CORE: an intelligent transportation system in Calabria

Francesco Santoro[1]

[1]Department of Mechanical, Energy and Management Engineering, University of Calabria

Contents

Chapter 1

Introduction

Transport is a strong point of Italian and European economy: it generates 7% of EU Gross Domestic Product and creates more than 8 million jobs. However, it is also responsible for more than 20% of CO2 emission in the EU and is one of the principal causes of death: every year there are more than 40 thousand deaths and almost 20 million injuries due to road accidents. Things are not different in Italy. Traffic density in the country is among the highest in Europe, with 937.276 million passengers-km/year, more than 92% of which is road traffic, and a freight traffic size of 239.518 million tons-km/year, with over 65% of it covering road distances measuring more than 50km. Traffic is expected to grow over the next few years, forecasting bad consequences for environment, quality of life, security and, as a direct consequence, for the whole economy. On the other hand, the satisfaction of current and future mobility demands is a mandatory goal to achieve in order to ensure economic development of the poorest areas and to support a higher level of competitiveness for the more developed ones. Adopting a transport strategy promoting even growth of different areas and supporting a sustainable development strategy is a national as well as community priority: in 2001 the General Plan of Transports and Logistics was compiled. One of the main initiatives is represented by the Transport National Operating Program, an instrument to develop the southern transport system. In particular, it has financed strategic infrastructures pointing out that a synergy between environmental sustainability, infrastructures and technology may improve quality of life.

EU Commission [17] pointed out that transport is not sustainable as it is. Making a 40-year projection of the future, the system will be 90% oil dependent and energy made out from renewable sources will constitute 10% of the overall production. In 2050, CO2 emissions will be higher than in 1990 by one-third; costs due to traffic congestions will be higher by 50%, social costs due to accidents and acoustic pollution will also keep growing. Reducing mobility is not an option.

It is necessary to estabilish new transport paradigms in order to carry higher volumes of goods and number of passengers by using more efficient trasport modes or combination of different modes. Individual transport should be discouraged and made only for the last step of a journey with greener vehicles. Future development should rely on:

- Improvement of energy efficiency of vehicles in every transport mode, by using sustainable sources;
- Optimization of multimodal logistics chain's effectiveness, by improving the process of resource exploitation when other innovations may be not sufficient (e.g. long distance freight transport);
- Improvement in the efficiency of infastructure exploitation by applying better information and traffic management systems, advanced logistics and market policies, such as an European integrated market for railway transport, improvement in fare management, etc.

In this scenario, a key role is played by research and technological innovation. Applied research has a dramatic impact on industry and local development. Intelligent Transportation Systems (ITS) have a central role, thanks to the important results they have achieved over the past few years in terms of sustainable mobility. This has been highlighted by the massive investments made at national and community level during the last decade. Within the 7th Framework Program, 4 billion Euros (nearly 13% of the total resources) were allocated with the aim of supporting scientific research intended to develop greener, safer and more intelligent transport systems. This trend has been confirmed by the Horizon 2020 Program. ITS are a fundamental instrument for managing mobility, road safety and development of a sustainable mobility model.

ITS collect, process, manage and exchange data related to vehicles, infastructures' status and users, and integrate them in an intelligent way. This integration is the key factor for facing mobility issues effectively in an organic way, improving security, transport efficiency and quality of life of commuters, and reducing environmental impact. Thus information and management operate in synergy, by optimizing infrastructures, vehicles and resources' exploitation with the aim of improving environmental efficiency and sustainability. Therefore transports become an integrated system thanks to ITS. An EU Commission survey reported that a number of applications made in EU countries have reduced travel time by 20%, increased network capacity by 5-10% and improved security by 10-15%, thanks to the application of pre-existing telematic technologies to transport.

These positive results showcase the advantages generated by ITS and confirm that nowadays they are necessary to achieve mobility related goals. Moreover, ITS are a development opportunity for companies operating in the transport industry and IT systems and thus also boost the growth of a whole country.

In this scenario, Regione Calabria has set up an ITS in order to provide functionalities related to infomobility, transport service management and certification of journeys. C.O.RE. is the software platform produced by a joint project by the University Of Calabria and Regione Calabria and will be presented in this work together with some scientific applications.

1.1 Main Contribution

This thesis contributes in the following topics

- ITS platforms
- Shift scheduling in public transport
- Driving style analysis

Regarding ITS platforms, as described in chapter 3, an information system has been developed from the scratch, based on open source technologies, in the scope of a project carried out jointly by the University of Calabria and Regione Calabria. The mostly used data model for transport systems has been adapted to the local context and a binary protocol has been designed in order to carry out efficiently Automathic Vehicle Location functionalities. Moreover, the main data standard have been studied in 2, in order to understand which one would fit as well as possible to the local context, since no one could be adopted as-is. In the end we decided to use TransXchange data format and we provided some guidelines to Regione Calabria to convert their database to one that is very close to TransXchange. One of the biggest challenges was, in fact, to design standard processes for timetabling and data manipulation. In the field of shift scheduling, a model for Crews and Vehicles Scheduling has been proposed: it takes into account constraints taken from EU and Italian legal frameworks. Heuristic algorithms have been designed in order to solve jointly che Crews and Vehicles Scheduling Problems and have been tested on data taken from CORE and other italian context. The AVL data that are collected by CORE consist in a valuable source for a number of studies related to the driving style. In this thesis these data have been processed and a subset has been selected in order to perform some statistics analysis and find out interesting trends in the drivers' behaviour.

1.2 Outline

This thesis is organized as follows:

- Chapter 2 provides a description of the context of application and a state-of-the-art analysis of its systems and their key role in the development of urban areas;
- Chapter 3 gives an extensive description of CORE project and its results;
- Chapters 4 and 5 present two scientific applications related to the ITS world and falling within the mid-term objectives of the CORE project. Since C.O.RE. is a valuable source of data on planned service as well as on automatic vehicle location applications (AVL [61]), it easily supports scientific studies aiming at improving the quality of the transport system and also its cost effectiveness. In section 4 an integrated algorithm for

Vehicles and Crews Shift Scheduling is presented; in section 5 a study on the behaviour of the drivers is performed, based on the data collected by the AVL system provided by C.O.RE. The former can lead to applications allowing companies to save resources by reducing the number of vehicles and crews needed in order to perform the same number of journeys, and also by reducing dead runs (i.e. journeys made to relocate vehicles and crews to another node of the network). The latter is aimed at implementing applications drivers' behaviour analysis and at providing users with information and suggestions, thus resulting in a greener and safer drive;

- Chapter 6 concludes this work with a summary of the main results and discusses interesting topics for further extending CORE.

Chapter 2

ITS Systems

2.1 Public Transport

A public service is a service provided by a government to the community living under its jurisdiction, either directly (when directly managed by the government) or by financing provision of services, for example by financing private subjects who act as service dealers on behalf of the government. The Communitary legal framework compels the supplier of the public service to provide its service to everybody, without interruptions (universality and continuity principles of the supply), to offer good quality at a reasonable price, to handle security concerns, transparency and to allow users to choose the service, the supplier and the payment mode they prefer. Public transport is a public service consisting of a set of transport means and ways of organization allowing citizens to move across different areas by any mean other than private: bus, taxi, train, etc. The use of public transport (also referred to as PT in the following) systems allow to reduce costs related to transport, air and sound pollution, traffic congestion and risks connected with accidents. On the other hand a citizen is discouraged from using public transport for several reasons, i.e. hygienic, uneven distribution, delays, longer run times due to intermediate stops.

A survey made by the Ministry of Infrastructure and Transport in 2014 [22] points out that private mobility is still the preferred option and that users prefer those means that are more reliable in terms of travel time and have the shortest possible waiting times (see table 2.1). The aggregate high-end score (6 to 10) exceeds 90% for airplanes (95.7%), motorized two-wheel vehicles (94.2%), cars (92.8%) and bikes (91.5%). Metro is just below, having 88.3% of positive marks, and it is followed by high speed trains, rated positively by 84.8% of users. Urban road transport scores an average mark of 6.2 and it is rated positively by 67.7% of users; extra-urban road transport is slightly above, with an average mark of 6.6 and 75.4% positive evaluations. The worst rated mean is local or regional rail transport, having a barely sufficient average score and 63.1% of satisfied users.

Table 2.1: Passenger satisfaction by transport mean

	2013		2014	
Transport mean	6-10 score	Avg score	6-10 score	Avg score
Moto/scooter	95,1%	8,4	94,2%	8,3
Bike	91,8%	8,4	91,5%	8,4
Car	92,1%	8,2	92,8%	8,2
Metro	87,3%	7,6	88,3%	7,6
Local or regional train	62,4%	6,0	63,1%	6,0
High speed train	85,7%	7,4	84,8%	7,3
Urban bus/tram	67,5%	6,2	67,7%	6,2
Extra-urban bus	78,2%	6,6	75,4%	6,6
Plane	94,1%	8,2	95,7%	8,2

Table 2.2: Demand share by means

	2012	2013	2014
Bike or walk	17%	17%	19%
Motorized	83%	83%	81%
Motorbike	*4%*	*4%*	*4%*
Private	*83%*	*83%*	*81%*
Public	*13%*	*13%*	*15%*

The share of demand by means of transportation reflects these results as private means remain the first choice for more than 80% of passengers (see table 2.2)

Moreover the demand for mobility is constantly increasing, being one of the most significant indicators of the development and vitality of a community.

Table 2.3: Mobility demand

	2012	2013	2014
% moving population	75,1	75,4	79,7
average transfers per individual	2,68	2,74	2,83
avg km run	34,7	37,8	33,3

In the last three years, in Italy a lifestyle change has resulted in a 4% increase of the moving population (from 75.4% in 2013 to 79.7% in 2014) and in the volume of transfers: 11.5 million more in 2014 than in 2013. In the year after the 2012 crisis, data are quite promising:

- mobility rate is close to 80% as it used to be from 2006 to 2010;

- average transfers per day per individual have risen to 2.83;

- the average transfer distance in km is down to 33.3: since transfers per individual are not lowering, people are probably moving towards areas closer to their everyday activities.

This trend mirrors what is happening in Europe: freight transport has increased by 31% and passengers transport by 20% in the last 10 years . In high growth rate areas such as China and India, increments are even higher. This leads to energy issues, as well as to an increase in traffic and pollution, with negative effects on quality of life. The equal distribution among travel modes, infrastructure boosting and low environmental impact vehicles seem to be the most important choices facing current mobility problems. Nevertheless, it is not always viable to build new infrastructures, especially in the short-term, for cost-related reasons as well as for the high environmental impact of such structures: Intelligent Transportation System (ITS) are indeed quite promising and effective in coping with mobility problems, at least in the short and mid-term. This is probably because ITS tend to follow a "system" approach, that has been the key factor behind their success in many international experiences: a synergy of information, management and control favouring an optimal use of vehicles and infrastructures from a multimodal perspective. Indeed, ITS, founded on the interaction between Information Technology and mobility, allow to turn the transport system into an integrated system making it possible for it to achieve more efficiency, productivity and security.

Experiences so far show clear evidence that ITS lead to environmental benefits and efficiency, citizen security and improvements in competitiveness, and by now they are necessary to reach sustainable mobility. There are also benefits on production system and employment rate: in Italy, for example, a number of highly technological SMEs (small/medium enterprises) observe a 10% yearly growth in employment [26]. This growth has been encouraged by the development of telecommunication, information technology and electronics, which are the foundation of advanced telematic services aimed at improving mobility resources' exploitation.

2.2 ITS Services

ITS services result in:

- accident reduction;
- travel time reduction;
- more efficiency in intermodal chain;
- lower environmental impact;

These goals are achieved thanks to a large number of services and systems applied to all aspects of the transport field. The following macro-areas are identified:

- payment systems;
- access control systems;
- load/traffic management systems;
- security management systems;

2.2.1 Payment systems

Some Electronic Tolling (electronic toll payment) are used on highways, providing devices on vehicles (OBU) which interact with a stable infrastructure. This infrastructure, capable of detecting the vehicle's passage, automatically triggers the toll payment. Telepass technology is used on the Italian highways. It leads to:

- travel time reduction;
- time consumption (on the toll booth) reduction;
- lower pollution level.

GPS-driven tolling systems represent an innovation in the ITS field. They are implemented with on board tools broadcasting entrance and exit of the vehicle on the highroad to an Operations Center. For this reason, it is no longer mandatory to have an infrastructure for tolling collection. For parking payments an sms (teleparking) can be sent to the service authority: when paid time is almost up, the user is alerted by an sms, so they can opt to extend it without having to reach the car. Otherwise RFID tag cards can be used. Containing data related to the vehicle mobility, they can be attached to the windshield, to allow traffic police to read it by means of a mobile device. Finally there is an electronic payment system, used in PT, which considers the usage of certain tickets like magnetic or smart cards, and e-terminals as validators. Those are introduced to reduce the limitations of traditional tickets, and used for dynamic discounts and pricing, travel and tourist cards, checking-in and checking-out. Another of its advantages is the possibility to use different modes and to travel across different fare zones.

2.2.2 Access control systems

ITS systems have a key role for access management, along with demand control in certain areas by way of using devices like cameras for plates recognition, or electronic payment systems. Limited Traffic Zones (ZTL) are areas where traffic is restricted in some time slots or to certain vehicles, in order to improve security for pedestrians during peak hours, or to lower pollution. Congestion Pricing is an access control policy which records the vehicle entrance in a predetermined area through a toll payment. Its scope is to discourage private vehicle usage for environmental reasons. The first Italian city that adopted this choice is Milan where ZTL access is ruled by a toll. Measured benefits are: 9% CO2 reduction and 11% NOX reduction per year, as well as 3.4% traffic decrease. ITS systems support access control policies thanks to cameras filming the registration plates of those cars that are infringing the rules, or BOA integrated portals which allow to detect authorized OBUs. This strategy is very useful for insurance companies, which can control geolocal data in order to offer new vehicle policies. Drivers are incentivized to reduce their mileage and to adopt an eco-friendly behaviour while driving, which in turn results in lower pollution levels.

2.2.3 Traffic management systems

ITS embedded systems allow to aid commuters in their journeys, improving their perceived service quality. An example could be traffic related data sent via the Internet, vehicle information systems, PMV (Message variant panels). Those tools can be private or shared, either installed on personal devices or shown via screens installed along e.g. on highways or at bus stops. In relation to public transport, these tools are especially useful in providing more accurate infomobility: commuters are informed about timetables as well as potential delays on their routes.

2.2.4 Security management systems

Security management systems include solutions aimed at increasing vehicle supervision, avoiding side or frontal collision, automatizing vehicle operation (i.e. driver/vehicle monitoring system, automatic driving system, etc.) and developing security devices. In some cases, speed limit control systems like Tutor (in Italy) are in place, together with lane control systems which monitor the way a driver behaves with regard to overtakings ([62]). For the ITS purpose a combination of knowledge and sophisticated transport tools are required. Moreover, some data collection, exchange, distribution and analysis is needed, in order to track information on passengers, vehicles and goods on the road. Some critical aspects may obstruct the ITS development in this field, such as obsolescent equipment, different technologies integration, incomplete legislation.

2.3 Economical and environmental impacts of ITS

The smart transport system has undergone a wide expansion. ITS technologies have been developed during a critical time period marked by increasing passenger demand without a proportional growth of the infrastructures. However, investment in the infrastructures themselves would not be enough to satisfy the increasing demand. Therefore, ITS technologies represent an effective tool to optimize the management of preexisting infrastractures. Europe, USA and Japan already experienced the benefits of ITS in the fields of safety, efficiency, environmental impact and overall productivity of the transport system. Investigations by the European Commission following ITS spread showed a decrease in travel time, energy consumption and pollution levels of the road network. Additionally, the research reported a reduction in number of car crashes, an improvement in the economy, productivity and quality of service in freight logistics and public transportation. ITS costs include investment, set-up, maintenance, coordination and vocational education. Furthermore, they depend on their temporal application and on the technologies applied, so they can be variable. The ITS system can be divided in 3 levels, each of which with its own expenses:

- *Information harvesting*: expenses for detectors, interfaces creation, softwares, infrastructure, etc.;
- : *Communication*: expenses for telecommunication and supplementary devices for the integration;
- Information processing: expenses for information systems, knowledge systems, software and hardware components etc.

The above mentioned costs are lower than the expected benefits in terms of energy saving, emission reduction and social outcomes. More specifically, very often the benefit related to the environmental impact are underestimated. Cooperative systems (such as Vehicle2Vehicle and Vehicle2Infrastructure ([50]))

tend to lower CO_2 emission as well. There exist different approaches promoting eco-friendly transportation such as traffic control systems, speed regulators aiming at minimizing energy and reducing the stop-and-go. These strategies use real time vehicle localization and real time route planning.

Drivers can be sensitised to save fuel and adopt an efficient driving behaviour by means of eco-friendly promotional campaigns. More specifically, drivers are pushed towards an eco-friendly attitude thanks to several functionalities provided by the vehicle and/or by internet-based applications. By having the right driving style one can save up to 10% of fuel, an improvement from both an economical and environmental point of view. This is about teaching drivers which gear is most suitable considering the engine RPMs and helping them to avoid busy areas. They will be offered a series of alternative routes they can keep a constant speed on, prompted to turn off their vehicles when stopping, and so on. Users' navigation systems can be managed in a centric way as they should suggest different paths to different users (multipath methods, [42]), thus reducing global congestion. Even those users not using navigators will perceive advantages.

2.4 ITS in Italy

To face the challenges related to the continuous increment in mobility demand, which in itself is a trend similar to other European countries, the Ministry of Infastructure and Transport has pointed out the need to "think the transport system" in a new way, as a whole system where information, management and control operates in synergy. ITS in Italy have been active sector since the eighties and had a dramatic development in the 1990s, on par with the most industrialized countries. Central and local administration, companies, research institutes, universities and public and private network managers all took part in the Research and Development Framework Program by the European Commission, with significant results. Traffic and mobility management systems are in place in a number of Italian cities, such as Rome, Turin, Milan, Florence, Bologna, Genoa, Perugia, Naples, Brescia, Salerno. Moreover, almost 50% of PT companies employ tracking systems, aimed at improving the offer.

A recent survey ([56]) made in 2014 within the Infocity Project of Elisa Program, financed by the Regional Affair Ministry, reveals that a wide number of local authorities have adopted a mobility plan which includes ITS and has already started (or completed or planned for the next 3 years) actions related to traffic and infrastructures management, local public transport, information to the commuters, road pricing, electronic ticketing and fare integration, freight transport and road security.

At a regional level, as agreed in May 2007 by Government, Regions and autonomous Provinces, ANCI, UPI and UNCEM in the Unified Conference, many Regions adopted Regional Plans of Infomobility, that is a strategic document related to actions aimed at promoting the development and sustainability of innovative processes applied to public and private mobility systems. There are a number of projects promoted in different locations placed on the strategic corridors of freight transport with the aim of supporting the development of intermodality and integrated logistics. In relation to this last sector, among the main national initiatives financed by the Ministry of Infastructures and Transports, the UIRNET project is worth of mention: a telematic platform built with the purpose of improving efficiency and security of the entire national logistics system, with important advantages both for single users and the whole system. This platform offers a series of services in matters of mobility and interoperability information as well as specific services for the transport of dangerous goods. Finally, worthy of mention are the ITS projects financed by program PON (National Operation Program - Programma Operativo Nazionale in Italian) Transport 2000-2006 and 2007-2013 that have been or are in the course of being carried out in the Objective 1 Regions (Basilicata, Calabria, Campania, Apulia, Sardinia, Sicily), the ITS projects promoted by the Elisa Program and financed by the Ministry of Regional Affairs and also those projects financed by the Industry Sustainable Mobility Program 2015.

In terms of infrastructures, Italian tollways are a natural environment of experimentation and application of innovative systems and technologies. The toll network has been created with the aim of efficiently connecting areas of great importance, both from an economic and social point of view, characterized by heavy traffic flows Thus it has been necessary to have in place advanced management and monitoring systems, as well as information systems for users and for toll collection. For decades, Italian tollways have been front and center of ITS and traffic management projects. In particular, the Telepass system for automatic toll collection is an Italian excellence now also adopted in other European Countries, with advantages to the national industry. Lastly it is important to notice that in March 2003 the Ministry of Infrastructures and Transports has published version 1 of Italian Telematic Architecture for Transport System (ARTIST - ARchitettura Telematica Italiana per il Sistema dei Trasporti in Italian). ARTIST's objective is to establish guidelines for making ITS application compatible, integrable and interoperable with each other. ARTIST's architecture has been modelled after KAREN European Architecture to ensure interoperability between what ARTIST propose and what is currently in use in the rest of Europe.

2.4.1 Legal framework

The most relevant juridical initiative regarding the development and adoption of ITS in Italy is Decree Law 18 Oct 2012 n. 179 *Other urgent measures for the growth of the country* (*Ulteriori misure urgenti per la crescita del paese* in italian), also known as Development Decree Bis, converted, with amendments, into Law 17 Dec 2012 n. 211. Art. 8 of this Decree Law, *Measures for the innovation of transport system*, adopts European Directive 2010/40/EU *on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport* ([53]) and stresses two main points:

- the need of promoting the adoption of an electronic payment system which should be interoperable at national level. It estabilishes a term of 90 days to issue the necessary technical rules for the public transport companies to support the adoption of interoperable electronic payment system, in accordance with existing solutions;
- the need of proprietary subjects, managers of infrastructures, parking and service areas, intermodal hubs, to set up a database of their respective infastructure and/or service and to keep it constantly updated.

Furthermore, art. 8 tightly regulates the terms of implementation of the 2010/65/EU Directive of 20 Oct 2010, in particular with regards to declaration of ship arrival and departure from the ports of the Member States, which should be managed by means of SafeSeaNet system, the EU system for maritime data exchange, or by means of PMIS (Port Management Information System), that is an information system for the administration of port activities. Ministerial Decree n. 39 of 1 February 2013, published on GURI on 26 March 2013 n. 72, completes the national legal framework in terms of development of ITS systems. Interministerial Decree 446/14 identifies the Ministry of Infrastructures and Transports as the national authority entrusted with the adoption of "National Plan of ITS System Development" as well as the related communications with the European Commission.

2.5 Data format Standards

Interoperability is defined as the capability of a system to cooperate and exchange data or services with other systems or products with reliability and resource optimization. This characteristic guarantees the correct communication between heterogeneous systems, so the need of a standardization arises. Knowledge representation is one of the main aspects linked with standardization because it allows a system to be flexible enough to fully represent the context in which it operates. In this section some of the data formats which have become best practices (although they are not ISO standards) will be presented. Particular interest will be paid to the TransXChange format as it was the reference used for the implementation of the data model of the ITS system proposed.

2.5.1 GTFS

GTFS (General Transit Feed Specification) is a common data format from Google. It is used to represent public transport information and to publish it on Google so developers can create applications which make use of it.

A GTFS feed contains information related to:

- stop point;
- routes;

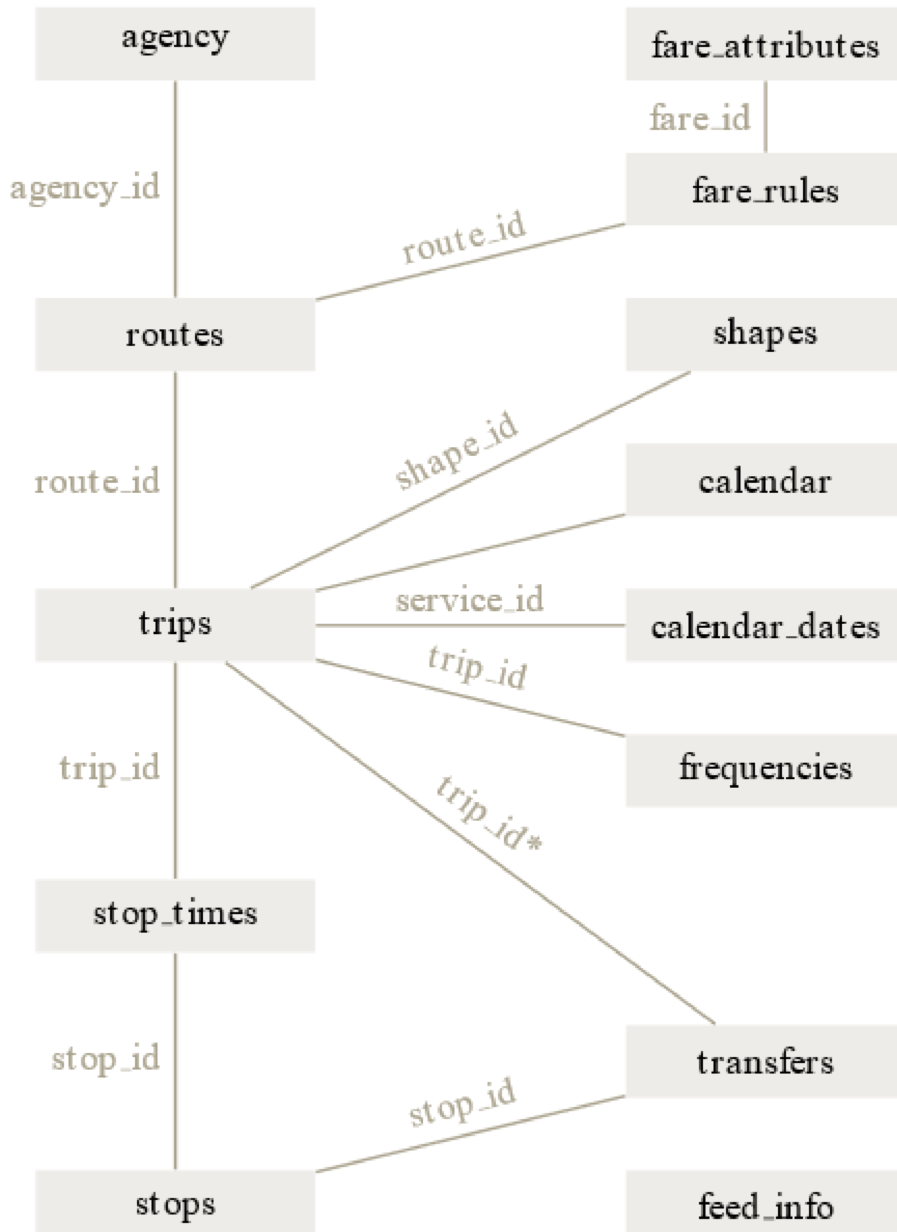


Figure 2.1: GTFS structure

- timetables;
- service information.

There are currently two specifications: GTFS static, used to represent static transport data, and GTFS realtime, an extention of the former which is used to represent real time transit data. In this work, only the former one has been analyzed. As said, a GTFS feed is a set of files whose relationships are shown in 2.1. Here follows a description of the required files of a feed:

- agency.txt: contains the description on one or more agencies (i.e. public transport companies), including name, website and contact information;
- routes.txt: identifies distinct routes;
- trips.txt: indentifies distinct trips. A trip is an instance of a route characterized by timing information;
- stop_times.txt: contains the timing information (arrival and departure) at each stop point for each trip;
- stops.txt: contains the geographic location of each stop in the transit system as well as some amenities associated with those stops;
- calendar.txt: defines the service pattern associated with a trip such as, for example, which days of the week the trip operates and which period (from date to date).

