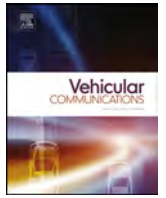




Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom



Spatial crowdsourcing with mobile agents in vehicular networks

Oscar Urrea^{a,*}, Sergio Ilarri^{a,b}

^a Department of Computer Science and Systems Engineering, University of Zaragoza, Zaragoza, Spain

^b I3A, University of Zaragoza, Zaragoza, Spain

ARTICLE INFO

Article history:

Received 5 October 2018

Received in revised form 10 February 2019

Accepted 20 March 2019

Available online xxxxx

Keywords:

Vehicular ad hoc networks

Spatial crowdsourcing

Mobile agents

Data management

Environment monitoring

ABSTRACT

In the last years, the automotive industry has shown interest in the addition of computing and communication devices to cars, thanks to technological advances in these fields, in order to meet the increasing demand of “connected” applications and services. Although vehicular ad hoc networks (VANETs) have not been fully developed yet, they could be used in a near future as a means to provide a number of interesting applications and services that need the exchange of data among vehicles and other data sources.

In this paper, we propose a spatial crowdsourcing schema for the opportunistic collection of information within an interest area in a city or region (e.g., measures about the environment, such as the concentration of certain gases in the atmosphere, or information such as the availability of parking spaces in an area), using vehicular ad hoc communications. We present a method that exploits mobile agent technology to accomplish the distributed collection and querying of data among vehicles in such a scenario. Our proposal is supported by an extensive set of realistic simulations that prove the feasibility of the approach.

© 2019 Published by Elsevier Inc.

1. Introduction

The continuous development of information and telecommunication technologies has brought advances to an increasing number of fields in industry. The automotive sector is not an exception and today's cars are more secure, respectful with the environment, and energy-efficient than never before, thanks to the inclusion of a wide range of computer devices and sensors. In this context, it is very relevant to provide cars with communication abilities so that they can exchange information with other cars (vehicle-to-vehicle communications) as well as with the road-side infrastructure and with other actors outside the roads (vehicle-to-infrastructure communications and vehicle-to-pedestrian communications). To accomplish this, the concept of vehicular network (also called vehicular ad hoc network or VANET) [1–3] has been proposed, as a set of vehicles driving along the roads of a certain geographic area, that can send/receive data to/from their neighbors by using short-range wireless communication devices. Vehicular networks open up many opportunities for the development of interesting applications, but they also imply a number of significant challenges from a data management perspective [4].

According to [5], luxury cars have more than 100 sensors. If conventional cars were equipped with appropriate sensors (e.g., temperature, humidity, pollution gases, etc.) to obtain information from their surroundings, and given that they are constantly moving along wide areas and that the number of vehicles could be rather high, the amount of information that can be retrieved and stored in this way could be considerable. Additionally, if the owners of those vehicles are willing to share those data and a suitable mechanism exists to manage and process them properly, then it would be possible to generate *knowledge* that could be used as a basis for the creation of other value-added services. Moreover, some vehicles could physically move to a spatial area to measure certain environment parameters there or just to bring a specific query processing task to that area (i.e., to transport some code for its execution in that area). This form of *spatial crowdsourcing* could be used, for example, to obtain driving patterns from the vehicles belonging to the VANET, based on the hour of the day or the weather conditions, and thus offer to the drivers the most optimal route to reach their workplace. Alternatively, as another example, this ability could be exploited for surveillance or environment monitoring (e.g., to measure the pollution or noise level in different areas of a city). Moreover, according to a report by ABI Research [6,7], the use of crowdsourcing to obtain data by connected, sensor-equipped vehicles can be used to improve already-existing applications (e.g., parking or traffic-related services), and can also bring new business

* Corresponding author.

E-mail addresses: ourra@itainnova.es (O. Urrea), silarri@unizar.es (S. Ilarri).

<https://doi.org/10.1016/j.vehcom.2019.03.004>

2214-2096/© 2019 Published by Elsevier Inc.

opportunities. In the same report, the number of cars contributing sensor data are estimated to grow to over 60 million by 2023.

In previous work [8], we have proposed the use of mobile agents for query processing in vehicular networks. In this paper, we present a system that obtains information stored on vehicles equipped with sensors while they are moving along the roads of a city or region based on the use of spatial crowdsourcing (unlike in [8]) to ease the process of locating and retrieving the required data. Our underlying research questions can be summarized as follows: can spatial crowdsourcing using mobile agents provide benefits for data collection in vehicular networks? how can we define a suitable and feasible approach? how could we quantify the benefits and costs of such an approach and the impact of different parameters, in different scenarios? As far as we know, this is the first study that analyzes the potential use of spatial crowdsourcing for vehicular networks using mobile agents (with the exception of our short paper presented in [9], where we introduce basic initial ideas and a very preliminary experimental evaluation with a single experiment). Moreover, an extensive experimental evaluation included in this paper shows the feasibility and the interest of the presented approach.

As an example of use case where the proposed approach can be applied, we can consider the case of air quality monitoring using low-cost air quality sensors aboard vehicles. For example, in the context of the European Project TRAFair – Understanding Traffic Flows to Improve Air quality (<http://trafair.eu>), one of the activities of the project tackles the deployment of an urban sensor network in several cities in Italy and Spain. In a scenario where vehicles carry air monitoring sensors, then it would be possible to perform flexible monitoring tasks to track air quality data about any area of the cities in a flexible way (rather than considering fixed monitoring areas covered by static sensors). In general, the spatial crowdsourcing approach described in this paper can be applied in any scenario where we need to monitor the environment by exploiting vehicles' sensors (e.g., for surveillance purposes [10], to create noise maps of the city [11], to obtain information about the availability of road-side parking spaces [12], etc.). As an example, SoundCity [13] is an Android application for mobile users, developed by Inria, that allows to measure one's personal exposure to noise pollution; however, it has not been developed for vehicular networks.

