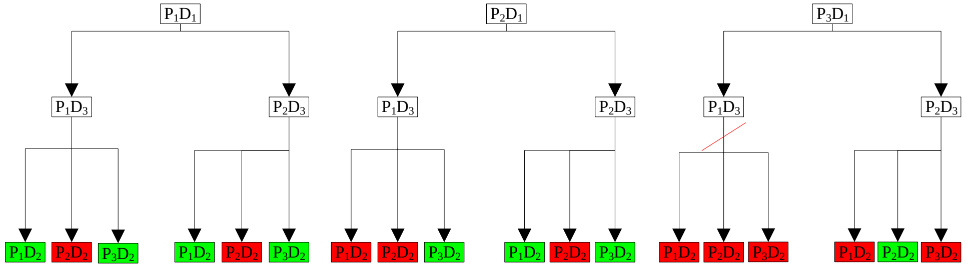
$$\sum_{D_k \in 2^D \setminus \varnothing} \Big(\prod_{d \in D_k} |\mathcal{P}(d)| \Big) \tag{8.4}$$

where 2^D represents the power set. This may be very large; moreover, every solution is some subset of these paths which is SD-CFP, so the computation time may be very large

In order to avoid infeasible executions times, computations are stopped after 60 minutes.



(A) An exhaustive search where the paths of drivers D_1, D_2 and D_3 are combined. With respect to the infeasible combinations, in this case, 19 checks are needed.



(B) An exhaustive search where the paths of drivers D_1, D_3 and D_2 are combined. With respect to the infeasible combinations, in this case, 21 checks are needed.

FIGURE 8.6: An example of the exhaustive search algorithm. P_iD_j stands for path i of driver j. Leafs in red are not feasible, while leafs in green are feasible. In total there are 18 possible combinations. Every arrow indicates a feasibility check. A red line through an arrow indicates that this check is not necessary any more since the previous feasibility check failed.

8.8.2 Heuristics 1: Focusing first on DCC

This is a two-stage heuristic. In the *first* stage only the CFPs in the SD-CFP family that lead to the DCC are considered. In other words, only the drivers who drive to the DCC are taken into account. The heuristic attempts to find a routing for them that picks up as many DCC guests as possible. For each driver a route is chosen, so that all drivers pick up as many DCC guests as possible. Once such a routing is found for all the drivers who reach the DCC, the set of TLs covered by these routes are considered. In the *second* stage, as many DCC guests as possible are collected at these TLs (respecting the constraints of the path). A greedy approach for these subproblems are used since they are NP-hard.

8.8.3 Heuristics 2: Focusing first on Transfer Locations

In comparison with the two stage heuristic as described in Section 8.8.2, the first stage of this algorithm only considers the CFPs leading to dedicated transfer locations (even if the trip could lead to the DCC). In this case the algorithm attempts to bring as many DCC guests to these locations in order to reduce the number of pick-up locations for the chartered buses. In the second stage, remaining CFPs going to the DCC (if any) are taken into account. For the experiments, nine schools in the surrounding area of the DCC were selected to be used as dedicated transfer locations. Other predefined home-based transfer locations were ignored in this solution.

8.8.4 Scoring

Determination of Bus Trips

As indicated in Section 8.3.2 it is not possible for legal and operational reasons to solve the problem in an integrated way. In order to find a good solution for the carpooling part it is assumed that the bus operator serves the DCC independently of any other client. For each carpooling proposal, a CVRPTW needs to be solved for the bus service. Open-source solvers such as OptaPlanner [OptaPlanner, 2017] and GraphHopper [GraphHopper, 2017] have been considered. It was decided not to use these tools because of the following reasons: (i) it is not known whether these tools are deterministic in the sense that they always give the same results with the same input and (ii) set-up costs and execution times are high.

A non-expensive approximate solution for the CVRPTW is required because many cases need to be evaluated. Therefore, for the morning flow, the cost for each bus is estimated by assuming that it starts at a location where a passenger is to be picked up. This is realistic because the usual contracting rules specify that only the distance driven with at least one passenger on board can be charged. The bus is assumed to move counter-clockwise around the DCC and does not insert wait periods at serviced locations. The DCC location is used as the origin of a polar coordinate system and the angular argument corresponding to the consecutively visited locations is non-decreasing. The bus picks up as many passengers as possible; the trip leads to the DCC and respects the vehicle capacity constraint as well as the timing constraints for each passenger and transfer location. The chronological pick-up order coincides with the counter-clockwise location visit order for each particular bus but not necessarily for the set of all passengers. A bus may need to skip a location in the counter-clockwise order due to timing constraints and because a bus does not wait at service locations. Additional bus trips are scheduled until all passengers reach the DCC and each trip is served by an additional bus.

The quality of the approximation is assessed by considering the total distance driven with passengers on board; this is used to determine the amount to invoice. (It is obvious that the cost per kilometre will increase if more *empty kilometres* are required.) Assume that a solution with N_B buses is found by the proposed approximation. Let $N_L(b)$ denote the number of locations served by bus b. The j-th location served by bus b is denoted by $L_{b,j}$ and the shortest distance between locations L_i and L_j is denoted by $d(L_i, L_j)$. Then the minimum total distance D_{N_B} driven by N_B buses is limited by

$$D_{N_B} \leq \sum_{b=1}^{N_B} \Big(\sum_{j=1}^{N_L(b)-1} d(L_{b,j}, L_{b,j+1}) \Big) \tag{8.5}$$

Due to the triangle inequality the total length D of all bus trips (irrespective of the number of buses used) cannot exceed the distance for a single-bus star-based solution in which each subtrip between consecutive pick-up locations passes at/near the DCC so that:

$$D \leq d_1 + 2 \cdot \sum_{i=2}^{n} d_i \tag{8.6}$$

where n is the number of locations served and d_i is the distance between the i-th location and the DCC.

The distance associated with the solution of the CVRPTW is at least

$$\sum_{i=1}^{n} d_i = D_{min} \leq D_{CVRPTW} \tag{8.7}$$

Due to the triangle inequality and because the number of subtrips driven twice in equation (8.5) cannot exceed the number of subtrips driven twice in equation (8.6)

$$D_{min} \leq D_{CVRPTW} \leq D \leq 2 \cdot D_{min} \tag{8.8}$$

$$D_{min} \leq D_{N_B} \leq D \leq 2 \cdot D_{min} \tag{8.9}$$

for all values of N_B. Hence, the total distance D_{N_B} found by the proposed approximation and the total distance found by a CVRPTW solver can differ by at most a factor two.

Scoring Functions

Before implementing algorithms and deploying proposed solutions, the economic feasibility is to be investigated. The *effectiveness* of carpooling acting as a feeder for bus stops could be expressed by (i) the fraction of DCC guests delivered by carpools at the DCC, (ii) the fraction of locations where people need to be picked up by the chartered bus and (iii) the number of buses and the amount of kilometres needed to pick up the stuck DCC guests. Two scoring functions are developed, one is based on (i) and (ii) (minimising stuck locations), while the other one is based on (iii) (minimising chartered bus costs).

For the *minimising stuck locations scoring function*, the following variables are taken into account: (i) the number of DCC guests reaching the DCC by voluntary drivers (*nDccGuests*), (ii) the number of transfer locations in which DCC guests are stuck (*nStuckTl*), (iii) the number of origins in which DCC guests are stuck (*nStuckOrigin*), (iv) the number of locations in the input (*nLocations*), (v) the total number of DCC guests in the input (*nGuests*).

This leads to the scoring function represented in Equation 8.10.

$$score = \frac{nDccGuests}{nGuests} + \left(1 - \left(\frac{nStuckTl + nStuckOrigin}{nLocations}\right)\right) \tag{8.10}$$

The first term is a measure for the number of DCC guests whose travel problem was completely solved. The second term accounts for the number of locations that need to be visited by the (expensive) bus.

For the *minimising chartered bus costs scoring function* other variables are of interest: (i) the number of chartered buses *nCharteredBuses* and (ii) the average amount of kilometres travelled by the chartered buses (*nAvgDistance*). This leads to the scoring function as can be seen in Equation 8.11.

$$score = -1 \cdot (nCharteredBuses \cdot 60 + nCharteredBuses \cdot nAvgDistance \cdot 0.5) \tag{8.11}$$

It is assumed that 60 €/bus covers the cost of a driver and 0.5 €/km covers the cost of the vehicle per kilometre. Because the algorithm is a minimiser, the score is reversed by multiplying it by -1.

Note that both scoring functions will yield different results for the exhaustive search. In Section 8.9.2 it will be clear that minimising the number of stuck locations does not mean minimising driving costs of a chartered bus and vice versa.

8.9 Experiments

8.9.1 Scenarios

The goal of this paper is to find out whether cost savings can be achieved by applying carpooling solutions. Since no real case data about participant's preferences are available, a number of combinations should be computed in order to be able to give some advice to DCCs. Two variables will be configurable during the simulation: (i) maximum detour time (MDT) and (ii) departure/arrival time window width (TWW). The used values for the TWW (in minutes) are $\{5, 15, 30, 45\}$ and the used values for MDT (in minutes) are $\{5, 15, 30, 45\}$. This results in 16 different *combinations* to examine.

The flow of the experiments is as follows: first a random *case* is generated based on a given MDT and TWW as described in Section 8.5.2. Such a case represents a realistic case study for a given DCC. For this random case a solution is computed for every day of the week (except Saturday and Sunday) based on the combination of the three algorithms and the two scoring functions as explained in Section 8.8.4. Four different experiments are conducted: (i) the exhaustive search method with the minimising stuck locations scoring function, (ii) the exhaustive search method with the minimising chartered bus costs scoring function, (iii) the two-stage heuristic (DCC) with the minimising chartered bus costs scoring function and (iv) the two-stage heuristic (transfer locations) with the minimising chartered bus costs scoring function. This allows us to compare the combination of the algorithm and scoring functions because they are executed on exactly the same data. In our experiments ten random cases are generated per combination MDT and TWW. This means that in total 800 solutions are computed per algorithm and scoring function combination. An overview of the experiment flow can be seen in Figure 8.7.

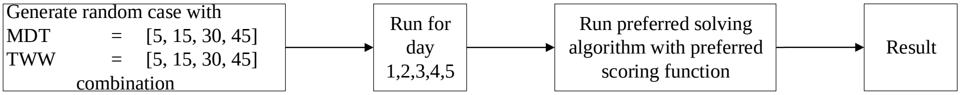


FIGURE 8.7: An overview of the execution of the experiments.

8.9.2 Results

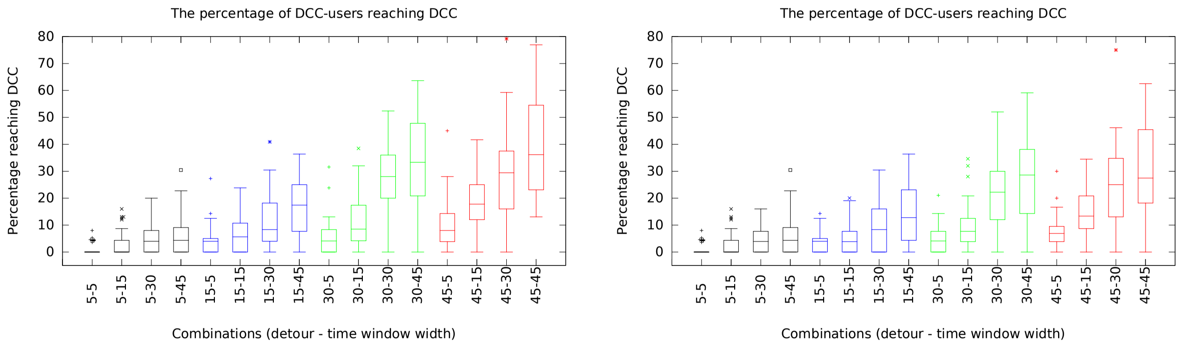
The algorithms and scoring functions are applied to exactly the same data, the results of the cases without voluntary drivers can be found in Table 8.2. Column headers indicate the combination maximum detour time (MDT) and time window width

(TWW), the number of stuck locations (nStuckLoc.), the number of buses needed to serve all the DCC guests (nBus) and the average distance travelled per bus (Avg. dist/bus). Note that these numbers are average values. On average around 24 DCC guests need to go to the DCC and depending on the value of the TWW, the number of needed chartered buses vary between 5 and 8. To have a realistic idea, one should take the ceiling of the number (2.4 chartered buses means 3 in reality for example).

TABLE 8.2: Results of the cases without voluntary drivers. Note that this are average results of 50 cases (five days per iteration and ten iterations).

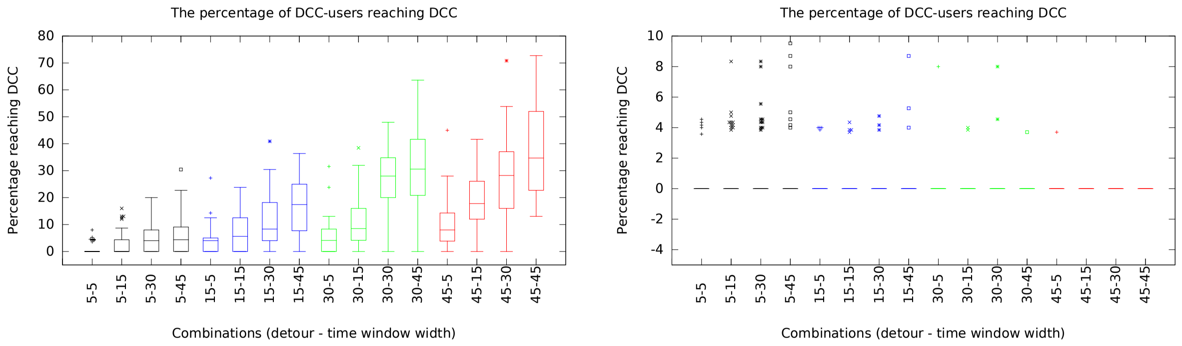
Combination		**nStuckLoc.**	**nBus**	**Avg. dist/bus [km]**
MDT	**TWW**			
5	5	23.44	5.32	23.27
5	15	23.82	7.20	27.93
5	30	23.96	6.06	35.93
5	45	23.36	4.96	39.60
15	5	24.16	6.28	22.49
15	15	24.72	7.16	28.92
15	30	23.96	5.76	34.63
15	45	23.84	5.02	38.38
30	5	24.52	6.10	24.67
30	15	24.30	7.14	27.27
30	30	23.72	5.94	34.42
30	45	23.96	5.06	39.97
45	5	24.52	6.26	23.39
45	15	24.22	7.06	28.18
45	30	24.26	6.18	33.86
45	45	24.08	5.10	39.94

Figure 8.8 shows box plots for the percentage of DCC guests reaching the DCC by carpooling with voluntary drivers. Each box plot emerges from a set of randomly sampled cases for a particular pair of *MDT* and *TWW* values.



(A) Results of the exhaustive search method with the minimising stuck locations scoring function.

(B) Results of the exhaustive search method with the minimising chartered bus costs scoring function.



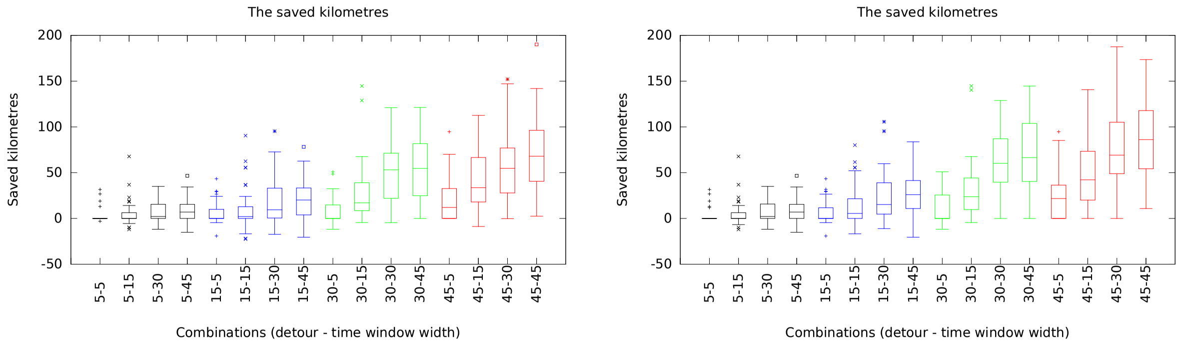
(C) Results of the two-stage heuristic (DCC) with the minimising chartered bus costs scoring function.

(D) Results of the two-stage heuristic (transfer locations) with the minimising chartered bus costs scoring function.

FIGURE 8.8: An overview of the results of the percentage of DCC guests actually reaching the DCC by using voluntary drivers.

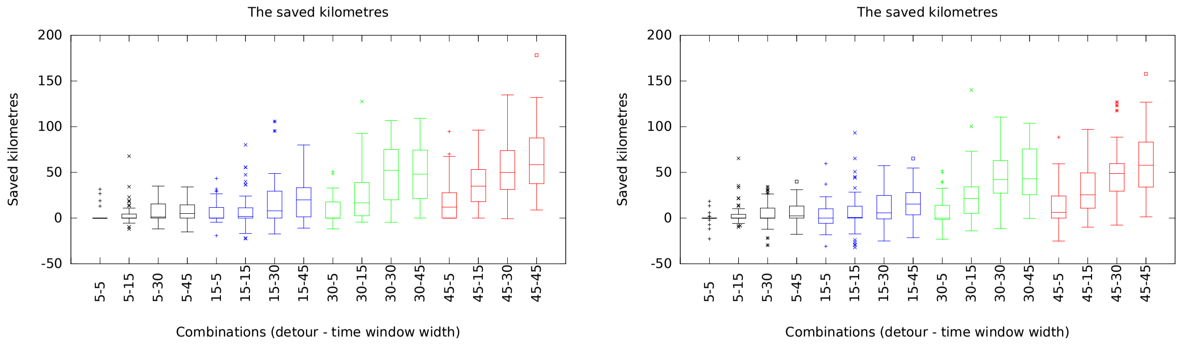
One can immediately observe a low percentage of DCC guests reaching the DCC by use of volunteers in Figure 8.8d. This is expected: the heuristic does not aim to bring as many DCC guests to the DCC by volunteers, but aims to minimise the number of stuck locations. Nevertheless, the saved amount of kilometres is comparable with the other simulations as will be discussed later on. For the other three figures, similar results can be observed. There is a large dispersion within the results of each combination of MDT and TWW. For one particular combination even up to 80 % can reach the DCC by volunteers. However in the same combination, there are cases with 15 % as well. One average decent results can be found with a MDT of 30 minutes and TWWs from 30 minutes on. In that particular case, on average 30 % of the DCC guests reach the DCC by using volunteers.

Another interesting measurement is the total saved amount of kilometres; this is a combination of the number of saved chartered buses and the amount of kilometres they travel. These results can be seen in Figure 8.9.



(A) Results of the exhaustive search method with the minimising stuck locations scoring function.

(B) Results of the exhaustive search method with the minimising chartered bus costs scoring function.



(C) Results of the two-stage heuristic (DCC) with the minimising chartered bus costs scoring function.

(D) Results of the two-stage heuristic (transfer locations) with the minimising chartered bus costs scoring function.

FIGURE 8.9: An overview of the results of the saved amount of kilometres for the chartered vehicles.

On overall, one can see that the exhaustive search method and the two-stage heuristic (DCC) have similar results. The two-stage heuristic (DCC) and the two-stage heuristic (transfer location) are performing similarly although almost no DCC guests reach the DCC by carpooling. When taking a closer look at Figure 8.9b, it is clear that this method is the best. In best cases, up to 200 km can be saved, but on average in realistic cases it will be between 75 km and 100 km. Note that values can have negative results as well. This could happen when more vehicles were needed. This could be the case when particular DCC guests were picked up, which made the route of the chartered buses less efficient. The reference value is the situation were no voluntary drivers are present.

A more detailed overview can be found in Table 8.3. For the percentage of DCC guests reaching the DCC by volunteers, similar results can be observed for $S1$, $S2$ and $S3$. The results for $S1$ are slightly better because the used scoring function is created to maximise the number of DCC guests reaching the DCC. Results for $S4$ are disastrous for this measurement, but it was expected since the method is not aimed at reaching the DCC by volunteers. The next measurement is about the number of stuck locations. All experiment types have similar results regarding this measurement. In this case, it is hard to find out which experiment type outperforms the others. As can be seen in Table 8.2, on average the simulations start at around 24 stuck locations. Depending on the MDT and TWW variables, a reduction of up to ten locations can be achieved. Again for the number of buses and the amount of driven kilometres per bus, the four experiment types perform similarly. By comparing it with Table 8.2, one can see that in many cases up to two vehicles can be saved.

TABLE 8.3: Overview of the results with (i) $S1$: exhaustive search method (minimising stuck locations scoring function), (ii) $S2$: exhaustive search method (minimising chartered bus costs scoring function), (iii) $S3$: two-stage heuristic (DCC) (minimising chartered bus costs scoring function) and (iv) $S4$: two-stage heuristic (TLs) (minimising chartered bus costs scoring function). Grey cells indicate that at least one of the simulations were stopped after an hour of computation. Column headers are (i) combination of maximum detour time (MDT) and time window width (TWW), (ii) the percentage of DCC guest which reach the DCC with carpooling (% reach. DCC), (iii) the number of stuck locations (nStuckLoc.), (iv) the number of buses needed for the stuck passengers (nBus) and (v) the average distance per bus to serve the stuck passengers (Avg. dist/bus).

Combination		**% Reach. DCC**				**nStuckLoc.**			
MDT	**TWW**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
5	5	0.94	0.94	0.94	0.43	23.22	23.22	23.22	23.34
5	15	3.02	2.94	3.02	1.01	23.10	23.12	23.10	23.52
5	30	4.42	3.84	4.34	1.00	22.88	22.98	22.92	23.48
5	45	6.34	6.08	6.34	0.86	21.70	21.72	21.88	22.72
15	5	4.14	3.81	4.14	0.33	23.16	23.22	23.16	23.78
15	15	6.72	4.61	6.80	0.32	22.94	23.32	23.04	23.70
15	30	11.10	9.85	10.93	0.25	21.00	21.24	21.34	22.16
15	45	16.61	14.09	16.53	0.34	19.58	19.98	19.90	21.26
30	5	4.89	4.24	4.89	0.16	23.32	23.48	23.32	23.90
30	15	11.52	9.05	11.36	0.16	21.46	21.86	21.54	22.30
30	30	27.23	21.42	26.48	0.25	17.04	18.04	17.44	18.92
30	45	32.47	26.29	30.55	0.08	16.12	17.04	16.64	17.78
45	5	9.05	7.01	9.05	0.08	22.28	22.74	22.30	22.98
45	15	17.84	14.70	17.84	0.00	19.84	20.48	19.90	20.88
45	30	30.01	25.06	28.44	0.00	16.88	17.80	17.36	18.24
45	45	38.87	30.81	36.71	0.00	14.64	15.74	15.24	16.36

Combination		**nBus**				**Avg. dist/bus [km]**			
MDT	**TWW**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
5	5	5.30	5.30	5.32	5.38	22.97	22.87	22.86	23.10
5	15	7.06	7.06	7.06	7.02	27.84	27.83	27.86	28.03
5	30	5.86	5.82	5.86	5.90	35.78	35.88	35.75	36.34
5	45	4.72	4.70	4.76	4.80	39.49	39.67	39.28	39.63
15	5	5.98	5.96	5.96	6.18	22.74	22.69	22.73	22.55
15	15	6.80	6.72	6.84	6.74	28.97	28.59	28.82	29.78
15	30	5.32	5.20	5.30	5.40	33.88	33.84	34.13	34.80
15	45	4.38	4.26	4.38	4.54	38.90	38.38	38.90	38.14
30	5	5.80	5.62	5.74	5.96	24.70	24.98	24.67	24.36
30	15	6.28	6.04	6.34	6.24	26.80	27.08	26.78	26.95
30	30	4.76	4.32	4.90	4.90	32.33	33.26	31.46	33.28
30	45	3.86	3.52	4.14	4.02	37.91	37.00	37.07	38.04
45	5	5.52	5.30	5.52	5.76	22.93	23.08	23.03	23.12
45	15	5.72	5.50	5.82	6.02	27.74	27.00	27.97	27.96
45	30	4.72	4.02	4.84	4.92	32.11	32.79	32.40	32.93
45	45	3.58	2.98	3.84	3.74	36.50	37.48	36.65	38.24

8.10 Discussion

The results of the simulations should be carefully interpreted. In this paper, the results were in most of the cases averaged over the number of runs conducted for every combination of maximum detour time (MDT) and time window width (TWW). It was observed that there was a very large dispersion between the results. In some cases even up to 80 % of the DCC guests could reach the DCC without making use of a chartered bus, however in some cases nobody did. The authors suppose that a MDT of 30 minutes in combination with a 30 minute TWW is realistic for a majority of the participants. The idea of adding fixed bus stops resulted in similar results as for the two stage heuristic (DCC). In order to judge both alternatives better, the amount of driven kilometres of the volunteers should be taken into account. It may be that volunteers would drive significantly less kilometres when they can bring passengers to surrounding transfer locations instead of to the DCC. The exhaustive search method gave the best results depending on the goal. Nevertheless, the developed heuristic method approached this method very well, it is faster and results did not differ too much.