Beyond the above mentioned files, there is a set of optional files which adds more detail and allow to express some more information and rules:

- calendar_dates.txt: lists all the exceptional days when a trip is not available or when a trip operates. This can be used in combination with calendar.txt to express complex operating profiles:
- fare_attribute.txt: contains the fare information for routes;
- fare_rules.txt: specifies rules for applying fare to routes;
- shapes.txt: contains rules for drawing lines on a map to represent routes;
- frequencies.txt: allows to express frequency schemas for routes;
- transfers.txt: contains rules to make transfers between points in order to optimize connection between journeys;
- feed_info.txt: contains metadata on the feed itself, such as publisher, version and expiration information.

The static GTFS has been object of deep analysis because the ITS system implemented within the present work let the user export data into GTFS file format in order to be compliant with Google.

2.5.2 Transmodel

Transmodel (formally CEN TC278, Reference Data Model For Public Transport, EN12896) is the CEN European Reference Data Model for Public Transport Information. It provides an abstract model of common public transport concepts that can be used in many scenarios to build many different kinds of public transport information system, dealing with the following aspects: timetabling, fares, operational management, real time data, journey planning, etc. This is with no doubt the reference standard mostly used all over Europe and it is promoted also by UIP (International Association of Public Transport - UITP, from the French: LUnion internationale des transports publics) and by municipalities (e.g. BIP project in Regione Piemonte [11]) and by the big players in the market of public transport information systems.

Transmodel defines a set of $POINTS$ as the base elements for describing network. A point is the smallest entity on a network and may have one or more types, such as:

- stop-point: that is a bus stop on the network;
- timing-point: a point associated with timing information, used to calculate timetables;

Several points may be clustered according to different principles (tipically spatial) in order to create $STOPAREAS$, which are particularly useful while building infomobility applications or journey planners. Moreover the specification allows to deal with GDF (geographic data files), such as shapefiles. Between two points a $LINK$ may be estabilished that is an oriented edge associated with spatial information (distance between the connected points). A $LINK$ connecting two timing-points is called $TIMING-LINK$ and it is associated with the run time information. Entities representing the service are created by using these base concepts:

- Journey Pattern: specify the route template of the actual journeys and are composed by set of links;
- Vehicle Journey: consist of a particular instance of their relevant journey pattern and are characterized by a start time. This is used in combination with links' run times in order to calculate time tables.

The above mentioned entities will be further explored in 2.5.4 which describes in detail TransXchange data format: a format deriving from Transmodel defining an XML based model for exchanging public transport data.

2.5.3 IFOPT

IFOPT (Identification of Fixed Objects in Public Transport) is a Technical Specification made by CEN that provides a Reference Data Model for describing the main fixed objects required for public access to Public transport. It allows to describe hubs (such as airports, stations, bus stops, etc.) as well as their characteristics, internal spaces, amenities.

Such a model is a fundamental component of the modern Public transport information systems needed both to deliver Public transport and to inform passengers about services. IFOPT identify four components: Stop Place, Point of Interest, Gazetteer/Topographical and Administrative Area. Each component is described in a sub-model with relations to other sub-models, and each object is related to a geographic position.

- Stop Place sub-model: is defined in order to manage all stops from the simple stop in rural areas to a complex urban multi-modal terminal, with several transport modes on different levels. The sub-model also includes paths to describe traveller navigation options and each object can have accessibility information;
- Gazetteer/Topographical sub-model: describes the overall geographic model for countries and cities, down to small residential and parish areas;
- Point of interest sub-model: is an address or a point where the trip starts or ends. The sub-model includes complex objects as tourist attractions, parks and governments facilities such as police stations, hospitals and schools, with multiple entrances and levels;
- Administrative Area sub-model: describes administrative responsibilities for the objects.

Two additional sub-models Parking and Address, are added in order to support multi-modal transport.

2.5.4 TransXchange

TransXChange (TXC in the following) is a UK national XML based data standard for the interchange of bus route and timetable information among those who are involved in the provision of passenger information. It is based on Transmodel and although it is currently used mainly to exchange bus timetables, it may also be used for schedules for rail and other modes. A TXC document contains information related to:

- Public transport planned service;
- Presentation of the data to the users

The main entities are the following:

- Service: describes a transport service. This is delivered on certain territory by following different paths. A service may be of two kinds:
  - Standard Service: composed by fixed routes;
  - Flexible Service: composed by routes that are not fixed (e.g. transport on demand).

- Route: describes a path as a sorted sequence of arcs connecting stops. Each arc is described by RouteLink entity;
- StopPoint: describes a stop point on the network. Each RouteLink connects two StopPoint entities;
- Track: describes a list of geographic points. It is used in order to connect the TXC based model with GDFs;
- JourneyPattern: describes the abstract schema of a bus route as an ordered sequence of arcs. These arcs are described by JourneyPatternTimingLink entity, as they are characterized by run times;
- VehicleJourney: it is an instance of JourneyPattern associated with a departure time. Defines a single route. It is defined by an ordered list of VehicleJourneyTimingLink objects which corresponds to the JourneyPatternTimingLinks composing the relevant JourneyPattern and overwrite their properties if needed;
- Line: it is the public name the transport service it is known by. A transport service may have one or more lines. Each VehicleJourney belongs to a Line;
- Operating Profile: describes the operating pattern of a service: calendars, days of operation, days of non operation, etc.;
- Validity Period: it is the period during which the service is valid;
- Operator: the transport company delivering transport service.

In the following sections further details about the main entities of the TXC model will be provided.

Service description

The basic entity for describing and modelling the public transport service is Service. It can contain two kinds of services:

- Standard Service: it is composed by a set of journeys which follow a predefined path;
- Flexible Service: describes a transport service which has not any fixed route, but it operates within an area and targets a specific market.

In the following sections details on standard service only will be given, as this model has been used as the starting point for implementing C.O.RE data model.

TP data model

Data model related to the public transport network and to the journeys is described by the following entities:

- Route: it is described as a sequence of arcs (RouteLink) connecting two stop points. Stop points are identified by geographic points, while arcs are described by sequence of mapping points which are not related to any particular reference system but can be projected onto different coordinate systems by using Track elements;
- Track: they represent the route shape in terms of spatial coordinates;
- RouteLink: represent arcs on a route; they can be grouped by RouteSection elements so that the same sequence of RouteLinks may be used in different journeys;
- JourneyPattern: it represents the base schema for bus journeys as a sequence of links associated with timing information. Each link is represented by a JourneyPatternTimingLink entity which contains the distance between the points it connects and the run time. Each stop at the ends of a JourneyPatternTimingLink may be associated with information related to the actions undertaken at that stop (JourneyPatternStopUsage): pickup, setdown, etc. JourneyPatternTimingLinks may be grouped into JourneyPatternSection so that whole sequences may be reused into more than one JourneyPattern;
- VehicleJourney: represents a bus journey as a instance of the JourneyPattern it is associated with. Each VehicleJourney must follows the same stops of its JourneyPattern and may overwrite timing properties by using VehicleJourneyTimingLink entities. Each VehicleJourney has an absolute start time that may be combined with the information stored into JourneyPatternTimingLinks/VehicleJourneyTimingLinks in order to calculate timetables;

The above mentioned entities are related to each other according to a flexible abstraction schema which allows the reuse of many parts, thus reducing the memory space needed and redundancies.

More in detail, it is a three-layer representation (fig. 2.2):

- Network layout schema: Route, RouteSection, RouteLink, Track;
- Journey template schema: JourneyPattern, JourneyPatternSection, JourneyPatternTimingLink, JourneyPatternStopUsage;
- Journey schema: VehicleJourney, VehicleJourneyTimingLink.

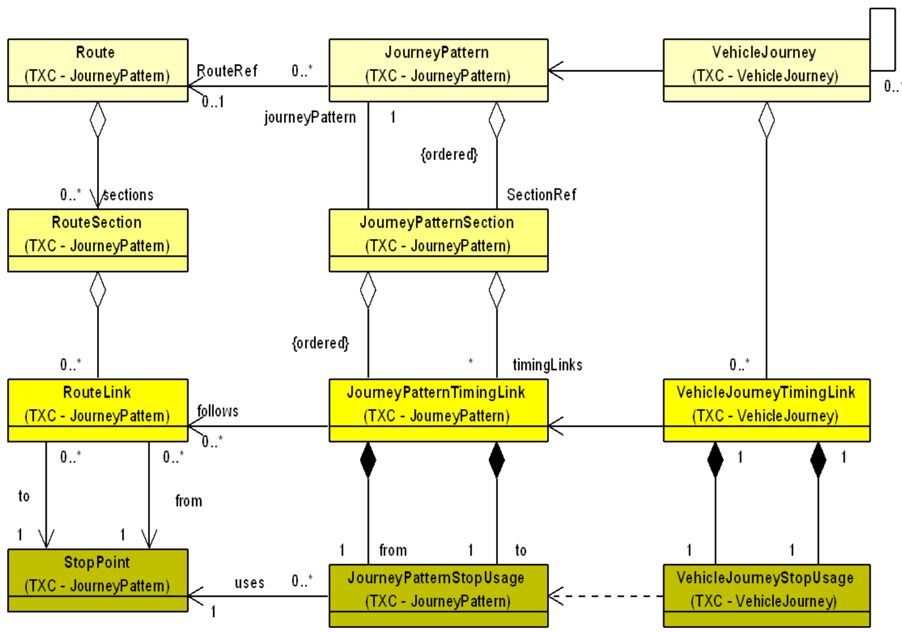


Figure 2.2: TransXchange basic data model

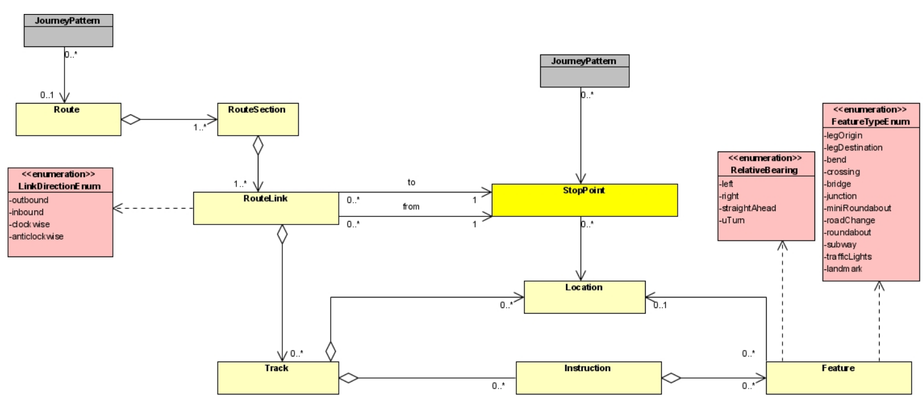


Figure 2.3: TransXchange route data model

Route model

Route entity (fig. 2.3) represents a path as an ordered sequence of RouteLink arcs. Each RouteLink describes a link between two StopPoints. They can be grouped by RouteSections, so that the same set of RouteLinks can be reused in different Routes. Track elements offer details about physical path followd by a RouteLink and are made by:

- Mapping element: the set of spatial points needed to project the RouteLink on a map;
- TrackInstructionElement: optional data used to describe the step-by-step path of the RouteLink (e.g. *turn right after the roundabout in Mazzini street*).

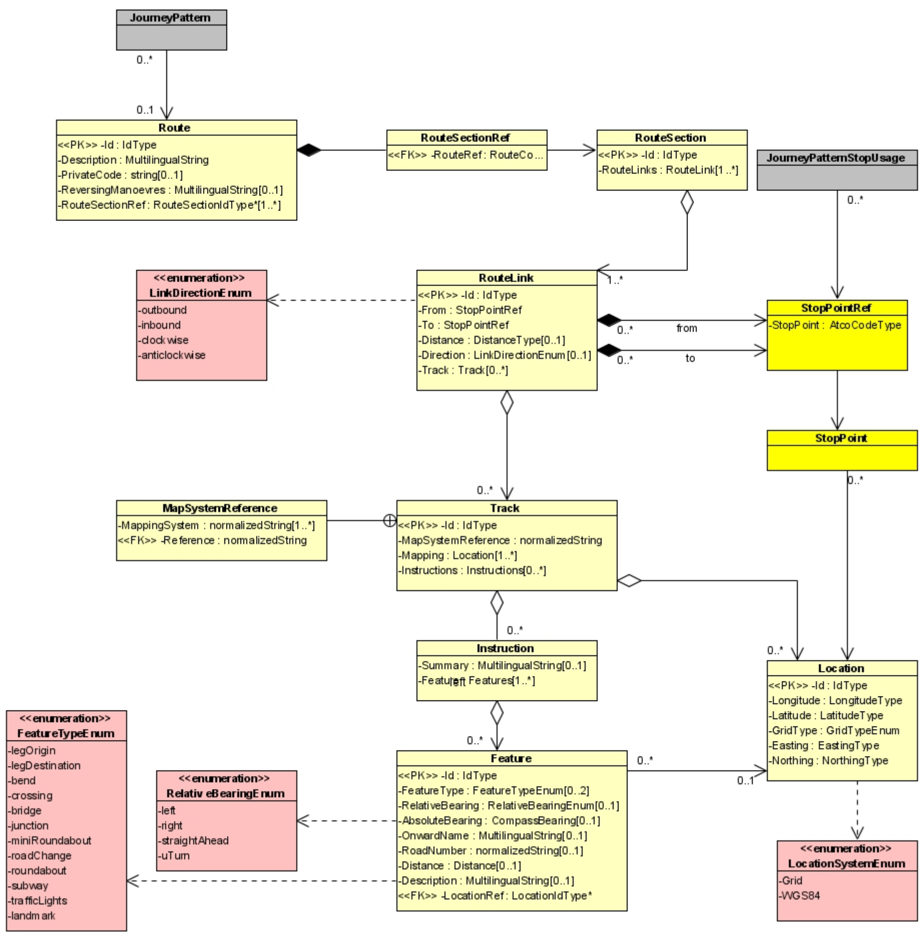


Figure 2.4: TransXchange detailed route data model

UML diagram in fig. 2.4 shows the Route releted hierarchy in detail. Here follows a detailed description of the main entities. Route is a container entity for a path. Its main attributes are:

- ID: unique identifier;
- PrivateCode: a code used for external system exchange;
- RouteSections: a list of RouteSection elements which compose the path;
- ReversingManeuvers: describes eventual reversing maneuvers that are needed along the path.

RouteSection is a section of a route. It is modelled as an ordered list of RouteLink elements and can be reused in different Routes. Each RouteLink describes the path between two StopPoints, in terms of distance, direction (i.e. inbound, outbound, etc.) and the actual road edges to be run. Road edges are modelled by one or more Track entities, defined as a set of geospatial points (MappingPoints) and a set of instructions, which can be free text or structured. Each Instruction entity refers to one or more Features, that are directly related to a road edge. Relevant fields are the following:

- LocationRef: reference to a geospatial file feature;
- FeatureType: type of the feature (e.g. bridge, crossing, junction, etc.);
- RelativeBearing: the way to go to the next step relative to the current one (e.g. right, left, etc.);
- AbsoluteBearing: absolute bearing to go to the next step;
- OnwardName: name of the street associated with the next step;
- RoadNumber: number of the street associated with the next step;
- Distance: distance between the current and next feature;
- Description: text description of the current step.

Each georeferenced point is modelled by a Location object. This can be described by using one or both following modes:

- Grid coordinates: coordinates are easting and northing of a defined reference system;
- WGS84 coordinates: coordinates are expresses as WGS84 degrees.

StopPoint model

TransXchange deals with stop points by means of the following entities:

- StopPoint: used to define a stop point;
- StopArea: used to group a set of StopPoint. StopAreas are useful in those areas where a cluster of points is present but the user does not know (or, equally, he is not interested in) any particular stop point, but he just knows the area by its name;
- Locality: represents a place, i.e. city or common;
- AdministrativeArea: represents the organization which is responsible to manage data related to the stop point.

TXT allows a double declaration for stop point:

- Local, by means of StopPoint entity;
- Global, by using any stop point defined into the NAPTAN database (the national dataset of public transport access points, e.g. bus stops, rail stations etc.).

It's worth to say that in the current work NAPTAN ([49]) is not object of study as in the ITS implemented and described in chapter 3 does not rely on any preexistent point database. As it will be shown in section 3.3, one of the main challenging tasks was to collect a stop point data and set up a database.

StopPoint model UML is shown in fig. 2.5; the gray entities are associated with NAPTAN system and thus are out of the scope of the present work.

JourneyPattern model

In the abstraction hierarchy of route model, JourneyPattern is the middle level element, which describes a bus route template.

Other relevant elements are the following:

- JourneyPatternSection: as for RouteSection, it is composed by an ordered sequence of object describing a link between two stops, named JourneyPatternTimingLinks;
- JourneyPatternTimingLink: describes an arc, i.e. a link between two stop points, associated with spatial and temporal information;
- JourneyPatternStopUsage: defines the access mode of a specific stop (e.g. pickup, setdown, etc.).

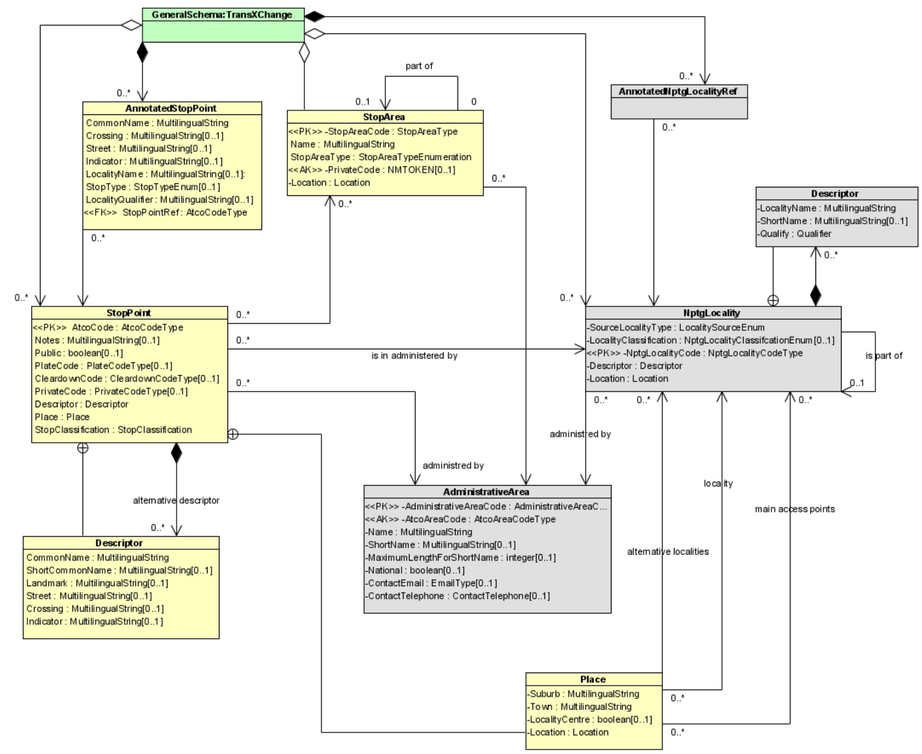


Figure 2.5: TransXchange detailed stop point data model

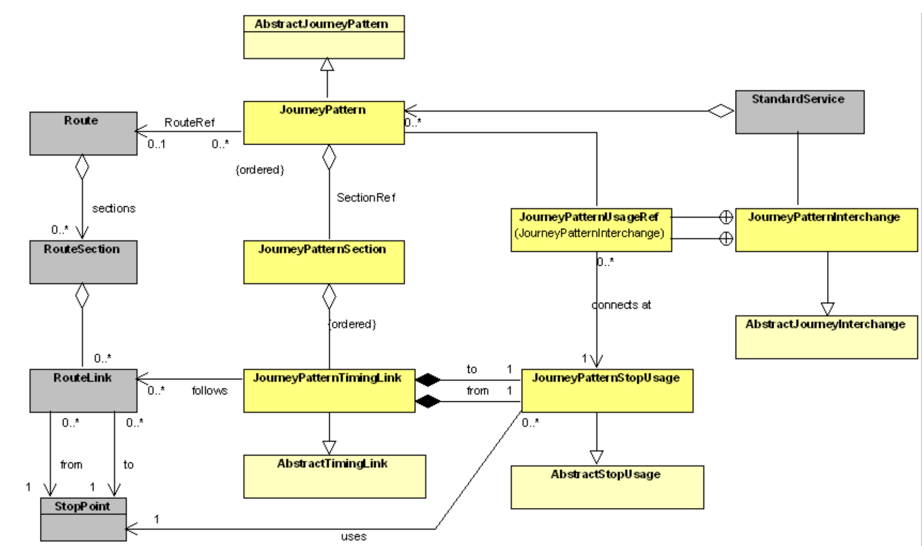


Figure 2.6: TransXchange journey pattern data model

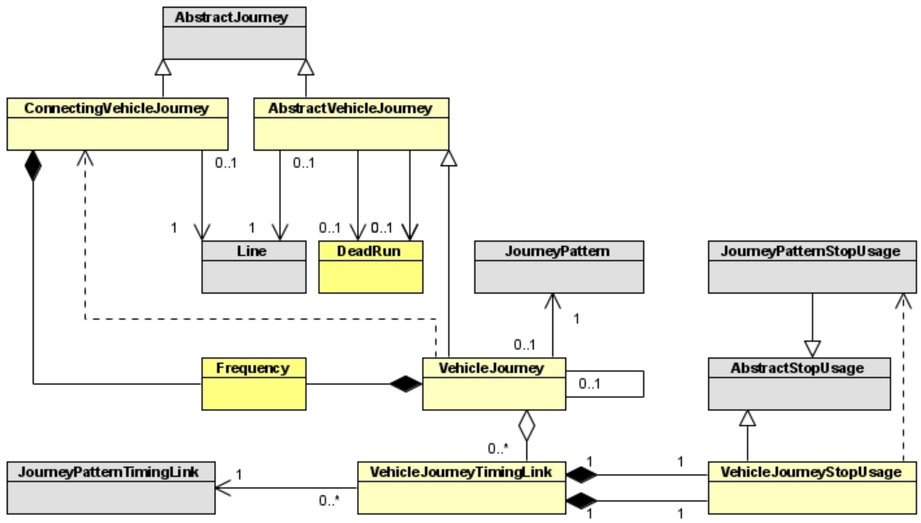


Figure 2.7: TransXchange vehicle journey data model

VehicleJourney model

VehicleJourney schema models service data related to the journeys which are based on the patterns described by JourneyPattern schema. Each VehicleJourney has an absolute start time (e.g. "13:02") which can be combined with the temporal information stored into each JourneyPatternTimingLink / VehicleJourneyTimingLink in order to calculate arrival and departure time for each stop and thus infer timetables. The order of stops traversal must be the same as defined into the JourneyPattern which this VehicleJourney references to. Equally, each VehicleJourneyTimingLink must correspond to a JourneyPatternTimingLink and thus inherits its stop usage information. Stop usage information may be redefined locally at a VehicleJourneyTimingLink, thus timetables will be computed by using, for each link, the most specific information.

Periodic services

Frequency entity allows the definition of periodic services (see fig. 2.8). Regular and irregular services may be defined through this model. EndTime attribute is the time after which frequency information are no longer valid (i.e. a journey runs every X minutes until 8 o' clock). MinutesPastHour, Interval and PartialFrequency are intended to be used in mutual exclusion, according to the following rules:

- MinutePastHour is an integer array: it means a service which repeats with regular frequency at the minutes specified into the array (which can be at most 60 elements long);

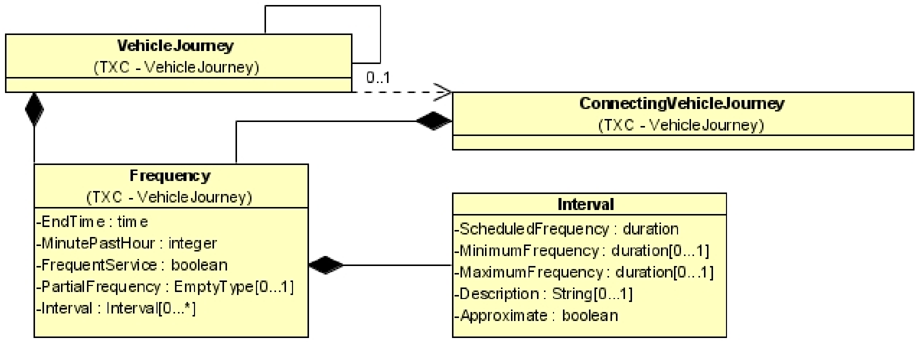


Figure 2.8: TransXchange frequency data model

- Interval describes a frequent service with irregular periodicity;
- PartialFrequency describes a frequent service just for some sections.

Chapter 3

ITS in Calabria: CORE

3.1 What is C.O.RE.?

One of the most critical needs of Transport Operators has to do with service monitoring and the improvement of service quality as perceived by the end users. The growing interest in location-aware devices allows the development of applications that enable the transport operator to track and monitor its services as well as providing more and more accurate information to their customers. C.O.RE. (REgional Operating Center - referred to as CORE from now on) is a web platform for infomobility and sustainable transport system created by the University of Calabria together with the Transport Department of Regione Calabria (ITALY). CORE supports the processes of the Transport Department and facilitates the collaboration with the transport operators in charge, on behalf of the Department, to deliver the transport service. Moreover it allows the monitoring and certification of all the journeys run by each operator and provides infomobility functionalities to the end users, such as a trip planner and departure screens. All transport related data can be manipulated inside the web platform itself by those users with appropriate privileges, so that all data remain consistent and the processes related to the service planning and designing are completely dematerialized. The system is currently deployed in the cloud and it has been recognized by the Regione Calabria as a critical improvement in the overall Transport System: it now comprises all the stop points located in the whole region and the intercity service, for a total amount of 624 lines and more than 1800 buses.

3.1.1 Context

Regione Calabria has signed an agreement with the University of Calabria aiming to build a system for the Public Transport management. This initiative falls into the objectives of the Regional Observatory of Mobility, instituted by Regional Law no. 23/1999 (art 7). The general aim is to bring PT services closer to citizens' and tourists' needs, to make them more effective and accessible, and boost local territory development. PT is currently delivered by 28 companies on behalf of the authority, which is the Transport Department of Regione Calabria. These companies are grouped into 6 consortia as of Regional Law no. 18/2006:

- A.D.M. - Autoservizi dei Due Mari;
- Co.Me.Tra. - Consorzio Meridionale Trasporti;
- Tr.In.Cal. - Trasporti Integrati Calabresi;
- T.R.C. - Trasporti Regionali Calabresi;
- S.C.A.R. - Societ Consortile Autolinee Reginoali
- Consorzio Autolinee Due

It's worth noticing that a number of smaller companies have been adjoined to others. The situation depicted in 3.1 is the most updated one as of 2016. The total amount of km run per year is around 59.2 million, 47.2 of which related to extra-urban services and 12 to urban services.

