

UNIVERSITY OF MODENA AND REGGIO EMILIA

Department of Sciences and Methods for Engineering

Doctorate School in
INDUSTRIAL INNOVATION ENGINEERING

XVIII Doctorate Period

Socially Aware Mobility Services

Ph.D. Candidate
Eng. ANDREA SASSI

Research Advisor
Prof. Eng. FRANCO ZAMBONELLI

School Coordinator
Prof. Eng. MAURO DELL'AMICO

March 21, 2016

Abstract

Nowadays, urban environments are widely populated by ICT devices seamlessly connected to the Internet. Mainly pushed by the Internet of Things paradigm, widespread sensor networks, actuators, and computational units constitute the Smart City ecosystem with the aim of addressing everyday needs with innovative solutions that pervade the whole city with collective sensing, awareness, and actions.

The dense, distributed, heterogeneous, and networked infrastructure deployed within the city is smoothly integrated with people and their personal mobile devices, by shaping a large-scale socio-technical Superorganism. This thesis has the main aim of supporting collaborative urban mobility of social communities that collect and spread mobility data through Socially Aware Mobility Services, by exploiting the new technological scenario.

The environment will be adapted through the coordination and solicitation of its infrastructural components to best suit people's needs at different scales, and so to serve at the same time individual-level and urban-level goals. For example, the aim of a parking infrastructure enriched by real-time occupancy can be dual. On one hand, individuals can know in advance where to park and they do not lose time in seeking for an available spot. On the other hand, the consequences of not losing time in cruising for parking can be measured as a potential global reduction of traffic congestions and air pollution.

A general-purpose coordination framework for the sensing, computing and actuating infrastructure is proposed to effectively provide socially aware mobility services to individuals and to the city as a whole. The infrastructural requirements and challenges of superorganisms are presented, and a set of metrics are defined to classify the enabled services with regard to their active contributions in the collective awareness generation, along all the phases of the sensing, actuating, computing feedback loop.

A couple of exemplar scenarios related to the deployment of socially aware mobility services, influenced by social aspects and interactions, are studied: Ride Sharing and Crowd Steering.

When a Ride Sharing service is put in place, both infrastructural and social requirements have to be satisfied. This work brings its contributions to Ride Sharing

in (i) proposing a data-driven methodology to infer mobility traces and traffic flows, (ii) estimating its potential application in terms of mobility reshaping over the road network, and (iii) evaluating the effects of non-monetary rewards, such as social benefits, on the penetration of the service. Three Ride Sharing use cases are presented and evaluated.

For the sake of avoiding overcrowded situations, the capabilities of superorganisms can enable the steering of pedestrians moving in a crowd, through the provisioning of mobility recommendations.

In this work, several Crowd Steering strategies both for indoor and outdoor scenarios are depicted. The classification of the strategies is proposed according to (i) the local or global level of information taken into account, (ii) the local or global nature of the pursued goals, and (iii) the unicast or multicast communications put in place to spread recommendations across the crowd. Any Crowd Steering strategy can have consequences both at individual scale and at a global one. Simulation-based and data-driven experiments are reported to show that Crowd Steering can effectively drive the crowd towards a desired direction or situation, by providing individual benefits, such as a reduced walking time, and global ones, such as a less dense, evenly distributed, crowd. Finally, the design and the development of two Crowd Steering real case systems are presented.

Despite the promising experimental results, the effectiveness of socially aware mobility services is threatened by several infrastructural and functional challenges. Open issues and future research perspectives conclude this work.

Contents

1	Introduction	1
1.1	The ICT Enabled Smart City Scenario	1
1.2	Urban Superorganisms	2
1.2.1	Vision	4
1.2.2	Challenges	5
1.2.3	Mission	9
1.3	The MUCCA project	9
1.3.1	Needs	10
1.3.2	Proposed Solutions	10
1.3.3	Motivations	11
1.4	Contributions of this Work	12
2	Collective Awareness for Socially Aware Mobility Services	15
2.1	Awareness Feedback Loop	15
2.1.1	Superorganism Capabilities	16
2.1.2	Impacts on Urban Life	17
2.1.3	Individual and Global Support	19
2.1.4	Human - ICT Interactions	19
2.2	Socially Aware Mobility Services (SAMS)	21
2.2.1	Exemplar SAMS	23
2.2.2	A Classification Approach	25
2.3	A General-Purpose Coordination Framework	28
2.3.1	The Architecture	29
2.3.2	Open Architectural Challenges	33
2.3.3	Exemplar Case Study	35
2.4	State of the Art	39
2.5	Conclusions	41
3	SAMS: Ride Sharing	43
3.1	Introduction	43
3.1.1	Goals	45

3.2	Mobility Demand Study	45
3.2.1	Problem formulation	46
3.2.2	Data Source and Methodology	47
3.2.3	Mobility Analysis	49
3.2.4	Experiments	57
3.3	Mobility and Social Demands Study	61
3.3.1	Problem formulation	62
3.3.2	Data Source and Methodology	64
3.3.3	Mobility Analysis	67
3.3.4	Social Analysis	69
3.3.5	Experiments	74
3.4	Real-world Case Study: SmartHitch	83
3.4.1	FIWARE Technologies	84
3.4.2	Service Design	85
3.4.3	Revenue Model	86
3.4.4	Technical Implementation	86
3.5	State of the Art	95
3.6	Conclusions	100
4	SAMS: Crowd Steering	103
4.1	Introduction	103
4.1.1	Goals	104
4.1.2	Technology	106
4.1.3	Strategies	107
4.2	Indoor Real Case Study: FairSteering	110
4.2.1	Telecom Italia Project	111
4.2.2	Experiments	112
4.2.3	Demo Architecture	119
4.3	Outdoor Real Case Study: TogetThere	126
4.3.1	PETRA project - Venice Use Case	129
4.3.2	Incentive Mechanism	131
4.3.3	Experiments	132
4.3.4	Architecture	134
4.4	State of The Art	142
4.5	Conclusions	144
5	Conclusions	147
5.1	Summary	147
5.2	Open Challenges	148
5.3	Future Directions	149
	Acknowledgments	152
	References	155
	List of Publications	167

Chapter 1

Introduction

Future urban scenarios will be characterized by the close integration of ICT devices and humans. Citizens using their own capabilities and the ICT infrastructure could collaboratively constitute a large-scale socio-technical system to support collective urban awareness and activities. This will eventually contribute to define a dense ecosystem whose individual components (i.e. citizens, devices, software services, sensors and actuators) will enable collaboration and adaptiveness between ICT devices and humans, towards the realization of advanced urban services. Such services are going to impact on our way of living in the urban environments and to change their nature.

1.1 The ICT Enabled Smart City Scenario

The adoption of sensor networks, actuators and computational resources capable of interacting with users' devices is transforming urban environments [150].

People are increasingly equipped with smart phones with powerful capacities in terms of battery life, sensing technology, computational power and connectivity. At the same time, autonomous ICT infrastructures (sensor networks, security cameras, robots, etc.) are likely to pervade cities in the near future. Accordingly, the future urban environment is becoming a sort of very dense digital ecosystem.

The components that are going to pervade urban environments are characterized by heterogeneous and complementary *sensing, computing, and actuating* (SAC) capabilities, which can cooperate in a goal-directed way.

Future pervasive urban services, rather than being limited to sensing what happens in the city (as most current approaches do) will enable the complementary SAC capabilities of humans and ICT devices. Such services can contribute towards the Smart City vision along several dimensions from Intelligent Transportation Systems, to environmental sustainability and participatory governance [82, 139]. Such services will impact on our way of living in the urban environments. Even more,

they will change the very nature of our urban environments, turning them into socio-technical urban superorganisms [72, 150].

The urban system *as a whole* will be able to:

- combine a wide range of information sources (e.g., environmental data from sensor networks, mobility data and social network posts) and sense, either opportunistically or in a participatory way, the current state of the city;
- perform advanced reasoning on the data to extract patterns and routines that are taking place;
- engage in large-scale coordinated tasks to achieve specific goals (e.g., optimize traffic flow in the city, make it more environmentally sustainable, etc.).

That is, the overall urban environment will act as a single superorganism made up of individual organisms (humans and ICT devices) and capable of directing its behaviour towards the achievement of specific urban-level goals. However, engineering collaborative and coordinated services capable of harnessing human and ICT capabilities at large scales challenges current engineering practices and middleware architectures.

1.2 Urban Superorganisms

The foreseen widespread adoption of sensor networks, actuators and computational resources capable of interacting with users' devices will transform our urban environments [42]. Citizens will have the possibility of being continuously connected in a situation-aware and socially-aware way, both with each others and with the objects/entities around (e.g., typically via some situation-aware social networking infrastructure [150]).

In future ubiquitous computing scenarios, the very large number of interconnected entities that can be found in urban environments, whether humans or ICT devices, can potentially be exploited to create what has been defined as a superorganism [72]. In particular, closing the sensing, understanding, and acting capabilities of the interconnected agents in a loop, and making such activities collaborative ones, it is possible to realize coherent collective behaviours, as it is observed in many natural situations [17].

For example, in ant colonies, a single ant has very limited local sensing and acting capabilities, and little or no cognitive abilities. Yet, ants can indirectly coordinate their movements and activities, via spreading and sensing of pheromones in the environment, so as to exhibit, as a colony, very powerful capabilities of sensing (finding food in the environment), understanding (finding the shortest path from food back to nest), and acting (carrying large amounts of food in the nest). These capabilities make the whole colony seemingly intelligent and certainly adaptive in its foraging activities.

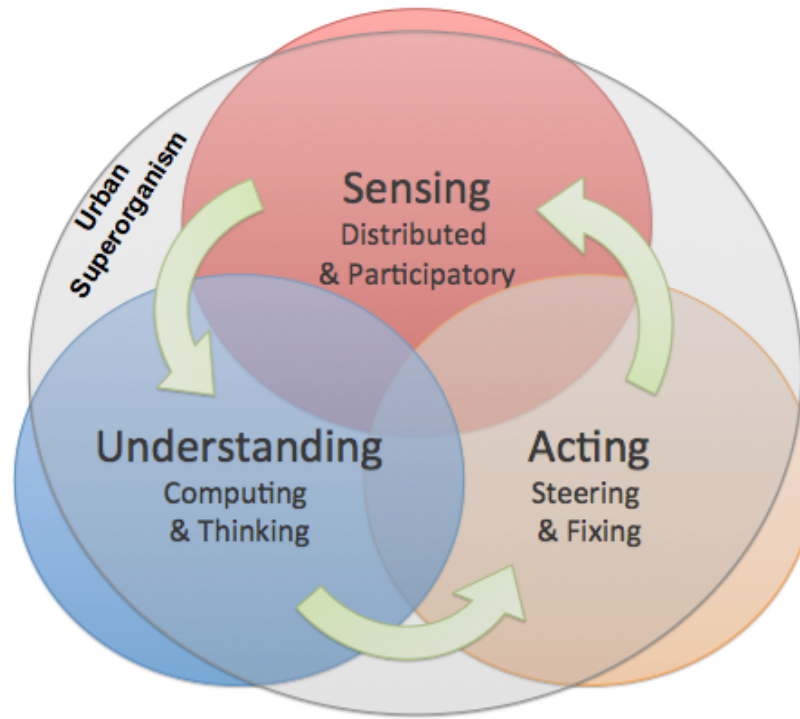


Fig. 1.1 The sensing-understanding-acting feedback loop of urban superorganisms.

In a similar way, future citizens can be engaged in large-scale coordinated activities allowing the city as a whole (i.e., the superorganism) to realize complex coordinated tasks like large-scale car sharing and energy saving campaigns.

The overall behavior of such superorganism will be that of a smart ecosystem, in which a variety of urban services will be enabled to improve the overall quality of life of individuals by providing them with services to better interact with the urban environment, and also by shaping the activities of the urban environment itself, to suit their own needs. In addition, by continuously monitoring and directing the behavior of its constituent agents, it will tackle city-scale problems.

After an initial learning phase in which raw data from sensors is collected, processed and classified, the superorganism will be skilled in context inference and anticipatory computing, as examples, and it will suggest tailored recommendations to the actuators.

Closing the capabilities in a loop means that the superorganism can measure the goodness and the adoption rate of the suggested recommendations, by making clear their causal relation with the effects they generate. The process results in the generation of collective awareness, which can describe both individual and large-

scale (i.e., global) characters, related respectively to single agents and to a collection of those [12].

Figure 1.1 illustrates the reinforcement feedback loop that creates collective awareness in the superorganism. Advanced, finalized and coordinated activities are the result of:

- *Sensing* activities, in which users supported by ICT devices and services get information about the current state of the environment (e.g., people location data);
- *Understanding* activities, in which advanced forms of context information are derived from the sensed data (e.g., citizens mobility patterns are identified from the collected location data);
- *Acting*, in goal-directed coordinated tasks supported by the extracted information (e.g., traffic management on the basis of the identified mobility patterns, car sharing on the basis of people mobility routines, etc.). The result of the activities being performed are then sensed again closing the feedback loop.

Further details on the capabilities of the urban superorganisms are given in Section 2.1.1.

1.2.1 Vision

Nowadays, despite many cities are soaking up the meaning of what a Smart City is, we are far away to observe effective infrastructural deployments that exploit a dense and distributed network of devices to pursue either urban level goals or individual ones. The lack of effective collaborative models impacts on the quality of the provided services as well, and a relevant gap is observable between the superorganisms depicted in literature and their deployment in real-world cases. The reason to this proven fact is related to the existence of a very challenging scenario that expands from technical and infrastructural requirements to social and collaborative ones. Furthermore, literature do not propose even a unified vision towards the engineering and the coordination of *smart services*. Section 1.2.2 describes most relevant challenges related to the enabling of urban superorganisms, while Section 1.2.3 defines the contributions of this work to the mission that would actually enable Smart Cities.

In literature, there are plenty of previous works that explore the SAC capabilities of ICT and humans engaged in urban superorganisms. However, these works lack of a general and unified vision, and do not completely deal with the complexity of the global scenario, i.e. they do not explore all the possible convergences of humans and ICT devices. Moreover, they do not fully make use of the very large number of interconnected individuals and their complementary capabilities to realize collective awareness.

Previous works in opportunistic and participatory sensing have tried to involve users by making use of their devices as sensors [81, 117]. On the opposite side, other

works try to detect events or situations by observing users activities on online social networks [122].

Some recent works try to address the "smart city" scenario [36]. Most of current approaches to smart cities (e.g., as those that are carried on by IBM [82]) are related to the "sensing and understanding" facets of the urban superorganism scenario, i.e., collecting (typically in a centralized way) data about various aspects of a city life and get a meaning out of it, for the sake of driving decision makers in planning future urban infrastructures. The acting aspect, i.e., the possibility of dynamically involve citizens and ICT actuators to dynamically influence the city is mostly disregarded. Only a few studies in this direction have been performed, and mostly oriented to steer crowd via mobile phones (e.g., the "tag my lagoon" project in Venice¹), or at directing traffic towards zones with available parking space (e.g., the Santander Smart City Project²).

Bio-inspired mechanisms are presented in the research field of self-adaptive and self-organizing systems in the form of frameworks and technique catalogues [55], with the intent to overcome the limit of ad-hoc implementations that prevent their systemically reuse. Thus, the basic idea is that of providing the bio-inspired self-organizing pattern modules with a set of reusable patterns that could be used to ease engineering of artificial and collective behaviors of urban superorganisms.

It is probable that the complementary capabilities of humans and ICT devices, and the ability to coordinate and organize them, could overcome current approaches and promote collective awareness and complex behaviors. In line with [55], bio-inspired algorithms could be helpful to achieve complex and coordinated behaviors. On the basis of such patterns, it will be possible, for the socio-technical system made of humans and devices, to put in place a variety of collaborative collective behaviors, expressing various forms of urban awareness and intelligence, and promising to dramatically change the way we move, live, and work, in our urban environments [10, 65, 74, 131, 147].

1.2.2 Challenges

Engineering collaborative and coordinated services capable of harnessing human and ICT capabilities at large scales challenges current engineering practices and middleware architectures.

This work identifies several challenges towards the construction of an infrastructure supporting urban superorganisms. A prerequisite in this challenging scenario is an high degree of awareness. As technologies improve, new types of sensors become available and affordable. All these challenges have to be solved to support, the superorganism scenario.

¹ <http://www.tagmylagoon.com>

² <http://www.smartsantander.eu>

Heterogeneity and interoperability

Whether they are humans equipped with a mobile phone or autonomous ICT devices, the software architecture has to be able to realize an abstraction layer in order to achieve the same goal on different individuals, by adapting and exploiting their heterogeneous and complementary SAC capabilities [64, 104]. The complementary use of sensing and understanding capabilities of humans and ICT devices has to lead the coordinated learning process towards reconstructing a collective awareness of the state of the city.

A common example that fits this situation is expressed by the capturing of pictures of a traffic jam both from users cameras and *Closed Circuit TeleVision* (CCTV) cameras, to make sure that all the images are properly tagged with information automatically generated by devices (e.g., amount of vehicles involved) and further enriched by humans (e.g., the reason they are stuck in traffic), due to their higher classification capabilities. The challenge is to realize an abstraction layer able to continuously observe the superorganism status and plan strategies to reach specific goals by making use of heterogeneous individuals with evolving sensing, actuating and computing capabilities.

Interconnection

As emerged in the previous section, individuals have to be connected and able to exchange messages for both (i) supporting collective behaviors (e.g. the coordination effort required to steer individuals to take photos of a road intersection), and (ii) gathering individual awareness to infer complex situations that involve the superorganism rather than specific individuals. It is worth noticing how this requirement calls for innovative data fusion and knowledge fusion [159] techniques. In fact, at both the individual and superorganism levels, multiple data streams of information sources have to be processed and put together to build up a coherent picture of the operating conditions.

Privacy Preservation

Consider a service supporting urban mobility. The data collected from users in movement is of particular interest but affected by privacy concerns. By analyzing this type of data, it is possible to infer users whereabouts and patterns of mobility that are very important for mapping the dynamics of the city [31, 97]. In fact, the pervasiveness of this kind of data, permits to analyze user behavior creating un-precedent opportunities for information retrieval almost in real-time [122].

A very important role is played by geo-referenced data coming from social networks, but also every other type of micro-data as preferences or opinions expressed, e-commerce transaction data etc. By performing different combinations of data analytics, it is possible to infer more information and use it to improve the results of computations, to increase user's engagement and the system awareness itself.

It is finally worth noticing that the superorganism infrastructure must support the privacy of data. Even if data is totally anonymized, the availability of different di-

mensions on data, cross-correlation techniques and uniqueness of mobility patterns, increases the success of re-identification attacks on mobility data [105]. In fact, the risk of matching a signal coming from a personal device and a name record in a database increases with the accumulation of data. Therefore the more can be done and kept within the personal handset, the greater the preservation of privacy [140].

Behavioral Steering

The effectiveness of the infrastructure supporting the urban superorganism is determined by the amount of people actually involved in the collective SAC phases. So, from a social perspective, it depends mainly on how deep individuals are steered by collaborative behaviors. The engagement of individuals should be stimulated by worthy mobility services that actually address their needs while provide the superorganism with a collective perspective on urban mobility.

Academic literature present several studies on modeling and evaluating behavioral changes driven by ICT devices [83, 87, 152]. As precisely described by Klein et al. [83], behavioral changes can be supported by taking a closer look at underlying determinants of behavioral change, focusing on how users can be persuaded to establish a desired behavior. This practice results in the identification of behavioral patterns composed by a sequence of behavior determinants, which are solicited through proper Human-Computer Interaction (HCI) techniques (e.g., provide the user with automated reminders, valuable suggestions, and tailored feedback on her activity), and then evaluated with a related computational model based on theoretical frameworks of behavioral change, to classify the degree of individuals motivation, awareness, and commitment to adopt new desired habits. However, a satisfying evaluation of behavioral change model needs to be protracted for years, in order to distinguish short-lived changes from permanent ones.

Dynamic re-configurability

As emerged by the heterogeneity challenge, the software architecture should show a certain degree of flexibility. Thus, the ability to dynamically execute heterogeneous code with heterogeneous sensing, actuating and computing capabilities is a key challenge to be tackled.

For instance, in the case of an accident reported by a citizen, the architecture has to support a service that requires the user interaction and thus has a user interface; while, in the case of the same task executed by a remote camera, the latter has to support services exploiting the SAC capabilities of devices. Furthermore, dynamic service composition and re-configuration is needed. The design of an architecture dealing with these challenges will make applications interoperable and able to self-reconfigure and self-optimize depending on the execution context.

Dynamic selection

One should evaluate which individuals are more suitable to be involved by taking in account different constraints. For instance, many CCTV cameras and humans could be available at the same time to capture traffic information to feed any Intelligent Transportation System.

Strategies taking into account different constraints (e.g. geographical areas, individuals status and sensing, actuating and computing capabilities) are needed to pick up the most affine individuals for the desired behavior. These strategies could be based both on explicit interactions (e.g., sending a message to an individual) or implicit interactions (e.g., sub-sampling the whole population of individuals satisfying specific constraints).

Context Awareness

A prerequisite in this challenging scenario is an high degree of awareness. In this direction, as technologies evolve, new types of sensors become available. Chemical, electric, optical, proximity and position sensors can provide data about environment, weather, presence or movements of and between different entities part of the city life. Furthermore humans can also act as a type of social sensor through social networks or their mobile phone signals [64, 104]. These data sources, made of humans and ICT devices, will produce, continuous streams that will generate a very big data set, specially, if the temporal dimension of data is considered.

From a computational concern, the issue is about finding the suitable kind of algorithms, and from this perspective, machine learning algorithms are useful for extracting high-level knowledge from sensed data [10]. These types of algorithms will support the awareness shaping. More specifically, one of the most significant issues pointed out by Zhang et al. [155] concerns the heterogeneous learning: comparison and fusion of data provided by multiple sensors is a challenging task that can be addressed to machine learning practices.

The high-level awareness acquired by the individuals through sensors should be used to infer complex situations and create a sort of collective awareness of the surrounding physical and social environment. The important challenge is to find ways of combining them together so that results coming from one data can validate and further describe results from other data. [11, 70, 97] propose some early approaches in this direction. More in detail, the software infrastructure here should be able to:

- Opportunistically sense and recognize many heterogeneous kind of situations.
- Self-reconfigure the awareness subsystem itself. The reconfiguration should be based on the partial awareness acquired by the infrastructure and is aimed to (i) improve the recognition accuracy by optimizing the classification modules and to (ii) preserve computational resources.
- Exploit the awareness of other individuals of the superorganism to infer complex situations and behavior in the urban environment that involve groups of individuals.

Mixing bottom-up and top-down design

Designing with a top-down approach means that all the requirements of a software architecture have to be taken into account a priori; systems engineered in this way have a predictable and measurable behavior but are not capable of coping with dynamic execution-context; while systems designed with a bottom-up approach are more robust and suitable for a pervasive environment but predicting their behavior and controlling them by design is not an easy task.

In the design of architectures for urban superorganisms, both of the two approaches are needed and finding and tuning the optimal trade-off between them is a key challenge to be tackled [42, 144]. A first exemplary case is the one of the coordination between individuals, affected by the dynamism of their activities and mobility patterns.

One can claim that the resulting superorganism should rely on self-organization in order to take advantage of collective behaviors, by taking inspiration from natural metaphors. However, emergent behaviors have to be controllable by design and thus understanding the trade-off of the two approaches and to what extent they can co-exist is a key challenge to be tackled to design controllable collective behaviors.

1.2.3 Mission

This work proposes innovative approaches and techniques to perform *Behavioral Steering* that enable collaborative and cooperative actions of the users involved. These contributions would tackle also the *Context Awareness* challenge that affects the deployment roadmap of urban superorganisms.

The reference scenario that is taken into account is represented by urban mobility. Innovative and smart mobility services will be proposed thanks to a deep study of mobility models that brought to the design of supporting collaborative tools.

The goals pursued by this Thesis are completely in line with the ones defined in the Emilia Romagna SPINNER2013 project MUCCA, which has mainly supported the work described in next Chapters.

1.3 The MUCCA project

MUCCA project aims to foster the active cooperation of citizens and their participation to social activities, through their smart phones, while they are moving within the urban area of a city. MUCCA is an Italian acronym ("Mobilità Urbana Cooperativa in una Città Amichevole") that stands for "Cooperative Urban Mobility in Smart Cities".

Pushed by the Smart City scenario, MUCCA wants to explore innovative way of interactions among citizens to enhance the quality of urban services and to enable new services influenced by the cooperative approach, and, of course, enabled by

the dense surrounding ICT environment. The born of new urban communities is pursued, with the main goal of having a source of information and resources always up to date and available to all the users in mobility.

1.3.1 Needs

The resulting scenario should address both individual and city-wide mobility needs, by enabling a set of services (i.e., mobility capabilities) that are provided to users in ad-hoc fashion, in terms of being tailored with respect to users location and contextual needs (e.g., current activity, specific preferences), and which can be even provided by the users themselves to other users that express a certain need.

In particular, recent technological advances of urban ICT infrastructures can offer good opportunities to establish innovative collaborative mobility models, such a real-time journey planner aware of the traffic on the road network, in which users can quickly and precisely report and share information of a congested area with each others.

1.3.2 Proposed Solutions

This work focuses on Socially Aware Mobility Services (SAMS) realized by finalizing the SAC capabilities of the superorganisms and of their agents at the improvement of urban mobility. That is, to increase the effectiveness of individual mobility while at the same time improving the overall sustainability of urban mobility.

In this field, human agents will play a fundamental role in the deployment of mobility services, since they can act both as consumers and as providers (e.g., via their private cars or simply by supplying information) of the services. Human social interactions can be pushed through a precise dynamic orchestration of the enabled data streams coming from both humans and ICT devices of the urban smart ecosystem (see Figure 2.2).

The contributions to the project are grounded on presenting how, in the scenario introduced above, the vision of the urban superorganism for the provisioning of integrated SAMS, can be effectively realized by a specifically suited coordination framework presented in detail in Section 2.4. Such coordination framework will be proposed as capable of supporting the iterative closed process of:

- Detecting mobility events related to the moving agents on the infrastructure, by harnessing the surrounding portion of the mobility data network shaped by the infrastructure itself, and also by processing the stream of incoming requests for mobility services;
- Identifying the possible solutions to satisfy expressed mobility needs based on the current state of things and of requests; anticipate future situations and future (or latent) mobility needs;

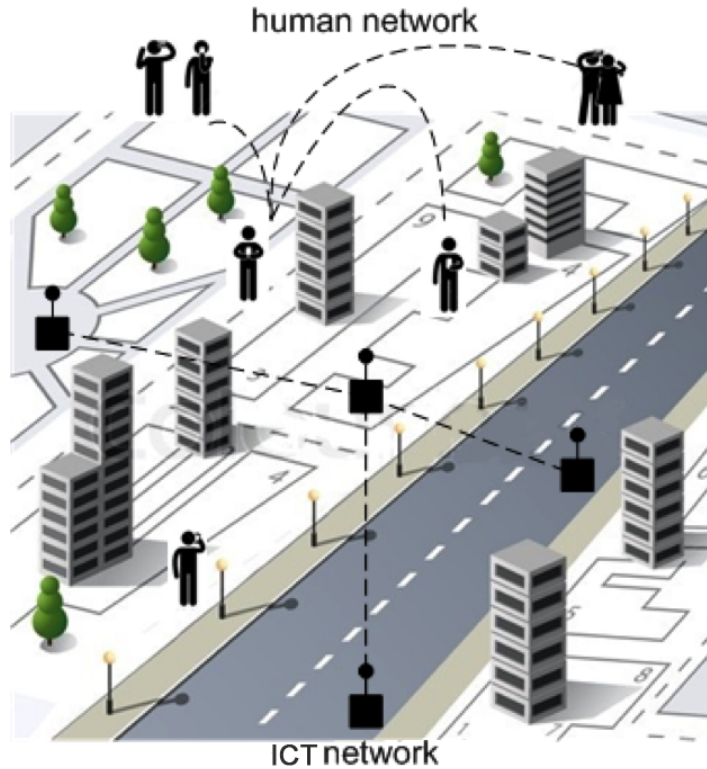


Fig. 1.2 Humans and ICT devices take part to the urban smart ecosystem

- Putting in act the necessary actions on hardware/software actuator agents, or persuade human agents to act in certain ways, so as to end up realizing a coherent and sustainable set of services to satisfy the recognized needs.

Furthermore, proposing exemplary case studies of SAMS and tentative designs and implementations of their software architectures, and community models, can offer useful inspirations for urban mobility authorities and can bring positive insights for the viable deployment of such mobility services. Therefore, a real-world service related to the sharing of private cars (Ride Sharing) in Chapter 3 and a couple of SAMS conceived to support pedestrian mobility (Crowd Steering) in Chapter 4 are presented.

1.3.3 Motivations

Collaborative behaviors oriented to the improvement of the life quality in the city and of its public services can contribute to the development of a new and relevant

citizenship feeling. Furthermore, cooperation and sharing in mobility can globally reduce air pollution, and so offer an improved quality life in urban environments. MUCCA encourages the born and the development of an urban superorganism able to infer citizens' situations, through their interactions with ICT city's infrastructure, and to provide recommendations oriented to satisfy individual and community needs.

Nowadays, everyone who approaches the SAMS scenarios, and aims to establish effective urban superorganisms, faces lacks in different directions. In particular:

- an efficient models for the spontaneous and distributed collaboration among users in movement is missing;
- an effective, coordinated, dense, and distributed ICT infrastructure is missing;
- privacy policies that manage data usage and limitations are scarce and not unified;
- crowdsourced mobility models are growing vertically, in a service-oriented way (e.g., Waze App³), and not horizontally, oriented to the city as a whole.

1.4 Contributions of this Work

This work brings its contributions through *(i)* the definition of a new model for the distributed coordination of users and ICT technologies, and *(ii)* the real testing of SAMS services that aim to enhance urban mobility, which embrace most of the challenging aspects related to this topic.

In particular:

- *Chapter 2* introduces the concept of Collective Awareness, the enabling technologies, and how users are involved in the process of awareness generation within the Urban Superorganism schema. Furthermore, a presentation of SAMS is made, by depicting some relevant examples, proposing a classification approach based on the level of awareness required to effectively work, defining a general-purpose coordination framework to orchestrate the human-ICT interactions, and showing an exemplar case study of a Ride Sharing SAMS to better explain the tasks of the coordination framework;
- *Chapter 3* introduces the exemplar SAMS of Ride Sharing (a.k.a. Car Pooling). A total of three studies are presented and evaluated. While in the first only the mobility aspects of such a service are evaluated, the second, which has been developed in collaboration with IBM Research Lab. Ireland, analyses also the social impacts of it, and offers an evaluation of the sharing opportunities enabled by taking into account how likely people would enjoy to share their rides. The last case study concern the actual deployment of the Autonomous Ride Sharing service *SmartHitch*, in the context of the EU FP7 Project FrontierCities;

³ <http://www.waze.com>

- *Chapter 4* depicts the SAMS Crowd Steering, which is conceived for pedestrian mobility. After an initial deep introduction on the enabling technologies and strategies, a couple of real-world Crowd Steering use cases are presented. The first use case has been developed in collaboration with Telecom Italia Joint Open Lab Turin, and it consists in an evaluation of Crowd Steering strategies in indoor environments, and in a actual design and implementation of a demonstrative architecture for supporting visiting users of an exhibition fair. The latter has been developed in collaboration with IBM Research Lab. Ireland, in the context of the EU FP7 Project PETRA, and it concerns an outdoor Crowd Steering scenario with the implementation of an Activity Planner for tourists in the city of Venice;
- *Chapter 5* concludes the Thesis with a brief summary of its contributions, open challenges and the directions of potential future works.

Chapter 2

Collective Awareness for Socially Aware Mobility Services

Penetration of smart phones and localization technologies are enabling a variety of smart location-based social mobility services that can notably improve the quality of life of citizens on the go. Most existing solutions are shaped from a common set of data sources processed through different computational flows to provide a specific class of services, and rarely strive for generality and urban-scale goals. A general-purpose coordination model is currently missing for supporting the dynamic composition of a variety of *Socially Aware Mobility Services* (SAMS) and for satisfying functional and infrastructural requirements to address both individual and urban-scale mobility needs.

2.1 Awareness Feedback Loop

Smart cities are characterized by the close integration of ICT devices and humans. However, the vast majority of current deployments of smart technologies principally relies on sensing devices collecting data and data mining techniques squeezing little meanings out of them. Nevertheless, citizens integrated with ICT technologies could collaboratively constitute large-scale socio-technical superorganisms supporting collective awareness and behaviors. Novel means of pervasive collaboration in the urban environment could be designed to exploit the complementary capabilities of individuals in an adaptive way. This chapter clarifies the vision on urban superorganisms and proposes a prototype architecture supporting their development.

Future collectives of socio-technical entities (i.e., citizens and ICT devices) will be most likely continuously, and possibly invisibly involved in distributed and participatory sensing campaigns. Their involvement can happen both individually or in a crowd. The results of such collective sensing activities will then promote a sort of participatory perception that can enable collective awareness (by computing, by thinking, and by communicating) of urban facts and issues, and urban dynamics. Individual actions and crowd-oriented recommendations can be suggested at the same

time, since the superorganisms take care of both *individual* and *global* collective awareness.

Based on such shared understanding, it will be possible to plan for specific collective actions aimed at fixing problems, addressing needs or adaptively steering urban dynamics.

2.1.1 Superorganism Capabilities

In this section, more details on the sensing, computing and actuating (SAC) activities, to be integrated for the sake of enabling the collective awareness feedback cycle (depicted previously in Figure 1.1), are provided.

People are increasingly equipped with smart phones with powerful capacities in terms of battery life, sensing, computational power and connectivity. At the same time, autonomous ICT infrastructures (sensor networks, security cameras, robots, etc.) are likely to pervade cities in the near future. Accordingly, the future urban environment is becoming a sort of very dense digital ecosystem.

	ICT	Human
Sensing	Sensor networks, camera networks, RFID tags, opportunistic access to smart phone sensors	5 human senses, facts-opinions-feelings posted on social networks, proactive usage of smart phone sensors
Understanding	Data analysis, data aggregation, basic situation recognition	Pattern analysis, advanced situation recognition, emotion recognition
Acting	Traffic lights, digital signage, pervasive public displays, actuating devices of critical infrastructures such as water distribution, energy grid, etc.,	Physical movements of individuals and of manned vehicles, physical actions, social persuasion

Fig. 2.1 The table summarises sensing, computing, and actuating capabilities of both humans and ICT devices that could mutually interact within an urban superorganism.

The components that are going to pervade urban environments are characterized by heterogeneous and complementary SAC capabilities, that can cooperate in a goal-directed way. In particular:

Sensing Capabilities

- *ICT Side.* The capabilities in sensing from the ICT side are provided by (i) mobile phones equipped with GPSs, accelerometers and cameras; (ii) sensors networks and smart objects that follow the Internet of Things paradigm; (iii) tags that exploit the near field communication technologies (NFC, RFID and Bluetooth).
- *Human Side.* From the human side, the five senses of humans, which in many situations can supply and be more accurate than ICT sensors, can be put at work for the community, due to the possibility of continuous access to social networks.

In addition, users can make available via social networks any other information, thus acting as sorts of social sensors [121].

Computing Capabilities

- *ICT Side.* The capabilities in computing from the ICT side makes it possible to collect and digest very large amounts of urban data in a short time, and to perform some limited pattern analysis on such data.
- *Human Side.* From the human side, on the other hand, one can exploit the capability of recognizing complex situations and patterns (so called human computation [148]), which machines can hardly tackle.

Actuating Capabilities

- *ICT Side.* The capabilities in actuating from ICT side can be provided by (i) traffic controllers supporting pervasive solutions in the mobility dimension; (ii) public displays that will be exploited to promote adaptable citizens behavioural steering; (iii) all kinds of actuators related to critical infrastructures (e.g., energy grid).
- *Human Side.* From the human side the key actuating element involved is the user himself, which can perform a variety of actions related to moving or moving items around or changing the properties of some physical entities. In other words, citizens could accomplish actions, by realizing an impact on the environment.

The goal-directed integration of the above capabilities and activities, will allow to close the collective-awareness feedback loop enabling large-scale coordinated behavior among humans and ICT devices and services.

2.1.2 Impacts on Urban Life

In the next future innovative collaborative and collective behaviors, expressing various forms of urban awareness and intelligence, will take place. These will dramatically change the way we move, live, and work, in our urban environments. Here an exemplary list (not exhaustive) of potential aspects of the urban life that will be impacted by the deployment of any application for the superorganism is presented.

Mobility

Among many capabilities that future urban superorganisms will exhibit, the first that one can expect to be in place, and for which embryonic examples have been observed around already, will relate to urban mobility. Specifically, it will relate to the capability of affecting the movement of vehicles or pedestrians (i.e., steering), and thus improving overall efficiency of urban mobility while reducing the stress of people in movement.

A variety of sensors already exist to detect the conditions of traffic or crowd in urban environments. In addition, users are increasingly provided with the possibility

to contribute to such sensing activities by posting information on social networks or by opening access to their navigators and smart phone sensors. All this information can be used to understand how to improve traffic flow or how to avoid overcrowding. To this end: actuators such as traffic lights and digital traffic signs can be put at work for vehicles; public (wall mounted) and private (smart phone) displays can be exploited to suggest directions to pedestrian. However, one could push the capabilities of superorganisms much beyond. For instance, one can think at dynamically matching the similarity of planned vehicle routes and of goods deliveries to dynamically self-organize a very flexible ride sharing and shipment services.

In general, urban superorganisms induce a change in the dominant paradigm for the provisioning of mobility services: from sensing mobility patterns and adapt existing services to them, to dynamically collect mobility needs and self-organize the role and mobility patterns of vehicles accordingly.

Sustainability

As an additional example of how the capabilities of future urban superorganisms can impact urban life, just imagine sensing in real-time information related to energy consumption, to compute sorts of instantaneous urban carbon footprints for specific areas of the city or for specific groups of citizens, other than for the city as a whole.

Public displays can then be exploited to share this information along with some summaries of the factors contributing to it, and personal displays can be possibly exploited to let individual and groups to become aware of their own contributions to the urban carbon footprint. On these bases, one could think at steering the behavior of individual citizens towards more energy efficient behaviors, or at engaging groups of citizens in self-organized collaborative actions aimed at solving/improving specific energy problems in specific urban areas to supply the lack of actuators suitable to the purpose (e.g., detecting open windows and closing them).

Taking Care

Via similar means, it could be possible to dynamically involve citizens in proactively helping to take care of the city (e.g., to help keeping it clean or making it a safe place for everyone). For instance, one can think at dynamically engage people to temporarily take care of (or simply take a look at, for some minutes) children on their way to school, whenever the current activity and known habits of some persons suggests.

Ideally, in the presence of enough matching persons willing to be involved, and possibly complementing sensors (e.g., cameras) and actuators (e.g., robots) already in place for that purpose, one can make sure that the whole path from home to school of every children in a city is properly covered and taken care of.

Feeling Part of It

Beside thinking at measurably useful objectives and services for which urban superorganisms can be put at work, their advantages could also be in the (not easily measurable) way by which they will improve our experience of living in urban environments. In particular, acting and moving around in a city by being given feedbacks on the effect of our own existence in it, and by making possible to observe ourselves in relation with our environment, and with the other citizens, can make most of our everyday actions inherently more pleasant and rewarding, and can promote a renewed and stronger sense of citizenship. New urban communities can arise.

2.1.3 Individual and Global Support

The dramatic progress in embedded and mobile computing technologies, smart phones in primes, along with the pervasive diffusion of social networking tools, let's envision the emergence of a dense networked ICT infrastructure. In such infrastructure, coordinated human agents (i.e., the citizens) and software/hardware agents will interact with each others so as to serve – at the same time – individual-level and urban-level goals, as if they were part of a single socio-technical urban superorganism [150].

This means, the enabled collective awareness aims to reach goals at different scales. By following the bio-inspired approaches offered, for instance, by ant and bee colonies, entities need to coordinate their individual and precise activities with the ones performed by the others in the colony, and the colony itself needs the contributions offered by any individual entity to survive.

Entities, taken individually, have very limited capabilities that turn into very powerful ones only if they collaborate each others for the survival of the colony. At the same way, superorganisms can support both individual and global needs of the socio-technical entities within the urban smart ecosystem.

2.1.4 Human - ICT Interactions

As introduced in previous sections, smart cities are characterized by the close integration of ICT devices and humans. However, the vast majority of current deployments of smart technologies principally relies on sensing devices collecting data and data mining techniques squeezing little meanings out of them. Nevertheless, citizens integrated with ICT technologies could collaboratively constitute large-scale socio-technical superorganisms supporting collective awareness and behaviors.

This section:

- clarifies the vision on how humans and ICT technologies can collaborate together, in a coordinated manner, to support urban superorganisms, by proposing a prototype architecture;
- presents an exemplary set of SAMS that can be established thanks to the deployment of coordinated and distributed socio-technical infrastructures;
- identifies the key challenges towards the deployment of such infrastructures;
- analyses a smart social mobility case study to explore the tasks of the proposed coordination framework for human - ICT interactions.

In order to ground the discussion and exemplify the above concepts, let's now introduce a case study scenario related to the governance dimension in smart cities. In particular, the focus is on a city-wide *Intelligent Transportation System* (ITS): an infrastructure aimed at relieving congestions, improving safety and enhancing productivity [145]. The achievement of these three main goals can be realized in terms of the superorganism's perspective.

Following the superorganism's paradigm, citizens can cooperate with each other to steer the behavior of the city (in terms of mobility dynamics), by interacting with its transport infrastructure and by sensing the environment.

Collective Sensing

Both humans and ICT devices concur in sensing activities for the ITS. In addition, humans can exploit their personal mobile devices to share any traffic-related event they are dealing with, or any mobility activity they are doing, and ICT devices can, for example, autonomously monitor in real-time the amount of vehicles at main intersections. Sensed data plays a key role in the effectiveness and efficiency of the ITS to provide real time data feed on what is happening in the city.

Collective Awareness

Real-time data streams provided by people and sensors deployed in the city can be processed by some specification-based and/or learning-based techniques [147] owned by a context inference platform (such as the two-tier middleware architecture proposed in next Section), in order to compute the actual situations faced by people on the move and ICT devices installed on the transport infrastructure. Thus, for example, the situation that occurs during traffic congestions can be inferred by gathering together the road characteristics context, the vehicle velocity and model contexts, and the weather context [74]. On the basis of these assumptions, the ITS could be endowed with situation-aware modules that aim to (i) analyze specific recurrences and (ii) infer the traffic network flow forecasting [65].

Collective Actions

The ITS manager could model current traffic conditions dynamically and take steering decisions accordingly, in order to optimize traffic flow by enhancing efficiency

(e.g., by recommending re-routings, intermodal routes and viability changes) and safety (e.g., by inferring accident-like situations before they occur, and acting to avoid them). Therefore, the ITS manager can suggest corrective actions directly to the citizens, to the infrastructure and to the urban planning authority of the city, accordingly to the fulfillment of the ITS intents introduced above.

2.2 Socially Aware Mobility Services (SAMS)

SAMS will soon hit the streets of our cities. However, most of existing solutions so far are not integrated with each other, and the understanding of how to design a general-purpose framework, supporting a variety of integrated services and promoting direct users involvement, is still missing. Thus, a conceptual vision of SAMS is depicted, by focusing on the cooperation and interoperability of the actors involved.

Then, the infrastructural requirements to enable such SAMS are analyzed, and the characteristics of a general-purpose framework for the provisioning of them, conceived as a distributed and open agent coordination infrastructure, is presented.

Trying to improve urban mobility is and has always been a challenge for urban designers and city administrators. The recent dramatic progresses in ICT technologies, in particular, have led to the emergence of a very broad area of research in ITS.

ITS, in general, represent the most advanced way to establish a real-time transportation management, and consists in harnessing ICT technologies to better address users mobility needs and to support urban authorities' decisions [5, 145].

ITS aim to improve urban transport performance, and can address in turns the problems and issues of pedestrians, cyclists, private vehicles, public transports, and roadside infrastructures. However, the application of ITS is often limited to the provisioning of on-demand web-services, with little or no interactions between users and contributions from user themselves. Furthermore, ITS do not offer a unified and integrated approach to support urban mobility in all its aspect, and often they own independent approaches for different mobility needs.

ITS drive the establishment of infrastructures for real-time transportation management. Advanced sensing (such as traffic metering) and actuating (such as adaptive traffic lights) infrastructures, can be put at work to better address – possibly in real time – users' mobility needs, and to support urban authorities' decisions (a comprehensive overview on ITS is available elsewhere [5]). Such infrastructures, along with the progresses on vehicular technologies, from vehicle to vehicle interaction to self-driving cars, are contributing to dramatically improve urban mobility.

However, current ITS approaches do not exploit the best possibilities for engaging smartphones, users interactions, and social networking tools, which could instead contribute to a more integrated, powerful, and flexible approach to support urban mobility at different levels and for different needs.

In general, the shift from ITS to SAMS must pursue the desired comfort for citizens and the satisfaction for urban authorities at the same level, by improving traffic

efficiency and road capacity on the transportation network at an integrated, global, level. Moreover, the services focus to impact on the development of increased social participation of citizens, where they are no longer simply requestors of mobility services, but can in turn play a role in the provisioning of services. Such an endeavor can feed collaboration, cooperation and sharing practices with incentives and regulations.

SAMS consist of all the mobility solutions enabled by pulling data from the available set of agents, generating higher information out of them, and enabling potential social interactions between them. The utility information is returned to agents in such a way as to reinforce their interaction.

The shift from ITS to what SAMS consists in:

- Exploiting in an integrated and cooperative way the mentioned infrastructural ICT agents (sensors and actuators) the sensing of individual needs (either explicitly specified by users or implicitly captured by smart phone and social sensing), user contributions and interactions (again, by using social networking tools);
- So as to orchestrate the provisioning of a variety of user-centered mobility services, capable of better satisfying user needs and increase social participation.

The basic idea of SAMS is to dynamically sense the specific mobility needs of users and dynamically check them against not only the set of actual mobility services (e.g., public buses) but also against the potential mobility capabilities (e.g., a private vehicle going to a specific place) that could be turned into mobility services (e.g., by having the private vehicle offering a ride).

As shown in Figure 2.2, the overall envisioned framework for SAMS can be described as realizing an iterative self-regulating and self-organized collective process (see also [150]), shaped as a close feedback loop. There, humans and ICT devices collaborate towards the achievement of multi-scale mobility goals, both at an individual level and an urban level. More in particular, such process consists in:

- Detecting mobility needs and mobility capabilities through the sensing of raw mobility data coming from the sensors deployed over the environment, as well as from the evaluation of the incoming mobility needs expressed by the users on the move.
- Identifying a set of matching solutions, that is, understanding how mobility capabilities can be put at work to satisfy at the best mobility needs, possibly also by trying to anticipating future events or by identifying which missing capability should be called into action via some means.
- Coordinating the activities of the agents involved (that is, controlling the behavior of ICT agents and recommending human agents to act in certain desired ways), so as to enable an effective set of mobility services to be realized, and to eventually satisfy the recognized needs.