Unfortunately, the research could only make use of a very limited data set. As described in Section 8.5.2, only data about home locations were available, other information was sampled based on statistical information about Flanders. Being in possession of the additional needed data could give us a very good insight into our solution space. Due to the large dispersion in the results, the average is quite low.

It was decided to keep the MDT and TWW fixed for all the participants. In daily life, this will not be the case, but it was necessary to be able to produce comparable results. If different MDTs and TWWs were randomly assigned to participants, it would be very hard to interpret the results.

For the cases presented in Section 8.9.2, the actual calculation of a solution, given the possible paths of a driver took on average 53.97 seconds (remember that simulations were stopped after 1 hour) for the exhaustive search method, while it only took 0.25 seconds to do the same with a heuristic. In order to have better insight in the results, simulations were conducted for which the optimal result could not be found within an hour. This was done by increasing the number of potential DCC guests from 30 to 150. However, similar results were found. The heuristic methods came close to the approximation for the optimal solution found by the truncated exhaustive search method. It became also clear that a larger number of stuck locations did not mean that more buses were needed. This can be a very useful insight for new heuristics. It could be beneficial to distribute DCC guest as much as possible over various transfer locations in order to be picked up later on by a chartered bus.

8.11 Conclusion and Future Research

In this paper, the impact of two variables (*maximum detour time* and *time window width*) on the solution of the DCC problem was investigated. Results were somewhat disappointing as a solution to a transportation problem in the sense that it is not possible to take more than 50 % of the DCC guests directly to the DCC without using a chartered bus. The possibility to save chartered buses seemed to be hard as well. The data that was sampled is based on real data and should give a more or less accurate view of the situation.

In this paper, data of a DCC in Flanders was used. On average only 24 DCC guests (30 DCC guests sampled with 80 % probability) would visit the DCC every day. One fourth of them had a voluntary driver. Due to the limited search space, most of the cases could be simulated by the exhaustive search method. For these cases heuristics are not really needed. When the number of DCC guests and/or drivers increases, the exhaustive search method fails to complete in many cases within an hour and hence heuristic solutions are very useful. The already developed heuristics approximate the results of the exhaustive search method quite well.

The final conclusion is that the flexibility (and probably also the number of volunteers) will be a major factor with regards to the feasibility of the different solutions. The goal here is finding a set of volunteers together with specific constraints for which the average results remain more or less stable.

In future research the proposed methods can be applied to other use cases such as schools and companies. Since there will be many more passengers, it could be possible that the exhaustive search technique will not be usable. However, research can be conducted to sort the drivers in a specific order, in order to exploit the pruning steps mentioned in Section 8.8.1 and hence, reducing the execution time. The influence of the number of voluntary drivers could also be investigated. It is clear that if the number of drivers increases, the solution would be better. The idea here is to find out as of which number of drivers results are getting substantially better. Furthermore, research on the division of the costs between the participants should be conducted.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors for the staff members of UHasselt - Hasselt

University. The cooperation with Irith Ben-Arroyo Hartman was funded by UHasselt - Hasselt University with BOF-project R-7238, "Methods from graph theory to support sustainable transport".

The authors would like to thank Nadine Smeyers, Mandy Snoeks and Jan Vuurstaek for reviewing this paper and for their valuable comments.

Bibliography

N. Agatz, A. Erera, M. Savelsbergh, and X. Wang. Optimization for Dynamic Ride-Sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012. ISSN 0377-2217. doi: 10.1016/j.ejor.2012.05.028.

M. Bögl, K. F. Doerner, and S. N. Parragh. The School Bus Routing and Scheduling Problem with Transfers. *Networks*, 65(2):180–203, 2015. ISSN 1097-0037. doi: 10.1002/net.21589.

R. Buliung, K. Soltys, C. Habel, and R. Lanyon. Driving Factors Behind Successful Carpool Formation and Use. *Transportation Research Record: Journal of the Transportation Research Board*, 2118:31–38, 2009. doi: 10.3141/2118-05.

G. Cich, I. Ben-Arroyo Hartman, L. Knapen, and D. Janssens. A Simulation Study of Commuting Alternatives for Day Care Centres. *Future Generation Computer Systems*, 2018. ISSN 0167-739X. doi: 10.1016/j.future.2018.05.009.

Committee for Education and Equal Rights and Committee for Mobility and Public Works. Knelpuntnota. `http://issuu.com/kinderrechten/docs/8797dd86-fdca-4ced-83c8-61a9503c5228/16?e=6593254/5429626`, 2013-2014.

K. Declercq, D. Janssens, and G. Wets. Flemish Household Travel Survey 5.1 (2015–2016) Table Report. Technical report, UHasselt - Hasselt University, IMOB, Hasselt, Belgium, December 2016. (in Dutch).

M. Dillen. Draft Resolution on Improving the Transportation of Students in Special Education. `https://docs.vlaamsparlement.be/docs/stukken/2013-2014/g2402-1.pdf`, 2014. (in Dutch).

Federal Government. Best Mobility Practices for the Residential Work at Companies and Government Institutions: Report based on Residential Work Diagnostics 2008. `https://mobilit.belgium.be/sites/default/files/downloads/diagnostique_best_practices_nl.pdf`, 2008.

Flemish Centre for Volunteering. Kilometre fees. `http://www.vlaanderenvrijwilligt.be/wetgeving/kosten-en-vergoedingen/kilometervergoedingen/`, 2017. Accessed: 2017-11-29 (in Dutch).

Flemish Government and Traject. Research into Obstacles and Conditions for the Successful Promotion of Carpooling. `http://www.mobielvlaanderen.be/studies/carpoolen02.php?a=18`, 2006.

Google. Google Maps. `https://www.google.be/maps/`, 2017. Accessed: 2017-11-07.

GraphHopper. The GraphHopper Directions API Route Planning For Your Application. `https://www.graphhopper.com/`, 2017. Accessed: 2017-10-16.

J. Li, P. Embry, S. P. Mattingly, K. F. Sadabadi, I. Rasmidatta, and M. W. Burris. Who Chooses to Carpool and Why? Examination of Texas Carpoolers. *Transportation Research Record: Journal of the Transportation Research Board*, 2021:110–117, 2007. doi: 10.3141/2021-13.

A. Neven, K. Braekers, K. Declercq, G. Wets, D. Janssens, and T. Bellemans. Assessing the Impact of Different Policy Decisions on the Resource Requirements of a Demand Responsive Transport System for Persons with Disabilities. *Transport Policy*, 44:48–57, 2015. ISSN 0967-070X. doi: 10.1016/j.tranpol.2015.06.011.

OptaPlanner. What Is OptaPlanner? `https://www.optaplanner.org/`, 2017. Accessed: 2017-10-16.

P. R. Oxley and M. J. Richards. Disability and Transport: A Review of the Personal Costs of Disability in Relation to Transport. *Transport Policy*, 2(1):57–65, 1995. ISSN 0967-070X. doi: 10.1016/0967-070X(95)93247-V.

J. Park and B. Kim. The School Bus Routing Problem: A review. *European Journal of Operational Research*, 202(2):311–319, 2010. ISSN 0377-2217. doi: 10.1016/j.ejor.2009.05.017.

Regional Transportation Authority Chicago. RTA Trip Planner, 2017. `http://www.rtachicago.org/plan-your-trip` Accessed Jul-11.

A. Rizzoli, R. Montemanni, C. Serra, F. Minori, D. Kehagias, E. Silk, M. Bulgheroni, and C. Bonizzi. SocialCar: Open Social Transport Network for Urban Approach to Carpooling. EU H2020 Project No 636427 Deliverable Deliverable D2.1, SUPSI, EURECAT, MOVENDA, CERTH, LIFTSHARE, Ab.ACUS, January 2016.

K. Roberts and D. Lawton. Financial assistance for families with severely disabled children and transport costs. *Children & Society*, 13(5):333–345, 1999. ISSN 1099-0860. doi: 10.1111/j.1099-0860.1999.tb00130.x.

Taxistop. Schoolpool. `https://www.schoolpool.be/nl`, 2017. Accessed: 2017-11-07 (in Dutch).

F.-J. Van Audenhove, O. Korniichuk, L. Dauby, and J. Pourbaix. The Future of Urban Mobility 2.0: Imperatives to Shape Extended Mobility Ecosystems of Tomorrow. Technical report, Arthur D. Little, January 2014.

T. Vanoutrive, E. Van De Vijver, L. Van Malderen, B. Jourquin, I. Thomas, A. Verhetsel, and F. Witlox. What Determines Carpooling to Workplaces in Belgium: Location, Organisation, or Promotion? *Journal of Transport Geography*, 22(Supplement C):77–86, 2012. ISSN 0966-6923. doi: https://doi.org/10.1016/j.jtrangeo.2011.11.006. Special Section on Rail Transit Systems and High Speed Rail.

Part III

Data Collections by Prompted Recall

Chapter 9

Threshold settings for TRIP/STOP Detection in GPS Traces

The work presented in this chapter was published on March 7, 2016 in the Journal of Ambient Intelligence and Humanized Computing [Cich et al., 2016]. It was an extended version of the peer-reviewed work [Cich et al., 2015] presented at the 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015) in London, United Kingdom held on June 2-5, 2015.

Keywords: GPS Recording; Trace Segmentation; Stop Detection; Sensitivity Analysis; Trace Generation

Abstract

This paper presents two methods to extract stops and trips from GPS traces: the first one focuses on periods of non-movement (stops) and the second one tries to identify the longest periods of movement (trips). A stop corresponds to a location where the individual halts with the intention to perform an activity. In order to assert the quality of both methods, the results are compared to cases where the stops and trips are known by other means. First a set of traces was used for which the stops were identified by the traveller by means of a visual tool aimed at alignment of manually reported periods in the diary to automatically recorded GPS coordinates. Second, a set of synthetic traces was used. Several quality indicators are presented; they have been evaluated using sensitivity analysis in order to determine the optimal values for the detector's configuration settings. Person traces (as opposed to car traces)

were used. Individual specific behaviour seems to have a large effect on the optimal values for threshold settings used in both the trip and stop detector algorithms.

Accurate detection of stops and trips in GPS traces is vital to *prompted recall surveys* because those surveys can extend over several weeks. Inaccurate stop detection requires frequent corrections by the respondent and can cause them to quit.

9.1 Problem Context

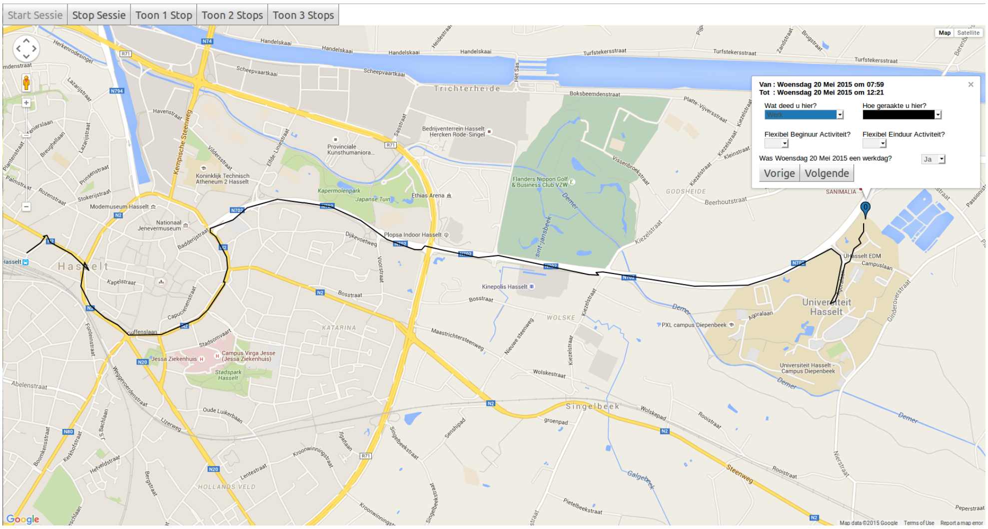
GPS traces are widely used as a data source in mobility science, e.g. researchers extract information about *stops* (locations where people reside for some time in order to perform an intended activity) and *trips* or *moves* linking those stops in time and space. The required accuracy of the results depends on the purpose of the research. This paper focuses on the problem to determine stops from person traces (containing mixed sequences of walking, public transport and car movements, etc.) that will be used in a *prompted recall* survey to capture the traveller's intention. The prompted recall concept is explained in [Stopher et al., 2010].

Following concepts are distinguished:

- **activity location**: a region in space visited one or more times to perform an activity on purpose. The location is visited intentionally. Those can be identified by spatial clustering and visit frequency analysis.
- **standstill**: a contiguous region in space-time where the speed of the monitored individual is near to zero. A standstill corresponds to a single particular visit of a location. Its identification can be supported by using equations governing movement (e.g using Kalman filtering).
- **stop**: a *standstill* at an *activity location* with the purpose to perform an intended activity. This research focuses on stops.

The aim is to detect *regions in space-time* corresponding to a single *stop* (as opposed to frequently visited activity locations which correspond to *regions in space*). A stop is characterised by an area, a time period and a purpose. The associated purpose is not uncovered by the GPS trace but needs to be determined in a stage that follows the *stop detection*. Periods of non-movement due to congestion or waiting for traffic lights are not considered to correspond to be stops in this context. The challenge is to accurately find stops while making use of coordinates and temporal data only. In a *prompted recall* survey, participants are urged to annotate their stops as soon as possible in order to ensure accurate reporting. Hence, it is not possible to postpone *stop* identification until a sufficiently large trace is available that allows for accurate *standstill* detection that makes use of parameters extracted from the particular trace (e.g. by spatial clustering). Many researchers use space and time thresholds to detect stops and several thresholds values are reported in the literature without argumentation. This paper reports on an experiment used to derive space and time threshold values for a specific stop detector. Thereto, traces were used for which the corresponding stops had been reported by the participants in cleaned diaries. The thresholds are determined for which the stop detector results in the best approximation of the set of reported stops.

GPS traces are collected by means of smartphones owned and carried by the survey participants. Those GPS traces are frequently uploaded to a server on which a stop detection algorithm runs every 3 minutes. The actual prompted recall survey is conducted using a web application. On this web application the stops and trips are presented on a map and the traveller is urged to specify for each stop location the intention why it was visited and by using which transportation mode. Prompted recall surveys often extend over a period of several weeks. Hence, accurate stop detection is crucial in order to avoid missing stop locations or presenting false positives to the respondent. Finally, since the interactive stop location annotation is performed before all data for the complete survey period become available, incremental stop detection is required. The GPS recording device uploads data as they come available and the traveller can decide to annotate new recordings several times a day. A screenshot of our prompted recall tool can be seen in Figure 9.1.



Van : Woensdag 20 Mei 2015 om 07:59
Tot : Woensdag 20 Mei 2015 om 12:21

Wat deed u hier? Werk

Hoe geraakte u hier? [redacted]

Flexibel Beginuur Activiteit?

Flexibel Einduur Activiteit?

Was Woensdag 20 Mei 2015 een werkdag? Ja

FIGURE 9.1: An example of a our prompted recall study tool.

The study presented in this paper compares two tools: (i) a stop detector based on identification of spatial clusters of GPS coordinates and (ii) a trip detector that tries to identify sequences of GPS recordings corresponding to movements. The methods are in some sense complementary and the research aims to support the selection of the method that is best suited to feed the prompted recall survey.

This paper is organised as follows. In Section 9.2, a brief overview of relevant related work is given. Sections 9.3 and 9.4 focus on algorithms to detect stops and trips. In Section 9.5 some important insights regarding the experiments are discussed. Section 9.6 presents some techniques to evaluate our TRIP/STOP detection algorithms. In Section 9.7 these algorithms are compared and our findings are discussed. In Section 9.8 the robustness of our TRIP/STOP detection algorithms is analysed with the help of *synthetic traces*. Finally, Section 9.9 gives a brief summary of the paper.

This paper is an extended version of the work published in [Cich et al., 2015]. Our work is extended with a new sensitivity analysis method. Besides the new method, an algorithm to create synthetic GPS traces is developed in order to be used by the TRIP/STOP detection algorithm.

9.2 Related Work

In the literature, different types of TRIP/STOP detection methods are found. A first class makes use of *car traces*. Those methods are based on monitoring the engine of the car, namely when the car engine is turned on, the GPS recording starts and when the car engine is turned off, the GPS recording stops. Schönfelder et al. [2002] use such method in order to replace paper based travel data surveying by automatic data collection via GPS recordings. Automatic data collection is more convenient to capture longitudinal data and reduces the burden for the respondents. Two types of data collection are discussed, namely *passive monitoring* in which only data is collected and no input of the respondent is required and *hybrid approaches* in which besides the automatic data collection, input of the user is also required. The paper describes data capturing using a GPS recorder in a car. With this method, they roughly capture all the trips and stops. Nevertheless they encountered some difficulties, namely when a respondent performs a short stop (for example at the bakery). If the car engine is not turned off, this stop is not initially captured by the device. Another problem is caused by recording failures. In some cases the GPS signal was lost or zero speed measurements were reported. The authors suggest post processing of standstills such as calculating the time the vehicle does not move. If the car does not move for less than 120 seconds, the standstill is not considered as a trip end. In the other case, the standstill will be seen as a trip end.

Du and Aultman-Hall [2007] propose a trip identification algorithm which distinguishes between identification during a GPS signal loss period and identification in normal situations. They assume that the driving pattern of a vehicle did not change during signal loss, hence they estimate the time a vehicle should have driven to reach the next captured point. To estimate this time, they take into account a number of points before and after the signal loss. If the estimated time equals the real time within a given threshold, the signal loss period should be part of a trip. In the other case, if the difference between the real time and the estimated time is larger than a given threshold, the signal loss is assumed to induce a trip end. When there is no signal loss, the dwell time is used. The authors use a minimum and a maximum dwell time. For the minimum dwell time, they used 20 sec, 30 sec and 40 sec; for the maximum dwell time, they used 60 sec, 100 sec and 140 sec. Hence, when the dwell time was more than the maximum dwell time, the authors assume that there was a stop; in the other case, when the dwell time was between minimum and maximum dwell time, further calculations need to be done. They propose two mechanisms, one uses the heading change of the vehicle, while the other takes into account the distance to the real vehicle roads (network link).

Guidotti et al. [2015] discuss (activity) location detection based on GPS coordinates. This differs from the problem to be solved in the *prompted recall* context. It does not make use of time information. The authors seem to consider GPS records with near zero speed only. Several parameter-free and parameterised methods are discussed and the problem of (lack of) robustness of parameterised methods is mentioned as well as the consequent problem of inability to derive parameter values when the *base truth* is unknown. Internal (to be used when no base truth is known) and external evaluation methods are discussed. A two-step method (named TOSCA) consisting of X-means clustering and Single-Linkage (SL) clustering based on the minimum distance between members in clusters is presented. The SL clustering proceeds by first combining the two clusters having the minimal distance (dendrogram). The distribution of those inter-cluster distances is considered and is observed to typically show a peak value. SL clustering stops as soon as the next distance value in the monotonically increasing sequence can be considered to be an outlier (according to the mentioned distribution). Three techniques for outlier detection are used. Finally, evaluation of the TOSCA algorithm was done (i) using synthetic data sets generated using a null model and (ii) using a mobility based model (locations distributed using radius of gyration and location visits distributed according to Zipf's distribution). For the (i) data set TOSCA is equivalent to other methods. For the (ii) data set TOSCA outperforms the other methods.

A second class includes points of interests (POI) to examine trips and stops. Alvares et al. [2007] attempt to answer questions about movements of people, e.g. “Which are the places most frequently visited by people attending a conference in a touristic city?”. When using raw GPS trajectories, complex queries are required to answer such questions. For this reason, the authors try to add semantic information to the raw GPS data, i.e. they attempt to find “stops” and “moves” in the data. Their algorithm uses “interesting zones” to find stops. For example if a person stays for a minimum period of time in the neighbourhood of the “Notre Dame” in Paris, this is flagged as a stop. Otherwise, if he does not stay a minimum of time in this zone, it is a move. Notice that this method requires a set of *interesting zones* and therefore it is application dependent. When conducting research about tourism, touristic POI are required. For example, a traffic light is no interesting zone for tourism but could be in another application context.

Spinsanti et al. [2010] attempt to match stops with POI. For their experiment, they use GPS recorders that are located in a vehicle. When a person stops at a parking and then walks two kilometres to a shop, this method will miss this two kilometre trip. To detect stops, the authors use an algorithm that detects sufficiently long periods during which the vehicle does not move. After finding a stop, they attempt to match this stop to a POI, taking into account the opening times of the POI. For example when a person arrives at a parking nearby a shop and a museum, the algorithm will analyse whether or not the shop and/or the museum are open at that moment in time. Furletti et al. [2013] describe an improvement/extension of this paper.

In the present paper, the aim is to conduct a prompted recall survey. For such research *person traces* are needed, but individual POIs are not known. Hence, a TRIP/STOP detection algorithm is needed that can operate with minimal information. Our work aligns with the work reported by Yan and Spaccapietra [2009]; they attempt to add semantics to raw GPS trajectories. Their algorithm uses five steps, namely “preparation”, “data preprocessing”, “trajectory segmentation”, “stop identification” and “semantic enrichment”. For stop identification, the authors have three methods. The first method is “velocity based”, i.e. they calculate the speed between GPS points and when the speed is low enough, the algorithm assumes it is a stop; this corresponds to the trip detection described in Section 9.4. The second method is “density based” and corresponds to our method mentioned in Section 9.3. This method also considers, besides speed, the distance the object has travelled during a time period. The third method is “time series based” [Yan, 2010]. This method uses the forecast speed to detect stops.