Each company may deliver urban, extra-urban or mixed public transport service. The breakdown is in table 3.2.

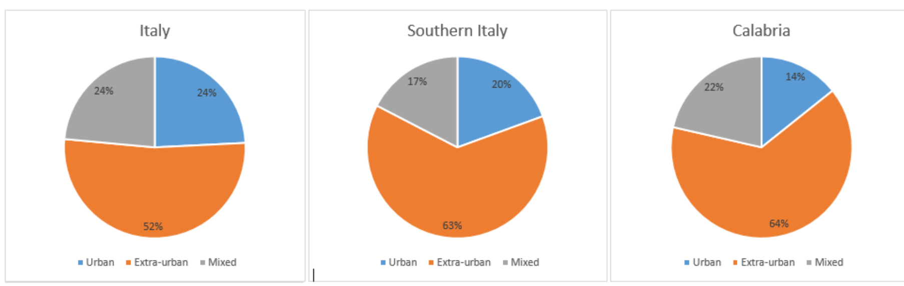


Figure 3.1: Companies by service type

As observed by the Italian Institute of Statistics in 2015 ([25]), the percentage of people who resort to public mobility ranges between 10% and 13%, resulting in low customer retention as well as low customer satisfaction. The average customer satisfaction is dramatically different between North and South of Italy, ranging between 40 and 70% for the former and 30 and 60% for the latter. Calabria, Apulia and Lazio are where the lowest satisfaction levels have been recorded. The most critical aspects are the cost of tickets, comfort at the stops and vehicle cleanliness.

The overall cost for the whole Public Transport System is higher than 22 million euros, being one of the most expensive in Italy. For each company, revenues consist in:

- funding from Authority;
- tickets

Table 3.1: Consortia and companies delivering PT service

Consortium	Company
A.D.M.	Preite
	Bilotta
	Scura
	Zanfini
	Romano
Co.Me.Tra.	A.M.C.
	A.M.A.Co.
	Brosio
	Ferrovie Della Calabria
	Fersav
	G.B.V.
	Ferloc
	Multiservizi Lamezia Terme
Trincal	Piana Palmi Multiservizi
	A.T.A.M.
T.R.C.	Perrone
	P.R.A.
	F.A.T.A. S.R.L.
	Genco
	T.N.C.
	S.A.T.
S.C.A.R.	Mediterranea Bus
	Federico
	Tripodi
	Costa Viola Bus
	Lirosi
	S.C.A.R. S.R.L.
Consorzio Autolinee Due	Consorzio Autolinee
	S.A.J.

Table 3.2: Companies by service type

Area	Total	Urban	Extra-urban	Mixed
Calabria	28	4	8	7
Southern Italy	540	105	341	94
Italy	997	248	535	241

The average cost for each km (urban and extra-urban) is 4.10. The total income from tickets is:

- 49.3% of the overall cost, for urban transport;
- 72% of the overall cost, for extra-urban transport

At the end of each quarter the Authority pays all the Consortia up front for the next quarter's service, minus a 2% deduction that is then released after completion of the monitoring processes.

3.1.2 Improvement opportunities

During the past 3 years, local road PT system has been analyzed and the following issues or inefficiencies have been detected:

- lack of a consistent source of information;
- lack of an information system;
- lack of a control system;
- lack of modern infomobility systems.

Regarding the source of information, there wasn't any database containing stop points or line related information. Stops were only known to individual companies and there was no central database or geocoded information. Office processes were often carried out manually or by using tools such as office automation softwares resulting in a set of files manually aligned, prone to errors and often inconsistent. There was no tool to track the fleet of vehicles servicing public transport: the only way was to monitor stop points at the expected passing times or to rely on users complaints. Moreover, infomobility was a task performed independently by each operator (website, flyers, ...) but there was no centralized or unified infomobility system. The overall transport system was thus easily accessible only to those who were aware of the services and/or the territory.

In this scenario the need of an information system has arisen.

3.2 Architecture

CORE is now the standard public transport control system that has been in production for 2 years and allows to monitor all the vehicles delivering public transport in the Regione Calabria.

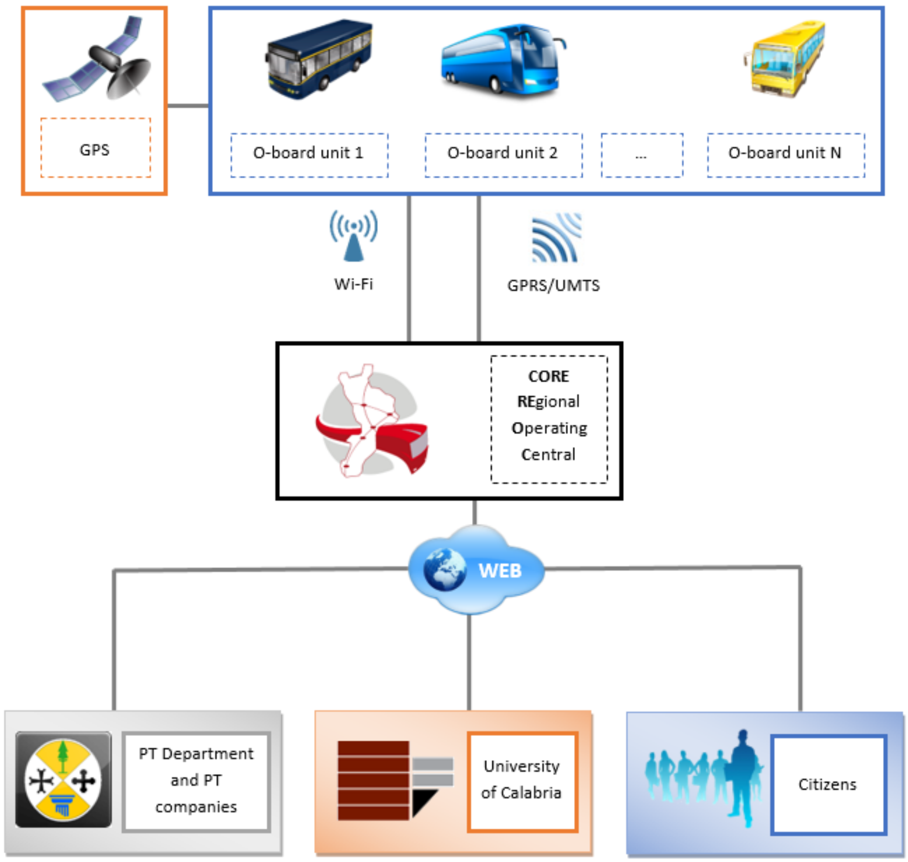


Figure 3.2: CORE - logical architecture

As shown in figure 3.2, the overall architecture can be broken down into three layers:

- Vehicle layer: an on board unit is deployed on each vehicle and it is endowed with GPS and GSM modules;
- User layer: consisting of a web portal addressing providing different functionalities (ranging from infomobility to certification of the service delivered) to different user levels;
- Application Programming Interface (API) layer: consisting of a set of

APIs that developers can interact with in order to build their own applications leveraging the database provided by C.O.RE. At the moment it only delivers static information (i.e. timetables).

The software architecture is shown in figure 3.3.

Figure 3.3: CORE - softwawre architecture

The vehicle layer is proprietary for each company providing public transport and its only required to comply with a communication protocol between each vehicle and the AVL server. Such protocol has been designed with the aim to be as much lightweight as possible in terms of bandwidth and connection frequency. Each data frame is 42 bytes long and its stored locally at the vehicle until a connection does not succeed. Connections are secured via HTTPS with mutual authentication and occur once per minute: thus more than one data frame is transferred in a row. The AVL server, built on top of JBoss 7, keeps track of the status of each vehicle and runs a certification algorithm as soon as a vehicle journey is closed. The web platform is built on top of Liferay CE 6.1, deployed on JBoss 7. A number of portlet (JSR 286) have been implemented in Vaadin 6.7 and deployed onto Liferay. The data layer is based on MySQL CE and its accessed via stateless EJB3 modules and JPA/Hibernate framework. The data format is derived from the TransXChange specification (schema version 2.4) and a number of optimization have been applied in order to speed up queries. The same data access layer is used by the SOA, built on top of JAX-RS/Resteasy, in order to provide the same information on the API channel.

3.2.1 On Board systems

On board systems are out of scope from the CORE project and thus apart from what the University of Calabria has jointly performed with the Regione Calabria. A specification has instead been released and each transport company was expected to buy a number of on board units complying with that specification. The on-board system (SBV - Sistema di Bordo Veicolo, in italian) consists of a set of hardware and software components installed on board of each vehicles composing the fleet. SBVs must send tracking data to the CORE and interact with the driver with the aim of delivering information related to the service. Here follow the main requirements:

- Real-time communication with CORE (for a detailed specification of the way the SBV interacts with the CORE, please refer to 3.5.2);
- Degraded mode: if GPRS signal is down, the SBV must cache data and send it at the first successful connection;
- Dead Reckoning: the SBV must be able to reconstruct position if GPS signal is temporarly down;
- Configurability: user must be able to change system parameters;
- Driver authentication: driver is authenticated via badge or other interface;
- Alarms (e.g. speed excess or heavy delay) to the driver;
- Stop detection: detection of the stop the vehicle is currently passing by.

Hardware requirements are specified in table 3.3

Besides these functionalities, one of the objectives of the project is to set up a hardware environment on board of vehicles which is capable of carrying out GPS tracking as well as other eventual functionalities, such as:

- Transport on demand: SBV must be easily extended in order to provide transport on demand functionalities, such as:
  - route display: the complete route with details related to each stop;
  - information related to the passengers;
  - information related to the fares;
- Fare management: SBV must be easily expandable in order to be connected to a ticket validation system and to emit tickets.

Table 3.3: SBV Hardware Requirement

CPU	CPU 400 Mhz
RAM	128 MB
Memory expansion	1 SLOT Secure Digital 2 GB o Compact Flash or equivalent
Multimedia	external viva-voce
FLASH	64 MB
Monitor	7” touch screen
	3 Serial ports RS232
Interfaces	2 Serial ports RS485
	1 Ethernet connector
	1 external relay connector
	2 digital inputs (vehicle doors status)
	Badge reader
	High impedence odometric input
	1 CAN interface
Dead Reckoning	with integrated gyroscope
Power Source	9 to 32Vcc
GPRS	GPRS/UMTS
GPS Receiver	GPS star III EGNOS, external antenna
WIFI	Wi-Fi IEEE 802.11 2,4 GHz

3.3 Knowledge representation

The main issues related to the data were the following:

- There was no global stop point knowledge base;
- Lines and journeys were represented into unparseable files;
- There was no geocoded information at all.

The main objectives of this phase were to collect stop point data along with GPS information and to gather service information in a well structured way. Thus a number of file formats have been proposed in order to collect data in files that could be parsed in deterministic way. This process has been carried out togheter with public transport companies, since stop points were known only by a restricted set of drivers. The data model have been designed based on TransXchange v.44. An Android application has been thus implemented and distributed in order to geocode stop points and to build the service on top of them.

3.3.1 Stop point database

A centralized stop point database has been created and named DURF (Database Unico Regionale delle Fermate - Regional Unique Stop Point Database) with the aim to create a central repository for every public transport company. One of the main issues, in fact, was that the same stop point was identified more than once and named differently in the same company and across different companies. For each company stop points have been collected and then normalized:

- Duplicates have been removed;
- Naming conventions have been applied;
- Different stops with the same name have been corrected.

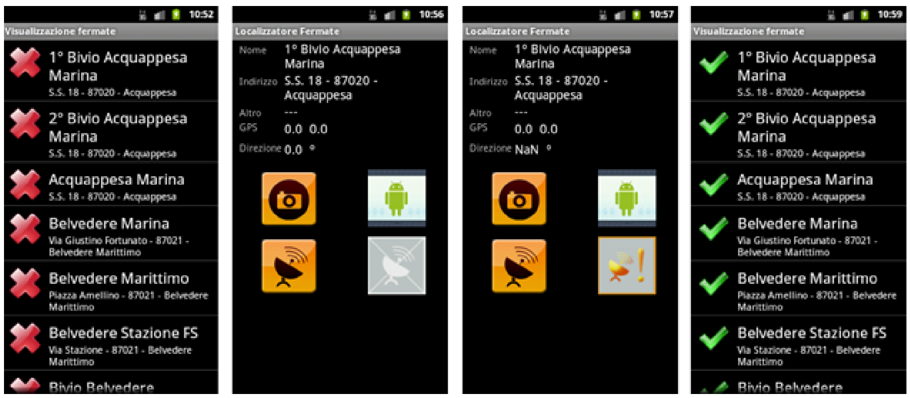


Figure 3.4: Android screen for stop point locator app

An Android 2.3 application has been implemented in order to take pictures and GPS coordinates of each company. Each company has been, in fact, endowed with smartphones with a list of stop points in memory so that they have been mostly geocoded during the normal service.

Stop point database was modelled after TransXchange and thus the following information is recorded:

- Name
- Description
- Directions to easily identify the stop point
- Wheter the stop is located at a crossing or not
- City, zip, address
- Type (on street, station, ...)

3.3.2 Service database

As soon as the stop point database has been completed for a single company, the service it delivers can be written down in an EXCEL file with a strictly deterministic format. The old file format (shown in figure 3.5) used for recording

Società Consortile a r.l. Autoservizi dei Due Mari
Zanfini

AUTOLINEA N. 113 : Serricella - Acri - Pietre Marine - Contrada Salice (limitatamente al periodo 01/07 – 31/08)

rario corse andata	Km	Fermata	Comune	Km	Orario corse ritorno
7.00	0,000	Serricella di Acri	Acri	45,600	20.00
7.02	0,600	Bivio Acri	Acri	45,000	19.58
	9,600	Acri (solo itinerario)	Acri	36,000	
7.35	10,300	Merolini		35,300	19.25
7.40	11,000	Bivio Merolini		34,600	19.20
7.45	11,800	Croce Don Paolo		33,800	19.15
7.50	13,100	Pietremarine Inferiore		32,500	19.10
7.55	14,200	Pietremarine Centrale		31,400	19.05
8.00	16,300	Pietremarine Superiore		29,300	19.00
8.05	18,100	Bivio Crista		27,500	18.55
8.10	19,000	Aria delle Donne		26,600	18.51
8.11	19,100	Casino Cofone		26,500	18.50
8.15	20,500	Crista Inferiore		25,100	18.45
	38,700	Bivio SS 106 (solo itinerario)		6,900	
9.00	45,600	Salice di Corigliano		0,000	18.00
T					T
MA-ME-GI-VE-SA-DO					LU-MA-ME-GI-VE-SA-DO

Figure 3.5: Service file format used prior to CORE

service information was not appropriate for the following reasons:

- there was a variable number of columns and thus the central column, which is the stops list, was not easily recognizable;
- there was not any distinction between inbound and outbound journeys in terms of stop points;
- there was not any link between stop points name and an eventual external data source; in other words there was no code which could identify stops;
- operating profiles was used as a free text field;
- it was impossible to record the daily operating profile of a journey (e.g. it runs on Monday but does not run on Thursday).

For these reasons a new file format has been designed in order to record information properly and to overcome the current limitations. A set of tools have been then implemented in order to import these files into CORE. Alike the stop point database, the service database was modelled after TransXchange.

3.4 Functionalities

A number of functionalities are currently provided:

- Anonymous users are provided with information regarding stop points, companies and lines. A travel planning allows to plan a journey by matching more than one line and company and a simple AVL system enable users to see the real time status of the trips they are interested in;
- Users belonging to a transport company are provided with visual editing features allowing easy and consistent data manipulation. The whole process related to any modification proposal and subsequent approval by officers of the Regione Calabria is completely dematerialized and mapped into the portal. Thus the entire process is simplified and data is kept consistent. For each trip, the AVL system generates a certification report that can be analyzed by users. Should any controversy or unpredictable event arise, a ticketing system helps Company and Regione Calabria users in the settlement;
- Officers of Regione Calabria are provided analytics and detailed information on the execution of each journey. These individuals manage the whole database and monitor in real time all the fleets belonging to each transport company.

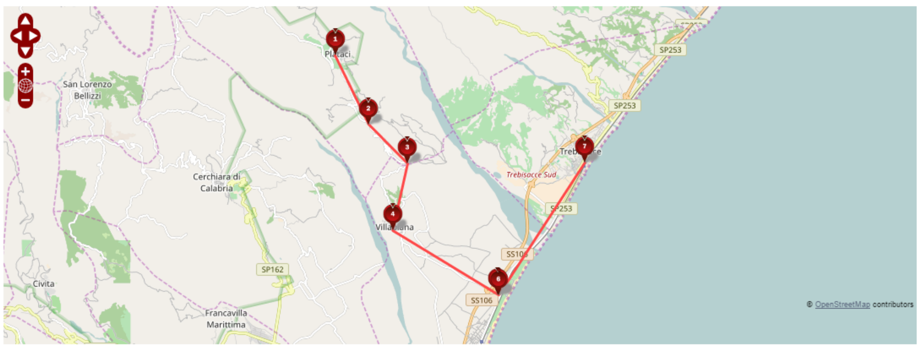
At the time of writing C.O.RE. has introduced a breaking change into the processes of the Regione Calabria Transport Office, since data management have been totally moved from an Excel based process to one that is fully managed by the platform. Finally, each vehicle can be tracked and real time information is available for every journey; thus a plethora of real time infomobility related application can be built atop of it.

3.4.1 Guest user

This section describes the functionalities provied to the guest users, eg. tourists or citizens seeking for public transport information.

Service visualization

Users can browse the service delivered by each company and make search on it. By means of a cartographic map based on the OpenLayers library, timetables, routes and other related information are displayed. (3.6).



Corse di andata | Corse di ritorno

Fermata			
Plataci - Plataci	07:10		
Plataci - Piano del Giudice	07:15		
Plataci - Curva Brunetti	07:25		
Villapiana - Villapiana	07:30	07:45	13:30
Villapiana - bv. Villapiana	07:35	07:50	13:35
Villapiana - bv. Ss 106 per Villapiana	07:40	07:55	13:40
Trebisacce - Trebisacce - P.le Staz. F.S.	07:45	08:00	13:45
Profilo operativo	LU MA ME GI VE SA DO Fe Sc	LU MA ME GI VE SA DO Fe Sc	LU MA ME GI VE SA DO Fe Sc

Si esegue nel giorno indicato
Non si esegue nel giorno indicato

Figure 3.6: A timetable as can be seen by guest users

Companies visualization

Users can browse the list of consortia and see which companies compose each consortium. There is information such as email and telephone number.

Stop points visualization

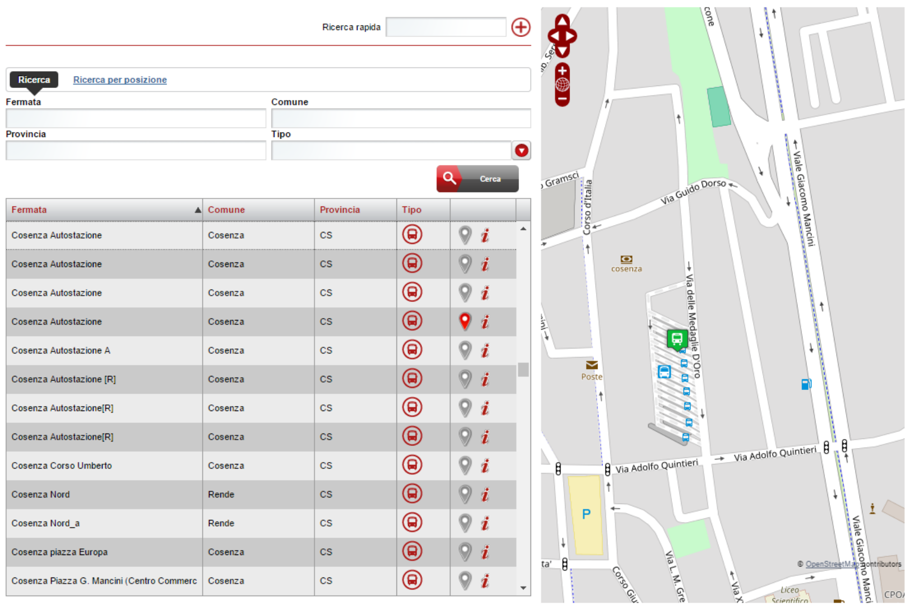


Figure 3.7: Stop points visualization

Users can browse the list of stop points and make searches according to different filters:

- city;
- distance from a given point;
- name;
- code.

For each point a picture is taken in order to help user correctly identifying the stop point; relying on services like Google Street View may not be reliable.

Service monitoring

Users can see all the journeys that are currently serviced by every operator. For each journey the current position of the vehicle is displayed as well as the path followed so far and all the stop points that are expected to be passed by (3.8).

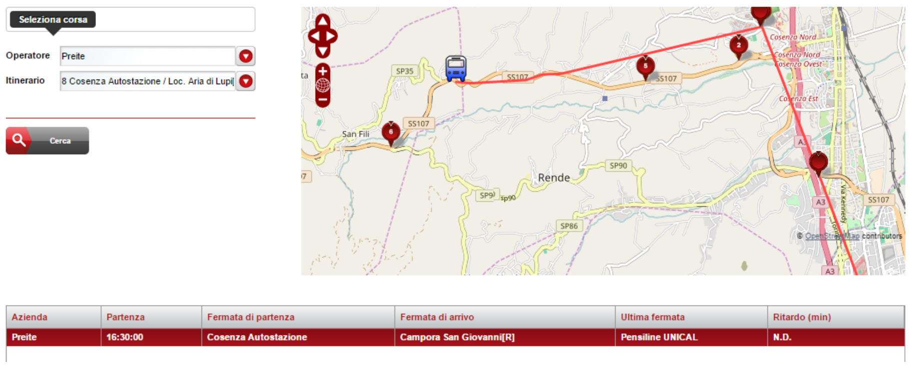


Figure 3.8: Stop points visualization

3.4.2 Transport user

There are several types of roles related to the transport user level:

- Regione Calabria
- Consortium
- Company

Each role can operate on a view of data that is under its responsibility: thus a company operates only on its data, a Consortium on all data related to its companies and Regione Calabria on data related to every Consortium.

Each page acts differently according to the role of the logged in user.

Fleet visualization and monitoring

Users can download X509 certificate for each vehicle. As one can read in section 3.5, no connection to the Automatic Vehicle Location system is granted without estabilishing a proper SSL connection. Moreover each fleet of each company is real time monitored and for each vehicle the on board journal is recorded and displayed. The on board journal records every event that has been detected and sent to the CORE. For further information please refer to 3.5.

Data editing

Users can edit all the data they are responsible for. A company may edit lines, journeys and vehicles that are under its responsibility; a Consortium may edit all these data for all the companies which are associated with it and Regione Calabria may edit everything. Every edit operation is tracked and a version history for each database entity is recorded. This feature has been implemented by using Hibernate Envers Framework.

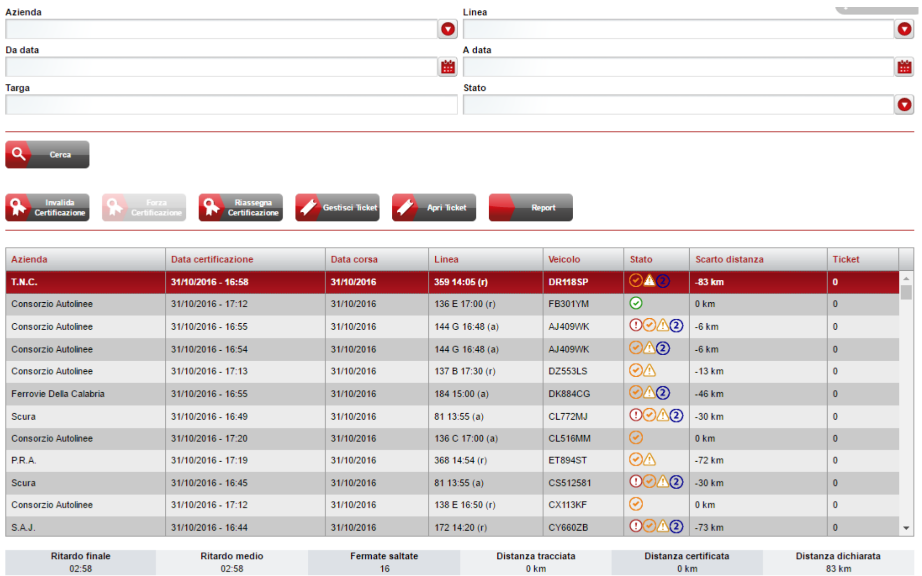


Figure 3.9: Certification page

Certification

Each vehicle is expected to send to the AVL system its location in real time together with some other information useful to find out which journey is currently serviced. As soon as a vehicle ends a journey, an algorithm runs in order to detect if the journey has been properly done. For each journey a so called *certification* is calculated and for each certification user can see which vehicle has serviced which journey, the route that has been run and the stop points that have been passed or skipped along with timing information. For further information please refer to 3.5.4. In case of anomaly, users may open a ticket associated with a certification and provide explaination.

Data exportation

Service data can be exported in a number of formats and projections, eg. EXCEL, PDF or GTFS. Moreover, some other data can be exported such as vehicle X509 certificates.

3.5 AVL system

3.5.1 AVL data frame

As said in 3.2.1, the scope of the activities performed within the project was to define the communication interface between the SBV layer and the CORE. For security reasons the tracking server is reachable only via HTTPS with mutual authentication and thus each vehicle is endowed with an X.509 certificate. The communication protocol has been designed in order to have a small impact on the SBV cpu time as well as on the bandwidth: to this aim each data frame consist of a 42 bytes long byte array. The data frame is shown in table 3.4

Table 3.4: Tracking data frame

Field	**Lenght**	**Total**
version	1 bytes	42 bytes
timestamp	8 bytes	
flags	1 bytes	
location	12 bytes	
heading	4 bytes	
speed	4 bytes	
accuracy	1 bytes	
trackingInfo	1 bytes	
vehicleJourney	4 bytes	
journeyOrdinal	2 bytes	
parameter	4 bytes	

Here follows the description for each field of the data frame:

- version: packet version;
- timestamp: time when the packet has been constructed;
- flags: a set of boolean values, specified later;
- location: current vehicle's position;
- heading: vehicle's direction with respect to the north;
- speed: current speed of the vehicle;
- accuracy: gps accuracy;
- trackingInfo: type of packet along with subtype information;
- vehicleJourney: the identifier of the journey currently serviced;
- journeyOrdinal: the occurrence of the journey in case it is frequency based;
- parameter: depends on the packet type, contains extra data.

For each of the above listed fields, the following sections contain full details.

Version

Version indicates the major version of the tracking packet that is sent. Currently the most recent version is 1 and thus *version* field is 00000001.