The rest of this paper is structured as follows. In Section 2, we describe basic background concerning intelligent vehicles, vehicular networks, and mobile agents. In Section 3, we describe a system for monitoring and querying data by using a VANET, as well as the use of spatial crowdsourcing techniques to improve the monitoring process. In Section 4, we perform several experiments to evaluate the performance of the proposed spatial crowdsourcing approach. In Section 5, we present some related work. Finally, in Section 6, we summarize our conclusions and present some lines of future work.

2. Technological context

In this section, we describe the background technologies that are used in our spatial crowdsourcing approach. Firstly, in Section 2.1, we focus on vehicular networks. Secondly, in Section 2.2 we describe the basics of mobile agent technology.

2.1. Vehicular networks

Enhancing existing vehicles with autonomous behavior and intelligent features, which has given rise to the concept of *intelligent vehicles*, is a relevant ingredient for the development of the so-called *Intelligent Transportation Systems (ITS)* [14,15]. These vehicles have the capability to sense the environment and communicate among themselves to share data relevant to the driver. A possibility

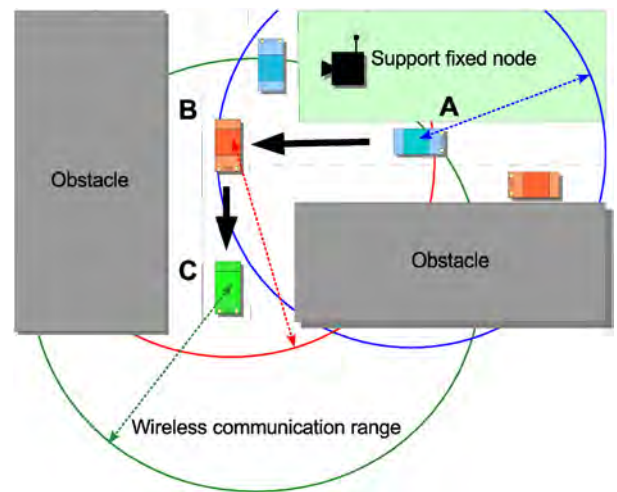


Fig. 1. Example of a VANET including a support fixed node and obstacles/buildings. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

to communicate these data is the use of cellular communications (e.g., 3G/4G), but they depend on a mobile telephony infrastructure, which does not cover all the areas (for example, inside tunnels and in certain shadow areas where the available coverage is limited) and have an economic cost that must be paid by the user to the mobile phone operator company. Other alternative options are publicly-available wireless networks deployed in certain cities by their local councils, but these are not present in all the cities and they usually do not cover all the city area either. Finally, we could consider the use of ad hoc communications, as an easy and non-expensive way of sharing resources would encourage the needed cooperation among users.

A Vehicular Ad Hoc Network (VANET) [2,3,16] is a highly-mobile network whose nodes are vehicles traveling along roads or highways. In a VANET, vehicles can establish connections with other nearby vehicles and, in this way, they can exchange different types of information. This makes the development of new applications relevant to drivers (as well as passengers) possible, such as applications related to security, monitoring, entertainment, or data sharing. The link between those nodes is established directly in a peer-to-peer way by using any of the available short-range wireless communication technologies (such as Wi-Fi based on the 802.11b/g/n standard or the more specific 802.11p, WiMAX, Bluetooth, Zigbee, etc.); specifically, the current communications standard for VANETs is WAVE [17,18].

Whereas VANETs open up new opportunities for the development of interesting applications, they also require the design of appropriate techniques from the perspective of data management [4]. Most difficulties are due to the high mobility of vehicles and the limited amount of time during which two vehicles may be within communication range of each other. Since the vehicles are constantly moving (usually at high speeds) and the wireless signal can be blocked by buildings and other obstacles, these connections may last only for a few seconds and therefore the network topology is constantly changing. For example, when a vehicle needs to communicate data to a specific target vehicle, a multihop routing protocol [19] must be usually used (e.g., see Fig. 1, where vehicle B acts an intermediate relay point for the data sent from vehicle A to vehicle C).

2.2. Mobile agents

A mobile agent [20–22] is a software program (composed by code and data) that can move from one computer or mobile de-

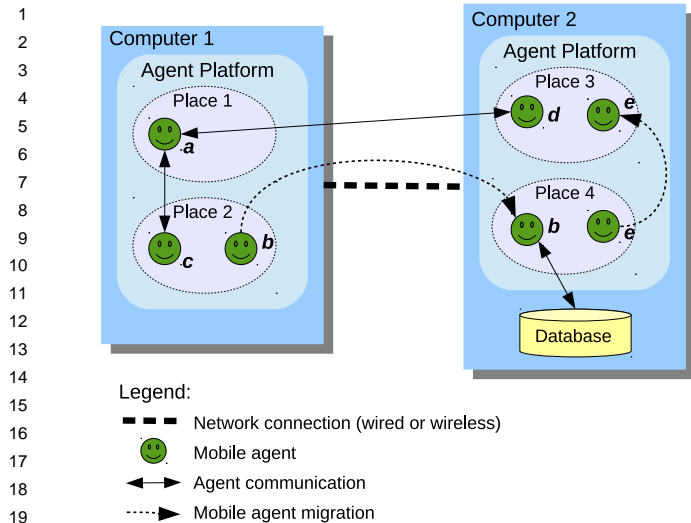


Fig. 2. Different elements in a mobile agent architecture.