The considered category of SAMS consists of all the mobility solutions enabled by pulling data from the available set of agents, generating higher information out of them, and enabling potential social interactions between a pair of agents. The utility information is returned to them in such a way as to reinforce their interaction.

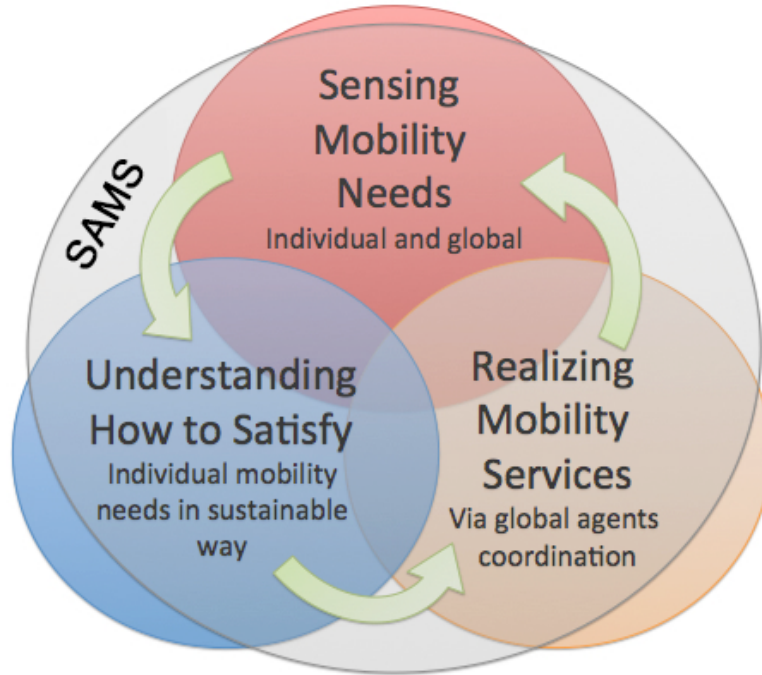


Fig. 2.2 Socially Aware Mobility Services (SAMS) in urban superorganisms.

Citizens with mobility needs receive collaborative recommendations built on the matches with the services provided by other citizens, thanks to the supporting ICT infrastructure. Such recommendations can be strengthened if users have a similar profile, especially in terms of collaborative behavior. Data from online social networks can detect social communities with same interests and mobility habits [21, 22, 153].

The system will monitor the eventual adoption of the recommendation, and its effectiveness (was the service actually available?). Finally, it will update the profiles of the involved agents, to provide more useful recommendations in the future.

2.2.1 Exemplar SAMS

In this Section some examples of such SAMS are presented. Furthermore, in Section 2.4 some approaches presented in literature and related to these examples will be analyzed.

Parking Match

Assume that a driver will look for a parking space close to some area of a town (as detected by its car navigator). Then, another driver leaves a parking spot in the same area. Thanks to the (opportunistic) incoming data collected from their smartphones, from the signals coming from their vehicles, or from the parking sensors installed in the parking lot [35, 101, 109], a match between the parking need and the parking capability takes place, and the driver that is approaching her destination is prompted with a recommendation.

Itinerary Match and Ride Sharing

When the spatio-temporal analysis on the mobility data, coming from a set of users, reveals that their co-location happens regularly over time (as seen in [36, 63, 69]), a possible pool of commuters is identified. Thus, they could be steered to switch to ride sharing and persuaded with the benefits they can obtain by switching to a collective mean of transportation. Some small modification of the usual itineraries of some vehicles could be also suggested to better match the route of these vehicles with the possibly slightly different required route of the users.

Available ride sharing services show how struggling is to reach a critical mass, hence social incentives are crucial (an exhaustive analysis of the related issues is presented here [66]).

Taxi Sharing

Taxi service can be shared among people that actually are looking for a ride towards close destinations. So it becomes more accessible for the users, in terms of lower cost per passenger, and the revenue for the taxi driver is increased, in terms of higher amount of rides achieved (to evaluate the market potential of such a service see [99]). This service differs from the Itinerary Match in terms of how the matches take place. Itinerary Match mainly evaluates historical usual trips that happen routinely, while the latter considers real-time data.

Multimodal Rides

A person explicitly declares a destination from her starting location, asking for directions. The superorganism selects the spatio-temporal portion of data streams that can provide multimodal directions to reach that destination. Current traffic level and rides availability of multiple means of transports on the transportation network is evaluated and several complex pattern matching mechanisms are put in place. The best multimodal alternatives to reach the desired destination are prompted to the user. Several approaches come from Operational Research [16, 50, 61]. In [40], authors have considered ride sharing as a complementary solution to usual means of transports in multimodal trip planning.

Chaperone Services

When parents, relatives or friends cannot look after a child that needs to be shepherded to school, one can share the path the child is going to follow, at a certain time, to look for someone that can take charge of assessing her presence at any intermediate checkpoint (e.g., a bus stop, a crossing, a grocery). Proximity sensors and reliable citizens can probe the child is actually there and send real-time feedback to her parents.

The above examples in any case see a clear distinction between provider of a service and requestor of a service, and consider that providers of a service are not influenced by the request. However, in a really integrated system, the mean to provide a service can be dynamically shaped upon the request, in a process of mutual influence. Indeed, those who provide a service is because they have a need to satisfy.

It is thus possible to let the distinction between requestor and provider vanish, and dynamically adapt the shape of services depending on the need, also with some supra level objectives in mind behind the opportunistic self-interest of the involved parties.

2.2.2 A Classification Approach

A tentative classification of SAMS is proposed here. On the basis of the strength of the interactions happening on the feedback loop, one can think to classify SAMS according to the level of collective awareness enabled on them.

Of course, the main effects on the awareness generation feedback loop are caused by the human and ICT actors involved in the superorganism. In particular, the *complementarity* and the *automation* of its sensing and actuating capabilities represent the dimensions with a direct influence on the effectiveness of the SAMS put in place. Moreover, also the *coordination* and the *goal* of the computing capabilities impacts on the level of collective awareness that could be generated.

Complementarity

The set of humans and ICT devices, that are engaged into the provisioning of a SAMS with sensing and actuating roles, is supposed to offer a wide set of capabilities that complement each others. The more the set is heterogeneous, the higher the amount of its capabilities.

The level of complementarity, of the socio-technical entities, and thus their *heterogeneity* or *homogeneity*, has directly proportional effects on the level of collective awareness that could be generated. This is true both in knowledge mining actions, in which capturing data from various sources of information can lead to a better

representation of knowledge, and in service provisioning, where the orchestration of the activities impacts on the quality of the service itself.

To support this idea, for instance, one can think to a board meeting of a company: the board is usually composed by a representative of any company department; since during its meetings the next strategic actions to take are discussed, the board needs the diversity on perception put in place by its heterogenous members to better know what actions should be put in place; then, corrective actions will be taken by the employees of the company departments, with different tasks performed thanks to their different competences and capabilities. The same behavior, can be reached also without a central coordination unit, where decisions are taken locally, individually, but actions are performed by a set of individuals. A fitting example is offered by colonies of insects.

However, according to the definition of superorganism, the heterogeneity of the entities involved in the loop should consists of humans and at least one kind of ICT device. People and ICT entities should be present at least in one phase of the loop to get a proper socio-technical system.

Automation

Another dimension that concurs to the collective awareness qualification is represented by the level of automation with whom sensing and actuating phases of the feedback loop are performed.

In details, sensing activities can happen *opportunistically* or in a *participatory* way. The first involves an high level of automation in which the entities are unobtrusively solicited, when it is opportune for the sake of the service, in capturing data and sharing it with the computational unit. The latter, concerns more flexible and autonomous entities that performs sensing tasks whenever they believe they should participate on it.

Actuating activities can be defined as *actions* and *recommendations*. The first are mainly directed to ICT devices, which can be fully controlled and driven from a computational unit, thus they could not refuse to act, except if they are not working properly. The latter are mostly sent to humans, which have free will to act as they please.

Humans can be engaged either with opportunistic sensing and actions (automated through their smart phones) or with participatory sensing and recommendations. ICT entities, in turn, are mostly engaged automatically by other entities, but sometimes they can be endowed with individual logical capabilities that could trigger sensing and actuating tasks.

Since in any SAMS can emerge the necessity of performing some corrective and sudden actions, enabling automated sensing and actuating activities could strengthen the robustness and the stability of the system, thus creating trust and reputation for the service itself. This lead to an increased level of collective awareness.

Coordination and Goal

As introduced earlier in this Chapter, computational entities on the SAC feedback loop have the main role of understanding the data collected from sensors, and proposing actions and recommendations to the actuators on the infrastructure. The goal pursued by computational units is intrinsically defined by the SAMS that is put at work. In turn, the established SAMS is deployed to support mobility needs and to address related issues.

Both needs and issues can be seen from a *urban – global – perspective* and a *user – individual – perspective*, and they can be addressed either by local interventions, or by global ones, or by a coordinated combination of them. Basically, the more distributed and inter-dependent are the computational operations, the higher is the level of collective awareness. This can be seen as a complex network of interactions between entities that through adequate *coordination* aims to give a solution to any kind of emerged need, thus enabling *individual collective awareness* and *global collective awareness*. The first can be defined as the solution discovered by a sample of entities to an individual need, the latter as the solution discovered to solve city-wide problems. An high coordination level among computational units let to enable cross-domain operations through *knowledge fusion* techniques [159], while their poor coordination lead to more traditional *data fusion* mechanisms. These considerations lead to define a second dimension able to describe computational operations: the *goal*. In particular, the diversity of goals that they pursue directly impacts on the collective awareness generation. Goals on a *single scale* (individual or global) contribute less in generating collective awareness with relation to the ones on a *multiple scale* (individual and global).

One can think to a journey planning SAMS that exploits crowd sourced information to suggest the quickest route to an individual. While, a traffic lights support system, which lets congestions disappear by acting on the traffic lights and digital signals control, consists in a tool to globally distribute traffic across the city. If the two systems could take advantage from the coordination of both central computational units and local ones, they should exploit both individual understanding capabilities (i.e., enabled by the brain on humans, and by logic circuits on ICT devices and smart phones) and global ones (i.e., cloud data-centers that run analytics on collected data and send either unicast or multicast actions and recommendations), as well they should pursue individual and global goals at the same time.

Figure 2.3 suggests a classification for every phase of the SAC feedback loop, with regards to the dimensions introduced above. The impacts on the collective awareness generation of any phase increase accordingly to the heading of the grey arrows.

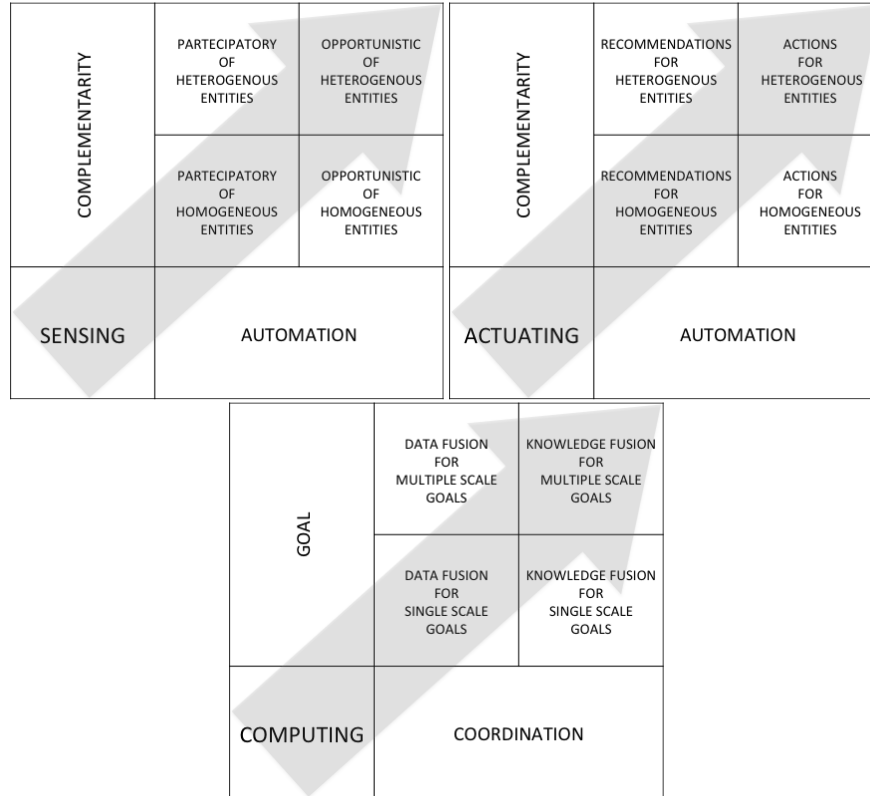


Fig. 2.3 A proposed classification for SAMS, according to the level of collective awareness enabled by each one of the SAC phases: **(top-left)** sensing involves participatory and opportunistic mechanisms on heterogeneous and homogeneous sensing devices; **(top-right)** actuating consists in recommendations and actions for heterogeneous and homogeneous actors; **(bottom)** computing concerns data fusion and knowledge fusion techniques to pursue single or multiple scale goals. Grey arrows heading point to their increasing contribution on collective awareness generation.

2.3 A General-Purpose Coordination Framework

A general-purpose coordination framework that supports the shaping and the provisioning of SAMS will represent a powerful tool for urban designers and city administrators to make urban mobility services more efficient in terms of cost of the infrastructure (by harnessing the same sensors and actuators that self-reconfigure themselves upon specific requests, as well the same software architecture) and amount of data collected and processed (by sharing them among services with different purposes, instead of replicating similar operations for each service).

Furthermore the provisioning of integrated solutions for different mobility needs can increase their individual adoption rate, towards the aim of reaching a critical mass of users, and thus can increase their effectiveness.

In the framework, citizens are intended to collaborate in crowdsourcing initiatives driven by combining mobility requests with fitting SAMS. Citizens are strongly centered in the framework, and their social interactions can shape *Social Collective Intelligence* (SCI) [119]. The goal of SCI is to harness real, physical, social interactions to address problems that require broad collaborations among people of the socio-technical superorganism.

Several distributed data streams on the transportation network are dynamically enabled, collected, processed and evaluated. Such data are generated by mobility agents, which can be ICT devices installed on the roadside, other humans that share their travel updates, or virtual social networks.

2.3.1 The Architecture

Beside the availability of a general underlying ICT infrastructure, the existence of a suitable coordination framework is needed to effectively enable the envisioned SAMS, and so to match with each other mobility needs and mobility capabilities, and to orchestrate the activities of the agents involved. Accordingly, a general-purpose coordination infrastructure for the provisioning of SAMS is needed.

The envisioned coordination infrastructure can be deployed over on some centralized server infrastructure, where to collect all data and host all needed computations. However, it could also effectively support a distributed implementation, in which multiple servers are devoted to collect data in their spatial proximity and analyze, typically in coordination with each other, mobility needs. The infrastructure should essentially ground (as from Figure 2.4) on:

- A Shared Information Space – to be accessed by all involved ICT and human agents – in which to collect information about mobility capabilities that could be possibly turned into mobility services and about mobility requests;
- A Matching Engine to process data collected the shared information space. The activities of the matching engine are triggered by incoming mobility requests, and are continuously re-shaped due to the feedback loop between actuating and sensing activities.

As shown in Figure 2.4, the framework grounds on a matching engine that processes several data streams from a dense distributed tuple space, which is made of information concerning mobility status, requests, and services, generated by the agents on the mobility network.

The rationale of the matching engine is triggered by incoming mobility requests, which in turns drive a continuous flow of consequent activities. After several computing iterations on the available relevant information, the matching engine discovers and builds services on the mobility network, which finally result in mobility recommendations for the requesting user.

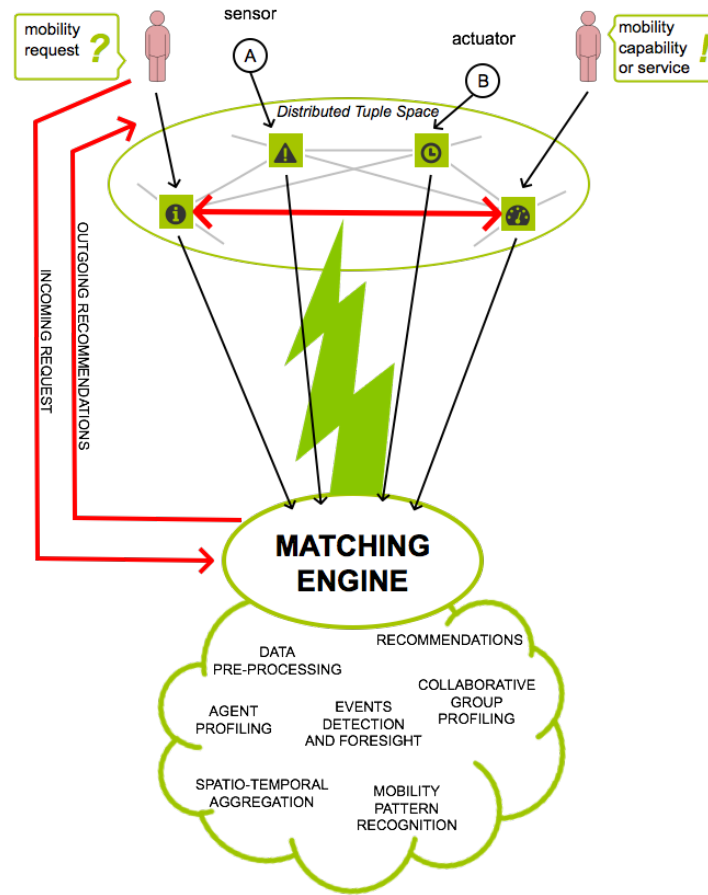


Fig. 2.4 Conceptualization of the coordination framework that matches mobility services with mobility requests. Agents on the mobility network include humans and ICT sensors and actuators such as smart traffic lights (which count the approaching vehicles) and smart signals (which change the displayed information). Data streams are dynamically selected and processed through the activities of the matching engine.

2.3.1.1 Distributed Tuple Space

The heterogeneous and unstructured nature of the information on the shared information space – and its inherent distributed nature – naturally suggests its realization in the form of a set of distributed tuple spaces (i.e, a non-organized multiset of information tuples).

Operations on the distributed tuple space can be managed through a middleware infrastructure that store and share knowledge. For instance, a coordination infrastructure such as the SAPERE one [30, 149, 151] can fit the needs faced during the implementation of the framework, since it integrates advanced and semantic pattern matching mechanisms, it defines a context-aware and spatial tuple space model, which is populated with specific rules to dynamically select, evaluate, and propagate information, and it can associate specific agents to react to events occurring within the tuple space.

Agents on the mobility network can implicitly or explicitly generate contextual information related to their mobility status, requests, and services. A middleware infrastructure based on a set of networked tuple spaces [137] could represent a viable and suitable solution to store and share knowledge among all the agents interested in some particular generated contents, as well to properly feed the matching engine with the necessary information.

2.3.1.2 Matching Engine

The matching engine can be described as an iterative process of sub-activities on the data that is living on the tuple spaces. The engine is conceived to exploit data from heterogeneous entities at the same time, and be able to recognize mobility needs and capabilities, as published on the tuple spaces, and build the necessary mobility services accordingly. Let's give a brief description of such phases.

Data Pre-processing and Spatio-temporal Aggregation

At a first place, incoming data is filtered, cleaned and aggregated. The process of course needs a considerable amount of data, collected over time, until the skills of the superorganism can build reliable recommendations to users. At further iterations, each incoming raw data will be filtered, cleaned and aggregated again, according to spatio-temporal constraints of the incoming request. Data modeling supports upper-level meaning abstractions, by generating complex data structures useful to understand a special mobility pattern of the considered agents.

These two initial steps represent the sensing tasks of the engine, and an initial computing phase is also put in place. The incoming data is filtered, cleaned, and aggregated, in order to obtain an upper-level meaning abstraction of the data, which is needed to infer mobility requests. Finally, data is stored in adequate data structures useful to make the agents mobility pattern easier to identify.

Mobility Pattern Recognition and Events Detection and Foresight

Machine learning techniques enable regular patterns identification and anomalies detection on the aggregated input data. Really well trained classifiers can perform effective anticipatory computing [111] that can be crucial in dynamic environments.

Not only agents on the move own mobility patterns (inferred, as example, by mining their mobility routes from GPS data). Roadside sensors can shape the mobility status on the mobility network as well, and so they let creation of tuples that characterize the mobility context of a geo-fenced area in a specific time interval. So, it is clear that it will be possible to detect and anticipate the occurrence of significant mobility events and have a real-time distributed representation of them.

Agent and Collaborative Group Profiling

Each entity on the network is characterized by its own capabilities, which let it play specific activities with proper tasks, which are, in turns, driven by the nature of the entity itself. These conjectures bring the necessity to model agents behavior to the foreground (the Belief-Desire-Intention (BDI) model [118] is one of the approaches suggested by the literature in this field). The distributed tuple space should be populated with profiling contents related both to individual agents and to groups of them.

Interactions among group of agents is actually a crucial aspect to model. A survey with some proposals is presented in [25]. Humans interactions offers a good starting point in collaborative behavior understanding. The discovering of interaction reasons, modes, and effectiveness is pursued, in order to bring collaboration aspects to the shared knowledge. In order to motivate users in deeper collaborations, behavioral changes can be stimulated through tailored incentives and mechanisms taken from persuasion theory [57].

Recommendations and Feedback Impacts

Once the platform is able to infer the up-to-date context of the active agents, the process goes on, to the evaluation of which mobility events can be useful in addressing mobility requests. The eventual set of identified alternatives is then sent to the requesting user, as recommendations, in the form of available services. The last two steps involved in the matching process shape the actuating phase.

The discovering of interaction reasons, modes, and effectiveness should describe the collaborative aspects of the considered agents. Their collaboration character should be brought to the shared knowledge, and tailored recommendations can be put in place to suggest a fitting mobility service by harnessing their capabilities.

The closing loop determines the continuous learning of the system, which becomes aware of how effective have been the recommendations, and how much relevant are the benefits generated thanks to them. The computing capability of the superorganism will continuously improve, and so its matching feature will do.

A similar mining can be performed when a reconfiguration of the ICT components on the mobility network is needed. Consider, as example, the increase of the sampling rate for a traffic sensor, according to the increase of variation in the traffic level measured. In that case, the superorganism needs to increase the granularity of the data collected. Since it has the direct control on the entities (hardware sensors and actuators), it is able to send the optimal self-reconfiguration rules to them.

2.3.2 *Open Architectural Challenges*

To effectively shape SAMS and make them viable, a set of challenging infrastructural and functional requirements should be met. These requirements can determine the technical viability of SAMS deployment.

Infrastructural Requirements

Interconnection. Based on the the Internet of Things paradigm [6], the distributed agents that populate the urban environment need to be connected and able to exchange messages each others. The distributed network of humans and mobile ICT-devices will enable the superorganism capabilities only if the information can flow seamlessly among a defined set of entities, despite network dynamics and made ephemeral.

Heterogeneity. The inter-connected components of the ICT infrastructure are highly heterogeneous. This feature has not to be considered its weakness. Their complementary role in knowledge mining has to be taken as an advantage. As example, one can consider a fixed entity on the roadside acting as a traffic sensor (e.g.: smart traffic light, CCTV camera). The data collected can be enriched with the one provided by mobile agents (e.g.: pedestrian, cars, buses), and hence its interpretation is made easier. Events detection and anticipation accuracy can improve as well. However, nodes heterogeneity challenges their interoperability.

Interoperability. Interoperable agents encourage combination of concurrent data streams from different locations, enabled in precise spatio-temporal patterns. The coordination framework is based on the orchestration of such different data sources, dynamically selected due their complementary role, according to the incoming requests. Nevertheless, energy saving and classification accuracy should imply specific conditions that drive the concurrent activation of certain data sources and classifiers as well.

Functional Requirements

Individual tasks. Each agent has to share her knowledge among a collection of agents that provides complementary skills to her ones, in order to (i) "measure" the context of the surrounding environment, (ii) infer a certain situation, so become aware that is happening something relevant, (iii) and finally adapt the behavior of the actuators accordingly. Human actions and interactions are crucial during the whole process, and they can be tracked by explicit or implicit sensing of data through both personal devices like smart phones or smart vehicles, and through public interactive displays.

Collective intelligence. The design of the software architecture capable of building the brain of the superorganism should be balanced between a top-down and a bottom-up approach. In top-down design approaches all the requirements are defined and satisfied a priori. This usually results in very predictable and measurable systems that lack in reactivity in high dynamic contexts.

On the other hand, bottom-up approaches based on spontaneous self-organization of agents, towards the emergence of a desired global behavior, suit to cope with pervasive computing in decentralized systems, but at the same time their behavior is not always predictable, nor easy to be engineered. Thus, in order to aim both at individual and collective goals, a trade-off solution between the two approaches must be pursued.

System safety. The system should own only a finite set of reachable states, which should be known in the design phase, and tested during the development. The aim here, is to avoid risks related to the eventual system's evolution towards uncontrolled situations. To enable this feature a deep understanding of system dynamics, and how to deal with them, is needed. In other words, citizens should feel safe to contribute in social collective intelligence initiatives, because they trust the superorganism and its potentials, and find it useful in any circumstance. The overall system dynamics should be deeply understood, in order to avoid risks related to the emergence of uncontrolled states.

This feature strongly impacts on the reliability and the potential perceived by the users, which would like to deal with social collective intelligence initiatives without worries.

Information propagation. The inferred information should pervade the nodes of the infrastructure till it can actually reach any potential agent that can be interested in it. Data should be packed in efficient structures, and routed via peer exchanges. A middleware architecture can be harnessed to reach these goals, and thus to support the purpose of the coordination framework, which is expected to become active supporter of agents interactions and facilitator of information propagation [29].

Data management. For the infrastructure to be practically usable, efficient storing, querying, and analysis of big amount of spatio-temporally distributed data is needed. Also, such approaches should take into account the need to distributed information across a set of multiple – and geographically distributed – tuple spaces. Several approaches can be found in literature ([110], and [86] among the others), even in the area of distributed tuple space infrastructures. However, a unified and sound approach, suitable to the need of SAMS, is still missing.

Users privacy. Discovering matches between needs and services implies computation on different data sources coming from a set of agents. This task of the superorganism can contemplate the sharing of confidential information among the involved agents. Privacy concerns and sharing policies must be dealt on user agreements and should consider innovative practices to balance the value of the data shared with the value expressed by service enabled through the sharing of someone else [157].

User incentives. The effectiveness of SAMS deeply relies on some behavioral changes that need to occur in how people are used to move within the city. Tailored incentives and mechanisms taken from persuasion theory can stimulate them (for an overview of perspectives enabled by the so called Persuasive Computing see [57]).

A reputation system can also establish an adequate level of trust needed to involve a greater number of users, as well introducing a monetary form of compensation can support the occurrence of social interactions.

Yet, it is still plenty of fascinating research challenges to make SAMS an every-day reality.

2.3.3 Exemplar Case Study

To evaluate the effectiveness of the proposed coordination framework in the provision of SAMS, a set of algorithms that aims to reproduce the main conceptual activities involved in the matching engine described above has been developed .

The efforts have been focused on an Itinerary Match and Ride Sharing service, as described in Section 2.2.1. The aim here is to evaluate potential matches between mobility needs and capabilities. Commuters with a similar typical daily route should be detected and recommended to join ride sharing opportunities.

Even if the framework is expected to collect real-time data coming from the distributed tuple space, an offline experiment has been undertaken, by simulating the matching engine activities on a large dataset previously collected.

Raw data involved in the study covers one week of detections in the city of Turin, and it consists in Call Description Records (CDRs) collected by a mobile network operator, through the cellphone network. However, one can assume that data can be collected opportunistically from a set of drivers, through an application installed on their smartphones, and propagated on the nodes of the infrastructure.

The work reported in this Section has been improved with the case studies presented in Chapter 3.

User	MMC	Time	Tower	Coord	Radius
3dd2b	222	7346286	123	(41.2,13.9)	450

Fig. 2.5 Structure of the CDR dataset. Every time a user send or receive calls and text messages, a CDR with information about the user (hashed) id, the MMC (Mobile Country Code), the timestamp of the event, the code of the cell tower and the coordinates and coverage radius of the cell tower, is generated.

Basically, each time a user starts a call, or sends a text message, a spatio-temporal record is created. In Figure 2.5 it is shown the structure of a CDR record. Each record comprises a user (hashed) id , the MCC (Mobile Country Code) representing

the country where the SIM card has been registered, the timestamp of the event, the code of the cell tower and the coordinates and coverage radius of the cell tower. Thus, the spatial resolution of CDR localization is the cell radius.

The following paragraphs describe the processing tasks to put at work on the collected CDRs in order to discover ride sharing matches and suggest tailored recommendations to users.

Towards Agents Classification

According to the conceptual model of the coordination framework, the first step involves an initial **pre-processing of raw data**. In this case study, data is filtered in a way that tries to exclude non-commuters.

In particular, commuters are defined as all the users that generate at least one event in both a pair of enough distant geographic zones (let's call them A and B), during working days. Furthermore, the definition of commuter is narrowed by considering two particular regions to perform that filtering. Urban mobility is the main topic that has to be studied, so an area that covers the inner part of Turin as the zone A (about 100Km² wide), and a geo-fence of a broader zone (about 3000Km² wide), which surrounds the city center (suburban area), as the zone B, are considered.

No considerations on the mean of transportation used by the users are done, because the input data is too fragmented and sparse. Best practice to succeed in this activity consists in excluding all the commuters that are used to move along railways, cycling paths, metro stations, or bus stops.

Next phase has involved the **spatio-temporal aggregation** of the selected CDRs into mobility traces. Data has to be modeled into upper-level meaning abstractions, useful to better understand mobility patterns of the considered users. A mobility trace is defined as the conjunction of a pair of temporally adjacent events, which represents the origin-to-destination path covered by a certain user in a defined temporal interval. This process has resulted in the detection of sequential mobility traces (the destination of the first matches with the origin of the second) that can cover wide areas and time intervals.

Each user is characterized by a long set of mobility traces that can be reduced in length by doing some further spatio-temporal aggregation. The aim here is to reduce the amount of data linked to each user, by merging the mobility traces through their sequential relationships. Thus, they shape brand new, more spatio-temporally extended, mobility traces. Mobility traces that were close each others have been merged together in both spatial and temporal domains. The amount of merging occurrences has also been stored in the resulting mobility trace.

Spatial closeness has been computed on the pair of geographical points that characterize the origin or the destination on the pair of the involved traces. This is easy to compute with any point-to-point distance formula (e.g.: haversine, euclidean). Temporal closeness has been computed on the time interval associated to the same pair of mobility traces. This task is more tricky and it concerns the evaluation of several

temporal relations. The approach presented by Van Beek and Manchak [134] has been followed.

For each user, the **mobility pattern recognition** phase has contemplated the inference of the most visited mobility path described by the mobility traces. In particular, this process has first resulted in the application of a K-means clustering algorithm on the spatial dimension of the mobility traces. The evaluation of the clustered points has been done on the amount of occurrences related to them.

Only the two most populated clusters have been evaluated (origin and destination candidates). The typical **daily route** of each user has been discovered. Figure 2.6 shows an extract of the daily routes in a 1-hour time lapse.

A daily route is defined as the more frequent outward and inward mobility traces generated by the same user from/to an origin to/from a destination. Actually, the daily routes mining has returned a significant result only for the 10% of the considered users, since most of the results have revealed the same amount of occurrences on multiple candidates in the same cluster (too much ambiguity on the data). This sudden loss of significance can be tackled by evaluating more temporally distributed data (one week of CDRs does not provide enough significance to the study).

The user's typical daily route can be useful to **detect and anticipate mobility events**. As examples:

- it describes the **daily journey** the user is used to perform (i.e., if the user is a commuter), and so it represents a daily event itself;
- it reveals the **expected presence** of an agent on the underlying road network, during a certain time span;
- it can be harnessed to anticipate any **expected traffic congestion** on the underlying road network at a certain time;
- it lets to locate a moving probe that can be queried just in case in the future to detect **mobility status and alerts**.

The contributions to the superorganism provided by each agent, whether it is a requesting agent or a potential service provider, have been used to classify them, by creating their **agent profile** (in the distributed tuple space) with classifying labels and higher information contents.

Towards Agents Recommendation

In this case study, each agent profile is related to a commuter, and it initially contains only information about its daily route. In a real case scenario it can additionally include other mobility routines, agent's personal details (such as demographics and interests) that can be collected through online social networks, its mobility preferences (such as its usual mean of transportation, and its willingness to do ride sharing), the most likely home and work locations, its belonging to a community of agents (due their commute similarities), and the most relevant historical events detected.

Furthermore, one can think to add a ranking information to the agent profile that quantifies how much it has been involved in crowdsourcing and collaborative

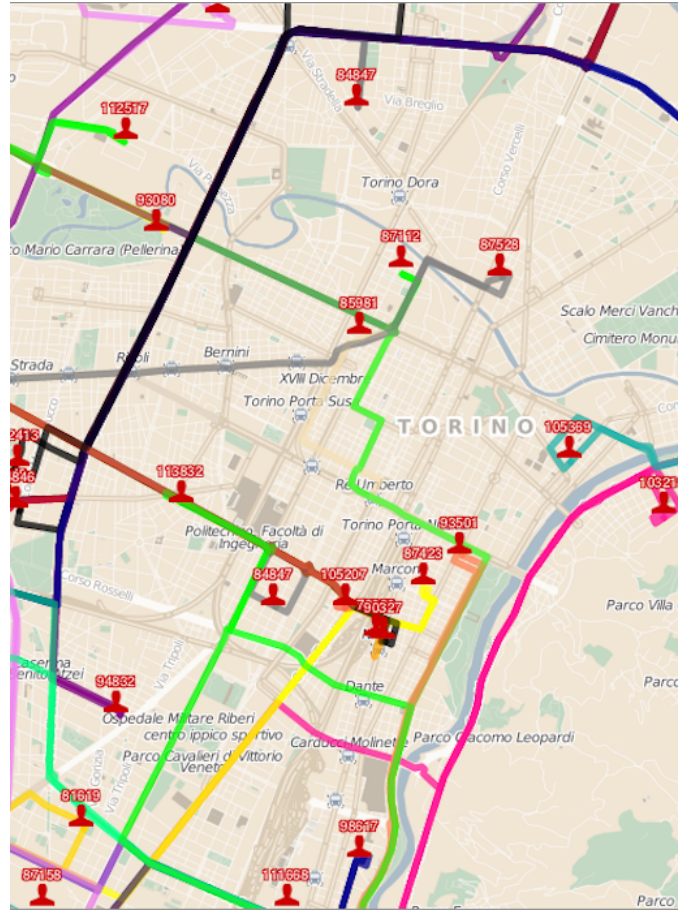


Fig. 2.6 Partial representation of the users daily routes in the city center of Turin at a given time interval.

initiatives. This data can narrowly reflect social interactions among agents and their resulting benefits. Thus, it can outline the rise of **collaborative group of agents**.

The next step in computation has required to mine pool of users with similar daily routes. This activity has been done through an exhaustive search on all the users, by assuming that all of them are currently moving alone, by driving a private car with 5 seats capacity. So, each user expresses at the same time its availability in offering ride sharing services, and its necessity to find more efficient mobility solutions (in terms of vehicle occupancy rate). The study has been limited to consider uniformed users that are characterized by the same mobility desiderata.

However, on mining pool of users, one should consider maximum detour distance and time admitted as individual factors of the driver and each one of the passengers. Of course, a further improvement of the algorithm should contemplate the evalua-

tion of the existing collaborative groups, in order to distinguish reliable users from untrustworthy ones.

For any driver, the pair of mobility traces that composes its daily route has been selected, and each one of those is then compared with the whole set of concurrent mobility traces (within a confidence time interval) generated by other users. All the compared users with a mobility trace detected along the one generated by the selected driver represent potential passengers in ride sharing pools, which can outline new potential collaborative groups of agents. The resulting pools contain information about the most suitable sequence of timed-stops to pick-up and drop-off passengers.

Once the pools are detected, **ride sharing recommendations** can be sent out. Their should push social interactions between the involved users, and let new collaborative communities to be shaped. The superorganism can track how they affect users mobility behavior and update both their individual and collaborative profiles.

2.4 State of the Art

Finding new approaches to enable SAMS has recently received a lot of attention. However, most of the studies are far from reaching autonomous, effective and integrated solutions (from the opportunistic collection of the requests to the provision of the most suitable recommended service).

As a *Parking Match* example, for instance, [109] proposes an approach that aims in discovering when a driver gets back to his/her car by exploiting Wi-Fi signed parking lots. Nevertheless, authors do not provide an automatism for direct matching the user that is leaving the parking and anyone that is looking for an available one.

As a *Itinerary Match and Ride Sharing* example, the work presented in [69] offers an accurate mobility analysis on GPS traces collected from 178 users to extract their regular routes and generate ride sharing plans among them. Despite the presented approach succeed in inferring recurrent mobility routines of users by starting from raw trajectories, the optimization algorithm put at work to assign users to shared cars does not suit with a real-time ride sharing service requirements, due to the need of processing batches of data at the same time. Mainly, every work that comes from Operational Research [16, 50, 61], in the *Multimodal Rides* example, presents the same limit.

As discussed previously, most of current ITS approaches do not offer a unified and integrated approach to support urban mobility in all its aspect, and often they own independent approaches for different mobility needs [5]. Also, in the proposed framework, and unlike most of ITS proposals, citizens are active agents of the overall infrastructure, by collaborating implicitly and explicitly towards the provisioning of SAMS.

Of particular interest to this work is the role of the urban superorganism, which supports interactions and information exchange among the agents on the socio-technical ecosystem, and its involved in the generation of distributed intelligence.

So far, the best examples in this field that deal with the underlying infrastructure are the work of Harnie et al. [68], which aims to specify urban-area applications with tuple spaces abstraction, and the work of Julien and Roman [80], which proposes a middleware to enable context-aware mobile applications. The former enables intelligence through moving buses that carry the tuples, the latter propagates intelligence through vehicle-to-vehicle short range communications.

The works of Qu et al. and of Yang et al. [114, 146] introduce the concept of Intelligent Transportation Spaces as the integration of various ITS modules, vehicles, and roadside infrastructure. They mainly analyze safe and effective communication technologies to enable pervasive intelligence without impacting too much on drivers workload. However, neither of the works mention social interactions in matching mobility needs and services.

The growing interest in SAMS is testified by a number of research projects and research approaches that address issues related to the definition of general architectures for their effective provisioning. Most of the approaches and project so far, such as the the EU-funded MOBINCITY¹ and SYNCRO² projects, aim at developing the hardware and networking infrastructures to support SAMS, but they do not strive for the development of a general purpose framework to act as engine for the actual building and provisioning of their services.

More on the software side, the MyWAY³ project has the goal of developing a general centralized server architecture where to collect information about the possible mobility services and the actual mobility needs of a town, so as to facilitate users in selecting the best ways to satisfy their mobility needs. This makes MyWAY quite in line with the vision presented, although it does not touch the issue of dynamically composing mobility services from potential capabilities.

Concerning research approaches, a number of special-purpose algorithms and software frameworks have been proposed to tackle specific aspects of SAMS (e.g., dynamic ride sharing, taxi sharing, park search), as it can be seen in the recent proceedings of conferences like the *IEEE Conference on Intelligent Transportation Systems* or the *IEEE Conference on Pervasive Computing and Communications*.

Existing works in literature do not give their contributions on proposing new approaches that could enhance social interactions.

Most of the mobility services presented in literature (as, e.g., [59, 74]) merely offer tailored solutions, without worrying about the creation of a coordinated methodology that deals with the dynamic orchestration of heterogeneous data streams.

A general-purpose comprehensive and unified approach grounded on the envisioned framework, which models sensing, computing, and actuating capabilities of a socio-technical ecosystem of mobility agents, is currently missing.

¹ <http://www.mobincity.eu/>

² <http://www.syncromobility.eu>

³ <http://www.myway-project.eu/>

2.5 Conclusions

Social interactions among humans and ICT devices could strengthen the awareness of what urban mobility needs are, and how they can be addressed with SAMS. Social collective intelligence can be enabled, and so its utility can hit citizens and convince them to collaborate and cooperate each others through innovative sharing practices regulated by suitable incentives.

The ICT enabled environment is becoming dense and the penetration of smart phones has reached high levels in several cities around the world already. These facts bring relevant contributions to the envisioned Smart City scenario. In the future, the more heterogeneous the ICT urban infrastructure will be, the better Context Awareness will be enabled by the socio-technical system.

Chapter 3

SAMS: Ride Sharing

Recent developments in smart phones and social networking tools allow collecting large amounts of data about mobility habits. These data can support advanced forms of *ride sharing* (a.k.a. car pooling). In this chapter, a couple of methodologies are presented, which have been applied to specific case studies, and based on mobility traces mining, to identify rides along the same trajectories that are amenable for ride sharing. These results can be used to evaluate the potential of a ride sharing system before its actual deployment and to actually support an opportunistic ride sharing recommender system.

3.1 Introduction

The increasing adoption of smart phones and social networking tools allows to collect large-scale information about mobility habits of people, and can support advanced forms of sharing, coordination and cooperation that can drastically reduce the overall demand for mobility [12, 42, 130, 150].

Ride sharing is a scheme in which people share a vehicle in order to reach common or nearby destinations. Despite its clear advantages in reduced costs, pollution, or time spent on finding a car park, there are still a few obstacles that prevent it from being the preferred way to move, like safety of passengers or sub-optimal mobility matches, among others.

First, people are concerned about safety, as the driver, or the other passengers, may be not known well enough in advance. Second, it is often difficult to find a good match between systematic movements that would enable two or more users of the system to find effective savings by sharing a car. Third, besides the clear advantages, additional incentives promoted by carpooling operators, mobility managers, or cities, may not be enough to encourage the needed behavioral switch. In addition, there is the risk of not enjoying the trip, i.e. people may be worried about sharing boring trips with a group of people on a everyday basis for work-home traveling.

While there exists a number of research and commercial proposals supporting ride sharing for *planned* and *long-distance* travels (e.g., BlaBlaCar¹), the novel opportunities offered by pervasive technologies can be used to shift ride sharing to a on-the-fly enabled service for *short-term* journeys within the city. In fact, pervasive technologies and mobile computing allow to automate the process of identifying and negotiating matching rides among users without requiring them to explicitly enter their availabilities and needs. This kind of unobtrusive and proactive approach could facilitate the spread of related applications and improve their overall effectiveness [58].

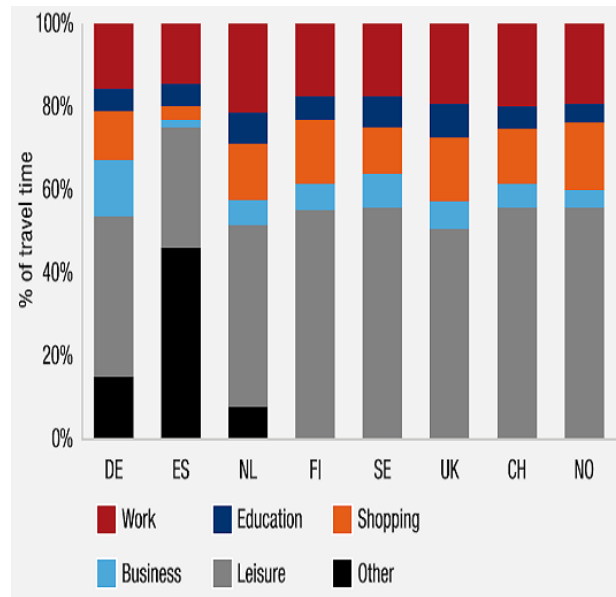


Fig. 3.1 Reasons for travel in percentage of travel time for selected European countries (Eurostat (2007): Passenger mobility in Europe: Europeans spend most of their travel time in cars. Statistics in focus. Transport 87/2007 (ISSN 1977-0316). Image taken from [132].

As reported in [132], mobility demand spans all aspects of our life and although home-work commute is an important part, also other mobility demands have a relevant effect as showed in Figure 3.1. Accordingly, effective systems to reshape mobility demand have to consider all aspects of our life, including free time and leisure.

¹ <http://www.blablacar.com>

3.1.1 Goals

Ride sharing, i.e. the sharing of vehicles to reach common destinations, or, more in general, to ride along common paths, is often performed to reduce costs and pollution. In the last years social networks have gained more and more value in exploring the interests/hobbies of the people for a variety of purposes, ranging from commercial to political fields. The goal is to exploit the main advantage of the Ride sharing idea, i.e. lower number of cars on the road, but also to ensure that the passengers will enjoy traveling together and sharing the same interests. This is a crucial point to motivate people to share their rides instead of traveling alone. On the one hand, two methodologies to extract suitable information from mobility traces, and so to identify users' mobility patterns, are presented. On the other hand, the advantages in terms of mobility reshaping of ride sharing opportunities, by taking into accounts the influence of both mobility and social measures, are evaluated.

3.1.1.1 Mobility Measures

The core enabling conditions of ride sharing are related to the mobility profiles of people. Drivers of private vehicles, which depart from the same neighborhood, at the same time, and drive along the same main road, or the ones that need to reach the same destination, by the same time, and have a portion of the journey in common, are potentially amenable to ride sharing recommendations. Next Section 3.2 proposes a set of unobtrusive tools to analyze mobility demand and to find mobility matches for the recipients of such recommendations.

3.1.1.2 Social Measures

Based on the assumption that the attractiveness of ride sharing could increase if besides mobility match also social aspects and, more generally, non-monetary rewards are taken into account, next Section 3.3 studies the impacts of social measures on ride sharing by proposing a methodology that models pair of users in terms of social demand, to offer them *more enjoyable* ride sharing recommendations.

3.2 Mobility Demand Study

This use case aims at studying how mobility can be reshaped by deploying an opportunistic ride sharing system endowed with an advanced mobility analysis tool and a set of algorithms to link together users with similar mobility trajectories, thus to provide them with ride sharing recommendations. Accordingly, the key novelty of the proposal is that the system is fully autonomous. Users do not enter their availabilities and needs explicitly. Sharing opportunities are automatically identified by

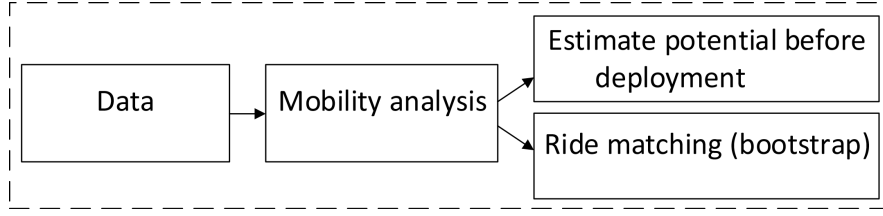


Fig. 3.2 Overview of system architecture. The proposed recommender system extracts mobility routine from telecom dataset. Then uses a ride matching algorithm to identify ride sharing opportunities.

analyzing people mobility patterns, and are recommended when suitable conditions arise. This kind of unobtrusive and proactive approach could facilitate the spread of related applications and improve their overall effectiveness [58].