Timestamp

It's the instant of time when the packet is constructed (i.e. all its relevant information is detected from sensors). It is represented in UNIXTIME and it's therefore a 64-bit integer number. It can be read from a GPS NMEA sentence (i.e. $\$GPRMC$ or $\$GPGGA$).

Flags

Table 3.5: Tracking packet: flags

degraded 1 bit	deadReckoned 1 bit	notFixed 1 bit	unused 5 bit
1 byte			

Flags are boolean values containing the following information:

- degraded: 1 bit. If it is equal to 1 then the packet has not been sent in real time due to lack of GPRS signal;
- deadReckoned: 1 bit. It if is equal to 1 then the GPS position is reconstructed by the dead reckoning system;
- notFixed: 1 bit. If it is equal to 1 then the GPS signal has not yet been fixed
- 5 bits reserved for future use.

Location

Table 3.6: Tracking packet: location

latitude 4 bytes	longitude 4 bytes	altitude 4 bytes
12 bytes		

It is the vehicle's current position expressed into GPS coordinates (WGS84 / EPSG 4326 projection). It is composed by the following fields:

- Latitude: 4 bytes. Represents a floating point number;

- Longitude: 4 bytes. Represents a floating point number;
- Altitude: 4 bytes. Represents a floating point number.

Heading

It is the angle with respect to the North the vehicle is heading to. It is a 4 bytes floating point number.

Speed

The instant speed in km/h. It is a 4 bytes floating point number.

Accuracy

GPS precision in meters. It is a 1 byte integer number.

TrackingInfo

Table 3.7: Tracking packet: trackingInfo

type 4 bits	event 4 bit
1 byte	

A tracking packet is sent based on one or more modes. A vehicle can send a packet after a certain time has passed or after a certain amount of meters or when an event occurs. The type field allows to specify the reason why the packet has been sent. More precisely, type can have the following values:

- Event = 1: the packet has been generated based on an event. The exact type of event is stated by the event field;
- Distance = 2: the vehicle has run a certain amount of meters
- Timed = 3: a certain amount of seconds has passed since the last transmission.

When type is equal to 1, the field *event* (which defaults to 0) must have one of the following values:

- Departure = 1
- Arrival = 2
- Doors opening = 3
- Doors closing = 4

- Stop point reached = 5. A stop point has been reached. The stop point ID must be stated by using the *parameter* field
- Stop point skipped = 6. A stop point cannot be reached (i.e. road is closed). The stop point ID must be stated by using the *parameter* field
- Departure as replacement = 7: it must be used if another vehicle was performing the same journey but it had to stop for any inconvenience
- Vehicle failure = 8
- Vehicle journey change = 9. The last vehicle journey ID that has been sent to the CORE was wrong
- Cancel = 10. There was an error in transmission, all the packets sent so far must be discarded

VehicleJourney

ID of the vehicle journey as stored into the CORE database.

JourneyOrdinal

Currently not used, must be 0. If a vehicle journey is frequency based, i.e. is performed more than once every day, this is the ordinal of the occurrence of the same journey in the same day. For instance, if a journey is performed at every hour starting from 6 a.m., the 8 a.m. occurence will have $journeyOrdinal = 2$, the one starting at 9 a.m. will have $journeyOrdinal = 3$ and so on.

Parameter

Contains extra information according to values assumed by the *event* field of the data frame. Apart from the cases in table, it must be 0.

Table 3.8: Tracking packet: parameter

Event type	Parameter meaning
Stop point reached	Stop point ID
Stop point skipped	
Vehicle journey change	Old vehicle journey ID

3.5.2 Tracking protocol

Generally speaking the communication between SBV and CORE is based on the transmission of a set of packets:

- Start packet P_S
- N intermediate packets P_{T_i}
- End packet P_E

The following subsections show some examples related to the correct implementation of the tracking protocol.

Start packet

The start packet P_S is sent whenever a vehicle starts a new journey. An example is shown in table

Table 3.9: Tracking protocol: Start packet example

Version	
Timestamp	
Flags	
Location	
Heading	
Speed	
Accuracy	
TrackingInfo	00010001 (departure)
VehicleJourney	
JourneyOrdinal	
Parameter	journey ID

Start packet with not fixed GPS

Whenever a vehicle must start its journey but the GPS has not yet fixed, the start packet is modified accordingly to tell the tracking server that the position is not accurate. The location must be set to the first stop point of the journey.

End packet

The end packet P_E tells the system that the journey has been completed. In this case the $trackingInfo$ field will contain value 1 in the most significant bits, stating that the packet is raised upon an event, and value 2 in the least significant bits.

Table 3.10: Tracking protocol: Start packet with not fixed GPS example

Version	
Timestamp	
Flags	00100000 (not fixed)
Location	(first stop location)
Heading	
Speed	
Accuracy	
TrackingInfo	00010001 (departure)
VehicleJourney	
JourneyOrdinal	
Parameter	journey ID

Table 3.11: Tracking protocol: End packet example

Version	
Timestamp	
Flags	
Location	
Heading	
Speed	
Accuracy	
TrackingInfo	00010010 (arrival)
VehicleJourney	
JourneyOrdinal	
Parameter	journey ID

Tracking packet

The generic tracking packet P_T may be sent upon event or after elapsed time. Table 3.12 shows the relevant field of time based packet. As one can see the field $TrackingInfo$ contains value 3 in the four most significant bits.

Table 3.12: Tracking protocol: Tracking (timed) packet example

Version	
Timestamp	
Flags	
Location	
Heading	
Speed	
Accuracy	
TrackingInfo	00110000
VehicleJourney	
JourneyOrdinal	
Parameter	journey ID

Frequency

Packet frequency F may change during a route in order to optimize network usage and the tracking server resources. Frequency changes based on vehicle speed: the slower the vehicle goes, the less packets are needed. In example, when a vehicle stops to a traffic light, frequency may be lowered and raised up again as faster as the vehicle goes.

Bulk transmission

Since on board units may cache packets and send them as soon as they have GPRS connection, a bulk transmission may be used in order to make a single connection to the tracking server. Thus a bulk packet is generated by putting together more than one tracking packet with the following limitations:

- packets must be sorted by timestamp in a bulk packet;
- packets must refer to the same vehicle journey.

Offline vehicles

Whenever a vehicle does not send any data for more than 6 hours, it is considered offline and thus its current journey is closed by the system. In this case the next packet sent by the same vehicle must be a start packet P_S, otherwise it is rejected (see 3.5.2).

Sending more than one start packet

If a vehicle sends a start packet while in the middle of a journey, this journey is closed and another one is opened for the same vehicle.

Server responses

The responses of the tracking server have been set up in order to be as more clear as possible to those who implement SBVs. Each response is an HTTP status following the $RFC2616$. Here follows the responses list:

- 200: OK, if the packet (or the bulk packet) is correct. It is correctly processed;

- 202: ACCEPTED, if the SBV has sent too many packets in a short time frame (i.e frequency is too high);

- 406: NOT ACCEPTABLE, if a packet is send after 6 hours from the last transmission from the same vehicle and it is not a start packet;

- 601: INVALID-DATA, if the packet size is not valid;

- 602: NOT-RECEIVED-START, if the SBV tries to send a tracking packet without sending a start packet first;

- 603: INVALID-PACKET, if GPS values are not valid;

- 604: INVALID-TIMESTAMP, if *timestamp* field is not valid (e.g. it refers to *tomorrow*);

- 606: INVALID-LEGACYCODE, if vehicle journey ID is not recognized by the tracking server;

- 607: INVALID-FINGERPRINT, if certificate is not accepted;

- 608: INVALID-LEGACYCODE-CHANGED, if the vehicle journey ID of two packets referring to the same journey is different;

- 610: INVALID-DURF, if the stop point ID is not recognized;

3.5.3 Tracking data model

TransXChange data format does not aim at dealing with real-time data, thus an extention to that has been proposed in order to keep track of the service that is delivered every day.

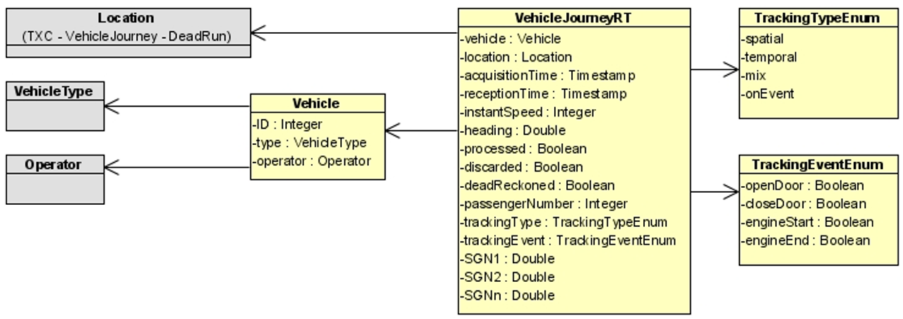


Figure 3.10: CORE - software architecture

The main entities are shown in figure 3.10 and are described in the following:

- Vehicle, representing a vehicle of a particular operator;
- VehicleJourneyRT, containing the basic information for vehicle tracking. Relevant attributes are:
  - Location: geographic position of the vehicle;
  - Processed/discarder: states whether the tracking data has been correctly processed or not;
  - DeadReckoned: states if the data is detected from a GPS or is reconstructed by the dead reckoning system;
  - TrackingType: indicates the tracking data type (i.e. based on time lapse or event);
  - TrackingEvent: the event which has occurred;
  - PassengerNumber: available for future use, whenever a ticketing system will be available;
  - Other sensor data (i.e. SGN attributes).

3.5.4 Service certification

As soon as a journey J ends, all the packets referring to J are collected and a certification algorithm is run in order to detect if the journey has been properly done.

Let $S = s_1, s_2, ..., s_n$ the set of stop points relative to J. Let $P = p_1, p_2, ..., p_n$ the set of tracking packets sent for J. Let d the distance measured by interpolation of all p_i.

ALGORITHM 1: Certification algorithm

```
Data: S, J, P
Result: Certification C for J
d = 0 ;
S_pass = ∅ ;
P_pass = ∅ ;
S_skip = ∅ ;
j_last = −1 ;
foreach s_i ∈ S do
    p_j = the closest point to s_i : j > j_last ;
    if p_j ≠ ∅ then
        j_last = j ;
        S_pass = S_pass ∪ s_i ;
        P_pass = P_pass ∪ p_j ;
    else
        S_skip = S_skip ∪ s_i ;
    end
end
foreach (p_k, p_{k+1}) ∈ P do
    d += distance(p_k, p_{k+1});
end
```

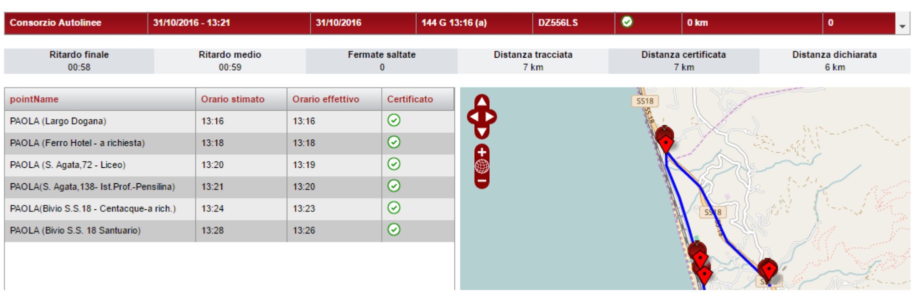


Figure 3.11: Example of certification

A certification C for a journey J is thus a set of stop points that have been passed, S_{pass} and a set of packets which correspond to the passed stops. Thus it is possibile to reconstruct the actual passing times at the stops.

Moreover a distance d is measured in order to figure out how many kilometers have been run by the vehicle, computed by summing all segment lengths joining all the tracking packets. Each segment length is calculated by the haversine formula. At the time of writing there are plans to introduce a more accurate way of computing d, i.e. by measuring the actual distance between points by using a shapefile and a map matching algorithm.

Since a vehicle can erroneously declare a different journey from the one it is actually doing or a journey may be only partially done (for example if road is closed to traffic), user can open a ticket, by using the internal ticketing feature, in order to explain his reason and to let a user from Regione Calabria force manually the output of the certification algorithm if his reason is accepted.

3.6 Impacts

At the time of writing a number of vehicles are running every day and thus a number of journeys are being tracked in real time. With each passing day, more and more vehicles are provided with SBV so that they can be tracked. This makes it possible to provide real time information to users, thus enabling them to:

- effectively monitor the provided service and alert Regione Calabria in case of misconduct;
- Have a more accessible public transport service and thus be encouraged to shift from private to public means of transport.

Before CORE information was fragmented across the websites of each company and of Regione Calabria, with the different sources often being inconsistent with each other. Now information is centralized and consistent across different formats (pdf, excel files and so on). Regarding infomobility a number of third party applications may be implemented in order to provide several tools for users and decision makers.

One of these has been implemented during the present work activity and consists of a bus departure board for the Lamezia Terme International Airport (international code: SUF). In the last year lines servicing Lamezia Terme have been deeply modified in order to pass by the Airport, which is now easily reachable from all over Calabria; thus the need of a departure board arises, in order to make passengers aware of the new services. The departure board has been implemented with a look and feel preserving the visual identity of the CORE itself while adapted to the screen size and ratio it is shown on. Currently it only displays static information but there are plans to add some real time data such as delays or cancellations.

Furthermore thanks to a centralized repository of information, some tools have been implemented with the aim to ease the work of the Public Transport Office of Regione Calabria.

3.6.1 Dematerialization of processes

Some processes of the Public Transport Office have been dematerialized and implemented in CORE:

- Timetables change requests made by the public transport

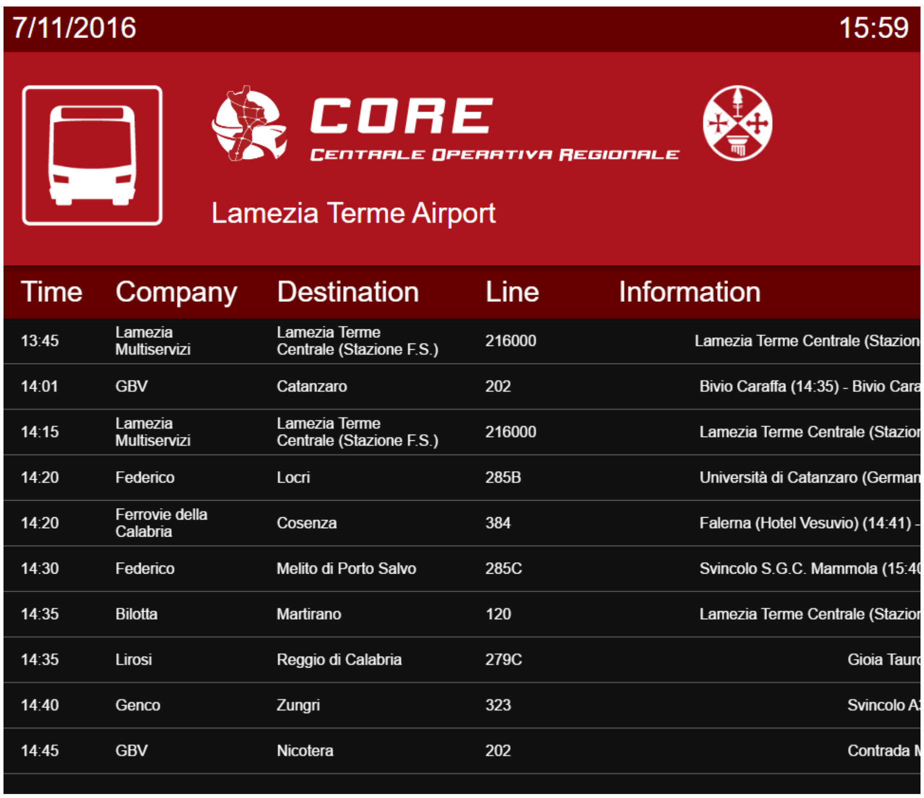


Figure 3.12: Departure board in Lamezia Terme International Airport

- Periodic cost reporting

Timetables change requests

Timetables often vary for different reasons. In this case the company responsible for the service must issue a formal request to its consortium which must first approve the request and then send it to the Regione Calabria for the final approval. This process has been completely dematerialized. Thanks to CORE, the changes made by each company to its timetables are forwarded automatically. Users from the consortium are then notified of these proposals, followed by notification of the users from Regione Calabria. As soon as a proposal receives the second approval (the one from Regione Calabria), a change to the data is triggered so that final users can see the new version of the timetables. While in the past this process used to take days, it may now take just several minutes to be completed.

Periodic cost reporting

A company is refunded by Regione Calabria of a certain amount of money for each km, which may vary from line to line. Before CORE, each change in the planned service had to be reflected in several EXCEL files in order to make information consistent across different files. This process was obviously prone to errors. For this reason, CORE implements a feature allowing users to export yearly cost reports with one click without risk of inconsistencies between the planned service and the report data. This feature allows the Office to save many person months.

3.7 Scientific works

CORE provides a field laboratory for scientific studies and experiments. In the following chapters two scientific applications will be presented:

- PT Shift Scheduling problems;
- Driving Style Analysis

3.7.1 Shift Scheduling

We present an integrated approach to solve two shift scheduling problems for local public bus companies. The first one aims at finding a schedule for vehicles, given a set of rides to execute. The second one aims at assigning drivers to vehicle schedules. The first subproblem to be faced is the *Multiple Depot Vehicle Scheduling Problem* that is known to be NP-hard. Therefore, heuristic algorithms are needed to find feasible solutions for real-life instances. In this work a starting solution for this problem is found by using a greedy algorithm. This solution is then improved by a simulated annealing strategy that exploits several local search techniques. The second problem to deal with is the *Crew*

Scheduling Problem where each trip is assigned to a driver. This problem is still NP-Hard. An initial solution for the Crew Scheduling Problem is firstly found with a classical sequential approach. This solution is then modified by changing the allocation of trips on vehicles in order to minimize the combined objective function. Both problems have been modelled taking into account as many real-world constraints as possible. Several constraints take into account the European Union restrictions related to how driver shifts must be arranged. The proposed problem is different from the ones presented in the literature, as the mathematical model, and the related algorithm, are designed based on real-world requirements. Computational results have been carried out on large real world instances. The results show that the proposed algorithm is able to quickly find good solutions within a limited computational time.

3.7.2 Driving style analysis

The collection of AVL data enables the analysis of driving style. Safety and fluidity of road transportation depend on many variables, the most critical among which is certainly the human factor. In fact, the driving style of the drivers determines the traffic flow, and, if not in compliance with current regulations and common sense, can trigger events with different risk levels, including dramatic ones. In this paper we define a novel driving style classification methodology, based on human driving behavioral patterns. The latter are extracted from databases containing aggregated trajectory logs, obtained from low-cost sensors placed on vehicles. In chapter 5 we report a study based on over 11 million GPS sample taken from the AVL system. Some definitions of aggressiveness will be proposed in order to trace a profile of Calabria PT drivers.

Chapter 4

Shift Scheduling

4.1 Introduction

The main two cost items for a public transport company are the salaries of the drivers and the operational costs related to the use of the vehicles [54]. Therefore the use of algorithms aimed at reducing the number of vehicles and drivers employed is a topic of great interest. The goal of these algorithms is to allocate each trip to a single vehicle and a single driver, while minimizing a given objective function. Usually this problem is split into two subproblems where the output of the first becomes the input for the second [57]. In the first problem, the *Multiple Depot Vehicle Scheduling Problem* (MDVSP) [6, 19], a set of routes performed by a set of vehicles located in different depots is determined. The constraints of this problem are related to the features of the available vehicles and the trips to execute. In the second problem, the *Crew Scheduling Problem* (CSP) [3, 9], each trip is assigned to one driver according to safety conditions and contractual restrictions. However, in many cases even if this approach allows to reduce significantly the computational time, the overall solution may not be satisfactory because it is obtained by solving separately two distinct problems. Therefore, it does not take into account adequately the integration between these problems. The aim of the proposed technique is to keep the benefits given by a sequential approach, while considering at the same time the link between the two problems. Furthermore these two problems are subject to several constraints drawn from the EU legal framework. EU regulation no 561/2006 defines the rules that a driver shift must abide by. Other constraints come from an analysis of the domain of application. Thus, the proposed algorithm is of practical interest to industry. The following sections are organized as follows. First the problem is introduced; a taxonomy and the state of the art are presented. Then two subproblems in which the overall problem is decomposed are described with a heuristic algorithm for each subproblem. Finally computational results are presented and discussed.

4.2 The transport service

The service provided by a Public Transport Company is made of a set R of *transportation activities* performed by a set of **vehicles** that run within the boundaries of a geographic area and that can be of different types according to the service type (i.e. bus, coach, long bus, etc.). A vehicle is driven by a **driver** or, more generally, by a **crew**. A transportation activity is named **trip** (or **ride**) and it is characterized by a route that links two points in the geographic area. Thus each trip is a point-to-point link that occurs at predetermined times: departure and arrival time are defined for each trip. A trip $r \in R$ can be represented as follows:

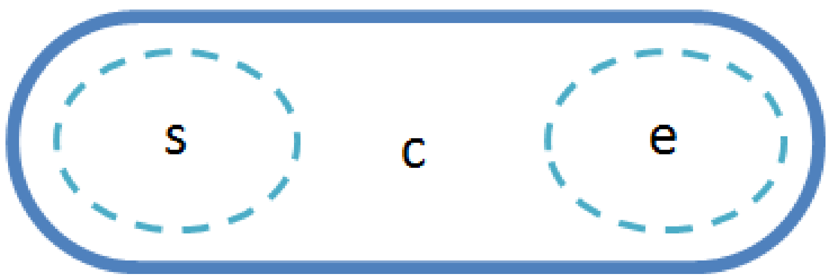


Figure 4.1: Representation of a trip

where $p_s(r)$ and $p_e(r)$ are, respectively, the starting point and the end point of r (Fig. 4.1). The geographic area is where a company provides its service: thus a set of **points of interest** is defined. A point of interest is a well-known point that can be involved by the service planning phase. A point of interest can be of one or more of the following types:

- **stop**: a point where passengers can board on or get off vehicles. Generally speaking, for security reasons, a vehicle cannot lay over at stop for more time than needed to let passengers get in/out; however, some stops can be such that parking is allowed;

- **depot**: a point where vehicles start and end their daily shifts. A depot can host a certain amount of vehicles of a given type;

- **parking point**: a point where a vehicle can lay over for a long time. It is used when a vehicle is inactive for a long time during its shift so that it cannot lay over at stop waiting for the next trip to execute;

- **refuelling station**: a point where a vehicle can reload its autonomy.

4.3 The shift planning process

For a Public Transport Company, the objective of the shift planning process is to execute all the trips that have been considered during the service design phase. To this aim the following resources must be exploited at best:

- vehicles
- crews

The shift planning process gives a set of daily shifts for the resources listed above as output. A shift σ is composed by a sequence A of activities sorted by time. For each activity $a \in A$ the following data are given::

- $p_s(a)$: point of interest where a starts from;
- $p_e(a)$: point of interest where a ends at;
- $t_s(a)$: time when a begins;
- $t_e(a)$: time when a finishes;

Two activities $a_1 \in S$ and $a_2 \in S$ are said to be *time continuous* (or merely *continuous*) if it holds that $t_e(a_1) = t_s(a_2)$. Two activities $a_1 \in S$ and $a_2 \in S$ are said to be *space contiguous* (or merely *contiguous*) if it holds that $p_e(a_1) = p_s(a_2)$. A sequence $A = \{a_1, ..., a_k, ..., a_n\}$ is *sound* if for each (a_k, a_{k+1}), these are time continuous and space contiguous. A shift σ is sound if it is composed by a sequence A that is sound in turn. Finally the objective of the shift planning process is to find a set of shifts that are sound and comply to all the constraints that are defined in the following sections.

4.4 Vehicle scheduling

The vehicle scheduling planning phase aims at optimizing the usage of a fleet of vehicles. This phase defines which trip of R must be assigned to which vehicle. For a given vehicle v, a shift ω_v is composed by all the daily activities, named *runs*, assigned to v, which departs from a depot d (**pull-out**) and returns to. The activities that form a vehicle shift ω_v may be as follows::

1. *Ride run*: activities consisting of vehicle services to the passengers. Each ride activity correspond to a trip $r \in R$ according to the plans of the local transport company;
2. *Dead run*: activities needed to move a vehicle across the geographic area (i.e. from a point of interest to another) without passenger transportation. The set of all the dead runs is $R' = \{r'_1, ..., r'_i, ..., r'_k\} : \{R \cap R' = \varnothing\}$. The dead run types are detailed in the following:

- *pull-out/pull-in*: each vehicle starts the service by traveling from the depot to the starting point of the first trip assigned to it and ends the service by traveling from the the end point of the last trip to the depot. Such dead runs are needed to place a vehicle from a depot to the first stop point, and from the last stop point to the depot;
- *transfer*: for each pair of consecutive trips (r_i, r_j) served by a vehicle v, if $p_e(r_i) \neq p_s(r_j)$ a transfer is needed to move v from $p_e(r_i)$ to $p_s(r_j)$. There are two different type of transfers:
  - *simple transfer*: if v can move from $p_e(r_i)$ to $p_s(r_j)$ and does not need to lay over at parking point or, equally, parking is permitted at least in one of $p_e(r_i)$ and $p_s(r_j)$ or the time need to move from $p_e(r_i)$ to $p_s(r_j)$ is such that parking is not needed at all (Fig. 4.2);
  - *compound transfer*: if it is none of the cases listed above, it is necessary to perform two simple transfers: one from $p_e(r_i)$ to a parking point p_p and one from p_p to $p_s(r_j)$(Fig. 4.3).