vice to another by using a network connection. That is, while it is running on a certain computer or device it can pause its execution, transfer itself to another computer or device, and resume its execution in the new destination, as long as the transfer process ends successfully. For this to be possible, it is necessary that all the computers/devices involved (i.e., the origin computer/device and the target computer/device where the agent moves) execute a light middleware called the *mobile agent platform* [21] (e.g., Aglets, Voyager or SPRINGS/GeoSPRINGS [23,24], to cite some examples), which manages the movements of the agents and offers them a number of additional services [22] related to security, efficiency, communication, etc.

In Fig. 2, an example that summarizes the basics of a mobile-agent based architecture is shown. There are two computers that can communicate by means of a network connection. *Computer 1* executes a mobile agent platform with two *places* or execution environments for mobile agents, *Place 1* and *Place 2*. Similarly, in *Computer 2* there are two other places called *Place 3* and *Place 4*. An agent executing in *Place 1* (*Agent a*) establishes a communication with another agent (*Agent c*) that is executing in a different place inside the same platform (*intra-platform* communication), and also communicates with *Agent d*, that is placed in a different computer (*Computer 2*) using the network. Another agent (*Agent b*) is mobile and decides to change its execution environment, so it moves (by means of a network connection) to *Place 4*, which is hosted in a different computer (*Computer 2*), and then establishes a connection to a local database (which might be not accessible from a remote computer) to query data. Similarly, *Agent e* is also mobile and performs a migration to another place inside the same computer (*Place 3*).

We think that mobile agent technology can provide relevant advantages for VANETs. Indeed, in a previous work, we have studied the potential use of mobile agents for data management in vehicular networks and analyzed the performance and benefits offered by that technology [8]. In these scenarios, the vehicles have computers capable of executing a mobile agent platform, as well as some wireless communication device. In this way, mobile agents can execute in such platforms and move from one vehicle to another as they travel, in order to process data in the distributed environment. As an example of application, mobile agents can be used to search and filter data that can be captured by different vehicles located along the streets of a city, using their sensors. The agents can reach spatial areas of interest by moving to them in two different ways, as shown in Fig. 3: through *agent transmission*, where

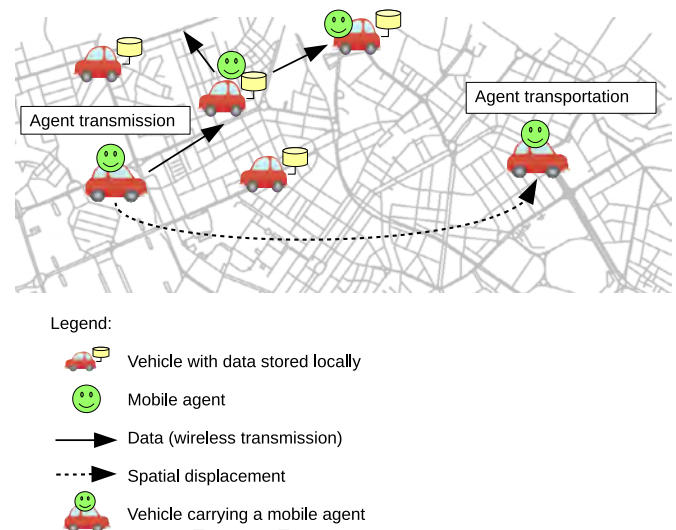


Fig. 3. Mobile agents in a VANET.

the mobile agent uses wireless communications to transfer itself from the execution environment located in an origin vehicle to another destination vehicle, and through *agent transportation*, where the mobile agent is physically transported to a different location by the vehicle that carries the computing device where the agent platform is being executed.

3. Monitoring an interest area by using VANETs and mobile agents

One proposed application for VANETs is the monitoring of certain parameters in an area by the moving cars present in it [25], and the possibility of executing queries about information related to that area (or any sub-area), that are processed by using the most updated data present in the VANET.

In this scenario, a percentage of equipped vehicles are assumed to carry a computer or device which collects data about its surroundings, that may be of interest for other drivers or even for users outside the roads. As an example of the first case, we could mention information about available parking spaces, the prices and status of gas stations or mechanical garages, the presence and occupation of recharging points for electric cars, the traffic density, etc. Besides that, it would also be very interesting if the vehicles carried different types of sensors to collect environmental information in the different places where the vehicle may travel to (e.g., to measure the ambient temperature, noise, solar radiation, or pollution gases such as CO₂, NO₂, O₃, etc.).

In order to make this monitoring process more effective and the collected data more representative of the area, it is necessary to obtain readings from sensors in different places that, in addition, should be distant enough from each other. A simple way of achieving this is by dividing the interest area in portions or cells with the same surface following a grid layout. Then, samples can be taken once the involved sensors are located within each of those cells, which implies that if the sensors are carried by vehicles, they should visit all those places at some time, in order to obtain measured values from all the cells.

3.1. Basic monitoring approach using mobile agents

Traditional sensing approaches based on the use of stationary sensors and/or sensing stations can be expensive, which motivates the development of alternative or complementary solutions [26–28]. We argue that by using mobile agents in a VANET

Table 1
Summary of advantages and disadvantages of several hop strategies described in [8].