Schüssler and Axhausen [2009] propose an algorithm to process raw GPS data. It starts with a cleaning step in which they attempt to remove systematic errors. The number of used satellites together with the Position Dilution of Precision (PDOP) value is an efficient measure. Unfortunately, this was not available in their research. Hence, they used the altitude and they attempted to remove jumps by measuring the speed between two consecutive points. They introduce a Gauss kernel method to clean/smooth the points as well. For trip identification, they process two situations, one in which no signal was lost and one in which there were signal losses. If there are ongoing GPS recordings, the authors use two methods: one detects an activity when the speed between the points stays very low (0.01 m/sec for $\geq$ 120 sec). Another technique attempts to find bundles of points which are close together. When there is a signal loss, the time between two consecutive points is calculated. When this time reaches 900 sec, they assume an activity has happened. The authors introduce an activity joining mechanism; when two activities happen with $\leq$ 15 GPS points in between, those two activities are joined. For mode detection, the authors use a fuzzy algorithm which is based on the median of the speed distribution and the 95^{the} percentile of the speed and acceleration.

Rasmussen et al. [2015] present a complete method to process raw GPS data automatically. They propose a data cleaning step, a trip identification step and a trip segmentation step which are adopted from Schüssler and Axhausen [2009]. The cleaning step is conducted in order to remove erroneous objects which takes into account the altitude level, number of used satellites and their dispersion. For smoothing the data, the authors use a Gauss kernel smoothing technique. Trips are identified by means of the stationarity of the GPS device. Note that this is not meant as not moving at all. They attempt to identify activities when the GPS (i) does not log any position for a specific amount of time ($\geq$ 120 sec), (ii) has a very limited speed for a specific amount of time (speed $<$ 0.01 m/sec for $\geq$ 60 sec) or (iii) stays within a limited area for a specific amount of time ($\geq$ 60 sec).

After finding activities, they attempt to segment trips into *trip segments*, such segments could be parts of a multi-modal trip. They calculate this by taking into account the characteristics of walking because they assume that for every mode change a walking activity needs to be performed. Besides trip and activity identification, they also describe a mode identification algorithm which consists of several steps and is based on a GIS approach and the speed characteristics of a trip. Mode identification is out of the scope of this paper.

Tsui and Shalaby [2006] discuss a four step model in order to process raw GPS data. They start with a data filtering process in which they remove GPS points which have (i) $<$ three satellites (ii) HDOP value $\geq$ 5 (iii) zero directional heading and zero speed and (iv) jumps (developed by Eui-Hwan and Amer [2005]). The second step is trip

identification. They describe a trip as a connection between two consecutive activities, hence when the activities are found, the trips are found as well. They propose a rule based algorithm for which they use a dwell time of 120 sec. They distinguish two main situations. The first situation is activated when the GPS does not suffer from signal losses. In this case they assume that an activity takes place when there are zero speed measurements for a period of 120 sec. The other case is when the GPS suffers from signal losses. In this case they distinguish three situations: (i) short duration indoor activities (duration between 120 sec-600 sec and distance travelled < 50 m), (ii) long duration indoor activities (duration > 600 sec) and (iii) underground or indoor travelling (duration > 120 sec and distance travelled > 500 m). The third step is mode segment identification for which they assume walking is the intermediate travel mode between any other mode. Finally they attempt mode identification for which they use (i) average speed, (ii) 95^{the} percentile maximum speed, (iii) positive median acceleration and (iv) data quality. They discuss also an extension/improvement by using a GIS system in order to conduct mode identification.
Bothe and Maat [2009] conduct a GPS-based travel survey. Their research consists of two parts, namely a interpretation process and a validation process. In the interpretation process, they first conduct a cleaning step in which unreliable GPS points are deleted. They use distances and times between GPS points and the speeds to conduct this cleaning. They attempt to extract trips by using a dwell time of three minutes. Hence, when a respondent stays for three minutes in a certain area, they assume this respondent did an intentional stop. When the trips and stops are found, they attempt to extract trip purposes which is done by using a GIS in which POIs are stored. When a stop is close to a certain POI, they assume this was the place where the activity took place. Finally they conduct a travel mode prediction in which they attempt to predict the travel mode of the trip. They use the speed of the trip and a GIS to predict this. The validate their data, they use a web application; respondents can go through a day and they can correct the predicted data if necessary. They can adjust trip times, travel modes and purposes, merge or split trips and move or add locations.

Other researches aiming at finding similarities between users according to their location histories need a stop detection algorithm as well. Li et al. [2008] and Xiao et al. [2014] both use the same approach to find (in their terminology) *stay points*, which is very similar to the *stop location* mentioned in Section 9.3.1. A stay point is a region bounded by a distance threshold in which a user stays for a minimum amount of time. Multiple GPS points which are captured within a region are averaged in order to keep one GPS point representing the stay point. Besides the coordinate of the stay points, the arrival and leaving time are calculated as well. For their experiment they used a time threshold of 30 min and distance threshold of 200 m. These stay points are then used to calculate the location history, which are used to measure the similarity between users. Ma et al. [2012] attempt to find social relationships between users by using their trajectories. Their algorithm consists of five steps, namely "trajectory parsing", "user entropy computation", "area entropy computation", "co-location record extraction" and "social relationship measurement". The trajectory parsing is done by calculating stay points, which is similar as discussed earlier.

In Section 9.3, an implementation of a stop detection algorithm is discussed in which an attempt to find stop clusters is made. Stop clusters are GPS points which are close together in both time and space. In Section 9.4 a trip detection algorithm which finds stops by using the speeds between the GPS points is discussed.

9.3 Stop Detection

The purpose of the work described in this paper is to detect *stops* (i.e. a stay at a given location in between two consecutive movements (trips)). The aim is to detect *regions in space-time* (as opposed to *regions in space*). A stop is characterised by an area and a time period. Since time is involved, processing GPS data chronologically is the natural way of working. Furthermore, the GPS traces are delivered as a chronologically ordered sequence of recordings. Other tools detect *stop locations* (regions in space) where people reside during multiple disjoint periods and spend a lot of their time (in which case spatial clustering can be performed without taking timing into account).

A stop detector processing GPS records in two passes is presented. The algorithm inspects the last few recordings received and performs spatial clustering. The first pass attempts to detect stops and trips by classifying every GPS record either as a stop member or a trip member and by assigning it the number of the stop or trip it belongs to. This can be accomplished by identifying clusters of points that are sufficiently close to each other. Points are represented as a triple $\langle x, y, t \rangle$ with x the longitude coordinate, y the latitude coordinate and t the timestamp. The second pass conducts cleaning of the stops and trips and it replaces the set of points constituting a stop by a single representative point for which the attributes $\langle x, y, t \rangle$ are averaged over the cluster members.

9.3.1 Terminology

StopDurationThreshold denotes *the minimum duration for a period* a person needs to stay in a specific area in order to qualify the area as a stop location.

StopDistanceThreshold denotes the *size of the area* where a person needs to stay in order to qualify the area as a stop location.

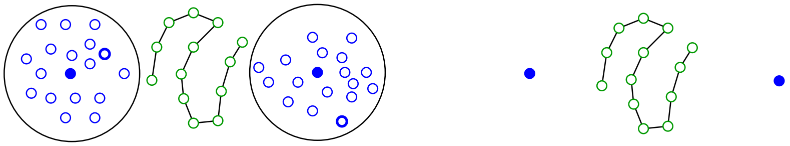
A *stop cluster* is a circular area with a radius of *stopDistanceThreshold*. Every stop cluster has a *cluster centre*, i.e. the centre point of the circle that represents the stop cluster.

Last known stop point is the last point of a stop cluster, i.e. the point with the most recent timestamp in the stop cluster.

A *Stop* location (abbreviated to *Stop*) consists of a specific set of GPS points represented by a single $\langle x, y, t \rangle$ triple together with the *stopDistanceThreshold* radius and the *stopDurationThreshold* period.

A *Trip* consists of a specific number of GPS points. Notice that the sets of GPS points corresponding to trips and stops respectively are mutually disjoint. Each recorded GPS point belongs either (i) to a trip or (ii) to a stop or (iii) is discarded as an outlier.

In Figure 9.2a, a conceptual view explaining the terminology is shown. The larger circles represent stop clusters along with their associated member points. The filled point in the stop cluster is the *cluster centre* and the point having a thick line perimeter is the *last known stop point*. The points that are connected with a line are part of a trip.



(A) Stop detection with two stop clusters and one trip.

(B) Stop detection with combined stop clusters.

FIGURE 9.2: An example of stop detection and the corresponding combination of several points into one representative point.

9.3.2 Algorithm: First Pass

In Algorithm 1, the first pass of the algorithm is presented as pseudo-code.

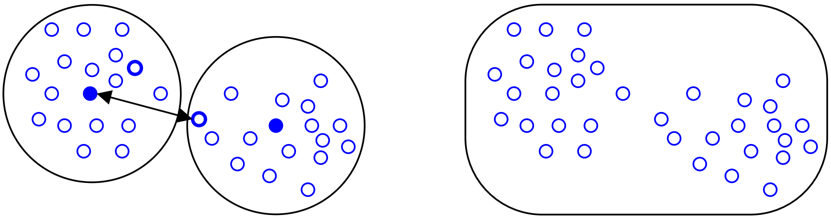
The input to the stop detection algorithm consists of a GPS trace for a given person. Such trace is a chronologically ordered sequence of $\langle x, y, t \rangle$ triples. The initial step is to label the first GPS record as the centre of a provisional cluster, this can be noticed in Line 1. From this point the algorithm continues by sequentially processing every GPS point fetched from the input, shown in Line 4.

Line 5 analyses whether or not *gpsRecord* spatially belongs to the current stop cluster; if so, the *gpsRecord* is labelled as a potential stop point. If not, it is certain that the current calculated stop cluster cannot be extended, but it is not known yet whether or not it is a valid stop cluster. The test in Line 8 analyses the duration of the current stop cluster. The function `duration(Set of gpsRecs)` returns the time difference between the chronologically last and first entries in the set. If the duration of the current stop cluster is less than the *stopDurationThreshold*, all potential stop points are labelled as trip members, shown in Line 18. If the condition in Line 8 holds, it is certain that the current stop cluster is a valid stop. At this point in the algorithm, it is known that a trip, if any was constructed, is complete. If there are potential trip points, it has to be analysed whether or not the trip is a valid one (Line 10). Not every sequence of GPS recordings does represent a trip; therefore, the newly added sequence consisting of all $P \in potTripPoints$ is evaluated to find out whether or not it meets the criteria to constitute a (partial) trip. In a first experiment, following straightforward conditions were used: (i) the duration of the trip shall be strictly larger than zero, (ii) the average speed shall be sufficiently high (2.5 km/h) and (iii) the number of GPS points in the sequence shall be not less than a specified minimum (10). Using only those conditions, "*ghost*" trips were observed that arise from point sequences that seem to contain measurement errors. It is assumed that those sequences occur when a person was in a building or tunnel or when the measurements got disturbed by the *urban canyon* effect. GPS errors caused many outliers. This problem was solved by considering the theoretical recording frequency of the GPS recording device. For this purpose, the condition specified by Equation 9.1 was used.

$$numberOfTripPoints > \frac{StopDuration}{2 \times TheoreticalRecordingRate} \tag{9.1}$$

The right hand side in Equation 9.1 estimates the number of points that should have been recorded for the duration of the trip. The factor two in the denominator means that at most half of the expected recordings are allowed to be dropped due to device problems.

When all previously stated conditions are satisfied, all $P \in potTripPoints$ are written to a file, otherwise they are discarded. In Line 13 the *potTripPoints* are cleared because the trip points are handled in the previous step. After handling the trip, the current stop have to be written to the output file. Remember that this was a valid stop, because the GPS points in the *potStopPoints* remained in the stop cluster for a period of *StopDurationThreshold*. At this point, when the algorithm detects two consecutive stops, it will analyse the distance between the two stops in both time and space. Spatial and chronological distance are calculated between the last point found in the first cluster and the centre of the second cluster. If the distance is sufficiently small, both clusters are merged into a single one. This scenario is illustrated in Figure 9.3.



(A) Two stop clusters separated by a small distance. (B) A merge of the two stop clusters shown in Figure 9.3a.

FIGURE 9.3: An example of the used merge process.

Finally, the *gpsRecord* being processed becomes the centre for an new provisional cluster; it is also added to the *potStopPoints* (Line 21 and 22).

9.3.3 Algorithm: Second Pass

The second pass consumes the output of the first pass. First, all points belonging to the same cluster are replaced by a single representative point. The result can be seen in Figure 9.2b. After the first pass, every GPS point has either received a *stop id* or a *trip id*. GPS points with the same *stop id* belong to the same stop cluster, while GPS points with the same *trip id* belong to the same trip. To calculate the representative point of the stop cluster, the algorithm goes through every GPS point. The longitude and latitude coordinate of GPS points with the same *stop id* are averaged and hence, one representative GPS point will remain.

The second task consists of cleaning both *trips* and *stops*. For the trip cleaning the following steps are executed: for all but the last point in a trip, the average speed between the point and its successor are calculated (speed is calculated from exactly two GPS recordings). If the average speed between consecutive points A and B is too high, point B is deleted. For the average speed 150 km/h was used in our experiments. Hence, GPS recording problems can be solved where a GPS point B appears at a distance of thousands of kilometres from the chronological predecessor

Algorithm 1 Stop Detection Algorithm

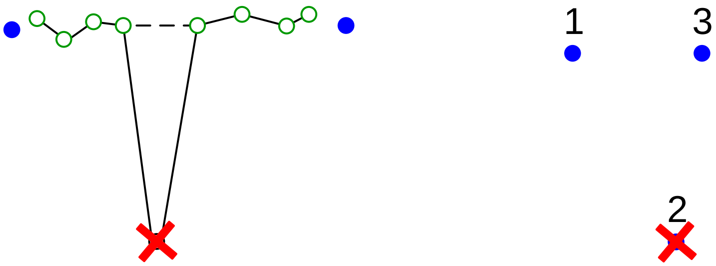
```
$clusterCenter \leftarrow$ first GPS record
$potStopPoints \leftarrow \emptyset$ {// Potential stop points}
$potTripPoints \leftarrow \emptyset$ {// Potential trip points}
for all $gpsRecord$ in input source do
  if distance($gpsRecord$, $clusterCenter$) $<$ $stopDistance$ then
    $potStopPoints \leftarrow potStopPoints \cup \{gpsRecord\}$
  else
    if duration($potStopPoints$) $>$ $stopDuration$ then
      {// Sufficient duration to create stop location}
      if isTripValid($potTripPoints$) then
        writeToFile($potTripPoints$)
      end if
      empty $potTripPoints$
      checkToMergeClusters()
      writeToFile($potStopPoints$)
    else
      {// Duration at location too short}
      $potTripPoints \leftarrow potTripPoints \cup potStopPoints$
    end if
    $potStopPoints \leftarrow \emptyset$
    $clusterCenter \leftarrow gpsRecord$
    $potStopPoints \leftarrow potStopPoints \cup \{gpsRecord\}$
  end if
end for
```

A while the successor of B is again near to A. An illustration of such a situation can be found in Figure 9.4a.

The *stop cleaning* part is similar to *trip cleaning*. All stops are processed (i.e. the stop cluster representatives) and when the average speed between stop A and stop B is too high, point B is deleted, as shown in Figure 9.4b. Again, problems with outlier GPS points can be solved.

9.3.4 Algorithm: Complexity

Every step described in the previous Sections consumes the GPS points and because only two consecutive points are compared, only one pass per step is needed. Hence the complexity of this algorithm is $\mathcal{O}(n)$ where n is the size of the input (GPS points). In order to give the reader an idea: a file consisting of 110k GPS points needs 1.5 sec to be processed.



(A) An example of trip cleaning. (B) An example of stop cleaning.

FIGURE 9.4: An example of trip and stop cleaning

9.4 Trip Detection

In trip detection, the longest possible subsequence for which (i) the moved distance is sufficiently large (similar condition as for stop detection), (ii) the speed values start from near zero, grow and drop again to near zero, and (iii) which does not have large gaps caused by missed recordings is considered to constitute a trip. Sequences having almost zero speed are considered to be *stops* and the remainder of the recordings is considered to be junk.

The trip detector scans the GPS records and maintains a variable size sliding window containing the last records seen. Those records have not yet been finally qualified as *stop*, *trip* or *junk*. Each time a record is read, several quantities are evaluated: instantaneous and smoothed speed and acceleration, window size and period, etc. Specific changes in the evaluated quantities lead to event firing. The events are fed to the finite state machine (FSM) shown in Figure 9.5 that controls the qualification of the subsequence contained in the window. Definitely qualified records are dropped from the window. The speed condition is required because part of the traces to be processed by the trip detector were recorded using devices that can be turned on/off by the respondents while driving. Plausible evolution of speed and lack of large gaps caused by missed recordings were required to assure the extraction of complete trips (as opposed to junk parts).

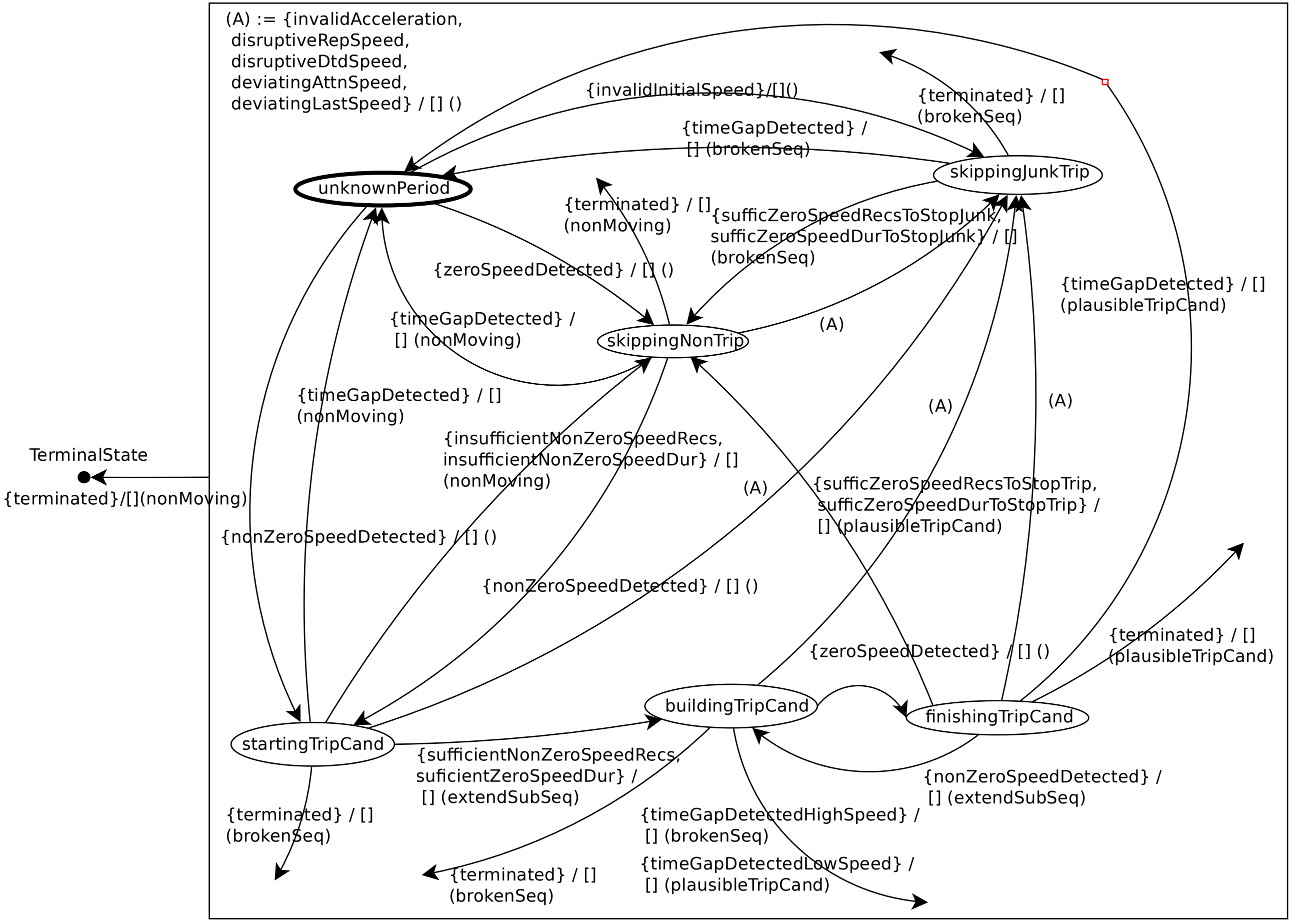


FIGURE 9.5: FSM controlling the trip detector software while scanning a stream of GPS recordings. Arrows not ending in an oval leave the composite state and point to the *TerminalState*.

TABLE 9.1: Trip detector finite state machine.

State	Description
unknownPeriod	The initial state
skippingNonTrip	Consuming data records in a period identified as a *stop*
startingTripCand	Consuming records that are the first ones in a sequence of non-zero speed records
buildingTripCand	Consuming records belonging to a started trip candidate
finishingTripCand	Consuming records that are the first ones in a sequence of zero-speed records
skippingJunkTrip	Consuming records that were classified as garbage, looking for sequence of zero-speed records

The states of the FSM are defined in Table 9.1. A state corresponds to a provisional qualification of the GPS records contained in the window being processed i.e. the tail of the sequence of GPS records inspected but not yet finally qualified. Such records cannot be qualified individually: the qualification collectively applies to all records in the window. As a consequence, the state transitions correspond to the decision to qualify the content of the window differently.

In Figure 9.5 the transitions are labelled using the notation `{E}/[C](A)` where $\{E\}$ denotes the set of events that can trigger the transition, $[C]$ is the set of conditions required to enable the transition and (A) is the action executed which is represented here by the label attached to the records collected in the window.

The speed values reported in the records (RepSpeed) are used to compute the attenuated speed (AtnSpeed) using a *moving average* technique operating on a given time interval $[t_0 - \Delta t, t_0]$ where t_0 is the time associated with the record being processed and Δt is a given parameter (typically the time required to accelerate to regime speed or to decelerate back to zero speed: 5 sec was used). The number of recordings in the interval is not fixed since recordings are allowed to be missing and due to the variance on the recording period length. As a consequence, the weights need to be recalculated each time a new recording is submitted because each recording's weight depends on the time difference between that recording and the new most recent one. The value of the weight at the beginning of the interval is given by a parameter $W_{min} = 0.05$. The weight function is given by

$$w(t_0 - dt) = e^{-\alpha \cdot dt} \tag{9.2}$$

$$dt = \Delta t \Rightarrow w(t_0 - dt) = A_{min}$$

$$\Rightarrow \alpha = \frac{-log(A_{min})}{\Delta t} \tag{9.3}$$

The attenuated speed is given by

$$v(t_0) = \frac{\sum\limits_{t_k \geq t_0 - \Delta t} v(t_k) \cdot w(t_k)}{\sum\limits_{t_k \geq t_0 - \Delta t} w(t_k)} \tag{9.4}$$

The acceleration is computed from the attenuated speed values by $\frac{v(t_0 - t_1)}{t_0 - t_1}$. If the acceleration (deceleration) exceeds a given threshold the data is discarded based on the assumption that the recording device was turned on/off while driving.
Speed smoothing is essential when speeds are computed using the most recent recordings at $t_1 < t_0$ respectively by dividing the distance driven in the interval $[t_1, t_0]$ by the length of that interval. If ϵ_s is the error on the position, the error on the speed can be as large as $\frac{2 \cdot \epsilon_s}{t_0 - t_1}$ which is problematic when high frequency recording (order of 1 Hz) is used.