Figure 4.2: Direct transfer between two consecutive trips

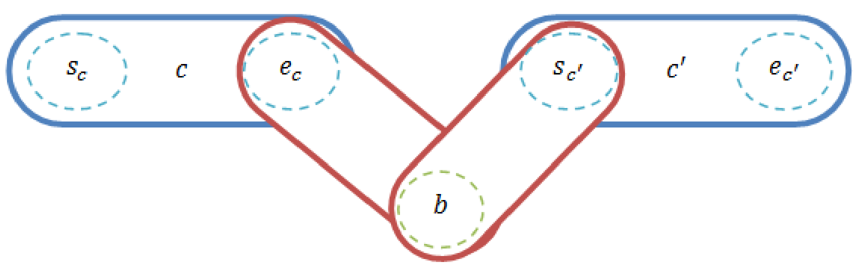


Figure 4.3: Transfer between two consecutive trips using a bus stop

Based on the above definition, the schema for a potential vehicle shift ω_v for a given vehicle v is represented in Fig. 4, where p_1, p_2, p_3, p_4 and p_5 are stops; d_1 and d_2 are depots; r_1, r_2 and r_3 are trips; r'_1, r'_2 and r'_3 are dead runs. More specifically r'_1 is the pull-out from the depot d_1, r'_3 is the pull-in to the depot d_2 and r'_2 is a simple transfer made to move v from p_3 to p_4. Generally, it is not needed for a route to start and end at the same depot ($d_1 \neq d_2$). The condition to be ensured is to have the same number of vehicles of each type in every depot by the end of the planning horizon. However, in real cases the depots are not

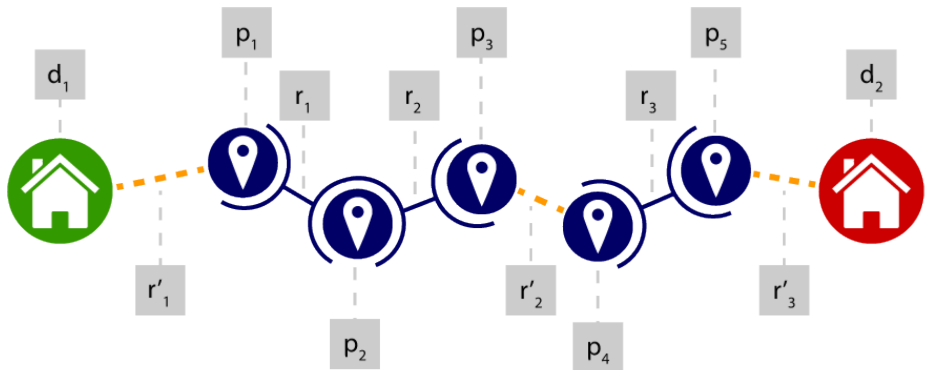


Figure 4.4: Schema for a potential vehicle shift

geographically distributed, therefore, imposing that $d_1 = d_2$ does not produce a significant deterioration of the solution.

4.4.1 Objective of the vehicle scheduling planning phase

In this section the objective of the planning phase is described. The input data are presented as well as all the constraints that have to be satisfied. The vehicle scheduling planning phase aims at:

- minimizing the number of vehicles;
- minimizing the dead runs in terms of total mileage.

4.4.2 Input data

The input data for the vehicle scheduling problem are:

- the set $R = \{r_1, ..., r_i, ..., r_k\}$ of trips. For each $r_i \in R$ the following features are defined:
  - $p_s(r_i)$: stop where r_i starts from;
  - $p_e(r_i)$: stop where r_i ends to;
  - $t_s(r_i)$: time when r_i begins at;
  - $t_e(r_i)$: time when r_i finish at;
  - $l(r_i)$: length of r_i in km;
  - the set $V(r_i) = \{\tau_1, ..., \tau_i, ..., \tau_q\}$ of all the vehicle types that can be assigned to r_i.
- the set $V = \{v_1, ..., v_i, ..., v_u\}$ of vehicles available to perform the trips. For each $v_i \in V$ the following features are defined:

- its type τ_{v_i};
- the amount k_{v_i} of kilometers that v_i can run without refuelling.

- the set $S = \{s_1, ..., s_i, ..., s_w\}$ of stops; for each $s_i \in S$ the corresponding coordinates are defined.
- the set of depots;
- the set of parking points;
- the set of refuelling stations.

Distances between points of interest

In both cases, we denote by $l^v_{r,r'}$ and $t^v_{r,r'}$ the distance and the time that vehicle v requires to move from the end location of trip r (e_r) to the start location of trip r' ($s_{r'}$), respectively. These parameters depend on vehicle v because some bus stops can be used only by a particular type of vehicle. Therefore, the distance traveled to connect two trips could change according to the vehicle type used to perform it. The length $l^v_{r,r'}$ of each transfer is computed by using the Manhattan distance.

$$l^v_{r,r'} = \begin{cases} |x_{e_r} - x_{s_{r'}}| + |y_{e_r} - y_{s_{r'}}| & \text{if } (\tau_v \in V(e_c)) \vee (\tau_v \in V(s_{r'})) \\ |x_{e_r} - x_p| + |y_{e_r} - y_p| + & \text{otherwise} \\ \quad + |x_p - x_{s_{r'}}| + |y_p - y_{s_{r'}}| & \end{cases} \tag{4.1}$$

where p is the parking point that minimizes the distance $l^v_{r,r'}$. The transfer time $t^v_{r,r'}$ is computed as the ratio between $d^v_{r,r'}$ and an average speed.

A vehicle is a resource whose utilization is an objective to be optimized by transport companies. For each vehicle $v \in V$, its starting depot d_v, its category τ_v and the maximum distance to be covered without refuel, named k_v (or equally *autonomy*), are given. A *vehicle service* is the collection of activities performed by a vehicle departing from its depot (**pull-out**) and returning to its ending location (**pull-in**). The activities performed by a vehicle $C(v)$ may be classified as follows:

Depot

A depot is the place where vehicles have to start and finish their routes. For each depot $d \in D$ its geographic coordinates (x_d, y_d), the set of available vehicles V_d, the time window within vehicles start and end their routes (s_d, e_d) are given.

Route

A route is a set of consecutive trips performed on the same vehicle. Each route starts and ends at a depot and must be performed within a given time window. Each route must comply with several constraints that are described in the next sections.

4.4.3 Literature Review

Based on the previous definitions, for a given set T of trips, a set V of vehicles located at the set D of depots, , the problem aims to assign every trip to a compatible vehicle minimizing the number of used vehicles and the total travel distance between two consecutive trips. Generally, it is not needed for a route to start and end at the same depot ($d_1 \neq d_e$). The condition to be ensured is to have the same number of vehicles of each type in every depot within the end of the planning horizon. Heuristic approaches for the MDVSP problem were developed since 1970. These approaches can be grouped as follows:

1. heuristics based on Mixed Integer Programming (MIP) formulations;
2. heuristics based on local search techniques.

In the first group are classified all the methods reaching good solutions within large computing times. The second group collects all the local search techniques quickly providing solutions for large size instances of the problem. More precisely, the MIP heuristics are based on column generation [51], branch and bound [8] and/or Lagrangian relaxation [55]. When using column generation the MDVSP can be formulated as follows:

$$\min z = \sum_{d \in D} \sum_{\omega \in \Omega_d} c_\omega x_\omega \tag{4.2}$$

$$\text{subject to} \sum_{d \in D} \sum_{\omega \in \Omega_d} a_{r,\omega} x_\omega = 1 \qquad \forall r \in R \tag{4.3}$$

$$\sum_{\omega \in \Omega_d} x_\omega \leq |V_d| \qquad \forall d \in D \tag{4.4}$$

$$x_\omega \in \{0,1\} \qquad \forall \omega \in \Omega \tag{4.5}$$

where Ω_d is the set of all the feasible routes that can be served by a vehicle starting the first trip from depot d, c_ω is the cost of route ω and x_ω is a binary variable equal to 1 if route ω is activated, 0 otherwise. $a_{r,\omega}$ is a binary variable equal to 1 if trip r appears into route ω. The main issue of this formulation is the high number of feasible routes that should be generated. This problem is solved through a column generation technique where an auxiliary subproblem is solved to iteratively generate a new route that is inserted in the master problem. For an in-depth discussion on column generation for Mixed Integer Linear Programming (MILP) problems, the reader is referred to Barnhart et al. [2]. Several column generation algorithms are presented in the literature, in which columns are found by solving subproblems that take into account different type of constraints to model limitations imposed by the transport companies ([47]). Other methods for the MDVSP are based on network flow models. Two different flow model methods were presented in literature. The first is an arc-oriented model leading to a multicommodity flow problem, while the second is a path oriented model ([59]). A different approach to solve the MDVSP problem is based on the use of local search [4] or tabu search techniques [18]. Kliewer et al. proposed a heuristic in which the main problem is split into $|D|$ subproblems [38]. Each of these problems is a Single Depot Vehicle Scheduling Problem that can be solved in polynomial time. The paths with the highest frequency are considered optimal and the new problem consists in allocating the remaining trips. Several exact formulations have been proposed in the last years by Dell'Amico [21], Bodin et al (1983), Ribeiro and Soumis [58] and Lobel [43]. However these formulations can only be used in practice to solve small size instances or real instances under unrealistic assumptions.

4.4.4 Mathematical formulation

In practical contexts a solution of MDVSP must satisfy several constraints to be feasible. In our formulation all the following constraints are considered:

1. each trip $r \in R$ is assigned to exactly one vehicle;
2. each trip $r \in R$ is assigned to a compatible vehicle;
3. the set of trips assigned to each vehicle must be compatible with its autonomy (if the autonomy is not sufficient to complete the trips, then a refuel is required);
4. the length (expressed in km) of each activity without passengers must be smaller or equal to a given value l_{max};
5. each vehicle $v \in V$ can wait in a bus stop unsuitable for parking for not more than a given value;
6. a specified number of vehicles of each type must start from and finish at each depot $d \in D$;
7. each vehicle $v \in V$ must perform its route within the time window of the depot in which it starts the service;
8. each vehicle $v \in V$ starts its route from and finishes at the same depot;
9. for each pair of consecutive trips (r, r') performed by the same vehicle, the time interval between $t_s(r')$ and $t_e(r)$ must be greater than the travel time $t^v_{r,r'}$;

The variables of the proposed MDVSP model are defined as follows:

$$x^v_{d,j} = \begin{cases} 1, & \text{if vehicle } v \text{ executes trip } j \text{ as its first trip starting from the depot } d \\ 0, & \text{otherwise} \end{cases}$$

$$x^v_{i,j} = \begin{cases} 1, & \text{if vehicle } v \text{ executes trip } j \text{ after trip } i \\ 0, & \text{otherwise} \end{cases}$$

$$x^v_{j,d} = \begin{cases} 1, & \text{if vehicle } v \text{ executes trip } j \text{ as his last trip before come back to the depot } d \\ 0, & \text{otherwise} \end{cases}$$

$$y^v_{d,j} = \begin{cases} 1, & \text{if vehicle } v \text{ travels from the depot to the ending point of trip } j \text{ without refueling} \\ 0, & \text{otherwise} \end{cases}$$

$$y^v_{j,d} = \begin{cases} 1, & \text{if vehicle } v \text{ travels from the ending point of trip } j \text{ to the depot without refueling} \\ 0, & \text{otherwise} \end{cases}$$

$$y^v_{d,d} = \begin{cases} 1, & \text{if vehicle } v \text{ executes the overall route without refueling} \\ 0, & \text{otherwise} \end{cases}$$

The following mathematical formulation holds:

$$\min z = \sum_{v \in V} \sum_{j \in R:\{d_v \ \ j\}} (l^v_{d,j} + K) x^v_{d,j} + \sum_{v \in V} \sum_{j \in R:\{j \alpha d_v\}} l^v_{j,d} x^v_{j,d} + \sum_{v \in V} \sum_{i \in R} \sum_{j \in R:\{i \alpha j\}} l^v_{i,j} x^v_{i,j} \tag{4.6}$$

$$\text{subject to} \sum_{v \in V:\{d_v \alpha j\}} x^v_{d,j} + \sum_{v \in V} \sum_{i \in R:\{i \alpha j\}} x^v_{i,j} = 1 \qquad \forall j \in R \tag{4.7}$$

$$x^v_{d,j} + \sum_{i \in R:\{i \alpha j\}} x^v_{i,j} = x^v_{j,d} + \sum_{i \in R:\{j \alpha i\}} x^v_{j,i} \qquad \forall j \in R \quad \forall v \in V \tag{4.8}$$

$$\sum_{v \in V} \sum_{j \in R:\{j \alpha d_v\}} x^v_{d,j} \leq |V_d| \qquad \forall d \in D \tag{4.9}$$

$$y^v_{d,d} + \sum_{j \in R:\{d_v \alpha j\}} y^v_{d,j} + \sum_{j \in R:\{j \alpha d_v\}} y^v_{j,d} \geq \sum_{j \in R:\{j \alpha d_v\}} x^v_{d,j} \qquad \forall v \in V \tag{4.10}$$

$$y^v_{d,d} + \sum_{j \in R:\{d_v \alpha j\}} y^v_{d,j} + \sum_{j \in R:\{j \alpha d_v\}} y^v_{j,d} \leq 2 \qquad \forall v \in V \tag{4.11}$$

$$y^v_{d,j} \leq x^v_{d,j} + \sum_{i \in R:\{i \alpha j\}} x^v_{i,j} \qquad \forall j \in R \quad \forall v \in V \tag{4.12}$$

$$y^v_{d,j} = y^v_{j,d} \qquad \forall v \in V \qquad \forall j \in R \tag{4.13}$$

$$y^v_{d,j} \left(\sum_{i \in R\{t_e(i) \leq t_e(j)\}} x^v_{d,i}(l_i + l^v_{d,i}) + \sum_{i \in R} \sum_{i^1 \in R:\{t_e(i^1) \leq t_e(j)\}} x^v_{i,i^1}(l_{i^1} + l^v_{i,i^1}) \right) \leq k_v \qquad \forall v \in V \qquad \forall r \in R \tag{4.14}$$

$$y^v_{j,d} \left(\sum_{i \in R\{t_e(i) \geq t_e(j)\}} x^v_{i,d} l^v_{i,d} + \sum_{i \in R} \sum_{i^1 \in R:\{t_e(i^1) \geq t_e(j)\}} x^v_{i,i^1}(l_{i^1} + l^v_{i,i^1}) \right) \leq k_v \qquad \forall v \in V \qquad \forall r \in R \tag{4.15}$$

$$y^v_{d,d} \left(\sum_{i \in R} x^v_{d,i}(l_i + l^v_{d,i}) + \sum_{i \in R} x^v_{i,d} l^v_{i,d} + \sum_{i \in R} \sum_{i^1 \in R:} x_{i,i^1}(l_{i^1} + l^v_{i,i^1}) \right) \leq k_v \qquad \forall v \in V \tag{4.16}$$

$$x^v_{i,j},\ x^v_{d,j},\ x^v_{j,d} \in \{0,1\} \qquad \forall i \in R \quad \forall j \in R \quad \forall v \in V \tag{4.17}$$

$$y^v_{i,j},\ y^v_{d,j},\ y^v_{j,d} \in \{0,1\} \qquad \forall i \in R \quad \forall j \in R \quad \forall v \in V \tag{4.18}$$

The objective (4.6) tries to minimize the total used vehicles and the total dead runs. Note that K represents the cost of activation of a vehicle. Constraints (4.7) impose that each trip is assigned to exactly one vehicle. Constraints (4.8) allow to build feasible routes. A route assigned to vehicle v is feasible if it starts from and ends to the same depot and if v is able to perform all the trips. The number of vehicles that can be used for each depot is limited by constraints (4.9). Constraints (4.10) and (4.11) impose that whether a vehicle is used, then it can execute the assigned service either refueling or not. More precisely, if vehicle v performs refuelling, then variable $y^v_{d,d}$ will be equal to 0 and constraints (4.11) will be satisfied for equality:

$$y^v_{d,d} = 0 \tag{4.19}$$

$$\sum_{j \in R:\{d_v \alpha j\}} y^v_{d,j} + \sum_{j \in R:\{j \alpha d_v\}} y^v_{j,d} = 2. \tag{4.20}$$

Otherwise the variable $y^v_{d,d}$ will take value of 1 and constraints (4.11) will be met for inequality:

$$y^v_{d,d} = 1 \tag{4.21}$$

$$\sum_{j \in R:\{d_v \alpha j\}} y^v_{d,j} + \sum_{j \in R:\{j \alpha d_v\}} y^v_{j,d} = 0. \tag{4.22}$$

Constraints (4.12) are used to represent the fact that a vehicle can travel without refueling from depot d_v to the end of trip j ($p_e(j)$) only if the trip is assigned to it. Constraints (4.13) impose that if a vehicle refuels after trip j, then the fuel load must be sufficient to complete the route. Constraints (4.14),(4.15),(4.16) limit the distance to be traveled before refuelling. More in details, constraints (4.14) ensure that the maximum distance traveled by a vehicle without refueling cannot be greater than k_v. Whenever a refueling occurs, then the route traveled by vehicle v is split in two parts. The maximum distances traveled before and after refueling cannot be greater than k_v as expressed by constraints (4.15) and (4.16), respectively. Finally, constraint (4.17),(4.18) define the domains of variables x and y, respectively. The problem described in this section is an IP problem whose feasible region is defined by $O(|R| \cdot |V|)$ constraints, some of which are non linear constraints, i.e, constraints (4.14),(4.15) and (4.16), and by $O(|R|^2 \cdot |V|)$ binary variables. Therefore, this problem cannot be solved to optimality in case of large size instances, even in case of linearization of the non linear constraints. However, the structure of this problem can be used to compute a lower bound for the proposed model, in which constraints related to the possibility to refuel or not the vehicles are not taken into account. In fact, dropping these constraints from formulation (4.6)-(4.18), the relaxed model reduces to an assignment problem with route constraints (4.25) and constraints on the number of vehicles to use, as expressed in the following formulation:

$$\min z = \sum_{d\in D}\sum_{j\in R:\{d\alpha j\}} (l_{d,j}+K)x_{d,j} + \sum_{d\in D}\sum_{j\in R:\{j\alpha d\}} l_{j,d}x_{j,d} + \sum_{i\in R}\sum_{j\in R:\{i\alpha j\}} l_{i,j}x_{i,j} \tag{4.23}$$

$$\text{subject to} \sum_{d\in D:\{d\alpha j\}} x_{d,j} + \sum_{i\in R:\{i\alpha j\}} x_{i,j} = 1 \qquad \forall j\in R \tag{4.24}$$

$$\sum_{d\in D:\{d\alpha j\}} x_{d,j} + \sum_{i\in R:\{i\alpha j\}} x_{i,j} = \sum_{d\in D:\{j\alpha d\}} x_{j,d} + \sum_{i\in R:\{j\alpha i\}} x_{j,i} \qquad \forall j\in R \tag{4.25}$$

$$\sum_{d\in D}\sum_{j\in R:\{d\alpha j\}} x_{d,j} \le |V| \tag{4.26}$$

$$\sum_{d\in D}\sum_{j\in R:\{j\alpha d\}} x_{j,d} \le |V| \tag{4.27}$$

$$x_{i,j}\in\{0,1\} \qquad \forall i\in R \quad \forall j\in R \tag{4.28}$$

where $x_{i,j}$ is a binary variable that has value 1 if trip j is performed after trip i, 0 otherwise. Parameter $l_{i,j}$ represents the minimum distance between the end point of trip i and the start point of trip j, calculate as the minimum distance between i and j run by a vehicle v ($l^v_{i,j}$):

$$l_{i,j} = \min_{v\in V} l^v_{i,j}. \tag{4.29}$$

Parameter $l_{j,d}$ is the distance traveled to come back to the nearest depot d from $p_e(j)$, whereas $l_{d,j}$ is the minimum distance from a depot d to $p_s(j)$. The objective function (4.23) minimizes the number of used vehicles and the total mileage. Solving to optimality this model through a general purpose MIP solver allows to obtain a lower bound for the MDVSP. Despite its weakness, computing this lower bound allows to reach a good compromise between quality and resource time consumption. The MDVSP problem is a NP-Hard problem [40]. Therefore finding a good solution in a reasonable computational time implies using heuristic techniques. We propose an Iterated Local Search (ILS) algorithm to find a MDVSP solution, in which a neighborhood search followed by a perturbation phase are iterated as long as an exit criterion is met. In Algorithm 1, we provide a pseudocode of the overall algorithm. The various steps will be detailed in the following sections by examining separately the initialization, the local search, the shaking and the acceptance decisions.

ALGORITHM 2: Iterated Local Search Algorithm for the MDVSP

Data: Set of trips R, vehicles V, depots D, bus stops P
Set $it_{current}, t_{current} = 0$, $f^* = f(s)$ and $s^* = s$;
while $it_{current} < it_{max}$ *and* $t_{current} < t_{max}$ *and* $f^*/f_{lb} > \varepsilon$ **do**
  Set $f(s_l) = +\infty$;
  foreach $type \in localSearchTypes$ **do**
    $s_t = LocalBestSolution(type, s)$;
    if $f(s_t) < f(s_l)$ **then**
      $f(s_l) = f(s_t)$;
    end
  end
  if $f(s_l) < f^*$ **then**
    $s = s_l$;
    $f^* = f(s_l)$;
  else
    foreach $s_l \in Neighborhood(Nshift, s)$ **do**
      Set $p' = N(0, 1)$;
      Set $p = e^{\left(\frac{f(s_l)-f^*}{T}\right)}$;
      if $p' < p$ **then**
        $s^* = s_l$, $f^* = z(s_l)$;
      end
    end
  end
  Update $T, t_{current}, IT_{current}$;
end

4.4.5 Initial solution

In this phase each trip is iteratively assigned to one vehicle at the least cost. Before starting the algorithm, the trips are sorted according to their departure times, if two trips start at the same instant the one with the lowest arrival time is examined first. The rationale of using this criterion is motivated by the fact that the feasibility of assigning a trip to a vehicle is performed before by checking its workload. The following definitions are needed:

- Let $\overline{V} \subseteq V$ be the set vehicles that are used. Initially $\overline{V} = \varnothing$.
- Let $\hat{V}_r \subseteq V$ be the set of vehicles that can be assigned to trip r. Initially $\hat{V}_r = \varnothing, \ \forall r \in R$.
- Let rf_v be the residual fuel of vehicle v. Initially $rf_v = k_v, \forall v \in V$
- Let $\overline{r}$ be the last trip assigned to the vehicle v. Initially $\overline{r} = \varnothing$, $t_e(\overline{r}) = 0$ and $p_e(\overline{r}) = d_v$.

At each iteration of the algorithm only one trip is examined. The first step consists of identifying the list $\hat{V}_r$ of the current trip r. Vehicle v is inserted in $\hat{V}_r$ if the following conditions are satisfied:

$$v \in \hat{V}_r \rightarrow \begin{cases} \tau_v \in V(r) \\ t_s(r) - t_e(\overline{r}) - t^v_{\overline{r},r} \geq 0 \\ rf_v \geq l^v_{\overline{r},r} + l_r + l^v_{r,d} \\ l^v_{\overline{r},r}, l^v_{r,d} \leq l_{max}. \end{cases} \tag{4.30}$$

The conditions expressed in (4.30) indicate that vehicle v is inserted in $\hat{V}_r$ if: a) it belongs to a compatible category, b) it has sufficient time and fuel to perform the trip, c) constraints on the maximum length of an activity without passengers are not violated. Under these conditions, trip r is assigned to vehicle v' that minimizes the increment of the cost function Δfo : $v' = argmin_{v \in V} \Delta fo_v$, where Δfo_v is calculated as follows:

$$\Delta fo_v = \begin{cases} l^v_{\overline{r},r}, & \text{if } v \in \overline{V} \\ l^v_{\overline{r},r} + K, & \text{otherwise} \end{cases} \tag{4.31}$$

where K is a fixed cost paid for using an additional vehicle.

4.4.6 Local Search

The solution found by the algorithm above described is generally not optimal due to the large number of vehicles employed and the total amount of distance traveled. Hence, different neighborhood operators are used with the aim of improving the initial solution. A in-depth explanation of local search algorithms designed for Vehicle Routing Problems (VRP) can be found in [7].

Nshift

Starting from a solution s, this operator builds a neighborhood by considering all the solutions that are obtained by shifting a trip between two vehicles (Fig. 5). A feasibility check is performed whenever a trip is shifted in order to ensure that the services of the involved vehicles remain feasible. Moreover, if such a move produces an improvement of the local best solution, named s^* , then s^* is updated. This procedure ends whenever it is not possible to find a better solution or after a given number of iterations it_{max}.

Nswap

This operator moves trip r_j serviced by vehicle v_i to vehicle v_{i^1}, and trip r_{j^1} serviced by vehicle v_{i^1} to vehicle v_i (Fig. 6). The aim is to find the best reduction of the objective function value.

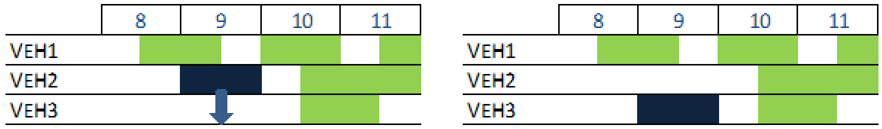


Figure 4.5: Nshift procedure

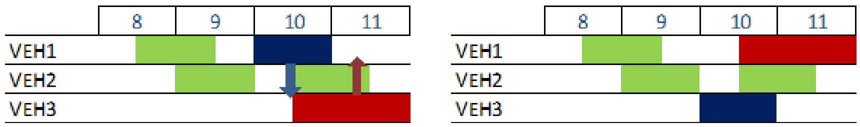


Figure 4.6: Nswap procedure

Blocks

This operator builds a neighborhood by splitting a sequence B of consecutive trips $B = \{r_j, r_{j^1}, ..., r_{j^n}\}$ serviced by vehicle v_l, and assigning them to vehicles different from v_l. Before making these operations a check on the feasibility of the remaining part of the route assigned to v_l is performed (Fig. 7). If the best solution found in the neighborhood is better, then the local best solution is updated..

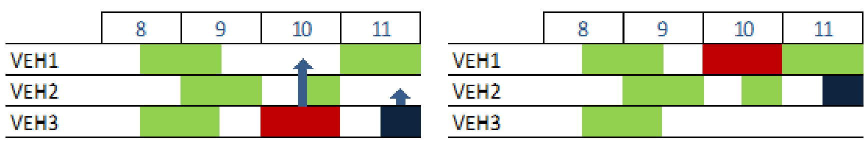


Figure 4.7: Blocks procedure

2opt*

In the 2*opt** technique the services performed by vehicles v_i and v_{i^1} are split in two parts. The solution will be modified if it is possible to obtain a reduction of the objective function by joining the first part of the service performed by v_i to the second part of the service performed by v_{i^1}, and vice versa (Fig. 8).

Changing vehicles

In the changing vehicle procedure all trips performed by vehicle v_i are assigned to vehicle v_i^1. If vehicle v_i^1 already contains some trips, the existing trips will be merged with the news ones to reconstruct the route traveled by v_{i^1} (Fig. 9).