Hop strategy	Description	Advantages	Disadvantages
EUC	Euclidean distance	Simple to compute	It does not consider the street layout The shortest distance is the only factor considered
ANG	Frontal angle	It considers the vehicle's heading	It does not consider the street layout The direction of the vehicle is the only factor considered
EP	Encounter probability	It considers the vehicle's heading It tries to estimate the probability that the vehicle will reach the target area	It does not consider the street layout
MAP	Street map distance	It considers the street layout	It needs a digital map It is computationally costly
MapTraj	Trajectories using maps	It considers the vehicle's trajectories and the street layout	It needs a digital map The planned trajectories must be known
Optimal	Optimal route	It returns the optimal sequence of hops	It is unrealistic (based on information that cannot be available)

it is possible to flexibly monitor potentially any geographic area required (*interest area* or *target area*) by using conventional vehicles and without the need to deploy a fixed sensor infrastructure. The whole monitoring process would consist of the following four steps:

- 1. Submission of the monitoring task.** First, the user defines the parameters to be monitored within the bounds of a spatial area (e.g., the temperature in the city center during the afternoon), and immediately a mobile agent starts its execution with the defined monitoring parameters.
- 2. Travel to the interest area.** In the second step, the mobile agent travels towards the interest area (in the example, the city center) by using the vehicles in the VANET to hop from one vehicle to another, using a specific hop strategy [8] (see Table 1), until it reaches the destination (transportation via wireless communications). Every time the mobile agent arrives at a vehicle, the next hop may not occur immediately, since the number of potential vehicles to hop to might be scarce (in the worst case, there might even be no other vehicles in the surroundings). In such cases, the mobile agent will stay in the same vehicle while waiting for a more suitable one. There is also the possibility of moving from one place to another by just staying in the current vehicle (transportation via locomotion). This combination of transportation facilities is sometimes called *carry-and-forward* or *store-carry-and-forward* in the context of vehicular networks [29].
- 3. Data collection.** Once the mobile agent reaches the interest area, it will visit the different cells into which the area was partitioned. Upon its arrival to each of them, it will read the required sensor data and it will continue its travel to the next cell, repeating this process until all (or a preconfigured portion) of them have been visited. In order to reach each spatial cell in the grid, the mobile agent will likely need to hop from one vehicle to another in the same way as in the previous step. Notice that the vehicle where the agent is executing may move out of the interest area, and therefore in that case the mobile agent will need to find a way to come back to that area.
- 4. Return of the results.** Finally, in the last step, the mobile agent returns the data collected in the interest area to its originator (e.g., a static node or a vehicle); again, ad hoc communications can be used to reach the place where the query originator is located. This stage of the process is challenging, as the originator could be a vehicle that may keep moving and thus constantly changing its spatial position. Thus, in this case, the mobile agent must estimate the position of the originating vehicle at every moment and, in case the estimation is wrong, keep re-evaluating and "chasing" the vehicle until it is successfully reached. The estimation of the position of the originating

vehicle can be calculated according to its last known position and its intended destination (which will likely be the interest area). Such estimation will surely not be an exact position, but instead an area where the vehicle is expected to be present. The wider the area, the higher the existing uncertainty for the mobile agent to find the originator [8]; if needed, additional strategies could be used, for example, based on the use of vehicle-to-infrastructure communications through Road Side Units (RSUs) [4] or mailboxes to store the query results at fixed locations from which they can be easily retrieved [30].

One advantage of using mobile agents is that, instead of sending only data, they can carry along with them the logics or algorithms needed to acquire and filter out those data. Also, they can act as intelligent entities that can decide at every moment how to proceed according to the information they receive from their surroundings. This feature is especially useful in the scenario we are describing, in at least two aspects. Firstly, the mobile agent must reach the interest/target area by using the vehicles as intermediate nodes, being necessary to hop from one to another until the destination is reached. However, as explained before, one difficulty of VANETs is that their nodes move constantly, so it is not possible to use classic routing tables or compute the route to follow in advance; instead, the next node to be transferred to must be constantly evaluated just a few seconds before hopping to it, by using the information available to the mobile agent at that moment (its position and heading, the remaining distance to the target, etc.). Secondly, another advantage of carrying the logic with the mobile agent is that the agent can carry the specific algorithm needed to acquire and filter out the data involved in the monitoring process and thus minimize the amount of data transported (only the required data are carried). For example, the data acquisition task could be different if the monitoring is about the number of available parking spaces in a neighborhood than if it is about the average level of pollution gases in the area. In this way, instead of using a generic and potentially-heavy algorithm to acquire and filter out all the possible types of monitoring parameters, a more specialized and *small* code can be used for each case.

We believe that this is an interesting application for VANETs, that makes it possible to extract valuable information by using the vehicles as moving sensing platforms and the available sensors in a more efficient and dynamic way, if we compare it to the traditional solution based on static measuring stations deployed at fixed locations along fixed and static interest areas. Also, the use of mobile agents is adequate since their flexibility and autonomy makes them a suitable solution for a fast-changing scenario such as a VANET. The mobile agents encapsulate code to intelligently take decisions, based on available data about the environment, with the goal of overcoming difficulties appearing in a VANET environment, such as the limited time available for exchanging data (e.g., if there

is not enough time to hop from one vehicle to another, due to the movements of the vehicles, the jump will fail and the mobile agent will retry or take an alternative decision). To reinforce this idea, we performed experiments to evaluate the basic monitoring approach described in this section, by using a mobile agent VANET simulator [31], which showed the feasibility of this proposal [8].