Overall, an algorithmic suite to create a ride sharing recommender system is proposed (see Figure 3.2). The system accesses mobility data (*Call Detail Records* - CDR) representing people whereabouts. Then, it runs a series of data mining algorithms to extract home-work commutes and general mobility routines of individuals. On this basis the recommender system can match multiple users' routine to enable ride sharing. Since CDRs are collected from a large user population (depending on the telecom operator's market share) without the need of any application installed on the users' phone, the analysis can support the estimation of ride sharing potential before its actual deployment. Moreover, it can support the bootstrap phase of the recommender system when the application still does not have enough GPS points to infer user's mobility routines (by assuming that any ride sharing application - once installed - will collect GPS traces of user mobility with a better spatio-temporal resolution than CDR ones).

Two main scenarios are evaluated. One is characterized by riders of users' pools that walk from their origin location to the most suitable meeting point for being picked up and from a drop-off point, on driver's route, to their destination. The other one assumes that drivers are going to divert from their original path to collect riders and to drop-off them at their destination. Figure 3.4-(b) depicts better these concepts.

3.2.1 Problem formulation

The ride sharing problem taken into account in this use case is modeled as a classical mobility-based ride sharing problem where mobility data is processed in batches. Each batch basically contains a set of mobility trajectories identified by a time-stamped origin location and a time-stamped destination location. The peculiarity of this computational system is that it could provide promising results with no effort required from users in planning and publishing their trips. The goal is to conceive a fully autonomous recommender system able to achieve a lower amount of circulat-

ing cars on the road, in which users just need to accept/reject the suggestions, since the data is automatically collected and processed.

3.2.2 Data Source and Methodology

This section firstly describes an approach to automatically extract home and work places for each user from a set of CDR data, and then it presents some experiments to evaluate accuracy of the extracted results. On these basis, the average commuting hours for each user are identified. Assuming that each day of the week users commute between home and work, the number of rides that could be shared on the basis of a number of parameters (e.g., willingness to tolerate detours or time delays) is estimated.

A large set of mobility data has been collected and obtained from an Italian telecom operator. In particular, it contains anonymized CDR data for about 3.5 million persons living in the same region spanning 1 month (CDR data format has been depicted previously in Figure 2.5).

For each user, CDR data logs the approximate location of the user whenever (s)he use the phone to send or receive calls or text messages. The user's location is given in terms of the cell network antenna the user was connected with. The area covered by a given antenna can be approximated by a circle with a given center and radius.

It is worth noticing that, in contrast to a number of other works, this work does not try to improve the estimation of the coverage of a cell network by using Voronoi tessellation. A simpler identification of a cell represented by a circle with a given center and radius is chosen. In [133], it is shown that the approach do not change the user's location accuracy.

Figure 3.3-left illustrates the daily average number of CDR produced for a given percentile of users. It is possible to see that a large fraction of the users produces a large number of CDR events per day. In particular, the most active 10% produces more than 10 CDR events per day thus allowing a rather fine-grained mobility tracking. Of course, the resolution of CDR data is far inferior to GPS traces than can pinpoint user location on a minute by minute basis with about a meter resolution.

Figure 3.3-right illustrates the radius of gyration for a given percentile of users. The radius of gyration is a synthetic and easy-to-compute parameter describing the spatial extent of user traces. It is defined as the deviation of user positions from the corresponding centroid position, and it is given by:

$$r_g = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - p_{centroid})^2}$$

where p_i represents the i^{th} position recorded for the user and $p_{centroid}$ is the center of mass of the user's recorded displacements obtained by:

$$p_{centroid} = \frac{1}{n} \sum_{i=1}^n (p_i)$$

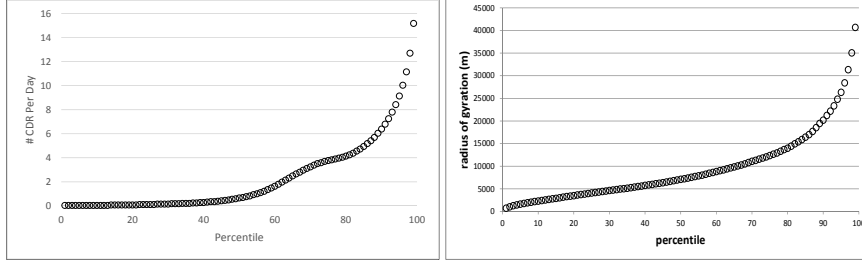


Fig. 3.3 Daily average number of CDR produced for a given percentile of users (**left**). Radius of gyration for a given percentile of users (**right**).

It is possible to see that the first quartile tend to be associated to sedentary people with $r_g < 5Km$. The bulk of the distribution (25^{th} - 75^{th}) percentiles can be associated to urban mobile people as the diameter of peri-urban areas of main cities in the region is about 15Km. Users beyond the 75^{th} percentile are associated to commuters spanning on the wide region area.

All the available trips are temporally and spatially aggregated by matching similar values in each of the fields. Temporal windows of 1 hour on daily commutes and of 2 hours on general mobility routines are set.

As mentioned above, to deal with space, the actual street networks is used: given *OriginLocation* and *DestinationLocation* for each trip, OpenStreetMap is used to acquire a topological representation of the path between them. Furthermore, the capacity boundary of cars is considered, so the aggregation of more than 4 trips together is avoided. The algorithm used for spatially aggregating trips is further detailed in Algorithm 1 and works as follows.

Given an initial trip (i.e., trip A), and another trip (i.e., trips B, C, D) sufficiently close in time, three cases might arise:

- *Trip B*. Both source and destination of trip B are within a radius r from a node of Trip A. In this case, Trips A and B are aggregated together, the user associated to Trip A becomes the driver of the aggregated Trips, while the user of Trip B becomes a rider of a potential users' pool.
- *Trip C*. None of source and destination of trip C are within a radius r from a node of Trip A, but both source and destination of trip A are within the same radius r from a node of Trip C. In this case, Trips A and C are aggregated together, the user associated to Trip C becomes the driver of the aggregated Trips, while the user of Trip A becomes a rider of a potential users' pool.
- *Trip D*. None of source and destination of trip D are within a radius r from a node of Trip A and viceversa. In this case, Trip D is considered as a new trip that could be eventually aggregated with others.

Algorithm 1 prioritizes the matching with any trip of the already discovered potential users' pools (lines 4-9). If the available pools do not offer any possible match,

Algorithm 1 - Assignment Procedure ($\mathcal{T}$)

```

1:  $\mathcal{P} \leftarrow \text{emptyPoolsCollection}$ 
2: for  $i \in \mathcal{T}$  do
3:   if  $\text{assigned}(i) == \text{FALSE}$  then
4:     for  $j \in \mathcal{P}$  do
5:        $PA_j \leftarrow \text{getRidersAssigned}(j)$ 
6:       if  $PA_j < 4$  and  $\text{match}(i, j) == \text{TRUE}$  then
7:          $PA_j \leftarrow PA_j \cup \text{getUser}(i)$ 
8:          $\text{assigned}(i) \leftarrow \text{TRUE}$ 
9:         break
10:    if  $\text{assigned}(i) == \text{FALSE}$  then
11:      for  $k \neq i \in \mathcal{T} : \text{assigned}(k) == \text{FALSE}$  do
12:        if  $\text{match}(i, k) == \text{TRUE}$  then
13:           $\text{assigned}(i), \text{assigned}(k) \leftarrow \text{TRUE}$ 
14:           $\mathcal{P} \leftarrow \mathcal{P} \cup \text{newPool}(i, [k])$ 
15:          break
16:        else if  $\text{match}(k, i) == \text{TRUE}$  then
17:           $\text{assigned}(i), \text{assigned}(k) \leftarrow \text{TRUE}$ 
18:           $\mathcal{P} \leftarrow \mathcal{P} \cup \text{newPool}(k, [i])$ 
19:          break
20:    if  $\text{assigned}(i) == \text{FALSE}$  then
21:       $\text{assigned}(i) \leftarrow \text{TRUE}$ 
22:       $\mathcal{P} \leftarrow \mathcal{P} \cup \text{newPool}(i, [])$ 
23:    break
24: return  $\mathcal{P}$ 

```

the algorithm looks for any match with the concurrent trips (by the temporal window) not aggregated yet (lines 11-19). If neither the individual trips offer a solution, then the trip is considered as a new pool for aggregating the following trips (line 22). The three cases of aggregation strategies are depicted in Figure 3.4-(a).

Overall, the identified shared rides can be used as a benchmark to evaluate the possible impact of a ride sharing system before its actual deployment (i.e., before the ride sharing application has been installed by a critical mass of users). The methodology, using CDRs, can also be used in a bootstrap phase of the system use when the ride sharing application still does not have collected enough GPS points to infer users' mobility routine (after then bootstrap phase, GPS data will likely outperform CDR measures).

3.2.3 Mobility Analysis

The analysis of CDR data allows to identify the mobility patterns of users. This kind of analysis is fundamental for an autonomous ride sharing system because on the one hand it allows to predict where a user is heading. Thus, the system can give sharing recommendation in advance. On the other hand, it allows to predict mobility routines where data is missing (if on a given day the user does not use his/her phone,

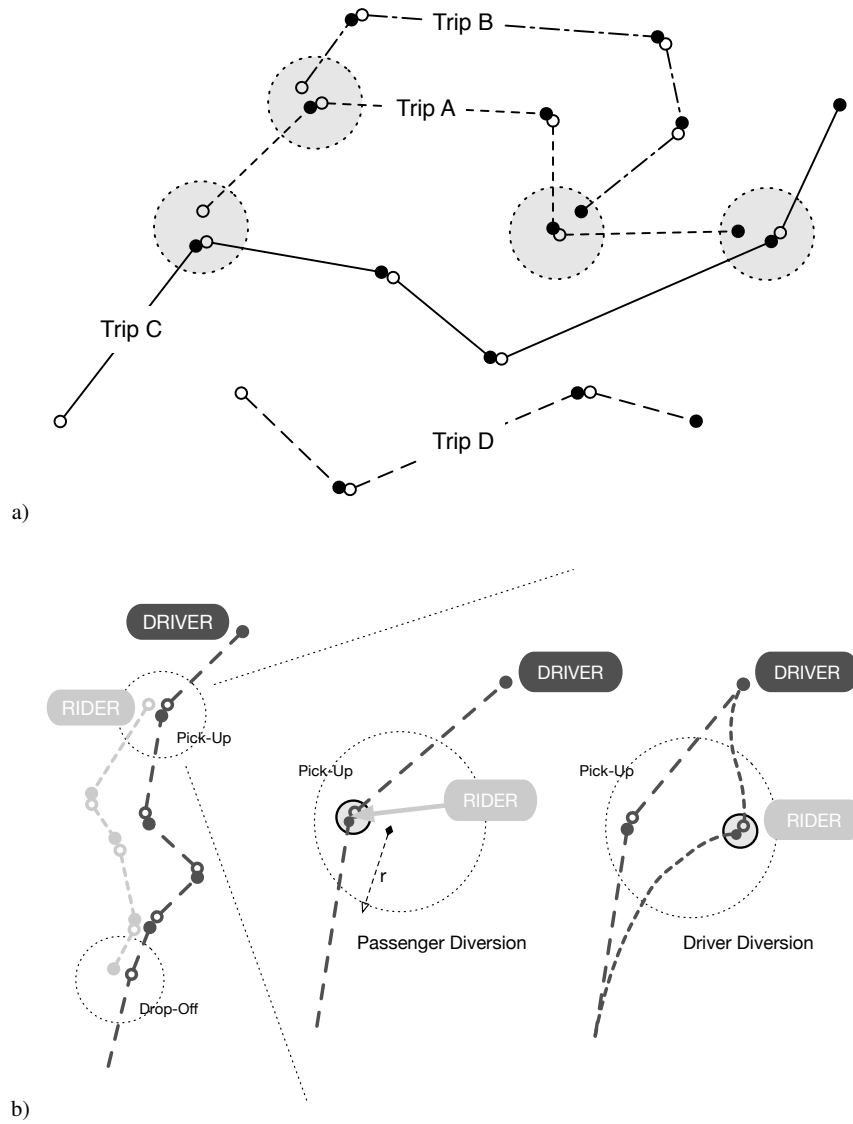


Fig. 3.4 Three cases of aggregation strategies for the analyzed urban trips (a). Further, an example of rider and driver diversions (b).

the system does not have any information about the user's whereabouts. Nevertheless, via pattern analysis, it can infer user's mobility on the basis of past behaviors).

This Section details the methodology put at work to study and to identify the mobility profiles of the considered users.

3.2.3.1 Daily Commutes Identification



Fig. 3.5 Proposed approach to identify home and work places. (1) CDR events for each user are selected. (2) A clustering algorithm partitions events into well-defined spatial regions. (3) A weighing mechanism gives each event a weight on the basis of the days and times in which the event is generated and also on the basis of other events nearby. (4) Home and Work places are selected as the center of the most clusters weighted clusters.

Following an approach similar to [27, 75], all the mobility events of each user are considered for clustering.

The proposed approach is illustrated in Figure 3.5. (1) For each user, all his/her CDR events are collected. (2) Then, events are clustered on the basis of their spatial distances so that the resulting clusters identify well-specified geographical areas. (3) Clusters are then weighted on the basis of the number of days in which the user has been there. (4) The centers of the heaviest clusters are considered to be home and work places. Following paragraphs go more in detail.

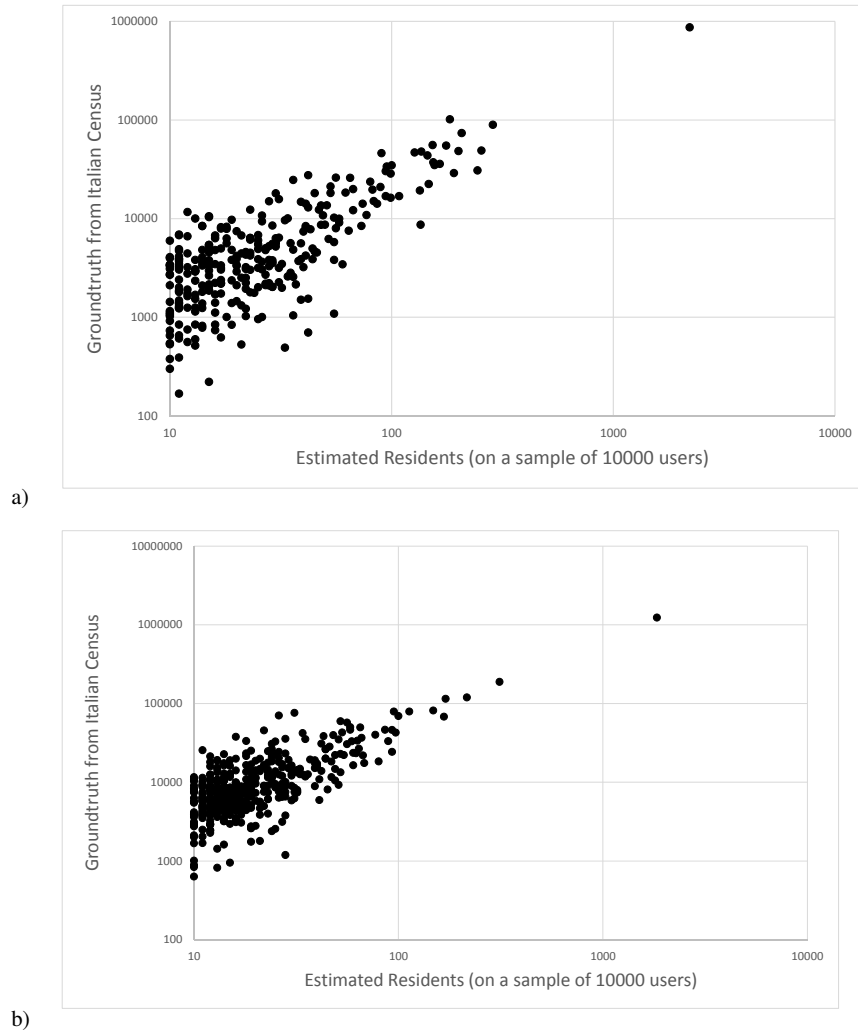


Fig. 3.6 Correlation between the number of people living in Piemonte (a) and Lombardia (b) as measured by the proposed approach and by the national census.

Clustering

Since several geographic areas are covered by multiple cells, CDR events produced from the same area can be associated to anyone of the covering cells. Thus, treating cells as disjoint entities can hamper the recognition of relevant places. For example, it can happen that the CDR events of a user calling from a given place (e.g. home) are scattered across many cells. Each cell will thus receive only few events, and it will resemble a place that is not much frequented.

Accordingly, to avoid CDR events fragmentation into disjoint cells, CDR events happening close to each other are clustered into coherent groups. After testing a number of algorithms, a simple agglomerative algorithm using the geographic distance among CDR events as the basis for clustering [76] is adopted.

The motivations behind this choice are the following: (i) all the algorithms (e.g., K-Means and Spectral Clustering) requiring as input the number of clusters to be found are rather inadequate. This is because the CDR events can be spread over a large area with a very large and unpredictable number of clusters. (ii) Density-/competitive- based algorithms (e.g., DBScan, SOM) solve the issue, but do not easily constrain the size of the clusters to be identified [76]. For example, in the some early experiments DBScan was adopted, but the need of relying on a rather complex mechanism to constraint the size of the clusters rose.

Constraining the size of the clusters is important in the context of CDR data, as the number of cells that can be associated with a certain position is limited to the cells "around" that position, thus the resulting cluster will have necessarily a limited scope.

Since the basic idea of the agglomerative clustering algorithm is to cluster the elements that fall within a given radius from a given place, this algorithm is a good starting point.

Weighting

Relying on commonsense assumptions, a time window associated to "home-based events" comprising all the events happening from 9pm to 6am, and a time window associated to "work-based events" comprising all the events happening from 11am to 4pm, were both defined. A "home-weight" and "work-weight", equal to the number of days in which the user has at least one event respectively, were associated to each cluster. A cluster with a high weight is likely to contain the user home/work place.

Thresholding

Home and work places are selected among the weighted clusters. The assumption, here, is that, considering the weight distribution of all the clusters, home and work places should be outliers with respect to that distribution, in that they are visited much more often than other places.

Following [142], the mean μ and standard deviation σ of the weight distribution (for home and work places separately) is computed. Then, all the clusters having a weight greater than $\mu + k\sigma$ are marked as *outliers*. These represent the areas around homes and the work places. In the analysis, $k = 1$ was empirically found as producing good results. Once a cluster is selected, the corresponding place (home or work) to the centroid of the cluster itself is set.

Validation

Given the lack of ground-truth information about people habits, it is difficult to provide sound measures. Since only home-work commutes are involved, all the mobility data generated on Saturdays and Sundays are excluded. Furthermore, a random sample of 10,000 individuals is considered, and the results of the place identification algorithm are compared with data coming from the Italian census, for both Lombardia and Piemonte regions.

Figure 3.6 illustrates the results that well conform to the linear correlation. The graph is in a log-log scale to compress the variation of sizes among cities. A linear regression model is run – forcing the intercept to 0 – on the data. The best linear model for Piemonte is: $census = 369.6 \cdot telecom$. This model has correlation coefficient 0.91. The best linear model for Lombardia is: $census = 656.9 \cdot telecom$. This model has correlation coefficient 0.90. It has been verified that correlation coefficient do not change notably when considering only larger cities.

Once home and work places of individuals have been identified, it is possible to estimate their home-work commuting behavior. For each user, their home and work places are represented on a map. Then, they are connected with the shortest path in the road network of the city (street map has been gathered via *openstreetmap.org*).

3.2.3.2 General Routines Identification

As discussed in the introduction, home-work commute is not the only mobility pattern affecting the overall mobility demand. Accordingly, effective systems aiming at reshaping mobility demand have to consider all aspects of the whereabouts. In this section, a topic modeling approach generalizing home-work commute is first presented. The topic modeling approach identifies recurring user habits characterizing the typical behavior of individuals. Then, experiments for evaluating the quality of the extracted patterns are evaluated. On these basis, the number of rides that could be shared is finally estimated.

Following the approach proposed in [51, 53], a topic model [15] to identify the user mobility routines is applied. From a general perspective, a topic model works in two steps. First, user mobility traces are represented as bags of words. In this use case, a word for each movement of the user has been created. Then, the actual topic model finds regularities and repeating patterns in those words. Such patterns (called topics) describe users' typical whereabouts.

Bag of Word

All the CDR events of a user are grouped on a daily base. Then, a sliding window over the events of each day is run, to consider couples of subsequent events. For each couple a word consisting of the location of the first cell, the location of second cell, and the time stamps in which the trace takes place, is created. In particular, 24 time labels, one for each hour of the day, are considered.

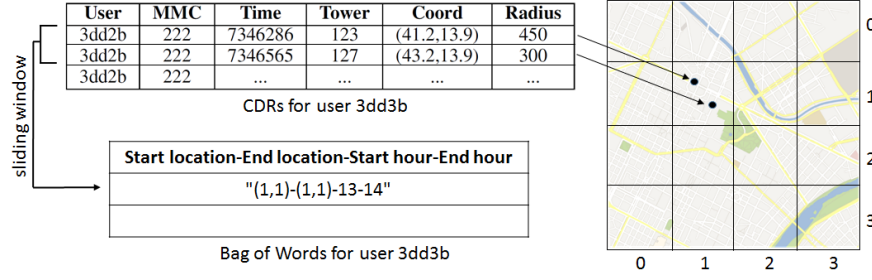


Fig. 3.7 Bag of words representation. For each couple of CDRs, a word consisting of the location of the first cell, the location of second cell, and the time stamps in which the trace takes place, is created.

The result is a set of words each one containing 2 location letters and a time label. Each word also keeps a reference to the actual geographic coordinates in order to localize the place, and the distance constrained by the street topology that is covered during the transition.

The resulting *bag of words* summarizes the original dataset and it is the input data structure for the algorithm to extract routine behaviours (i.e., topics) and add semantics (see Figure 3.7).

Latent Dirichlet Allocation (LDA)

LDA is a probabilistic generative model [15] used to cluster documents according to the topics (i.e., word patterns) they contain. LDA is an unsupervised learning mechanism that do not require a labeled (difficult to be acquired) training set. Moreover, LDA has two key advantages compared to other clustering mechanisms such as k-means: (i) the LDA model allows complex representation of topics. More specifically, it allows probabilistic distributions of days given all topics whereas other clustering algorithms assigns only one cluster per day; (ii) topics can be represented as meaningful word distributions. Topics are based on discriminative location sequences characterizing routines [53].

More in detail, LDA is based on the Bayesian network depicted in Figure 3.8. A word w is the basic unit of data, representing user location at a given *time-label*. A set of N words defines a day of the user. Each user has a dataset consisting of M documents. Each day is viewed as a mixture of topics z , where topics are multinomial distributions over words (i.e., each topic can be represented by the list of words associated to the probability $p(w|z)$). For each day i , the probability of a word w_{ij} is given by:

$$p(w_{ij}) = \sum_{t=1}^T p(w_{ij}|z_{it})p(z_{it})$$

where T is the number of topics. $p(w_{ij}|z_{it})$ and $p(z_{it})$ are assumed to have multinomial distributions. Mixture parameters are assumed to have Dirichlet distributions

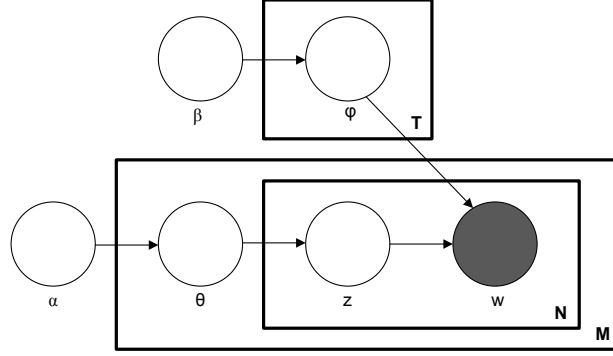


Fig. 3.8 Plate notation representing the LDA model. α is the parameter of the uniform Dirichlet prior on the per-document topic distributions. β is the parameter of the uniform Dirichlet prior on the per-topic word distribution. θ_i is the topic distribution for document (day) i , ϕ_j is the topic distribution for word j . These variables are modeled with a Dirichlet distribution. z_{ij} is the topic for the j -th word in document i , and w_{ij} is the specific word. These variables are modeled as multinomial distributions. The w_{ij} are the only observable variables, and the other variables are latent variables.

with hyper-parameters α and β respectively. Hyper-parameters α and β were both set to 1 representing an uninformative uniform distribution. LDA uses Gibbs sampling to learn the model parameters. The implementation involved the use of the library Mallet² to perform these computations. Once the model parameters have been found, Bayesian deduction allows to extract the topics best describing the routines of a given day (rank z on the basis of $p(d|z)$).

It is worth emphasizing that this work is not proposing an extension of the LDA model. However, it applies LDA in a novel way for identifying ride sharing opportunities.

Validation

Unfortunately, as discussed above, given the lack of accurate ground-truth information, it is difficult to provide sound measures about the accuracy of the obtained results (this is a general limitation of the current state of the art). To partially validate the results, a performance check on how well the LDA model actually *generalizes* the home-work commute described in Section 2 is done.

For each user, the two most probable locations extracted by LDA, associated to nightly and daily events, are computed. These locations are then compared with the home and work locations described above. For each user, the distance between home and work and the top locations based on topics (see Figure 3.9) is considered. It is possible to see that for more than 90% of the users, the distance between these points is less than 1 km.

² <http://mallet.cs.umass.edu>

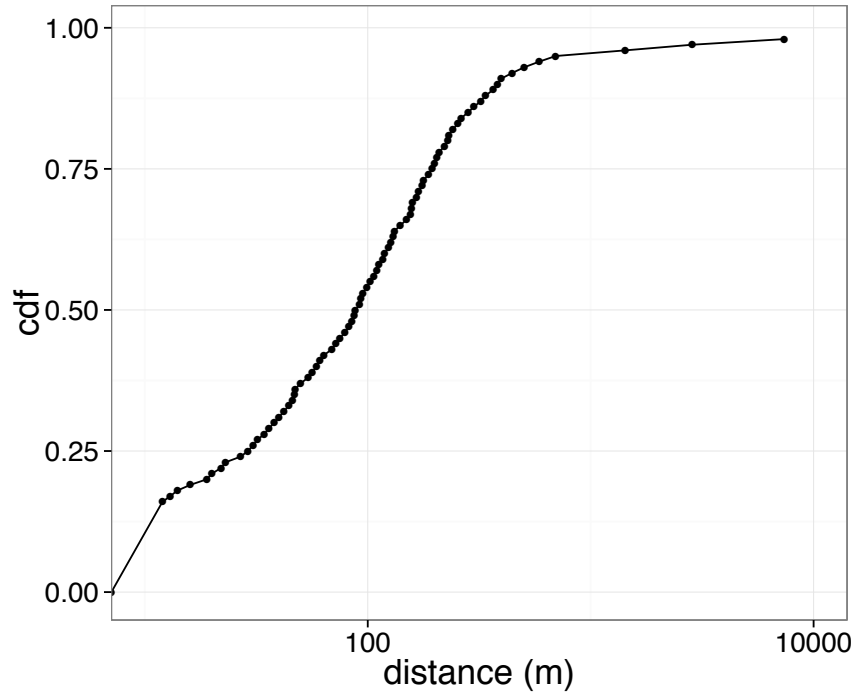


Fig. 3.9 Evaluation of topic modeling approach. fraction of daily movements that are represented in the top words of the most probable topic.

3.2.4 Experiments

To assess the proposal, experiments were run on two of largest Italian cities (i.e., Torino and Milano), by considering both (i) daily commutes and (ii) general mobility routines, as well as the cases in which (iii) passengers divert from their path to be collected from a driver that is passing by, and (iv) drivers take a detour in order to collect them. In each of the four originating cases, savings provided by the ride sharing assignments, in terms of both the amount of trips and km saved (see Figure 3.10), were measured. Each trip extracted from CDR data is described by a tuple like the following: *OriginTime*, *OriginLocation*, *DestinationTime*, *DestinationLocation*, *Route*. While time and location are straightforward, *Route* represents the list of locations composing the actual path between source and destination, constrained within OpenStreetMap network.

Regarding the cases based on riders diversions, the matching mechanism was managed by varying the distance they were willing to walk (*delta*) to reach a shared (see Algorithm 2). For each pair of driver-rider paths, the rider should not walk more than δ , neither driver nor rider have to wait more than a temporal window τ

at the pick-up meeting point (line 10). Furthermore, the expected drop-off time of the rider should happen not farther than τ with relation to her/his destination time (line 12). The achievements are depicted in Figure 3.10-(a,c). In Torino, the number of trips could be halved by adopting a large-scale trip sharing system and accepting to walk at most 500m to reach a shared vehicle. In Milano, instead, the number of trips could be halved by walking around 1000m. This difference is probably related to specific features (i.e., road network, people distribution) of the two cities.

Algorithm 2 - Matching Mechanism with Rider Diversion -
`match(driverPath, riderPath)`

```

1:  $\tau \leftarrow \text{temporalWindow}$ 
2:  $\delta \leftarrow \text{walkingThreshold}$ 
3:  $D_P \leftarrow \text{getPathPoints}(\text{driverPath})$ 
4:  $R_{OL} \leftarrow \text{getOriginLocation}(\text{riderPath})$ 
5:  $R_{DL} \leftarrow \text{getDestinationLocation}(\text{riderPath})$ 
6:  $R_{OT} \leftarrow \text{getOriginTime}(\text{riderPath})$ 
7:  $R_{DT} \leftarrow \text{getDestinationTime}(\text{riderPath})$ 
8:  $\text{match} \leftarrow \text{FALSE}$ 
9: for  $p_i \in D_P$  do
10:   if  $\text{distance}(p_i, R_{OL}) \leq \delta$  and  $\tau \leq |R_{OT} - \text{getTimeAt}(p_i)|$  then
11:     for  $p_j \neq p_i \in D_P$  do
12:       if  $\text{distance}(p_j, R_{DL}) \leq \delta$  and  $\tau \leq |R_{DT} - \text{getTimeAt}(p_j)|$  then
13:          $\text{match} \leftarrow \text{TRUE}$ 
14:         break
15:       if  $\text{match} == \text{TRUE}$  then
16:         break
17: return  $\text{match}$ 

```

Instead, when drivers are supposed to make diversions for collecting other passengers in their car, their willingness to share their rides could be influenced by the entity of the diversions related to the length of their routes if they would drive in single occupied vehicles. Thus, in these scenario, the spatial threshold on the percentage distance covered by drivers was varied.

In Algorithm 3, possible assignments were computed by assuming that drivers would make a maximum diversion in distance equal to a certain percent (δ) of their original path (D_D) per each rider they collect and another equal diversion per each rider they drop off. According to this relation, one could assume that, in order to keep positive savings on the amount of driven km, a minimum distance constraint on the potential rider's original route (R_D) equal to 2 times the detour ratio threshold is required: $2 \times \delta \times D_D \leq R_D$ (line 11). Spatio-temporal closeness is evaluated both for pick-up and drop-off meeting points (lines 12-19).

In this study, to overcome the complexity of the problem, haversine distances [135] between the source-destination pair of points of the passenger and their closest points within the route of the driver have been exploited. However, each route is computed on the actual road network topologies. Due to this, the minimum length constraint on the rider's route does not assure to keep positive savings of driven kms.

Algorithm 3 - Matching Mechanism with Driver Diversion -
 $match(driverPath, riderPath)$

```

1:  $\tau \leftarrow temporalWindow$ 
2:  $\delta \leftarrow detourRatioThreshold$ 
3:  $D_P \leftarrow getPathPoints(driverRoute)$ 
4:  $D_D \leftarrow getPathDistance(driverPath)$ 
5:  $R_{OL} \leftarrow getOriginLocation(riderPath)$ 
6:  $R_{DL} \leftarrow getDestinationLocation(riderPath)$ 
7:  $R_{OT} \leftarrow getOriginTime(riderPath)$ 
8:  $R_{DT} \leftarrow getDestinationTime(riderPath)$ 
9:  $R_D \leftarrow getPathDistance(riderPath)$ 
10:  $match \leftarrow FALSE$ 
11: if  $2 \times \delta \times D_D \leq R_D$  then
12:   for  $p_i \in D_P$  do
13:     if  $distance(p_i, R_{OL}) \leq \delta \times D_D$  and  $\tau \leq |R_{OT} - getTimeAt(p_i)|$  then
14:       for  $p_j \neq p_i \in D_P$  do
15:         if  $distance(p_j, R_{DL}) \leq \delta \times D_D$  and  $\tau \leq |R_{DT} - getTimeAt(p_j)|$  then
16:            $match \leftarrow TRUE$ 
17:           break
18:       if  $match == TRUE$  then
19:         break
20: return  $match$ 

```

Then, this constraint has been adapted by a multiplying factor that should help to save more km by doing ride sharing (α):

$$\alpha \times 2 \times \delta \times D_D \leq R_D$$

In the experiments, best results have been achieved with $\alpha = 2$. The amount of both saved trips and km decreases while increasing of α as shown in Figure 3.10-(b,d).

It is worth emphasizing that both the cases allow interesting results. The case concerning passengers diversions could lead to a reduction of 50% of the number of circulating cars by accepting to walk at most 1,000m (depending on the city) to reach a shared car. Drivers diversions, instead, are less efficient allowing at most a reduction of 40% of circulating cars in the best simulated case. However, even though the number of actually driven kms might increase because of the diversions drivers are asked to take, the reduction of circulating cars could lead to a reduction of many other traffic-related issues. It is possible to see that these results are in line with those reported by UberPool [54].

Finally, it is interesting to note how specific features of the observed cities led to noticeably different results. For example, in Torino the number of trips could be halved by asking riders to walk 500m, while in Milano the same result is achievable by walking almost 1000m. It is likely that these peculiar differences are related to the structure of the city, the topology of the road network, the spatial distribution of home and work places. Understanding these factors, might lead to the possibility of building more efficient cities in future.

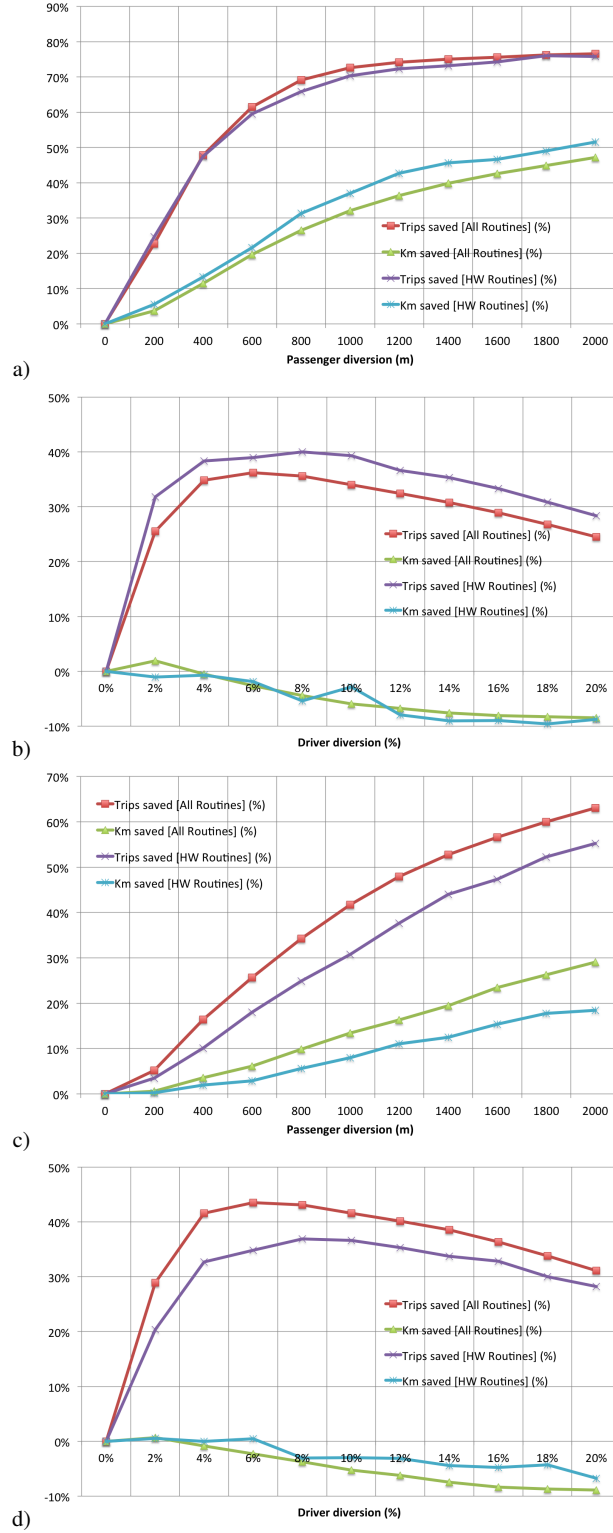


Fig. 3.10 Experimental results for both Torino (a,b) and Milano (c,d). In both the cases, the number of trips can be halved by adopting a large-scale trip sharing system and accepting to walk at most 1km to reach the shared vehicle (a,c). Instead, in case of driver diversions, the number of trips can be reduced up to 40% by accepting diversions (for each passenger) around 5% of the overall planned trip (b,d). Furthermore, it seems that Torino allows a sharper reduction of the number of trips compared to Milano. This evidence is probably related to specific features (i.e., road network, people distribution) of the two cities.

These experiments have studied the impacts of ride sharing by taking into account mobility demand exclusively. Furthermore, the results have been obtained by assuming that all the recommendations were followed by the users. Obviously, this represents a best-case scenario that is quite impossible to reproduce in reality. Next Section will extend the mobility demand by taking into account also social measures that should encourage the adoption of a ride sharing service.

3.3 Mobility and Social Demands Study

Since the sole study of mobility demand could drive to too much optimistic impacts on urban mobility, this section presents **GRAAL**, a data-driven methodology for *GR*een And *so*ciAL *ride sharing*, which gives the same influence to mobility and social demands on ride sharing impacts.

GRAAL optimizes a carpooling system not only by minimizing the number of cars needed at the city level, but also by maximizing the *enjoyability* of people sharing a trip. A measure of enjoyability based on people's interests, social links, and tendency to connect to people with similar or dissimilar interests, has been introduced. GRAAL computes the enjoyability within a set of users from crowd-sourced data, and then uses it on real world datasets to optimize a weighted linear combination of number of cars and enjoyability. To tune this weight, and to investigate the users' interest on the social aspects of carpooling, an online survey on potential carpooling users has been conducted. The 39%, over the 237 answers received, was in favor of a social solution.

The results of applying GRAAL on real world crowd-sourced data from the cities of Rome and San Francisco are presented from both the city and the user perspective. With the crowd-sourced weight, GRAAL is able to save up to 57% of the cars needed, while the total enjoyability is up to double. From the user perspective, how the entire per-car distribution of enjoyability is increased w.r.t the baselines is shown.

Inspired by works on smarter mobility like [39, 115, 130], this formulation of the Ride Sharing problem takes into account a data-driven methodology to automatically derive mobility and social matches to be used as recommendations for a Ride Sharing systems. Based on this, GRAAL optimizes a carpooling system, at the city level, not only by minimizing the number of cars needed, but also by maximizing the *enjoyability* of people traveling together.

Defining social enjoyability of a group of users is a challenging topic. An *enjoyability* measure that takes into account two factors can be proposed: (i) *like-mindedness*, i.e. a topic similarity between any two users; and (ii) *homophily*, i.e. the tendency of a person to group with similar ones.

Previous attempts try to put together in a car people who are friends [39]. However, by looking at only the direct (or even the two-hop) friends, good chances for optimization may be lost, as the set of potential drivers (or co-passengers) is usually much larger than the typical number of friend pairs in a social network. An alter-

native would be putting together people with *similar interests*, regardless of their friendship. However, not everybody may enjoy sharing an ordinary trip with people talking about similar topics: socio-cultural diversity is often researched to make a discussion more enjoyable.

Along these lines, can we come up with a measure, automatically computed from available data, grouping together people that are *socially compatible* according to both their interests and their potential preference for ending up in the same vehicle with people with either similar, or different interests?

To answer this, carpooling recommendations to people who potentially do not know each other, but who appear to be socially compatible according to their existing set of interests and friends, can be provided.

In GRAAL, a multiobjective optimization based on a weighted linear combination of two components is introduced: i) the number of cars (which is minimized) and ii) the total enjoyability of the users in the system (which is maximized). In the experiments, this weight, α , varies between 0 and 1.

Moreover, a crowd-sourced value for α is learned by means of an online survey. The survey has the double effect of both confirming the interest of potential carpooling users to a more social solution, and to provide an estimation of an α which is more meaningful to people. 237 answers have been collected, with 39% in favor of a social solution.

The results of applying GRAAL on real world crowd-sourced data from Twitter, geo-located in the cities of Rome and San Francisco are presented from both the city-wide and the user perspective. Then, they are compared with different baselines:

- a random model;
- a heuristic model aimed at maximizing the user enjoyability;
- GRAAL ran with a value of α that makes GRAAL minimize only the number of cars (which is derived from the state of art of carpooling);
- GRAAL ran with the opposite value of α , such that only the maximization of enjoyability is performed.

Results show that with the crowd-sourced α , GRAAL is able to save up to 57% of the cars needed, while the total enjoyability is up to double. From the user perspective, how the per-car distribution of enjoyability is increased w.r.t the baselines is shown. The algorithm is also compared with the methodology described in [39], to the extent of only the room for optimization, as the goals of the two approaches are different.

3.3.1 Problem formulation

The final objective of the carpooling optimization problem proposed is the minimization of the total number of cars in the system, together with the maximization of the *enjoyability* experienced by the users traveling together. The goal is to follow the main advantage of the carpooling idea, i.e. lowering the number of cars

on the road, while ensuring that the passengers will enjoy traveling together. This may serve as an additional, non-monetary, incentive to motivate people to share a car. The user study presented in 3.3.4.1 shows how the potential users are actually sensitive to this.

Given the enjoyability and mobility patterns similarity described in detail later in 3.3.3 and 3.3.4, the problem is formulated using an *Integer Linear Programming* (ILP) approach.

The starting point is a set of users that will be grouped together into cars. Within each car only one of the users is a driver while the others are defined as passengers. The number of drivers in the system indicates the number of cars allocated by the algorithm for the entire set of users. The grouping process is regulated by two aspects: i) trajectory containment; ii) enjoyability between users. The optimization procedure takes as input the enjoyability values and the set of recommendations and generates the optimal assignment A_{RU} .

From the recommendation set R_U three sets can be built: D , the set of candidate drivers in the system; P , the set of candidate passengers (D and P may overlap in the recommendations, but not in an assignment); C , the set of possible couples (i, j) driver-passenger.

It has been useful to define:

- a binary variable x_{ij} that describes the assignments between drivers and passengers, specifically $x_{ij} = 1$ if i is the driver of passenger j , $x_{ii} = 1$ if i is a driver;
- a binary variable y_{jki} indicating whether two passengers share the same car, specifically $y_{jki} = 1$ if passengers j and k share the same car with driver i ;
- a continuous number m_{ij} describing the normalized trajectory distance, with $m_{ij} \in [0, 1]$ if driver i can give a ride to passenger j . $m_{ij} > 1$ is set otherwise. M is the set of all m_{ij} with $i, j \in U$;
- a continuous number $\bar{e}_{ij}$ that describes the unenjoyability of two users traveling together, $\bar{e}_{ij} \in [0, 1]$ where 1 indicates that users i and j are not prone to travel together and 0 indicates that users i and j are prone to travel together. $\bar{e}_{ii} = 1$ so as to indicate that a user will not enjoy traveling alone.

The optimization model finds the minimum, over x_{ij} , of the following objective function:

$$\alpha\rho \sum_{i \in D} x_{ii} + \beta \left(\sum_{(i,j) \in C} \bar{e}_{ij} \cdot x_{ij} + \sum_{i \in D} \sum_{(i,k) \in C, j \neq k} \bar{e}_{jk} \cdot y_{jki} \right) \quad (3.1)$$

where the parameter ρ is the cost of adding a new car to the system. The purpose of ρ is to balance the two sides of the optimization function. The optimization is subject to:

$$\sum_{j \in P} x_{ij} \leq \gamma, \forall i \in D \quad (3.2)$$

where the maximum number of passengers per car is set to γ .

$$\sum_{i \in D} x_{ij} = 1, \forall j \in P \quad (3.3)$$

where one driver has to be assigned only to one car.

$$m_{ij} \cdot x_{ij} \leq 1, \forall (i, j) \in C \quad (3.4)$$

a limit different than 1, within $[0, 1]$ may be taken instead, to restrict the set of recommendations to take into account. For the sake of broader optimization, all of them have been taken.

$$y_{jki} \leq x_{ij} \quad (3.5)$$

$$y_{jki} \leq x_{ik} \quad (3.6)$$

$$y_{jki} \geq x_{ij} + x_{ik} - 1 \quad (3.7)$$

$$\forall i \in D, j \in P, k \in P : (i, j) \in C, (i, k) \in C, j \neq k$$

that are used to linearize the relation $y_{jki} = x_{ij} \cdot x_{ik}$.

Here, α and β are weights respectively for the first term, i.e. number of cars, and the second term, i.e. total unenjoyability in the system (the lower the better). The data-driven method to compute α , β and ρ is explained in next Section 3.3.2.

The algorithm proposed aims at minimizing the number of cars jointly with maximizing the enjoyability of the system (formulated as minimization of the unenjoyability for convenience here). Its result is to group passengers and at the same time ensure that they will enjoy the ride in each car.

3.3.2 Data Source and Methodology

Twitter was used as data source here, though the method is easily modifiable to other online sources like Flickr, or Facebook, and it can be made more general to other offline data sources as well, to supply information regarding mobility, interest, and social connections.

The full GRAAL methodology (as well as some baselines) is presented, to derive an optimal assignment starting from Twitter data, with a few parameters.

3.3.2.1 Assumptions

Twitter may be not the perfect source of data for any of the three need dimensions (text, trajectory, and co-presence). However, it is among the few public ones providing some information in all of them. The arising problems of not having ideal data were tackled as follows:

- Co-presence** the co-presence of two users in a cell at the same time is estimated, and thus the mobility demand of users, by aggregating several days of data;
- Trajectory** as geo-tagged tweets are too sparse to track users between origin/destination pairs, every user is assumed to follow the best path between them, which is computed by running the same journey planner for every pair of user locations;
- Topic mining** tweets are short, and the typical usage of Twitter include typos, abbreviations and slang. However, topic extraction via *Latent Dirichlet Allocation* [15] is typical on documents, and shown to be providing usable results also from Twitter [156].