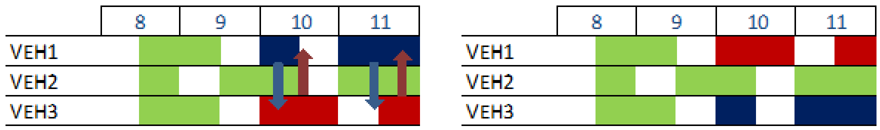


Figure 4.8: 2opt* procedure

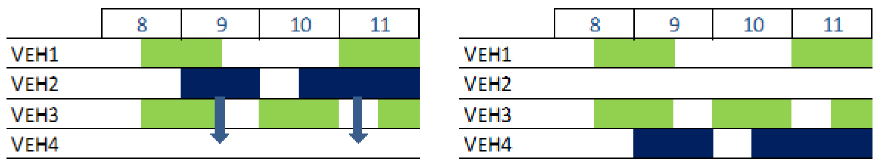


Figure 4.9: Changing vehicle procedure

4.4.7 Perturbation

We use the simulated annealing paradigm to escape from local optima. Simulated annealing is a probabilistic meta-heuristic used to find a good approximation of the global optimum of a given function [37, 35]. The name of this technique is inspired by the annealing in metallurgy, a technique involving heating and controlled cooling of material. At each iteration, an exploration of the feasible region around the current solution s, denoted by $N(s)$, is performed through the operators described in the previous Sections. If the best solution $s' \in N(s)$ has a cost lower than the cost of s, this solution is accepted and the current optimal value is updated. However, even if the cost is greater than the global best solution, this solution is accepted under certain conditions. Let f and f' be the cost of the current and the new solution, respectively. Let p' a random number with a uniform distribution between [0,1]. Let T the so-called temperature, whose value decreases during the iterations. A solution is then accepted under the following conditions: the following conditions:

$$p \leq p' \tag{4.32}$$

$$p = e^{\left(\frac{f-f'}{T}\right)} \tag{4.33}$$

It is crucial to find the starting value of T because high values allow accepting a large number of bad solutions. On the other hand, small values limit the capability of the algorithm to explore different portions of the space solution. An appropriate starting value of parameter T is chosen such that the initial probability p ranges between 0.6 and 0.9. For example, with $p = 0.8$, parameter T is computed according to the following schema. Initially a number of K iterations is performed with $T = 0$. These iterations are useful to collect statistical information regarding the average value of the increment Δ_f of the objective function.

$$\mu_{\Delta_f} = \sum_{k=1}^{K} \frac{f_k - f}{K} \tag{4.34}$$

$$T = \frac{\mu_{\Delta_f}}{\log 0.8} \tag{4.35}$$

Hence, the temperature used in the different iterations is modeled according to the residual computational time or the remaining iterations:

$$T = \min\{T_{it}, T_{time}\} \tag{4.36}$$

$$T_{it} = T_0 - T_0 \frac{IT_{current}}{IT_{max}} \tag{4.37}$$

$$T_{time} = T_0 - T_0 \frac{t_{current}}{t_{max}} \tag{4.38}$$

where IT_{max} and t_{max} are the maximum iteration and time, respectively, while $IT_{current}$ and $t_{current}$ are the current value of these parameters.

4.4.8 Stop criteria

According to our extensive preliminary testing, we found it profitable to use different criteria to stop the ILS algorithm. The following criteria are used:

1. the processing time exceeds a maximum time;

2. the number of performed iterations exceeds a maximum number;

3. the gap between the best objective value and the lower bound is smaller than a given threshold.

4.5 Crew Scheduling Problem

The salary of the drivers usually covers a percentage between 50% and 75% of the total costs. Therefore it is important to minimize the number of drivers using efficiently all the available resources. The crew scheduling problem presents many similarities with the MDVSP but also many differences, because of constraints which model safety conditions and law requirements. The trips assigned to each driver are grouped as follows:

1. driver-trips: all the trips assigned to a given vehicle driver.

2. passenger-trips: all the trips assigned to a vehicle that carries the driver as passenger. Each passenger-trip must be concatenated with a driver-trip assigned to the same or a different vehicle driven by this driver. The passenger-trips are associated with activities in which a driver must be moved in a geographical area to execute trips.

Each driver belongs to a given *Service Unit*. A Service unit U is composed by a set of bus stops where the driver can get on or get off the vehicle The shift ω_c of a driver c is defined by n activities chronologically sorted.

$$\omega_c = \{a_1, a_2, ..., a_n\} \tag{4.39}$$

$$a_1 \leq a_2 \leq ... \leq a_n \tag{4.40}$$

A shift can be divided into three subsets, named D,B,P that represent the set of driver-trips, breaks, and passenger-trips, respectively.

$$D, B, P \subseteq \omega_c \tag{4.41}$$

Let c_N denote the time between the beginning of the first activity and the end of the last, that is:

$$c_N = \sum_{i=1}^{n} t(a_i) \tag{4.42}$$

where $t(a_i)$ is the time required to perform the activity a_i. Let c_W denote the working time defined as the time period in which the driver is paid by the company. Note that c_W is equal to c_N minus all the time breaks included in B longer than μ,where μ is a fixed time. The following definitions will be helpful in the following:

1. maximum driving time c_D;

2. maximum driving time without a break $c_{\overline{D}}$: the time between two consecutive activities is considered a break only if it is greater than μ. Based on the value of μ, set B is is partitioned as follows:$B_{\geq\mu} = \{a \in B : t(a) \geq \mu\}$ and $B_{<\mu} = \{a \in B : t(a) < \mu\}$. The driving time without a break is calculated (as reported in the EU regulation no 561/2006) according to the following cases:

   - if the shift contains at least a trips longer than 50Km

   $$c_{\overline{D}} = \sum_{a \in B_{<\mu}} t(a) + \sum_{a \in D} t(d) \tag{4.43}$$

   - if the shift does not contains a trips longer than 50Km

   $$c_{\overline{D}} = b_{max1} + b_{max2} + \sum_{a \in D} t(d) \tag{4.44}$$

   $$b_{max1} = \max(t(a) : a \in B_{<\mu}) \tag{4.45}$$

   $$b_{max2} = \max(t(a) : a \in B_{<\mu} \backslash \{b_{max1}\}) \tag{4.46}$$

All these time parameters must be satisfy the following inequality:

$$c_N \geq c_W \geq c_D \geq c_{\overline{D}}. \tag{4.47}$$

4.5.1 Literature Review

The urban bus crew scheduling problem has received a great attention in the last decade, and several model and solving methods were developed. The most used models can be classified into set covering, set partitioning [46] and multi-objective models [45, 29].These models were solved either through exact methods like column generation [24] and branch and bound [31] or heuristic algorithms like tabu search [10], genetic algorithm [41] and colony algorithm [23]. Yunes et al. [64] proposed an algorithm based on a set partitioning model. Clement and Wren [13] designed a greedy algorithm based on a genetic algorithm where new solutions are found by using the operators of combination, mutation and selection. A multi-objective model was formulated by Jian and Chou [34] where different costs like idle time, number of crews, and number of layovers were considered as different objectives,and the final solution is found by evaluating the minimum distance from the utopia point. In the last decade several integrated approach for the vehicle and crew scheduling problem were proposed. The impact of using an integrated algorithm is usually more effective in the suburban or extra-urban transit system because the distance between the terminal points to be connected may be very large. The first integrated approach was proposed by Freling [28]. This approach is based on the following steps:

1. a quasi-assignment model for vehicle scheduling,
2. a set-partitioning model for crew scheduling,
3. requirements to ensure the compatibility between vehicle and crew schedules.

Huisman et al. [32] proposed an algorithm based on a combination of column generation and Lagrangian relaxation. The algorithm starts solving the MDVSP and the CSP problem for each depot by generating an initial set of columns used to calculate a lower bound through Lagrangian relaxation. Subsequently some columns with positive reduced costs are deleted and new columns with negative reduced costs are generated to solve a Lagrangian dual problem that produces a feasible solution. Proll and Wren [27] divided the problem in two smaller instances according to a geographic partition. One way to simplify solving the problem consists of splitting it into single-depot vehicle and crew scheduling problems (SD-VCSP). Splitting may be performed according to the following rules:

- the distance between the start location of the trip and the depot;
- the distance between the end location of the trip and the depot;
- solve the MDVSP problem and assign each trip to the depot where it is assigned in the solution of the problem.

Haghany et al. [30] described different techniques to solve a problem where each vehicle route is assigned entirely to one driver, and the only constraint taken into account is the maximum route length. In this context, our approach is different from the ones presented in the literature. Such a difference relies on the fact that the integration between the MDVSP and the CSP is designed to consider that finding a solution for the second problem implies to consider more realistic constraints that modify the solution found for the first problem.

4.5.2 Mathematical formulation

Constraints that a bus driver shift must comply with are specified as follows:

- each driver starts and ends the shift in a bus stop that belongs to his Service Unit;
- each driver starts the service within time window $S_c = [c_m^S, c_M^S]$ and ends the service in time window $E_c = [c_m^E, c_M^E]$
- the distance traveled by a bus driver is limited;
- a minimum number of breaks , named b_{min}, must be assigned to each bus driver ;
- a bus driver can change vehicle only on a subset of bus stops, named relief points.

The crew scheduling problem can be formulated on the basis of the following variables:

1. $x_{i,j}^c = \begin{cases} 1, & \text{if driver } c \text{ executes trip j as a driver after trip i} \\ 0, & \text{otherwise} \end{cases}$
2. $y_{i,j}^c = \begin{cases} 1, & \text{if driver } c \text{ executes trip j as a passenger after trip i} \\ 0, & \text{otherwise} \end{cases}$
3. $x_{u,j}^c = \begin{cases} 1, & \text{if trip } j \text{ is the first activity performed by driver c} \\ 0, & \text{otherwise} \end{cases}$
4. $x_{j,u}^c = \begin{cases} 1, & \text{if trip } j \text{ is the last activity performed by driver c} \\ 0, & \text{otherwise} \end{cases}$

The following mathematical formulation holds:

$$\begin{aligned} \min\ z = &\sum_{t \in T} \sum_{j \in C \cup C'} (f_t + a_j - d_j) x_{u,j}^t + \\ &\sum_{i \in C \cup C'} \sum_{j \in C \cup C'} (a_j - d_j)(x_{i,j}^t + y_{i,j}^t) + \\ &\sum_{i \in C \cup C'} \sum_{j \in C \cup C' : \{d_j - a_i \leq \mu\}} (d_j - a_i)(x_{i,j}^t + y_{i,j}^t) \end{aligned} \tag{4.48}$$

$$\text{subject to} \sum_{t \in T} x^t_{u,j} + \sum_{t \in T} \sum_{i \in C \cup C'} x^t_{i,j} = 1 \qquad \forall j \in C \cup C' \tag{4.49}$$

$$x^t_{u,j} + \sum_{i \in C \cup C'} (x^t_{i,j} + y^t_{i,j}) = x^v_{j,u} + \sum_{i \in C \cup C'} (x^t_{j,i} + y^t_{j,i}) \qquad \forall j \in C \cup C' \quad \forall t \in T \tag{4.50}$$

$$x^t_{j,u} = 0 \qquad \forall j \in C \cup C' : \{e_j \notin U_t\} \tag{4.51}$$

$$x^t_{u,j} = 0 \qquad \forall j \in C \cup C' : \{s_j \notin U_t\} \tag{4.52}$$

$$x^t_{i,j} = 0 \qquad \forall i \in C \cup C' \quad \forall j \in C \cup C' : \{i \ \overline{\alpha} \ j\} \quad \forall t \in T \tag{4.53}$$

$$T^S_m \leq \sum_{j \in C \cup C'} d_j x^t_{u,j} \leq T^S_M \qquad \forall t \in T \tag{4.54}$$

$$T^E_m \leq \sum_{j \in C \cup C'} a_j x^t_{j,u} \leq T^E_M \qquad \forall t \in T \tag{4.55}$$

$$\sum_{j \in C \cup C'} a_j x^t_{j,u} - \sum_{j \in C \cup C'} d_j x^t_{u,j} \leq T^{max}_N \qquad \forall t \in T \tag{4.56}$$

$$\sum_{j \in C \cup C'} (a_j - d_j) x^t_{u,j} + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} (a_j - d_j) x^t_{i,j} \leq T^{max}_D \qquad \forall t \in T \tag{4.57}$$

$$\begin{aligned} &\sum_{j \in C \cup C'} (a_j - d_j) x^t_{u,j} + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} (a_j - d_j)(x^t_{i,j} + y^t_{i,j}) + \\ &\sum_{i \in C \cup C'} \sum_{j \in C \cup C' : \{d_j - a_i \leq \mu\}} (d_j - a_i)(x^t_{i,j} + y^t_{i,j}) \leq T^{max}_W \qquad \forall t \in T \end{aligned} \tag{4.58}$$

$$\sum_{i \in C \cup C'} \sum_{j \in C \cup C' : \{d_j - a_i \geq \mu\}} x^t_{i,j} + y^t_{i,j} \geq n - breaks_{min} \qquad \forall t \in T \tag{4.59}$$

$$x^t_{u,j}, x^t_{i,j}, x^t_{j,u} \in \{0,1\} \qquad \forall t \in T \quad \forall i \in C \cup C' \quad \forall j \in C \cup C' \tag{4.60}$$

$$y^t_{i,j} \in \{0,1\} \qquad \forall t \in T \quad \forall i \in C \cup C' \quad \forall j \in C \cup C' \tag{4.61}$$

The objective function (4.48) minimizes the number of drivers used and the total working time. The working time is given by the sum of the duration of all the trips assigned to the driver plus the sum of all the breaks shorter than a parameter μ. The first two set of constraints (4.49),(4.50) are used to ensure that each trip is assigned to a driver and to build feasible duties. A route is feasible if the start location of each trip can be reached by the end location of the previous. The next two set of equations (4.51),(4.52) are used to ensure that each shift starts from and ends at a bus stop belonging to the Service Unit of the driver. The constraints (4.53) fix the value of the variables $x^c_{i,j}$ for each pair of not compatible trips. Two trips, i and j, are compatible if the driver has enough time to reach the departure stop of the $j-th$ trip from the end stop of the $i-th$ and if one of these conditions is met:

1. both the trips are served by the same vehicle;

2. the departure stop of the $i-th$ trip and the end stop of the $j-th$ are two relief points.

Equation (4.54) and (4.55) are introduced to respect the constraints on the time window for the start and the end of the shift while equation (4.56), (4.57) and (4.58) guarantee that the limits on the length of the shift, the working and the driving time are not violated. Finally equation (4.59), (4.60) and (4.61) are used to assign a sufficient number of breaks to each driver and to fix the value of the variables to 0 or 1. Also in this case the model contains integer variables $x^c_{i,j}$ and $y^c_{i,j}$ and has a size (number of variables and constraints) excessively large to be solved exactly. So to find a good solution of this problem it is necessary the use of heuristic approaches. However also in this case, the structure of this problem can be used to compute a lower bound of the the crew scheduling problem, in which some constraints are not considered. More specifically, in the relaxed model we relaxed the constraints related to the service unit and the minimum/maximum driving time. Moreover the relaxed model allows to assign a trip to more than one driver. However, since the driving time is not considered, then it is unnecessary to use variable set y. The mathematical formulation used to calculate a lower bound of the crew scheduling problem is therefore as follows:

$$\min z = \sum_{t\in T}\sum_{j\in C\cup C'}[f_t + W(a_j - d_j)]x^t_{u,j} + \sum_{i\in C\cup C'}\sum_{j\in C\cup C'} W(a_j - d_j)x^t_{i,j} + \sum_{i\in C\cup C'}\sum_{j\in C\cup C':\{d_j - a_i \leq \mu\}} W(d_j - a_i)x^t_{i,j} \tag{4.62}$$

$$\text{subject to } \sum_{t\in T} x^t_{u,j} + \sum_{t\in T}\sum_{i\in C\cup C'} x^t_{i,j} \geq 1 \qquad \forall j \in C\cup C' \tag{4.63}$$

$$x^t_{u,j} + \sum_{i\in C\cup C'} x^t_{i,j} = x^v_{j,u} + \sum_{i\in C\cup C'} x^t_{j,i} \qquad \forall j\in C\cup C' \quad \forall t\in T \tag{4.64}$$

$$x^t_{i,j} = 0 \qquad \forall i\in C\cup C' \quad \forall j \in C\cup C' : \{i\ \overline{\alpha}\ j\} \qquad \forall t\in T \tag{4.65}$$

$$\sum_{j\in C\cup C'} a_j x^t_{j,u} - \sum_{j\in C\cup C'} d_j x^t_{u,j} \leq T^{max}_N \qquad \forall t\in T \tag{4.66}$$

$$\sum_{j\in C\cup C'}(a_j - d_j)x^t_{u,j} + \sum_{i\in C\cup C'}\sum_{j\in C\cup C'}(a_j - d_j)x^t_{i,j} + \sum_{i\in C\cup C'}\sum_{j\in C\cup C':\{d_j - a_i\leq\mu\}}(d_j - a_i)x^t_{i,j} \leq T^{max}_W \qquad \forall t\in T \tag{4.67}$$

$$\sum_{i\in C\cup C'}\sum_{j\in C\cup C':\{d_j - a_i \geq \mu\}} x^t_{i,j} \geq n - breaks_{min} \qquad \forall t\in T \tag{4.68}$$

$$x^t_{u,j}, x^t_{i,j}, x^t_{j,u} \in \{0,1\} \qquad \forall t\in T \quad \forall i\in C\cup C' \quad \forall j\in C\cup C' \tag{4.69}$$

4.5.3 Local search algorithm

The outline of the algorithm used to find a solution of the CSP problem is similar to the one proposed for the MDVSP. Initially a feasible solution is found using a greedy approach and then such a solution is improved using local search operators. However the main difference in this algorithm is the link between the two problems. The local search techniques is devoted to find a solution of the MDVSP problem that minimizes either the number of vehicles or the number of drivers.

Initial solution

The basic idea in the algorithm is to generate a number of tasks and assign a subset of these tasks to the drivers in such a way that each trip is assigned to at least one driver. A task T is a set of consecutive trips performed by the same vehicle that starts and ends in a relief point that belongs to the same service unit. To ensure the respect of the constraints on the maximum driving time without a break, only tasks with a length smaller of $c_{\overline{D}}$ are considered. At each iteration a task and a driver are selected by minimizing the following quantity:

$$\frac{w_1 \cdot \Delta c_N + w_2 \cdot \Delta c_W + w_3 \cdot \Delta c_D}{|R_T|} + \Delta f_c \tag{4.70}$$

where $|R_T|$ is the number of not yet assigned trip in the task, w_1, w_2 and w_3 are weights assigned to c_N, c_W and c_D whereas f_c is a fixed cost that is paid if c_N is greater than 0. Trips already assigned are not taken into account in the denominator of this function and they are considered like passenger trips for the driver. An initial solution is found according to the following steps:

1. selection of the task to assign to a driver at least cost;
2. updating drivers' parameters;
3. updating the number of trips not assigned for the remaining Tasks;
4. dropping all the fulfilled (tasks without unassigned trips).

The algorithm ends when all the trips are assigned to at least one driver or is not possible to assign a new task. The trips are not assigned assigned in accordance with the cost function values. The solution returned by this procedure could be far from the optimum, but also infeasible. The constraints that could be violated are ones related to the minimum working time of the driver, and those that impose the assignment of each trip. Therefore, to restore the feasibility and improve the quality of the solution several well-known local search techniques and a new operator named $Shiftunion$ are used.

Local Search

The initial soluzion could use a large number of drivers. A reduction of this number represents a goal for the company. The $ShiftUnion$ operator aims at joining the shifts of two drivers without violating constraints More precisely, given two different shifts c_1, c_2, a new shift is created when a task T exists such that $c_1 \cup c_2 \cup T$ is feasible. The shift returned by the union of two drivers may have a length T_N equal, lesser of greater than the sum of the joined shifts. We assume that the fixed cost of using an additional driver is greater than the variable costs associated with shifting the driver. This assumption implies that the ShiftUnion tends to increase the number of trips where the drivers are moved as passengers or the total working time of the drivers. In some cases it is possible to join the shift of two drivers with no new activity. To take into account this option, a dummy task that does not contains any trips is introduced. Therefore the union of two shifts can be performed by joining them directly, or through a new set of activities. A sketch of these two possibilities is showed in Fig. 4.8 and Fig 4.11.

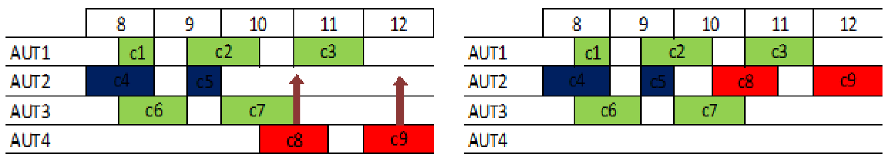


Figure 4.10: Direct union of the working schedules of two drivers

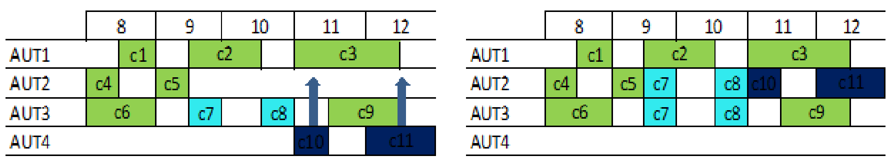


Figure 4.11: Joining of the schedules of two drivers in which the involved trips are used to move drivers

Post-optimization algorithm

After generating a the Crew scheduling solution, an improvement is executed by using the local search techniques illustrated in the previous sections. The output data of the MDVSP is a set of trips $R \cup R'$ and routes $\sigma = \{\sigma_{v_1}, \sigma_{v_2}, ..., \sigma_{v_m}\}$. Set R does not depend on the solution, while R' changes by altering the allocation of the trips in the MDVSP. A solution of the MDVSP is feasible if each trip is assigned to exactly one vehicle:

$$\sigma_{v_1} \cup \sigma_{v_2} \cup \cdots \cup \sigma_{v_m} = C \cup C' \tag{4.71}$$

$$\sigma_{v_j} \cap \sigma_{v_z} = 0 \;\; \forall (j, z) \in V \tag{4.72}$$

The output data of the CSP is a set of shifts $C_t = \{c_1, c_2, ..., c_n\}$ that is assigned to each driver $t \in T$ such that a trip can be assigned to more than one driver. The local search techniques are performed by taking into account also the new allocation of the trips to the drivers. The operators *ShiftAllocation* and *ShiftUnion* are executed on each current solution, and the current optimum is updated whenever the sum of the objective functions of the MDVSP and the CSP is reduced. Due to the limit on the number of activities assigned to a driver, the new allocation can produce infeasible solutions. The feasibility is recovered by evaluating new MDVSP solutions where the trips that overcome the workload of the assigned drivers, are assigned to those drivers with a small workload. An example of the *Nshift* technique is reported in Fig. 10:

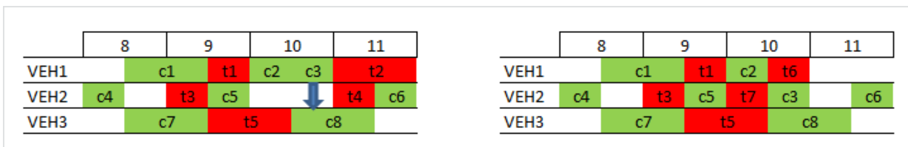


Figure 4.12: The feasibility of the overall solution is ensured by the procedure that is outlined in Algorithm 2

ALGORITHM 3: Global optimization

```
Data: Set of trips C ∪ C', vehicles V, drivers T and a feasible solution s_0
s = s_0 → current solution;
z_0 → objective function of the current solution;
c_i → select a random trip ∈ C;
v_j → current vehicle of c_i;
set C_{v_j}=C_{v_j} − c_i;
if C_{v_j} is feasible then
    foreach v ∈ V do
        set C_v=C_v ∪ c_i;
        C'_new → new set of activities without passenger;
        foreach c ∈ {C' − C'_new} do
            t → current driver of c;
            set C_t=C_t − c;
        end
        foreach c ∈ {C'_new − C'} do
            ShiftAllocation(c,T);
        end
        ShiftUnion(T);
        if s is not feasible or z(s) > z_0 then
            s = s_0;
        end
    end
else
    put C_{v_j}=C_{v_j} ∪ c_i;
end
```

4.6 Numerical Test

This section illustrates the results of an extensive computational study performed on the data set of instances that are related to a public transport company. The goal is to prove the effectiveness of the proposed ILS (Iterated Local Search) algorithm. The main computational analysis is clearly devoted to the logistic problem in which a MDVSP and a CSP are integrated together, and is given in Section 4.1. As far as we know, no benchmark instances exist in the literature for an integrated problem like the one under analysis. Therefore, we generated three sets of instances. A set of small-sized instances with not more than 150 trips drivers are solved by using the ILS algorithm, and the corresponding values are compared with the lower bounds obtained by solving the model introduced at the end of Section 2.1 with a general purpose optimization software. The second set is made by medium-sized instances with up 300 trips. These instances were designed from real data provided by Italian transport companies, and are used to investigate the impact of every neighborhood involved in the local search algorithm. The third set is made by large-sized instances with more than 300 trips. These instances were designed by taking into account as many data as possible coming from the available real data. Further investigation are performed on these instances to study the influence of the neighborhoods sequences. The fine tuning of the algorithms parameters is also discussed. Finally, in Section 4.2, we present the results obtained by the ILS algorithm. The ILS algorithm was implemented as single thread code in Java and all tests are performed on a desktop computer equipped with an Intel Core i3 processor with 2.2 GHz, 6 GB RAM, and running Windows 7 Professional.

4.6.1 Characteristics of the MDVSP instances

Data with different numbers of trips, depots and bus and park stops are used to test the performance of the ILS algorithm. The instances are grouped in three sets: small, medium and large. All the features of the instances are reported in Tables 1-2-3.

Table 4.1: Table 1 Small Instances

Id problem	N° trips	N° stops	N° depots	Total km	N° Park stops
1	82	334	26	1732.43	26
2	90	361	26	1814.87	26
3	83	335	26	1781.35	26
4	147	404	26	3380.58	30
5	147	404	26	3129.57	30
6	142	327	5	2987.32	10
7	128	224	10	2567.89	12
8	128	224	12	2567.89	12
9	102	276	12	2078.45	12
10	75	98	12	2234.56	12
11	78	92	20	2459.19	20
12	127	296	26	2478.99	26
13	122	404	26	2326.67	26
14	125	398	12	2589.78	15
15	125	275	12	2384.91	15

For each instance we impose the following limits for each vehicle service:

1. autonomy of the vehicle: 500Km;
2. maximum length of an activity without passenger: 40Km;
3. average speed of the vehicle for activities without passenger: 40Km/hour;

Table 4.2: Table 2 Medium Instances

Id problem	N° trips	N° stops	N° depots	Total km	N° Park stops
16	158	19	1	1823.36	1
17	273	19	1	3622.11	1
18	229	404	26	9705.57	42
19	180	404	26	7047.47	42
20	227	404	26	8617.57	42
21	276	19	1	8413.24	1
22	246	19	1	7212.57	1
23	221	19	1	5743.34	10
24	298	19	1	5626.89	10
25	189	376	16	7821.58	16
26	176	302	16	6237.24	22
27	158	309	16	6496.69	22
28	291	305	22	6209.97	42
29	256	356	22	6668.02	42
30	220	378	22	7314.65	42

Table 4.3: Table 3 Large Instances

Id problem	N° trips	N° stops	N° depots	Total km	N° Park stops
31	492	18	1	6476.57	18
32	326	17	1	4014.08	1
33	326	19	4	4014.08	4
34	621	19	1	8425.47	1
35	650	19	1	9147.47	3
36	620	19	4	8371.63	19
37	631	18	4	6721.11	18
38	653	17	4	8659.04	17
39	648	17	1	8792.54	4
40	498	17	1	9631.66	4
41	532	18	1	8150.54	4
42	578	18	4	7827.04	4
43	712	19	4	8919.10	15
44	679	19	1	7097.38	19
45	498	19	1	6854.86	12

4. maximum time that a vehicle can stand in a bus stop where parking is not allowed: 3 minutes;

5. a vehicle can perform a refuel at any depot.