3.2. Disadvantages of the basic monitoring approach

The previously-described monitoring process has some drawbacks. For example, if the traffic density is too low, the mobile agent may not find a path to reach the interest area in a reasonable time, or even it may not reach it at all. Similarly, if a vehicle visited by a mobile agent leaves the area, or just parks in an underground parking before the agent hops away, it could get *trapped* or *lost* and the monitoring process would be interrupted.

Two possible strategies to try to limit these drawbacks are the setting of a timeout and the use of agent's clones. With the first strategy, once the mobile agent with the required monitoring parameters is launched, a timeout is set, and if the agent does not return any result before the time limit expires, the agent is assumed to be lost and launched again. In the second strategy, the mobile agent creates a number of copies of itself to increase the probability of successfully reaching the interest area by using different alternative routes [8]. This strategy increases the use of the network bandwidth, but maximizing the likelihood of obtaining a result can compensate this cost.

A potential crowdsourcing strategy could be that certain drivers (e.g., commuters) could make available the route that they will follow, to help the mobile agent to decide if a vehicle is worth to hop to or not. For privacy reasons, the complete routes are not expected to be provided with all the details. Indeed, this strategy does not necessarily require a full disclosure of the travel intentions of the driver. In general, only a minimum amount of information is needed, for an agent to know if a certain street is in the intended route or not. In exchange of this information, the driver can be compensated with *virtual money* (e.g., discount points for gas stations, virtual coins that can be used to pay for other useful information or even be exchanged by real money, etc.). This virtual money could be equal to 0 in the case of altruistic drivers that share their routes without requiring a profit, or in the case of public transportation vehicles that follow well-known and regular routes (such as bus or tramway lines).

Another strategy for enhancing the behavior of the monitoring process, which is the focus of this paper, is the use of spatial crowdsourcing techniques. For example, in a certain scenario, the mobile agent might find it difficult to reach certain areas of a city, but the user that starts the monitoring process may be willing to pay a certain amount of virtual money for some help from other vehicles. So, the mobile agent gets the ability to negotiate, with the drivers in the VANET, a way to reach the interest area faster or straighter than in the usual way. Recalling the steps followed to complete the monitoring process (described in Section 3.1), when the mobile agent hops to a vehicle in the second step of the process, it would not only look for other vehicles that seem to travel towards the interest area, but may also try to negotiate to be physically carried there in exchange of a certain amount of virtual money.

3.3. Approach based on the use of spatial crowdsourcing

As defined in [32], spatial crowdsourcing implies "location-specific tasks that require people to physically be at specific locations to complete them". With spatial crowdsourcing, a vehicle could be willing to physically transport a mobile agent closer to

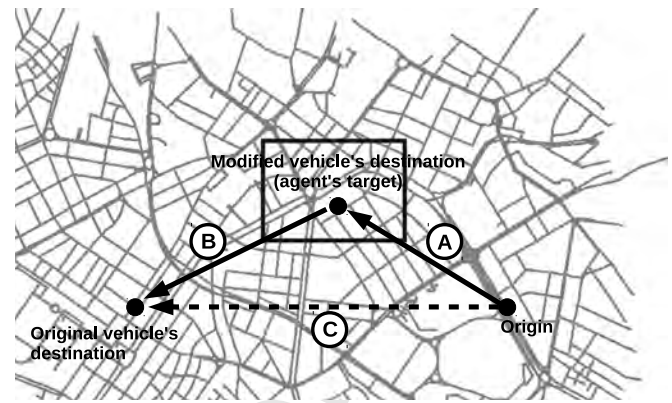


Fig. 4. Example of social cost calculation for a single vehicle.

its destination; we call such a vehicle a *collaborator* or *collaborating vehicle*. In this section, we describe our approach. Firstly, in Section 3.3.1, we introduce the concept of social cost. Secondly, in Section 3.3.2, we present the role played by virtual money in the spatial crowdsourcing approach. Thirdly, in Section 3.3.3, we present the basic workflow that describes the spatial crowdsourcing approach applied to travel to a target area. Finally, in Section 3.3.4, we describe the use of spatial crowdsourcing for data collection.

3.3.1. Social cost

Spatial crowdsourcing comes at a cost, as it usually requires an effort by the collaborating vehicles. Specifically, we define the *social cost* as the time needed by the collaborating vehicles to deviate from their original routes to carry the mobile agent towards its destination and, once the agent leaves the vehicle, to recover their previous destinations and continue traveling towards there. In other words, the social cost is the time that collaborators need to invest to help the mobile agent. More specifically, as shown in Fig. 4, the social cost is computed as the time needed by the collaborator to reach the agent's destination (A), plus the time needed to travel back to the driver's original destination (B), minus the time that it would have taken to directly follow the original route (C). If the agent would have not required this help, the collaborator would have not spent extra time traveling, so this cost needs to be compensated somehow.

3.3.2. Virtual money

Specifically, to encourage the collaboration from other vehicles, we propose an approach where a mobile agent can pay *virtual money* to them in exchange of being transported closer to the target area: the agent has a specific budget to process the query and tries to use that budget in the best possible way to reach its objectives and at the same time minimize the amount of money spent. The drivers of the collaborating vehicles could later use the virtual money received to pay other vehicles for similar services. The negotiation required to establish if a driver is willing to alter his/her current trajectory to follow the agent's objectives could be performed automatically whenever the agent is looking for the next vehicle to hop to; in order to accomplish this, the drivers of the area would state previously if they would like to participate in these exchanges and to what extent they are willing to detour from their route to bring the mobile agent closer to its destination. As an alternative, the driver could be explicitly asked if he/she is willing to follow a specified detour.