Moreover, by having divided the space into rectangular cells through a geo-hash function, the geo-match is performed on the center of these cells (v) having divided the space into rectangular cells (thus walking distance and time are computed between their centers).

3.3.2.2 Algorithm

Algorithm 4 shows the steps performed by the methodology to solve the socially-optimal carpooling problem. The algorithm takes five parameters, and Section 3.3.5 explains how to tune the last two: i) the bounding box where to perform carpooling, ii) a spatial threshold δ , iii) a temporal threshold τ to define the time-stamped locations, and to compute trajectory containment, iv) α , to balance enjoyability and number of cars, and v) ρ , the cost of adding a car to the system. Lines 1 – 3 are used to get a geo-tagged corpus of tweets from the bounding box, to derive a set of users from it, and to filter those users with poor data. Namely, users with an average tweet per day ratio below a certain threshold (see 3.3.5.1 for details), and with a ratio between average number of distinct words and number of tweets below 1, are removed.

This last step aims at removing automated tweets, and spammers. In lines 4 – 10, for each user, his/her tweets (not necessarily geo-located) to build a larger corpus (geo-located tweets constitute a small fraction of the entire set of tweets), cleaned by removing stopwords and performing stemming, are retrieved. Then the users' friends list, i.e. the other users that the user is following, are retrieved.

In line 7 the vector of most visited (systematic) time-stamped locations of a user, given δ and τ , is computed. In particular, a spatial grid over the *boundingbox*, consisting in rectangular cells of width δ , is defined, while time is sliced in non overlapping slots of duration τ . From this set, in line 8 a journey planner is queried to derive trajectories connecting any two time-stamped locations in each users' L_u .

Finally, in line 11, the vector of topics contained in the users' documents is computed. This is done by running a *Hierarchical Dirichlet Process (HDP)* [127] on the users' tweets texts. *HDP* is a parameter-free version of *Latent Dirichlet Allocation (LDA)* [15] that automatically infers the number of topics. Lines 12 – 13 compute the like-mindedness between any two users, and then, for each user, use the median value of it to compute the homophily in lines 14 – 15. Lines 16 – 18, compute the

Algorithm 4 GRAAL ($boundingbox, \delta, \tau, \alpha, \rho$)

```

1:  $\mathcal{G} \leftarrow getTweets(boundingBox)$ 
2:  $U \leftarrow getUsers(\mathcal{G})$ 
3:  $U \leftarrow filterUsers(U)$ 
4: for  $i \in U$  do
5:    $D_i \leftarrow getTweets(i)$ 
6:    $F_i \leftarrow getFriends(i)$ 
7:    $L_i \leftarrow computeTimeStampedLocations(D_i, \delta, \tau)$ 
8:    $T_i \leftarrow computeTrajectories(L_i)$ 
9:    $\mathcal{T}_U \leftarrow \mathcal{T} \cup T_i$ 
10:   $\mathcal{D}_U \leftarrow \mathcal{D} \cup D_i$ 
11:   $\{\mathbf{t}_i\} \leftarrow computeTopics(\mathcal{D}_U)$ 
12:  for  $i, j \in U$  do
13:     $lm_{i,j} \leftarrow computeLikemindness(\mathbf{t}_i, \mathbf{t}_j)$ 
14:  for  $i \in U$  do
15:     $h_i \leftarrow computeHomophily(i, F_i)$ 
16:  for  $i, j \in U$  do
17:     $e_{i,j} \leftarrow computeEnj(lm_{i,j}, h_i, h_j)$ 
18:     $E \leftarrow E \cup e_{i,j}$ 
19:  for  $tr_i, tr_j \in \mathcal{T}$  do
20:    if  $tr_i \sqsubseteq_{\delta, \tau} tr_j$  then
21:       $R_U \leftarrow R_U \cup (i, j, tr_i, tr_j)$ 
22:   $M \leftarrow computeMobilityMatches(R_U)$ 
23:   $A_{R_U} \leftarrow optimize(\alpha, \rho, E, M)$ 
24: return  $A_{R_U}$ 

```

enjoyability values between any two users. Lines 19 – 21 generate the recommendations from the set of mobility demands. Line 22, builds the matrix of mobility matches from the recommendations.

Finally, the multiobjective optimization in line 23 is performed by finding a set of assignments minimizing the objective function described in Section 3.3.1.

3.3.2.3 Baselines

GRAAL is compared to a number of baselines (i.e., a random approach, a heuristic approach maximizing the enjoyability, against GRAAL used with two particular values of α , and against an approach based on the same rationale behind [39], maximizing the number of friends in a car), described here informally.

All the baselines start from a set of recommendations R_U computed as described in this Section. Then, they each return a (potentially different) subset of it, together with the recommendations on the single occupancy vehicles that constitute different sets of assignments A_{R_U} . To describe the first two baselines, consider the set of recommendations R_U as a directed graph G_{R_U} built by putting a directed edge (i, j) if j can get a ride from i . Then:

Random the edges of G_{R_U} are ranked randomly, then the first edge (i, j) in the rank is taken, and, if i has not been already selected as a passenger and there are

less than $\gamma = 4$ assignments (see Sec. 3.3.1) with i as driver, then with probability 0.5, all the edges linked to j are removed and the assignment (i, j) is produced. Otherwise, the next edge is taken, and the procedure is repeated for all subsequent edges in the ranking. If, at the end of the procedure, there are nodes (passengers) for which no final recommendation was made, they become drivers of single-occupancy vehicles.

Heuristic the enjoyability is maximized with a greedy approach. The procedure follows the *random* approach with the only difference that the edges of G_{RU} are ranked by descending enjoyability e_{ij} .

Social this is basically GRAAL with $\alpha = 0$, i.e. the optimization is run only for total enjoyability (which is maximized).

Green this is basically GRAAL with $\alpha = 1$, i.e. the optimization is run only for total number of used cars (which is minimized).

Lastly, the comparison with [39] is made in terms of user impact, in Section 3.3.5. Same Section presents results for: i) GRAAL ran varying α from 0 to 1 with 0.05 increments (this thus include the Social and Green baselines); ii) GRAAL ran with a particular value for α as learned as described in Section 3.3.4.1; iii) all the baselines as described above. The experiments were conducted by combining two possible values for δ (500m and 70m), and two possible values for τ (60min and 30min).

3.3.3 Mobility Analysis

An important step when considering ride sharing is the analysis of the users' mobility demand. To avoid lack of generality, a *location* l as any geo-referenced shape is defined.

Depending on the available data, a location may be a pair of (lat, lon) GPS coordinates, a geo-hashed area belonging to a geo-index, or any shape with associated geographical information used in a GIS system. In the experiments, the areas of interest are divided in rectangular cells of either 500m or 70m of width. Each user i may be found in different *locations* l over time. *Time-stamped location* is defined as a pair $tsl = (l, ts)$ where l is a location and ts is an associated relative time-stamp. The condition to consider two time-stamped locations close in space and time is so defined :

Definition 3.1 (Close time-stamped locations). Given two time-stamped locations $tsl_1 = (l_1, ts_1)$ and $tsl_2 = (l_2, ts_2)$, tsl_1 is close to tsl_2 ($tsl_1 \simeq_{\delta, \tau} tsl_2$) iff

$$\text{space-dist}(l_1, l_2) \leq \delta \wedge \text{time-dist}(ts_1, ts_2) \leq \tau$$

where $\text{space-dist}(\cdot, \cdot)$ and $\text{time-dist}(\cdot, \cdot)$ are two functions of spatial and temporal distance.

The choice of the specific functions is left for the specific application. Examples for space distance include the euclidean, spherical, or Manhattan, and for time dis-

tance simply the time difference. In this work, the spherical distance between two rectangular cells of the grid defined above, and the time difference, are used. If ts_1 or ts_2 are undefined, then the $\simeq$ operator considers only $\text{space-dist}(l_1, l_2) \leq \delta$.

A *trajectory* is a sequence $tr = \{tsl_1, \dots, tsl_n\}$ of time-stamped locations. Each user i has a set of trajectories, which constitutes the *mobility demand* $T_i = \{tr\}$ for that user. $\mathcal{T}_U = \{T_i\}$ indicates the mobility demand of all the users.

For the sake of ride sharing, a match between two trajectories, i.e. the trajectory of the user who will be the candidate driver, and one who will be the candidate passenger, has to be defined.

A match is forced on the two initial time-stamped locations of the two trajectories, and on the final time-stamped location of the candidate passenger's trajectory with any of the locations on the candidate driver's trajectory, including (where possible) the final time-stamped one. In carpooling terms, this means that the driver-passenger pair should depart from their initial locations (the first on their trajectories), but the driver is allowed to drop the passenger on any of the locations along the associated trajectory which are close to.

More formally, the following condition is defined.

Definition 3.2 (Trajectory containment). Given two trajectories $tr' = \{tsl'_1, \dots, tsl'_n\}$ and $tr'' = \{tsl''_1, \dots, tsl''_m\}$, tr' contains tr'' ($tr' \sqsubseteq_{\delta, \tau} tr''$) iff

$$tsl'_1 \simeq_{\delta, \tau} tsl''_1 \wedge \exists n, 1 < n \leq \bar{n} \text{ s.t. } tsl'_n \simeq_{\delta, \tau} tsl''_m$$

Note that this definition can be extended to any origin and destination, if required by the final application. In practice the maximum walking distance from the passenger's departure/arrival locations to pick-up/drop-off points (set by the driver) is fixed as δ , and the maximum time difference in departure and arrival times as τ . Given the above definition, two users i and j having trajectories tr_i and tr_j in their mobility demand, respectively, generate a recommendation for carpooling if tr_i is contained in tr_j or viceversa. More formally, a *recommendation* is defined as follows.

Definition 3.3 (Recommendations). Given a set of users U , R_U is the set of *recommendations* w.r.t the users in U . $R_U = \{r_{ij}\}$ where $i, j \in U$ are users and $r_{ij} = (i, j, tr_i, tr_j)$ denoting that *passenger* j is recommended to *driver* i because

$$\exists tr_j \in T_j \wedge tr_i \in T_i \text{ s.t. } tr_i \sqsubseteq_{\delta, \tau} tr_j$$

where T_i and T_j are the mobility demands of i and j , and j is the *passenger* and i is the *driver*.

By Definition 3.2, a passenger has to walk no more than δ , and wait no more than τ .

All the recommendations in a set R_U , containing all the possible recommendations between any pair of users in U , can be grouped. Following the recommendations in R_U , D is the set of possible drivers, and P is the set of possible passengers. For each recommendation the variable m_{ij} is computed, as the sum of the walking distances for pick-up and drop-off point and then normalized by the maximum.

This is referred to as *normalized distance between trajectories*. Note that m_{ij} exists within the interval $[0, 1]$ only if a recommendation between i and j exists, i.e. the passenger does not walk more than δ and wait more than τ .

Given all the definitions above, the objective of the optimization method is to find a set A_{R_U} of *assignments* containing a subset of recommendations of R_U , such that the following constraints are satisfied: i) no user is both passenger and driver; ii) each vehicle holds no more than γ passengers; iii) each user can be found in only one vehicle; iv) the total number of cars required to satisfy $\mathcal{T}_U$ is minimized; v) the total enjoyability of the system is maximized.

3.3.4 Social Analysis

Enjoyability takes into account not only whether two users share the same interest, but also whether they tend to connect to people with similar or dissimilar interest.

Let U be a set of users. Every user $i \in U$ may consider other users in U as friends, or interesting in general, and such set of users is denoted as F_i .

Each user i generates, or is interested in, a set of documents D_i . Given i and D_i , a vector of topics $\mathbf{t}_i$ can be built, where each topic is weighted by its relative importance, i.e. frequency, within the documents. This weights are computed by running a *Hierarchical Dirichlet Process (HDP)* [127] on the users' tweets texts. *HDP* is a parameter-free version of *Latent Dirichlet Allocation (LDA)* [15] that automatically infers the number of topics.

Like-mindedness is a measure of how much two users are interested in the same topics, as follows.

Definition 3.4 (Like-mindedness). Given two users i, j their *like-mindedness* is the number:

$$lm_{ij} = 2 \frac{\mathbf{t}_i \cdot \mathbf{t}_j}{\|\mathbf{t}_i\| \|\mathbf{t}_j\|} - 1$$

i and j are like-minded, i.e. they share a set of interests, if $lm_{ij} \approx 1$, not-like-minded if $lm_{ij} \approx -1$.

Two different categories of people are taken into account: those who are more prone to be in contact with other people with similar interests (*homophilous* people), and those who tend to connect with people with dissimilar interest (*heterophilous* people). For this reason user's tendency to connect with people with whom he/she has a high or low like-mindedness has been evaluated. In social networks, the concept of homophily is well known [102].

Definition 3.5 (Homophily). Given a user i his/her *homophily* is computed as the median of the like-mindedness between i and other users in F_i :

$$h_i = \text{median}_{j \in F_i} lm_{ij}$$

If $h_i \approx 1$, i tends to be homophilous, while if $h_i \approx -1$ i tends to be heterophilous.

The objective here is to relate the like-mindedness of couples of users with the homophily/heterophily of the single user. Thus, enjoyability is defined as:

Definition 3.6 (Enjoyability). Given two users i, j , their like-mindedness lm_{ij} and their homophily values h_i, h_j , the *enjoyability* of them being together is:

$$e_{ij} = \frac{lm_{ij}h_i + lm_{ij}h_j}{2}$$

E represents the set of enjoyabilities computed between each pair of users. Note that $e_{ij} \approx 1$ if either: i) both i and j are homophilous and like-minded; or ii) i and j are heterophilous and not like-minded. In the other cases, $e_{ij} \approx -1$. This means that the enjoyability will be maximum in two cases: i) if a user tends to connect with similar users and it is actually connected with a user that shares its same interests, ii) if a user tends to cluster with different users and it is connected with a user that shares different interests.

The objective function of the multi-objective formulation is a linear combination of two components: number of cars and total enjoyability. As the number of cars is minimized, the *unenjoyability* of the system is taken into account, rather than the enjoyability, to minimize this as well. The unenjoyability is computed as:

$$\bar{e}_{ij} = 1 - \frac{1}{2}(e_{ij} + 1)$$

In defining enjoyability, not only the set of possible common interests of two users has been taken into account, but also the tendency of each single user to associate with similar or dissimilar users has been considered.

Let's define the set of possible users as U , where every user $u \in U$ may consider other users in U as friends, and denote such set of users as F_u .

Each user u generates, or is interested in, a set of documents D_u . Given u and D_u , a vector of topics $\mathbf{t}_u$ is built, where each topic is weighted by its relative importance, i.e. frequency, within the documents.

First of all, *like-mindedness*, as the measure describing how likely a couple of users shares the same interests has been defined.

Definition 3.7 (Like-mindedness). Given two users u, v , their *like-mindedness* is the number:

$$lm_{u,v} = 2 \frac{\mathbf{t}_u \cdot \mathbf{t}_v}{\|\mathbf{t}_u\| \|\mathbf{t}_v\|} - 1$$

The like-mindedness quantifies the similarity of interests between any two users.

u and v are like-minded, i.e. they share some or all their interests, if $lm_{u,v} \approx 1$, not-like-minded if $lm_{u,v} \approx -1$.

Moreover, two different categories of people have been taken into account: people that are more prone to be in contact with other people with similar interests, and people who tend to connect with different people. For this reason, user's tendency to connect with people with whom he/she has a high or low like-mindedness has

been evaluated. The homophily is typically defined as the tendency of individuals to associate with similar others. On the other hand the heterophily is defined as the tendency of individuals to collect in diverse groups. The objective is to relate the like-mindedness of couple of users with the homophily/heterophily of the single user.

Moreover, user's tendency to connect to people is evaluated with whom (s)he has a high like-mindedness, or a low one.

Definition 3.8 (Homophily). The *homophily* of a user u is the median of the like-mindedness between him/her and all him/her friends F_u :

$$h_u = \text{median}_{v \in F_u} lm_{u,v}$$

where median returns the median value of a certain set of values.

In social networks, the concept of homophily is well known [102], and network analysts often compute it on the degree of a node, defining the assortativity of a network the phenomenon for which most of the users tend to be homophilous by degree (i.e. they link to nodes with similar degree). If $h_u \approx 1$, u is *homophilous*, while if $h_u \approx -1$, u is *heterophilous*.

Finally, the enjoyability can be defined.

Definition 3.9 (Enjoyability). Given two users u, v , their like-mindedness $lm_{u,v}$ and their homophily values h_u, h_v , the *enjoyability* of them being together is:

$$e_{u,v} = \frac{lm_{u,v}h_u + lm_{u,v}h_v}{2}$$

Note that $e_{u,v} \approx 1$ if either both u and v are homophilous and like-minded, or u and v are heterophilous and not like-minded. In the opposite case, $e_{u,v} \approx -1$. This means that the enjoyability will be maximum in two cases: 1. if user u tends to connect with similar users and u and v share the same interests, 2. if user u tends to cluster with diverse users and u and v do not share the same interests.

The enjoyability in each vehicle is defined as:

Definition 3.10 (Vehicle Enjoyability). Given a vehicle V holding at least two users, $V \subseteq U$ s.t $|V| > 1$, its enjoyability is defined as the average enjoyability computed between any two occupants, or, in formula:

$$e_V = \text{avg}_{u,v \in V} e_{u,v}$$

If the vehicle has no passengers (i.e., there is only the driver), the enjoyability of it is set to 0.

3.3.4.1 Online Survey

In order to assess the real importance of the enjoyability in carpooling versus other factors like sustainable mobility, a survey with potential end-users of the proposed

methodology has been conducted. This section presents the design of the study, as well as the numerical results obtained with it.

The survey was sent via direct Twitter messages, other social networks (e.g. Facebook, LinkedIn, etc., including dedicated carpooling groups), and direct e-mail or mailing list interactions. A link to a webpage containing the survey has been shared, which was organized as follows. Users have been directed to a webpage, depicted in Figure 3.11, presenting the two resulting cars from the Green and Social baselines, in the optimizations obtained with $\delta = 500m$ and $\tau = 60min$.

The Green model was chosen as it constitutes the main baseline for carpooling, and it is well understood by people. Then, Social was used, as counterpart, as the enjoyable output, as it represent the opposite extreme of the multiobjective optimization. The goal of this user study was to learn a crowd-sourced value for their α weight.

To generate a landing page, a user $i \in U$ in the dataset was picked up, the cars he/she would end up into in both the approaches were identified, and presented in the same page, to the user of the survey. The page shows then the corresponding trip between two points in either Rome or San Francisco. For each trip, the two solutions (i.e. the cars) were presented. Each solution showed:

- i) a bar indicating the average enjoyability among the occupants of the car;
- ii) a bar indicating the “greenness” of the solution, computed as the global amount of cars saved by the city-wide system if all the users were to click on this choice;
- iii) a button to choose that particular car.



Fig. 3.11 Part of the landing web page of the survey

The two cars were presented in random order, to minimize the probability of clicks performed on a given column. The two presented solutions can be referred to as “social choice”, and “green choice”. The first one is the car with higher enjoyability but lower greenness value (obtained by Social), while the second choice is the car with lower enjoyability but higher greenness value (obtained by Green).

Note that, while the enjoyability is a property local to its car, the greenness is a global, city-wide, property. That is, there are only two values of greenness for a city: the one obtained if every user were to click on the social choice, and the one obtained if every user were to click on the green one.

After this choice, the users were directed to a subsequent set of general questions on carpooling, including the following: “which of the following would make carpooling more attractive to you? Savings, sharing the car with interesting people, or sustainability of the solution?”

The collected answers and values could be exploited to weigh the multi-objective optimization model, by combining the costs of the two starting carpooling solutions. Thus, the two parameters α and β were learned, which represent how much the users are more likely to prefer the Social car w.r.t the Green one, and vice-versa.

Each page presents two cars, with their enjoyability values of e_S (the enjoyability of the Social car) and e_G (the enjoyability of the Green car). If their difference $|e_S - e_G|$ is high, meaning that the social car has a high value of enjoyability, while the green car has a low value for it, the user is expected to be tempted to click on the social car, rather than the green one.

As the greenness values of the Social and Green car for a given city are fixed (i.e., they do not change if a different pair of solutions is displayed), they are not taken into account in the learned weight. Instead, the difference of enjoyabilities between the green and the social car, which depends on the pair of solutions displayed, is considered. The following two values have been defined. For the Green car, the value v_G is defined as:

$$v_G = |e_S - e_G|$$

For the social car (i.e., the *enjoyable* one), the value v_S is computed by taking into account the complement to 1 of the absolute value of the difference between the average enjoyabilities:

$$v_S = 1 - |e_S - e_G|$$

Given S as the set of the social choices obtained from the survey, G the set of the green choices, and the v_S and v_G values computed on their elements, the weight α is defined as the ratio:

$$\alpha = \frac{\sum_{j=1}^{|G|} v_{G_j}}{\sum_{i=1}^{|S|} v_{S_i} + \sum_{j=1}^{|G|} v_{G_j}}$$

In the same way, β can be defined as the complement to 1 of α .

Intuitively, α and β are sums, over all their respective choice sets, weighted by the likelihood of making a particular choice against the other.

Concerning the survey results, from the 237 answers collected, the 39% was in favor of a social solution. Another encouraging result was provided by the answers to the additional survey question: 24% of the people was more attracted by sharing the car with interesting people, while 41% by the savings provided by carpooling, and 35% considered the sustainability to be the most attractive aspects of ride sharing. These numbers can be definitively considered as a measure of the potential impact of a ride sharing system taking into account *also* the enjoyability of a car, rather than just minimizing the cars.

3.3.5 Experiments



Fig. 3.12 Geographical areas analyzed.

This Section shows the results of running GRAAL and the baselines on real Twitter data, using different sets of parameters. In details, it presents the data and tools used, the parameter tuning (this includes the results from the user study), the results of computing the social measures, and the results of GRAAL and all the baselines. The results of optimization were assessed from a city-wide perspective (i.e. looking at the total values of the components of the objective function), and from a user perspective (i.e. looking at the distribution of the enjoyability of single cars, and user impact).

3.3.5.1 Data

The Twitter's Streaming API³ was exploited to obtain two large datasets of geo-tagged tweets. The API was queried using two bounding boxes on the area of *Rome* (Fig. 3.12(a)), and the bay of *San Francisco*, hereafter referred to as San Francisco (Fig. 3.12(b))⁴, for 50 days from the beginning of October 2014. As a result, 558,000 geo-tagged tweets are collected from 17,600 users in Rome, and 3,286,000 geo-tagged tweets from 113,000 users in San Francisco. Rome was chosen as it is the city with the largest population in Italy, while San Francisco was chosen for its popularity in ride sharing studies [9, 37, 46].

The application of user and tweet filtering ended up with the statistics reported in Table 3.1. In particular, the objective was to consider the top 1,000 users for each city. The users in Rome having less than 40 tweets, and users in San Francisco having less than 300 tweets, were filtered out. This resulted in 1,106 users in Rome, and 1,052 users in San Francisco.

Dataset	Rome	San Francisco
Users	1,106 (06.53%)	1,052 (00.93%)
Tweets	237,351 (42.19%)	681,597 (20.74%)

Table 3.1 Users and tweets statistics after filtering. Numbers in brackets are the percentages over the initial unfiltered data.

3.3.5.2 Tools

GRAAL was written in Java and C, making use of external libraries for some specific tasks. A publicly available Java implementation of HDP⁵ was used to perform non-parametric topic modeling. To execute route planning OpenRouteService⁶ was used, a public Java library. As *space-dist* and *time-dist* the geo-spherical distance and the absolute difference, respectively, were used. Finally, to perform the optimization steps, the C APIs of IBM CPLEX⁷ was used.

³ <https://dev.twitter.com/docs/streaming-apis>

⁴ GPS coordinates bounding box: Rome (12.234498, 41.655642, 12.85576, 42.141028), San Francisco (-122.667, 36.8378, -121.2949, 38.0771)

⁵ <https://github.com/arnim/HDP>

⁶ <http://openrouteservice.org/>

⁷ <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>

3.3.5.3 Parameters

To run GRAAL on the collected data, besides the parameters of Algorithm 1, a sample of the data (in number of days) and a number of topics to put in the topic vectors were chosen. The bounding box, and the spatio-temporal parameters δ and τ were kept as choices available to the analyst, for different analyses. This allows to try different optimizations in function of different temporal and spatial resolutions.

In the experiments, results for δ set to 500 and 70 meters, and for τ set to 30 or 60 minutes, were reported. Note that the combination $\delta = 500m$ and $\tau = 30min$ agrees with common sense, or best practice, in journey planning: users are typically willing to walk distances up to 500 meters, and have a flexibility of waiting up to 30 minutes to find a means of transport.

The 3 most frequent locations were considered, as they typically cover home, work, and the so called “third place”. Home and work place detection were out of the scope for this paper. To decide the number of days of data to take, the ratio of people for which at least one of the top 3 locations changes on x consecutive days was considered. After $x = 40$ it dropped. Thus 40 consecutive days of data were taken in the sample.

Next, a description of how a suitable number of topics was learned, α and β , and ρ , follows.

3.3.5.4 Number of Topics

Topic modeling is recognized to be able to capture meaningful information in textual data [15, 33]. However, one drawback of one of the main approaches to topic modelling, i.e. Latent Dirichlet Allocation (LDA), is being parametric in the number of topics to extract. As stated above, a nonparametric HDP algorithm was used to estimate the number of topics automatically. Since HDP is nondeterministic, it was ran 2,000 times on the data, obtaining on average 25.48 topics ($\sigma = 1.56$) on Rome and 25.61 ($\sigma = 1.54$) on San Francisco. The results for the two datasets are reported in Figure 3.13. According to this, results relative to a number of topics of 25 were selected to construct vectors $\mathbf{t}_i$.

3.3.5.5 Tuning ρ

The ρ parameter is defined as the cost of adding a car to the system within the optimization model. By varying ρ from 0 to 100, it was observed that $\rho = 2$ had the largest impact on the number of cars saved.

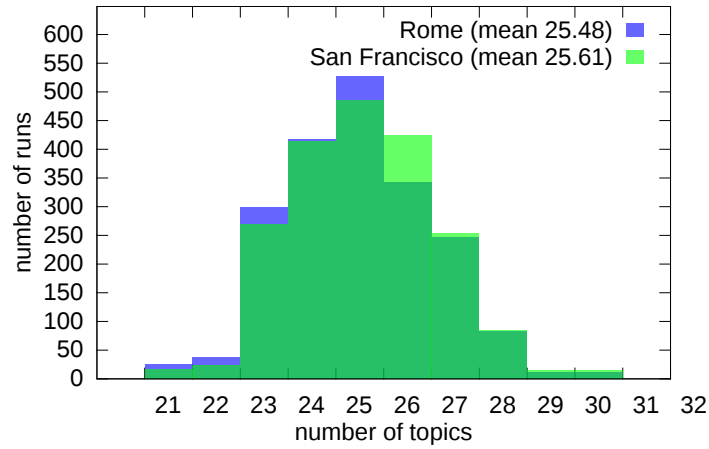


Fig. 3.13 Number of topics distributions on 2,000 runs of HDP.

3.3.5.6 Tuning α

The α , and thus β , parameters were learned looking at the results of the user study conducted as described in Section 3.3.4.1. 237 responses were collected from three different sources: 2% came from direct messages sent via Twitter; 12% came from sharing the survey in other social networks; 86% came from direct e-mail or mailing lists sharing. In total, 39% of people clicked on the social choice. This is encouraging, as it confirms the need for a social-aware carpooling system. This result was studied further.

Figure 3.14 reports the percentages of clicks on the social or green choice (y axis) versus the difference in enjoyability of the two proposed cars $|e_S - e_G|$ (x axis). No clear correlation between the two quantities was reported, meaning the users made an educated choice, not conditioned by the above difference. As the distribution is also non-uniform, also a non-random choice between Social and green Cars was reported. The final value obtained for α , computed as explained earlier is 0.36 ($\beta = 0.64$).

3.3.5.7 Results on Social Measures

Figure 3.15 presents the distributions of like-mindedness (top left), homophily (top right) and enjoyability between pairs of users (bottom row) for all the users. There was no significant differences in like-mindedness and homophily between Rome and San Francisco. As depicted from the first plot, computing a similarity based only on the like-mindedness may end up recommending connections in a limited number of pairs of users. On the other hand, the second plot depicts that most of the people are heterophilous. If the two things are combined into the enjoyability, as depicted

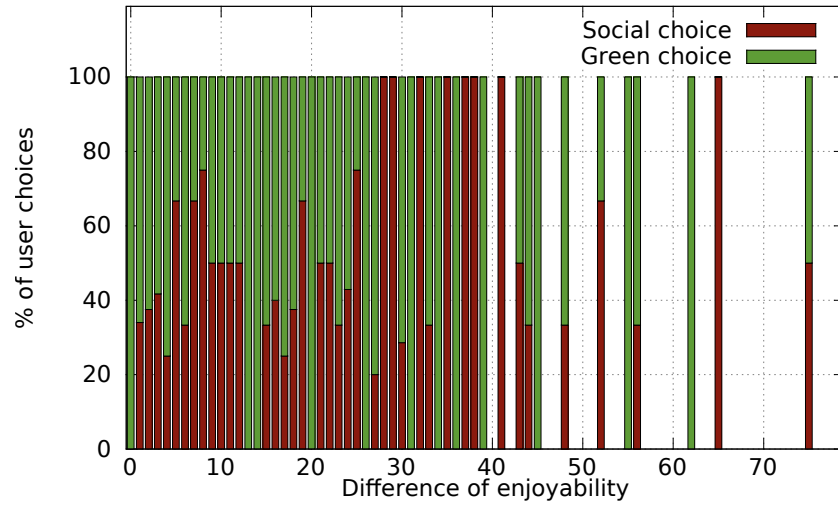


Fig. 3.14 Distribution of the user choices by enjoyability difference

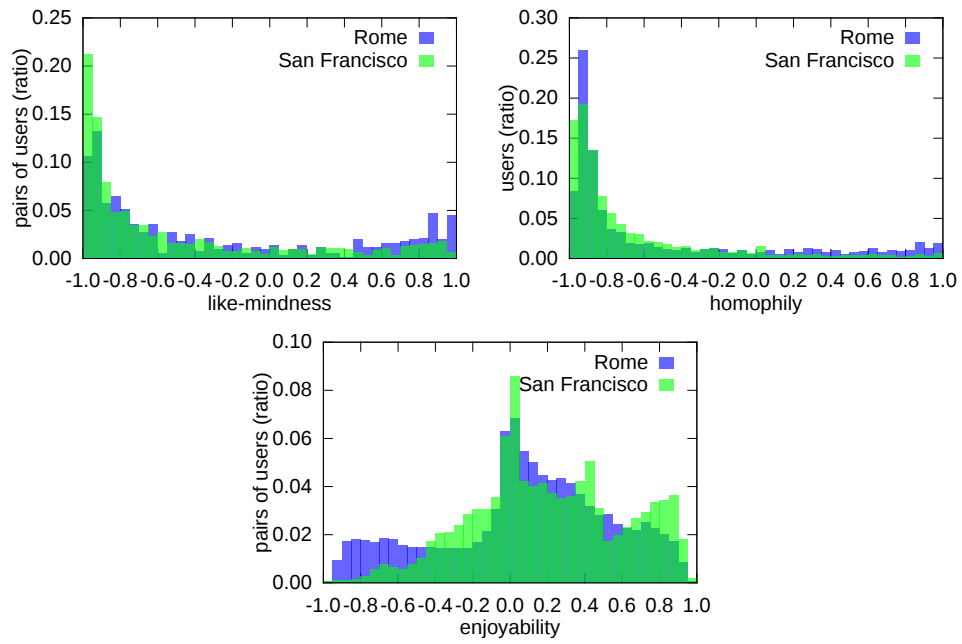


Fig. 3.15 Social measures for all the couples of users.

in the third plot, there is broader space for recommendations based on this measure, rather than the like-mindedness. Moreover, the combination of the first two measures produces two different distributions for Rome and San Francisco, highlighting that the enjoyability is capturing a different phenomenon than just the like-mindedness.

3.3.5.8 Results on Recommendations

δ	τ	Rome				San Francisco			
		$ R_U $	$ S_1 $	$ S_2 $	$ Z $	$ R_U $	$ S_1 $	$ S_2 $	$ Z $
500	60	6,883	81.56%	76.04%	18.44%	2,298	68.63%	57.41%	31.37%
500	30	5,870	79.84%	73.51%	20.16%	1,106	54.37%	36.88%	45.63%
70	60	349	26.85%	15.46%	73.15%	245	16.73%	9.60%	83.27%
70	30	309	24.68%	13.92%	75.32%	250	16.44%	9.79%	83.56%

Table 3.2 Statistics on user recommendations by δ and τ for GRAAL. $S_1 \subseteq U$ is the set of users with one or more recommendations, $S_2 \subseteq U$ is the set of users with two or more recommendations, $Z \subseteq U$ contains the users with no recommendations.

Table 3.2 reports some statistics for the obtained recommendations in the two areas using different combinations of spatio-temporal resolutions. The first column reports the number of recommendations, in column S_1 the percentage of users with one or more recommendations is shown, column S_2 reports the percentage of users with two or more recommendations, (for which the optimization has more impact), while in column Z the percentage of users with no recommendations is shown (these will end up being drivers of single occupancy vehicles in all the models). For the sake of optimization, these users are dropped, as no optimization is possible for them, in line with the literature [39].

The table shows the clear effects of taking the same number of users in the two cities having very different geographical structure. In particular, San Francisco Bay Area is a much larger area than Rome. As carpooling in San Francisco works actually across the entire area, while it would not make much sense to keep the same user density per area and reduce the area over San Francisco, no corrective actions were taken. In this way, one could also assess the effects of having different recommendation densities on the performances of the optimization. Thus, a larger room for optimization in Rome in general is reported, and for $\delta = 500m$ in general as well. In San Francisco, only $\delta = 500m$ provides significant room for optimization. This is expected to be seen in the results at the city level.

3.3.5.9 City-wide perspective

This section presents the results of the optimization at the city level, i.e. at the level of the entire optimization. The effects of varying $\alpha \in [0, 1]$ (with steps of 0.05), and

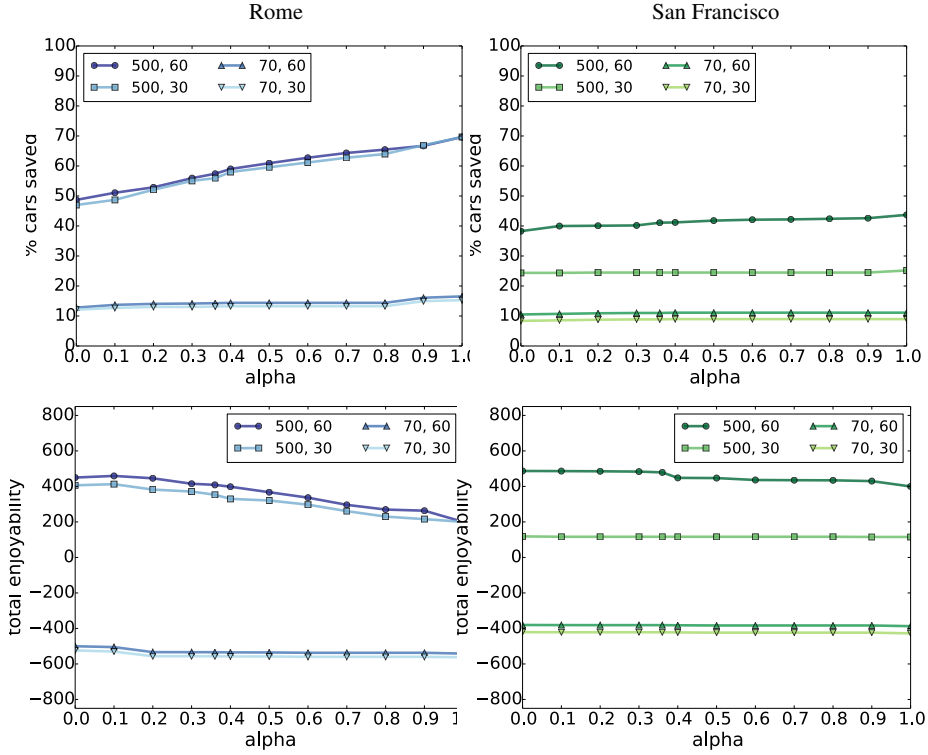


Fig. 3.16 Cars saved (top row) and total enjoyability (bottom row), in Rome (left column) and San Francisco (right column) by running GRAAL with 20 values of α and different values of δ and τ . For all the plots, higher is better.

in particular $\alpha = 0.36$, on the total number of cars saved and the total enjoyability of the system, were studied. GRAAL for $\alpha = 0.36$ was compared with all the baselines.

Figure 3.16 reports the number of cars saved (in the top row) and the total enjoyability (in the bottom row) for Rome (in the left column) and San Francisco (in the right one). The best performance are reached for the city of Rome with $\delta = 500m$ and for any values of τ . In all the other cases (and in San Francisco as well) is reported a mostly flat behavior, which means that with less room for optimization, α can not make a big difference in the results. Moreover, $\delta = 70$ implies the lowest number of cars saved and the lowest enjoyability in both cities. This agrees with the lowest numbers in the S column of Table 3.2. Moreover, where the Z column of Table 3.2 is very high, negative values of enjoyability are present. This is due to the large number of people going alone, which have an enjoyability score of -1 , as described in Section 3.3.1.

Figure 3.17 shows the percentage of cars saved (in the top row) and total enjoyability (in the bottom row), in Rome (left column) and San Francisco (right column) by running GRAAL with $\alpha = 0.36$ and all the baselines. As expected, the highest number of cars is saved by the Green approach. One encouraging result is that So-

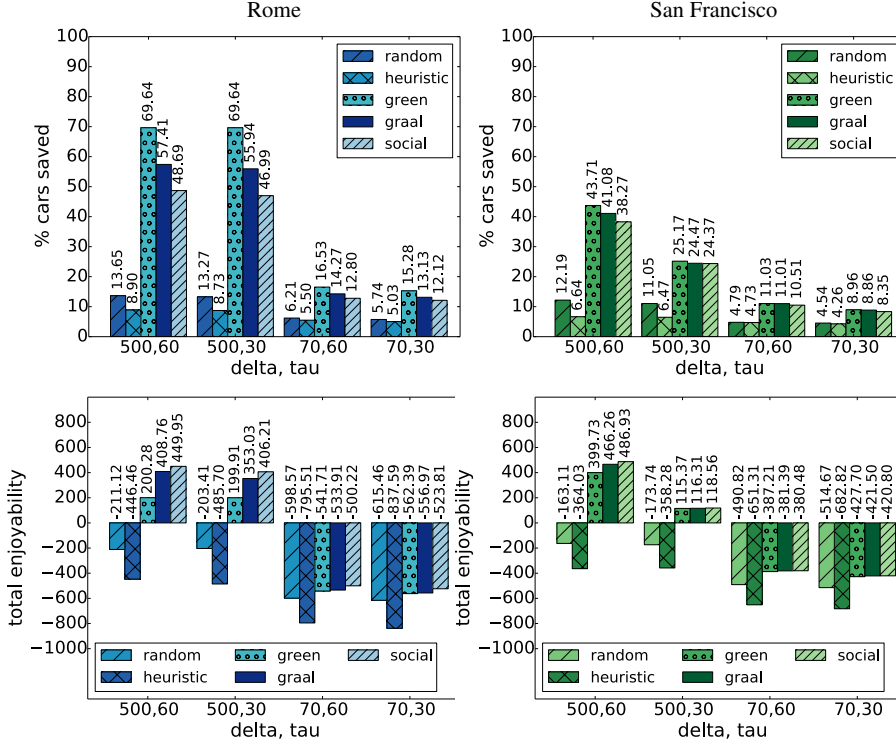


Fig. 3.17 Cars saved (top row) and total enjoyability (bottom row), in Rome (left column) and San Francisco (right column) by running GRAAL with $\alpha = 0.36$ and all the baselines. For all the plots, higher is better.

cial saves a significantly higher number of cars w.r.t Random and Heuristic. This is due to the choice of assigning -1 as enjoyability to a person traveling alone. As a consequence, even if Social does not directly minimize the number of cars, it tends to put more people together anyway. The GRAAL approach with $\alpha = 0.36$ is a trade-off between Social and Green (which are basically GRAAL with the two possible extreme values for α).

Consider now the bottom row of Figure 3.17, with $\delta = 500m$. The negative total enjoyability confirms that in those cases there is a significant number of people going alone. This is avoided by GRAAL with all alpha values (as reported in Figure 3.16). In accordance with Table 3.2, reporting a high number of single occupancy vehicles for $\delta = 70m$, there are only negative total enjoyability for all models for this value of δ .

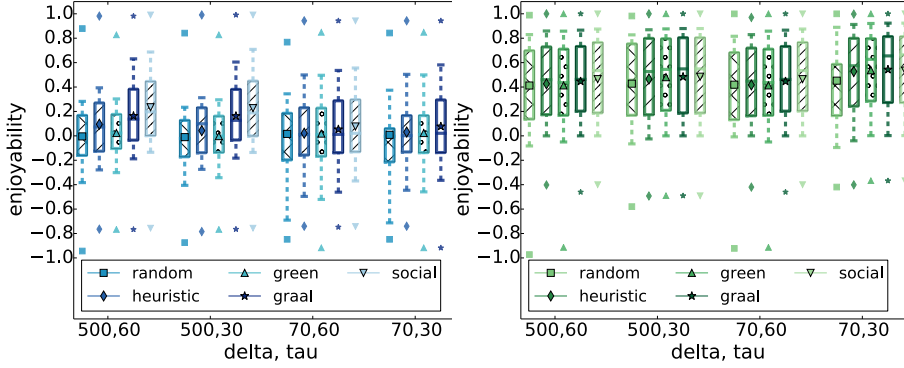


Fig. 3.18 Enjoyability per cars (min, max, 10th, 25th, 50th, 75th, 90th percentiles, and average across all cars), for Rome (left) and San Francisco (right). Higher is better.

3.3.5.10 User perspective

The results from the user perspective were assessed in terms of enjoyability in the single cars. As aggregates, minimum, maximum, average, 90th, 75th, 50th, 25th, and 10th percentiles of the distribution of the enjoyability across vehicles are reported, in order to understand the improvement introduced for a user, in Figure 3.18. For this assessment, only the users who received a recommendation were considered. That is, most of the effects of considering an enjoyability equal to -1 for a high number of people were removed in these plots. The first clear result is that there is a globally higher enjoyability in San Francisco, compared to Rome. This is coherent with the results on the distribution of the enjoyability per city reported in Figure 3.15, which shows both a higher negative tail in Rome for the enjoyability, and a higher positive tail for San Francisco.

Despite the globally higher enjoyability, there is again the problem of the results being more flat. Consider now the results in Rome, with $\delta = 500m$. In the Green model, where the optimization disregards the enjoyability, the results are inline with Random, while Heuristic does a better job. This is also true for the other value of δ , although less evident.

δ	τ	Rome				San Francisco			
		$ R_U $	$ S_1 $	$ S_2 $	$ Z $	$ R_U $	$ S_1 $	$ S_2 $	$ Z $
500	60	189	11.36%	7.42%	88.64%	148	7.57%	4.89%	92.43%
500	30	183	11.12%	7.02%	88.88%	120	7.50%	4.23%	92.50%
70	60	53	9.37%	4.40%	90.13%	46	5.82%	2.31%	94.18%
70	30	51	9.40%	4.32%	90.60%	45	4.35%	2.25%	95.65%

Table 3.3 Statistics on user recommendations by δ and τ for [39]. $S_1 \subseteq U$ is the set of users with one or more recommendations, $S_2 \subseteq U$ is the set of users with two or more recommendations, $Z \subseteq U$ contains the users with no recommendations.

Table 3.3 tries to compare GRAAL with the method described in [39], which tries to put friends (i.e., direct Twitter links) together, as their concept of enjoyability. However, the final comparison would not be fair, as their methodology uses the friendship connections to filter out the users before the optimization by cars, and not vice-versa. This would reduce dramatically the number of users in the system. Also, their goal is to optimize by friendship, while GRAAL utilizes the enjoyability to enlarge the set of users impacted by the optimization.

For these reasons, an evaluation against them in terms of impact on users is reported in Table 3.3, which contains the same columns as Table 3.2. It was produced by running Green first, and then by applying brute force to optimize by friendship (friends are put together in a car). One can clearly see that the number of recommendations between friends is much smaller than what can be achieved with GRAAL, due to the sparsity of the friendship connections in Twitter (and in the real world, too), as opposed to the fact that the enjoyability could be computed between any two users. Thus, the room for optimization here is much smaller, with numbers in S_2 not reaching two digits. The purpose of putting the S_1 column in the two tables was actually to give more chances to this approach. In fact, even if room for optimization is less, with at least one recommendation, friends can be put together. Nevertheless, numbers go up to slightly more than 11%.

A comparison with the same approach ran with a 2-hop network was not performed for a clear reason: 2-hop friends (i.e., friends of friends), when they are not direct friends, are people with whom you can not give any guarantee on the enjoyability from a topic perspective, and neither they are direct friends. On the other side, 2-hop friends could be at least more trustworthy than unknown (but enjoyable) people. However, neither the applied methodology, nor the one in [39] are meant to be seeking a higher trust in the system, which is then left as future work.

3.4 Real-world Case Study: SmartHitch

This Section presents **SmartHitch**, a real-world ride sharing service built on top of the case study presented in Section 3.2 that let deploy on the field the set of algorithms tested in laboratory. The service has been developed by *Epoca S.r.L.*, a former spin-off of University of Modena and Reggio Emilia, and it has been powered by funds from the EU *FIWARE FrontierCities* Accelerator⁸, which aims to promote, or accelerate, the use of *Future Internet* – FIWARE⁹ – technologies in order to increase the presence of smart mobility applications in European cities. SmartHitch, with the use of such as latest technologies, aims to create an innovative dynamic interaction between commuters, and citizens in general, within a city and its transportation network.

⁸ <http://www.fi-frontiercities.eu>

⁹ <http://www.fiware.org>

The service, opposed to current ride sharing services on the market, is conceived to be the less unobtrusive as possible for its users, by being grounded on the novel idea offered by the opportunistic ride sharing recommender system analyzed in Section 3.2.

SmartHitch proposes a solution for reducing congestion and pollution via collaborative mobility supported by a "smart hitch-hiking" mobile app. The core of the service is a smart engine that: (i) autonomously collects the mobility needs of people, and the mobility patterns of drivers, to learn the typical mobility routes of individuals; (ii) identifies in real-time groups of users that shares compatible routes; (iii) suggests individuals to dynamically hitch a ride by matching drivers, including suggesting a driver not to take the car and share a ride.