Event firing is controlled by thresholds settings loaded from a configuration file. Two of them are used in sensitivity analysis: (i) the minimum period during which speed is below the *almostZeroSpeed* to fire *sufficZeroSpeedDurToStopTrip* and (ii) *bDistMinFirstAny* the minimal bird's eye view between the first point and at least one other point in a sequence to qualify the sequence as a trip component.

The meaning of most events is clear from their name and from the transition they trigger (which can be found in the state transition diagram). The events collectively labelled with *(A)* require additional explanation. Speed value names are:

v^{rep}	speed reported by the recording device
v^{dtd}	speed calculated from last two records using $\frac{\Delta s}{\Delta t}$
v^{att}	attenuated speed computed by v^{dtd} smoothing

Time t_0 refers to the last recording and t_1 refers to its predecessor. The threshold value is denoted by $\overline{\Delta}$. The events are defined by:

- **invalidAcceleration**: excessive acceleration or deceleration
- **disruptiveRepSpeed**: $|v^{rep}(t_0) - v^{rep}(t_1)| > \overline{\Delta}$
- **disruptiveDtdSpeed**: $|v^{dtd}(t_0) - v^{dtd}(t_1)| > \overline{\Delta}$
- **deviatingAttnSpeed**: $|v^{att}(t_0) - v^{att}(t_1)| > \overline{\Delta}$
- **deviatingLastSpeed**: $|v^{rep}(t_0) - v^{dtd}(t_0)| > \overline{\Delta}$

9.5 Preliminary Note on Results Comparison

9.5.1 Difference between the Stop and Trip Detectors

Both the trip and stop detectors partition the ordered set of GPS recordings constituting a trace so that each part (cell) consists of consecutive recordings that are either unclassified (junk) or belong to the same particular trip/stop and so that all recordings for a particular trip/stop belong to exactly one part. Since the trace is an ordered set, the parts in the partition can be ordered using the recording timestamps. In the trip detector result, no two consecutive parts have the same type (junk, trip, stop). In the stop detector result, two consecutive parts can be stops; this occurs when a part of the trace is missing (teleporting occurs).

9.5.2 Parameters Range Used in the Analysis

The threshold settings for the stop detector mentioned in Section 9.3 and two equivalent settings for the trip detector described in Section 9.4 were given 10 values each (resulting in 100 runs for each respondent by each detector). The following sets of values were used:

$$\mathit{StopDurationThreshold[sec]} =$$

$$\{60, 90, 120, 150, 180, 210, 240, 300, 360, 420\}$$

$$\mathit{StopDistanceThreshold[m]} =$$

$$\{25, 50, 75, 100, 125, 150, 200, 250, 300, 400\}$$

In Section 9.6, the reference data together with the different quality indicators will be discussed. Section 9.7 discusses the results of the experiments.

9.6 Sensitivity Analysis

9.6.1 Reference Data

In order to find optimal threshold settings for stop detection, traces are used for which the corresponding diaries (timed sequences of activities) are available. The set of *person* (as opposed to *car*) traces was captured using smartphones during periods ranging from three to six weeks. The surveyed people participated in a pilot project evaluating electric vehicles. The smartphone was used for GPS recording and for interactive diary data entry: this involves recording information of activities conducted at locations where resided, travel mode used, etc.

A well known phenomenon of those diaries is that they are filled in by the respondents in a very inaccurate way. Examples of mistakes people make are (i) filling in diaries ahead and not correcting these when something changes with respect to the planning, or (ii) filling in diaries only after some days which means that respondents do not remember everything they did.

As a result, the diaries and the corresponding GPS traces can be mutually inconsistent. Two researchers cleaned the diaries by shifting reported trips and activities in time so that they match the GPS traces. The diaries have been aligned with the GPS traces by means of a dedicated interactive tool. Period start and end times were adapted such that there was no movement during periods labelled as *home*, *work*, *shopping*, etc. For more information about these techniques, the reader is referred to [Raza et al., 2015]. The resulting cleaned diaries were used as *base truth* in the sensitivity analysis.

9.6.2 Quality Indicators

Several quality indicators have been computed: those are quality assessment quantities applying to the entire trace (which is a sequence of recordings reflecting a sequence of trips and stops). The indicators allow to compare the sets of automatically extracted trips and stops to the trips and stops reported by the participants and used as a reference. Each indicator is identified by an acronym and a symbol used in equations. The overview is found in Table 9.2.

The *avgDistInd*, the *avgDurInd* and the *avgSpeedInd* are used for plausibility checking (e.g. to evaluate trip length under-estimation in case the correct number of trips is found); the *nTripsInd* and *tempInd* are significant to assert the quality in the context of prompted recall surveys where false positives and false negatives during stop detection should be avoided. Other indicators focusing on the time-shift between reported and recorded trips and their difference in duration have been considered too. Those are only useful if the number of reported trips equals the number of recorded trips (which does not frequently occur).

In [Cich et al., 2015] a *ranked squared error* method to perform a sensitivity analysis was used. For each parameter combination (i) the deviation between the reported and calculated number of trips I_{num} and (ii) the temporal trip matching indicator I_{temp} were determined. Error values for (i) and (ii) were squared and summed over the set of traces. For both indicators, the parameter combinations were ranked based on increasing error. A combined ranking for each combination was determined. Since ranking is based on the error values but ignores their proportions, the

TABLE 9.2: Quality indicators.

Identifier	Symbol	Description
nTripsInd	I_{num}	the *number of trips* found in the trace.
tempInd	I_{temp}	a *temporal trip matching indicator* based on the time intervals corresponding to the reported and the detected trips respectively; the indicator is the ratio of the duration of the intervals intersection period to the intervals union period. The ratio equals one if and only if the total trip times are equal and all intervals pairwise exactly overlap.
avgDistInd	I_{dist}	the *average trip distance* which is the accumulated euclidean distance between coordinates for consecutive recordings associated with the trip.
avgDurInd	I_{dur}	the *average trip duration* which is the difference between the timestamps of the last and the first recordings associated with the trip.
avgSpeedInd	I_{speed}	the overall *average trip speed* where the trip speed is computed by dividing the trip length by the trip duration.

method was not efficient. Furthermore, not all available indicators were taken into account.

Therefore, the use of *Least Weighted Mean Squared Error (LWMSE)* is proposed in order to find suitable *dur-dist-combinations*. Hence, the indicators *avgDurInd*, *avgDistInd*, *nTripsInd* and *tempInd* are used to calculate the *Mean Squared Error (MSE)*. For each *dur-dist-combination* p, the MSE is computed using Equation (9.5) for each indicator $i \in I = \{I_{num}, I_{temp}, I_{dist}, I_{dur}\}$. Variable N_U stands for the number of respondents whose data are included in the sensitivity analysis, $I_{u,i}$ is the *base truth* for a respondent u for indicator i and $I^*_{u,i,p}$ is the estimated indicator i found by the TRIP/STOP detector for a respondent u with *dur-dist-combination* p.

$$MSE_{i,p} = \frac{1}{N_U} \sum_{u=1}^{N_U} \left(\frac{I_{u,i} - I^*_{u,i,p}}{I_{u,i}} \right)^2 \tag{9.5}$$

The use of Equation (9.5) results in the set of following indicators: (i) MSE_{num} based on the *number* of trips found, (ii) MSE_{temp} based on the *temporal trip matching indicators*, (iii) MSE_{dist} based on the *average distance* of trips found and (iv) MSE_{dur} based on the *average duration* of trips found. The MSE values are assigned a weight, specified in the Table 9.3. As discussed above, it is assumed that the variables *avgDistInd*, *avgDurInd* and *avgSpeedInd* can be used for plausibility testing, while *nTripsInd* and *tempInd* are significant to assert the quality of the results in the context of a prompted recall study. Therefore, it was decided to give MSE_{num} and MSE_{temp} a weight of one and MSE_{dist} and MSE_{dur} a weight of 0.5. These weights were given in an attempt to achieve a good balance between the significant and plausible variables. The *Weighted Mean Squared Error (WMSE)* is given by Equation (9.6).

$$WMSE_p = \sum_{i=1}^{N_I} w_i \cdot MSE_{i,p} \tag{9.6}$$

Finally, the *dur-dist-combination p* resulting in the minimum WMSE are of interest; this is called the *Least Weighted Mean Squared Error (LWMSE)* and given by Equation (9.7).

$$LWMSE = \min_p (WMSE_p) \tag{9.7}$$

TABLE 9.3: An overview of the weights given to the different indicators to calculate the Least Weighted Mean Squared Error.

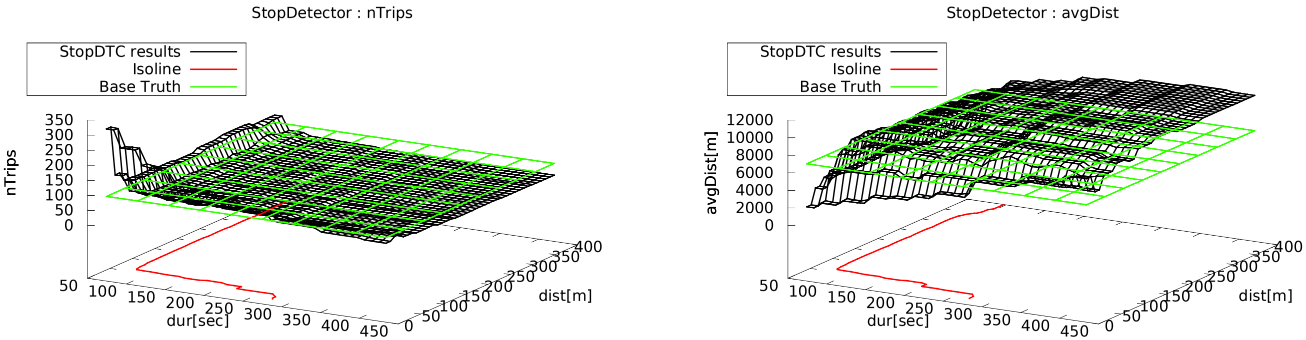
Symbol	**Weight**
MSE_{num}	1.0
MSE_{temp}	1.0
MSE_{dist}	0.5
MSE_{dur}	0.5

9.7 TRIP/STOP Detection Comparison

9.7.1 Sensitivity Analysis Results

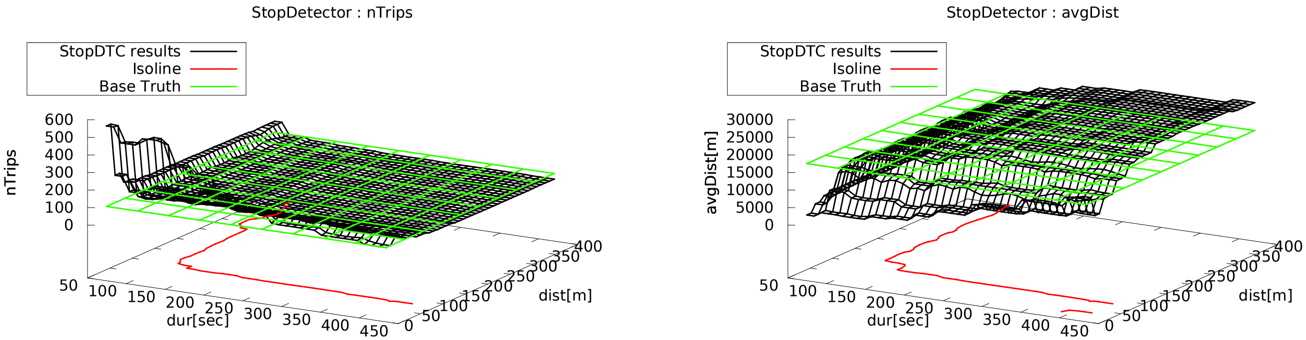
Sensitivity to threshold settings was analysed in two ways: graphs and LWMSE (discussed in Section 9.6) have been compared.

For a representative set of respondents, the 100 indicator values have been plotted in a 3D diagram using a 10 × 10 grid. Examples are shown in Figure 9.6. The horizontal plane drawn as a grid (green) represents the *true value* extracted from the interactively aligned diary. The intersection of the surface representing the detector results (black) for the given thresholds with the *truth plane* has been projected on the horizontal lower plane (red). This projection is the geometric place consisting of pairs of threshold settings that deliver the value reported by the respondent.



(A) Results (nTrips) of user 103.

(B) Results (avgDist) of user 103.



(C) Results (nTrips) of user 129.

(D) Results (avgDist) of user 129.

FIGURE 9.6: Sensitivity analysis indicators: the number of trips and the average trip distance for two users. The horizontal green rectangular plane grid represents the value reported by the respondent. The red line is the projection of the intersection of this plane with the detector's output (black).

When error values below a given threshold are allowed, the geometric place consists of a (not necessarily connected) region.

The iso-lines were compared (i.e. the *geometric place* projection mentioned above) in an attempt to identify similar patterns.

In around 75 % of the cases, roughly similar iso-line patterns were found. For the *average distance* determined by the stop detector, a typical L-shaped pattern was found as shown in Figure 9.6. In a minority of cases other iso-line shapes occurred; in some cases the distance was always over- or under-estimated and hence, no iso-line was discovered. There seems not to be a single point belonging to each of the geometric places. An attempt was made to identify a small region that is crossed by all geometric places (iso-lines) for the *avgDistInd*, *avgDurInd* and *nTripsInd*. In the majority of cases, an intersection was found in the following intervals:

duration[sec]: [100, 200] and distance[m]: [50, 100]

The visual representation of the different parameters gives us a good initial idea about suitable threshold settings. However a quantitative method is required to compare the results of the TRIP/STOP detector with the *base truth*. Hence, the WMSE as given in Equation (9.6) in Section 9.6 was calculated. The five best results of the stop detector and the trip detector are shown in Table 9.4 and Table 9.5 respectively.

TABLE 9.4: The five best results based on the WMSE for the stop detector. The different components of the WMSE and the original temporal indicator are reported. This table is sorted on the WMSE column in increasing order.

Dur[sec]	**Dist[m]**	$\mathbf{I}_{temp}$	$\mathbf{MSE}_{num}$	$\mathbf{MSE}_{temp}$	$\mathbf{MSE}_{dist}$	$\mathbf{MSE}_{dur}$	***WMSE***
180	100	0.6320	0.1225	0.1541	0.0399	0.1562	0.3747
240	100	0.6184	0.1220	0.1650	0.0415	0.1409	0.3782
240	125	0.6265	0.1120	0.1590	0.0425	0.1956	0.3901
210	100	0.6262	0.1384	0.1586	0.0426	0.1461	0.3913
180	125	0.6316	0.1147	0.1542	0.0381	0.2113	0.3936

Based on the WMSE of Table 9.4 and Table 9.5, it is clear that the stop detector gives better results than the trip detector. On average, a tempInd of 63 % can be observed in Table 9.4. At a first glance, this does not seem to be ideal but careful interpretation is required. This result might be due to the average length of the trips. When calculating the overlap of several trip intervals, it is possible that the trip periods identified by the TRIP/STOP detector are shifted relative to the *base truth* trips. Such situation is illustrated in Figure 9.7. The horizontal bars represent the trips. The green bars show high overlap situations. The red bars show situations

TABLE 9.5: The five best results based on the WMSE for the trip detector. The different components of the WMSE and the original temporal indicator are reported. This table is sorted on the WMSE column in increasing order.

Dur[sec]	**Dist[m]**	$\mathbf{I}_{temp}$	$\mathbf{MSE}_{num}$	$\mathbf{MSE}_{temp}$	$\mathbf{MSE}_{dist}$	$\mathbf{MSE}_{dur}$	***WMSE***
120	100	0.4711	0.1033	0.3049	0.0402	0.0979	0.4773
120	125	0.4712	0.1034	0.3049	0.0401	0.0980	0.4774
120	75	0.4712	0.1034	0.3049	0.0403	0.0979	0.4774
120	25	0.4701	0.1041	0.3052	0.0403	0.0966	0.4777
120	50	0.4710	0.1038	0.3050	0.0403	0.0976	0.4778

where the overlap indicator value is low. The overlap in red is very small, but due to the limited trip length, the overlap indicator value amounts to 50 % or less. If the trace contains many of such cases, the overlap indicator will be low, although the calculated trips might be acceptable for use in a prompted recall survey. However, our comparison is strengthened by taking into account the *nTripsInd*, the *avgDurInd* and the *avgDistInd* as well.

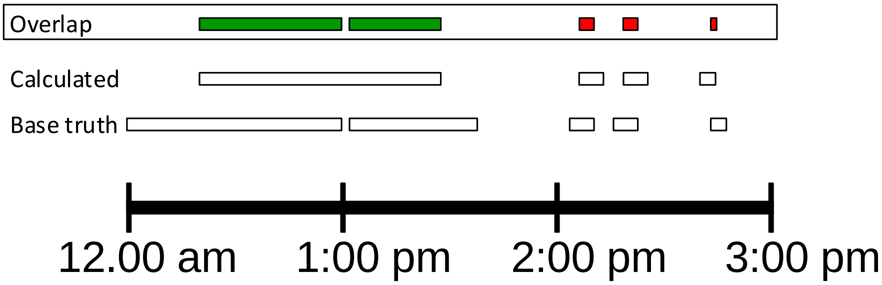


FIGURE 9.7: An illustration of the overlap between calculated trips and *base truth* trips.

Finally, Table 9.4 confirms the conclusion drawn from the visual analysis of the iso-lines. As a result, threshold values *180 seconds* for the duration and *100 metres* for the distance were chosen to use in the prompted recall survey.

9.7.2 Reasons for High Degree of Variability

As discussed in Section 9.7.1, no single pair of threshold parameters suits all cases. This is assumed to be caused by either the use of person traces as opposed to car traces or the accuracy and correctness of the *base truth* diaries.

The slow movement of people (e.g. walking) can be the cause for missing trips, especially in combination with short trips. Examples are social visits or daily shopping at a small distance from home. Due to the limited trip length and duration and the low speed of the trip, there is a large probability that this trip is just discarded or annotated as a stop. In general, trips are discovered more accurately when the trip is made using a fast moving vehicle and over a decent amount of kilometres. Therefore, it is likely that person characteristics such as work, preferences and possession of a car might be reasons for the diversity between the respondents.

The use of person traces means that each respondent carries a GPS recorder all the time, i.e. in the car, when walking or in buildings. Problems may occur when people enter a building. The GPS signal will lose much accuracy and outlier GPS recordings will show up. These outlier GPS points are problematic because they can pretend to represent trips. A solution was discussed in Section 9.3.2, however this solution might not work for every possible case.

Another reason can be found in the fact that the used *base truth* diaries are not as accurate as expected. The diaries were corrected by experts but some of their decisions follow from personal interpretation (opinion) and hence, might contain errors. In this process they used a lot of criteria in order to judge whether a person had a good diary and hence, could be selected to be corrected. The strongest criteria stated that activities and trips performed by a respondent should not be filled in more than 24 hours in advance; and 90 % of their activities had to be filled in at most 48 hours after they finished it. Based on those criteria only five of our 33 *base truth* diaries were found to qualify. The other 28 diaries were selected based on other (less strict) criteria. Therefore, it can be assumed that not every diary in our *base truth* set has the same quality. This is shown by plotting the LWMSE for every individual respondent. This can be seen in Figure 9.8. On the x-axis, 33 values which represents the different used diaries can be seen. On the y-axis the LWMSE can be found. Note that the LWMSE values are sorted in ascending order.

It is clear that (based on the LWMSE) the stop detector gives very good results for some respondents (LWMSE < 0.2), but one can notice also some outliers (LWMSE > 0.3). To show the difference, the stop detector together with the LWMSE were run on the five *base truth* diaries which met the strong criteria which as described above. This resulted in Table 9.6.

Compared to the WMSE results of Table 9.4, there is approximately a difference of 0.2 between the top five best results. One can also notice that there is almost no difference between the WMSE values of Table 9.6, which means that *dur-dist-combinations* give almost the same results. Although the analysis of the diaries which meet a strong criteria gives better results, the complete data set will be used in the

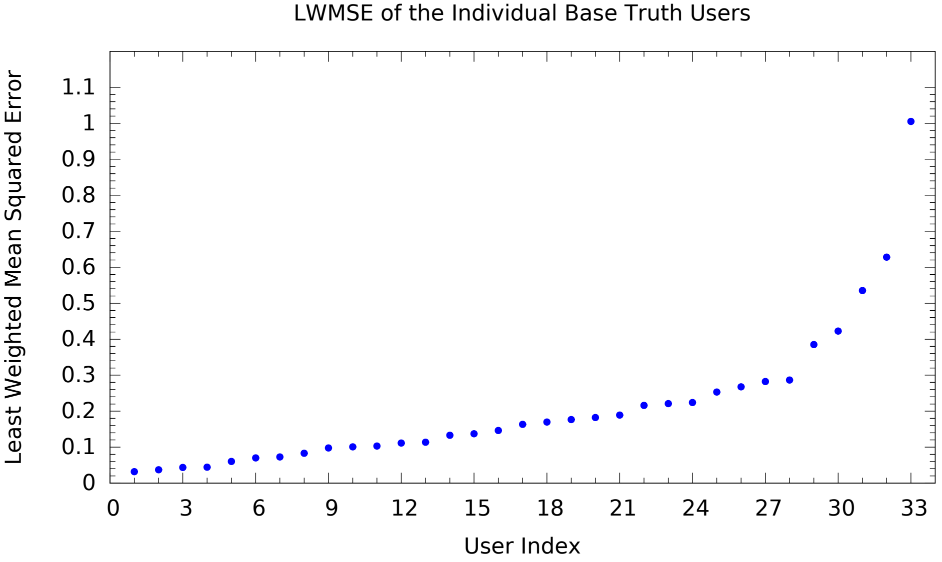


FIGURE 9.8: LWMSE of individual respondents calculated by the stop detector. X-axis represents the index of the respondents and the y-axis represents the LWMSE.

TABLE 9.6: The five best results based on the WMSE for the *base truth* diaries which met the strong correctness criteria for the stop detector. The different components of the WMSE and the original temporal indicator are reported. This table is sorted on the WMSE column in increasing order.

Dur[sec]	**Dist[m]**	$\mathbf{I}_{temp}$	$\mathbf{MSE}_{num}$	$\mathbf{MSE}_{temp}$	$\mathbf{MSE}_{dist}$	$\mathbf{MSE}_{dur}$	***WMSE***
180	125	0.6699	0.0970	0.1264	0.0422	0.09627	0.2927
210	150	0.6692	0.0862	0.1253	0.0396	0.12315	0.2930
180	200	0.6631	0.0681	0.1281	0.0185	0.17760	0.2942
180	100	0.6733	0.1155	0.1248	0.0460	0.07140	0.2990
180	150	0.6655	0.0849	0.1277	0.0377	0.14126	0.3021

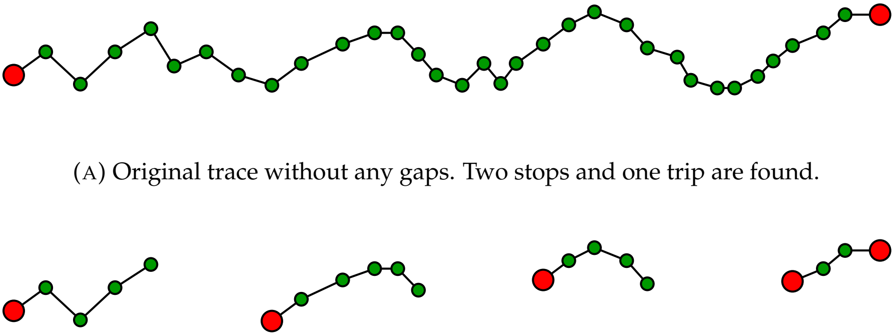
remainder of the paper. The reader shall keep in mind the limited quality of the available *base truth* data set.