Several payment schemes could be considered. In our prototype, the amount paid by an agent is directly proportional to the time spent by the vehicle carrying the agent and there is no minimum fare, as shown in Table 2. The agent is assumed to have

Table 2
Payment model: payment from an agent to a collaborator.

	Initial budget	Final budget
Vehicle where the mobile agent is created (V_{origin})	$C_{V_{origin}}$	$C_{V_{origin}} - C_{agent}$
Mobile agent	$C_{agent} (\leq C_{V_{origin}})$	$C_{agent} - t$
Vehicle transporting the agent during t time units	$C_{V_{collaborator}}$	$C_{V_{collaborator}} + t$

an allowable budget (provided by the vehicle where the agent is created, that is, the monitoring node that wants to retrieve data from the interest area) and the agent will pay one unit of virtual money per every time unit (i.e., every second) spent in the vehicle acting as a “taxi” (i.e., following the route required by the agent). This route will likely be different than the one the vehicle was following originally. However, it might also happen that a driver obtains virtual money even if he/she does not really need to modify his/her trajectory to transport the agent (because the route required by the agent already matches the future trajectory of the vehicle) or that he/she gets a considerable amount of money in exchange of a slight deviation from the previous trajectory. This is unavoidable, as the intended trajectories of the drivers are unknown to the agent, and so the agent cannot know that it could achieve its objectives just by staying in the vehicle, even without paying for the vehicle to change its trajectory. Even though the basic payment scheme proposed is only based on transportation time, other additional costs could be easily added, if needed (e.g., the cost of traversing road tolls in case of interurban trajectories).

3.3.3. Basic workflow of the spatial crowdsourcing approach

The basic workflow of the proposed spatial crowdsourcing approach is shown in Fig. 5, where SC is used as an abbreviation of spatial crowdsourcing. The following are the main steps of the process:

- In order to save virtual money, the mobile agent will first try to reach the interest area by hopping from one vehicle to another without asking vehicles to collaborate by modifying their original routes. Only if the speed at which the mobile agent approaches its destination (*approaching speed*) is lower than a certain value (1), which we call the *minimum speed threshold*, the agent will look for nearby collaborating vehicles (2) and will negotiate with them if they are willing to carry the agent. More specifically, the *approaching speed* of a mobile agent (ag) to the target area at time instant t is defined as indicated in Equation (1), where $d(ag, area, t)$ represents the distance between the agent (ag) and the target area ($area$) at time t , and $d(ag, area, 0)$ represents the initial distance between the agent and the target area; for simplicity of computation, we consider the Euclidean distance, but alternative distance metrics could also be considered. It should be noted that this approaching speed may not be the same as the speed at which the agent is traveling. For example, the agent may be in a vehicle moving at 50 km/h, but if its trajectory is inside a street located parallel to the target area then the approaching speed would be 0, as that trajectory does not bring the agent closer to the target area. Similarly, if the mobile agent is traveling in the opposite direction to the target area, its approaching speed will have a negative value.

$$ApproachingSpeed(ag, t) = \begin{cases} \frac{d(ag, area, t) - d(ag, area, 0)}{t}, & \text{if } t > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The minimum speed threshold must be either a positive value or 0; a value of 0 for the minimum speed threshold means

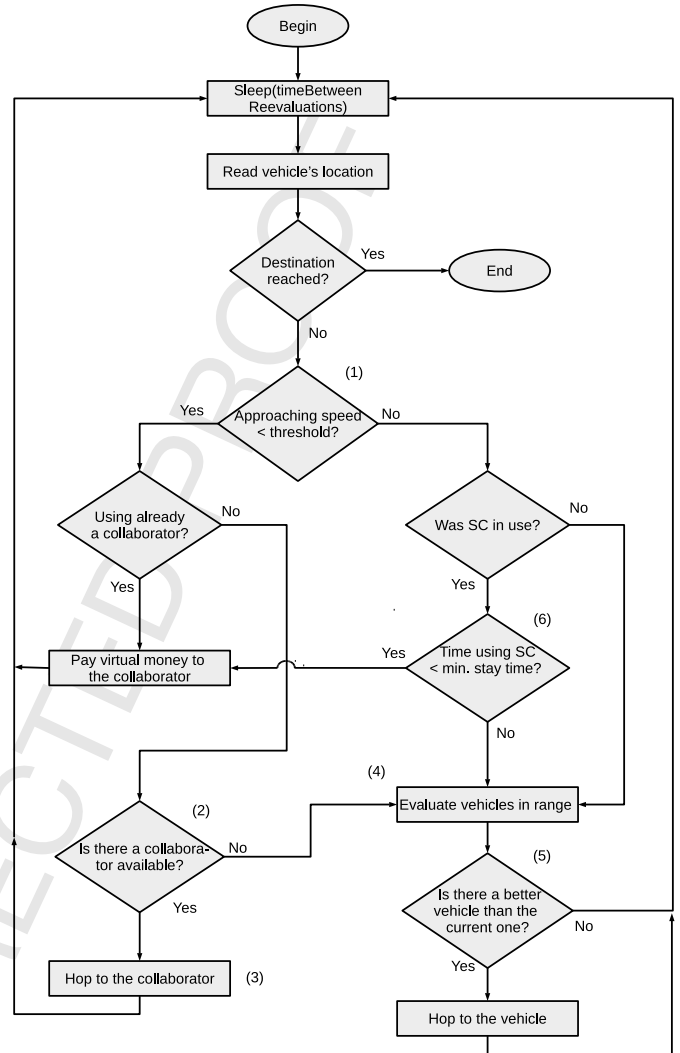


Fig. 5. Basic workflow of the SC approach followed by a mobile agent to travel to an area.

that the agent will never ask for help to potential collaborating vehicles (i.e., no spatial crowdsourcing will be used).