The main objective of SmartHitch is to develop a software infrastructure able to:

- simplify the adoption of ride sharing as an option for commuting;
- decrease the number of vehicles used for daily commuting, so also pollution and traffic;
- promote more sustainable mobility habits;
- reduce the cost of commuting for riders and drivers.

SmartHitch targets large private and public institutions, where a large number of commuters travel to a relatively low number of workplace locations and allows for efficient car pooling; moreover, the platform becomes especially relevant for directional and business districts, where facilities of more large private/public companies are located in a relatively confined area, hence opening up opportunities for finer and more efficient inter-organization optimizations.

3.4.1 FIWARE Technologies

FIWARE's mission states that the rising community, the set of tools and resources made available to developers have the aim of building an open sustainable ecosystem around public, royalty-free and implementation-driven software platform standards that will ease the development of new Smart Applications in multiple sectors.

The FIWARE catalogue contains a rich library of components (i.e., *Generic Enablers* – GE, tools, and frameworks) that allow to put into effect advanced functionalities such as the connection to the Internet of Things or Big Data analysis. In particular, technologies are designed to adapt to domain specific conditions, such as, for instance, the ones in agri-food, e-health, energy, media, manufacturing, mobility.

In the definition of the SmartHitch software architecture, four FIWARE components were included in it:

KeyRock Identity Manager GE. It covers secure and private authentication from users to devices, networks and services, authorization & trust management, user profile management, privacy-preserving disposition of personal data, and Single Sign-On.

- Cosmos** Big Data Analysis GE. It is intended to deploy means for analyzing both batch and stream data, by exploiting an *Hadoop As A Service* engine.
- Orion** Publish/Subscribe Context Broker GE. It exposes standard interfaces to perform several operations for context registration, update, and query.
- Cygnus** Context Broker Connector tool. It is in charge of persisting data coming from Orion (intercepted on the subscription/notification feature of Orion) in certain pre-configured storages, with the aim of creating a historical view of such data.

3.4.2 Service Design

SmartHitch is a ride sharing service which allows users to share their cars and offer lifts to other people who need to travel the same route. The project target are commuters, or those who have a daily need to reach their workplace in at least 20 minutes, traveling by public or private transports.

During the first phase of the project, three main challenges related to the design of a collaborative mobile platform for commuters were set:

- autonomously collects the mobility needs of people, and the mobility patterns of drivers, to learn the typical mobility routes of individuals;
- identifies in real-time groups of users that share compatible routes;
- suggests individuals to dynamically hitch a ride by matching drivers, including suggesting a driver to take the car and share a ride.

FIWARE technologies, applied to a complex software architecture and to an advanced algorithm that performs rides matching, will be exploited to solve these challenges.

The project goal is to improve the traveling experience from home to workplace and bring benefits to the end users (reduction in costs, increased comfort) and businesses (more sustainable choice, improved quality of life for their employees) and also on the whole community and environment (traffic pollution reduction).

SmartHitch should enter into the carpooling market as a different carpooling service from existing ones, also because it is oriented on a user category with very specific traits and special needs. For this reason, Epoca started with a research phase that has been carried out by a multidisciplinary team which adopted the *Design Thinking* approach¹⁰, and worked simultaneously with the technology and business team. The Design Thinking team focused and analyzed the potential competitors, identifying the user's needs and prototyping the features to be field-tested.

The field-test outlined SmartHitch basic features. One of these was related to the creation of a trusty user community. The lock-in effect could be reached by developing and supporting the community awareness. One other outlined feature was

¹⁰ A user-centered problem-solving technique used in the fields of process – product – service – system design, based on users' needs, first mentioned by Simon Herbert in his 1969 book "The Sciences of the Artificial, Cambridge, MIT Press".

to efficiently manage the users matching, which allow to put together passengers with common interests in the same car (e.g., Musical interests). Moreover, another relevant feature that should be enabled by SmartHitch, is the ability in managing unexpected situations, in order to establish a trusty system. For instance, in the case of no-show of the driver, riders need back-up solutions to the missed ride.

The final prototype introduced (i) a feedback system for the users, (ii) a matching algorithm able to take into account social affinity of users in the same car (according to the motivations that brought to the case study presented in Section 3.3), (iii) an external service of taxi booking in case the driver deletes the availability of her/his ride after at least one rider booked for it.

At the date of writing, the implementation of the service is still ongoing, thus the current deployment does not include both social aspects in matching discovering, and alternative rides for unexpected situations.

3.4.3 Revenue Model

In order of not providing a service that would get in taxi's own businesses, SmartHitch does not manage direct payments related to rides, so it does not require neither riders to pay a quota per kilometer ridden, nor drivers to pay the service provider (i.e., SmartHitch) per every ride shared. A time-based subscription is required to use the service, both for riders and drivers. The driver that has to pick up a rider, is notified with a price suggestion for sharing that ride according to the kilometers driven and the preferred tariff set within her/his preferences. However, a final negotiation can happen, and the actual monetary transaction is made by cash. In any case, SmartHitch platform does not work as a proxy for payments.

In the initial phase, and for a limited period, the service is launched for free to encourage people to download the mobile app and to boost its adoption.

In addition to the above, the business model considers licensing the use of SmartHitch to large companies (i.e., to their employees) and municipalities (i.e., to their citizens). One of the key values of SmartHitch is its fitting nature to employees of large companies located in confined areas. For instance, one could think to provide customized service plans to companies located in industrial districts to encourage the use of shared modes of transportation for employees used to commute alone in their private cars. SmartHitch considers a special subscription plan for industrial partners that would benefit of a "greener" image of them.

3.4.4 Technical Implementation

SmartHitch service platform relies on a complex software architecture that manages and coordinates the full stack of operations from user interactions on the front-end mobile application to the rides matching procedure on the back-end cloud envi-

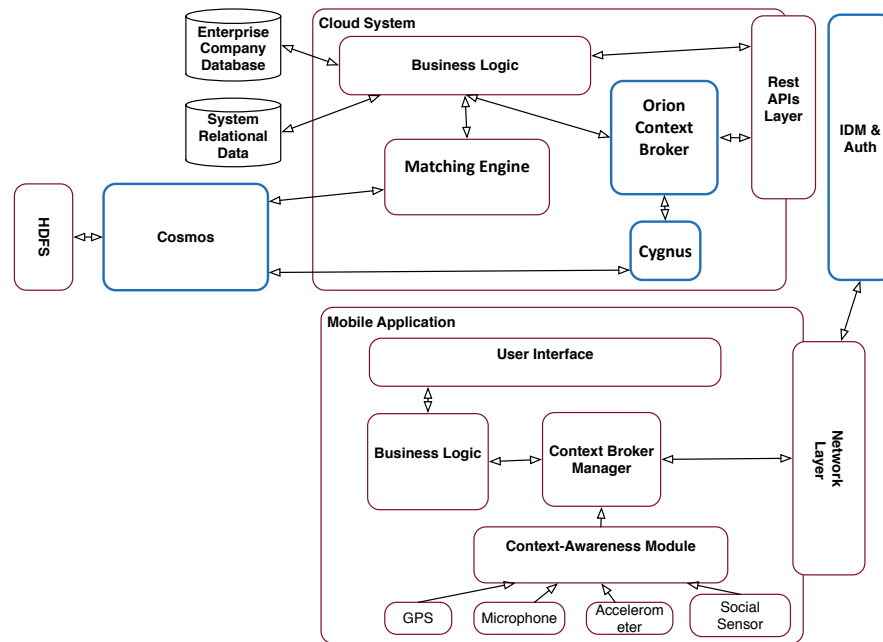


Fig. 3.19 SmartHitch software architecture diagram. The blue-bordered modules are the FIWARE components.

ronment. A diagram with the architecture components is shown in Figure 3.19. A description of the main modules follows.

3.4.4.1 Mobile Application

SmartHitch front-end is made of a mobile app developed both for iOS and Android devices that enables all the major features emerged in the design phase of the service.

Once the user logs in for the first time he/she is asked to fill a profile form that eventually includes the information about his/her owned car. After the initial setup of the user profile, recurrent itineraries can be added by specifying the departure time and location, the arrival location, the kind of recurrence (e.g., single day, weekly), the maximum acceptable spatio-temporal deviation for picking-up and dropping-off (in terms of kilometers walked/driven, and temporal difference at meeting points), and the role of the user in the trip (i.e., the user offers rides as a driver, or he/she looks for a driver which shares a ride with him/her). Figure 3.20 shows screenshots of the itinerary-adding process.

When the itinerary is added successfully, any potential match with other trips is sought. In the discovering of a matching driver-rider(s) pair of users, the rider(s) is provided with a in-app notification with the matching information. As soon he/she

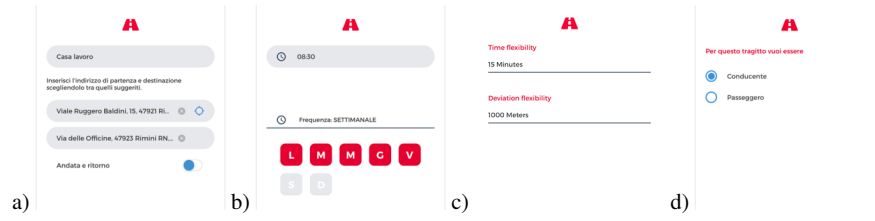


Fig. 3.20 SmartHitch itinerary-adding process consists in **a)** filling origin and destination locations, **b)** setting departure time and kind of recurrence, **c)** setting maximum spatio-temporal deviations, and **d)** choosing of being a driver or a rider of the trip.

books the ride, the driver is notified of the confirmed booking with the suggested meeting point for the pick-up and the drop-off of the rider. Figure 3.21 shows screenshots of the ride-booking process.

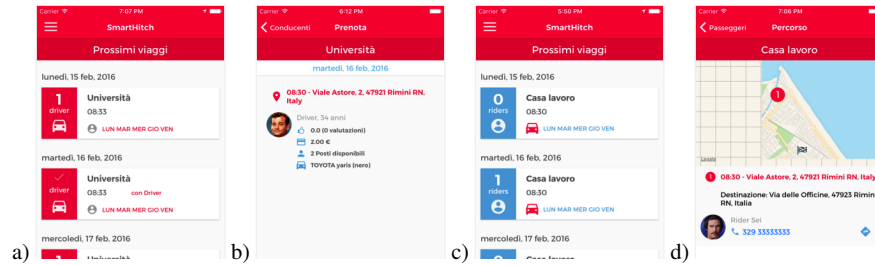


Fig. 3.21 SmartHitch ride-booking process starts from a rider that **a)** receives recommendations for any discovered driver that matches with his/her recurring trips. **b)** In the driver view, rider can evaluate matching information (i.e., meeting time and location, driver name and rating, ride suggested price, car model and seats availability) and book the ride. **c)** Then, the driver receives the notification for the booked trip. **d)** In the trip view, driver can evaluate information of the riders that have booked a ride on that trip (i.e., pick-up time and location, drop-off location, rider name and contact number).

Context Broker Manager

The Context Broker Manager deployed on the mobile app has the aim of granting the synchronization of the data between the mobile devices and the server, by keeping the context an integral part of the decisions of the synchronization modes.

The synchronization is guaranteed by pushing all the changes on the recurring trips to the cloud back-end and exploiting the Publish-subscribe pattern for updating the interested users' devices with all the updates regarding new matches, rides booking and feedbacks.

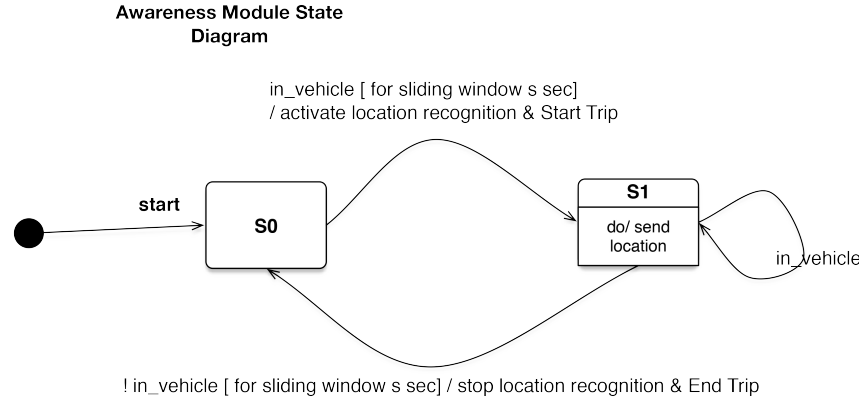


Fig. 3.22 Context-Awareness Module State Diagram.

Context-Awareness Module

The Context-Awareness Module in the mobile application continuously collects GPS traces to characterize users' mobility pattern and needs. In order to preserve battery life, the GPS traces mining will be switched on when the user is in a vehicle, by exploiting the Google Activity Recognition APIs¹¹, in Android, and the Apple Core Motion Framework¹², in iOS. The Android Fused Location Provider¹³ and the Apple Core Location Framework¹⁴ are used to retrieve user's location and tune the location accuracy to preserve battery life (e.g. lower accuracy when the user is not in a vehicle). Figure 3.22 shows the state diagram implemented for collecting users' GPS traces.

When the module detects that the user is in the vehicle for a time longer than s seconds, there is a state transition from $S0$ to $S1$. The state transition triggers the activation of the user's position tracking in order to characterize her/his route. When the user is out of the vehicle for a time longer than s seconds there is a state transition from $S1$ to $S0$. This time, the state transition triggers the deactivation of the user's position tracking and persists all the collected information as the potential recurring trip. All the potential recurring trips will be suggested to the user through the mobile application user interface to allow the user to publish them as recurring trip by adding the desired frequency or to allow the user to discard them.

A future release of the context-awareness module should bring some social features to the recurring trip collection. For instance, if the user logs on social networks such as Facebook, Foursquare, or LinkedIn, category, name, or other context-

¹¹ <https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi>

¹² <https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CoreMotion.Reference>

¹³ <https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi>

¹⁴ <https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation.Framework>

tual information of the visited places could be added to the trip, which could enable more relevant social matching among users with similar mobility profiles.

3.4.4.2 Business Logic

The structure of the Business Logic (*BL*), which is the core module within the back-end architecture, is built upon a Python/Django service exposing a REST APIs and a set of tools enabling all possible interactions between the front-ends and the Matching Engine. The developed APIs gives functions in two directions: to the front-end for the obvious user interactions, as well as to the Matching Engine, creating a call-back mechanism for the data, to be retrieved and easily reused.

The BL relies on two distinct databases to manage the persistence: a *MongoDB* instance to manage the persistence of the complex objects and a *Redis* database to manage access tokens and device details. The different approaches come from the different interactions needed in the various phases.

3.4.4.3 Matching Engine

The Matching Engine (*ME*) is a parallel computation module that aggregates users with similar mobility routines (recurrent trips) as passengers (riders) of the same car (hitch) leaded by a user (driver) that makes available that car for ride sharing.

Definition 3.11 (Hitch). A hitch can be described as a set of ride sharing recommendations for riders that potentially can join a shared car made available by its driver. It is worth noticing that the total amount of riders assigned by ME to the same hitch can exceed its maximum capacity (e.g., 4 seats) because its purpose is to provide users with all the ride sharing recommendations enabled by any available driver.

The ME is enabled by a data mining algorithm that is run for each new recurring trip discovered by the BL on a time span of 7 days (starting from the current day). To date, since the inferring of recurring trips made on the data collected by the mobile context-awareness tool is not working properly, the mobile app lets the users to insert trips manually (e.g., both by specifying origin and destination addresses and times and by tracking the mobile phone location while traveling).

Matching Rules

The similarity of a rider's mobility routine, with a potential driver's one, takes place whenever at least one of the following matching rules is satisfied:

Matching Departure Driver and rider Departure locations are close to each others within a given *Spatial Threshold* (Pick-up point colocated with driver departure), their Departure times differ less than a given *Temporal Threshold*, and at

least one geographical point exists, on the driver's route, which is close to the Arrival location and time of the rider's trip within the same Spatial and Temporal Thresholds (Drop-off point on the driver route).

Matching Arrival Driver and rider Arrival locations are close to each others within a given Spatial Threshold (Drop-off point colocated with driver departure), their Arrival times differ less than a given Temporal Threshold, and at least one geographical point exists, on the driver's route, which is close to the Departure location and time of the rider's trip within the same Spatial and Temporal Thresholds (Pick-up point on the driver route).

Matching Subtrip At least two geographical points exists on the driver's route, which are close, respectively, to the Departure location and time and the Arrival location and time of the rider's trip within the given Spatial and Temporal Thresholds (Pick-up and Drop-off points on the driver route).

The three matching rules (depicted, respectively, in Figures 3.23-a, 3.23-b, 3.23-c) show that three matching types (*MT*) are possible between any given driver-rider pair.

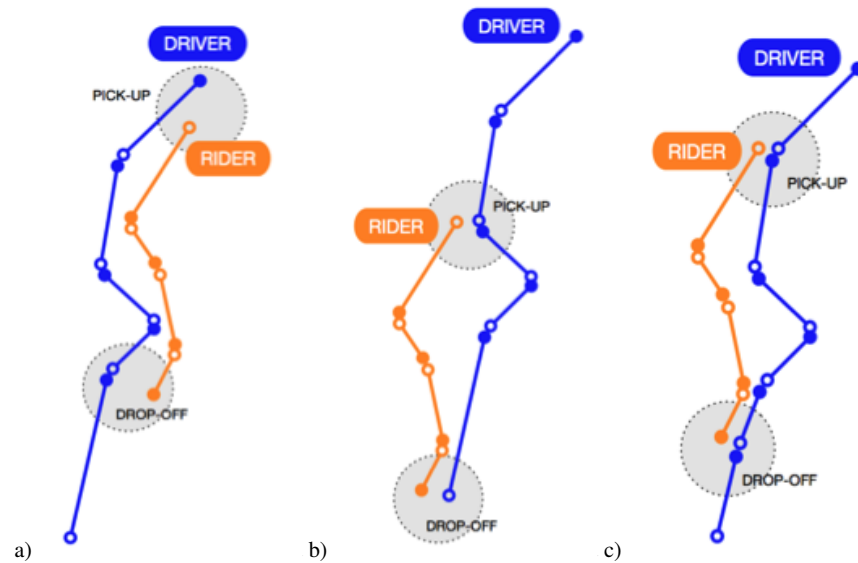


Fig. 3.23 The three matching rules defined: **a)** Matching Departure; **b)** Matching Arrival; **c)** Matching Subtrip.

Matching Workflow

The workflow of the ME starts by loading the given initial batch of recurrent trips. Entries are adapted for processing to add any missing routing information and to infer the trip trajectory (sequence of *GeoHash*¹⁵ described by temporal information and index of the geographical point in the *GeoJSON*¹⁶ geometry of the trip related to that GeoHash). Each trip is initially classified with regard to its attribute that define the user as a driver or a rider.

Once the potential matches are discovered for any MT over all the driver-rider pairs (*MATCH I*), and the possible hitches are populated (*HITCHES I*), the drivers that do not offer a suitable ride for any available rider in the set (non-matching driver - *ND*) are then considered potential riders of any suitable hitch found so far for a new run of the ME (*MATCH II* that incrementally originates *HITCHES II*).

After the second round of three MT discovering, the ND that do not match with the existing hitches (i.e., driver is not assigned to any hitch) are taken into account for a further matching discovering (*MATCH III*) with a duplicate of the ND collection.

With the third run of the ME, matches are scanned between pairs of ND (this process initializes the new set *HITCHES III*). In the first copy, ND are considered as drivers of new (empty) potential hitches, in the second one, ND are treated as potential riders.

Finally, *HITCHES III* is combined with *HITCHES II*, by generating the final set of recommendations so called *HITCHES IV*.

In general, the ME performs three matching discovering between pairs made of different matching objects (*MO*), and for each one of them the three MT are evaluated. The matching strategies described in [14] are fully implemented in the workflow, and adapted to the constraints and functionalities put in place by the *Apache Spark Framework*¹⁷ deployed on top of the parallel computation environment on FIWARE Cosmos instance. A schematic view of the workflow, implemented as a Spark tasks pipeline is shown in Figure 3.24.

In the actual implementation, Cosmos has been replaced by a private *Apache Hadoop*¹⁸ instance. This choice has primarily been taken since the FIWARE GE did not provide a deployment of Spark, which was necessary for performing matching.

Interfaces

The ME is deeply integrated in the whole cloud environment thanks to a set of available interfaces that connect it to the BL and The Hadoop BigData enabler.

Figure 3.25 offers a schematic view that demonstrates the integration of the ME within the back-end architecture. In particular, ME naturally resides on the same

¹⁵ <https://en.wikipedia.org/wiki/Geohash>

¹⁶ <http://geojson.org>

¹⁷ <http://spark.apache.org>

¹⁸ <http://hadoop.apache.org>

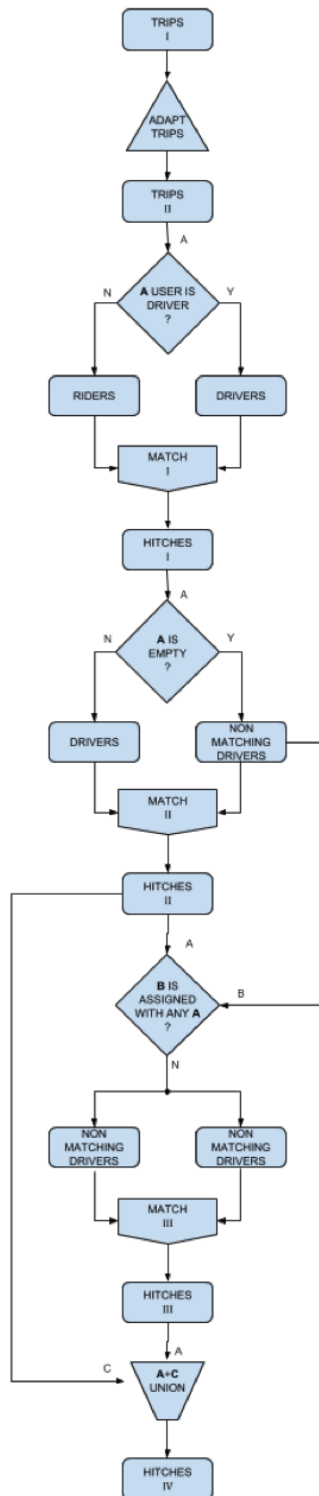


Fig. 3.24 The Matching Engine workflow developed in Apache Spark.

machine where an Hadoop data-node and a *Flask Service*¹⁹ are running. The first handles the needed computing resources by the Spark job and stores the outputs of the matching procedure, while, the latter, exposes a REST API to the BL, and it is exploited to submit the Spark job itself to Hadoop. Moreover, once matches are discovered and stored in the *Hadoop Distributed File System* (HDFS), a callback is sent by the ME to the BL. Data, then, flows from HDFS to BL via *WebHDFS*²⁰. There, it is processed through Orion, where matches are published and notified to users, and sent to Cygnus, which manages the history of found matches for each user (aggregated by user-Id) by persisting them back to Hadoop.

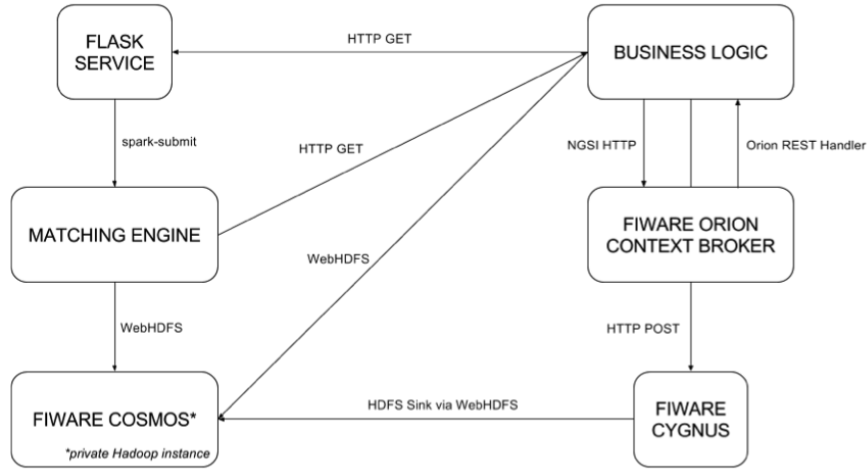


Fig. 3.25 The Matching Engine interfaces diagram.

3.4.4.4 FIWARE Orion Context Broker

The Orion Context Broker is a Publish/Subscribe FIWARE technology that performs registration, update and query of information according to certain contextual conditions. The use of Orion consists of a publish/subscribe mechanism that involves the storage of any recurrent trip and Hitch as *Context Entities*, and the managing of the subscriptions as *Matching Notifications* for users that will be recommended on a certain Hitch.

The step-by-step operations performed by Orion is depicted in Figure 3.26. Basically, in the first place (*Step A*) Context Entities are created (stored on local DB) for any recurrent trip and the already enabled Hitches that constitute inputs for a new ME run. (*Step B*) Each recurrent trip of every users is then subscribed to any

¹⁹ A RESTful Web Service written in Python.

²⁰ The public HTTP REST API exposed by Hadoop

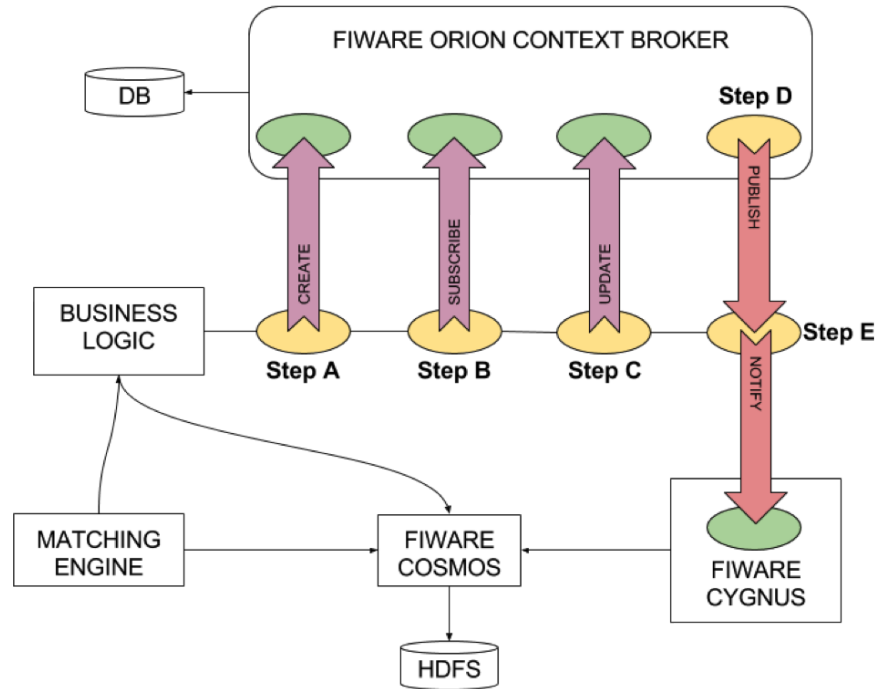


Fig. 3.26 A step-by-step view of the FIWARE Orion Context Broker operations: **(Step A)** Context Entities are created; **(Step B)** subscriptions on changes are performed for everyone of them; **(Step C)** Entities are updated whenever ME discovers new matching trips; **(Step D)** updates are published for the subscribed entities; **(Step E)** updates are notified to FIWARE Cygnus, which is in charge of tracking historical updates for statistics.

contextual change of it (i.e., a matching trip is found). Afterwards, once the ME discovers new Hitches made of users that could share a single ride to cover their recurrent trips, Orion is in charge of (*Step C*) updating the stored Context Entities with the matching information and then (*Step D*) publishing the contextual updates for the subscribed users, which will be provided with tailored ride sharing recommendations containing individual pick-up and drop-off locations and times. Finally, (*Step E*) the historical view of the updates is tracked thanks to FIWARE Cygnus (persisted on HDFS), which is notified at every entity update.

3.5 State of the Art

Ride sharing has increased its popularity in recent scientific literature and in real-world use cases. It is one of the most popular way of commuting, and maybe one of

the least understood. It has been analyzed from very different points of view going from Psychology [136] to Economics [32], passing through Computer Science and Transport Science. This Section presents related works for specific topics that support a large-scale ride sharing system, as well for exemplar applications of shared mobility services.

Data and Analytics

Across previous works, GPS traces and CDR events are the main data used for the generation of synthetic mobility routines. The CitySense project²¹ used GPS and WiFi data to cluster people whereabouts and discover hotspots of activity in the city area. In a similar work based on extremely large anonymized mobility data coming from Telecom operators, authors were able to extract the spatio-temporal dynamics of the city, highlighting where people usually go during the day. The authors were able also to identify the most visited areas by tourists during the day and the typical time of the visit (see, e.g., [26, 64]). While these works focus mainly on hotspot identification, the approach presented in Section 3.2 goes further and is able to identify and label patterns and routine behaviors that will be useful to identify ride sharing opportunities.

In [53, 56] authors proposed the use of probabilistic topic models to capture human routines from cell tower connections. The presented use case uses a more complex dataset, thus allowing to analyze the topic models method at a finer-grain scale with a higher number of places. As mentioned earlier in this Chapter, the geographic coordinates provided by GPS and CDR dataset allow to enrich the location vocabulary with a higher number of places (in contrast with the ‘home’, ‘work’ and ‘elsewhere’ label used in [53]). In addition, Section 3.2 also presents algorithms to automatically label topics in order to make them more understandable and usable.

The M-Atlas approach is a recent proposal to extract patterns from mobility data [129]. M-Atlas creates an origin-destination matrix counting the number of repeated trips in the dataset. High-values in this matrix represent recurrent patterns. The proposed topic model is more flexible in that, since it can identify patterns even if no direct trajectories between places are present.

The presented method in Section 3.3 exploits systematic behaviors in human daily travels to produce ride sharing recommendations. Therefore, a crucial point is the retrieval of spatio-temporal patterns of presence in geographical locations. Using fine mobility data, [130] and [69] extract not only patterns of presence in locations, but also systematic routes, from GPS trajectories. These routes are used to find user-to-user matches and to provide ride sharing recommendations. [88] developed a methodology working on GPS data that organizes trajectories in a trie to speed up geographical carpooling queries.

In both the proposed approaches, only users’ locations and users’ presences were extracted from mobility data, while they infer the movements among them through a route planner. Even if coarse grain data is used, and ride sharing matches may be

²¹ www.citysense.com

less accurate, on the other hand, it may be easier to enrich the service by taking into account also information other than mobility itself.

Mobility has been used in conjunction with other types of information to improve recommendations. For example, [37] proposes a location prediction model that combines users movements with a social network structure, highlighting how the social dimension may improve significantly the performances.

Geo-social data is used in [52], recommending individuals to join their friends during trips by using home location models and users' similarities. In [39] the authors derive home and work locations using Twitter and Foursquare data, then social ties are used to develop an algorithm for matching users with similar mobility patterns. Twitter data is used also in [60] as a complementary source of information for urban planning applications.

While in these works the additional knowledge is used mostly to improve performances or to filter the data for more focused recommendations, the work presented in Section 3.3 uses social connections and interests to enlarge user impact, and with the specific goal of providing more enjoyable trips. The methodology also helps users to overtake psychological barriers associated to riding with strangers. Similarly, [45] proposes a model to find compatible matches for traditional groups of users and also to find rides in alternative groups. Finally, [13] implemented an application for car sharing by exploiting a clustering algorithm applied to labeled trajectories, which is the precursor of the work presented here.

Users' Matching

Ride sharing is often modeled as an optimization problem, where it finds several solutions. The purpose of the previous studies are focused in producing recommendation while other works, like the following, mainly concern about discovering the largest amount of users' matches as possible, and so about optimizing the traffic scenario, such as minimizing the number of cars, minimizing the traveled distance, etc. In [43], the problem is faced as an optimization task reduced to the *chairman assignment problem* [128], while, in this proposed methodology an instance of the *transportation problem* [108] is used.

Authors in [91] consider simulated straight-line trajectories observing only origin and destination of trips and classify users as eligible or ineligible for carpooling by minimizing the time of the trip.

In the experiments presented in Section 3.3, the number of cars is minimized, like in [154], where the authors propose a carpooling based on taxicab. That is, they analyze the reduction of circulating taxis in presence of ride sharing.

In [93], instead of inferring mobility routines, authors present a dynamic ride sharing service to efficiently serve real-time requests sent by taxi users and to generate ride sharing schedules on actual trips that reduce the total travel distance. Both the two methodologies proposed in this Chapter do not concern a real-time service, but they are smart offline recommendation services.

In terms of real-time carpooling, [100] develops and implements this concept by allowing a large base of member passengers and drivers that declared their route

to be matched against each other automatically and instantly using mobile phone calls. [77] proposes a carpool system in which users can submit carpool requests and obtain matches that are among the most optimal, and operates with significantly less computational complexity to require less services computing time. The authors of [84] present an algorithm that provides carpooling advises by maximizing the expected value for negotiation success. The probability values are calculated by a learning mechanism using person features, trip characteristics, the negotiation feedback. In the case studies presented, negotiation was not taken into account, since users perhaps do not expect to be asked to share their rides, and the assumption they accept any suggestion was made.

With a spirit similar to GRAAL, [115] proposed a journey planner that maximizes (a different concept of) the enjoyability of the journey. Besides solving a different problem (route planning, as opposed to ride sharing), their concept of enjoyability differs from the one presented here, as it is based on characteristics of the journey itself (presence of trees, landscape, etc.), rather than on the relation among traveling users.

Shared Mobility Services

The first applications promoting car pooling and ride sharing have been developed almost 20 years ago. For example, systems like *www.mitfahrzentrale.de* provide advanced car pooling solutions since 1998. However, the vast majority of systems that are in place today supports ride sharing for *planned* and *long-distance* (> 100km) travels.

BlaBlaCar, for example, is a fully working web service enabling peer-to-peer ride sharing among users. Users can buy and sell rides on a eBay-like platform. While this system takes advantage of novel Web technologies, it still does not automate the process of identifying matching rides and users have to explicitly enter rides availability and needs. The system presented in Section 3.2, instead, provides an algorithm to perform the latter step in an automatic way on the basis of past user's routines. Accordingly, the mechanisms could be effectively integrated in systems like BlaBlaCar to provide an easy and automatic way to users that would like to participate in the system.

The work presented in [7] describes a multi modal journey planner that, relying on a number of mobility data, computes the probability distributions of end-to-end travel times, taking into account uncertain departure, travel and transfer times. While the goal of this system is rather different from the one presented, the basic mechanisms to extract information from underlying mobility data are similar.

Another example of using social interactions to ease traffic problems is described in [35]. Authors propose to use crowd-sourcing to collect, and keep updated the information about parking slots in the city. Although they focus on a different case study, some of the proposed algorithms to automatically extract places and routine behaviors could effectively complement crowd-source information collection.

[24] introduces a totally social-data-oriented ride sharing with a Facebook-based matching, a Twitter-based traffic monitoring, and a Flickr-based incident reporting application.

Recently, several researches tried to estimate the benefits of large-scale ride sharing platforms in terms of trips, vehicles, or congestions, saved. [13] describes a previous inspirational work on estimating ride sharing potential in Torino. However, in that work only matches among users having both the same starting and ending locations were considered, i.e., it does not consider the case of picking up users en-route. Under that assumptions, the proposed methodology obtained 35% trips saved for 1Km detours. The current work presented in Section 3.2 improves on that approach by considering en-route pick-ups and using the actual street map of the city. Moreover, a better comparison of the results deriving from home-work commute, and the ones from generalized whereabouts routines, was made.

From a more theoretical perspective, the work presented in [67] use a probabilistic model to analyze the likelihood that a person will be successful in finding a ride-match, given a pool size of potential ride matches. In the same work, the authors also observe that there are many obstacles, primarily in terms of communication and social norms, preventing ride sharing from actually happening. [19] analyze this latter issue further, trying to identifying the social enablers for ride sharing. Similar in this direction is the work on persuasive technologies [58, 116] that analyzes social and psychological aspects related to behavior-influencing technologies and crowd-source coordinated activities. These aspects are very important for the practical applicability and success of ride sharing systems. In this line of inquiry is also the analysis of incentives and reputation mechanisms [116] supporting ride sharing activities. The approach presented in Section 3.3 deals with this issues.

An important factor that affects the amount of ride sharing opportunities, independently from the analytical tools exploited or the matching mechanism put in place, is constituted by the city where the approaches are being tested. In particular, specific urban features such as road topology and people distribution can impact significantly on the results of large-scale sharing systems. The results presented for the case study in Section 3.2 show difference for potential ride sharing services implemented with the same methodology over Torino and Milano. Alternative results proposed by other research groups for various cities, have been summarized in the table below.

Study	City	Trips saved
Alexander, Gonzales [4]	Boston	38%
Bicocchi, Mamei [13]	Torino	35%
Cici et. al. [38]	Madrid	50%
Santi et. al. [123]	New York	95%
Uber (UberPool) [54]	N/A	32%
Trasarti et. al. [130]	N/A	55%

In [4], authors analyze the ride sharing potential by matching user CDR traces. Assuming 50% adoption rate, they found: number of vehicles -19%, vehicles mile

traveled -11%. They also studied the impact on traffic congestion obtaining a reduction of congested traffic time by 35%.

In [38] authors analyze mobile phone location data to measure the amount of overlapping in people daily mobility as a pre-condition to enable ride sharing opportunities. They show that simple ride-sharing among people having neighboring home and work locations can reduce the number of cars in the city at the expense of a relatively short detour to pick up/drop of passengers. Specifically, the report about 50% cars reduction for 1Km detours. The work presented in this thesis is very similar in spirit with this research, in addition the proposed mobility analysis methodology can improve and generalize mobility routine identification. The proposed topic model, in fact, can identify also routine behaviors other than home-work commute.

In [123] authors estimate the potential of shared taxi in New York City. This work is based on analyzing taxi GPS traces, and in applying a graph matching algorithms to identify ride sharing possibility. Results illustrate that in this setting 95% of the trips could be saved. Such a high percentage is the results of the regular grid of streets in Manhattan. Similarly, in [94] authors proposed a taxi-sharing system that accepts real-time ride requests sent from smartphones and schedules proper taxis to pick up them via ride sharing, subject to time, capacity, and monetary constraints. Monetary constraints have been proposed to provide incentives for both passengers and taxi drivers.

In [130] authors analyse GPS data to extract users' mobility profiles (i.e., frequent paths) via scan statistics and clustering mechanisms. Then they run ride sharing experiments obtaining that 32% of trips could be saved if users are willing to walk for 2.5Km to reach the ride and tolerate up to 1 hour delay. Authors generalize the analysis to GSM data (CDR) reporting larger ride shareability due to inaccuracies in the user localization both in space and time (i.e., inaccuracies soften the spatio-temporal matching constraints).

Finally, similar conclusions are described in a report from UberPool [54] estimating 55% of trips to be saved from a widespread use of their system.

3.6 Conclusions

Despite ride sharing is becoming a well known concept that counts several deployments all over the world, it actually faces many obstacles in its way to overcome private modes of transportation. Users may be asked to spend efforts on planning their trips (either recurrent or sporadic), and communication and social norms are often missing, thus preventing ride sharing from actually happening. Heavy users interactions and lacks of adequate social enablers can represent relevant issues of a ride sharing system.

The case study presented in Section 3.2 proposed a recommender system capable of identifying opportunities for sharing rides in urban scenarios by mining mobility traces collected from telecom operators. Experimental results showed the fea-

sibility of the approach and its potential to mitigate traffic-related issues. In future work, user studies can be conducted to evaluate the contribution of the identified whereabouts patterns in the development of behavior-influencing applications. Furthermore, the differences emerged between Torino and Milano might suggest that specific urban features such as road topology and people distribution impact significantly on the results of large-scale sharing systems. On these basis, in future work, additional cities will be analyzed to further investigate this issue eventually identifying policies to build trip-sharing-ready urban environments.

The results obtained were interesting in both the tested cities and cases. The case concerning passengers diversions could lead to a reduction of 50% of the number of circulating cars by accepting to walk at most 1,000m (depending on the city) to reach a shared car. Drivers diversions, instead, are less efficient allowing at most a reduction of 40% of circulating cars in the best simulated case. However, even though the number of actually driven kms might increase because of the diversions drivers are asked to take, the reduction of circulating cars could lead to a reduction of many other traffic-related issues.

The case study presented in Section 3.3 tried to go further, by enabling social measures on discovering the likelihood of riding a shared car. GRAAL solved a multiobjective model for carpooling recommendations as a weighted linear combination on the amount of cars used (which is minimized) and the total enjoyability (which is maximized).

The concept of enjoyability built up on top of like-mindedness (i.e., topic similarity) and homophily (i.e. how much a user tends to cluster with similar user), which are two well established concepts in social network analysis. The conducted survey showed the results of two optimizations (one minimizing cars, one maximizing enjoyability) to potential users of carpooling. By asking them to choose between the Social and Green solution, the tuning of the weight for the linear combination in the optimization function was possible. A total amount of 237 answers was received, 39% of which were in favor of the Social solution, motivating this work, and providing the needed weight.

Finally, the results of the multiobjective optimization in terms of cars saved and enjoyability were presented both from the city and the user perspective. With the crowd-sourced alpha, GRAAL was able to save up to 57% of the cars needed, while the total enjoyability was up to double. From the user perspective, the increasing of the entire per-car distribution of enjoyability was shown. Future works could focus on customized optimizations at the level of a single user. This means that each single user should have her/his own value for α , since different people may have different preferences. Moreover, one can think to extend the work by including other social factors in the optimization model, such as gender match, trust, and network similarity.

The real-world case study presented in Section 3.4 constitutes the actual implementation of SmartHitch, an advanced ride sharing system that takes into account the complex study of mobility and social similarities among the users of the service. SmartHitch would like to make ride sharing an easy to setup service, which enables automatic recurrent trip discovering to be totally unobtrusive, and which character-

izes the goodness of a matching according to adequate social measures to be trusty. However, the full set of features has not been developed yet. Thus, future works will cover further developments and some relevant piloting sessions with selected users' communities.

Chapter 4

SAMS: Crowd Steering

Smart phones and environmental sensors make it possible to dynamically monitor the positions and patterns of movements of people in urban areas and public spaces, identify or predict possible dangerous situations (e.g., overcrowded areas) or simply recognize the profitability of a different patterns of distribution and collective movements. Such technologies can be exploited to steer the movement of people in public and urban spaces (i.e., suggesting people where to move to eventually reach some desirable global configuration in terms of crowd distribution), both in indoor and outdoor environments. In this Chapter, the so called crowd steering problem is defined by overviewing and motivating the goals pursued, identifying the technologies that can be put in place, the possible strategies to direct people movements, and the open challenges related to solving it.

4.1 Introduction

Pervasive computing technologies [42], such as smart phones, embedded sensors, and cameras distributed in our urban environment make it possible to monitor at very fine-grained levels a variety of social and environmental phenomenas. These technologies are being already widely used to understand the dynamics of cities [36], and they have also been used to plan building structures and pathways therein (e.g., emergency exit routes) [97, 120].

Smart phones, in particular, introduce the possibility of providing users with information about the current crowd conditions and possibly suggest them directions in order to better move in the city (i.e., to avoid crowd or simply to get alerted about specific conditions). Providing the user with this kind of feedback makes a big shift from simply sensing to actually actuating (i.e., steering the behavior of people so as to serve both their individual needs and the global needs of the city [150]).

The technologies that can be put at work to enforce steering go beyond apps on smart phones, but can include advanced wearable systems and interfaces (such as vibration and sound) [125], and also technologies to collectively suggest all people

in a specific location (e.g., in a room) how to move, such as public sound announcements or public displays [47, 112]. In addition, societal or economic incentives can be put at work to encourage people to follow the suggestions [85, 89].

Clearly, in addition to technological means, there must be some strategies to identify what suggestions should be given. The main problem in this task is to devise strategies serving individual needs *co-existing* with possibly conflicting strategies that serve global needs of the environment in which users situate.

The contribution of this introductory section is to unfold the many issues related to crowd steering in urban spaces. In particular:

- Analyses the motivations to enforce strategies and techniques for crowd steering in urban spaces, both at the individual and at the global level (Section 4.1.1);
- Overviews the key technologies that can be put at work to support crowd steering (Section 4.1.2);
- Presents some of the more promising strategies that can be devised to support crowd steering at both individual and global levels (Section 4.1.3).

4.1.1 Goals

The problem of steering the movements of people can arise as a means to help individuals or groups in achieving their own mobility objectives, or as a mean for the institution managing a public space to achieve some sort of overall distribution property within (e.g., the institution managing an exhibition area wants to achieve better people throughput in the area). A crowd steering problem can contemplate both individual or global goals (even at the same time), toward the addressing of respective individual and institutional needs.

4.1.1.1 Steering to support individual needs

In a number of application scenarios, users would benefit from having solutions helping them get oriented in the environment, and move by avoiding the crowd.

Most navigation systems in vehicles already include solutions to suggest routes based on the detected traffic conditions, so as to avoid traffic jams and minimize travel time. However, when it comes to pedestrians, none of the existing navigation systems considers the actual density of people in streets and walkways to suggest directions. There are several reasons why individual people moving in an environment may want to be recommended on where to go, and such motivations differ from those typically involved in vehicles' navigation. In particular:

- When moving in environments such as big exhibitions or city fairs, pedestrians (unlike car drivers) are not simply willing to reach a specific destination. They rather want to visit different places (e.g., exhibitions booths), but may have no

clue on what could be the best order in which to visit them, to make the visit more enjoyable and efficient.

- As it happens to drivers, it may also occur that pedestrians can be slowed down by extreme crowd conditions. So it would be good to receive advices on how to avoid crowd. However, for pedestrians, avoiding crowd may be useful also for simply better enjoying a place.
- In some cases, pedestrians may wish to receive suggestions on how to move within an environment so as to avoid peculiar situations, such as moving across unsafe neighborhoods or extremely polluted streets.

All the above considerations are particularly critical for groups of people sharing similar interests or goals, and willing to share a mobility path across an environment. Examples may include groups of tourists or groups of teachers herding young children around. As a final consideration, the need of dynamically forming groups in which individuals can join and leave while the group walks across a path (as in walking school buses [143]), necessarily requires some form of steering to facilitate its dynamic shaping.

4.1.1.2 Steering to support institutional needs

Focusing on a more global – institutional – level, steering the movement of people can be used by the administrators of some public spaces (e.g., a large museum, an exhibition, a fair, a stadium, a large park) to promote a better or safer exploitation of the space. More specifically, the goals of an institutional crowd steering may include:

Load balancing to ensure a fair exploitation of the available space and an even distribution of people. For instance, load balancing may aim at avoiding that people during an exhibition concentrate in specific areas simply because that area is easier to be reached – an approach experienced at the Vienna City Marathon where a smartphone application guiding people to alternative (less crowded) spots [112] was deployed. In addition, load balancing may be aimed at ensuring that all areas of an exhibition receive a similar amount of visitors (e.g., consider the case in which exhibitors pay to be present and would not tolerate being penalized for being at an unfortunate location).