9.7.3 Processing Gaps in Traces

The TRIP/STOP detectors behave differently when processing gaps in trips.

The stop detector will divide one trip in multiple trips with multiple stops which leads to an overestimation in both trips and stops. This situation can be seen in Figure 9.9. Figure 9.9a shows the original GPS trace without any gaps. Green circles represent GPS points that are part of a trip, while the bigger red circles represent stops. One trip and its delimiting stops are represented. When the GPS trace contains some gaps, the situation as illustrated in Figure 9.9b occurs. Because the trip is cut into pieces, three *fake stops* are detected which leads to four trips.

The trip detector requires that each trip starts and ends at a decently low speed. When a traveller switches the recording device on/off while driving, part of the trace is rejected as junk. In general, the trip detector finds less trips and stops when the trace contains gaps.



(A) Original trace without any gaps. Two stops and one trip are found.

(B) Original trace with gaps. Five stops and four trips are found.

FIGURE 9.9: An example of processing gaps in a trace.

The presence of gaps in a trace largely depends on the recording device. A device that is fixed to a car in general is switched on/off with the engine and the gaps are caused by passing tunnels etc. When smartphones are used, the recorder can be switched on/off by the owner which increases the probability to find gaps in such traces. Furthermore, such devices can stop recording due to lack of GPS signal reception, most likely when it is stored indoors.

To take care of those phenomena, the trace generator discussed in Section 9.8 distinguishes between both cases by means of the `holeGen_*` parameter settings.

9.8 Experiments Based on Synthetic Traces

In Section 9.7, the optimal settings for the TRIP/STOP detectors have been identified using recorded traces for which the associated travel diaries were available. The diaries revealed the stops; this is used as *base truth*. The information about the revealed stops was derived from the diaries and in some cases expert opinion was required to make definite decisions. Hence, the timing derived for the stops might contain errors.

Therefore, it was decided to create synthetic GPS traces in order to conduct a sensitivity analysis. Note that the purpose of this analysis is not to calibrate the input parameters, but to show the robustness of the stop detector.

This Section first describes the method used to create synthetic GPS traces along with the used parameter settings. Next a sensitivity analysis is described, analogous to the analysis which was already conducted in Section 9.7.1.

Gamma distributions can be defined in multiple ways. This paper takes the case

$$f(x) = gamma(x, \alpha, \beta) \tag{9.8}$$

$$\mu = E(x) = \frac{\alpha}{\beta} \tag{9.9}$$

$$\sigma = VAR(x) = \frac{\alpha}{\beta^2} \tag{9.10}$$

where α is the shape parameter and β is the rate parameter.

9.8.1 Characteristics of Synthetic Traces

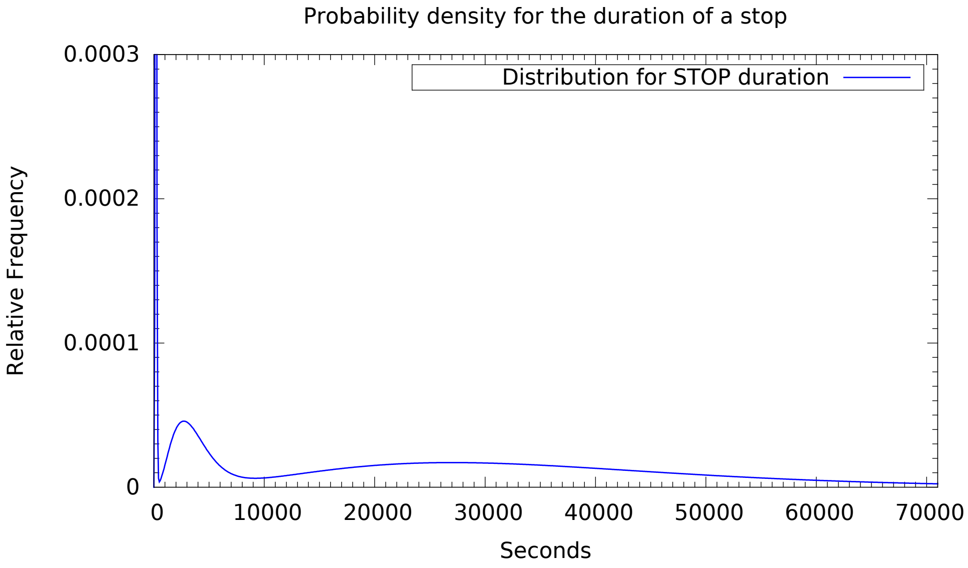


FIGURE 9.10: Probability density for the duration of a stop. The function is a linear combination of gamma densities having mean values 180 sec, 3,600 sec and 36,000 sec respectively. The combination coefficients used are 0.132, 0.183 and 0.686 respectively.

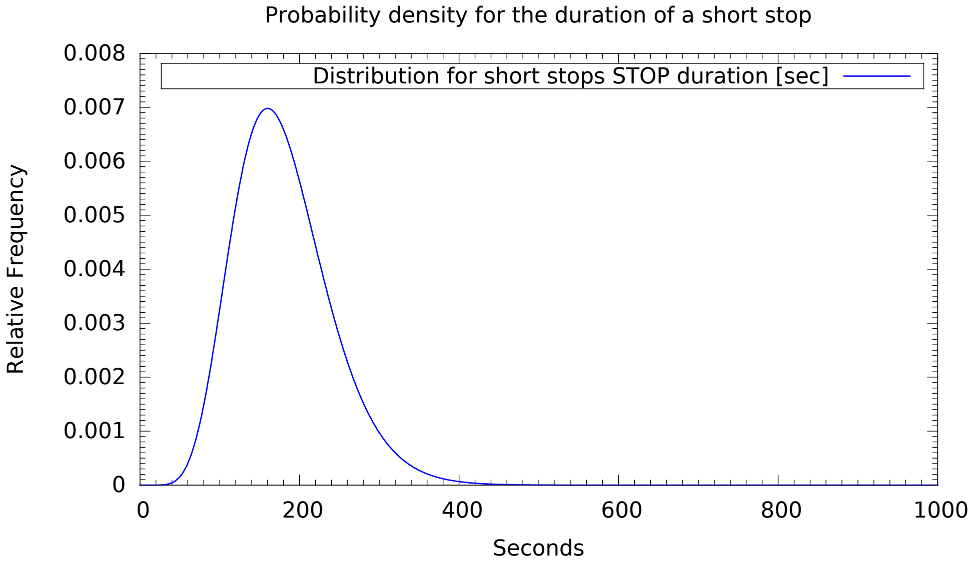


FIGURE 9.11: Probability density for the duration of a "*short*" stop.

Trip Characteristics

The synthetic trace generator is based on a path generator described in [Knapen et al., 2014], that delivers link sequences in a road network. For each traveller, a first stop location is sampled. Then a sequence of $\langle$trip, stop$\rangle$ pairs is generated. Each stop corresponds to a node in the network and each trip corresponds to a path in that network. The trip length is sampled from a uniform distribution.

Traveller Position during a Trip

The period Δt between successive GPS recordings is sampled from a gamma distribution. The mean and standard deviation (see Table 9.7) were estimated from the traces that were used to compare the TRIP/STOP detectors (see Section 9.6.1). A traveller is simulated to leave each node on the path starting at zero speed and to arrive in a node at near zero speed. The speed v on each link is derived by sampling from a uniform distribution for which the boundary values are determined from the allowed speed on the link.

The distance driven between successive recordings is $d = \Delta t \cdot v$. The distance D_l driven since the last visited node N_l is calculated by accumulating the d values and subtracting the distance $dist(N_0, N_l)$ between the origin N_0 and the last visited node N_l. It is expressed as a linear reference $r = D_l / L$ where L is the link length.

Finally, a coordinate pair $\langle lon, lat \rangle$ is generated using the link geometry and the linear reference r. Both *lon* and *lat* in the coordinate pair are disturbed with a random value sampled from a normal distribution in order to account for GPS accuracy. The standard deviation depends on whether the recording corresponds to a stop or to a trip because recordings in buildings may be less accurate.

Duration of a Stop

Stops correspond to *short* (e.g pick-drop), *medium* (e.g. shopping) and *long* (e.g. working) activities respectively. The duration of a stop is sampled from a density function that is a linear combination of gamma densities in order to account for differences between activity types. The density function used is given by

$$f(t) = \sum_{i=1,3} w_i \cdot gamma(\alpha_i, \beta_i) \tag{9.11}$$

$$\sum_{i=1,3} w_i = 1 \tag{9.12}$$

with (mean and standard deviation expressed in seconds). The w_i value is the probability for the activity to belong the the i^{th} class (short, medium, long). Figure 9.10 shows the probability density used to determine the stop duration values. Note the sharp peak for low values and the other two maxima. Figure 9.11 shows the distribution for the *short* stops; this corresponds to the sharp peak near the left in Figure 9.10. The weights shown in Table 9.7 were derived from a set of agendas predicted by the FEATHERS activity based model and hence, reflect the situation for Flanders, Belgium [Bellemans et al., 2010].

TABLE 9.7: Parameters for the gamma distributions used to sample the stop duration for "*short*", "*medium*" and "*long*" stops respectively.

	i	$\mathrm{w_i}$	Mean $\mathrm{m_i}$	StdDev $\mathrm{s_i}$	α_i	β_i
Short	1	0.132	180	60	9	20
Medium	2	0.183	3600	1800	4	900
Long	3	0.686	36000	18000	4	9000

Position Drift in Large Stop Locations

Moving around in a large area dwelling is modelled by the `stopDriftProbab` setting. Each stop location is identified by a coordinate pair $\langle lon, lat \rangle$ at which the person is assumed to reside. The GPS recordings are generated by adding a random error to the *lon* and *lat* values based on the receiver accuracy. The reference is moved to the last generated coordinate pair with the given probability `stopDriftProbab`. This models people walking around on a large site.

Holes (Gaps) in Recording Sequences

Generated pseudo GPS recordings pass a *hole generating filter* having *dropping* and *keeping* operating modes. This model accounts for interrupted recordings. Gamma distributions are used to sample (i) the time till the next *hole* and (ii) the duration of a hole. Whenever the hole generator receives a GPS recording while it is in the *keeping* mode, it moves to *dropping* mode with a given probability. If the mode changes, the duration for the hole is sampled from a distribution for which the parameters depend on whether the recording belongs to a stop or a trip. The probability to start generating a hole depends on whether the recording for which the decision is to be taken, belongs to a trip or to a stop. This accounts for disturbed GPS signal reception within buildings (but a hole starting during a stop can continue during the succeeding trip and vice versa, without changing its properties).

9.8.2 Setup

Parameter settings for the trace generator are subdivided in two categories. The first category applies to device and recording parameters that can be derived from a trace by the researcher extracting stops or trips. The recording frequency (or period) and the device accuracy belong to this category. For those parameters, the values were derived from recorded traces (details are explained below). The second category consists of settings that apply to the traveller behaviour which is unknown to the researcher. The expected value for the stop duration and for the trip distance mentioned in Section 9.8.1 belong to this category. Those parameters have been derived from a data set containing agendas predicted by the FEATHERS tool.

The recording period was derived from the recorded GPS traces as follows. The nominal value was known to be 30 sec. It was postulated that at most 10 consecutive recordings can get lost due to stochastic phenomena. Hence, for the traces being investigated, periods longer than 300 sec not containing any recording, are considered to constitute a *gap*. The frequency distribution for the length of the period between successive recordings was extracted from 33 traces. Values larger than 300 sec were discarded (this represents 0.23 percent of the cases. The mean m and standard deviation s for the remaining sample were determined and used to calculate values for α

TABLE 9.8: Settings used in the trace generator.

Parameter	Value
maxLengthLow	500
maxLengthHigh	30,000
lonLat_stdDevTrip	8.33
lonLat_stdDevStop	8.33
lonLat_stdDevStochastic	true
minSpeedFactor	0.7
maxSpeedFactor	1.2
minAccel	-2
maxAccel	1.5
samplePeriodMean	47.66
samplePeriodStdDev	24.94
stopDriftProbab	0.0001
teleportProbab	0.08
teleportTimeGapLow	300
teleportTimeGapHigh	86,400
holeGen_periodTillNext_mean	14,400
holeGen_periodTillNext_stdDev	2,400
holeGen_holeDurationInStop_mean	600
holeGen_holeDurationInStop_stdDev	150
holeGen_holeDurationInTrip_mean	90
holeGen_holeDurationInTrip_stdDev	30
holeGen_probInStop	0.7
holeGen_probInTrip	0.5
stopDur_a_weight	0.132
stopDur_b_weight	0.183
stopDur_c_weight	0.686
stopDur_a_mean	180
stopDur_a_stdDev	60
stopDur_b_mean	3,600
stopDur_b_stdDev	1,800
stopDur_c_mean	36,000
stopDur_c_stdDev	18,000

and β using

$$\alpha = \left(\frac{m}{s}\right)^2 \tag{9.13}$$

$$\beta = \frac{s^2}{m} \tag{9.14}$$

This is not a maximum likelihood estimation[1].

---

[1] `http://www.itl.nist.gov/div898/handbook/eda/section3/eda366b.htm`

In order to run the experiments, 98 traces were created. Each trace represents a synthetic person who performs 60 trips. Typically a trace covers a period of two weeks. This conforms to the average daily number of trips (which is 3.2) for an individual in Flanders. Each trace is generated by the same configuration file specifying the characteristics as described in Section 9.8.1. The numerical values for the setting are given in Table 9.8.

For each of these traces, the stop detector was executed with the same parameters as used in the previous experiment described in Section 9.7. Remember that ten different values were used for the stop distance and ten different values for the stop duration. Hence, in total $98 \times 10 \times 10 = 9800$ files were used to conduct the sensitivity analysis.

The technique described in Section 9.6 was used. Hence the *number of trips*, the *temporal trip matching indicator*, the *average distance* and the *average duration* are used to analyse the sensitivity. The results of this run can be found in Table 9.9.

TABLE 9.9: The five best results based on the WMSE for the synthetic traces for the stop detector. The different components of the WMSE and the original temporal indicator are reported. This table is sorted on the weight column in increasing order.

Dur[sec]	**Dist[m]**	$\mathbf{I}_{temp}$	$\mathbf{MSE}_{num}$	$\mathbf{MSE}_{temp}$	$\mathbf{MSE}_{dist}$	$\mathbf{MSE}_{dur}$	***WMSE***
150	200	0.8472	0.0096	0.0224	0.0034	0.0020	0.0347
150	150	0.8512	0.0095	0.0229	0.0032	0.0018	0.0349
150	250	0.8531	0.0096	0.0223	0.0041	0.0021	0.0350
150	125	0.8491	0.0095	0.0235	0.0032	0.0018	0.0355
150	100	0.8472	0.0094	0.0241	0.0031	0.0018	0.0359

9.8.3 Recurring *dur-dist-combinations*

Recurring *dur-dist-combinations* to which the algorithm is insensitive while generating high quality results are interesting. They are a measure for robustness of the technique and can suggest safe parameter settings for cases where the *base truth* is unknown.

Based on the presented results, there are some presumptions that *dur-dist-combination* range might be found in which the stop detector gives more or less the same results. Therefore, seven additional configuration files were created for which ten synthetic traces each containing 60 trips were generated. The parameter value settings are shown in Table 9.10. For each of these seven cases, the stop detector was run and the WMSE was calculated. The plot of these seven cases together with the *base truth* of Section 9.6.2 and the synthetic case used in Section 9.8.2 can be found in Figure 9.12.

TABLE 9.10: Parameter settings used to generate seven sets of synthetic traces. Only settings that differ among the cases are shown. *holeDurationInTrip* corresponds to the `holeGen_holeDurationInTrip_mean` setting in the configuration.

Name	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
stopDriftProbab	0.0001	0.0001	0.0001	0.0050	0.0050	0.0050	0.0050
holeDurationInTrip	90	180	180	90	90	180	180
stopDur_a_mean	120	180	120	180	120	180	120

There is a quite large set of *dur-dist-combinations* resulting in low WMSE values: they correspond to the near horizontal coinciding lines at the left in the diagram indicating that there is almost no difference in WMSE. To give a better view, a zoomed version is included in Figure 9.13.

FIGURE 9.12: An overview of the WMSE of different cases.

In Figure 9.13 one can see that for the first 30 *dur-dist-combinations* the WMSE value is less than 0.2 for all the cases except the *base truth* case. Reasons for the higher WMSE of the *base truth* case are discussed in Section 9.7.2. However, note that the shape of the diagram is similar for the real and synthetic cases. This means that, for each set of traces, there is a set of *dur-dist-combinations* that deliver similar accuracy. The accuracy is *set specific* (e.g. for the *base truth* traces the WMSE value is larger than the one for the synthetic traces). Also, the i^{th} *dur-dist-combination* is not necessarily the same one in each case because sorting was performed for each

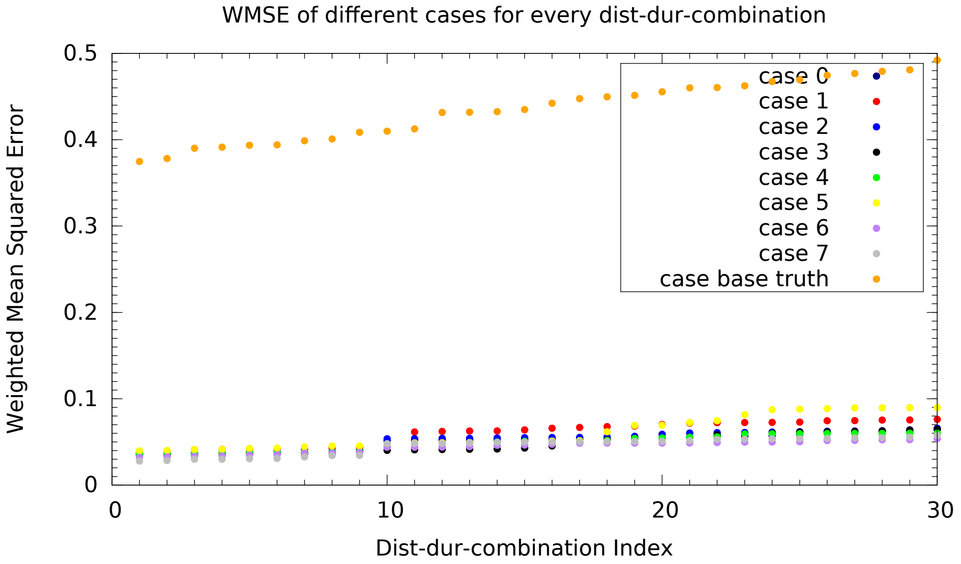


FIGURE 9.13: A zoomed version of Figure 9.12.

case independently. Then, one can wonder whether there are *dur-dist-combinations* that result in low WMSE values in *every* case. Therefore, the 30 *dur-dist-combinations* delivering the lowest WMSE for each case were investigated in order to find out which *dur-dist-combinations* do occur in all of the top 30 rankings. Nine *dur-dist-combinations* were present in every individual case; these *dur-dist-combinations* are listed in Table 9.11.

From this observation it is concluded that, when being faced with a new data set for which the nominal recording period is less than the value used in this paper (i.e. 30 sec) and which was collected using a recent standard quality GPS receiver, the highest accuracy can be expected using one of the *dur-dist-combinations* listed in Table 9.11. It is also advised to use multiple combinations and to compare the results to each other (due to lack of *base truth*) by means of the quality indicators specified in Section 9.6.2. Thereto, the set of trips and stops found using one particular *dur-dist-combination* randomly chosen from Table 9.11 shall be used as reference. It is obvious that this cannot be used to determine a *best option* but if large differences in the WMSE values are found, detailed investigation to find the cause is required.

TABLE 9.11: *Dur-dist-combinations* which give similar results.

Dur[sec]	Dist[m]
120	100
150	100
150	125
150	150
180	75
180	100
180	125
180	150
180	200

9.9 Conclusion

Several sets of real and synthetic GPS traces were analysed. First, a stop detector algorithm was compared to a trip detector algorithm in order to select one of them for integration in a *prompted recall* based survey. Overall, the stop detector performed better than the trip detector. This can be explained by the fact that the latter is selective with respect to speed variations while assessing GPS sequences to represent a trip and classifies some sequences as junk, which is not done by the stop detector.

A synthetic GPS trace generator was presented and sets of traces were created using different parameter settings. Those were used along with the real traces for which the base truth (trips and stops revealed by the survey respondents by means of diaries) was known to find optimal thresholds for distance (*stopDistanceThreshold*) and time (*stopDurationThreshold*). Due to the high variability of the results, this turns out to be a cumbersome task. Several combinations for the parameter settings were used. It was observed that, for both the synthetic and real traces, there seems to exist a collection of settings that deliver low errors for the weighted mean squared errors (WMSE) and to which the algorithm is quite insensitive. Such collection was identified and is advised to be used as a set of suitable values to detect stops in traces for which no *base truth* is available.

Bibliography

L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macedo, B. Moelans, and A. Vaisman. A Model for Enriching Trajectories with Semantic Geographical Information. In *Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems*, GIS '07, pages 22:1–22:8, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-914-2. doi: 10.1145/1341012.1341041.

T. Bellemans, B. Kochan, D. Janssens, G. Wets, T. Arentze, and H. Timmermans. Implementation Framework and Development Trajectory of FEATHERS Activity-Based Simulation Platform. *Transportation Research Record: Journal of the Transportation Research Board*, Volume 2175(Volume 2175 / 2010):111–119, December 2010. ISSN 0361-1981.

W. Bothe and K. Maat. Deriving and Validating Trip Purposes and Travel Modes for Multi-Day GPS-Based Travel Surveys: A Large-Scale Application in the Netherlands. *Transportation Research Part C: Emerging Technologies*, 17(3):285–297, 2009. ISSN 0968-090X. doi: 10.1016/j.trc.2008.11.004.

G. Cich, L. Knapen, T. Bellemans, D. Janssens, and G. Wets. TRIP/STOP Detection in GPS Traces to Feed Prompted Recall Survey. *Procedia Computer Science*, 52:262–269, 2015. ISSN 1877-0509. doi: 10.1016/j.procs.2015.05.074. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015).

G. Cich, L. Knapen, T. Bellemans, D. Janssens, and G. Wets. Threshold settings for TRIP/STOP detection in GPS traces. *Journal of Ambient Intelligence and Humanized Computing*, 7(3):395–413, Jun 2016. ISSN 1868-5145. doi: 10.1007/s12652-016-0360-9.

J. Du and L. Aultman-Hall. Increasing the Accuracy of Trip Rate Information from Passive Multi-Day GPS Travel Datasets: Automatic Trip End Identification Issues. *Transportation Research Part A: Policy and Practice*, 41(3):220–232, 2007. ISSN 0965-8564. doi: 10.1016/j.tra.2006.05.001.

C. Eui-Hwan and S. Amer. A Trip Reconstruction Tool for GPS-based Personal Travel Surveys. *Transportation Planning and Technology*, 28(5):381–401, 2005. doi: 10.1080/03081060500322599.

B. Furletti, P. Cintia, C. Renso, and L. Spinsanti. Inferring Human Activities from GPS Tracks. In *UrbComp 13 Proceedings of the Second ACM SIGKDD International Workshop on Urban Computing*, Chicago, 2013. ACM.