- If the potential collaborator accepts (3), then the mobile agent will stay in the vehicle and will not hop off it unless the approaching speed exceeds again the threshold value.
- If the driver does not agree to deviate from his/her trajectory (4), then the mobile agent will continue trying to reach the destination with the usual (non spatial crowdsourcing) approach, by hopping from one vehicle to another (5), while at the same time it keeps looking for a potential collaborator.
- If the mobile agent is already using a collaborator but it finds a suitable vehicle nearby, it can jump there (6). However, a *minimum stay time* in the vehicles can be considered for the following reason: for low values of the minimum speed threshold, once moving by staying in a collaborator it is easy to reach the threshold, and thus the agent may immediately hop to a more promising vehicle (i.e., one that approaches the target

Table 3
Experimental settings.

Parameter	Default value
City maps	Madrid, Barcelona, and Zaragoza
Size of the interest area	0.25 km ²
Distance to the interest area	2 km
Density of vehicles	Medium (100 vehicles/km ²)
Speed of the vehicles	50 km/h \pm 10% (random variability)
Mobility model	Pathway mobility model
Hop strategy	MAP (map distance)
Data collection delay	5 s
Data to collect (number of samples)	6 (1 from each cell of a 3 \times 2 grid)
Data collection timeout	3 minutes
Total agent size (code, status, and amount of data carried by the agent)	200 KB
Communication range	250 m
Mobile agent's hop delay	1 s
Minimum approaching speed threshold	5 km/h
Strategy used for data collection with SC	SCCA
Buildings block communication signals	Yes
Percentage of potential collaborators	50%
Minimum stay time in a collaborating vehicle	0 s
Size of the cold area (low-traffic area around the interest area)	0 m
Percentage of trajectories crossing the cold area	15%
Time limit	20 minutes

area faster). This may render the behavior of the mobile agent somewhat unstable, and in addition it may require a higher use of bandwidth, since the agent could be constantly hopping to new vehicles. So, in order to limit this effect, a hysteresis value for the agent's stay time (minimum period that the mobile agent must stay in a collaborating vehicle before hopping to others) is considered. In this way, the bandwidth used and the likelihood of switching continuously between spatial crowdsourcing and hopping among vehicles can be reduced. The impact of the value of this parameter is evaluated in Section 4.4.3.

So, in our proposal, we combine the basic monitoring approach, based on hopping from one vehicle to another by exploiting wireless communications, with spatial crowdsourcing. It should be noted that the switching between these two methods can occur several times at any moment during the trip to/from the target area. For example, if the agent is traveling too slowly (maybe because it is in a low-traffic street/area that makes it difficult to find other vehicles traveling towards the desired destination), it can use a collaborator to leave that street/area faster, and once its speed reaches a higher value, use again the method of hopping from one vehicle to another until the destination is reached.

It should be noted that we advocate a completely decentralized approach, based only on the use of opportunistic ad hoc communications with nearby vehicles. In this way, there is no need of a central entity with an overall view of the network in order to take global decisions and control the distribution of crowdsourcing tasks. Instead, the mobile agent itself takes its own decisions based on local knowledge about its environment and by interacting only with vehicles in its surroundings.

3.3.4. Spatial crowdsourcing for data collection

Spatial crowdsourcing could be used not only for traveling to the interest area but also for data collection once in the interest area. In order to know to which extent the spatial crowdsourcing is useful or not for the data gathering process, we have tested two versions of the mobile agent with different behaviors exhibited once the agent reaches the target area:

- In the first version, that we call *SCCA (Spatial Crowdsourcing Collecting Agent)*, or *SC/SCCA*, the mobile agent always uses spatial crowdsourcing during the data collection phase (i.e., to

travel to the target area cells and to return to the area if the vehicle leaves it during the data collection).

- In the second version, that we call *PHCA (Pure Hopper Collecting Agent)*, or *SC/PHCA*, the mobile agent never uses the help of collaborators during the data collection phase; therefore, the agent tries to reach the cells within the target area by only hopping from one vehicle to another.

The difference between these two versions only affects phase 3 of the monitoring approach (see Section 3.1), since in both versions the mobile agent uses spatial crowdsourcing to travel to the area as well as to return the results to the originator vehicle. In Section 4.3, we show experimentally that the performance of the *SCCA* approach is better than one where spatial crowdsourcing is not used during the data collection phase.

4. Experimental evaluation

In this section, we present the experiments that we have performed to test the feasibility of the proposed monitoring approach in VANETs using mobile agents with spatial crowdsourcing abilities. For evaluation, we have used the MAVSIM simulator [31], that allows the simulation of both mobile agents and traffic in a realistic way by using real road maps extracted from OpenStreetMap [33]. Among its various functionalities, we can highlight the simulation of situations where the wireless signal is blocked by obstacles such as buildings, as this is more realistic.