Planning for evacuation to ensure that the distribution of people facilitates a quick evacuation from the area. Such an issue may be particularly important for dangerous situations but – independently from safety issues – it may also be important to ensure no one is penalized with this regard. For instance, at some airport (such as Heathrow) the waiting time at different security gates is advertised via public displays, implicitly enforcing local balancing through displays that implicitly invite people to move towards the less crowded gates.

Enforcing other specific strategies such as favoring the movements of specific classes of persons, or attracting people towards specific location at a specific time. For instance, crowd-sensing campaigns (e.g., exploiting people as sensors

by having them to report about some events in some specific locations [81]) may require dynamically mobilizing people according to some specific sensing goal.

Of course, whatever the goals to be achieved, any institutional strategy for crowd steering has to take into accounts the fact that a percentage of people can decide to ignore the suggestions, and follow their own schedule.

4.1.2 Technology

Crowd steering is enabled by a number of recently emerged ICT technologies.

4.1.2.1 Detecting People Location and Density

The first, necessary, ingredient of any crowd steering approach is the capability of detecting the location and density of people. With this regard, the number and accuracy of available technologies is constantly increasing.

For outdoor scenarios, smart phones currently have a very good localization accuracy (thanks to the concurrent exploitation of GPS, cellular telephony signal, and WiFi triangulation).

For indoor scenarios, the dense deployment of WiFi access point makes it possible to achieve high-levels of accuracy via WiFi triangulation even indoor [73]. In addition, novel low-cost technologies based on beacons [160] (small Bluetooth-enabled boxes to be detected by a smart phone, and enabling to effectively estimate its distance from it) can be very effective to achieve accurate indoor localization.

More specifically, information about people density in a certain area can be extracted by sniffing signaling messages sent by smartphone radio interfaces such as Wi-Fi and Bluetooth without the need of a specific application or hardware on the user side [124]. Naturally, cameras and computer vision can be also used to analyze the presence of people and their density.

4.1.2.2 Smart Phones and Mobile Apps

Smart phones, other than being a very important tool to localize people (as discussed in the previous subsection), can be very important to enforce crowd steering strategies. In fact, the primary means by which one can expect a crowd steering strategy to inform individuals about a suggested route, is via some mobile app. Perhaps, in the near future different personal appliances will emerge as the primary mean via which to receive crowd steering suggestions. Smart watches – if at all successful – will be definitely a suitable tool to adopt. It is also worth outlining that smart watches, other than being suitable for suggesting directions with traditional GUI, will enable exploring alternate means for suggestion (e.g., vibrations or sounds). Similarly, smart

glasses and augmented-reality goggles could have a strong role in supporting crowd steering activities.

4.1.2.3 Digital Signages and Public Displays

Public digital signals are already widely adopted in vehicular traffic to dynamically provide real-time traffic information, and to suggest routes and detours. However, we are assisting also to the application of digital signals and public displays to suggest directions to people, and to enforce specific crowd steering policies [48] (as it is in the already mentioned examples of displays that are signaling the length of queues at Heathrow airport).

In the future, with the increasingly dense deployment of interactive wall-mounted displays, it will be possible to exploit such displays to provide real-time information on the density of crowd at different locations, and will be possible to use displays to suggest passing-by people where to go, typically with the goal of enforcing some global-goal strategy. Clearly, exploiting public displays to enforce private-goals would raise privacy concerns, in that it would disclose information about individuals' intentions and goals [49].

4.1.2.4 Morphable Architectures

A last technology, possibly a bit more futuristic, is morphable architectures: places in which it is possible to dynamically change the topology of space (i.e., making walls and door appear and disappear) so as to force specific movement in the crowd.

To some extent, dynamic signages can already obtain similar effects (e.g., by making a one-way sign appears at the beginning of a street). However, recent conceptual experiments (e.g., the Slothbot moving wall¹) show that the concept could be applied by actually reshaping the environment.

4.1.3 Strategies

The previous section presented the enabling technologies for crowd steering. This section, discusses possible “crowd steering strategies” to analyze location and mobility data and decide which steering actions to actuate.

To create an effective taxonomy of possible strategies, three main axes, along which strategies might vary, were identified: *(i)* strategies with a local VS. global goal, *(ii)* strategies using local VS. global information, and *(iii)* strategies giving personalized recommendations to individual users, or multicasting recommendations to a group of users. Naturally there are not crisp boundaries among the categories and

¹ <http://arch-os.com/projects/slothbot/>

Local Info Global Info	Personalized Unicast	Multicast
	Local Goal	Global Goal
Local Goal	Recommend detours	Greedy steering
	Emergent Planning Navigator	Swarm Intelligent Mechanisms Traffic control

Fig. 4.1 Taxonomy of crowd steering strategies

the taxonomy space is a continuum of possibilities.

Local VS. Global Goal

The goal of steering strategies is to “arrange” people in the environment. A local steering goal is simply about sidestepping a local condition (e.g., an overcrowded room). A global steering goal is about achieving a global distribution of people in the environment. More specifically:

- Local goal strategies aim at recommending users about the next step to take. For example, a local goal could be “*avoid neighbor crowded rooms*”. The system would recommend the user not to visit a neighbor crowded room. This kind of strategies can be useful to better navigate across an environment sidestepping blocked or crowded areas.
- Global goal strategies aim at giving recommendations to influence the long term behavior of users. These strategies can be formulated with a user-centered perspective (e.g., “*minimize the time to visit the environment*”), or an environment-centered perspective (e.g., “*balance the crowd in the environment*”).

It is worth noting that the difference between local- and global-goal strategies is quite subtle. Necessarily, a global goal will be realized by a number of local actions by the agents. These local actions can be interpreted as enacting local goals. Vice versa, the repeated application of actions supporting a local-goal induces global emergent behaviors (associable to global goals) in the system. For example, one can think of a strategy that recommends the agents moving to less crowded nearby places. Its immediate local-goal is to “*avoid neighbor crowded rooms*”. However, the repeated actions by the agents will spread the crowd more evenly in the environment, thus having also global consequences.

Another dimension that characterizes the nature of the goal of a crowd steering strategy is the nature of needs that the goal itself tries to address. Thus, one can face individual- and global-goal strategies with regards to the amplitude of the effects generated and the needs addressed. For sake of simplicity, this discrimination is omitted in the proposed taxonomy, because of their usual coexistence in most of the crowd steering approaches (e.g., if the strategy aims to reduce visiting time per individual, at the same time, the crowd results even distributed across the environment).

Local VS. Global Information

The information available to the steering component might be local (i.e., the component provides recommendation on the basis of nearby crowd conditions), or global (i.e., the steering component oversees the whole environment).

- Local information strategies give recommendations by considering the user local neighborhood only. For example, a local information strategy can recommend user to visit the least crowded neighbor room.
- Global information strategies give recommendations on the basis of the knowledge of the whole people distribution across the environment.

Personalized Unicast VS. Multicast

The steering component might provide personalized recommendations to individual users (e.g., via their smartphone), or general recommendations to a whole group of people (e.g., via public displays).

- Strategies adopting a personalized unicast approach can give recommendations via the users’ smartphones. Using data from the smartphone (e.g., user location and planned itinerary) such strategies are able to highly personalize the steering recommendation.
- Strategies adopting a multicast approach can typically give recommendation via displays deployed in the environment. Recommendation can take into account approximate users location (as represented by the display own location) and other

information, but it cannot be specifically tailored to individual users, as all users would see the same steering advice.

Figure 4.1 illustrates the proposed taxonomy. A description of exemplary systems that would fit in each area of the taxonomy space follows:

Recommend detours and Greedy steering In these strategies the system recommends a next step (detour) on the basis of the crowd conditions in the user local neighborhood. In *Recommend detours*, the system can interact with users individually proposing detours compatible with the user planned trip. In *Greedy steering*, the system cannot give personalized advices so it tends to greedily steer users where most convenient at the moment. As already introduced, it is worth noting that the repeated execution of local goal strategies, often induces global “emergent” behaviors. On the one hand, these emergent behaviors can represent a global version of the local strategy. On the other hand, such behaviors are often hard to predict and control [71].

Gray areas These strategies are very uncommon. In fact, once global information is available, it is natural to pursue global goals rather than local ones. Global information is often an overkill for local goals.

Emergent planning and Swarm Intelligent Mechanisms These strategies try to exploit some kind of collective (i.e., swarm intelligence) mechanism to achieve a global goal on the basis of local information only. In *emergent planning* personalized recommendations can be given to individual users, accordingly the problem is to coordinate individual movements toward the achievement of the global goal. In *swarm intelligent mechanisms* control over individuals is much more coarse and self-organizing algorithms could be used [71, 96, 112].

Navigator and Traffic Control These strategies exploit global information to achieve a global steering goal. *Navigator* strategies guide individual users across the environment on the basis of all the information available. *Traffic control* strategies try to achieve a similar goal but without addressing individual users (e.g., coordinating traffic lights in a city to steer the traffic) [8].

4.2 Indoor Real Case Study: FairSteering

FairSteering is a mobility service provided to visitors of an exhibition fair to enhance their visiting experience. Users will be suggested with tailored itineraries according to their visiting preferences and to the real-time crowding level of the environment. They can be engaged by offering a mobile app containing a set of integrated services related to their visiting, such as:

- computation of the best visiting schedule according to every specific visiting preference (e.g., through an appropriate rating system, users can prioritize the visiting of booths);
- real-time notification for changes to the visiting schedule due to changing crowding levels at the booths;

- real-time notification for trendy events/booths;
- indoor navigation support with the exposition of contextual information (perhaps through a captivating augmented reality interface).

Indoor users' localization is performed by putting in place Wi-Fi hotspots (and running a triangulation based on sensed signals power) that offer free connectivity inside the fair, and Bluetooth beacons that enable contextual information of the closest booth to the user. The same technologies can help also in occupancy and crowding computation, which can be supported by other presence sensors such as CCTV cameras.

FairSteering aims to suggest users with a coherent visiting schedule according to their crowding preference (i.e., users may want to avoid overcrowded situations or they may prefer to visit trendy booths by seeking for crowd) and visiting preference (i.e., plan an itinerary to absolutely visit certain booths and to eventually skip other ones), and to provide fair organizers with an autonomous tool to even distribute the crowd across the fair in order to better manage evacuations in emergency situations.

4.2.1 Telecom Italia Project

In the project run in collaboration with the Telecom Italia Swarm Lab in Turin², two crowd steering strategies in a specific application scenario were evaluated: people visiting an indoor space structured in adjacent booths composing a trade fair or a more general event exhibition. The project consist in (i) modeling and running agent-based simulations to evaluate the goodness of the strategies and in (ii) proposing a demonstrative architecture of the FairSteering service by exploiting a real-time link to a simulated environment that computes the expected crowding levels at booth resolution in order to shape user recommendations.

The two strategies defined were:

- a *local info - local goal - multicast* strategy, referred to as Hot Potato (HP). This strategy recommends the user to visit the less crowded booth among the adjacent ones with regards to its current location, which belongs to the user's planned itinerary.
- a *global info - global goal - unicast* strategy, referred to as Shortest Path (SP). Given the user current location and its visit plan, this strategy takes into account the distance to all the booths in its itinerary, and the crowding level at every booth and on the path to reach it (i.e., all the components of the agent model – Figure 4.4) to provide individual recommendations.

This work focuses on these two strategies as they exemplify the difference between two major families of steering approaches: local info - local goal strategies in which agents follow simple rules with a local scope. Global info - global goal

² <http://jol.telecomitalia.com/jolswarm/>

strategies in which the behavior is more complex aimed at navigating the user more effectively. In any case, experiments with other strategies are in future work's plan.

The metrics used to assess the effectiveness of the crowd steering strategies are:

- Average visiting time per booth (individual goal);
- Crowd distribution variance (global goal).

The idea is that both metrics should be minimized. The recipients of the potential advantages enabled by such strategies are thus both visitors and fair organizers. On the one hand, visitors could avoid annoying situations if they are suggested to move towards less dense areas. On the other hand, fair organizers could manage evacuations and emergency occurrences thanks to a more equally distributed crowd.

4.2.2 Experiments

4.2.2.1 Simulation Setup

The scenario was modeled by using the open-source, agent-based, simulation engine GAMA³. The simulation is presented in terms of: (i) Environment model, (ii) Mobility model and Timings, (iii) Agent model, and (iv) Steering model.

Environment model

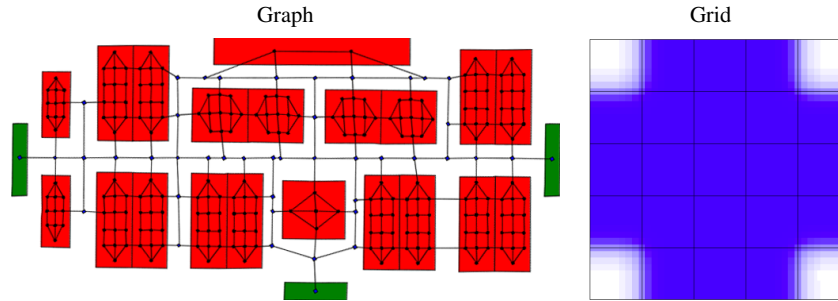


Fig. 4.2 The two GAMA simulation environments modeled. Graph Topology on GIS layered environment (**left**): traversable graph connecting entrances (green rectangles) and adjacent booths in fair pavilions (rectangular and triangular areas shaped by the graph over red rectangles). Grid Topology (**right**): adjacent booths (blue squares) and entrances (white squares)

The environment has been shaped by using different topologies. Experiments with both a traversable graph topology (on GIS layers – Figure 4.2-left) and a sim-

³ github.com/gama-platform

pler grid topology (Figure 4.2-right), to speed up computations, were ran. More in detail:

In the first place the experiments based on the graph topology as the simulation environment, and moving agents traversing it with shortest path algorithms, were ran. GAMA offers the possibility to model micro interactions among agents and between agents and the environment in a very detailed way. For instance, hallways can be modeled by using a proper parameter on the graph edges representing their actual width, or agents can be enabled with the capability to avoid each others through a proxemic bubble describing the required physical space to walk without changing their behavior[98] (i.e. by reducing their speed). While this approach is very detailed, it is also much demanding in terms of computational resources (with the available computation capabilities, only 1 experiment per day could be run at most).

Because of the long running times of the experiments, due to the too high computational complexity of the whole model, the analysis was complemented with a grid topology, in which the agents are hopping cell by cell to reach their target (without a long term path to follow), and micro interactions among agents are abstracted to an upper level of information. In particular, agents are not subjected to obstacles avoidance anymore, and their speed variation due to the crowd is translated to their hopping frequency cell by cell, since agents do not perform a continuous movement through cells but they literally jump into them.

Mobility model and Timings

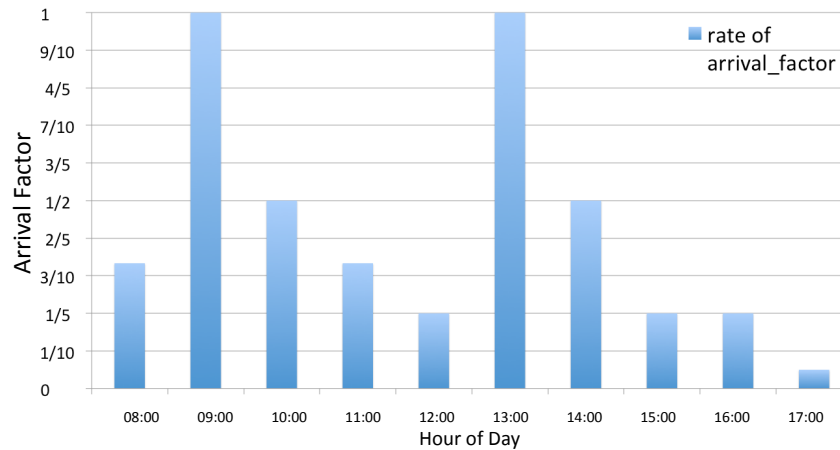


Fig. 4.3 Arrival distribution of agents on hourly base. The value returned by the timed function is then multiplied by the input parameter `arrival_factor`, which determines the actual number of agents that are going to be injected into the environment every minute of the specific time slot.

Agents have been modeled by assuming they were arriving at the fair with a certain arrival rate per hour, and they were moving in groups of people with a variable size defined between 1 and 4 on a Poisson distribution [106]. Among different experiments, the more adherent solution to a real fair scenario, to control the agents' arrival distribution, was best performed by a double peak function that mapped the rate per minute of a maximum value (called *arrival_factor*), which is set in input, on a hourly base (see Figure 4.3). Total simulation time has been set according to a typical fair opening time of 10 hours per day, where 1 step of simulated time was equal to 1 second.

Agents' speed and visiting time have been affected by the crowding in the surroundings of their locations[23], both while they were moving and while they were visiting a specific booth. This relation has been derived from the guidelines for the design of pedestrian spaces published by the US Department of Transport⁴. The queueing level-of-service described in the document suggests a threshold value on the physical space to identify visible effects of the crowding level on the visiting time. In particular, with lower or equal than 1.2 m² area available per agent, the crowding level would affect the behavior of people (i.e. the queue is long and people density is high) in terms of how much time they would need to visit that area.

Given the average visiting time in presence of a low crowding level $\bar{T}$ within a certain area A , the actual visiting time T linearly increases accordingly to the increase of the occupancies O in that area, by following this equation⁵:

$$T = \bar{T} * (1 + \frac{O - \frac{A}{1.2}}{\frac{A}{1.2}})$$

Agent model

Booths visiting order has not been scheduled a priori. Every agent is subjected to a detailed decision process that takes place after every single visit, and it determines the next booth to be visited. Agents' decisions have been driven by the combination of three main factors (Figure 4.4):

visiting plan The list of booths, thus goals, the agent would like to visit; it has been generated randomly per each agent, but composed by the same amount of booths; it also includes a fixed subset of popular booths that everyone visits;

local crowd The local perception of crowd measured in the surrounding space of agents' locations;

global crowd The global crowd measured by the crowd steering component. The idea is that the crowd steering component provides broadcast information to agents through broadcast displays in the form of a heat-map, and unicast recommendations that suggest the best booth to visit because of its adherence to the individual and, its low occupancy rate and it being on the shortest path to travel to. Agents' behavior will be then influenced by such information.

⁴ http://ntl.bts.gov/DOCS/11877/Chapter_8.html

⁵ http://ntl.bts.gov/DOCS/11877/Chapter_8.html

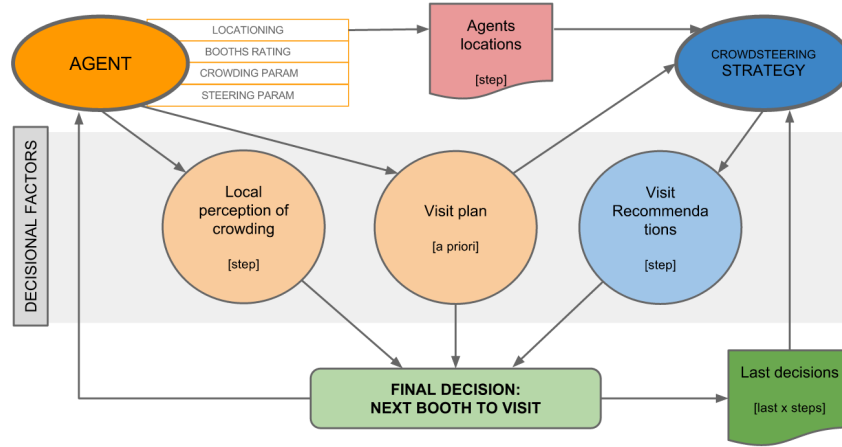


Fig. 4.4 The agent's decisional process. Agents' decisions are driven by the combination of: (i) visiting plan. The list of booths, thus goals, the agent would like to visit. (ii) The local perception of crowd. (iii) The global crowd measured by the crowd steering component. Each agent weights the three factors (the third one being the crowd steering recommendation) and decide the next place (booth) to visit.

At the end of each visit, the visiting plan has been reordered by ascending distance of each booth from the agent current location. Therefore, when the three factors were not able to distinguish a unique best target to visit next, or when the crowding was too low to interfere with speed and visiting time, the agents moved to the closest booth.

Finally, agent's willingness to follow the crowd steering suggestion was parametrized (with a parameter called *steering_influence*). This parameter was expressed as the fraction of moving decisions the agent takes in agreement with the recommendation provided by the steering component. The two crowd steering strategies were tested with different adoption rates.

Steering model

To put in place the two crowd steering strategies introduced earlier (HP and SP – Section 4.2.1), a central crowd steering component was exploited, with the main tasks of measuring the crowd at every booth, predicting the crowding dynamics for the whole visiting day, and effectively adopt one of the crowd steering strategies to shape users recommendations. The crowd steering component is continuously feed with (as depicted by Figure 4.4):

- last recorded agent location (agents locations are aggregated to measure the current crowding level at booths);

- agent's visiting plan (expected visiting times at every booth concur to identify future crowding levels);
- agent's taken decision for the given recommendation (the amount of decisions taken accordingly to the received recommendations impacts on the efficacy of the strategy – it represents the feedback loop of the awareness cycle presented in Section 2.1).

Referring to the presented agent model, in the HP strategy scenario, booths to be visited are re-ordered by the local crowd perception, and in the case any of the adjacent booths is planned to be visited by the agent, it then moves towards the closest one.

Whilst, in the SP strategy scenario, every booth is rated by linearly combine together the distance between the agent and every booth (i.e., a measure of the *centrality* of the agent position within the environment) and the current crowding level at every booth on the path to reach it from agent's location (i.e., a measure of the crowding level on the traversing path to reach every destination booth).

The booth at the top of the ranking will be the one recommended from the crowd steering component. By using two weighting parameters that complement to 1, the influence of each component on the rating is balanced. The parameters try to quantify how likely the distance affects the walking and visiting time towards a booth w.r.t. the crowding level on the way to it.

The ratio between the standard deviations of the mean distance and the mean crowding from the agent's location to all the booths determines the value of the weighting parameter for the distance component. Its complement to 1 determines the weight for the crowding level component in the linear combination.

4.2.2.2 Simulation Results

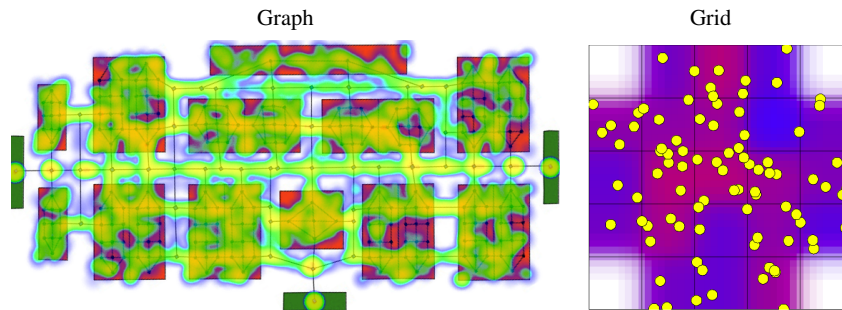


Fig. 4.5 Populated Graph Topology (**left**) and Grid Topology (**right**) with heat-map. This kind of maps allow to see clearly how users are spread across the environment.

A set of 4 experiments in total was ran. Each experiment has been characterized by a specific value of *steering_influence*, from 25% to 100% by 25% steps, as the

rate of decisions taken by agents according to the crowd steering recommendations. There were 10 different runs per each experiment, for both SP and HP strategies. In each run, the *arrivals_factor* changed from 20 to 200, by steps of 20, to reach different levels of crowd in the fair. All the experiments were based on the grid-topology environment model, as it was more computationally efficient. However, it was verified that results with the graph-topology environment model were obtaining coherent results.

Figure 4.5 shows an example of the running simulation in both graph topology (left) and grid topology (right) scenarios. The overlay (in graph topology) and the background color (in grid topology) of the booths are capable of showing a visual representation (heat-map style) of their occupancy updated in real-time.

Measures of **crowd variance** and **average visit time** were taken in each experiment (see Figure 4.6 for a comparison between the two crowd steering strategies and a no-steering scenario (NS)).

The experiments show an increasing gain in crowd balancing as an effect of any crowd steering strategy. In general, the more the agents follow the steering recommendations, the more equally distributed results the crowd. With a complete steering influence on the whole set of agents, the Shortest Path strategy is close to halve the crowding variance (44%), while the Hot Potato decreases it of the 18%. Nevertheless, both the strategies sharply impact only when at least half of the agents visit the recommended booths. It is worth noticing that the achievement in crowd balancing in terms of crowding variance do not negatively influence the average visiting time of the booths. The Shortest Path can even reach savings up to 72 seconds per booth in medium-high crowding conditions.

The fact the average visit time does not actually *decrease* due to the steering behavior is quite surprising. Although more experiments are needed, it could be due to the well-connectedness of the simulated floor plan. Given the large number of paths to reach a desired booth, agents are always able to circumvent overcrowded areas with short detours. In future experiments, test should be run on strategies in more constrained floor plans where steering should exhibit a larger impact on the agents' average visit time.

One not modeled aspect was the implicit overhead for the user to “see and understand” the recommendation (e.g., the overhead in focusing on the mobile phone screen to read the recommendation). This is an important aspect that could notably impact steering performance. This aspect will be addressed in future works [103].

In conclusion, other than illustrating the effectiveness of the presented strategies in balancing crowd distribution, the results showed also the possibilities offered by the simulation environment in testing different steering strategies, in different contexts, and at different levels of granularity.

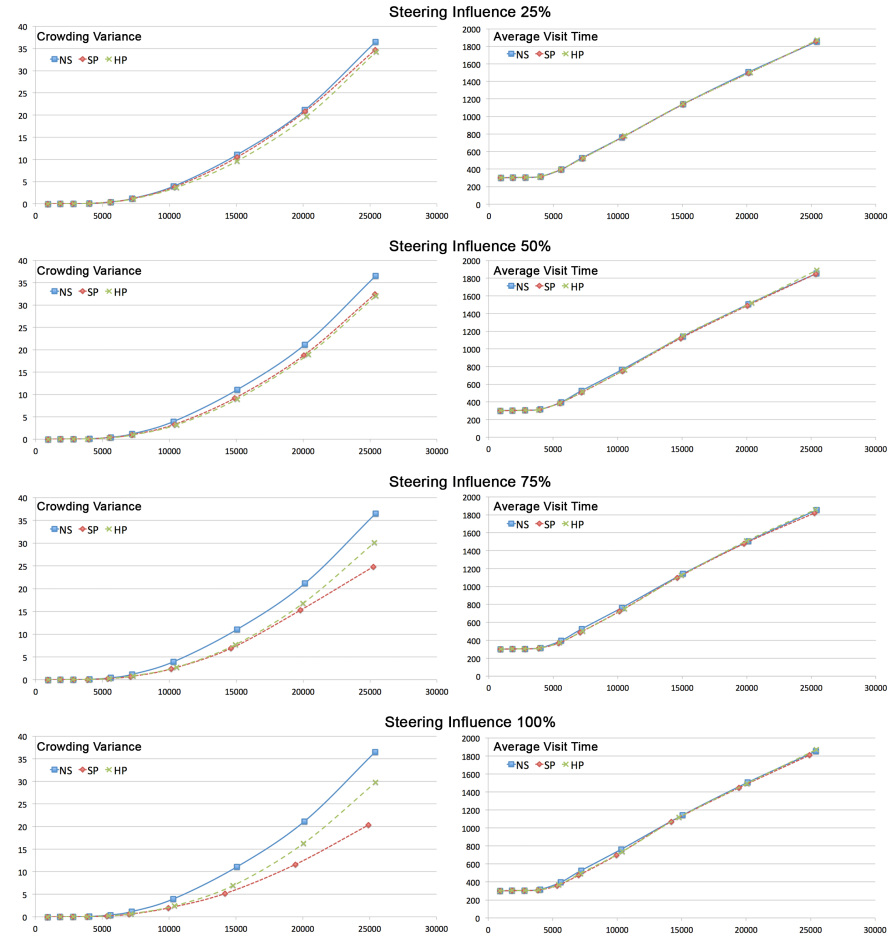


Fig. 4.6 Impacts of Shortest Path (SP) and Hot Potato (HP) strategies on the crowding variance (left column) and the average visit time per booth (right column) compared to a visiting strategy not influenced by any steering (NS). **a)** Performance achieved when 1/4 of the visiting choices are influenced by the steering strategy; **b)** Performance achieved when the half of the visiting choices are influenced by the steering strategy; **c)** Performance achieved when 3/4 of the visiting choices are influenced by the steering strategy; **d)** Performance achieved when all the visiting choices are influenced by the steering strategy.

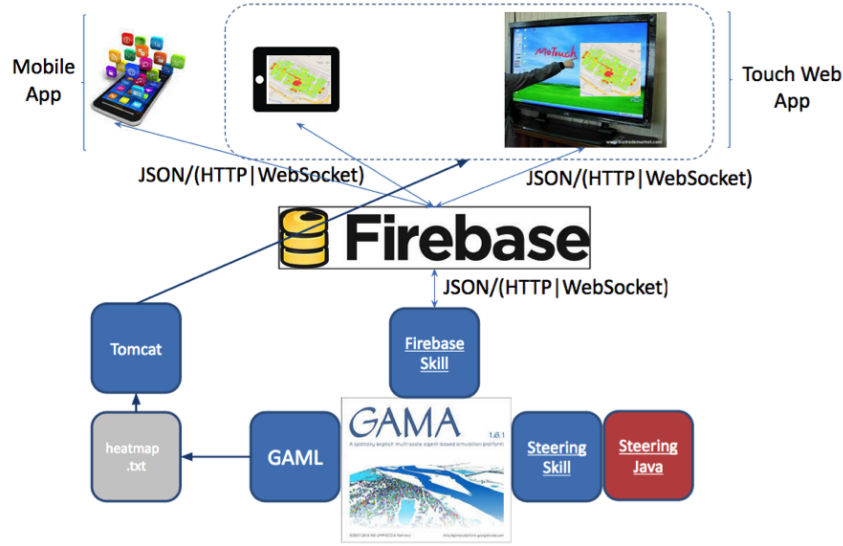


Fig. 4.7 Demonstrative software architecture of FairSteering.

4.2.3 Demo Architecture

A FairSteering demonstrative application was developed with the aim of showing the benefits provided by the crowd steering strategy SP to a typical user. The functionalities actually implemented are:

- heat-map to evaluate the global crowding level across the environment of the fair;
- tailored recommendation for the "best" next booth to visit at the end of each visit;
- freedom of choosing the next booth to visit, once aware of the crowd steering recommendation.

The development of the demo concerns the introduced applicative scenario through the software architecture depicted in Figure 4.7.

The software components of such architecture are:

GAMA simulation Since the demo is conceived to prove the concept of FairSteering to a sole user, presence and the movement of other users in the environment are simulated. Furthermore, the crowd steering component is run internally to the simulation engine;

Firestore⁶ Back-end as a Service useful to speed-up the developments (data exchanges) at such prototypal version of FairSteering;

Mobile App The personal user interface provided to interact with the crowd steering component;

Touch Web App The user interface that should be deployed over public displays across the fair.

4.2.3.1 GAMA simulation

The simulation model is characterized by a set of parameters that *(i)* initialize the environment, *(ii)* configure agents' behavior and *(iii)* regulate agents' interactions with each others and with the environment.

Environment

In the presented proof of concept, the environment is modeled through a graph topology for its higher adherence to a real case scenario, but for sake of simplicity only one building of the topology depicted in Figure 4.2-left is considered.

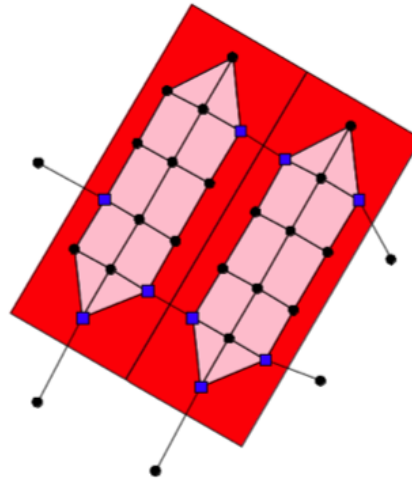


Fig. 4.8 GAMA graph topology modeled for FairSteering Demo.

The environment is described by four GIS (Geographic Information System) layers that contain spatial shapes and attributes to model rooms, booths and hallways of the fair (see Figure 4.8):

Building Polygons layer for the rooms (red rectangles);

Graph Lines layer for hallways and walking paths connecting booths and rooms.

Each line (edge of the graph) contains an Integer attribute that represents an approximate value for the width of the walking path with the aim of modeling the edge capacity in terms of maximum amount of agents hosted in each point of the edge;

Graph Junctions Points layer to fence booths (pink rectangles and triangles).

Each point (node of the graph) contains a Boolean attribute that distinguish the ones in which the agent is able to perceive the local crowding level in the adjacent edges (blue squares) to the ones in which the agent is not used to change

direction (black circles). The 5 points that do not overlap the building layer are the entrances to the rooms.

Bounds Polygon layer that delimits the simulation environment and overlaps all the elements in the other layers (not visible).

Every environmental element is modeled as a GAMA agent with specific tasks in the simulation, through the definition of species. Rooms, Roads and Intersections are created directly from the information contained into the GIS layers, while the definition of Booths required the discovery of contiguous sets of Intersections.

The crowding in the environment is monitored at two different levels: on the Roads and inside Booths. Every 60 simulation steps, each agent in both species count the "moving agents" (visitors) in their proximity (i.e., on the edge or inside the polygon described by contiguous nodes). Crowding data collected are then appended to two output files that will be used to depict the historical and punctual heat-maps on the Front-end applications, as overlays of the environment. The file content for the punctual heat-map is deleted every 3,600 steps.

Moving Agents

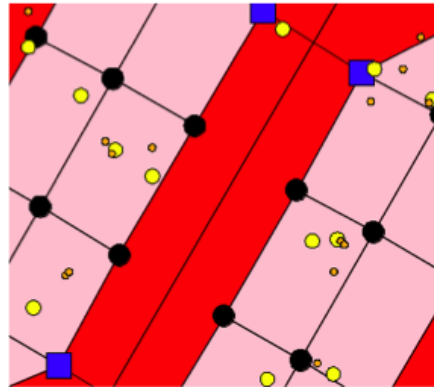


Fig. 4.9 Snippet of a running simulation that shows agents that are visiting the fair in groups, which are leaded by a Leader and composed by [0,3] Followers.

The visitors of the fair are modeled through two different species: Leaders and Followers. This choice is made by taking into accounts that visitors are used to go and visit public exhibitions in groups of individuals. Leaders (yellow circles in Figure 4.9) are the sole that own a complex behavior, while Followers (orange circles) limit their behavior to emulate the movements and the decisions made by their proper Leader. Each Follower is linked to a precise Leader, while each Leader has a number of Follower in the range [0, 3], according to the following probability distribution:

Followers	Probability
0	5%
1	50%
2	35%
3	10%

The agents' arrival distribution to the fair is modeled through the function already presented in 4.2.2.1.

From a spatial perspective, there is a perfect equivalence between the real dimensions of the fair space, and the ones measured by GAMA. Thus, the maximum speed of the agents is identified in the range [1.25 - 1.51] *meters/steps*, as reported by a study appeared in the *Road Engineering Journal* in 1997⁷.

The model is then programmed to adapt dynamically the speed with regards to obstacles and congestions on the walking paths. In particular, for each moving agent, an obstacle is anyone of the other moving agents that is located inside a certain radius of proximity from its current position. The default value for this parameter is set to 2 meters. Each time the visiting agent encounters an obstacle within the radius, its speed decreases with an exponential function, and only when the obstacle is cleared (i.e., the other agent goes out from the radius) it goes back to its maximum value. Agents' speed on the walking paths and visiting time at booths is regulated by the function defined by [23] and mentioned already in 4.2.2.1.

Leaders, in their attributes, own the personal ratings of the booths, which determines their visiting preferences, thus their visit itinerary, and the one of their followers. There are three classes of ratings for the booths: positive, negative and neutral. All the booths in the same class have the same initial value, which is based on the total amount of booths that belong to the class. The sum of the ratings constraints to +100 for the positives and -100 for the negatives (0 for the neutrals).

For every simulated moving agent, the amount of booths in each class is regulated by a normal distribution with mean and standard deviation defined via global parameters. The visit itinerary contains only the booths in the positive class. The visiting time for each booth is computed once the agent arrives at that precise booth, and it is regulated through a function that correlates a normally distributed value of the visiting time for that booth and the current occupancy level of it.

The local perception of crowding determines the capability of each Leader to observe and measure the crowding level on the adjacent edges. From a computational point of view, the goal that to pursue through the local crowd perception is to obtain a ranking of the remaining boots to visit with regards to the crowding level on the edges next to Leader's location. Thus, while a Leader performs this task, the booths are grouped in classes identified by the ID of the edge that must be traversed as first to reach them, departing from Leader's location. The ratings attributed to the classes follow the same rule introduced above in defining the visiting preferences, with the extra rule of creating only two classes for positive and negative booths in the case there are only two adjacent edges to the current node where the Leader is located.

⁷ <http://www.usroads.com/journals/p/rej/9710/re971001.htm>

The third and last (but not least) decisional factor for the Leaders is represented by the recommendation sent by the crowd steering component. Only the booths initially included in the visiting plan of the agent are taken into account by the steering strategy. Furthermore, the decisions taken by all the agents present in the fair in the previous 600 steps are taken into account, since they influence the crowding in the near future, thus they offer an estimation of the crowding level dynamics.

As a reminder, the implemented SP strategy has the multiple objective of minimizing the distance travelled and with the lowest crowding level. Once the matrices of distances and crowding levels are computed, and after a first normalization of the values, the metrics are balanced with two parameters that complement to 1, which represent the impact of the distances' standard deviation on the crowdings' one. Booths are finally grouped in the same positive, negative, neutral classes by following the usual rule for the rating value. It is worth noticing that even if the initial set of booths is composed by the only ones contained in the positive class determined by analyzing agent's visiting preferences, the processing put in place to generate recommendations leads to booths that belong also to the negative and the neutral class.

To wrap up, the visiting of the booths is regulated step-by-step by the evaluation of (i) the euclidean distance between the agent and the remaining booths (the visit itinerary is re-ordered at the end of each visit by increasing distance), and (ii) the decisional process depicted in 4.4, which is made by the Leader of the agents' group at the end of each visit (both local and global crowding level are taken into account).

For (i), every time agents face the eventual situation of choosing between more than one booth with the same rating, the final choice will regard the closest one. For (ii), at the end of each visit, agent's decision will be influenced by the linear combination of all the factors involved in its decisional process. Each Leader is characterized by two individual Double parameters:

Crowding parameter Defined in the range $[-1, +1]$ that describes how likely it will prefer not crowded areas instead of overcrowded ones, and

Steering parameter Defined in $[0, +1]$ that describes its willingness to follow the recommendations sent by the crowd steering component.

The next booth to visit will be the one with the greatest rating, after having applied a linear combination between the rating associated during the creation of the visit itinerary, the aggregation due the local crowding perception, and the recommendation processing on the global crowding level, or, in presence of more than one booth at the top of the ranking, the closest one.

4.2.3.2 Demo User

The user that interacts with the FairSteering Demo has an active role also in the simulation. It is modeled as a particular agent that acquires all the attributes from Leader specie, except the ones needed to generate the visit itinerary and to perform the decisional process. These two tasks are performed directly via the apps by the

user, which can express a rating for each booth, or just select the subset of preferred ones, and autonomously select the next booth to visit. The provided user interface shows the received recommendations generated by the crowd steering component and the heat-map with the different crowding levels.

Even if the interactions of the user happen only via the two Front-end developed, their effects impact also onto the GAMA simulation (that is continuously running) thanks to Firebase. This last component acts as glue between all the elements in the architecture by enabling JSON structured data exchange on a set of REST endpoints. More specifically, Firebase is able to store and synchronize data through a JSON exchange model read and written by a set of heterogeneous platforms. Developers here just need to call Firebase API from client-side and keep Firebase managing any server-side operation.

First of all, the simulation waits to know the departure location and the visit itinerary of the user. Then, it creates a new agent that is localized in the environment at its exact position and it sends the visit itinerary and the departure coordinates to the crowd steering component.

The user is prompted with the recommendation on the apps. It chooses to follow or not that recommendation with a touch event, which corresponds to the chosen booth, and this information is shared with the simulation engine. Now GAMA can move the correspondent agent on the walking path till it reaches the selected destination. It is worth noticing that during the movement, every 5 steps, the simulation writes the current position of the agent on a Firebase endpoint. This lets to show the actual position of the user even on the two Front-ends. Once the visit at the booth is ended, the process repeats starting from a new recommendation, until the list of remaining booths to visit is cleared.

4.2.3.3 Mobile App

The Mobile App is developed for iPhone iOS7+ only and it is able to exchange data with the simulation through Firebase. Once launched, the app asks the user to express visiting preferences for the booths of the fair through a 5-star rating system (see Figure 4.10-left). When the user is ready, she can click on a button to begin her visit. The initial location of the user has been mocked in this prototypal version. At the beginning of each visit, the app depicts the fair map as an overlay on the Apple map view. The booths follow these coloring rules:

- Red** Generic booth, not in the visit itinerary;
- Yellow** Booth in the visit itinerary, not yet visited;
- Orange** Booth in the visit itinerary, visited already;
- Blue** Booth recommended by the crowd steering component.

Once the visiting recommendation is sent, a push notification alerts the user. The notification shows a message inviting the user to visit the suggested booth (see Figure 4.10-center), and two button to accept or decline the recommendation. In case the user accepts the recommendation, the app sends the booth coordinates to the

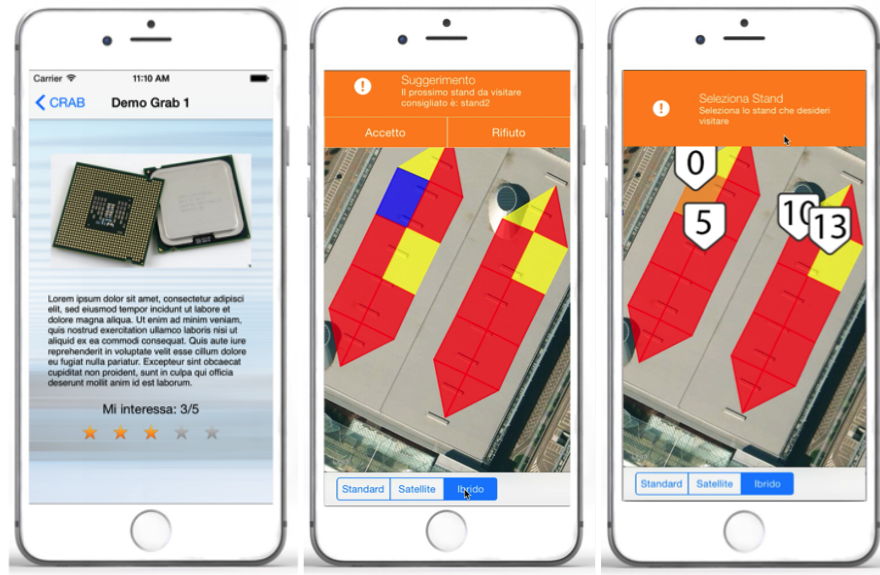


Fig. 4.10 Three screens of the Mobile App showing the rating of a booth (**left**), and the map of the fair shown with the recommended booth notification (**center**) and with the remaining booths to visit highlights (**right**).

simulation. While, if the user declines to visit the recommended booth, the remaining booths to visit will be highlighted to facilitate the selection of the next booth to visit (see Figure 4.10-right).

The Mobile App offers a heat-map view of the fair as well.

4.2.3.4 Touch Web App

Despite the actual aim of a Web App in a real case scenario is the one of showing the real-time crowding level across the environment on the available public displays, this Front-end HTML5/JS application owns a more extended purpose in the demonstration. An exemplar view is shown in Figure 4.11-left. It actually covers a triple role:

- it lets the creation of the agent correspondent to the user, by setting its initial attributes (i.e., departure and visit itinerary);
- it allows the evaluation of the global crowding level through a heat-map updated every 10 seconds;
- it provides a user interface with whom they can view the recommendations coming from the crowd steering component and they can choose which stand to visit next.

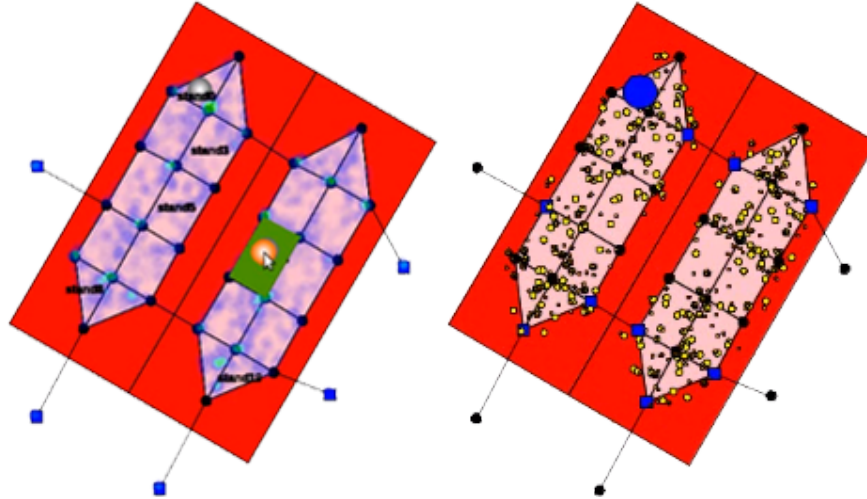


Fig. 4.11 Touch Web App (left) and GAMA simulation (right) synchronized views. In the Front-end, the booths' background is filled with the real-time crowding heat-map, the remaining booths are highlighted through labels showing their name, the recommended booth is colored in green, the grey circle is put where the user is currently located, while the orange circle is the place-mark for the next booth chosen to visit (the same suggested by the recommendation). In the simulation view, the big blue circle is located exactly where the user is, and agents are almost even distributed across all the environment.

4.2.3.5 Overall Perspective

Initially the two Front-ends were conceived to be complementary during the Demo setup. Thus, their functionalities were supposed to be different. In Figure 4.12 the Sequence Diagram of the activities involved on the entire software architecture is shown. The operations within the red border are repeated after each visiting of a booth.

However, during the implementation of data exchange operations with Firebase, some small changes raised on the Sequence Diagram to let the Demo setup from both the Front-ends:

- the initial location and the next destination of the user can be selected also via the Mobile App (and not only via the Touch Web App);
- the visit itinerary can be composed also via the Touch Web App (and not only via the Mobile App).