4.6.2 Characteristics of the CSP instances

In these computational tests the results of the MDVSP problems are used as input data for the CSP problems, where each trip has to be assigned to a driver. For each driver, the following parameters are given:

1. maximum distance walked by a bus driver: 0.5 Km;
2. maximum length of the shift: 12 hours;
3. maximum working time: 8 hours;
4. maximum driving time: 6 hours;
5. maximum driving time without a break: 2.5 hours;
6. minimum length of a break: 20 minutes;
7. minimum number of breaks assigned to a driver: 2;
8. maximum number of breaks assigned to a driver: 4.

Moreover, in the following tests, three classes of shifts are considered. Different time windows are used:

- morning shift: each driver of this class begins its turn within the time window [0:00-10:00] and ends to work within the time window [8:00-18:00];
- afternoon shift: each driver of this class begins its turn within the time window [8:00-18:00] and ends to work within the time window [16:00-24:00];
- evening shift: each driver of this class begins its turn within the time window [16:00-24:00].

4.6.3 Parameter Settings

The proposed algorithm tries to improve iteratively the quality of the solution by using local search techniques and simulated annealing with a value of the initial temperature T_0 that is chosen based on the probability p of accepting worse solutions. In order to have a deeper understanding of the algorithm behavior, several executions have been performed on a given instance of each class using different value of p in the range [0.4-0.8]. The optimal value of this parameter is chosen by performing four different runs for each problems and for each possible value of p. The results obtained are reported in the following Table.

The table shows that the best results are achieved using a starting value of p equal to 0.6.

Table 4.4: Table 4 Tuning of the p value

	Problem 1				Problem 16				Problem 31			
p	#Vehicles	Km Transfer	#Drivers	Working Time (hours)	#Vehicles	Km Transfer	#Drivers	Working Time (hours)	#Vehicles	Km Transfer	#Drivers	Working Time (hours)
0.4	11	295.48	13.00	84.12	14	882.61	25.75	171.39	35	321.22	55.00	381.80
0.5	11	297.31	12.25	83.71	14	872.38	25.25	176.13	35	311.18	55.50	406.43
0.6	11	295.40	12.00	81.29	14	860.20	24.75	166.00	35	312.24	54.50	356.09
0.7	11	298.11	12.25	79.91	14	856.75	25.00	173.25	35	314.09	56.25	364.03
0.8	11	297.31	12.50	78.13	14	883.87	26.25	182.13	35	322.73	57.00	393.55

4.6.4 Results

The results of the proposed ILS algorithm for the MDVSP and the CSP are compared with the lower bounds obtained using the models proposed in Sections 4.3 and 5.1. For all the instances we used a fixed cost K equal to 50 for each vehicle, a fixed cost equal to 100 for each driver and a cost equal to 10 for each additional working hour. The lower bound z_{lb} is used to calculate the percentage gap, denoted by $Gap\%$, with respect to the objective function of the best solution returned by the algorithm, named z_{ILS}. This gap is computed as follow:

$$Gap\% = \frac{z_{ILS} - z_{lb}}{z_{lb}} \tag{4.73}$$

where z_{lb}^{MDVSP} denotes the lower bound for the multi depot vehicle scheduling problem, z_{lb}^{CSP} denotes the lower bound of the crew scheduling problem, and z_{lb} indicates the overall lower bound calculated as the sum of the bounds for both problems. The values of the lower bound of each instances are reported in Table 5.
Due to the stochastic nature of the algorithm, 10 runs are performed for each instance. A time limit of five minutes is imposed to every execution of the algorithm related to the MDVSP, and an additional time of five minutes is given to optimize the overall solution. The results are reported in Table 6 where the columns indicate the number of vehicles, the number of kilometers for transfer activities, the number of drivers, the $Gap\%$ of the best solution and the average percentage gap, denoted by ($\mu_{Gap\%}$) obtained in 10 runs.
The table shows that the percentage gap of the best solution ranges between 2.01% and 15.65%. However, it is possible to see that the largest gaps are reached for the instances with 26 depots. The largest gaps are partially due to the deterioration of the lower bound for the MDVSP problem in the instances where the number of depots is high, because the length of the pull-in and pull-out trips are always evaluated by considering the nearest depot. Finally, to understand the impact of each neighborhood we executed the tests for the MDVSP problem using different versions of the algorithm in which each neighborhood was assessed individually. For each instance and for each neighborhood we considered not only the gap with respect to the lower bound, but also the quality of the improvement Δf measured as the difference between the final (z^{MDVSP}) and the initial solution (z_0^{MDVSP}) returned by the algorithm presented in Section 4.4. For this analysis we used two instances of each class.
The best results are achieved using $2opt^*$ and *Blocks* procedures. For each in-

stance, these neighborhoods are able to reach better results compared to the $Nshift$ and the $Nswap$. The analysis performed in Table 7 was used to determine the percentage of time allocated to each local search operator. Let n be the number of local search operators and let e_i be the average percentage gap using only the i-th operator. The percentage of time $\%t_i$ allocated to each operator is calculated as:

$$\%t_i = \frac{\frac{1}{e_i}}{\sum_{j=1}^{n} \frac{1}{e_j}} \tag{4.74}$$

As a result, we observe that 21% of the time is dedicated to the $Nshift$ operator, 20% of the time is devoted to $Nswap$, 30% of the time is dedicated to the 2opt* and 29% of the time is dedicated to the $Blocks$ operator.

4.7 Conclusion

The resolution of the *Multi Depot Vehicle Scheduling Problems* and the *Crew Scheduling Problems* is an extremely important step for the operation planning of a local transport company. Since 1970, different heuristic algorithms have been proposed for both problems. However only in the last decade the two problems have been solved with an integrated approach. Moreover these approaches solve problems based on several simplifying assumptions. The aim of this work is to present an integrated algorithm for the MDVSP and the CSP problems where different constraints according to EU legal framework (561/2006) have been taken into account. This technique is based on the principle that a sequential approach can be used to reduce the computational time using some procedure that allow to consider the connections of these two problems. The tests made show that a significant improvement of a starting solution can be obtained using the local search techniques presented. Moreover the quality of the final solutions is guaranteed by the Simulated Annealing operator that allows the search to escape from the local minima. The results demonstrate the quality of the method presented with which is possible to determine in a small time a solution for a real-world problem with many trips, depots, vehicles and drivers.

Table 4.5: Table 5 Lower Bound

Id problem	Vehicles	Km transfer	z_{lb}^{MDVSP}	Drivers	z_{lb}^{CSP}	z_{lb}
1	11	248.04	798.04	12	2070.35	2868.39
2	16	330.78	1130.78	16	2789.87	3920.65
3	14	315.34	1015.34	15	2473.64	3488.98
4	21	468.15	1518.15	25	4082.85	5601.00
5	22	488.13	1588.13	24	3966.60	5554.73
6	20	299.79	1299.79	22	3581.82	4881.61
7	22	469.54	1569.54	23	3680.53	5250.07
8	21	392.03	1442.03	22	3790.16	5232.19
9	22	385.05	1485.05	25	4347.10	5832.15
10	21	134.68	1184.68	24	4180.90	5365.58
11	23	39.71	1189.71	24	4100.57	5290.28
12	23	410.90	1560.90	23	3919.61	5480.51
13	21	223.69	1273.69	21	3497.61	4771.30
14	22	453.58	1553.58	18	3112.06	4665.64
15	21	405.93	1455.93	18	3015.41	4471.34
16	14	858.20	1558.20	16	2609.57	4167.77
17	21	1358.60	2408.60	23	3951.33	6359.93
18	38	1120.10	3020.10	42	6811.22	9831.32
19	31	843.50	2393.50	35	5644.35	8037.85
20	37	1149.15	2999.15	41	7040.07	10039.22
21	36	782.08	2582.08	41	6833.27	9415.35
22	32	1009.28	2609.28	40	6403.20	9012.48
23	27	907.45	2257.45	37	6105.37	8362.82
24	26	1331.70	2631.70	31	5087.16	7718.86
25	33	908.98	2558.98	36	5958.72	8517.70
26	29	1358.95	2808.95	32	5314.82	8123.77
27	29	1248.60	2698.60	30	5139.15	7837.75
28	28	1247.26	2647.26	31	5298.89	7946.15
29	31	1073.36	2623.36	37	6206.45	8829.81
30	32	799.04	2399.04	36	5830.24	8229.28
31	35	303.83	2026.98	54	9085.01	11111.99
32	22	1284.78	2369.79	46	7457.24	9827.03
33	21	789.66	1812.33	45	7431.98	9244.31
34	43	2555.55	4516.80	55	9300.45	13817.25
35	46	1678.58	3833.78	61	9971.91	13805.69
36	43	428.71	2521.47	59	9441.77	11963.24
37	42	879.16	2874.80	55	9530.95	12405.75
38	45	843.36	2954.50	65	10445.18	13399.68
39	41	1236.40	3168.53	61	10427.22	13595.75
40	45	901.14	3016.60	63	10144.83	13161.43
41	43	833.73	2923.79	56	9654.74	12578.53
42	39	744.49	2567.65	61	10497.92	13065.57
43	41	1163.61	3118.19	63	10600.25	13718.44
44	40	858.58	2702.89	56	9719.14	12422.03
45	40	932.35	2863.62	55	9589.14	12452.76

Table 4.6: Table 6 Summary table of the results achieved

Id problem	Vehicles	Km transfer	Drivers	Working time	z	$Gap\%$	$\mu_{Gap\%}$
1	11	295.40	12	89.97	2945.15	2.68	7.36
2	16	411.38	16	123.81	4049.44	3.28	7.45
3	14	403.72	15	102.95	3633.22	4.13	8.97
4	19	829.31	26	172.99	6109.22	9.07	12.12
5	22	649.29	25	170.52	5954.52	7.20	10.80
6	20	423.53	22	162.75	5251.05	7.57	11.56
7	22	588.20	24	184.62	5934.40	13.03	15.64
8	21	558.59	22	161.27	5421.30	3.61	8.57
9	22	449.20	26	196.58	6115.03	4.85	9.73
10	21	182.78	24	185.23	5485.05	2.23	5.12
11	23	171.41	24	173.45	5455.91	3.13	7.90
12	23	604.14	24	159.49	5749.06	4.90	7.76
13	21	367.49	22	158.44	5201.86	9.02	12.15
14	22	619.04	19	149.82	5117.29	9.68	12.82
15	21	505.95	18	129.06	4646.57	3.92	8.51
16	14	858.20	16	118.97	4347.94	4.32	8.41
17	21	1373.72	23	182.32	6546.91	2.94	5.77
18	38	1806.05	44	326.43	11370.37	15.65	17.78
19	31	1192.13	36	275.87	9100.85	13.22	17.50
20	37	1734.24	44	299.58	10980.07	9.37	12.53
21	36	833.29	43	283.46	9767.94	3.74	6.22
22	32	1367.28	43	289.41	10161.40	12.75	15.20
23	27	995.50	39	296.90	9214.49	10.18	15.12
24	26	1411.18	33	225.55	8266.66	7.10	11.07
25	33	1190.99	38	279.02	9431.18	10.72	14.01
26	29	1451.93	33	248.76	8689.50	6.96	10.79
27	29	1472.32	32	248.35	8605.84	9.80	11.84
28	28	1472.02	34	233.13	8603.37	8.27	11.52
29	31	1217.91	38	256.18	9129.68	3.40	6.73
30	32	983.05	38	264.90	9032.03	9.75	12.13
31	35	303.83	58	392.64	11780.26	6.01	10.93
32	22	1284.78	48	352.22	10706.97	8.95	12.34
33	21	789.66	47	313.88	9678.51	4.70	7.20
34	43	2555.55	59	420.35	14809.06	7.18	12.09
35	46	1678.58	64	467.53	15053.91	9.04	13.27
36	43	428.71	62	409.42	12872.88	7.60	10.52
37	42	879.16	57	409.43	12773.47	2.96	6.05
38	45	843.36	68	472.85	14621.88	9.12	12.41
39	41	1236.40	63	459.56	14182.00	4.31	8.44
40	45	901.14	66	435.16	14102.78	7.15	9.58
41	43	833.73	59	394.71	12830.83	2.01	5.46
42	39	744.49	63	444.69	13441.41	2.88	7.53
43	41	1163.61	65	481.34	14527.06	5.89	8.98
44	40	858.58	58	404.72	12705.82	2.28	7.18
45	40	932.35	56	446.09	12993.25	4.34	6.91

Table 4.7: Table 7 Numerical results by using single local search operator

Id problem	z_0	Neighborhood	Vehicles	Km transfer	z^{MDVSP}	$Gap\%$	Δf
1	882.26	Nshift	11	314.13	864.13	9.52%	18.13
		Nswap	11	315.59	865.69	9.71%	16.57
		2opt*	11	299.09	849.09	7.61%	33.17
		Blocks	11	304.41	854.41	8.28%	27.85
8	882.26	Nshift	21	571.34	1621.34	11.06%	56.95
		Nswap	21	567.23	1617.23	10.83%	61.06
		2opt*	21	564.89	1614.89	10.87%	63.40
		Blocks	21	566.21	1616.21	10.78%	62.08
16	1590.14	Nshift	14	871.32	1571.32	0.84%	18.82
		Nswap	14	868.49	1568.49	0.66%	21.65
		2opt*	14	854.38	1564.38	0.40%	25.76
		Blocks	14	853.97	1563.97	0.37%	26.17
23	1590.14	Nshift	27	1047.68	2397.68	6.21%	60.58
		Nswap	27	1058.12	2408.12	6.67%	50.14
		2opt*	27	1011.98	2361.11	4.63%	96.28
		Blocks	27	1019.67	2319.67	4.97%	88.59
31	2131.84	Nshift	35	323.57	2073.57	2.30%	58.27
		Nswap	35	324.08	2074.08	2.32%	57.76
		2opt*	35	317.75	2067.75	2.01%	64.09
		Blocks	35	322.91	2072.91	2.27%	58.93
38	1590.14	Nshift	45	871.32	1571.32	5.65%	36.91
		Nswap	45	868.49	1568.49	5.55%	39.74
		2opt*	45	854.38	1564.38	5.07%	53.85
		Blocks	45	853.97	1563.97	5.06%	54.26

Chapter 5

Driving style analysis

5.1 Introduction

Improving traffic flow management is a hard challenge with high impact in many dimensions: road safety, economics, social life, environment protection and research. The management process requires methodologies and techniques for monitoring the flow, collecting data to analyze, extracting patterns from the latter and finally interpreting these patterns into valuable knowledge to exploit in decision-making for traffic control.

Nowadays, this process is performed through static traffic analysis, whose data are collected by sensors on road like traffic cameras, police patrols, air pollution detectors and so on. This methodology has a strong drawback: it can only provide a short-sighted, localized and biased vision of the road situation, being unable to reconstruct the vehicle trajectories and the drivers' behavior. Actually, one of the most crucial variables that determines the traffic flow dynamics is definitely the human factor. An appropriate use of transportation vehicles (compliant with the traffic code) and a sufficient amount of common sense are the essential ingredients to ensure traffic stream quality. This quality is supposed to be understood in a broad sense, i.e. in terms of overall efficiency of the road system, reduced toxic air pollutant emission and road safety. Conversely, illegal and aggressive driving behaviors can lead to harmful events, which, in some cases, can even be of dramatic nature.

In this study we focus on human behavior on the road. Particularly, we take the data collected by the AVL system implemented for the CORE project and described in 3 to provide an aggressiveness-based classification of the drivers. The importance of driving style classification is undoubted: costs related to accidents, to fuel consumption as well as tyres and other consumables are directly influenced by the way drivers behave. If one would endow transport operators with a AVL system, a driving style based ranking of the drivers could be one of the most interesting features in order to give the decision makers a way to sensitize drivers on a correct behaviour. An example of such tools is provided by TomTom fleet management products ([63]).

5.2 Related work

A large area of research is focusing on analyses of the human behavior's impact to the traffic flow quality. Parker et al. [52] measure the tendency of drivers committing violations via a questionnaire to the drivers (DBQ - Driver Behaviour Questionnaire). Their results relate the behaviour to information such as age, frequency of fast driving and thoroughness in decision making. Moreover, as reported in Chung and Wong [12] it is possible to relate the habitual driving styles to sociodemographic factors affecting drivers' lives and relate speeding and accidents to personality traits like aggressiveness, anxiety, misjudgement, distraction and unawareness. Brown and Groeger [5], Jonah [36] and Mayhew et al. [20] proposed a work, in order to prevent car accidents, identifying a set of factors emerging as predictors of accident involvement. These factors include age, experience, ability to detect hazards quickly and tendency towards risk taking. In this case, Decision Making Questionnaire (DMQ) and Driving Style Questionnaire (DSQ) were submitted to the drivers, and the result was obtained by looking at their answers and insurance historical data, e.g. car accidents. Moreover, Rigolli and Brady [60] conducted a study on the driving style by modelling an agent-based environment. The result was a complex system, analyzing drivers' behaviours, using a kinematic traffic simulator, perceptive agents and reasoning models. In Mitrovic [48], another approach to model this enviroment was presented. In that work a simple and reliable method for the recognition of driving events using hidden Markov models was presented, and used on data from real vehicles in a normal driving environment, the kind of driving event the sequence was identified. A driving style software architecture was developed by Meseguer et al. [33], adopting data mining and neural network techniques to analyze and generate a classification of the route followed. The purpose of the work was to assist drivers at correcting bad habits in their driving behaviour, while offering helpful tips to improve fuel economy.

However, it is possible to apply a data mining approach to the data. Constantinescu et al. [65] proposed an analytical approach, using cluster and principal component analysis from exploratory statistics to identify and explain drivers by grouping them according to their driving behaviour. In all the works using real data, a preprocessing step was necessary for aligning a sequence of observed car positions with the road network on a digital map. A global map-matching algorithm for low-sampling-rate points was proposed in Lou et al. [44]. Studying the traffic environment, drivers behaviour and road conditions could allow us to improve the quality of the transportation service in terms of reducing time spent onboard, decreasing the number of car accident, and improving traffic safety and fuel economy. A correct driving behaviour, in fact, may lead to a safer driving as well as to a greener transport system, as pointed out by Akena et al. [1].

The next sections are organized as follows: Section 5.3 presents the data and then describes the data preprocessing and cleaning phase. Section 5.4 outlines the procedure used, section 5.5 as the results and in Section 5.6 some hints for future works are provided.

5.3 Data Presentation and Preprocessing

5.3.1 Data Presentation

The data is provided by a GPS tracking system that is part of the CORE system as described in 3.5. Data collected between October 2014 and March 2015 have been used to feed the present study. The data consist of:

- 11489592 GPS samples;
- 1196 different journeys;
- 1249 different tracked vehicles;
- 29 distinct companies divided into 6 different consortia;

Data is stored in packet_ext table and consist of the following fields:

- Id: an identifier of each row in table packet_ext;
- accuracy: GPS accuracy value (meters);
- flags: set of booleans stating if GPS point has low accuracy because there was no GPS signal at all at the moment of transmission;
- heading: vehicle direction in terms of angle with respect to the geographic north;
- journeyOrdinal: indicates the occurrence of the journey in the case it is recurring during the same day (journey with a frequency scheme);
- altitude: self descriptive;
- latitude: self descriptive;
- longitude: self descriptive;
- parameter: extra parameter associated with different packets, accoding to the value of data. It has not been used in this study;
- speed: actual speed of the vehicle;
- timestamp: date and time of the packet;
- data: code of the activity (open door, close door, start run, ...);
- vehicleJourneyID: code of the vehicle journey;

- version: the version of the transmission protocol associated with this packet. It has been introduced in order to support different implementation of the transmission protocol;
- receivedAt: timestamp of the packet receiving time;
- senderFingerprint: vehicle identifier;
- $certification\_id$: identifier of this specific run. Thus every same route run at different times, will have a different $certification\_id$. In other words, the $certification\_id$ is the identifier of the certification as results from the algorithm proposed in 3.5.4;

The dataset has been processed in order to obtain new columns which could support the present study. These new columns:

- ds: difference in terms of speed between this packet and the last one in the same journey;
- dt: difference in terms of timestamp between this packet and the last one in the same journey;
- Ec: value of the kinetic energy;
- prevspeed: speed of the previous packet;
- prevtime: timestamp of the previous packet;

5.3.2 Data Preprocessing

After collecting the data, we obtained the speed limit on the road for each GPS sample. In order to do that, we built a Java code using QGIS and the GeoTools library. This allowed to add new data to the packet_ext table:

- speedMinDistanceFeature: speed limit, obtained by retrieving the value from the shapefile;

Then another table is created, journey_data. Each tuple is a resume of a journey and thus is related to a set of packets stored into the packet_ext table. It has the following fields:

- certification_id: as above;
- vehicleJourneyID: as above;
- senderFingerprint: as above;
- Vmn: average speed kept during the route;
- Vsd: speed standard deviation during the route;

- ACBRsd: standard deviation calculated on the acceleration and the braking value (only positive values);
- ACmn: average acceleration (only positive values);
- ACsd: standard deviation on acceleration;
- BRmn: average braking (only negative values);
- BRsd: standard deviation on braking;
- V50: % of packets whose speed is above the speed limit;
- W: value of work, calculated by the physics formula;

The study also requires the day of the week and the moment of the day a journey has been run. Thus we added the following columns by extracting the relevant data from the *timestamp* field. Moreover we added the *points* field in order to support a study based on the speed limit violation. The formal definition of this field will be provided in the next paragraph.

5.3.3 Definitions of aggressiveness in driving style

The tables described above provide the basis for our experiments. After the preprocessing phase, it became necessary to define what would be an *aggressive driving style*. Based on the analysis of the journey components, and by performing some empirical tests, we decided to use three different definition for aggressiveness:

- Accelerating/Braking-based: based jointly on $BRmn$, $BRsd$, $ACBRsd$, $ACmn$, $ACsd$;
- Speed-based: based on Vmn, Vsd and $V50$;
- Score-based: similar to the speed-based but based on a scoring function assigned to each journey stored in the field *points* of the *journey_data* table. (described below);

Once defining the notion of aggressiveness, we could start mining informations from the data, deriving an interpretation from the result of PCA, varimax factor rotation and the application of data mining techniques such as clustering. The process and the results of these various experiments are discussed in the next sections. Moreover, the *points* field has been computed for each journey. The second definition, focuses more on the concept of exceeding speed limit. We considered that a GPS sampling that shows a speed slightly over the limit should be evaluated with a different weight rather than another one with a more excessive speed. For this reason, a new field was added to the previous, representing the score that we attached to the GPS sample by looking at the difference between the real speed of the vehicle and the speed limit. The score was calculated in the following way: Let i = (speed-speedMinDistanceFeature)/5. We give to a packet the score = Fibonacci(i), where Fibonacci(i) is the i-th number in Fibonacci sequence. Using this approach we could be sure that there will be an exponential increase in the score as the magnitude of the speed limit violation increases. This allows us to emphasize a big violation in terms of speed limits, which is characteristic of an aggressive behaviour.

5.4 Data Analysis

5.4.1 An algorithm for extracting driving style

Due to its multivariate nature, the first issue is to obtain an in-depth understanding of the data. We have to concisely consider (in terms of aggressiveness) not only the speed, but even the acceleration, braking, and the speed limit violations. In order to do that, we performed a Principal Component Analysis, taking in account 5 principal components. This was followed by a Varimax Factorial Rotation. After using the PCA on the data, we could obtain the 5 principal components by calculating the eigenvalues and eigenvectors. In Figure 5.1 we observe the correlation matrix.

Axis	Eigen value	Difference	Proportion (%)	Histogram	Cumulative (%)
1	3,916793	1,740299	48,96 %		48,96 %
2	2,176495	1,380073	27,21 %		76,17 %
3	0,796421	0,220319	9,96 %		86,12 %
4	0,576103	0,270759	7,20 %		93,32 %
5	0,305344	0,156958	3,82 %		97,14 %
6	0,148386	0,083185	1,85 %		98,99 %
7	0,065201	0,049944	0,82 %		99,81 %
8	0,015257	-	0,19 %		100,00 %
Tot.	8,000000	-	-	-	-

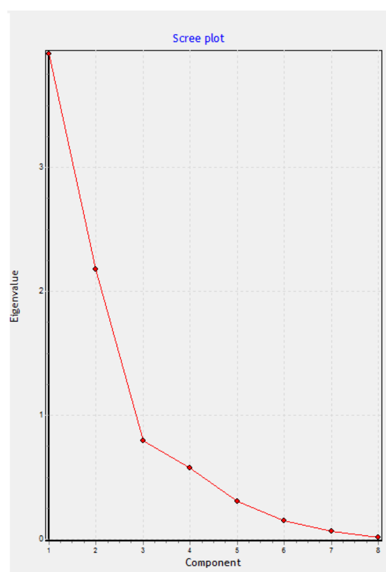


Figure 5.1: Principal Component Analysis

After that, we performed a Varimax Factorial Rotation, taking into account 3 factors. In that way, we could establish a correlation between the old variables and the new ones. In fig. 5.2 we show the results of the application of Varimax Rotation. It is clear that $BRsd$, $BRmn$, $ACBRsd$, $ACsd$ and $ACmn$ on PCA_Axis_1 are correlated, as well as Vmn, Vsd and $V50$ on PCA_Axis_2. In this way, we could assume the acceleration and braking variables compose one

Attribute	Mean	Std-dev	Axis_1	Axis_2	Axis_3	Axis_4	Axis_5
Vmn	32.2905913	12.4196646	0.1316667	0.6113754	0.0630361	0.2359958	0.1123461
Vsd	20.8093221	7.5193311	-0.1898655	-0.5178077	0.0975365	-0.1835100	-0.0819536
ACBRsd	0.4849852	0.4628895	-0.4837206	0.1517554	0.0195691	0.0076144	0.2717405
ACmn	0.3175414	0.1779660	0.4238720	0.0129082	0.4669806	0.00402444	0.5831392
ACsd	0.3225260	0.1185958	-0.1129130	0.0812103	-0.5657193	-0.0676660	0.3621565
BRmn	0.3662041	0.2234597	-0.4127073	0.1274781	0.4914057	0.0845358	-0.5163601
BRsd	0.3941272	0.5298498	0.4239861	0.174012	0.4544758	0.0054270	0.4010644
V50	34.1467578	13.7145419	0.1112348	0.4993065	0.0019482	0.8349027	0.0889524

Table 5.1: Factor Score Coefficients.