R. Guidotti, R. Trasarti, and M. Nanni. TOSCA: Two-Steps Clustering Algorithm for Personal Locations Detection. In *23rd ACM SIGSPATIAL*, Seattle, Washington, November 2015. ACM. doi: 10.1145/2820783.2820818.

L. Knapen, T. Bellemans, D. Janssens, and G. Wets. Canonic Route Splitting. *Procedia Computer Science*, 32:309–316, 2014. ISSN 1877-0509. doi: 10.1016/j.procs.2014.05.429. The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014).

Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma. Mining User Similarity Based on Location History. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, pages 34:1–34:10, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-323-5. doi: 10.1145/1463434.1463477.

C. Ma, J. Cao, L. Yang, J. Ma, and Y. He. Effective Social Relationship Measurement Based on User Trajectory Analysis. *Journal of Ambient Intelligence and Humanized Computing*, 29(1):1–16, Feb 2012. ISSN 1868-5137. doi: 10.1007/s12652-012-0120-4.

T. K. Rasmussen, J. B. Ingvardson, K. Halldórsdóttir, and O. A. Nielsen. Improved Methods to Deduct Trip Legs and Mode from Travel Surveys using Wearable GPS Devices: A Case Study from the Greater Copenhagen Area. *Computers, Environment and Urban Systems*, 54:301–313, 2015. ISSN 0198-9715. doi: 10.1016/j.compenvurbsys.2015.04.001.

A. Raza, L. Knapen, K. Declercq, T. Bellemans, D. Janssens, and G. Wets. Diary Survey Quality Assessment using GPS Traces. *Procedia Computer Science*, 52:600–605, 2015. ISSN 1877-0509. doi: 10.1016/j.procs.2015.05.045. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015).

S. Schönfelder, K. Axhausen, N. Antille, and M. Bierlaire. Exploring the Potentials of Automatically Colected GPS Data for Travel Behaviour Analysis: A Swedish Data Source. Technical report, Ecole Polytechnique Fédérale de Lausanne, 2002.

N. Schüssler and K. Axhausen. Processing Raw Data from Global Positioning Systems without Additional Information. *Transportation Research Record: Journal of the Transportation Research Board*, 2105:28–36, Dec 2009. ISSN 0361-1981. doi: 10.3141/2105-04.

L. Spinsanti, F. Celli, and C. Renso. Where You Stop Is Who You Are: Visited, 2010.

P. Stopher, C. Prasad, and J. Zhang. Can GPS Replace Conventional Travel Surveys? Some Findings. In *Australasian Transport Research Forum 2010 Proceedings*, Canberra, Australia, October 2010.

S. Tsui and A. Shalaby. Enhanced System for Link and Mode Identification for Personal Travel Surveys Based on Global Positioning Systems. *Transportation Research Record: Journal of the Transportation Research Board*, 1972(1):38–45, Jan 2006. ISSN 0361-1981. doi: 10.3141/1972-07.

X. Xiao, Y. Zheng, Q. Luo, and X. Xie. Inferring Social Ties Between Users with Human Location History. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):3–19, Feb 2014. ISSN 1868-5137. doi: 10.1007/s12652-012-0117-z.

Z. Yan. Traj-ARIMA: A Spatial-time Series Model for Network-constrained Trajectory. In *Proceedings of the Third International Workshop on Computational Transportation Science*, IWCTS '10, pages 11–16, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0429-0. doi: 10.1145/1899441.1899446.

Z. Yan and S. Spaccapietra. Towards Semantic Trajectory Data Analysis: A Conceptual and Computational Approach. In *VLDB2009 (PhD Workshop)*, 2009.

Part IV

Discussion and Conclusion

Chapter 10

Discussion

10.1 Future Research

10.1.1 Reducing Number of Comparisons in the Exhaustive Search DCC Algorithm

In Chapter 8, an exhaustive search algorithm was introduced. In cases were there were a lot of paths per driver, the algorithm was unable to find an optimal (based on a scoring function) solution, since it was not able to check all the feasible combinations within the allowed computation time interval. In Section 8.11, a suggestion was made that the pruning steps could be improved by ordering the drivers in a specific way before trying to combine them.

A method to order the drivers in a ascending order based on the number of paths is proposed. Assume that the drivers are handled in a sequence $D_1, \ldots D_n$ and that the number of paths for driver D_i is m_i. As seen in Section 8.8.1, the enumeration is represented by a tree: the number of leafs is the number of cases to evaluate ($\prod\limits_{i=1}^{n} m_i$). The presented trees (examples in Figure 8.6) are completed by adding a root element. At the first level no *real* checks are required. The root is at level zero. The level of a vertex is the level of its parent plus one. Consider level i in the tree: the number of vertices at level i is $|V_i|$ and $|V_i| = |V_{i-1}| * m_i$. This is equal to the number of edges linking level $i-1$ to level i and hence the number of checks to add to level $i-1$, then the number of edges is

$$\sum_{i=1}^{n} \Big(\prod_{j=1}^{i} m_i \Big) = m_1 + m_1 \cdot m_2 + m_1 \cdot m_2 \cdot m_3 + m_1 \cdots m_n \qquad (10.1)$$

and the number of *real* checks is

$$N_C = \sum_{i=1}^{n} \left(\prod_{j=1}^{i} m_i \right) - m_1 = m_1 + m_1 \cdot m_2 + m_1 \cdot m_2 \cdot m_3 + m_1 \cdots m_n - m_1 \tag{10.2}$$

$$N_C = \sum_{i=2}^{n} \left(\prod_{j=1}^{i} m_i \right) = m_1 \cdot m_2 + m_1 \cdot m_2 \cdot m_3 + m_1 \cdots m_n \tag{10.3}$$

This is minimised by ordering the drivers to increasing number of paths because this ensures that the largest m values occur in as few terms as possible. The gain only is large in cases where the m values heavily differ. To give the reader an idea of possible efficiency increases a sampling was conducted. Drivers and number of paths per driver were both sampled based on $X \sim U(1, 100)$. The formula was used for the random sample (*withoutSorting*) and for the sorted alternative (*withSorting*). Log values were used because of the very large numbers. Next, the gain was measured by $gain = e^{withoutSorting - withSorting}$. The results showed a mean increase of 5.23. The complete results can be seen in Figure 10.1.

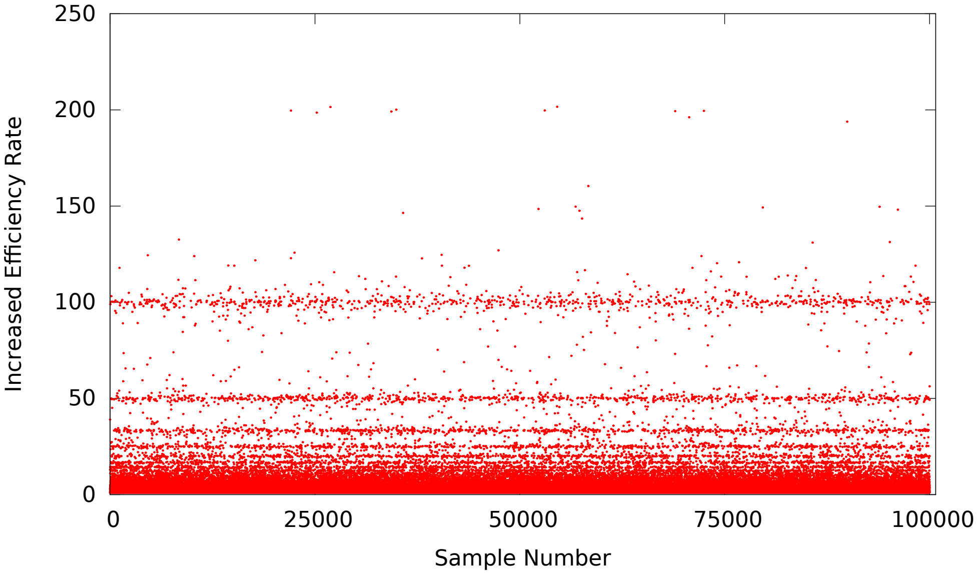


FIGURE 10.1: An overview of the increase rates if the input is sorted based on the number of paths per driver. The x-axis shows the sample number, while the y-axis shows the increase rate.

One should take in mind that despite some very high increased efficiency rates, in some cases the algorithm will not end within a specific time. However, the benefit is that a lot more solutions will be found.

10.1.2 New Method to Find Stops and Trips in GPS Traces

In this thesis two algorithms to find trips and stops were presented. In the meantime the idea was formulated to detect *stops* by independently analysing two time series of recorded coordinates. A *stop* is detected as soon as for a sufficiently long period of time both $x(t)$ and $y(t)$ are (nearly) constant. The method is based on linear regression and already implemented in a Bachelor's thesis [Vanherle, 2017] and the results were promising. However, more research should be conducted to improve this method.

Basic Concepts about Linear Regression

The conditional expectation represented in Equation (10.4) is a linear function of x, where β_0 is the intercept and β_1 the slope. The slope is of interest for this algorithm since it measures the trend of the data. Furthermore, a least squares linear regression method is used.

$$G(x) = E\{Y|X = x\} = \beta_0 + \beta_1 x \tag{10.4}$$

The goal is to find a function $\hat{G}(x)$ that approaches the data points as close as possible. This can be conducted by minimising the *residuals* which is the difference between the observed data y_i and the fitter value $\cap y_i = \cap G(x_i)$ for a given point in the data set. Following functions are used:

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} \tag{10.5}$$

$$\hat{\beta}_0 = \bar{y} - b_1\bar{x} \tag{10.6}$$

$$S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{10.7}$$

$$S_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{10.8}$$

In the equations, $\bar{x}$ stands for the mean of the predictors, while $\bar{y}$ stands for the means of the responses.

Now t-statistic should be used to test some hypotheses. As mentioned before, the slope is of interest, hence the following hypotheses are examined:

$$H_0 : \beta_1 = 0 \quad H_a : \beta_1 \neq 0 \tag{10.9}$$

The t value is calculated as follows

$$t = \frac{\beta_1 - 0}{\frac{s}{\sqrt{S_{xx}}}} \tag{10.10}$$

where s is the regression standard deviation

$$s = \sqrt{\frac{SS_{ERR}}{n-2}} \tag{10.11}$$

and where n represents the number of points taken into account in the regression. Furthermore SS_{ERR} is calculated as follows

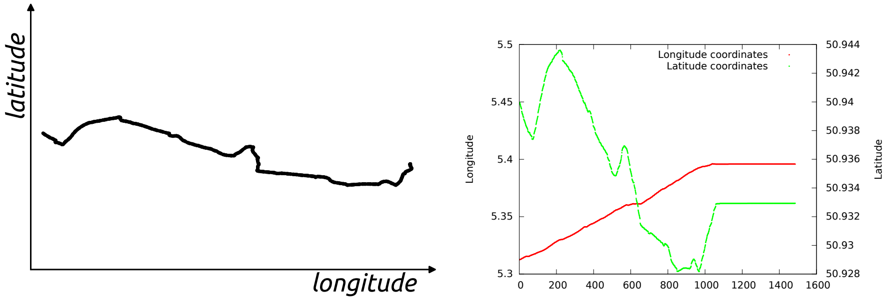
$$SS_{ERR} = \sum(y_i - \hat{y})^2 \tag{10.12}$$

Based on this t-statistic, the null-hypothesis can be rejected or accepted depending on the chosen significance level.

Regression Used for Stop Detection

The main idea is to use these statistics to determine whether a person stood still. The slope hypothesis testing is conducted on the x coordinates and y coordinates separately by using a sliding window. In order to conduct this method, the GPS trace $\{(x_1, y_1, t_1), (x_2, y_2, t_2), \ldots, (x_n, y_n, t_n)\}$ where x represents the longitude, y represents the latitude and t the timestamp are transformed into pairs instead of triples. The x and y coordinate will be separated as follows: $\{(x_1, t_1), (x_2, t_2), \ldots, (x_n, t_n)\}$ and $\{(y_1, t_1), (y_2, t_2), \ldots, (y_n, t_n)\}$. The timestamp acts as a predictor, while x and y act as the response value.

Hence, why are the slopes needed to investigate the traces? In Figure 10.2a, an example of raw GPS traces is shown, while in Figure 10.2b, the corresponding longitudes and latitudes are plotted as a function of time. The movement of the GPS traces goes from left to right. It is possible to reconstruct the GPS traces based on the plot in Figure 10.2b. An increasing slope of the longitude coordinates shows a movement in the east direction. A positive slope for the latitude coordinates means a movement in the north direction, while a decrease means a movement to the south. It is clear to see that when both plots have a slope of zero (more or less horizontal), this means that there is no movement any more in the traces. This is around point 1,000 in Figure 10.2b.



(A) The trace represented in a x-y frame.

(B) The traces represented as two time series. The red dots are the longitude coordinates and the green dots represent the latitude coordinates.

FIGURE 10.2: An example of a trace, consisting of a trip and a stop.

To find out whether a person has stood still for a certain amount of time, the stop time T is defined first. A sliding time window of the same size T is used. A sliding window scans the traces and the regression slope (for both time series) is calculated for all the GPS points within this window. Hence every point in the data set has a regression slope calculated based on the sliding window with size T. An example of this can be found in Figure 10.3. Two sample points with their corresponding window are represented.

To decide whether there is a stop in movement in both directions, the concept of a *directional stop* is introduced. A directional stop is a period of consecutive points (in a direction) for which the slope is proven to be zero *or* the point lies at most T time units before a point with slope zero. Such an example can be found in Figure 10.4

Finally, to detect actual stops, the directional stops in both directions are compared based on time. Overlap period of directional stops which last for more than T, are supposed to be actual stops as can be seen in Figure 10.5.

This method showed promising results, but it was only used for a small set of GPS traces. Without further research it is hard to say whether this method will perform better than the already presented methods.

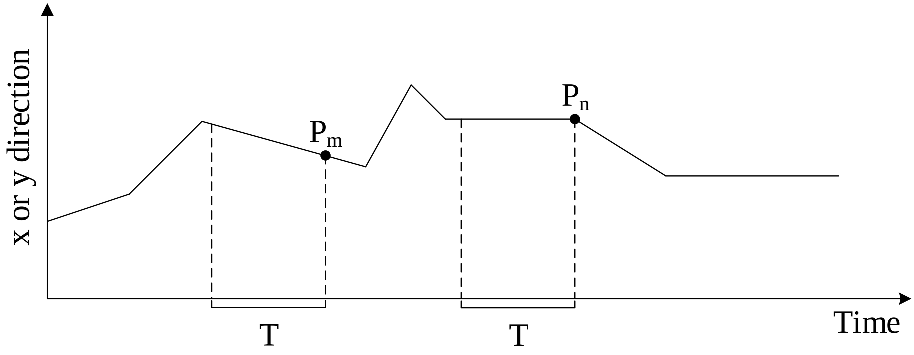


FIGURE 10.3: An example of two regression windows for two data points for the x or y direction.

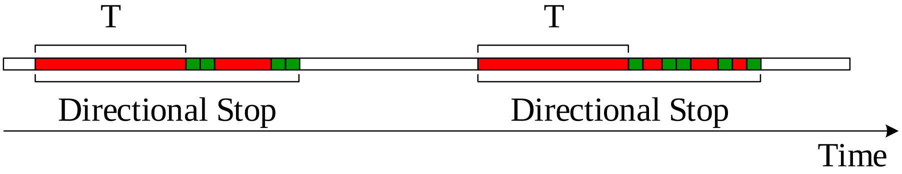


FIGURE 10.4: An example of the detection of directional stops. Green blocks represent points with a slope of zero, while the red blocks represent points which are positioned within *T* steps before a point with a regression slope of zero.

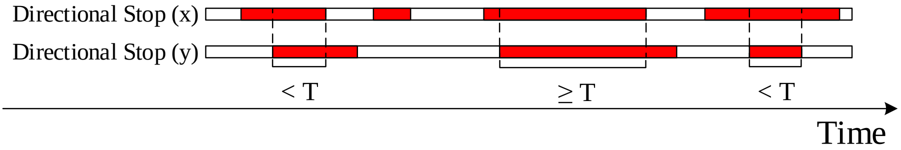


FIGURE 10.5: An example which shows how stops are extracted from directional stops both in x as y direction.

10.2 Increased Parallelism in Conservative Synchronisation

In Chapter 4, a conservative synchronisation approach was developed. In the mean time, this approach was studied well and a few improvements were implemented. These improvements increased the parallelism with some nice runtime reductions. Exactly the same experiment was conducted; the results can be found in Figure 10.6. Clearly, in the old version, almost no parallelism was allowed. It is clear that the execution time highly depends on the number of events. The increase is linear as was expected.

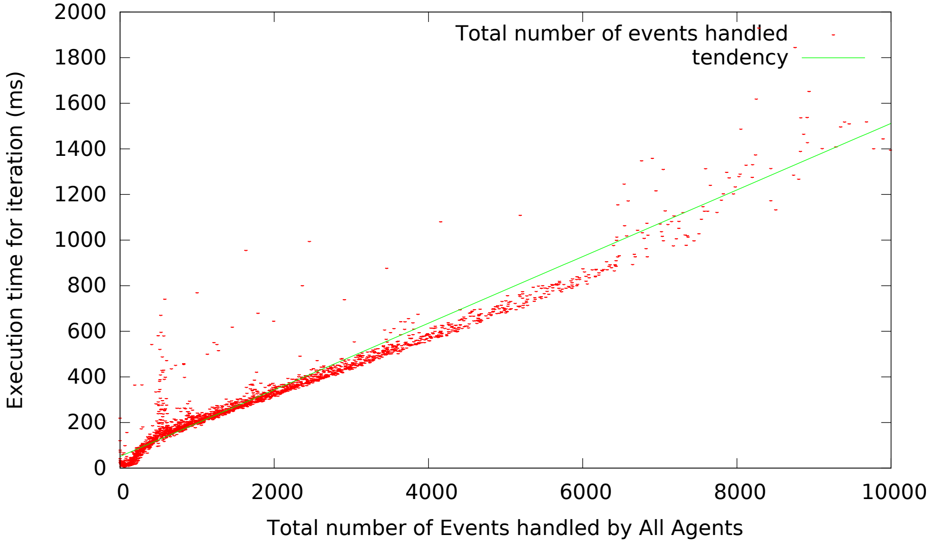


FIGURE 10.6: Graph that represents the total number of events handled in a specific iteration (x-axis) against the total execution time for that iteration in ms (y-axis) for the case of 200 agents and 2,500 iterations for the improved version.

10.3 Difficulties to Get Private Data

In Chapter 8, an extensive number of simulations were done on sampled data which was based on data of a real DCC. In the mean time, there was another opportunity to cooperate with a two additional DCCs. However, there was no time left to conduct the administrative duties to set up the cooperation. Since this data deals with highly personal data, the ethical commission needs to be contacted and furthermore, every participant of the DCC should give their permission. The introduction of the GDPR made it even harder.

This was solved by using the mean and standard deviation for the home-to-DCC distances together with the number of participants and the locations of the DCCs. This information was enough to redo the experiments which were explained in Chapter 8. Hence, the data is again sampled and the experiments were again tested against different values for the time window width (TWW) and maximum detour time (MDT).

Two DCCs will be investigated and for privacy reasons they will be referred to as DCC A and DCC B. The characteristics of both centres can be found in Table 10.1. The column header "DCC" indicates the DCC, "Number of Guests" indicates how many people need to reach the DCC every day, "Mean Distance" indicate the average distance from guests' homes to the DCC and "Std. Distance" indicates the standard deviation.

DCC	Number of Guests	Mean Distance [m]	Std. Distance [m]
DCC A	67	7,185.54	3,374.20
DCC B	18	6,414.93	3,115.66

TABLE 10.1: The characteristics of the two DCCs.

Results are presented analogously as in Chapter 8. Since, one of the cases is very large, it was decided to not take into account the case of 45 minutes for MDT and TWW. Furthermore, 45 minutes for MDT and TWW are quite high and probably not realistic.

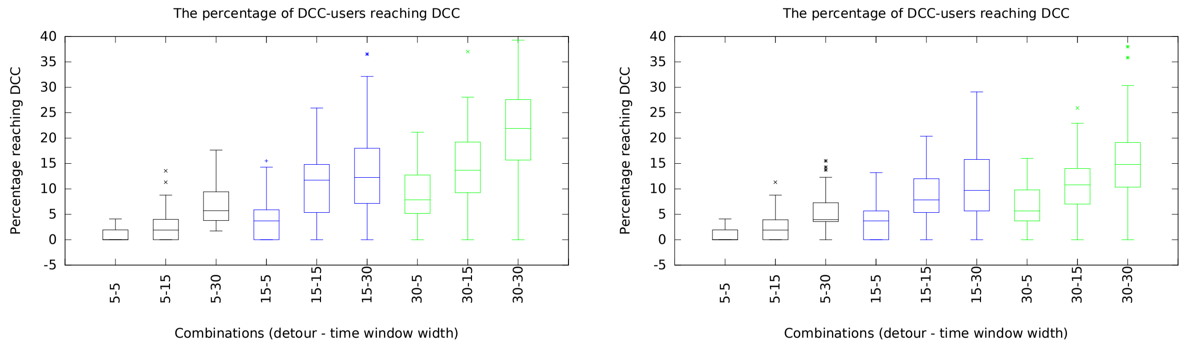
10.3.1 DCC A

For the two-stage heuristic (transfer locations), four neighbouring schools were used. The initial situation, where no voluntary drivers are present can be seen in Table 10.2. On average around 54 DCC guests have to go to the DCC every day. Depending on the TWW, they can be transported by 11 to 16 buses.

The results of the percentage of DCC guests reaching the DCC and the amount of saved kilometres are visualised in Figure 10.7 and Figure 10.8 respectively.

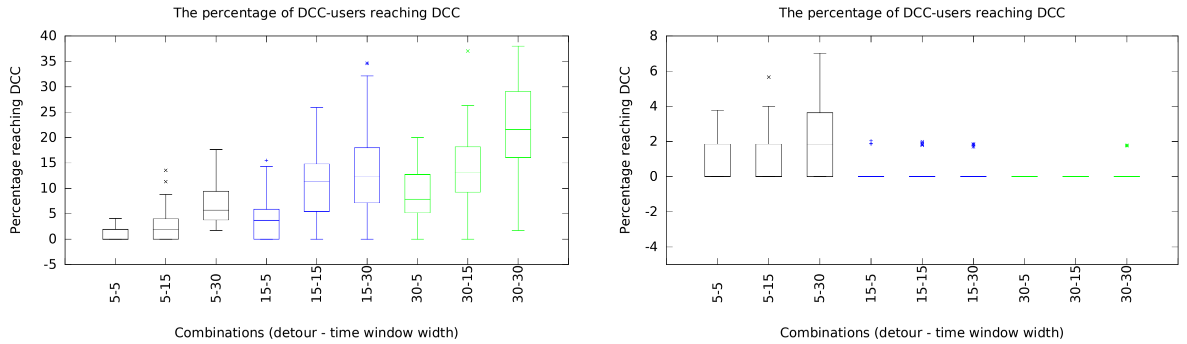
TABLE 10.2: Results of the cases without voluntary drivers for DCC A. Note that this are average results of 50 cases (five days per iteration and ten iterations). The column nStuckLoc. represents in this case the number of homes of the visitors.