4.3 Outdoor Real Case Study: TogetThere

Steering behavior of crowds and individuals can be performed also in outdoor environments. TogetThere is a mobility service provided to tourists with the aim of

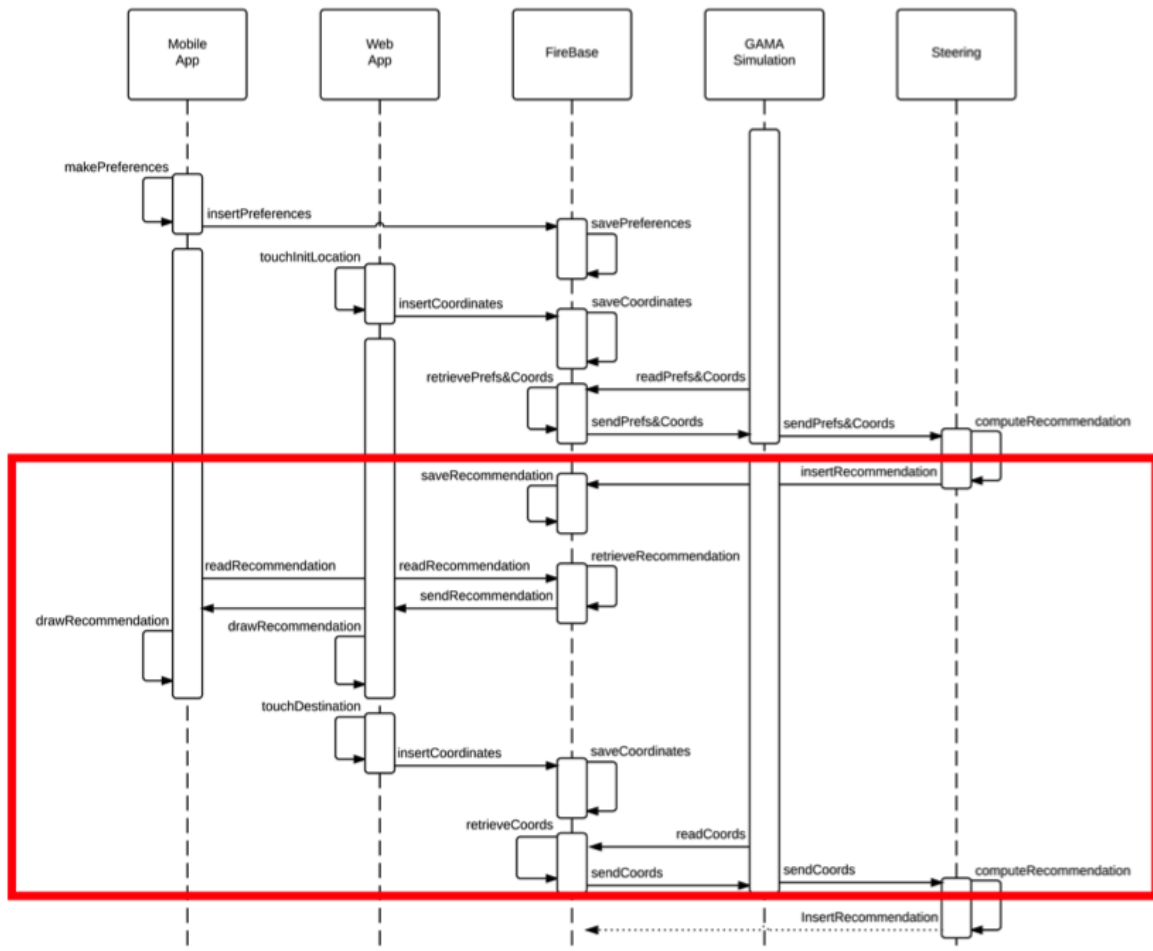


Fig. 4.12 Sequence Diagram of the activities between the components of the Demo architecture. The red rectangle highlights the one repeated after each visiting of a booth till there are remaining booths to be visited.

enhancing the activity planning to better enjoy their visits across a city. Users will be suggested with an activity plan by gathering their visiting demand in terms of Points of Interest (POIs), without affecting their visiting preferences, and preventing overcrowded situations both along the streets and at the selected POIs.

The proof of concept of the service has been developed for the historical city of Venice, in Italy, which has the peculiarity of being a vehicle-free city, except for the water buses that travel along its canals. Nevertheless, the same service can be deployed in other cities around the globe with the integration of a journey planner that takes into account other means of transport such as bicycles, cars, taxis, buses and trains.

There are averagely 60 thousands tourists per day that move across an area that is smaller than 20 Km², but during most popular events, such as the carnival celebration, more than 100-200 thousands people reach Venice. Pedestrian paths, squares, cathedrals, historical palaces, museums, cafes and restaurants get quickly overcrowded in those days and police interventions to disperse people is often needed. Furthermore, Venice already established a professional figure to steer people to less crowded areas⁸.

San Marco's square is a typical example of an attraction that everyone wants to visit in Venice. It is characterized by entrances linked to only three main paths and a total walkable and standing area of about 1.4 Km² where people are used to stay averagely 15-30 minutes before/after the visit to the adjacent San Marco's cathedral. A better distribution of (i) the temporal arrival rate of tourists in such a popular POI, and (ii) their spatial flow on the all available paths, even the less popular ones, is pursued.

Basically, the goal here is to influence the social phenomena of crowds by gathering the visiting demand from tourists and spatio-temporal scheduling it according to the expected crowding level on the paths and at the POIs. Business activities located on less known paths can take part as direct stakeholders of the service. One can think to a business model that incentivizes people to follow the recommended activity plan by providing them with meal deals and special offers at certain shops located on the suggested paths. On the other hand, business activities that are not used to accommodate a large amount of customers because of their outlying positions can increase their visibility. For further details on local incentives, see Section 4.3.2.

At the beginning of the day, the tourist provides a list of POIs to visit during the day. Typically a small number of attractions (e.g., 5) can be visited in one day. Other inputs user can include are parameters for journey planning, such as preferred transport modes and quotas on walking, cycling and number of legs in a journey. The user should be able to choose a quality criterion for the visit itinerary, such as minimizing the time spent in a queue, or the congestion experienced at the POIs visited.

The task is to provide the user with an order in which the POIs should be visited to better enjoy the visit. Furthermore, traveling directions should be provided from one location to the next in order to avoid popular and most crowded roads. The itinerary should be optimized to the crowd-based quality criterion at hand.

In the use case, outdoor crowding level computations are performed by exploiting complimentary data sources provided by Telecom Italia, an Italian telecommunication service provider, through smart phones, mobile apps and social medias. This data sources were made available in the Telecom Italia Big Data Challenge 2015⁹, in which we have competed (see Section 4.3.1.1).

The datasets were sparse, aggregated and related to a 60-day period. These properties make them acceptable to develop a prototypal version of the service, but they

⁸ Venice's official intruder: <http://www.comune.venezia.it/flex/cm/pages/ServeBLOB.php/L/IT/IDPagina/2595>

⁹ <http://www.telecomitalia.com/tit/it/innovazione/big-data-challenge-2015.html>

are not suitable for a production deployment of it. TogetThere is conceived to be fully fed with localization, density, and occupancy data from the sensing technologies embedded on users' smart phones (e.g., GPS sensor) and from the ICT services deployed across the city (e.g., Wi-Fi hotspots loggings, museum ticketing systems, CCTV and computer vision algorithms [3, 41, 126]).

The benefits provided by the service will be directed to a wide set of stakeholders:

- *individuals*, which will face less annoying and frustrating situations;
- *urban authorities*, which will have an even distributed crowd to handle in case of emergencies and evacuations;
- *business activities*, which will have a greater amount of visits on secondary streets and a better quality of service perceived on primary ones.

TogetThere is conceived as an Activity Planner for tourists that can help cities to better understand crowd dynamics and business activities to boost their revenues.

The user first selects a set of preferred activities, then she/he is provided with the visiting sequence that optimize the city-wide crowding level. Crowding levels are monitored and updated in real-time. Thus, at the end of each visit, the visiting sequence with the remaining touring activities is re-computed with the new crowding level estimation. In particular, the exploited methodology consists in:

- computing the crowding levels related to the visiting sequence by taking into account crowds on the followed paths and at the POIs, at an expected time;
- updating the projected crowding levels, after every plan computation, along the whole day, on the interested paths and POIs;
- tracking the user while she is visiting the activities, thus collecting implicit and explicit feedbacks;
- checking for eventual updates that improve the activity plan at the end of each visit;
- collecting the overall feedback of the perceived crowding level during the whole visiting experience and adapting the dynamic threshold that estimates the feeling of overcrowding.

4.3.1 PETRA project - Venice Use Case

The PETRA project is a EU FP7 funded project, entitled "PErsonal TRansport Advisor: an integrated platform of mobility patterns for Smart Cities to enable demand-adaptive transportation systems"¹⁰. Research partners in the consortium work on achieving a next generation personal travel advisory platform with demonstrative results in several cities. A use case on Venice mobility, which is conceived to support pedestrian mobility across the city, is included in the expected outcomes of PETRA. Thus, TogetThere is perfectly aligned with the goals of the project, and

¹⁰ <http://www.petraproject.eu>

it could be orchestrated with the modules of its software architecture (depicted in Figure 4.13).

The collaboration in PETRA project was made possible by a Joint Agreement signed with the *Smarter Urban Dynamics* team at IBM Research Lab. Ireland¹¹, which is one of the leading partners of the consortium.

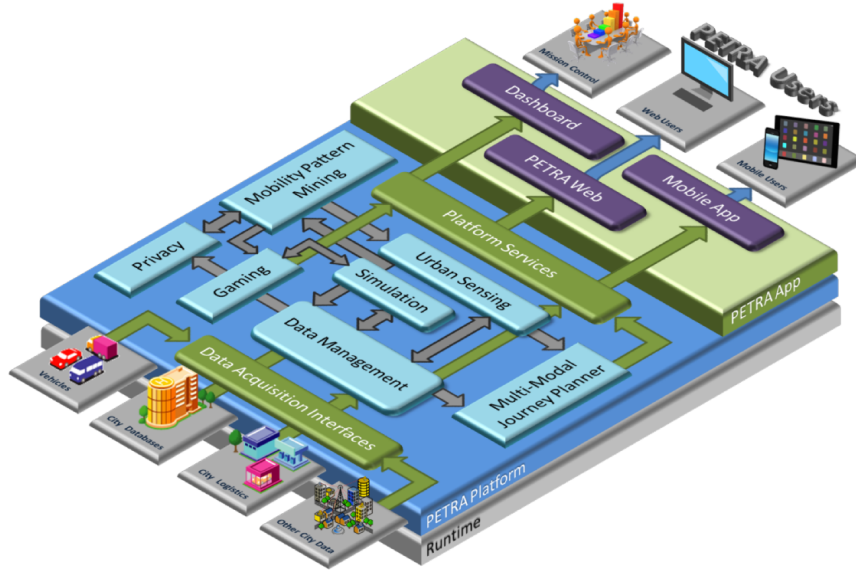


Fig. 4.13 The PETRA software architecture. The development of the Venice use case actually involves different modules: starting from the *Data Acquisition Interfaces*, passing through the *Data Management* module, the *Multi-Modal Journey Planner*, the *Urban Sensing* module, and the *Simulation* service, till reaching several *User Interfaces*.

In the collaboration with IBM Research, the assessment of the effectiveness of a crowd steering strategy, for the outdoor scenario, which aims to prevent overcrowded situations by balancing the distribution of tourists across the city of Venice, without affecting their visiting preferences, and by increasing the aggregate shops visibility across the city (by considering the limited capacity of a business activity as defined in [138]), is pursued. With regards to the taxonomy surveyed in Section 4.1.3, a *global info - global goal - unicast* steering strategy was evaluated.

The metrics used to assess the effectiveness of the crowd steering strategy are:

- Visiting time and walking distance covered;
- Business activities visibility.

The idea is that by avoiding overcrowded areas, streets, and POIs, even if the total covered walking distance can be slightly greater than the one offered by the shortest

¹¹ http://researcher.watson.ibm.com/researcher/view_group.php?id=2522

path, the total visiting time could decrease. Furthermore, the shops visibility on not popular streets can increase their visibility, and thus potentially their profits, while the ones located in main streets or in overcrowded areas would not be affected negatively, since their serving capacity is highly saturated. The main goal is to balance the crowd across the city.

With regard to the typical tourist visiting approaches, as baselines, one can consider to compare the crowd steering strategy with:

- *Shortest path approach.* Selected activities are sorted initially to minimize the total distance covered during the visit itinerary;
- *Greedy approach.* Next attraction in the sequence is the closest to the one just visited.

4.3.1.1 Telecom Italia Big Data Challenge 2015

The theme of the 2015 contest was "Big Data for Competitiveness Boost: how can data help a country grow and become more competitive?". The contest required to exploit a bunch of data collected in the main Italian cities in order to boost their competitiveness. We took part to the Industrial track with TogetThere and we won the first prize of the competition¹².

The data made available was collected from 7 main cities in Italy over a two months period. All the available datasets collected in Venice (Grid Topology, Telecommunications, Census, SocialPulse, Business Activities) were exploited in modeling the initial crowding levels and in measuring the impact of the activity planner on business activities. The full stack architecture of TogetThere, till the Front-end Mobile App, was finally implemented and submitted to the contest.

4.3.2 Incentive Mechanism

Consider visiting a crowded POI, such as San Marco's square. In such place, many tourists spend significantly more time than they actually need to enjoy the place and to retain good memories about it. This creates an unnecessarily high level of crowding, which negatively impacts multiple stakeholders involved in the tourism domain.

First off, the experience of the tourists will suffer because of the crowding. Secondly, business activities and other POIs in the area, or in other parts of the city, may suffer from having fewer visitors than their capacity. An approach to perform tourist load balancing, taking the surplus of tourists from a overcrowded area and encouraging them to spread around to other less crowded POIs, was proposed.

¹² <http://www.telecomitalia.com/content/tiportal/en/bigdatachallenge/news-social/tim-big-data-challenge-2015-finalisti/togetThere-leveraging-Big-Data-mobile-information-smarter-tourism.html>

The idea, based on local incentives, works as follows. Imagine that tourists exploit the envisioned service, which could be provided to tourists in partnership with the municipality of Venice, for instance. Through GPS measuring, TogetThere app can detect that a tourist is spending her time at a crowded POI. After the time spent exceeds a given threshold, set to allow enough time to enjoy the POI, a tourist receives an incentive on the mobile app, such as an offer from one or several POIs that want to attract visitors. This would encourage the tourist to leave the crowded place. For example, the offer could be a lunch deal together with a reserved table at a nearby cafeteria, if this happens around lunch time.

The mobile app keeps monitoring the GPS location, to infer whether the tourist leaves the area soon or they continue to stay there for a significant amount of time. In the latter case, the offer is lost. Otherwise, the offer remains valid, and the mobile app can provide detailed navigation instructions to reach the POI that has provided the offer. As mentioned earlier, the offers target tourists located at the crowded POI, and whose time spent at the POI exceeded a given threshold. This targeting is key to help reduce the crowding at the POI. Otherwise, if the offers were simply applicable to everyone, the desired effect of reducing crowding would be greatly diminished.

To date, this might be a novel idea. Focused on the crowding of tourist attractions, the approach addresses a real problem, pointed out, for instance, by representatives of the municipality of Venice that are partnering PETRA project. The idea can be implemented using current technology (e.g., GPS, smart phones) and its discoverability is trivially easy.

The current implementation of TogetThere does not provide local incentives to users, but this feature will be taken into account for sure in the future.

4.3.3 Experiments

Simulated users requests were ran to measure the impacts of the crowd steering strategy on tourists and business activities, in an average crowding day scenario. In particular, the walked distance and time between the ones expected with the crowd steering recommendations will be compared with the ones offered by the baselines, and the amount of tourists that pass close by a particular business activity (until the saturation of its serving rate) will be compared to the one offered by the baselines.

4.3.3.1 Simulation Setup

To evaluate the impact of the proposed solution, a simulation that combines real data extracted from the datasets provided by Telecom Italia Big Data Challenge and synthetic active tourists was devised. 60,000 tourists visiting the various attractions in Venice in one day, which corresponds to a peak day in Venice, were simulated. The number of attractions per visit has been defined on statistics provided by the city, and it is normally distributed. Visiting plans are defined on a weighted list of

attractions, which is based on the *Importance* tag linked in *Open Street Map* (OSM) entities retrieved¹³, and *Social Pulse*¹⁴ data.

The average visiting time was estimated with regard to the kind of each attraction (e.g., historical site, monument, museum, park, religious site, various attraction) in the itinerary.

Tourists are provided with visiting recommendations computed step-by-step. The approach involves a crowd-based optimization strategy that provides a visit itinerary with the remaining attractions, which can potentially be updated after each visit. The chosen baseline, to compare the crowd steering strategy with, was a pure heuristic-based shortest path (TSP solver).

A ratio of tourists that follow the recommendations was not set. The simulation was ran twice: during the first run every tourist visited Venice with the TSP path; in the second run everyone was supposed to be supported by TogetThere, thus following the less crowded path.

4.3.3.2 Simulation Results

The impacts on tourists offered by the applied strategy show that 56.9% of the itineraries (each dot in Figure 4.14) returned by the crowd balancing method are faster than the ones returned by the shortest path baseline. Only 4.9% are slower, due to the heuristic used. On average, plans are 0.93% longer, but 3.35% faster than the plans returned by the baseline.

In terms of shop visibility increase, given the tourist visit itinerary, the business activities that are traversed in a time-slot (15 minutes) were evaluated, thus the amount of tourist impressions near the shops was counted. Only an overall value as the aggregate impressions resulted by the 60,000 itinerary requests along the entire day was computed (see Figure 4.15). Using the crowd steering strategy the shop visibility increases of 4.5%. Considering the limited capacity of a business activity as defined in [138], a potential business impact increases of at least 3.8% is estimated.

Even if the proposed crowd balancing approach is still rough and in an initial phase, the obtained results are encouraging. The effectiveness of the crowd steering strategy was assessed, and the crowd was more balanced across the city. Nevertheless, choosing an heuristic-based baseline consists in a big limitation: some crowd-based paths result to be shorter (in distance) than the TSP-solved ones. This aspect should be fixed by implementing a mathematical model that computes the actual shortest paths for the visit itineraries. However, such first results obtained, let's envision the potentiality of actually deploying TogetThere in Venice.

¹³ XML entities retrieved through *Nominatim* API: <https://nominatim.openstreetmap.org>

¹⁴ Social Pulse dataset was provided in the Telecom Italia Big Data Challenge. It consists of a popularity index for the POIs. It is correlated to the social presence of the POI itself on social medias and networks (e.g., the rate of Tweets that contains the name of the POI in a periodic Tweet-crawling).



Fig. 4.14 Impacts on tourists. Each dot is an itinerary returned by the crowd balancing solution and a solution based on shortest path (for the same query): on the X axis the difference in walking distance (positive means the solution of the crowd balancing is longer) is shown; on the Y axis the difference in walking time (negative means the solution of the crowd balancing is faster and gives more free time) is shown.

4.3.4 Architecture

A TogetThere application was developed with regards to the introduced scenario through the software architecture depicted in Figure 4.16.

The software components of such architecture are:

- *Activity Planner*. The engine of the whole architecture. It is constantly fueled by the congestion data with the aim of computing visiting plans;
- *Congestion Module*. The Back-end service that computes up-to-date crowding levels on paths and POIs;

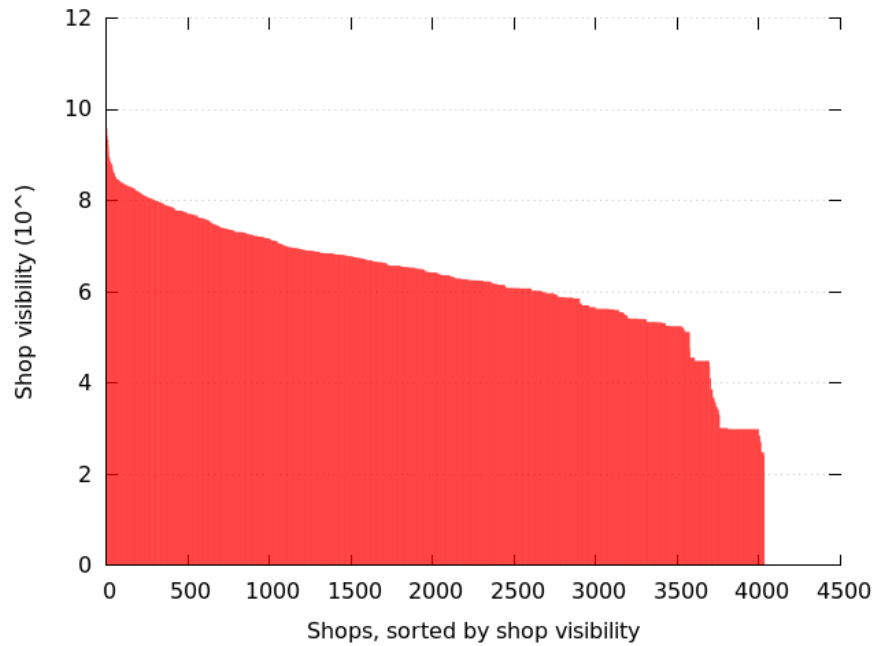


Fig. 4.15 Impacts on business activities. On the X axis the shop incremental ID of the shops (on a total of more than 4,000 shops) is reported, while on Y the amount of tourists in proximity of the shop location along the entire day is shown (it means that, in highly business-populated areas, each tourist can affect the visibility of more than one shop per time-slot, and she can affect the visibility of the same shop in more than one time-slot).

- *DOCIT Journey Planner*. The journey planner of PETRA is linked to TogetThere to provide traveling time with uncertainty [34] to the Activity Planner;
- *Mongo DB instance*. The sink where crowding levels and other data are stored;
- *Mobile App*. The Front-end for tourists where activity plans are requested and retrieved;
- *City Dashboard*. The Front-end for urban authorities where to monitor how the crowd is spread across the city.

Information flows across the architecture by following the following steps:

- The user inserts her visiting preferences (in terms of departure time and location, arrival location, and crowding preference), and she selects a list of activities across the city, with the Mobile App;
- The Activity Planner discovers the least crowded visiting plan by combining: the user's visiting preferences, the traveling times provided by DOCIT Journey Planner, and the crowding levels computed by the Congestion Module;
- The Congestion Module updates the expected crowding levels due to the new visiting plan computed;

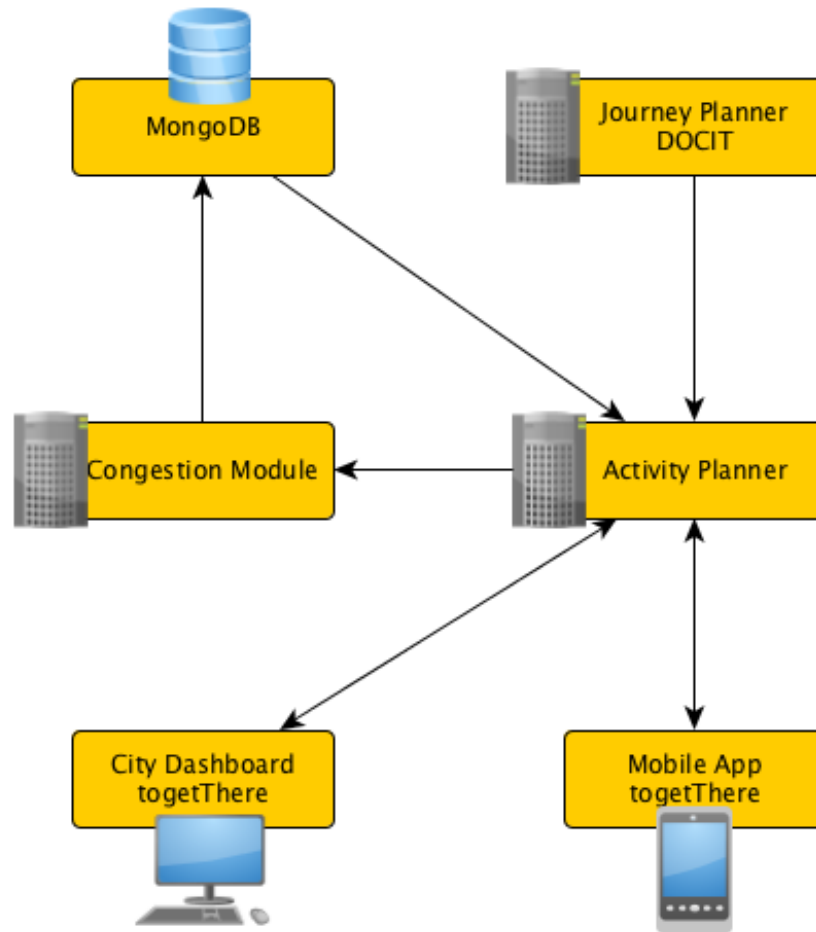


Fig. 4.16 TogetThere software architecture.

- The user provides explicit and implicit feedbacks during her visiting that concur to update the future crowding levels estimation;
- At any moment, urban authorities (such as mobility managers) can monitor the real-time crowding statistics on the City Dashboard.

Activity Planner

As already introduced above, the Activity Planner performs the computation of the less crowded (that results in the fastest) visit itinerary of the selected activities, without affecting the selection, but only the visiting order. Under the assumption that the

number of POIs to visit in one day is typically small (e.g., 5), even a brute-force enumeration of possible orderings would be feasible. Pruning could be employed to reduce the number of considered orderings to a subset. For each candidate ordering, one or several plans can be computed.

Initially, the brand new less crowded itinerary is shaped with the following procedure:

- Given the departure location and time of the tourist and her expected arrival location, the TSP problem among all the selected POIs is solved by taking into account the traveling time affected by the traveling distance only. The set of the first $K - TSP$ solutions (the K-shortest/K-fastest paths) is stored for further computations.
- The oriented graph of the K-TSP solutions¹⁵, with the departure location as the root node and the arrival location as the target node, is built. The weights on the edges represent the dynamic traveling time between any pair of POIs, which is computed by taking into account the crowding level on the walking path that connects them and the occupancies at the arrival's POI. Expected arrival times at POIs and their expected visiting times (that are affected by the POI's occupancy) are crucial to the selection of the right, time-based, crowding level.
- The graph is traversed with an instance of the A* algorithm, which builds the fastest sequence of activities, which is the less crowded one.

Afterwards, the recommended visit itinerary is provided to the tourist that begins its journey. At the end of each visit, a re-planning request is sent to the Activity Planner. In these cases, it repeats the procedure put in place for the shaping of the initial itinerary by considering the remaining POIs to visit, and it eventually updates the recommendation if expected crowding levels or occupancies during the day have changed due to the requests received since the last computation performed for the same tourist. Furthermore, crowding levels and occupancies are updated following the changes between the newly computed itinerary and the previous one.

Congestion Module

The Congestion Module is in charge of computing and storing (on the Mongo DB instance) crowding levels and occupancies. The benefit of a lower congestion experience applies not only to the tourist at hand, but also to other tourists, in a less direct way. This latter effect occurs more visibly when many tourists opt for itineraries that minimize both the encountered crowding levels and occupancies at POIs, which generate long waiting queues. Tourists that will choose to avoid crowded areas will not contribute to making those areas even more crowded. This can save time also to the tourists that are not covered by the service. Furthermore, as long queues

¹⁵ The choice of solving the $K - TSP$ problem was driven by the need of reducing the complexity of solving the A* algorithm. This approach lets to prune the full graph connecting the full set of selected POIs, which would have $O(N!)$ complexity (where N is the amount of selected POIs), to a graph with traversing complexity of $O(N^K)$

are closely correlated to big crowds, avoiding long queues also results in avoiding crowded areas.

To determine the initial crowding level, first of all, Venice area is divided on spatial cells by following the provided grid topology in the Telecom Big Data Challenge. For each cell, the walkable area is estimated by taking into account the areas of the overlapping streets to the cell itself (as a GIS line layer, extracted via OSM, in which each line has a width attribute chosen accordingly to the kind of street - Venice streets are 4 meters wide averagely).

Concerning the temporal resolution of the data used, the one provided in Telecom's datasets was kept: 15 minutes time-slots. For each time-slot and cell, then, the amount of outbound phone-calls made by users that do not belong to Venice residential population was aggregated (by exploiting the "Demographic Callouts" dataset from the Challenge - containing demographic statistics of users that are placing a call), and the maximum reported value along the day (one per cell) was stored. The initial congestion in a cell, in a certain time-slot, is computed as the ratio of the number of outbound phone calls in the time-slot over the maximum value reported in that cell.

The updating procedure begins with the loading of the "Census" dataset from the Challenge (containing statistic on residential population, provided by the Italian institute of STATistics - ISTAT ¹⁶), the average amount of tourists given by the municipality of Venice, and Telecom Italia market share¹⁷.

For the pair (cell, time-slot) that needs an updated crowding level, the aggregated amount of outbound phone calls from "Demographic Callouts" was split over the number of outbound phone calls contained in the "Telecommunication Callouts" dataset (collecting the amount of outbound calls events in the cell and time-slot) obtaining the probability of placing a call over Telecom Italia network, then the probability of placing a call was computed by estimating the calling population (through the Telecom Italia market share), and, finally, the current crowding level was increased/decreased according to a value which is function of the probability of placing a call.

Concerning the occupancies computation, the initial value is set to 0, and the updates are obtained by adding/subtracting 1 to the current value, when a tourist reaches/leaves the POI.

The dynamics of traveling and visiting times are regulated with the speed-density relation described in [23].

In particular, for the *Dynamic Traveling Time* ($\widetilde{TT}$) computation between any pair of POIs, we:

- Initialize $\gamma = 0.245$ (leisure/shopping scenario, from [23]);
- Initialize *max_congestion* as the maximum crowding level currently present among all the cells of the Telecom Italia grid;

¹⁶ <http://www.istat.it/>

¹⁷ It is about 35% at the time of writing.

- Set *current_congestion* as the average crowding level among the ones on the cells traversed by the path connecting the pair of POIs;
- Compute *Dynamic Traveling Time* as $\widetilde{TT} = TT \cdot (1 - e^{\gamma \cdot (\frac{\text{max_congestion}}{\text{current_congestion}} - 1)})$, where *TT* is the Static Traveling Time between the POIs, affected by the distance only.

While, for the *Dynamic Visiting Time* ($\widetilde{VT}$), we:

- Initialize $\gamma = 0.245$;
- Initialize *max_congestion* as the maximum occupancy currently present among all the POIs;
- Set *current_congestion* as the amount of people currently localized at the POI;
- Compute *Dynamic Visiting Time* as $\widetilde{VT} = VT \cdot (1 - e^{\gamma \cdot (\frac{\text{max_congestion}}{\text{current_congestion}} - 1)})$, where *VT* is the Static Visiting Time of the POIs, in a not crowded condition.

DOCIT Journey Planner

DOCIT is able to perform global optimization across transportation modes, and to provide users with a set of alternative journeys identified by a probability value of arriving at destination at the expected time. Figure 4.17 depicts an exemplar case. The Multi-modal journey planner of PETRA is not used in the current implementation of TogetThere since the uncertainty linked to the travel time of a journey in Venice is only related to the crowding levels on the streets, and not to the probability of missing a connection with a mean of transportation. However, the module has been added to the architecture since it can be exploited in future deployment of TogetThere on different cities.

Mobile App

TogetThere Mobile App is a multi-platform HTML5/JS Front-end developed with *AngularJS* and *Apache Cordova* frameworks. It is able to exchange data with the Activity Planner, through the exposed API, and it is the tool exploited by tourists to setup their journey preferences, select the activities they want to visit, request a crowd-based visit itinerary, and receive the recommended one.

Once launched, the app asks the user to fill a visiting preference form with her departure location and time, desired arrival location, the set of activities she wants to visit, and her crowding preference (see Figure 4.18-left). In particular, on selecting the activities, user can choose among a set of them split in 6 categories: historical sites, monuments, museums, parks, religious sites, various attractions.

Each activity is rated according to a popularity index inferred by the *Social Pulse* dataset provided by Telecom Italia Big Data Challenge (see Figure 4.18-right). Furthermore, crowding preferences slider let the user to choose between 4 different crowding conditions (i.e., fully empty, mainly empty, mainly crowded, fully crowded), in case she has a propensity or an adversity to crowded areas.

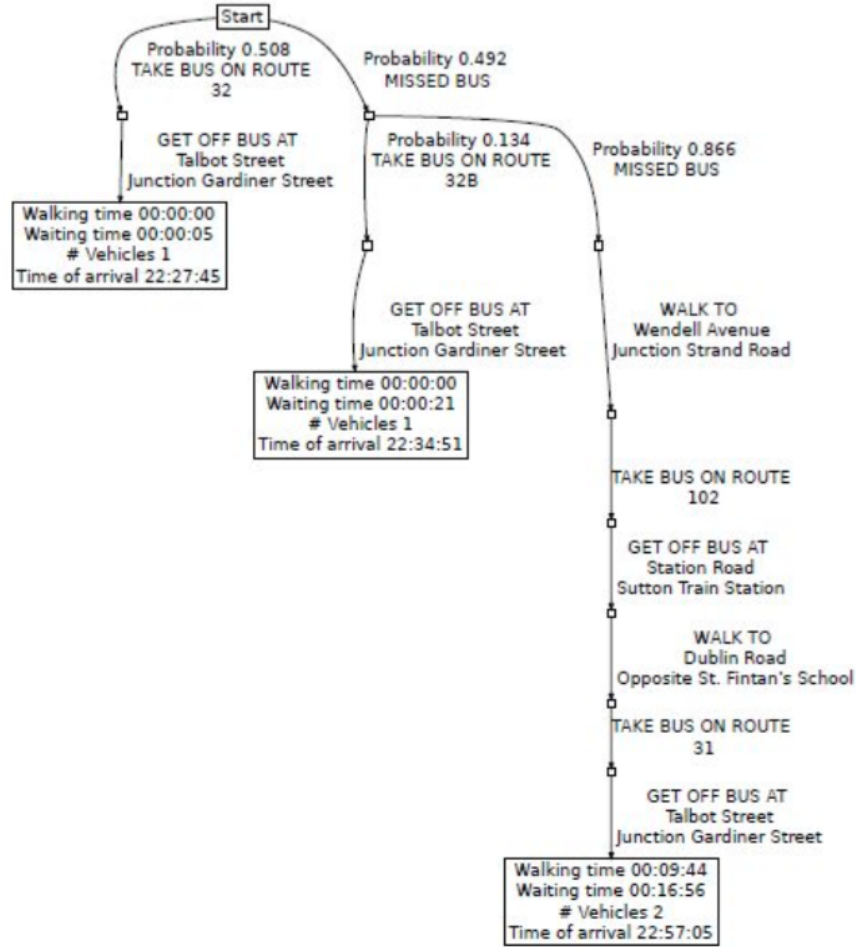


Fig. 4.17 A DOCIT exemplar case of multi-modal uncertain journeys.

Once ready, the user can tap on the *Get Visiting Plan* button, and a planning request is sent to the Activity Planner. The user is then recommended with the less crowded itinerary computed with the current crowding projections along the day (see Figure 4.19-left). At her arrival to each POI, she is asked to give a feedback for the crowd encountered on her way there (see Figure 4.19-right). At the end of the visit, a planning request containing the remaining activities is sent to the Activity Planner, and the process repeats till the end. In the last screen of the app, user is asked to leave a feedback related to how she liked the recommendations and how likely her crowding expectations were satisfied.

A demonstrative video of the Mobile App is available online¹⁸.

¹⁸ <https://ibm.biz/BdX6FD>

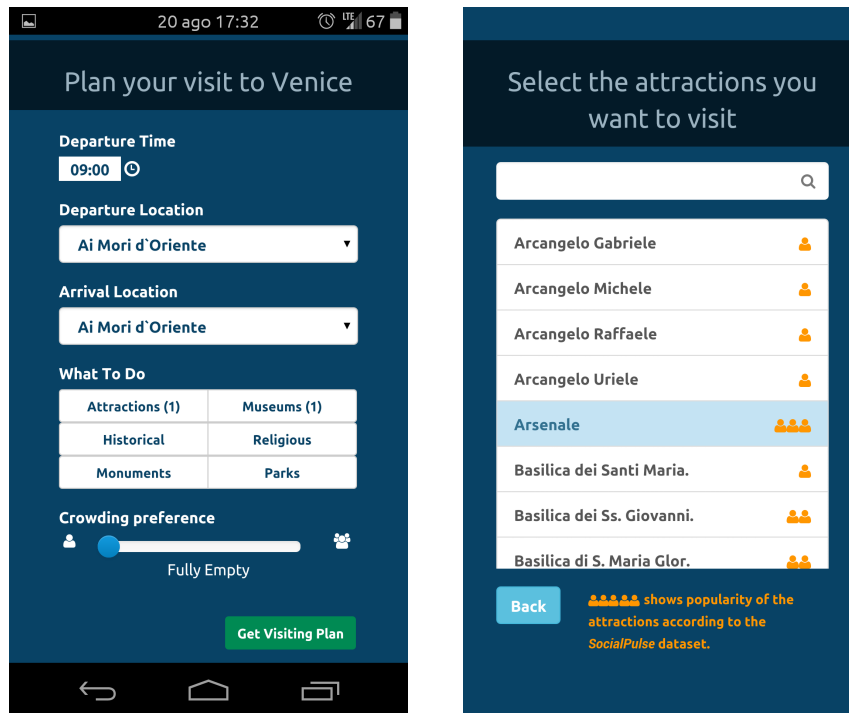


Fig. 4.18 Two screens of the Mobile App showing the visit setup form (**left**), and the activities' selection view (**right**).

City Dashboard

TogetThere comes out along with a second Front-end conceived as a dashboard for city's urban managers. The platform would be released as a Web Application showing an heat-map fashioned crowding map using Voronoi Tessellation of space, and a set of global statistics indicators such as road congestion, crowd balancing, and shop visibility coverage (see Figure 4.20). Furthermore, the tourists' interactions with TogetThere will be logged in real-time, as depicted in the demonstrative video mentioned earlier.

Another feature that should be enabled in the City Dashboard would be the managing of local incentives. Business owners should have access to a proper market place in which they can publish deals to be enabled in special days and hours. City managers should then setup automatisms on enabling/disabling certain offers according to the crowding levels reported in surrounding areas.

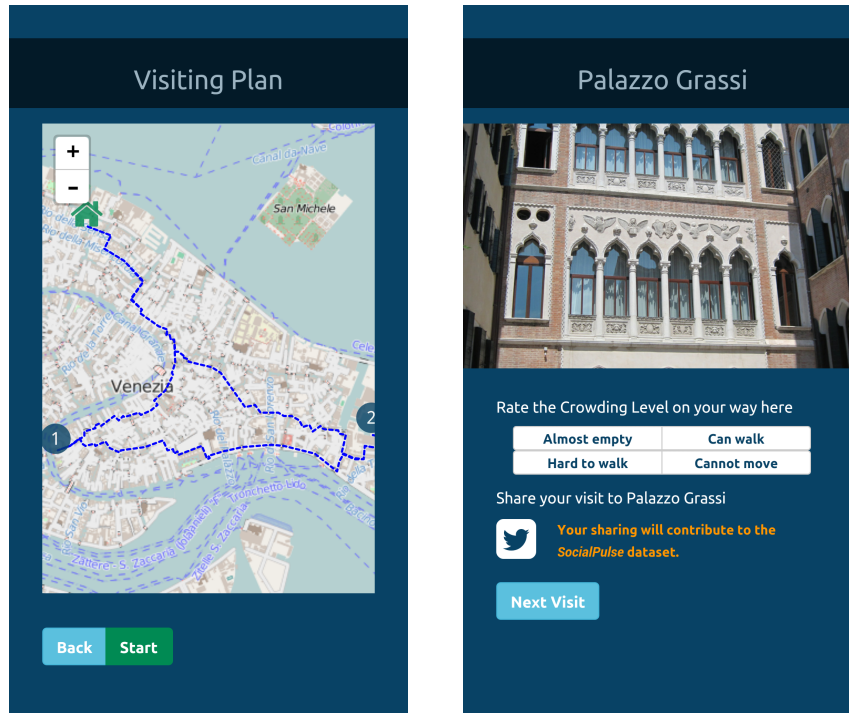


Fig. 4.19 Two screens of the Mobile App showing the recommended visiting plan by the Activity Planner (**left**), and the view showed when the next activity to visit is reached (**right**).

4.4 State of The Art

In the area of collective robotics [18], a large amount of work has been devoted to the problem of engineering to orchestrate the movements of a large number of robots. The kinds of collective movements that such approaches aim at orchestrating depends on the specific missions the robot swarm is deployed for. Accordingly, some of the identified strategies for crowd steering resemble those for robot swarms [28]. Yet, robots miss an individual goal, and move with a global cooperative goal in mind.

Dynamic load balancing [2] – a topic extensively studied over the past 20 years in the area of parallel and distributed high-performance computing – somehow relates to the study of crowd steering. Dynamic load balancing concerns re-allocating at runtime the processes composing the parallel computation so as to fairly balance the computational load on the processors, and eventually speed up the overall computation.

In load balancing, some of the strategies upon which to base the decisions on which processes to move on which processors can somehow resemble the discussed strategies for crowd steering [44], and so does the goal of balancing the overall

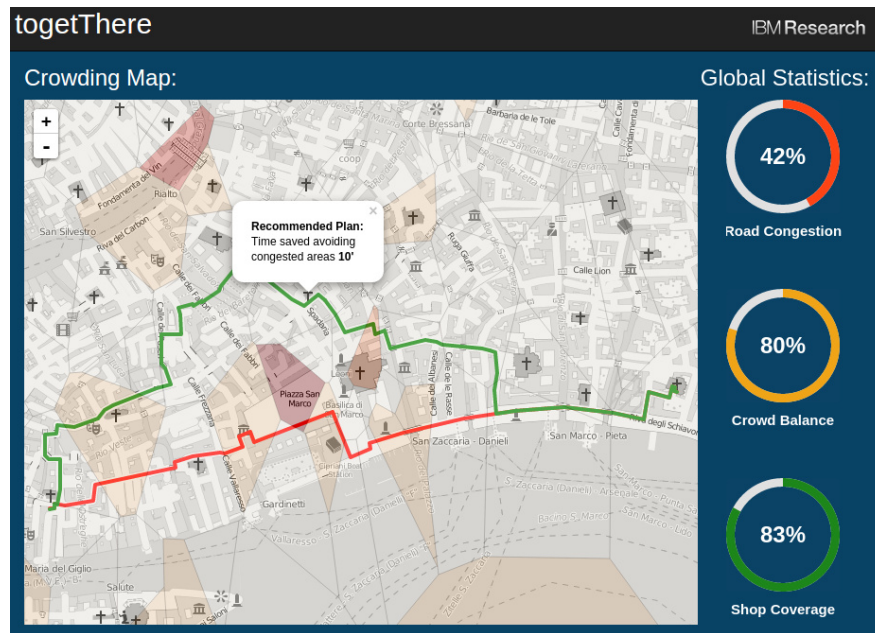


Fig. 4.20 The City Dashboard lets to monitor in real-time the crowd spread in the city and the planning requests generated by tourists. The map shows an example of a recommended plan from Santa Maria del Giglio to San Giovanni Battista in Bragora taking into account the crowding levels on the path.

distribution of processes. However, the approach are fundamentally different in that, in load balancing, movement of the processes is a mean and can be strictly enforced on processes, in crowd steering movement is also the goal of the individuals and, in most of the cases, the strategies cannot be strictly enforced on all individuals.

Most of past studies in indoor scenarios, relate to crowd dynamics (i.e., have focussed on analyzing and simulating mobility patterns and behaviors in an environment), and at designing the environment (e.g., changing size of corridors and moving doors so as to affect paths [1, 71]) so as to enforce specific behaviors. In most of the cases, the key goals were to avoid overcrowded situations, prevent the formation of waves of people moving in opposite directions, and more in general ensure safety of public environments and roads. This kinds of studies involve an accurate modeling and simulation of pedestrians' behaviors, and indeed a large amount of recent work deals with the issue of defining realistic models for such simulations [106, 107, 141]. Such simulation approaches, though, have different goals than the ones pursued in this work. In fact, they are mostly aimed at identifying how the behavior of individuals in an environment can possibly lead to the emergence of peculiar crowd phenomena.

Mamei and Zambonelli discussed already an approach to achieving load balancing inside a building such as a museum [95]. They used the concept of a crowd field,

a continuous mapping of the museum floor area into a field of "hills and valleys" that reflects the level of crowding in each room. The more crowded a room, the higher the "altitude" of the crowd field in that room. A hill-descending exploration strategy would encourage tourists to visit less crowded areas. The crowd field can be combined linearly with a mapping reflecting what rooms are more interesting to the tourist at hand, to obtain a strategy that allows to visit interesting locations while avoiding a high level of crowding.

In outdoor scenarios, some works evaluate the configuration of tourists in a touristic area (i.e., the number of tourists at each relevant location in the area) using an objective function. In [92] authors argue that the Variance and the Gini-Simpson Index [79], both used as objective functions in previous mathematical models for tourist load balancing, have their own limitations. The authors introduce an objective function called the *Tourism Utility Function* (TUF), which assigns a lower satisfaction score when a visited spot is overcrowded (i.e., when the number of tourists exceeds a given capacity).

In designing a strategy to dispatching tourists through a given touristic area, several authors represent the area as a graph with relevant locations and links [92, 62, 158]. Dispatching strategies discussed in the literature include stochastic (probabilistic) approaches. For example, the average strategy [92, 158], applied at a given node in the graph, picks any of the successor nodes with a uniform probability. In the same works, the space-time strategy favors nodes with a lower occupancy rate. [158] combines the two strategies to obtain a reduction in the duration of bottlenecks (i.e., overcrowded spots).

Authors in [90, 78] have considered Radio Frequency Identification (RFID) as a technology suitable for tracking tourists. This information can provide a view on the crowding levels at various locations within the monitored area, and it can be utilized as input data to various tourist dispatching strategies.

[113] studies the spatio-temporal distributions of visitors in a Chinese tourist site to provide a navigation for balancing the load of visitors within the site. The authors simulated the visitor data by assuming the capacity of sites, roads, visitor walking and sightseeing time, among others, and quantified the loads of each visitor spot. To balance the loads, the authors build a mathematical model that minimizes the variance of loads in each spot. The authors do not investigate how to find a solution to this mathematical model. The paper also points out that vehicle scheduling can be used to balance the loads within a site. It formalizes a plan for vehicle scheduling and presents a list of steps for simulating the scheduling. Again, it does not provide a method to solve the scheduling problem in this context.

4.5 Conclusions

Smart phones, pervasive sensing, and interactive displays can be effectively put at work in public spaces to suggest pedestrians how to move based on current crowd conditions, on their specific mobility needs, or on other contingencies. This chap-

ter surveyed a number of possible strategies to enforce crowd steering application and presented some exemplar use cases for both indoor and outdoor crowd steering scenarios. Furthermore, simulation has been used as a powerful tool to better understand the effectiveness of two exemplary strategies. Experimental results suggest that simple strategies based on local information can work as well as more global strategies, and sometimes even outperform them.