Unrotated Factor Loadings

Attribute	Axis_1		Axis_2		Axis_3	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)	Corr.	% (Tot. %)
BRsd	-0,83911	*70 % (70 %)*	0,25670	*7 % (77 %)*	0,40554	*16 % (93 %)*
BRmn	-0,81678	*67 % (67 %)*	0,18807	*4 % (70 %)*	0,43854	*19 % (89 %)*
ACBRsd	-0,95733	*92 % (92 %)*	0,22388	*5 % (97 %)*	0,01746	*0 % (97 %)*
Vmn	-0,26058	*7 % (7 %)*	-0,90196	*81 % (88 %)*	0,05625	*0 % (88 %)*
Vsd	-0,37576	*14 % (14 %)*	-0,80818	*65 % (79 %)*	0,08704	*1 % (80 %)*
V50	-0,22014	*5 % (5 %)*	-0,73662	*54 % (59 %)*	-0,00174	*0 % (59 %)*
ACsd	-0,81719	*67 % (67 %)*	0,12428	*2 % (68 %)*	-0,50489	*25 % (94 %)*
ACmn	-0,83888	*70 % (70 %)*	0,01904	*0 % (70 %)*	-0,41674	*17 % (88 %)*
Var. Expl.	3,91679	*49 % (49 %)*	2,17649	*27 % (76 %)*	0,79642	*10 % (86 %)*

Rotated Factor Loadings

Attribute	Axis_1		Axis_2		Axis_3	
-	Corr.	% (Tot. %)	Corr.	% (Tot. %)	Corr.	% (Tot. %)
BRsd	0,92444	*85 % (85 %)*	0,00165	*0 % (85 %)*	0,28260	*8 % (93 %)*
BRmn	0,91379	*84 % (84 %)*	0,06486	*0 % (84 %)*	0,23582	*6 % (89 %)*
ACBRsd	0,74154	*55 % (55 %)*	0,03033	*0 % (55 %)*	0,64506	*42 % (97 %)*
Vmn	-0,00063	*0 % (0 %)*	0,93999	*88 % (88 %)*	0,03172	*0 % (88 %)*
Vsd	0,12411	*2 % (2 %)*	0,88172	*78 % (79 %)*	0,09539	*1 % (80 %)*
V50	-0,02707	*0 % (0 %)*	0,76549	*59 % (59 %)*	0,06613	*0 % (59 %)*
ACsd	0,27075	*7 % (7 %)*	0,04590	*0 % (8 %)*	0,92884	*86 % (94 %)*
ACmn	0,31881	*10 % (10 %)*	0,16031	*3 % (13 %)*	0,86627	*75 % (88 %)*
Var. Expl.	2,43057	*30 % (30 %)*	2,27993	*28 % (59 %)*	2,17922	*27 % (86 %)*

Figure 5.2: Varimax Factorial Rotation

component, and all the variables related to the speed compose another component. With this two principal components we can provide a good correlation between a large part of variables.

5.4.2 Driving Behaviour Classification

For each definition of aggressiveness provided in section 5.4.2, three classes have been used in order to characterize drivers' behaviour. For the acceleration-based aggressivenes we define the following behavior classes. Constant values are empirically calculated by interviewing passengers on some test journeys made with the purpose of calculating these values:

- Very Aggressive: having a value of $BRmn$, $BRsd$, $ACBRsd$, $ACmn$, $ACsd$ greater than or equal to $3(km/h)/s$;
- Middle Aggressive: having a value of $BRmn$, $BRsd$, $ACBRsd$, $ACmn$, $ACsd$ greater than or equal to $1.5(km/h)/s$ but less than $3(km/h)/s$;
- Low Aggressive: having a value of $BRmn$, $BRsd$, $ACBRsd$, $ACmn$, $ACsd$ less than $1.5(km/h)/s$.

Definition of aggressiveness	Categories	# of routes	Percentage
Acceleration-Braking based	Very Aggressive	2550 routes	1.53 %
	Middle Aggressive	4007 routes	2.4 %
	Low Aggressive	160000 routes	96.06 %
Speed based	Very Aggressive	37794 routes	22.69 %
	Middle Aggressive	6998 routes	4.2 %
	Low Aggressive	121765 routes	73.1 %
Score based	Very Aggressive	29010 routes	17.41 %
	Middle Aggressive	30999 routes	18.61 %
	Low Aggressive	106548 routes	63.97 %

Table 5.2: Quantification of the number of routes among all the aggressiveness' definitions.

In the same way, speed-based aggressiveness can be split into 3 categories (constant values are empirically calculated):

- Very Aggressive: having a value of Vmn near $30km/h$, Vsd near $20km/h$ and $V50$ greater than 40;
- Middle Aggressive: having a value of Vmn near $25km/h$, Vsd near $15km/h$ and $V50$ greater than 25 but less than 40;
- Low Aggressive: less than 25.

Finally, the score-based can be split into 3 different categories (constant values are empirically calculated):

- Very Aggressive: having a value of $score$ higher than 2.5;
- Middle Aggressive: having a value of $score$ between 1 and 2.5;
- Low Aggressive: having a value of $score$ lower than 1.

In the next section these classes are applied to clustering and statistical analysis.

5.5 Results

The data have been categorized according to the aggressiveness classifications in section 5.4.2 and recorded in table 5.2.

As one can see, drivers are generally categorized as Low Aggressive with respect to all the definitions.

Regarding the attributes listed in 5.3, a statistical analysis is performed. The idea is to detect any interesting trend during the week day or in a particular time slot. In order to do this, the dataset was partitioned into 4 partitions:

- A: all data related to routes that are performed everyday;
- B: all data related to routes that are performed 6 day per week (Monday to Saturday);

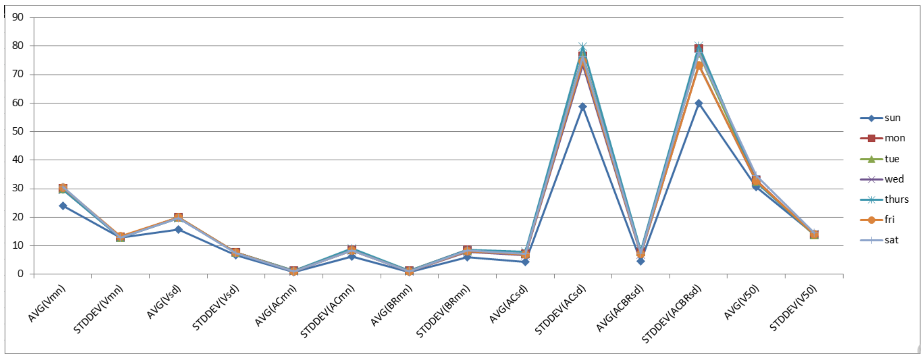


Figure 5.3: Attributes on dataset A per day

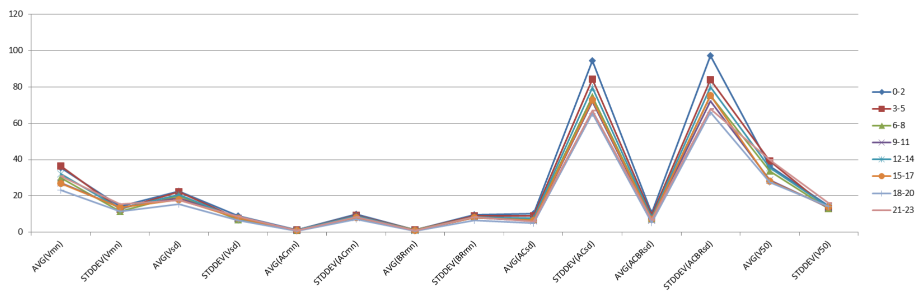


Figure 5.4: Parameters on dataset A per time slot

- C: all data related to the nightly routes (time slots 3-5 in fig 5.4 time slots and 0-5 in fig. 5.5).

From the analysis of the dataset A (fig. 5.3), one can notice that on Sunday the values of Vmn and $V50$ are significantly lower than the other days. This evident behaviour should be further explored as it does not directly reflect the traffic conditions registered on Sundays. A similar behaviour is registered with respect to accelerations ($ACBRsd$, $ACsd$ and $ACmn$). This behaviour should be further analyzed with respect other factors (i.e. social ones) in order to be fully understood. Our preliminary conclusiong is that these observations are due to the lower traffic level, as vehicles may proceed without accelerating (or breaking) too often.

The 24 daily hours are split in 8 slots (3 hours each). The 1st time slot is between 0.00am and 2.59am, the second one is between 3.00am and 5.59 am and so on. The result of the analysis with respect to time slots show an interesting trend in speed during late evening and early morning hours. In fact, $AVG(50)$ and $AVG(Vmn)$ are higher in these time slots (see fig. 5.4).

The third experiment on the dataset A tried to extract associative rules using the APRIORI algorithm, after applicating a DISCRETIZE filter on it (using default parameters). The results showed us that there are no associative

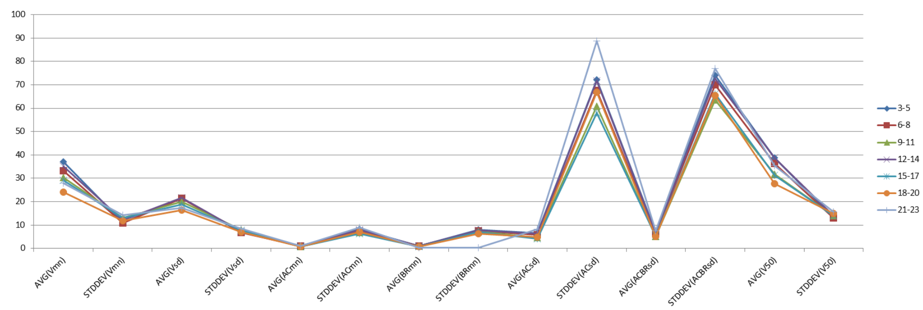


Figure 5.5: Comparing parameters on dataset B

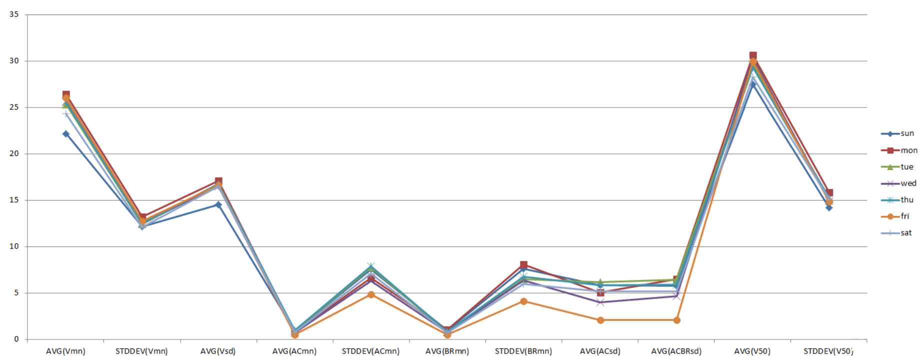


Figure 5.6: Comparison between days on dataset C

rules between the fields day, hour and $V50$, although there is a strict correlation between $BRmn$, $ACmn$, $ACsd$, $BRsd$ and $ACBRmn$, W.

Finally we tried to make clustering experiments by applying expectation maximization (fig. 5.7) to the whole dataset. The Y-axis is the score-based aggressiveness while the X-axis represents the mean value of accelerations and breakings. Low aggressiveness cluster are identifiable but no other natural clustering could be observed.

5.6 Conclusion

CORE is a valuable source of data that can feed studies related to the driver behaviour. Providing the Regione Calabria and the different operators with a tool to monitor drivers' behaviour could result in an improvement of the service. This would lead to:

- improved road safety;
- better quality perceived by commuters as the journey is safer and smoother;
- lower pollution;

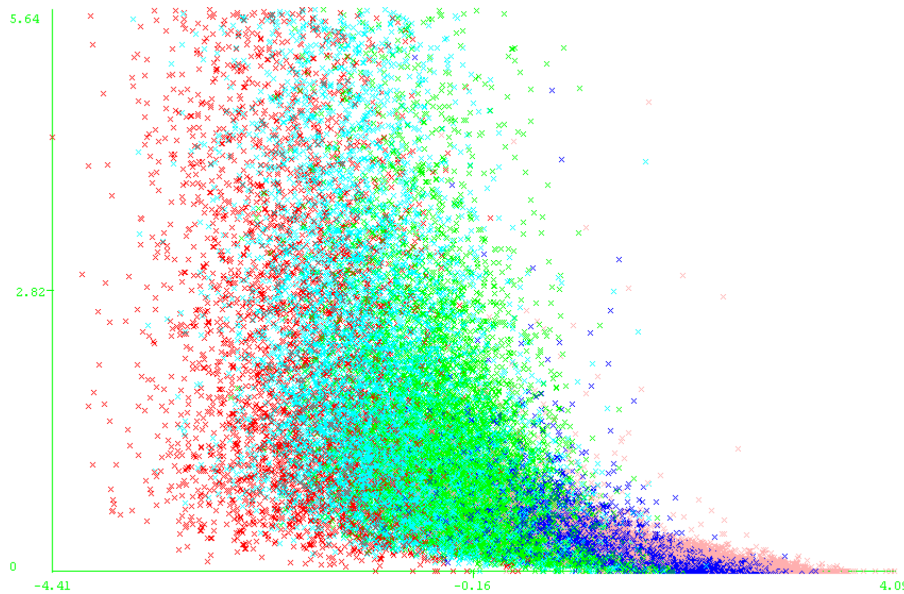


Figure 5.7: Clustering for speed-based aggressiveness acceleration

- longer durability of consumable components (i.e. tyres).

The study presented in this chapter may be further developed e.g. by the application of outlier detection algorithm and by the correlation between the results that have been presented and other factors, such as social ones. Future works may be start from the method proposed by Lei et al. [39], who estimate driving behaviour by analizing GPS samples with no need of any other sensor. This could be particularly useful in the context proposed in chapter 3 as bus are not endowed with any other sensor than GPS. As stated in the introduction, driving style based ranking may be one of the striking features of any Automatic Vehicle Monitoring application. By providing insights into the way drivers behave, they could be pursued to a safer and greener drive. The benefits obtained by the introduction of such applications may be measured in several ways which are out of the scope of this chapter and are up to the strategies one wish to enforce. For instance, several KPI may be introduced, related to fuel or consumables savings in a time period.

Chapter 6

Conclusion and future work

This work points out some important results and the main advantages deriving by the implementation of ITS, which are extensively described in chapter 2. Acknowledging this, Regione Calabria, together with University of Calabria, has set up an ITS aimed at providing the following functionalities:

- Infomobility;
- Data management;
- Service certification.

The CORE project resulted in the implementation of a software platform consisting of a website (http://calabria.coretpl.it), an AVL system and a distributed layer composed by a number of on-board units, as described in chapter 3.

CORE has introduced changes in the public transport system of Calabria: before its adoption there was no means to track vehicles or to provide consistent information to commuters. It also facilitates the relationship between the Regione Calabria officers and the consortia delivering public transport service on behalf of Regione, as pointed out in section 3.6.

At the time of writing more than 95% of the extra-urban service is loaded into CORE database and more than 70% of vehicles are monitored everyday. This number is expected to increase to at least 90% in the next year.

Furthermore CORE may support a plethora of applications allowing to boost the public transport system. These may be addressed by future releases of the platform. Here follows a list of the most relevant ones:

- Service Oriented Architecture (SOA) implementation. The overall architecture may be extended in order to provide the same functionalities via a SOA. CORE services will become accessible to developers; they will thus support a number of third parties applications. This could lead to the development of a software ecosystem based on a centralized and consistent data source while providing heterogeneous services customized to their particular purpose and context of application;

- Departure boards. As shown in section 3.6, a departure board has been set up in the Lamezia Terme International Airport. This could be the first of a series of infomobility boards that could be installed at critical nodes of the transport network;
- Mobile applications. CORE currently provides only a website as end user interface. As web is nowadays mobile first, one or more mobile applications could be implemented to support mobile browsing and to effectively exploit mobile's capabilities, such as GPS and/or other sensors;
- Route planning. One of the most critical application regarding infomobility applied to public transport is the multimodal route planning. A multimodal route planner allows user to plan a journey from an origin to a destination without no prior knowledge about the transport system. A single journey may be composed by several legs using different modes of transport. The introduction of this kind of applications is strongly supported by the European Commission, as demonstrated by the large number of funded projects. MODUM ([16]) and eCOMPASS ([14]), for example, implemented routing algorithms and multimodal planners that have been integrated in a more comprehensive system that has been piloted into 3 european cities within the scope of the HoPE Project ([15]), funded within the 7th Framework Program.

CORE is currently being refined and some features are being changed in order to comply with new specifications. As soon as it becomes stable, it would be the first Italian platform of its kind and could become a best practice to refer to, not just from a software perspective but also processe-wise.

Moreover, data collected may support scientific studies aimed at improving transport system. The present work introduces vehicles and crews shift scheduling studies and driving behaviour analysis. The first study came up with an integrated algorithm for solving both the crews and shift scheduling for public transport companies and could be an interesting starting point for building industrial applications in this field (4). The second study arises from the need to control drivers' behaviour in order to increase security, quality and optimize resources exploitation. In 5 data collection and data processing is performed togheter with statistical analysis on the driving style in Calabria. Some definitions of aggressiveness are provided and experimented on the available dataset coming from the AVL system described in 3.5.

As more and more data keeps being collected, this could result in other studies leading to new industrial applications.

Finally, surveys and interviews will be submitted in the next future in order to measure KPIs. Customer satisfaction, customer retention and loyalty are expected to increase thanks to the introduction of CORE.

Bibliography

[1] Robert Akena, Felix Schmid, and Michael Burrow. Driving style for better fuel economy. *Proceedings of the Institution of Civil Engineers - Transport*, 0(0):1–9, 0.

[2] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.

[3] John E Beasley and B Cao. A tree search algorithm for the crew scheduling problem. *European Journal of Operational Research*, 94(3):517–526, 1996.

[4] Jin-Kao Hao Benot Laurent. Iterated local search for the multiple depot vehicle scheduling problem. *Computers and Industrial Engineering*, 57(1):277–286, 2009.

[5] C. G. BOTTOM and R. ASHWORTH. Factors affecting the variability of driver gap-acceptance behaviour. *Ergonomics*, 21(9):721–734, 1978.

[6] Juergen Branke, Christian Schmidt, and Markus Withopf. A fast look-ahead heuristic for the multi-depot vehicle routing problem. In *Wirtschaftsinformatik (2)*, pages 411–428. Citeseer, 2007.

[7] Olli Bräysy and Michel Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation science*, 39(1):104–118, 2005.

[8] Giorgio Carpaneto, Mauro Dell'Amico, Matteo Fischetti, and Paolo Toth. A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, 19(5):531–548, 1989.

[9] P Carraresi and G Gallo. Network models for vehicle and crew scheduling. *European Journal of Operational Research*, 16(2):139–151, 1984.

[10] Luís Cavique, César Rego, and Isabel Themido. Subgraph ejection chains and tabu search for the crew scheduling problem. *Journal of the Operational Research Society*, pages 608–616, 1999.

[11] CEN. Transmodel implementations in italy. http://transmodel-cen.eu/?page_id=593.

[12] Yi-Shih CHUNG and Jinn-Tsai WONG. Investigating driving styles and their connections to speeding and accident experience. *Journal of the Eastern Asia Society for Transportation Studies*, 8:1944–1958, 2010.

[13] Ross Clement and Anthony Wren. *Greedy Genetic Algorithms, Optimizing Mutations and Bus Driver Scheduling*, pages 213–235. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

[14] European Commission. ecompass - eco-friendly urban multi-modal route planning services for mobile users. http://www.ecompass-project.eu/.

[15] European Commission. Hope - holistic personal eco-mobility. http://www.hope-project.eu/.

[16] European Commission. Modum - models for optimizing dynamic urban mobility. http://modum-project.eu.

[17] European Commission. *WHITE PAPER Roadmap to a Single European Transport Area Towards a competitive and resource efficient transport system.* European Commission, 2011.

[18] Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 1997.

[19] Benoit Crevier, Jean-François Cordeau, and Gilbert Laporte. The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773, 2007.

[20] H. M. Simpson D. R. Mayhew, R. A. Warren and G. C. Haas. Young driver accidents: Magnitude and characteristics of the problem. pages 27–33, 1996.

[21] Mauro Dell'Amico, Matteo Fischetti, and Paolo Toth. Heuristic algorithms for the multiple depot vehicle scheduling problem. *Management Science*, 39(1):115–125, 1993.

[22] Ministero delle infastrutture e dei trasporti. *Conto Nazionale delle Infrastrutture e dei Trasporti.* Istituto Poligrafico e Zecca dello Stato S.p.A, 2015.

[23] Guang-Feng Deng and Woo-Tsong Lin. Ant colony optimization-based algorithm for airline crew scheduling problem. *Expert Systems with Applications*, 38(5):5787–5793, 2011.

[24] Martin Desrochers and Francois Soumis. A column generation approach to the urban transit crew scheduling problem. *Transportation Science*, 23(1):1–13, 1989.

[25] Istituto Nazionale di Statistica. Transport and mobility survey. https://www.istat.it/storage/politiche-sviluppo/Trasporti_mobilit.xls.

[26] Eurostat. *Panorama of Transport*. Luxembourg: Office for Official Publications of the European Communities, 2007.

[27] Sarah Fores, Les Proll, and Anthony Wren. *Experiences with a flexible driver scheduler*. Springer, 2001.

[28] Richard Freling, Dennis Huisman, and Albert P. M. Wagelmans. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6(1):63–85, 2003.

[29] Keivan Ghoseiri, Ferenc Szidarovszky, and Mohammad Jawad Asgharpour. A multi-objective train scheduling model and solution. *Transportation Research Part B: Methodological*, 38(10):927–952, 2004.

[30] Ali Haghani, Mohamadreza Banihashemi, and Kun-Hung Chiang. A comparative analysis of bus transit vehicle scheduling models. *Transportation Research Part B: Methodological*, 37(4):301–322, 2003.

[31] Karla L Hoffman and Manfred Padberg. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39(6):657–682, 1993.

[32] Dennis Huisman, Richard Freling, and Albert PM Wagelmans. Multiple-depot integrated vehicle and crew scheduling. *Transportation Science*, 39(4):491–502, 2005.

[33] J. C. Cano J. E. Meseguer, C. T. Calafate and P. Manzoni. Characterizing the driving style behaviour using artificial intelligence techniques. IEEE LCN, 2013.

[34] Ming-Shen Jian and Ta-Yuan Chou. Multiobjective genetic algorithm to solve the train crew scheduling problem. In *Proceedings of the 10th WSEAS International Conference on Systems Theory and Scientific Computation*, ISTASC'10, pages 100–106, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS).

[35] David S Johnson, Cecilia R Aragon, Lyle A McGeoch, and Catherine Schevon. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations research*, 37(6):865–892, 1989.

[36] Brian A. Jonah. Accident risk and risk-taking behaviour among young drivers. *Accident Analysis and Prevention*, 18(4):255–271, 1986.

[37] Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986, 1984.

[38] Natalia Kliewer, Taieb Mellouli, and Leena Suhl. A time–space network based exact optimization model for multi-depot bus scheduling. *European journal of operational research*, 175(3):1616–1627, 2006.

[39] Yulong Lei, Ke Liu, Yao Fu, Xingzhong Li, Zhenjie Liu, and Shaohua Sun. Research on driving style recognition method based on drivers dynamic demand. *Advances in Mechanical Engineering*, 8(9):1687814016670577, 2016.

[40] Jan Karel Lenstra and AHG Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

[41] David Levine. Application of a hybrid genetic algorithm to airline crew scheduling. *Computers & Operations Research*, 23(6):547–558, 1996.

[42] Chang Liu, Fei Song, Huan Yan, and Sidong Zhang. Modeling of multipath transport. *International Journal of Computer Science Issues*, 10(3):458–467, 2013.

[43] Andreas Löbel. Solving large-scale multiple-depot vehicle scheduling problems. In *Computer-aided transit scheduling*, pages 193–220. Springer, 1999.

[44] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. Map-matching for low-sampling-rate gps trajectories. In *ACM SIGSPATIAL GIS 2009*. ACM SIGSPATIAL GIS 2009, November 2009.

[45] Helena R Lourenço, José P Paixão, and Rita Portugal. Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation Science*, 35(3):331–343, 2001.

[46] Roy E Marsten and Fred Shepardson. Exact solution of crew scheduling problems using the set partitioning model: Recent successful applications. *Networks*, 11(2):165–177, 1981.

[47] Marta Mesquita and José Paixão. *Exact Algorithms for the Multi-Depot Vehicle Scheduling Problem Based on Multicommodity Network Flow Type Formulations*, pages 221–243. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[48] D. Mitrovic. Reliable method for driving events recognition. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):198–205, June 2005.

[49] UK Department of Transport. Naptan. `http://naptan.dft.gov.uk/`.

[50] United States Department of Transportation. Vehicle infrastructure integration. `http://www.its.dot.gov/research_archives/vii/index.htm`.

[51] Amar Oukil, Hatem Ben Amor, Jacques Desrosiers, and Hicham El Gueddari. Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. *Computers & Operations Research*, 34(3):817–834, 2007.

[52] Dianne Parker, Robert West, Steve Stradling, and Antony S.R. Manstead. Behavioural characteristic and involvment in different types of traffic accident. *Accident Analysis and Prevention*, 27(4):571–581, 1995.

[53] European Parliament and Council. On the framework for the deployment of intelligent transport systems in the field of road transport and for interfaces with other modes of transport. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2010:207:0001:0013:EN:PDF.

[54] Eric Pels and Piet Rietveld. *COST FUNCTIONS IN TRANSPORT. IN: HANDBOOK OF TRANSPORT MODELLING*. Elsevier, 2000.

[55] Ann-Sophie Pepin, Guy Desaulniers, Alain Hertz, and Dennis Huisman. Comparison of heuristic approaches for the multiple depot vehicle scheduling problem. Technical report, Erasmus School of Economics (ESE), 2006.

[56] Progetto Infocity Programma Elisa. Indagine sull'infomobilità: I sistemi di trasporto intelligenti (its) negli enti pubblici del programma elisa.

[57] Samuel Raff. Routing and scheduling of vehicles and crews: the state of the art. *Computers & Operations Research*, 10(2):63–211, 1983.

[58] Celso C Ribeiro and François Soumis. A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42(1):41–52, 1994.

[59] Celso C. Ribeiro and Franois Soumis. A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research*, 42(1):41–52, 1994.

[60] Marco Rigolli and Michael Brady. Towards a behavioural traffic monitoring system. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 449–454, New York, NY, USA, 2005. ACM.

[61] Shashi Shekhar and Hui Xiong. *Automated Vehicle Location (AVL)*, pages 37–37. Springer US, Boston, MA, 2008.

[62] SkySet spa. Viabilità sicura - sorpassometro. http://www.skyset.com/dynamic/engine.aspx?lang_id=1&cont_id=215&ARG=SO.

[63] TomTom. Tomtom - green and safe driving. https://telematics.tomtom.com/en_gb/webfleet/fleet-management/green-and-safe-driving/.

[64] Tallys H. Yunes, Arnaldo V. Moura, and Cid C. de Souza. Solving very large crew scheduling problems to optimality. In *Proceedings of the 2000 ACM Symposium on Applied Computing - Volume 1*, SAC '00, pages 446–451, New York, NY, USA, 2000. ACM.

[65] C. Marinoiu Z. Constantinescu and M. Vladoiu. Driving style analysis using data mining techniques. *International Journal of Computers, Communications & Control*, 5(5):554–663, 2010.