Combination		**nStuckLoc.**	**nBus**	**Avg. dist/bus [km]**
MDT	**TWW**			
5	5	53.60	15.02	32.53
5	15	53.82	13.98	39.86
5	30	53.90	10.86	46.93
15	5	53.62	15.74	31.40
15	15	54.20	14.24	40.29
15	30	54.10	10.84	48.97
30	5	53.54	15.28	31.80
30	15	53.46	13.74	39.80
30	30	53.28	10.54	48.89



(A) Results of the exhaustive search method with the minimising stuck locations scoring function.

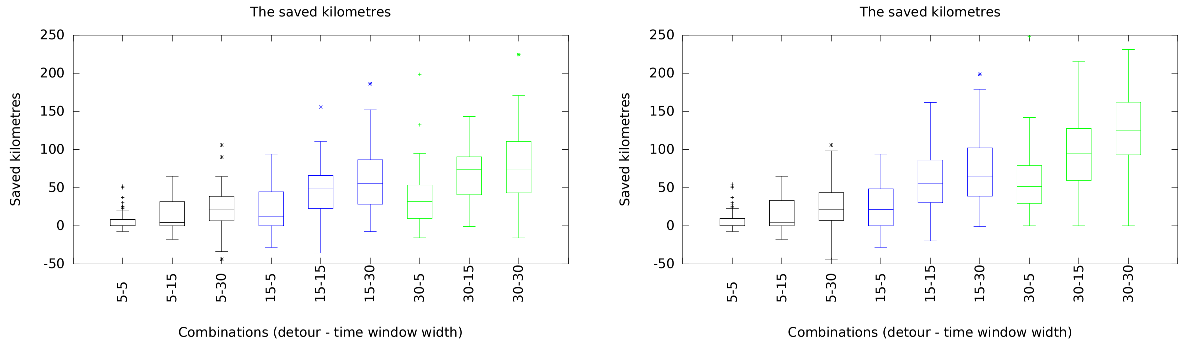
(B) Results of the exhaustive search method with the minimising chartered bus costs scoring function.



(C) Results of the two-stage heuristic (DCC) with the minimising chartered bus costs scoring function.

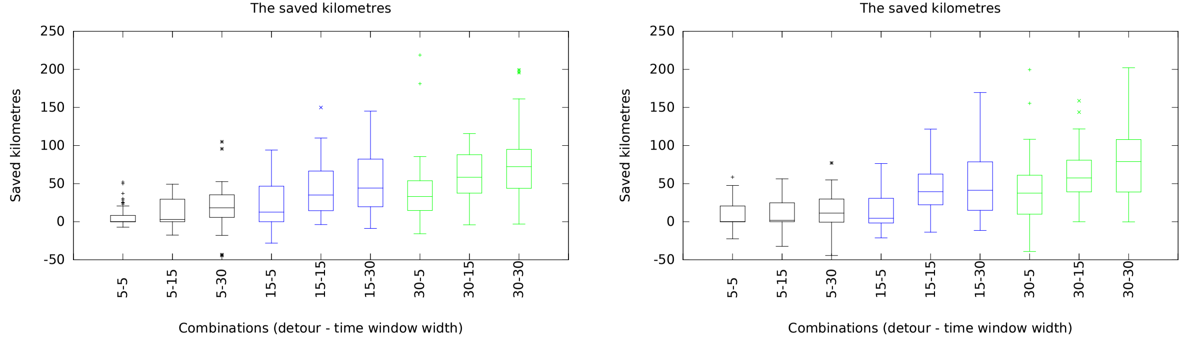
(D) Results of the two-stage heuristic (transfer locations) with the minimising chartered bus costs scoring function.

FIGURE 10.7: An overview of the results of the percentage of DCC guests actually reaching the DCC by using voluntary drivers for DCC A.



(A) Results of the exhaustive search method with the minimising stuck locations scoring function.

(B) Results of the exhaustive search method with the minimising chartered bus costs scoring function.



(C) Results of the two-stage heuristic (DCC) with the minimising chartered bus costs scoring function.

(D) Results of the two-stage heuristic (transfer locations) with the minimising chartered bus costs scoring function.

FIGURE 10.8: An overview of the results of the saved amount of kilometres for the chartered vehicles for DCC A.

The comparison between the different algorithms can be seen in Table 10.3. They all perform similarly. However the main goal to reduce the number of buses seems to be hard in this case. In some cases on average one to two buses can be saved. Note that since the case is quite large, a lot of exhaustive searches were stopped after one hour of computation. Optimal results for these algorithms were not obtained.

TABLE 10.3: Overview of the results for DCC A with (i) $S1$: exhaustive search method (minimising stuck locations scoring function), (ii) $S2$: exhaustive search method (minimising chartered bus costs scoring function), (iii) $S3$: two-stage heuristic (DCC) (minimising chartered bus costs scoring function) and (iv) $S4$: two-stage heuristic (TLs) (minimising chartered bus costs scoring function). Grey cells indicate that at least one of the simulations were stopped after an hour of computation. Column headers are (i) combination of maximum detour time (MDT) and time window width (TWW), (ii) the percentage of DCC guest which reach the DCC with carpooling (% reach. DCC), (iii) the number of stuck locations (nStuckLoc.), (iv) the number of buses needed for the stuck passengers (nBus) and (v) the average distance per bus to serve the stuck passengers (Avg. dist/bus).

Combination		**% Reach. DCC**				**nStuckLoc.**			
MDT	**TWW**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
5	5	1.20	1.20	1.20	0.56	52.64	52.64	52.66	52.96
5	15	3.04	2.78	3.00	0.83	51.48	51.58	51.66	52.42
5	30	6.64	5.30	6.57	1.98	49.62	50.14	50.06	51.56
15	5	4.05	3.79	4.05	0.11	51.56	51.68	51.64	52.44
15	15	10.30	8.04	10.11	0.18	48.28	49.24	48.72	49.90
15	30	13.58	10.81	13.31	0.11	45.36	46.40	46.36	47.74
30	5	9.00	6.42	8.81	0.00	47.26	48.46	47.42	48.20
30	15	14.45	10.73	14.22	0.00	45.02	46.50	45.72	46.68
30	30	21.28	15.42	21.95	0.07	41.80	43.74	42.32	42.66

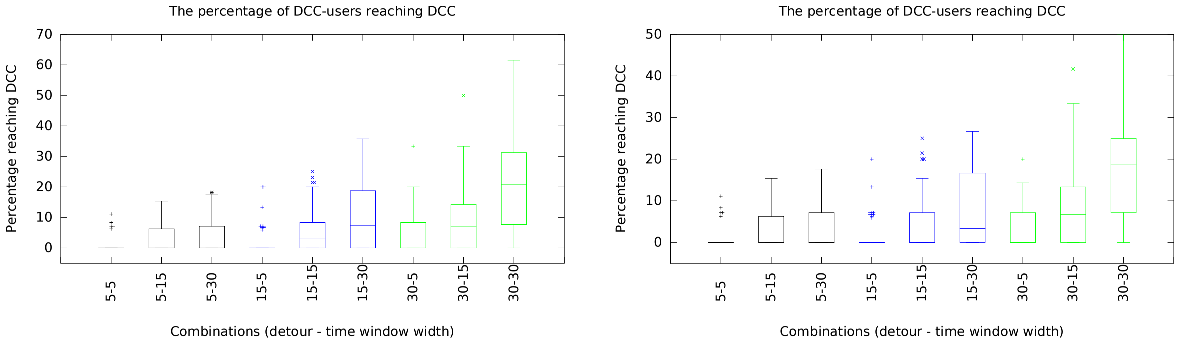
Combination		**nBus**				**Avg. dist/bus [km]**			
MDT	**TWW**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
5	5	14.76	14.74	14.76	14.76	31.99	31.96	31.97	31.90
5	15	13.38	13.34	13.40	13.48	39.73	39.77	39.83	39.88
5	30	10.56	10.40	10.60	10.70	47.04	47.43	47.06	47.41
15	5	14.84	14.78	14.88	15.08	31.09	31.01	31.00	31.15
15	15	12.78	12.36	12.80	12.76	39.44	39.80	39.65	39.82
15	30	9.68	9.18	9.90	9.96	47.76	48.73	47.54	47.55
30	5	13.44	12.80	13.26	13.32	32.15	32.12	32.36	32.21
30	15	11.86	11.08	12.18	12.16	40.58	40.72	40.33	40.21
30	30	9.54	8.20	9.64	9.42	47.39	49.61	47.43	47.91

10.3.2 DCC B

In Table 10.4, the results can be found where no voluntary drivers are present. Around 14 DCC guests need to go to the DCC every day. Between three and five buses are needed to bring them. The actual results can be found in Figure 10.9 and Figure 10.10. It can be seen that only a very small portion of the DCC guests can use carpooling. For the two-stage heuristic (transfer locations), three neighbouring schools were used. In this case, the reduction of buses is constrained to one bus.

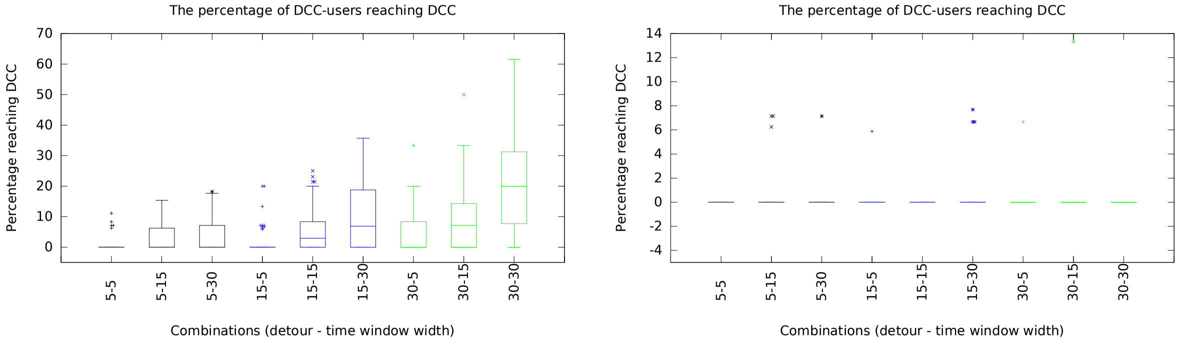
TABLE 10.4: Results of the cases without voluntary drivers for DCC B. Note that this are average results of 50 cases (five days per iteration and ten iterations). The column nStuckLoc. represents in this case the number of homes of the visitors.

Combination		**nStuckLoc.**	**nBus**	**Avg. dist/bus [km]**
MDT	**TWW**			
5	5	14.46	2.56	21.58
5	15	14.54	4.44	25.78
5	30	14.68	3.98	33.94
15	5	14.42	2.70	24.28
15	15	14.14	4.30	26.94
15	30	14.12	4.08	32.12
30	5	14.64	2.74	22.28
30	15	14.42	4.42	26.75
30	30	14.52	4.14	34.02



(A) Results of the exhaustive search method with the minimising stuck locations scoring function.

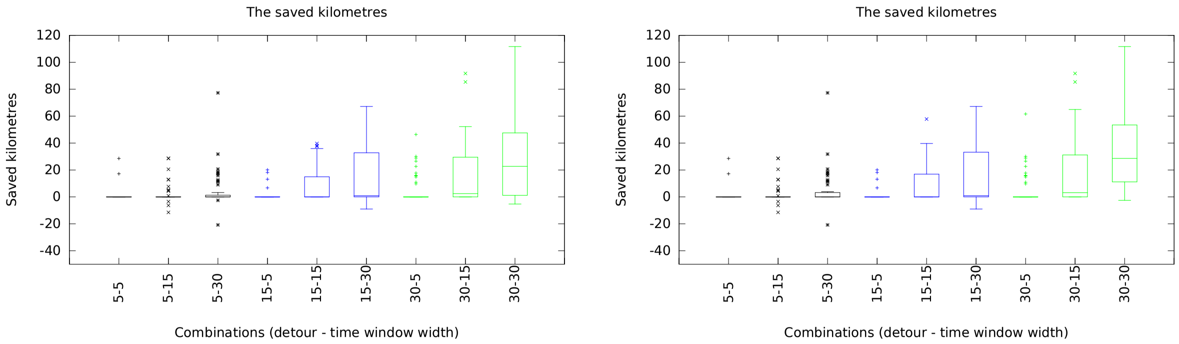
(B) Results of the exhaustive search method with the minimising chartered bus costs scoring function.



(C) Results of the two-stage heuristic (DCC) with the minimising chartered bus costs scoring function.

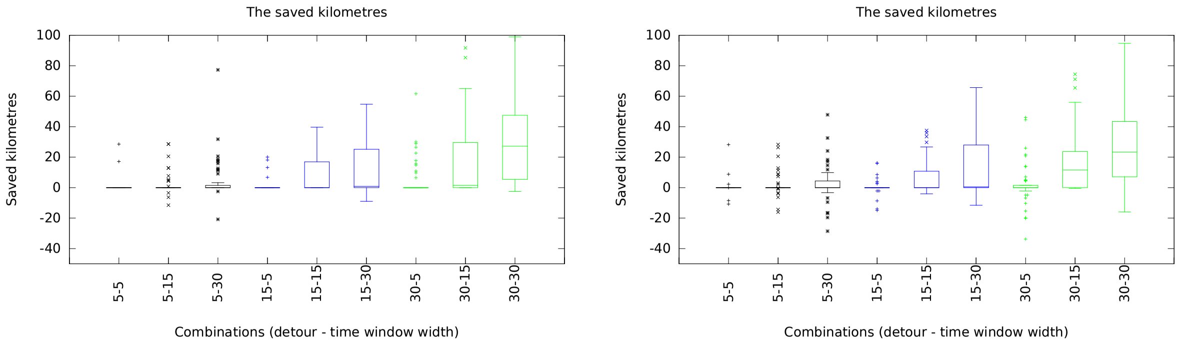
(D) Results of the two-stage heuristic (transfer locations) with the minimising chartered bus costs scoring function.

FIGURE 10.9: An overview of the results of the percentage of DCC guests actually reaching the DCC by using voluntary drivers for DCC B.



(A) Results of the exhaustive search method with the minimising stuck locations scoring function.

(B) Results of the exhaustive search method with the minimising chartered bus costs scoring function.



(C) Results of the two-stage heuristic (DCC) with the minimising chartered bus costs scoring function.

(D) Results of the two-stage heuristic (transfer locations) with the minimising chartered bus costs scoring function.

FIGURE 10.10: An overview of the results of the saved amount of kilometres for the chartered vehicles for DCC B.

TABLE 10.5: Overview of the results for DCC B with (i) $S1$: exhaustive search method (minimising stuck locations scoring function), (ii) $S2$: exhaustive search method (minimising chartered bus costs scoring function), (iii) $S3$: two-stage heuristic (DCC) (minimising chartered bus costs scoring function) and (iv) $S4$: two-stage heuristic (TLs) (minimising chartered bus costs scoring function). Grey cells indicate that at least one of the simulations were stopped after an hour of computation. Column headers are (i) combination of maximum detour time (MDT) and time window width (TWW), (ii) the percentage of DCC guest which reach the DCC with carpooling (% reach. DCC), (iii) the number of stuck locations (nStuckLoc.), (iv) the number of buses needed for the stuck passengers (nBus) and (v) the average distance per bus to serve the stuck passengers (Avg. dist/bus).

Combination		**% Reach. DCC**				**nStuckLoc.**			
MDT	**TWW**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
5	5	0.82	0.82	0.82	0.00	14.58	14.58	14.58	14.68
5	15	2.77	2.49	2.77	0.41	14.06	14.08	14.06	14.32
5	30	4.03	3.47	4.03	0.14	13.82	13.88	13.82	14.24
15	5	2.06	1.79	2.06	0.14	14.26	14.30	14.26	14.42
15	15	5.96	5.15	5.96	0.00	13.86	13.96	13.88	14.34
15	30	9.91	7.87	9.63	0.41	13.26	13.46	13.32	13.78
30	5	5.17	3.91	5.17	0.14	13.58	13.76	13.58	14.00
30	15	9.50	7.99	9.37	0.28	13.12	13.28	13.16	13.68
30	30	21.04	17.81	20.48	0.00	11.24	11.56	11.34	12.00

Combination		**nBus**				**Avg. dist/bus**			
MDT	**TWW**	**S1**	**S2**	**S3**	**S4**	**S1**	**S2**	**S3**	**S4**
5	5	2.62	2.62	2.62	2.66	25.26	25.26	25.26	25.22
5	15	4.16	4.16	4.16	4.22	25.76	25.76	25.76	25.56
5	30	3.88	3.88	3.88	3.94	33.41	33.37	33.41	33.60
15	5	2.68	2.68	2.68	2.76	22.55	22.55	22.55	22.42
15	15	4.24	4.22	4.22	4.40	26.30	26.41	26.53	25.98
15	30	3.88	3.86	3.96	4.02	32.93	32.94	32.82	31.93
30	5	2.68	2.68	2.70	2.90	22.21	22.05	21.98	22.33
30	15	4.00	3.92	4.02	4.08	26.47	26.39	26.45	25.99
30	30	3.48	3.26	3.42	3.48	30.51	30.53	30.92	30.77

10.3.3 General Remarks

The results of the two presented DCCs are not as good (at least in the field of transportation) as the results presented in Chapter 8, although the techniques were identical. Only the locations of the DCC and the average and standard deviation of the distances of the guests have changed. The average distance between the homes of guests and the DCC in Chapter 8 was around 5,870 metres with a standard deviation of 2,582 metres. There is a clear difference between these numbers and the numbers presented in this section, however it is only a difference of 1,000 to 2,000 metres. If the locations are taken into account, a more clear difference can be observed. The case in Chapter 8 has a DCC located in an urban area, while the two DCCs in this Section are located in more rural areas. As a result, it is harder for drivers to make detours and still arriving on time at their destination. Furthermore, it seems to be clear, that a detour of 30 minutes is an absolute minimum to get some reasonable results.

Bibliography

B. Vanherle. Stop detection in GPS trajectories, 2017.

Chapter 11

Conclusion

11.1 Main Findings

In this thesis Computer Science was successfully used in the field of Transportation Science. As a result three software tools were presented; all of them taking care of different aspects of thin flows.

In Part I, a robust **agent-based demand-supply micro-simulator** toolkit was presented. A newly developed tool was used to find **anomalies in OSM as well as in GTFS**. In OSM data for Flanders, a lot anomalies were found regarding attributes, such as speed, number of lanes etc. These were solved with a rule based algorithm. Furthermore, links were found with zero length and nodes which did not have any value for the network (= useless) were removed from the data set. Solving these issues resulted in a more reliable OSM network which is capable to be used for microscopic simulations. In GTFS, most of the problems occurred based on "missing connections" between the different files. Some stops were not used or trips did not belong to any route. This cleaned GTFS was used by OpenTripPlanner to act as an **external PT router API.**

The agent-based framework SARL was successfully extended with a **conservative synchronisation** mechanism. This method is ready to be used as a dedicated extension for SARL.

As a synthetic population, a **sampled subset of the thin flows** was used in the simulation. This synthetic population is considered to be similar to the real population on an aggregate level. The simulator was used for a specific use case in Flanders, Belgium. However, it could be used for different purposes, e.g. simulating effects of policy changes; in this thesis the simulator was used to check the **viability of transport providers** within the concept of the upcoming basic accessibility in Flanders, Belgium. Twelve cases were investigated where prices as well as fleet size were used as a variable. Results showed an expected shift in mode share and also in profit depending on the tariffs and fleet sizes. Furthermore, it is **hard for transport providers to collect sufficient fares to be viable**. Hence, a main suggestion is that ATS should be combined with normal taxis because the demand for those compensated providers is quite low in comparison with normal taxi providers. By doing this, the car capacities will be used more efficiently and it fits better in the new concept of basic accessibility.

In Part II a different subset of the thin flow population was investigated. Due to drastic economising by the Flemish government, subsidies for daily commuting to specialised centres such as a day care centre (DCC) for mobility impaired people are reduced. As a consequence, it becomes expensive to get picked up by a chartered bus at home in the morning and picked up at a DCC in the evening. Since in these cases, most of the people cannot drive themselves, a **carpool solution organised by volunteers** was proposed. The goal was to find a proper cooperation between carpooling and DRT providers. Both an **exhaustive search method**, as well as **heuristics** were developed and used on **sampled data based on a real case data set**. Results showed that some savings could be made, but the savings level highly depends on the flexibility of the drivers; on average a **cost reduction between 20 % and 30 %** could be achieved.

In Part III a stop detector algorithm to be used in a data collection tool was presented. This stop detector was compared with an existing trip detector. A tool to conduct a **sensitivity analysis** to find out the **best settings** was presented. The main advantage is that it can be used on **real data** (if present) and on sampled data. The sampled data mimics GPS traces quite realistically, even simulating connection losses. **Data containing previously interactively annotated stops and trips** were already present and were used for **validation purposes**. As mentioned before, the software was not used to follow thin flows people, although it would be perfectly suitable for them. However, the software tool proved to be useful in the ICOMFlex project, iScape project and a project in Tanzania.

Appendix A

Curriculum Vitae

Glenn Cich

Hasselt University Campus Diepenbeek
Transportation Research Institute (IMOB)
Wetenschapspark 5 bus 6
BE-3590 Diepenbeek Belgium

Tel: +32 (0)11 26 91 31
Fax: +32 (0)11 26 91 99
E-mail: glenn.cich@uhasselt.be

PERSONAL DETAILS

Place of birth	Heusden, Belgium
Date of birth	1991-Apr-05
Marital status	Legally cohabiting
Nationality	Belgian

EDUCATIONAL BACKGROUND

2012–2014	Master of Science: Master of Information Sciences, Databases Institute: UHasselt (Computer Science) Grade: Great Distinction Thesis: NoSQL: een vergelijkende studie
2009–2012	Bachelor of Science: Bachelor of Information Sciences Institute: UHasselt (Computer Science) Thesis: Massive Graphs
2003–2009	Latin-Mathematics Institute: Don Bosco College Hechtel

WORK EXPERIENCE	
2018-Sep-01 – present	InfoFarm Job title: Data Engineer
2014-Sep-01 – 2018-Aug-31	UHasselt/IMOB Job title: Researcher - PhD Student (supervisor prof. Dr. Davy Janssens)

Appendix B

List of Publications

B.1 As First Author

B.1.1 Peer-Reviewed Journals

Cich et al. [2016a]	(Published)
Cich et al. [2018a]	(Published)
Cich et al. [2018b]	(Submitted)

B.1.2 Peer-Reviewed Conferences

Cich et al. [2015]	(Published and presented by Glenn Cich)
Cich et al. [2016c]	(Published and presented by Glenn Cich)
Cich et al. [2016b]	(Published and presented by Glenn Cich)
Cich et al. [2017b]	(Published and presented by dr. ir. Luk Knapen)
Cich et al. [2017a]	(Published and presented by dr. ir. Luk Knapen)

B.1.3 Presentation

Cich [2015]	(Presented by Glenn Cich)

B.2 As Co-author

B.2.1 Peer-Reviewed Journals

Vuurstaek et al. [2018]	(Published)

B.2.2 Peer-Reviewed Conferences

Vuurstaek et al. [2017]	(Published and presented by Jan Vuurstaek)
Knapen et al. [2018]	(Published and presented by dr. ir. Luk Knapen)

Bibliography

G Cich. Demand-Responsive Collective Transportation Including Door-to-Door Services for Mobility Impaired People. *Mobility-management, Traffic-safety and Simulations (MTS) summer school*, Jul 2015.