In both the case studied, data driven approaches were proposed. In the indoor scenario, the input data concerning users' localization, booths' occupancy, and hallways' crowding, was simulated along with the crowd dynamics (i.e., users requests and steering recommendations). On the other hand, the outdoor scenario, has taken into account real data by exploiting the datasets provided in the Telecom Italia Big Data Challenge so as to estimate the crowding level on the streets and at the POIs. Of course, in the future, instead of using such as coarse-grained data, one can think to have more reliable and precise one coming from, e.g., Wi-Fi hotspots, museums, CCTV, drones¹⁹. However, both cases present the novelty of a user-in-the-loop approach to Smart City.

The case studied in Section 4.2 proposed the application of a *global info - global goal - unicast* strategy to the indoor crowd steering scenario through FairSteering mobility service. Group of people directed to the fair would find less annoying the visiting of the booths, by avoiding overcrowded hallways and booths themselves.

The case study presented in Section 4.3 concerned the implementation of an advanced and dynamic activity planner for tourists visiting the historical city of Venice. The development of the conceived architecture enabled an even distributed crowd of people across the city, by creating benefits for both business activities (i.e., a visibility increase for the ones not located in main roads) and tourists (i.e., a quicker journey for letting them visit more attractions along the day, avoid overcrowded areas, be provided with deals and promotions from local shops).

¹⁹ Recently, *Jakarta Police*, reported to plan the deployment of five drones for monitoring the traffic, <http://www.thejakartapost.com/news/2015/06/30/city-police-use-drones-monitor-traffic.html>

Chapter 5

Conclusions

5.1 Summary

Despite everyone talks about Smart Cities as well grounded socio-technical systems that directly address people needs through technology exploitation, the exemplar cases deployed around the world depicts that we are far away to see distributed infrastructures of coordinated sensors and actuators effectively put at work.

Driven by bio-inspired models and theories in the research field of self-adaptive and self-organizing systems, literature presents the concept of urban superorganism as an entity endowed with sensing, actuating and computing capabilities closed in a feedback loop to enable collective awareness, and to support collective actions. However, the presence of a distributed ICT infrastructure deployed across the urban environment is scarce, and the most enabling kind of device that impacts on the efficacy of nowadays urban superorganisms is smart phone.

Despite ICT infrastructures such as, e.g., sensor networks, security cameras, autonomous cars, are likely to pervade cities in the near future, the amount of people carrying a smart phone is quite high already, and the penetration is increasing. Embedded sensing technology, computational power, connectivity features, and actuating capabilities provided by smart phones let them play an important role in the transition towards the envisioned Smart City concept. Of course, in the future, the more heterogeneous the ICT urban infrastructure will be, the better Context Awareness will be enabled by the socio-technical system.

This Thesis focused on innovative approaches to perform Behavioral Steering of individuals by supporting collaborative and cooperative actions for the sake of solving mobility issues in urban spaces. In particular, a set of SAMS can be put in place to improve urban mobility. These services represent the coherent evolution of Intelligent Transportation Systems to manage real-time mobility. Indeed, urban mobility needs can be addressed in a more effective way if a unified and integrated approach provided by a general purpose coordination framework is put in place. Furthermore, user-to-user and user-to-infrastructure interactions should be encouraged and sup-

ported. The increasing participation of citizens will transform them from simple requestors of mobility services to providers of those.

Two assessed collaborative mobility services were analyzed and studied in this work, which are related respectively to the sharing of private vehicles and to enhance pedestrian mobility: Ride Sharing and Crowd Steering. The first aims at reducing the amount of circulating vehicles on the road network by matching a set of users to the same car, the latter aims at guiding individuals towards the shaping of an evenly distributed crowd. Impacts of both the scenarios benefit a set of stakeholders on different scales, from individuals to urban authorities through, e.g., business activities, touristic attractions, means of transportation.

5.2 Open Challenges

The observable gap between conceptual, mostly theoretical, distributed urban infrastructures and their actual deployment is linked to a complex, extremely advanced scenario where they are dived into.

Software architects must, on the one hand, deal with the needs of a very wide set of users, with different goals at different scales, and, on the other hand, achieve effective fusion and manipulation of the heterogeneous data streams on the computing nodes. In fact, at both individual and urban scale, multiple data streams of information have to be processed and put together to build up useful services. A suitable set of algorithms is needed to extract high-level knowledge information. The awareness acquired by the individuals through sensors should be used to infer complex situations and create a sort of collective awareness of the surrounding physical and social space. Furthermore, architectures need to provide a certain degree of flexibility by allowing dynamic service composition, and thus by being able to self-reconfigure and self-optimize themselves with regard to contextual conditions. Finally, in the design of architectures for urban superorganisms, the trade-off between a top-down and a bottom-up approach must be pursued: they must cope with a dynamic execution-context by keeping control an easy to achieve task.

From a functional point of view, of course the effectiveness of the enabled services is related to the amount of users actually involved in the sensing, computing, and actuating activities. The engagement of individuals should be stimulated by worthy mobility services that actually address their needs while provide the superorganism with a collective perspective on urban mobility. For instance, data collected and shared from users in movement is of particular interest. However, this data is deeply affected by privacy concerns and sharing policies. Users' whereabouts and mobility patterns are required for mapping urban dynamics, but constitute very sensible data. Another approach that impacts and stimulate users is represented by incentive mechanisms taken from game and persuasion theory. Reputation systems must be established to guarantee an adequate level of trust to involve a greater number of users. Introducing a monetary form of compensation can support the occurrence of social interactions as well. It is worth noticing that only on a long-term

perspective, when social interactions become recurrent and protracted for years, permanent behavioral changes can be distinguished from short-lived ones.

5.3 Future Directions

The roadmap to overcome the challenging scenario of superorganism can be covered by structuring the interactions and the coordination among humans and ICT devices. Future SAMS will involve advanced capabilities of pattern analysis, matchmaking and orchestration, calling for the identification and implementation of a general-purpose coordination infrastructure to integrate such capabilities and support the effective realization of such services.

The use cases presented in this work concern innovative approaches and implementations of SAMS such as Ride Sharing and Crowd Steering.

The experimental results obtained in the study on mobility demand (Section 3.2) show how the differences emerged between Torino and Milano might suggest that specific urban features such as road topology and people distribution impact significantly on the results of large-scale ride sharing systems. On these basis, future work might include the analysis on additional cities to further investigate this issue, by eventually identifying policies to build trip-sharing-ready urban environments.

In Section 3.3, by adding social dimension to the mobility demand study, emerged that different people may have different preferences. Taking into account this fact, the study would have required a different model, which was out of the scope of this work, as feasibility was only taken into account. Along these lines, future work could include: user ad-hoc optimization; traffic-dependent optimization (i.e. higher cost of adding a car during peak hours); distance-dependent optimization (i.e., relaxing the optimization for shorter trips). Furthermore, there may be several obstacles for a larger adoption of ride sharing: safety of passengers, mobility mismatches, social incompatibility, and so on. The use case aims at analyzing only one of them, and under only one definition of enjoyability. More possibilities will be investigated in the future, together with the inclusion of other factors like gender match, trust, network similarity, and others.

A real-time, on-demand ride sharing recommender system is the aim of the case study presented in Section 3.4. However, at the time of writing, it does not take into accounts social aspects of ride sharing matching. This task will be covered in the future.

In any of the presented ride sharing use cases, the choice for CDR, and Twitter data was supported by a sort of replicability considerations. One could easily mix the data sources to improve the quality and the quantity of data in all the dimensions of the mobility and social analysis problems: better location data, better textual content, weighted friendship information. Alternative data to estimate mobility and social demand includes, e.g., Flickr, Foursquare, Facebook, taxi-sharing data, journey planner requests, public transportation ticketing. However, the estimation

provided in these studies was needed just to assess the feasibility of the proposed methodologies. Future works might include their quality evaluation.

Concerning the crowd steering service, both the indoor and the outdoor scenarios depicted in Chapter 4 can be improved in the future. In general, further strategies surveyed in Section 4.1.3 can be experimented. Especially, the trade-off between global-local goals and global-local information can be assessed. Furthermore, an actual deployment of the demonstrative architecture for indoor environments presented in Section 4.2.3 is intended to bring relevant results towards the assessment of the required technological infrastructure to enable such service. The user engagement by means of a mobile application and interactive displays, which are able of giving them tailored suggestions and providing them with a global crowding perception, could be evaluated as well.

The incentive mechanism mentioned during the description of the outdoor scenario, in Section 4.3.2, might be investigated deeper, implemented and tested as well. Furthermore, an incentive-based system able to impact on a set of heterogeneous stakeholders should be put in place also in the indoor scenario.

A further step forward would be to extend both indoor and outdoor services to other contexts and other cities with different settings and challenges (e.g., at shopping malls, at airports, in a city with other transport modes).

Other insights could be faced with a centralized planning approach for multiple users: this could be suitable in particular circumstances, such as guiding organized groups of tourists across a crowded area, such as guided tours in natural parks, or museums.

Finally, one could model the tourist activity planning as a scheduling with temporal constraints and resources. Each user can be defined through a list of goals, and temporal constraints are defined over the opening hours of the POIs, the time needed to visit a POI, the time to spend in a queue, and the transfer time from one location to another. Problems can be encoded into a standard representation language such as *Optimization Programming Language* (OPL). An off-the-shelf solver, such as *CPLEX Optimizer*¹, can be used to attack the instances.

¹ <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>

Acknowledgements

This work was supported by:

- Emilia Romagna SPINNER2013 project MUCCA – Cooperative Urban Mobility in Smart Cities;
- EU FP7 Project ASCENS – Autonomic Service-Component Ensembles (FP7-ICT-257414);
- EU FP7 Project FrontierCities – European Cities driving the Future Internet (FP7-ICT-632853-020);
- EU FP7 Project PETRA – Personal Transport Advisor (FP7-ICT-609042);
- Telecom Italia, Turin Swarm Joint Open Lab.

References

1. Abdelghany, A., Abdelghany, K., Mahmassani, H., Alhalabi, W.: Modeling Framework for Optimal Evacuation of Large-scale Crowded Pedestrian Facilities. In: European Journal of Operational Research, Elsevier, vol. 237, n.3, pp. 1105-1118, 2014
2. Alakeel, A.M.: A Guide to Dynamic Load Balancing in Distributed Computer Dystems. In: International Journal of Computer Science and Information Security, vol. 10, n. 6, pp. 153-160, 2010
3. Albiol, A., Silla, M.J., Mosi, J.M.: Video Anaysis using Corner Motion Statistics. In: Workshop on PETS, pp. 31-38, 2009
4. Alexander, L., Toole, J., Gonzales M.: Asessing the Impact of Ride Sharing on Congestion using Mobile Phone Data. In: Netmob International Conference, Cambridge (MA), USA, 2015
5. An, S.-H., Lee, B.-H., Shin, D.-R.: A Survey of Intelligent Transportation Systems. In: 2011 3rd International Conference on Computational Intelligence, Communication Systems and Networks CICSyN, Bali, Indonesia, 2011
6. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. In: Computer Networks, vo. 54, n. 15, pp. 2787-2805, 2010
7. Baptista, A., Bouillet, E., Calabrese, F., Verscheure, O.: Towards Building an Uncertainty-aware Personal Journey Planner. In: IEEE International Conference on Intelligent Transportation Systems, Washington (WA), USA, 2011
8. Bazzan, A.: A Distributed Approach for Coordination of Traffic Signal Agents, In: Autonomous Agents and Multi-Agent Systems, vol. 10, n. 4, pp. 131-164, 2005
9. Bento, A.M., Hughes, J.E., Kaffine, D.: Carpooling and Driver Responses to Fuel Price Changes: Evidence from Traffic Flows in Los Angeles. In: Journal of Urban Economics, vol. 77, pp. 41-56, 2013
10. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of Context Modelling and Reasoning Techniques. In: Pervasive and Mobile Computing, vol. 6, pp. 161-180, 2010

11. Bicocchi, N., Castelli G., Lasagni, M., Mamei, M., Zambonelli, F.: Experiences on Sensor Fusion with Commonsense Reasoning. In: IEEE Workshop on Context Modeling and Reasoning, Lugano, Switzerland, 2012
12. Bicocchi, N., Cecaj, A., Fontana, D., Mamei, M., Sassi, A., Zambonelli, F.: Collective Awareness for Human-ICT Collaboration in Smart Cities. In: 2012 IEEE 22st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Hammamet, Tunisia, 2013
13. Bicocchi, N., Mamei, M.: Investigating Ride Sharing Opportunities through Mobility Data Analysis. In: Pervasive and Mobile Computing Journal, vol. 14, pp. 83-94, 2014
14. Bicocchi, N., Mamei, M., Sassi, A., Zambonelli, F.: Opportunistic Ride Sharing via Whereabouts Analysis. In: Proceedings of the 2015 IEEE 18th Int. Conf. on Intelligent Transportation Systems ITSC, Las Palmas de Gran Canaria, Spain, 2015
15. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. In: Journal of Machine Learning Research, vol. 3, n. 1, pp. 993-1022, 2003
16. Booth, J., Sistla, P., Wolfson, O., Cruz, I.F.: A Data Model for Trip Planning in Multimodal Transportation Systems. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology EDBT 09, Saint Petersburg, Russia, 2009
17. Bonabeau E., Dorigo M., Theraulaz G.: Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press, London (UK), 1998
18. Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M.: Swarm Robotics: a Review from the Swarm Engineering Perspective. In: Swarm Intelligence, vol. 7, n. 1, pp. 1-41, 2013
19. Brereton, M., Roe, P., Foth, M., Bunker, J., Buys, L.: Designing Participation in Agile Ridesharing with Mobile Social Software. In: ACM OZCHI Conference of the Australian Computer-Human Interaction, Melbourne, Australia, 2009
20. Brilhante, I., Macedo, J.A., Nardini, F.M., Perego, R., Renso, C.: Where Shall We Go Today?: Planning Touristic Tours with Tripbuilder. In: CIKM, 2013
21. Brown, C., Nicosia, V., Scellato, S., Noulas, A., Mascolo, C.: Social and Place-focused Communities in Location-based Online Social Networks. In: Computing Research Repository CoRR, ACM, 2013
22. Brown, C., Noulas, A., Mascolo, C., Blondel, V.: A Place-focused Model for Social Networks in Cities. In: Computing Research Repository CoRR, ACM, 2013
23. Bruno, L., Venuti, F.: The Pedestrian Speed-Density Relation: Modelling and Application. In: Footbridge 2008 3rd International Conference, Porto, Portugal, 2008
24. Byon, Y.J., Jeong, Y.S., Easa, S.M., Baek, J.: Feasibility Analysis of Transportation Applications based on APIs of Social Network Services. In: ICITST, 2013
25. Cabri, G., Leonardi, L., Ferrari, L., Zambonelli, F.: Role-based Software Agent Interaction Models: a Survey. In: Knowledge Engineering Review, Cambridge, vol. 5, n. 4, pp. 397-419, 2010

26. Calabrese, F., Reades, J., Ratti, C.: Eigenplaces: Analysing Cities Using the Space-time Structure of the Mobile Phone Network. In: *IEEE Pervasive Computing*, vol. 9, n. 1, pp. 78-84, 2010
27. Calabrese, F., Di Lorenzo, G., Liu, L., Ratti, C.: Estimating Origin-Destination Flows Using Mobile Phone Location Data. In: *IEEE Pervasive Computing*, vol. 10, n. 4, pp. 36-44, 2011
28. Capodiecì, N., Hart, E., Cabri, G.: Artificial Immune System driven evolution in Swarm Chemistry. In: *Proceedings of the 2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems*, Piscataway (NJ), USA, 2014
29. Castelli, G., Mamei, M., Zambonelli, F.: The Changing Role of Pervasive Middleware: From Discovery and Orchestration to Recommendation and Planning. In: *PerWare Workshop at the 9th IEEE International Conference on Pervasive Computing and Communications*, Seattle (WA), USA, 2011
30. Castelli, G., Mamei, M., Rosi, A., Zambonelli, F.: Multiagent Environment Design for Pervasive Human-ICT Systems: The SAPERE Approach. In: *Handbook of Human Computation*, Pietro Michelucci Editor, Springer, 2013
31. Cecaj, A., Mamei, M., Zambonelli, F.: Re-identification and information fusion between anonymized CDR and social network data. In: *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-14, 2014
32. Charles, K.K., Kline, P.: Relational Costs and the Production of Social Capital: Evidence from Carpooling. In: *The Economic Journal*, vol. 116, n. 511, pp. 581-604, 2006
33. Chemudugunta, C., Steyvers, P., Smyth M.: Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model. In: *NIPS*, vol. 19, 2007
34. Chen, B., Pinelli, F., Sinn, M., Botea, A., Calabrese, F.: Uncertainty in Urban Mobility: Predicting Waiting Times for Shared Bicycles and Parking Lots. In: *IEEE International Conference on Intelligent Transportation Systems*, 2013
35. Chen, X., Santos-Neto, E., Ripeanu, M.: Crowdsourcing for On-street Smart Parking. In: *Proceedings of the 2nd ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications DIVANET 12*, Cyprus, 2012
36. Cho, C.W., Wu, Y.H., Yen, C. Chang, C.Y.: Passenger Search by Spatial Index for Ridesharing. In: *2012 Conference on Technologies and Applications of Artificial Intelligence*, Los Alamitos (CA), USA, 2012
37. Cho, E., Myers, S.A., Leskovec, J.: Friendship and Mobility: User Movement in Location-based Social Networks. In: *KDD*, 2011
38. Ciciy, B., Markopoulou, A., Frías-Martínez, E., Laoutaris, N.: Quantifying the Potential of Ride-Sharing using Call Description Records. In: *ACM International Workshop on Mobile Computing Systems and Applications*, Jekyll Island (GE) USA, 2013
39. Ciciy, B., Markopoulou, A., Frías-Martínez, E., Laoutaris, N.: Assessing the Potential of Ride-sharing using Mobile and Social Data: a Tale of Four Cities. In: *UbiComp*, 2014
40. Coltin, B.J., Veloso, M.: Towards Ridesharing with Passenger Transfers. In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*

AAMAS 13, Saint Paul (MN), USA, 2013

41. Conte, D., Foggia, P., Percannella, G., Vento, M.: A Method Based on the Indirect Approach for Counting People in Crowded Scenes. In: AVSS, pp. 111-118, 2010
42. Conti, M., Das, S., Bisdikian, C., Kumar, M., Ni, L., Passarella, A., Roussos, G., Trster, G., Tsudik, G., Zambonelli, F.: Looking ahead in pervasive computing: Challenges and opportunities in the era of cyberphysical convergence. In: Pervasive and Mobile Computing, vol. 8, n. 1, pp. 2-21, 2012
43. Coppersmith, D., Nowicki, T., Paleologo, G., Tresser, C., Wu, C.W.: The Optimality of the Online Greedy Algorithm in Carpool and Chairman Assignment Problems. In: TALG, vol. 7, n. 3, 2011
44. Corradi, A., Leonardi, L., Zambonelli, F.: Diffusive Load-balancing Policies for Dynamic Applications. In: IEEE Concurrency, IEEE Computer Society, vol. 7, n. 1, pp. 22-31, 1999
45. Correia, G., Viegas, J.M.: Carpooling and Carpool Clubs: Clarifying Concepts and Assessing Value enhancement possibilities through a Stated Preference Web Survey in Lisbon, Portugal. In: Transportation Research Part A: Policy and Practice, vol. 45, n. 2, pp. 81-90, 2011
46. Dahlgren, J.: High Occupancy Vehicle Lanes: Not always More Effective than General Purpose Lanes. In: Transportation Research Part A: Policy and Practice, Elsevier, vol. 32, n. 2, pp. 99-114, 1998
47. Davies, N., Langheinrich, M., José, R., Schmidt, A.: Open Display Networks: A Communications Medium for the 21st Century. In: IEEE Copmuters, vol. 45, n. 5, pp. 58-64, 2012
48. Davies, N., Clinch, S., Alt, F.: Pervasive Displays: Understanding the Future of Digital Signage. In: Synthesis Lectures on Mobile and Pervasive Computing - Pervasive Displays, Morgan & Claypool Publishers, 2014
49. Davies, N., Langheinrich, M., Clinch, S., Elhart, I., Friday, A., Kubitza, T., Surajbali, B.: Personalisation and Privacy in Future Pervasive Display Networks. In: CHI Conference on Human Factors in Computing Systems CHI'14, Toronto, Canada, 2014
50. De Jonge, B., Teunter, R.H.: Optimizing Itineraries in Public Transportation with Walks between Rides. In: Transportation Research Part B: Methodological, Elsevier, vol. 55, pp. 212-226, 2013
51. Eagle, N., Pentland, A.: Eigenbehaviors: Identifying Structure in Routine. In: Behavioral Ecology and Sociobiology, vol. 63, n. 7, pp. 1057-1066, 2009
52. Elbery, A., ElNainay, M., Chen, F., Lu, C.T., Kendall, J.: A Carpooling Recommendation System based on Social VANET and Geo-social Data. In: SIGSPATIAL, 2013
53. Farrahi, K., Gatica-Perez, D.: Discovering Routines from Large-Scale Human Locations using Probabilistic Topic Models. In: ACM Transactions on Intelligent Systems and Technology, vol. 2, n. 1, 2011
54. Ferenstein, G.: Uber Shows new Carpooling Feature Reduces Traffic Congestion 50% in Pilot Areas. In: <http://venturebeat.com/2014/10/07/>, 2014
55. Fernandez-Marquez, J.L., Di Marzo Serugendo, G., Montagna, S., Viroli, M., Arcos, J.: Description and composition of bio-inspired design patterns: a complete overview. In:

- Natural Computing, vol. 12, n. 1, pp. 43-67, 2013
56. Ferrari, L., Mamei, M.: Classification and Prediction of Whereabouts Patterns from the Reality Mining Dataset. In: Pervasive and Mobile Computing, IEEE, vol. 9, n. 4, pp. 516-527, 2013
 57. Fogg, B.J.: Persuasive Computers: Perspectives and Research Directions. In: Proceedings of the SIGCHI Conf. on Human Factors in Computing Systems, Los Angeles (CA), USA, 1998
 58. Fogg, B.J.: Persuasive Technology: Using Computers to Change What We Think and Do. In: Science & Technology Books, Kaufmann, 2002
 59. Fougères, A.J., Canalda, P., Ecarot, T., Samaali, A., Guglielmetti, L.: A Push Service For Carpooling. In: 2012 IEEE International Conference on Green Computing and Communications GreenCom, Besancon, France, 2012
 60. Frias-Martinez, V., Soto, V., Hohwald, H., Frias-Martinez, E.: Characterizing Urban Landscapes using Geolocated Tweets. In: PASSAT, SocialCom, 2012
 61. Garaix, T., Artigues, C., Feillet, D., Josselin, D.: Vehicle Routing Problems with Alternative Paths: an Application to On-demand Transportation. In: European Journal of Operational Research, Elsevier, vol. 204, n. 1, pp. 62-75, 2010
 62. Ge, P., Zheng, W., Xiao, X.: A Simulation Study based on the Regional Space-time Load Balancing of Jiuzhaigou. In: Journal of Industrial Engineering and Engineering Management, CNKI, n. 2, 2013
 63. Gidófalvi, G., Herenyi, G., Bach Pedersen, T.: Instant Social Ride-Sharing. In: Proceedings of the Fifteenth World Congress on Intelligent Transport Systems, New York (NY), USA, 2008
 64. Girardin, F., Blat, J., Calabrese, F., Dal Fiore, F., Ratti, C.: Digital Footprinting: Uncovering Tourists with User-Generated Content. In: IEEE Pervasive Computing, vol. 7, n. 4, p. 36-44, 2008
 65. Gong, X., Liu, X.: A Data Mining Based Algorithm for Traffic Network Flow Forecasting. In: International Conference on Integration of Knowledge Intensive Multi-Agents Systems, Boston (MA), USA, 2003
 66. Graziotin, D.: An Analysis of Issues Against the Adoption of Dynamic Carpooling. In: arXiv preprint arXiv:1306.0361, 2013
 67. Hall, R., Qureshi, A.: Dynamic Ride-Sharing: Theory and Practice. In: Journal of Transportation Engineering, Elsevier, vol. 123, n. 4, pp. 08-315, 1997
 68. Harnie, D., D'Hondt, T., Boix, E.G., De Meuter, W.: Programming Urban-area Applications for Mobility Services. In: Transactions on Autonomous and Adaptive Systems TAAS, ACM, vol. 9, n. 2, 2014
 69. He, W., Li, D., Zhang, T., An, L., Guo, M., Chen, G.: Mining Regular Routes from GPS Data for Ridesharing Recommendations. In: Proceedings of the ACM SIGKDD International Workshop on Urban Computing UrbComp 12, Beijing, China, 2012
 70. Helaoui, R., Riboni, D., Stuckenschmidt, H.: A Probabilistic Ontological Framework for the Recognition of Multilevel Human Activities. In: ACM International Joint Conference on

Pervasive and Ubiquitous Computing, Zurich, Switzerland, 2013

71. Helbing, D., Buzna, L., Johansson, A., Werner, T.: Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions. In: *Transportation Science*, vol. 39, n. 1, pp. 1-24, 2005
72. Holldobler, B., Wilson, O.: *The Superorganism: the Beauty, Elegance, and Strangeness, of Insect Societies*. W. W. Norton and Co., New York (NY), USA (2009)
73. Hongbo L., Jie, Y., Sidhom, S., Yan, W., Yingying C., Fan Y.: Accurate WiFi Based Localization for Smartphones Using Peer Assistance. In: *IEEE Transactions on Mobile Computing*, vol. 13, n. 10, pp. 2199-2214, 2014
74. Hu, X., Wang, W., Leung, V.C.M. : VSSA: A Service-oriented Vehicular Social-Networking Platform for Transportation Efficiency. In: *DIVANet-12 Proceedings of the second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, New York (NY), USA, 2012
75. Isaacman, S., Becker, R., Cceres, R., Kobourov, S., Martonosi, M., Rowland, J., Varshavsky, A.: Identifying Important Places in Peoples Lives from Cellular Network Data. In: *International Conference on Pervasive Computing*, San Francisco (CA), USA, 2011
76. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. In: *ACM Computing Surveys (CSUR)*, vol. 31, n. 3, pp. 264-323, New York (NY), USA, 1999
77. Jiau, M.K., Huang, S.C., Lin, C.H.: Optimizing the Carpool Service Problem with Genetic Algorithm in Service-Based Computing. In: *SCC*, 2013
78. Jin, M., Zheng, D., Ren, P.: Application of RFID Based Low-Carbon Scenic Integrated Management System in Jiuzhaigou Area. In: *Proceedings of the 7th International Conference on Management Science and Engineering Management*, vol. 242 of *Lecture Notes in Electrical Engineering*, Springer Berlin Heidelberg, pp. 1269-1280, 2014
79. Jost, L.: Entropy and Diversity. In: *Oikos*, Blackwell Publishing Ltd, vol. 113, n. 2, pp. 363-375, 2006
80. Julien, C., Roman, G.C.: EgoSpaces: Facilitating Rapid Development of Context-Aware Mobile Applications. In: *Transactions on Software Engineering, IEEE*, vol. 32, n. 5, pp. 281-298, 2006
81. Kanhere, S.: Participatory Sensing: Crowdsourcing Data from Mobile Smartphones in Urban Spaces. In: *IEEE International Conference on Mobile Data Management*, Bengaluru, India, 2012
82. Kehoe, M.: *Understanding IBM Smart Cities*. Redbook Series, IBM Corporation, New York (NY), USA (2011)
83. Klein, M.C.A., Mogles, N.M., Van Wissen, A.: Why Won't You Do What's Good for You? Using Intelligent Support for Behavior Change. In: *Lecture Notes in Computer Science*, Springer, Salah, A.A. and Lepri, B. editors, v. 7065, pp. 104-115, 2011
84. Knapen, L., Keren, D., Yasar, A.U.H., Cho, S., Bellemans, T., Janssens, D., Wets, G.: Estimating Scalability Issues while Finding an Optimal Assignment for Carpooling. In: *Procedia Computer Science*, vol. 19, pp. 372-379, 2013

85. Koutsopoulos, I.: Optimal Incentive-driven Design of Participatory Sensing Systems. In: INFOCOM, 2013 Proceedings IEEE, 2013
86. Kwon, Y.C., Nunley, D., Gardner, J.P., Balazinska, M., Howe, B., Loebman, S.: Scalable Clustering Algorithm for N-Body Simulations in a Shared-Nothing Cluster. In: Lecture Notes in Computer Science: Scientific and Statistical Database Management, Springer, Gertz, Mi., Ludascher, B. editors, vol. 6187, pp. 132-150, 2010
87. Lathia, N., Pejovic, V., Rachuri, K.K., Mascolo, C., Musolesi, M., Rentfrow, P.J.: Smart-phones for Large-Scale Behavior Change Interventions. In: IEEE Pervasive Computing, vol. 12, n. 3, pp. 66-73, 2013
88. Lee, D.W., Liang, S.H.L.: Crowd-sourced Carpool Recommendation based on Simple and Efficient Trajectory Grouping. In: SIGSPATIAL, 2011
89. Lee, J.S., Hoh, B.: Dynamic Pricing Incentive for Participatory Sensing. In: Pervasive and Mobile Computing, Elsevier, vol. 6, n. 6, pp. 693-708, 2010
90. Leong, A., Li, X.: Digital Culture and E-Tourism: Technologies, Applications and Management Approaches. In: Study on Tourist Management in China Based on Radio Frequency Identification (RFID) Technology, chapter A, pp.190-203, 2011
91. Lerenc, V.: Increasing Throughput for Carpool Assignment Matching. In: US Patent App. 13/329,899, Google Patents, 2011
92. Lin, S., Yuan, Y., Zheng, W., Ren, P., Jin, M.: Evaluation Model Construction and Simulation Research on Tourist Diversion Strategy in Ecotourism Scenery Sport. In: Applied Mathematics and Information Sciences, vol. 7, n. 6, pp. 2249-2257, 2013
93. Ma, S., Zheng, Y., Wolfson, O.: T-share: A Large-scale Dynamic Taxi Ridesharing Service. In: ICDE, 2013
94. Ma, S., Zheng, Y., Wlfson O.: Real-Time City-Scale Taxi Ridesharing. In: IEEE Transactions on Knowledge and Data Engineering, vol. 27, n. 7, 2015
95. Mamei, M., Zambonelli, F.: Field-based Approaches to Adaptive Motion Coordination in Pervasive Computing Scenarios. In: The Handbook of Algorithms for Mobile and Wireless Networking and Computing, CRC Press, 2004
96. Mamei, M., Zambonelli, F.: Field-Based Coordination for Pervasive Multiagent Systems. In: Springer Series on Agent Technology, Springer-Verlag, 2005
97. Mamei, M., Ferrari, L.: Classification of Whereabouts Patterns From Large-scale Mobility Data. In: Proceedings of 11th Workshop nazionale "Dagli Oggetti agli Agenti" (WOA), Rimini, Italy, 2010
98. Manenti, L., Manzoni, S., Vizzari, G., Ohtsuka, K., Shimura, K.: Towards an Agent-Based Proxemic Model for Pedestrian and Group Dynamic. In: WOA, 2010
99. Martinez, L.M., Correia, G., Viegas, J.: An Agent-based Model to Assess the Impacts of Introducing a Shared-taxi System in Lisbon (Portugal). In: Proceedings of the 7th International Workshop on Agents in Traffic and Transportation, 2012.
100. Massaro, D.W., Chaney, B., Bigler, S., Lancaster, J., Iyer, S., Gawade, M., Eccleston, M., Gurrola, E., Lopez, A.: Carpoolnow-Just-in-Time Carpooling without Elaborate Preplanning.

- In: WEBIST, 2009
101. Mathur, S., Jin, T., Kasturirangan, N., Chandrasekaran, J., Xue, W., Gruteser, M., Trappe, W.: ParkNet: Drive-by Sensing of Road-side Parking Statistics. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services MobiSys 10, San Francisco (CA), USA, 2010
 102. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in Social Networks. In: Annual review of sociology, 2001
 103. Mehrotra, A., Musolesi, M., Hendley, R., Pejovic, V.: Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In: ACM International Joint Conference on Pervasive and Ubiquitous Computing, Osaka, Japan, 2015
 104. Mitchell, M.: Self-awareness and control in decentralized systems. In: AAAI Spring Symposium: Meta-cognition in Computation, Palo Alto (CA), USA, 2005
 105. de Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., Blondel, V. D.: Unique in the Crowd: The privacy bounds of human mobility. In: Nature Scientific Reports, vol. 3, Nature Publishing Group, 2013
 106. Moussad, M., Perozo, N., Garnier, S., Helbing, D., Theraulaz, G.: The Walking Behaviour of Pedestrian Social Groups and Its Impact on Crowd Dynamics. In: PLoS ONE, Public Library of Science, vol.5, n. 4, 2010
 107. Moussad, M., Helbing, D., Theraulaz, G.: How simple Rules determine Pedestrian Behavior and Crowd Disasters. In: Proceedings of the National Academy of Sciences, vol. 108, n. 17, pp. 6884-6888, 2011
 108. Munkres, J.: Algorithms for the Assignment and Transportation Problems. In: Journal of the Society for Industrial & Applied Mathematics, vol. 5, n. 1, pp. 32-38, 1957
 109. Nawaz, S., Efstratiou, C., Mascolo, C.: ParkSense: A Smartphone Based Sensing System for On-street Parking. In: Proceedings of the 19th Annual International Conference on Mobile Computing & Networking MobiCom 13, Miami (FL), USA, 2013
 110. Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S., Stonebraker, M.: A Comparison of Approaches to Large-scale Data Analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, Providence (RI), USA, 2009
 111. Pejovic, V., Musolesi, M.: Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges. In: arXiv preprint arXiv:1306.2356, 2013
 112. Pianini, D., Viroli, M., Zambonelli, F., Ferscha, A.: HPC from a Self-organisation Perspective: The Case of Crowd Steering at the Urban Scale. In: International Conference on High Performance Computing & Simulation HPCS 2014, Bologna, Italy, 2014
 113. Qiu, Y., Ge, P., Ren, P.: A Study on Temporal and Spatial Navigation based on the Load-balance of Tourists in Jiuzhaigou Valley. Resource Science, n. 32, pp. 118-123, 2010
 114. Qu, F., Wang, F.Y., Yang, L.: Intelligent Transportation Spaces: Vehicles, Traffic, Communications, and Beyond. In: Communications Magazine, IEEE, vol. 48, n. 11, pp. 136-142, 2010

115. Quercia, D., Schifanella, R., Aiello, L.M.: The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy routes in the City. In: *Hypertext*, p. 116-125, 2014
116. Rahwan, I., Dsouza, S., Rutherford, A., Naroditskiy, V., McInerney, J., Venanzi, M., Jennings, N.R., Cebrián, M.: Global Manhunt Pushes the Limits of Social Mobilization. In: *IEEE Copmuter*, vol. 46, n. 4, pp. 68-75, 2013
117. Rana, R., Chou, C., Bulusu, N., Kanhere, S., Hu, W.: Ear-Phone: An End-to-End Participatory Urban Noise Mapping System. In: *International Conference on Information Processing in Sensor Network*, Stockholm, Sweden, 2010.
118. Rao, A.S., Georgeff, M.P.: Modeling Rational Agents within a BDI-Architecture. In: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, San Francisco (CA), USA, 1991
119. Robertson, D., Anderson, S., Carreras, I., Miorandi, D.: White Paper on Research Challenges in Social Collective Intelligence. In: *Social IST*, Springer Verlag, 2014.
120. Roggen, D., Wirz, M., Tröster, G., Helbing, D.: Recognition of Crowd Behavior from Mobile Sensors with Pattern Analysis and Graph Clustering Methods
121. Rosi, A., Mamei M., Zambonelli, F., Dobson S., Stevenson G., Ye J.: Social Sensors and Pervasive Services: Approaches and Perspectives. In: *IEEE Workshop on Pervasive Collaboration and Social Networking*, Seattle (WA), USA, 2011.
122. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In: *World Wide Web Conference*, Raleigh (NC), USA, 2010.
123. Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S., Ratti, C.: Proceedings of the National Academy of Sciences (PNAS), vol. 111, n. 37, pp. 13290-13294, 2014
124. Schauer, L., Werner, M., Marcus, P.: Estimating Crowd Densities and Pedestrian Flows Using Wi-Fi and Bluetooth. In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, London, UK, 2014
125. Spagnolli, A., Chittaro, L., Gamberini, L.: Persuasive Technology. In: *Proceedings of the 9th International Conference PERSUASIVE 2014*, Padua, Italy, 2014
126. Subburaman, V.B., Descamps, A., Carincotte, C: Counting People in the Crowd Using a Generic Head Detector. In: *Proceedings of the IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance AVSS12*, pp. 470-475, 2012
127. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet Processes. In: *Journal of the American Statistical Association*, vol. 101, n. 476, 2006
128. Tijdeman, R.: The chairman assignment problem. In: *Discrete Mathematics*, vol. 32, n. 3, pp. 323-330, 1980
129. Trasarti, R., Rinzivillo, S., Pinelli, F., Nanni, M., Monreale, A., Renso, C., Pedreschi, D., Giannotti, F.: Exploring Real Mobility Data with M-Atlas. In: *European Conference on Machine Learning and Knowledge Discovery in Databases*, Barcelona, Spain, 2010
130. Trasarti, R., Pinelli, F., Nanni, M., Giannotti, F.: Mining Mobility User Profiles for Car Pooling. In: *KDD International Conference*, San Diego (CA), USA, 2011

131. Tubaishat, M., Zhuang, P., Qi, Q., Shang, Y.: Wireless sensor networks in intelligent transportation systems. In: *Wireless Communication Mobile Computing*, vol. 9, n. 3, pp. 287-302, John Wiley and Sons Ltd., Chichester, UK, 2009
132. Tuner, B.: One Planet Mobility: A Journey towards a sustainable future. In: *WWF Report*, 2008
133. Ulm, M., Widhalm, P.: Properties of the Positioning Error of Cell Phone Trajectories. In: *NetMob Proceedings*, Boston (MA), USA, 2013
134. Van Beek, P., Manchak, D.W.: The Design and an Experimental Analysis of Algorithms for Temporal Reasoning. In: *Journal of Artificial Intelligence Research*, pp. 1-18, 1996
135. Van Brummelen, G. R.: *Heavenly Mathematics: The Forgotten Art of Spherical Trigonometry*. Princeton University Press, 2013
136. Van Vugt, M., Van Lange, P.A.M., Meertens, R.M., Joireman, J.A.: How a Structural Solution to a Real-world Social Dilemma Failed: A Field Experiment on the first Carpool lane in Europe. In: *Social Psychology Quarterly*, pp. 364-374, 1996
137. Viroli, M., Casadei, M., Montagna, S., Zambonelli, F.: Spatial Coordination of Pervasive Services through Chemical-Inspired Tuple Spaces. In: *Transactions on Autonomous and Adaptive Systems TAAS, ACM*, vol. 6, n. 2, 2011
138. VV.AA., *Effective Crowd Management: Guidelines on maintaining the Safety and Security of your Customers, Employees and Store*, NRF, <http://NRF.com/crowdmanagement>, 2005
139. VV.AA., *Centre of Regional Science: Smart cities - Ranking of European medium-sized cities*. Vienna, Austria, 2007
140. Wicker, S.: The Loss of Location Privacy in the Cellular Age. In: *Communication of the ACM*, vol. 55, n. 8, pp. 60-68, 2012
141. Wijermans, N., Jorna, R., Jager, W., Van Vliet, T., Adang, O.: CROSS: Modelling Crowd Behaviour with Social-Cognitive Agents. In: *Journal of Artificial Societies and Social Simulation*, vol. 4, 2013
142. Wilcox, R.: *Introduction to Robust Estimation and Hypothesis Testing*. In: Elsevier Academic Press, Burlington (MA), USA, 2005
143. Winstanley, C., Davies, N., Harding, M., Norgate, S.: Supporting Walking School Buses. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, Seattle (WA), USA, 2014
144. Wurtz, R.: *Organic Computing*. Springer Verlag, Berlin, Germany, 2008
145. Yan, X., Zhang, H., Wu, C.: Research and Development of Intelligent Transportation Systems. In: *11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science*, Guilin, China, 2012
146. Yang, L., Wang, F.Y.: Driving into Intelligent Spaces with Pervasive Communications. In: *Intelligent Systems, IEEE*, vol. 22, n.1, pp. 12-15, 2007
147. Ye, J., Dobson, S., McKeever, S.: Situation Identification Technologies in Pervasive Computing: A review. In: *Pervasive and Mobile Computing*, vol. 8, n. 1, pp. 36-66, Elsevier

- Science Pub., Amsterdam, The Netherlands, 2012
148. Yuen, M., Chen, L., King, I.: A Survey of Human Computation Systems. In: International Conference on Computational Science and Engineering, Vancouver, Canada, 2009.
 149. Zambonelli, F., Castelli, G., Ferrari, L., Mamei, M., Rosi, A., Di Marzo Serugendo, G., Risoldi, M., Tchao, A.E., Dobson, S., Stevenson, G., Ye, J., Nardini, E., Omicini, A., Montagna, S., Viroli, M., Ferscha, A., Maschek, S., Wally, B.: Self-aware Pervasive Service Ecosystems. In: *Procedia CS*, vol. 7, pp. 197-199, 2011
 150. Zambonelli, F.: Toward Sociotechnical Urban Superorganisms. In: *IEEE Computer*, vol. 45, n. 8, pp. 76-78, 2012
 151. Zambonelli, F., Castelli, G., Mamei, M., Rosi, A.: Programming Self-organizing Pervasive Applications with SAPERE. In: *Intelligent Distributed Computing VII - Proceedings of the 7th International Symposium on Intelligent Distributed Computing*, Springer, vol. 511, pp. 93-102, 2014
 152. Zapico, J.L., Turpeinen, M., Brandt, N.: Climate persuasive services: changing behavior towards low-carbon lifestyles. In: *Proceedings of the 11th International Conference on Persuasive Technology*, New York (NY), USA, 2009
 153. Zhang, A.X., Noulas, A., Scellato, S., Mascolo, C.: Hoodsquare: Modeling and Recommending Neighborhoods in Location-based Social Networks. In: *Computing Research Repository CoRR*, ACM, 2013
 154. Zhang, D., Li, Y., Zhang, F., Lu, M., Liu, Y., He, T.: coRide: Carpool Service with a Win-win Fare Model for Large-scale Taxicab Networks. In: *SenSys*, 2013
 155. Zhang, J., Wang, F., Wang, K., Lin, W., Xu, X., Chen, C.: Data-Driven Intelligent Transportation Systems: A Survey. In: *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, n. 4, pp. 1624-1639, 2011
 156. Zhao, W.X., Jiang, J., Weng, J., He, J., Lim, E.P., Yan, H., Li, X.: Comparing Twitter and traditional Media using Topic Models. In: *Advances in Information Retrieval*, 2011
 157. Zhaoyu, G., Muyuan, L., Suguo D., Haojin Z.: Fairness-Aware and Privacy-Preserving Friend Matching Protocol in Mobile Social Networks. In: *IEEE Transactions on Emerging Topics in Computing*, IEEE Computer Society, vol. 1, n. 1, pp. 192-200, 2013
 158. Zheng, W., Ge, P., Jin, M., Ren, P., Gao, H.: The Simulation Study of Space-Time Shunt in Jiuzhaigou Based on Bottleneck Theory. In: *Applied Mathematics & Information Sciences*, Natural Sciences Publishing Corp., vol. 6-3S, n. 3, pp.993-1000, 2012
 159. Zheng, Y.: Methodologies for cross-domain data fusion: an overview. In: *Big Data*, IEEE Transactions on, vol. 1, n. 1, pp. 16-34, 2015
 160. Zhu, J., Zeng, K., Kim, K.H., Mohapatra, P.: Improving Crowd-sourced Wi-Fi Localization Systems using Bluetooth Beacons. In: *IEEE 9th Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2012

List of Publications

- Bicocchi, N., Mamei, M., Sassi, A., Zambonelli, F., "An Opportunistic Ride Sharing Recommender System". Submitted to Intelligent Transportation Systems, IEEE Transactions on, February 2016
- Berlingiero, M., Gheddar, B., Guidotti, R., Pascale, A., Sassi, A., "The GRAAL of carpooling: GReen And sociAL optimization from crowd-sourced data". Submitted to Transportation Research, Part C, Elsevier, November 2015
- Sassi, A., Borean, C., Giannantonio, R., Mamei, M., Mana, D., Zambonelli, F., "Crowd Steering in Public Spaces: Approaches and Strategies". The 2015 IEEE 13th Int. Conf. on Pervasive Intelligence and Computing (PICom) Proceedings, October 2015, Liverpool, UK.
- Bicocchi, N., Mamei, M., Sassi, A., Zambonelli, F., "Opportunistic Ride Sharing via Whereabouts Analysis". The 2015 IEEE 18th Int. Conf. on Intelligent Transportation Systems Proceedings, Sept. 2015, Las Palmas di Gran Canaria, Spain.
- Guidotti, R., Sassi, A., Berlingiero, M., Gheddar, B., Pascale, A., "Social or green? A datadriven approach for more enjoyable carpooling". The 2015 IEEE 18th Int. Conf. on Intelligent Transportation Systems Proceedings, September 2015, Las Palmas di Gran Canaria, Spain.
- Borean, C., Giannantonio, R., Mamei, M., Mana, D., Sassi, A., Zambonelli, F., "Urban Crowd Steering: an Overview". IDCS 2015 The 8th Int. Conf. on Internet and Distributed Computing Systems, September 2015, Windsor, UK.
- Sassi, A., Zambonelli, F., "Coordination Infrastructures for Future Smart Social Mobility Services". IEEE Intelligent Systems, 29(5), pp. 78-82, September-October 2014.

- Sassi, A., Mamei, M., Zambonelli, F., "Towards a General Infrastructure for Location-based Smart Mobility Services". The 2nd International Workshop on Location-based Services and Applications in Ubiquitous Computing (LSAUC), held at the IEEE Int. Conf. on High Performance Computing & Simulation (HPCS), pp. 849-856, July 2014, Bologna, Italy.
- Sassi, A., Zambonelli, F., "Towards an Agent Coordination Framework for Smart Mobility Services". Proceedings of the 8th Int. Workshop on Agents in Traffic and Transportation (ATT), pp. 64-71, held at the 13th Int. Conf. on Autonomous Agents and Multiagents Systems (AAMAS 2014), May 5-6, 2014, Paris, France.
- Bicocchi, N., Cecaj, A., Fontana, D., Mamei, M., Sassi, A., Zambonelli, F., "Social Collective Awareness in Socio-Technical Urban Superorganisms". Social Collective Intelligence, pp. 227-241, Springer International Publishing, 2014.
- Bicocchi, N., Cecaj, A., Fontana, D., Mamei, M., Sassi, A., Zambonelli, F., "Collective Awareness for Human-ICT Collaboration in Smart Cities". Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2013 IEEE 22nd Int. Workshop on. IEEE, 2013.