G. Cich, L. Knapen, T. Bellemans, D. Janssens, and G. Wets. TRIP/STOP Detection in GPS Traces to Feed Prompted Recall Survey. *Procedia Computer Science*, 52:262–269, 2015. ISSN 1877-0509. doi: 10.1016/j.procs.2015.05.074. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015).

G. Cich, L. Knapen, T. Bellemans, D. Janssens, and G. Wets. Threshold settings for TRIP/STOP detection in GPS traces. *Journal of Ambient Intelligence and Humanized Computing*, 7(3):395–413, Jun 2016a. ISSN 1868-5145. doi: 10.1007/s12652-016-0360-9.

G. Cich, L. Knapen, S. Galland, J. Vuurstaek, A. Neven, and T. Bellemans. Towards an Agent-based Model for Demand-Responsive Transport Serving Thin Flows. *Procedia Computer Science*, 83:952–957, 2016b. ISSN 1877-0509. doi: 10.1016/j.procs.2016.04.191. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.

G. Cich, J. Vuurstaek, L. Knapen, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Data Preparation to Simulate Public Transport in Micro-Simulations using OSM and GTFS. *Procedia Computer Science*, 83:50–57, 2016c. ISSN 1877-0509. doi: 10.1016/j.procs.2016.04.098. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / Affiliated Workshops.

G. Cich, S. Galland, L. Knapen, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Addressing the Challenges of Conservative Event Synchronization for the SARL Agent-Programming Language. In Yves Demazeau, Paul Davidsson, Javier Bajo, and Zita Vale, editors, *Advances in Practical Applications of Cyber-Physical Multi-Agent Systems: The PAAMS Collection*, pages 31–42, Cham, 2017a. Springer International Publishing. ISBN 978-3-319-59930-4.

G. Cich, L. Knapen, M. Maciejewski, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. Modeling Demand Responsive Transport using SARL and MATSim. *Procedia Computer Science*, 109:1074–1079, 2017b. ISSN 1877-0509. doi: 10.1016/j.procs.2017.05.387. 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017) / Affiliated Workshops.

G. Cich, I. Ben-Arroyo Hartman, L. Knapen, and D. Janssens. A Simulation Study of Commuting Alternatives for Day Care Centres. *Future Generation Computer Systems*, 2018a. ISSN 0167-739X. doi: 10.1016/j.future.2018.05.009.

G. Cich, L. Knapen, S. Galland, and D. Janssens. Viability Evaluation of Transport Supply for Thin Flows with Agent-Based Simulation. *European Journal of Transport and Infrastructure Research*, 2018b. Submitted.

L. Knapen, G. Cich, and D. Janssens. A Task Scheduling Method for Agent/Activity-Based Models. *Procedia Computer Science*, 130:761–766, 2018. ISSN 1877-0509. doi: 10.1016/j.procs.2018.04.132. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / Affiliated Workshops.

J. Vuurstaek, G. Cich, L. Knapen, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. GTFS Bus Stop Mapping to the OSM Network. *Procedia Computer Science*, 109:50–58, 2017. ISSN 1877-0509. doi: 10.1016/j.procs.2017.05.294. 8th International Conference on Ambient Systems, Networks and Technologies, (ANT 2017) / Affiliated Workshops.

J. Vuurstaek, G. Cich, L. Knapen, W. Ectors, A.-U.-H. Yasar, T. Bellemans, and D. Janssens. GTFS bus stop mapping to the OSM network. *Future Generation Computer Systems*, 2018. ISSN 0167-739X. doi: 10.1016/j.future.2018.02.020.

Appendix C

Education

C.1 Taught Courses

2014–2015	Management en Predictie van de Verkeersvraag
2015–2016	Inleiding tot Geografische Informatiesystemen/Geografische Informatiesystemen

C.2 Supervised Bachelor Theses

2016–2017	Philippe Goethals	Potentiële Rol van taxidiensten binnen Basisbereikbaarheid
2016–2017	Bram Vanherle	Stop Detection in GPS Trajectories

C.3 Supervised Master Theses

2015–2016	Sadaqat Ali	Daily Plan (Schedule) Adaptation in MATSim
2016–2017	Ahmad Adeel	Imputation of Missing Values in OSM Networks

C.4 External Jury Bachelor Theses

2017–2018	Karel Vandebergh	Drones in de EU: Regelgeving en Mogelijke Toepassingen

C.5 External Jury Master Theses

2015–2016	Hamid Hussain Sabir	Use of Quick Chargers: A Framework for Evaluation of DC Quick Chargers for Electric Vehicles Using Activity Based Travel Patterns Predicted by FEATHERS for Flanders (Belgium)
2015–2016	Iuliia Naumova	Building Traffic Models using Freely Available Data
2016–2017	Farhan Shakeel	Location Choice Models based on Open Data Sources
2017–2018	Maarten Jacobs	Onderzoek naar het Mobiliteitsprofiel van Woonzorgcentra en Assistentiewoningen
2017–2018	Ozge Schevernels	Benefits of Service Integration in Demand Responsive Services; Flanders Case

Appendix D

Doctoral Schools related Sessions

D.1 Discipline Specific Activities

D.1.1 Teaching

Fulfil at least 4 teaching assignments throughout your PhD

- Management & Predictie van de Verkeersvraag (2014–2015)
- Geografische Informatiesystemen (2015–2016)
- Master Thesis: Sadaqat Ali (2015–2016)
- Job Student (Internship): Hendrik Janssen (2015–2016)
- Master Thesis: Ahmad Adeel (2016–2017)
- Bachelor Thesis: Philippe Goethals (2016–2017)

Modules on education

- Activerende Werkvormen
- Begeleiden van Bachelor- en Masterproeven
- Werk- en evaluatievormen
- Education needs Global Citizenship. Reflecties op wereldburgerschap in het Hoger Onderwijs

D.1.2 Internationalisation & Global Citizenship

Follow an info session about (funding) possibilities to perform research abroad

- Knowledge on the Move

International stay (summer school, visiting a research group, stage/internship,...) (optional)

- /

D.1.3 Seminars & Advanced Courses

Follow at least 12 seminars/research presentations from international speakers throughout your PhD

- 1-3
  1. Prof. dr. Kris Luyten - Data Visualisation
  2. Dr. Eef Theunissen - Effecten van Legale en Illegale Drugs op Rijvaardigheid en Cognitie
  3. Dr. Gautier Krings - Real Impact Analysis
- 4-6
  4. Prof. dr. Kai Nagel & Prof. dr. Carlo Prato - Current Research
  5. Prof. dr. Gerard de Jong & Prof. dr. Tomás Ruiz - Current Research
  6. Prof. dr. Christopher Zegras - Current Research
- 7-9
  7. Prof. dr. Rasa Ušpalytė-Vitkūnienė - Why Public Transport?
  8. Prof. dr. Alfonso Montella - Current Research
  9. Eric Sempels - Basisbereikbaarheid en Basismobiliteit
- 10-12
  10. Dr. Nicholas Race - The Role of SDN & NFV in Building Future Networks
  11. Prof. Itzhak Benenson - Modeling Cruising for Parking
  12. Dr. Antonio Fernández Anta - Saving Energy by Powering Down Links

Follow 1 info session about "Computational science/supercomputer" or an equivalent

- HPC Introduction

Follow at least 1 advanced course (a master course, a tutorial, a technical course,...)

- Data Sim Summerschool 2014
- Snellezen
- Mobility-Management, Traffic-Safety and Simulations (MTS) 2015
- Agent-Based Modeling
- Graph Theory Seminars
- Impact and Innovation in H2020 - a guide for proposers
- Flames Annual Meeting - FAM2018: Implementing the EU General Data Protection Regulation. Data Scientist, wake up and be prepared for GDPR!

D.1.4 IP & Valorisation

Follow an info session about Intellectual property/patents, search for patents or an equivalent

- Information session Intellectual Property

Info sessions/courses about valorization of innovation, how to protect IP,... (optional)

- Impact and Innovation in Horizon 2020

D.2 Non-Discipline Specific Activities

D.2.1 Scientific & Generic Communication

Follow an info session about Literature Searching skills or an equivalent

- Literature Searching Skill

Follow at least 1 course about scientific and generic communication

- Academic English

Follow an info session about reviewing skills or an equivalent

- Reviewing Skills

D.2.2 Research Management

Follow at least one of the 2 sessions about PhD management or an equivalent

- Project and Time Management: Managing my PhD

Follow an info session on writing a successful project proposal or an equivalent

- /

Organization of lectures, symposia (optional)

- /

D.2.3 Ethics & Research Integrity

Follow an info session about ethics & research integrity or an equivalent

- Ethics & Research Integrity

D.2.4 Career & Personal Development

Follow at least 1 course / info session about career & personal development

- 4 keys to build valuable relations during events

Samenvatting

Aangezien technologie telkens weer evolueert, worden steeds meer en meer sectoren met elkaar gecombineerd om onderzoek uit te voeren. Elk jaar worden computers sneller en krachtiger en daarom ideaal om te gebruiken voor onderzoek. In deze thesis worden Computerwetenschappen en Mobiliteitswetenschappen met elkaar gecombineerd. De thesis is een bundeling van drie software tools die elk een afzonderlijke situatie behandelen. Het bevat tools voor simulaties, optimalisaties en data collectie.

Sociale uitsluiting kan een groot probleem voor de maatschappij zijn. Deze thesis belicht mogelijke problematische situaties en onderzoekt de bijhorende gevolgen en oplossingen. Bijzondere aandacht zal besteed worden aan bepaalde situaties in Vlaanderen, België, namelijk (i) de evolutie van *basismobiliteit* naar *basisbereikbaardheid* (ii) de introductie van het *compensatiedecreet* en (iii) het reduceren van *subsidies*.

Ondanks dat de focus van de thesis gevestigd is op de lokale situatie, kunnen de tools toch eenvoudig gebruikt worden voor andere regio's. De geschikte data moet dan wel verzameld worden.

In de mobiliteitssector is er altijd een vervoersvraag en een vervoersaanbod. *Vervoersvraag* wordt gebruikt voor de mensen die vervoer nodig hebben, terwijl *vervoersaanbod* gebruikt wordt voor entiteiten die vervoer aanbieden. In de thesis wordt het meeste aandacht geschonken aan het vervoersaanbod. Toch zal de vervoersvraag (in mindere mate) ook aan bod komen. De gebruikers van deze transportdiensten delen dezelfde kenmerken. Ze kunnen niet zelf met de auto rijden en daarenboven is het ook niet eenvoudig voor hen om gebruik te maken van Openbaar Vervoer (OV). Dit kan te maken hebben met hun mobiliteitsbeperking, met hun leeftijd of door het gebrek aan OV haltes in de buurt.

De thesis kadert in het huidige mobiliteitslandschap in Vlaanderen, waarin er momenteel een overgang bezig is tussen het huidige model van basismobiliteit naar basisbereikbaarheid. *Basismobiliteit* zorgt ervoor dat alle inwoners van Vlaanderen toegang hebben tot OV. Mensen die niet over een auto beschikken of mensen die niet (meer) zelf kunnen rijden, moeten OV kunnen gebruiken. Dit model is duidelijk "aanbod gedreven", wat wil zeggen dat bussen altijd bepaalde trajecten rijden, ook al is er op dat moment geen vraag naar. Regelgeving over locaties van de OV haltes, de frequentie, de amplitude (het tijdsvenster waarin OV wordt aangeboden) etc. werden ontwikkeld en neergeschreven, gebaseerd op de locatie (verstedelijkingsgraad) en tijd (daluren vs. piekuren). Dit soort van transport bleek zeer duur te zijn voor de Vlaamse Overheid. Sommige OV lijnen zijn succesvol en goed bezet, andere daarentegen zijn nauwelijks bezet; dit resulteert in een zeer inefficiënt gebruik van de buscapaciteit. Voor dit soort lijnen is het concept van de "belbus" bedacht. Dit is een soort van vraaggestuurd vervoer waarbij passagiers een dag op voorhand hun trips moeten aanvragen. De belbus zal enkel rijden als er tenminste één passagier vervoer heeft aangevraagd. De bus zal een vooropgelegde route afleggen en zal stoppen bij de OV haltes die voordien werden aangevraagd. Aangezien de Vlaamse OV aanbieder moet besparen en de kosten voor de belbussen zeer hoog zijn, is het concept van de belbus langzamerhand aan het verdwijnen. Het resultaat is dat een (klein) deel van de bevolking mogelijks problemen heeft om OV te gebruiken.

Het unieke concept van basismobiliteit was (en is nog altijd) zeer duur. Naast de hoge kosten, worden de buscapaciteiten niet optimaal benut en "multi-modaliteit" zoals het combineren van OV trips met ander vervoer wordt niet aangemoedigd.

Door al deze tekortkomingen werd het concept van *basisbereikbaarheid* geïntroduceerd. Met dit concept wil de Vlaamse Overheid al het beschikbare vervoer integreren in één geconnecteerd netwerk waarin passagiers zonder problemen kunnen overstappen van het ene vervoersmiddel naar het andere (combi-mobiliteit). Het belangrijkste uitgangspunt is hier dat mensen ergens moeten geraken en in vergelijking met basismobiliteit, is basisbereikbaarheid vraaggestuurd in plaats van aanbodsgestuurd. De Vlaamse Overheid wil een *gelaagd netwerk* introduceren dat bestaat uit (i) treinnet, (ii) kernnet, (iii) aanvullend net en (iv) vervoer op maat. Het *treinnet* is de ruggengraat van het vervoersnetwerk, zowel nationaal als internationaal en wordt verzorgd door treinen. Het *kernnet* connecteert belangrijke steden en andere belangrijke locaties. Dit vervoer wordt uitgevoerd door bussen, trams en metro's. Het *aanvullend net* waarin bussen opereren, connecteert kleinere steden met het kernnet en/of treinnet. Het fungeert als een feederdienst voor de andere netten en het zorgt voor piekuurdiensten voor huis-werk en huis-school verplaatsingen. Ten slotte, zorgt het *vervoer op maat* voor het lokaal vervoersaanbod. Het bestaat uit allerlei vervoersinitiatieven om gebieden met een lage vervoersvraag te kunnen ondersteunen. Het hoofddoel is om te dienen als aanvoer naar het kernnet [Mobiel Vlaanderen, 2018].

Deze maatregelen worden ondersteund door het "decreet tot compensatie van de openbaredienstverplichting tot het vervoer van personen met een handicap of een ernstig beperkte mobiliteit" [Vlaams Parlement, 2012] (in Vlaanderen beter bekend als het compensatiedecreet). Dit decreet is van toepassing sinds december 2012 en zorgt voor een vervoerssysteem dat gesubsidieerd en aangepast vervoer aanbiedt voor gans Vlaanderen. Het decreet introduceert het Openbaar Aangepast Vervoer (OAV) (eerder gekend als Dienst Aangepast Vervoer (DAV)). De DAVs zijn operationeel in 27 vervoersgebieden. In elk vervoersgebied is er één vervoersaanbieder gecompenseerd. Deze vervoersaanbieder dient in het bezit te zijn van ten minste één aangepast voertuig. Hoeveel compensatie een vervoerder ontvangt, hangt af van de afstand en of de passagier in een rolstoel zit. Klanten kunnen OAV gebruiken als ze voldoen aan bepaalde eigenschappen. Naast OAV, zijn er ook nog de *Minder Mobielen Centrales (MMC)* met striktere regelgeving. Alleen mensen die een inkomen hebben dat lager ligt dan twee keer het huidige leefloon, dat afhankelijk van de gezinssituatie tussen de €595,13 en €1 190,27 ligt, mogen beroep doen op de MMC [POD Maatschappelijke Integratie, 2018]. De chauffeurs van de MMC zijn vrijwilligers die hun eigen wagen gebruiken. De karakteristieken van OAV zijn meer situatieafhankelijk; OAV aanbieders moeten voor elke passagier zelf bepalen of ze gebruik kunnen maken van de compensaties, gebaseerd op verschillende criteria zoals de persoonlijke situatie of zelfs het weer. MMC aanbieders daarentegen gebruiken enkel het objective criterium van het inkomen. Momenteel zijn voor elke vervoersregio gecomenpenseerde vervoerders aangeduid.

In deze thesis wordt er onderzoek gedaan naar *dunne stromen* binnen het huidige mobiliteitslandschap in Vlaanderen. Een stroom wordt gedefinieerd als een *route* tussen een herkomst en een bestemming, samen met een *periodieke sequentie*. Voorbeelden van een periodieke sequentie zijn (i) elke maandag van september 2018 tussen 09:00u en 12:00u, (ii) elke dag van 12:00u tot 16:00u tussen 1 januari 2018 en 31 juli 2018. Een stroom ratio (intensiteit) kan vervolgens berekend worden door de grootte van de stroom te delen door de lengte van de periodieke sequentie. Vervolgens kan een stroom als "dun" worden beschouwd als deze ratio onder een bepaalde waarde ligt. Samenvattend kan er gesteld worden dat een dunne stroom, een bepaalde route in het wegennetwerk is waar er maar weinig vraag is naar vervoer op bepaalde tijdsstippen. Zoals reeds duidelijk werd, zijn dit soort stromen problematisch voor OV, aangezien het moeilijk is om dergelijke lijnen rendabel te houden.

Deze thesis is een bundel van verschillende peer-reviewed artikels zowel in proceedings als in journals. De thesis bestaat uit vier delen.

Deel I is het grootste deel en bestaat uit vijf hoofdstukken. In Hoofdstuk 2 wordt de netwerkvoorbereiding voor de simulatie besproken. Er wordt dieper ingegaan op de voorbereiding van *OpenStreetMap (OSM)* en *General Transit Feed Specification (GTFS)*. OSM wordt gebruikt voor het netwerk en GTFS wordt gebruikt om OV te simuleren. GTFS is een dataset die informatie geeft over locaties van OV haltes, de dienstregeling, de routes etc. Aangezien het hier over open-source data gaat, is de kans groot dat de data niet altijd even correct is. Het doel van dit hoofdstuk is dan ook om anomalieën uit de OSM en GTFS dataset te halen en zo nodig om ze te corrigeren. Dit is een belangrijke stap gezien er microscopische simulaties gebruikt zullen worden en deze nood hebben aan zeer gedetailleerde en correcte data. De GTFS dataset zal voornamelijk gebruikt worden als input voor de OV router OpenTripPlanner.

In Hoofdstuk 3 wordt een eerste design aanzet voor de agent-based software besproken. Hier wordt er voor het eerst kennisgemaakt met het agent-based framework SARL dat gebruikt zal worden voor de microscopische simulaties. Ook enkele belangrijke concepten van de simulaties worden hier besproken.

In Hoofdstuk 4 wordt een uitbreiding op het bestaande *agent-based framework* SARL besproken. Aangezien SARL geen mogelijkheid heeft om gesimuleerde tijd te synchroniseren, werd er een conservatieve synchronisatie methode voorgesteld en geïmplementeerd. Dit was nodig omdat in de simulaties, de tijd minuut per minuut moet verder gaan en de communicatie tussen de verschillende vervoersaanbieders chronologisch correct moeten verlopen.

Vervolgens wordt in Hoofdstuk 5 een belangrijke mogelijkheid van de software besproken, namelijk de koppeling met externe APIs. De mogelijkheid tot het integreren van externe APIs verlaagt het programmeerwerk van de programmeur van de applicatie. Complexe algoritmes zoals een Vehicle Routing Problem of een OV router moeten niet opnieuw ontworpen en ontwikkeld worden. Er moet enkel een connectie gemaakt worden tussen de applicatie en de gewenste API. Het gevolg is dat de onderzoeks- en ontwikkelingstijd drastisch zal dalen. In dit hoofdstuk wordt de integratie met het Multi-Agent Transport Simulation (MATSim) besproken.

Hoofdstuk 6 combineert de kennis en technieken, ontdekt in dit deel, om een werkende simulator te ontwikkelen en experimenten uit te voeren. Een sensitiviteitsanalyse wordt uitgevoerd gebaseerd op de tarieven van vervoersaanbieders en de grootte van hun vloot. Deze verschillende experimenten geven inzichten in verschillende metingen zoals (i) het aantal verplaatsingen dat een klant niet kan volbrengen, (ii) de totale inkomsten en uitgaven van een aanbieder en (iii) het totaal aantal aanvragen, afwijzingen, aanvaardingen en voorstellen. Het uiteindelijke doel is om te onderzoeken of vervoersaanbieders in een vraaggestuurde transport context rendabel kunnen zijn gedurende een specifieke periode. Om deze resultaten te kunnen bekomen, zullen klanten en vervoersaanbieders met elkaar de dialoog aangaan om

zo te onderhandelen over aanvragen en de bijhorende voorstellen. Vervoersaanbieders versturen voorstellen als ze de aanvraag kunnen behandelen en wijzen de aanvraag af als ze deze niet kunnen afhandelen. Bijgevolg kan het zijn dat in sommige gevallen, de klanten niet in staat zullen zijn om bepaalde trips te kunnen uitvoeren aangezien er geen aanbieder is die de aanvraag succesvol kan behandelen.
Ten slotte wordt er in Hoofdstuk 7 een kritische reflectie gegeven over deze simulatie.

Deel II bestaat uit één hoofdstuk. Hoofdstuk 8 beschrijft de optimalisatie algoritmes die gebruikt kunnen worden om transport te optimaliseren dat georganiseerd wordt door faciliteiten door gebruik te maken van carpoolen en vraaggestuurd vervoer. Het vervoer bestaat bijgevolg uit (i) vrijwilligers die faciliteitsbezoeker willen ophalen en afzetten bij de faciliteit of een andere gemeenschappelijke locatie en (ii) vraaggestuurde vervoersaanbieders. Door te carpoolen, moeten mogelijks minder faciliteitsbezoekers worden opgehaald door vraaggestuurde vervoersaanbieders of de routes van de vraaggestuurde vervoersaanbieders worden efficiënter aangezien faciliteitsbezoekers samengebracht worden bij een gemeenschappelijke locatie. Het resultaat is dat minder voertuigen gebruikt moeten worden en minder kilometers afgelegd moeten worden en dat er bijgevolg minder kosten zijn. Gegeven een reeks van deelnemers en bestuurders met bijhorende beperkingen zal de software proberen om een optimale oplossing te vinden. Deelnemers die geen deel uitmaken van de carpooloplossing, zullen nog steeds gebracht worden door een betaalde bus.

Deel III bestaat uit één hoofdstuk. Hoofdstuk 9 stelt een methode voor die gebruikt kan worden om data te verzamelen van een zeer specifieke groep mensen. Een stop detectie algoritme wordt geïntroduceerd. Een sensitiviteitsanalyse is uitgevoerd en resultaten zijn vergeleken met een bestaand trip detectie algoritme.

Deel IV beëindigt de thesis. In Hoofdstuk 10 wordt er een overzicht gegeven van mogelijk toekomstig onderzoek. Er wordt ook wat extra materiaal besproken. Ten slotte, wordt er een conclusie getrokken in Hoofdstuk 11.

Bibliografie

Mobiel Vlaanderen. Hoe werkt het? `https://www.mobielvlaanderen.be/basisbereikbaarheid/hoe.php`, 2018. Geraadpleegd: 2018-06-18.

POD Maatschappelijke Integratie. (Equivalent) leefloon. `https://www.mi-is.be/nl/equivalent-leefloon`, 2018. Geraadpleegd: 2018-06-11.

Vlaams Parlement. Ontwerp van decreet tot compensatie van de openbaredienstverplichting tot het vervoer van personen met een handicap of een ernstig beperkte mobiliteit. `http://docs.vlaamsparlement.be/pfile?id=1039564`, Dec 2012. Geraadpleegd: 2018-07-09.