4.1. Experimental setup

The main simulation parameters considered in the experiments carried out, with their default values, are shown in Table 3. In Section 4.1.1 we describe the main experimental parameters, and in Section 4.1.2 the metrics measured and the purpose of the different experiments performed.

4.1.1. Main parameters in the simulations

A few parameters included in Table 3 deserve further explanations:

- *City maps*. We have used map scenarios corresponding to different cities of Spain (Madrid, Barcelona and Zaragoza, as shown in Fig. 6), that exhibit different features regarding the presence of long straight road segments and short and/or

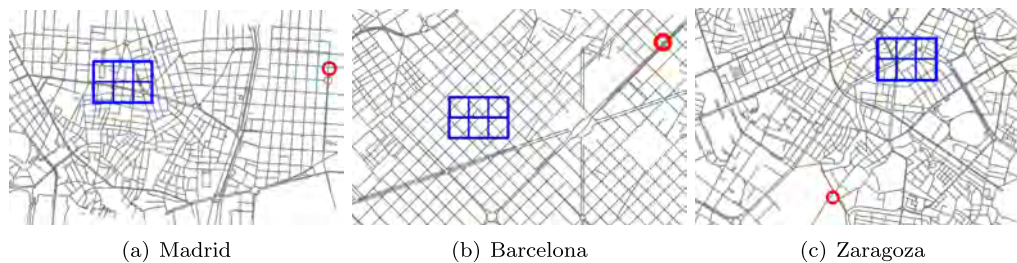


Fig. 6. Key section of the map fragments used for evaluation. The grid square shows the designated interest area, and the small circle is the point where the monitoring process is launched.

Table 4
Different metrics of the road graphs in the maps.

Map	Length (km)	Surface (km ²)	Road density (km / km ²)	Avg. edge length (m)	Vertices (V)	Edges (E)	δ
Madrid	162.49	24.95	6.51	3.72	3732	4367	2.34
Barcelona	217.93	14.62	14.91	3.92	5060	5555	2.20
Zaragoza	253.3	18.76	13.50	3.54	6030	7153	2.37

curved segments, that neither favors nor harms the wireless signal propagation. In this way, we assure that our proposed approach is generic enough to perform consistently in different topologies and that its behavior is not linked to any one in particular.

In Table 4, some metrics characterizing the map graphs are shown. The *average nodal degree* (δ) is defined as $2 * |E|/|V|$, where E is the set of edges/links in the graph and V is the set of vertices/nodes, and it is used as a measure of the mesh degree [34]. The *length* represents the total length of the roads in the map, the *surface* is the area covered by the map (in square kilometers), the *road density* is the ratio of road kilometers per surface unit (in square kilometers), and the *average edge length* is the average length of a road segment in the map.

- *Density of vehicles.* The vehicle density can be defined as the number of vehicles present per surface unit [35] and it can be measured in terms of vehicles per square kilometer. Another similar vehicle density measure unit is the number of vehicles per road length unit [36], that is, vehicles per kilometer or per mile; this unit is useful for measuring the traffic flow in a mostly-linear road topology (such as a highway), but it is not suitable for urban scenarios, where vehicles can communicate not only with others present in the same street but also with vehicles traveling along other nearby parallel or perpendicular streets. Of course, this is not always possible due to the presence of obstacles such as buildings but, even in that case, opportunities can appear to perform the communication in open city areas such as main squares or street junctions. For these reasons, we use the *vehicles per square kilometer* (or *vehicles/km²*) unit, since it is more appropriate for urban scenarios like those we are testing.
- *Speed of the vehicles.* The experiments are performed in an urban environment, so the speed of the vehicles is set to 50 km/h, which is a typical speed limit value in many countries for driving in a city. We introduce a random variability of $\pm 10\%$ to that value, since it is highly unlikely that all the vehicles move exactly at the same speed. These settings are the same ones presented in [8].
- *Hop strategy.* The hop strategy refers to the algorithm used by the mobile agent when it travels by hopping among vehicles and it must choose, among several candidate vehicles, the one to hop to (instead of just being physically carried by a crowdsourcing collaborator). The agent needs to choose the vehicle that it estimates that will travel closer to the agent's

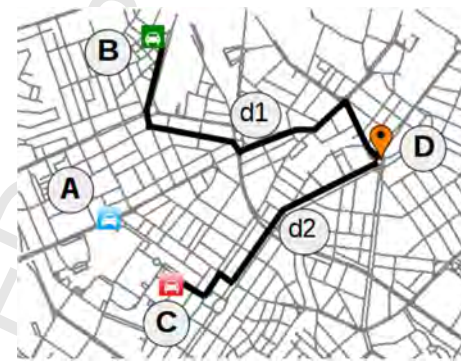


Fig. 7. The MAP hop strategy.

destination with the highest probability. For the experimental evaluation, we have chosen by default a strategy that uses as a criterium the remaining distance to the destination (*map distance*), computed by adding the lengths of the streets to traverse in the road network. This algorithm is called the *MAP hop strategy* and an example is shown in Fig. 7: The mobile agent in vehicle A travels towards its destination in D, and it must decide whether it hops to vehicle B or C; both vehicles have a similar straight (i.e., Euclidean) distance to D, but if the routes are computed following the streets, the distance $d1$ from B to D is longer than the distance $d2$ from C to D; therefore, the mobile agent will choose the vehicle C, since it is the one with the shorter route to the destination. This hop strategy has been shown to provide good results [8].

- *Mobile agent's hop delay.* This is the time needed by the mobile agent to complete a hop from one vehicle to another, which implies transferring the agent's code, status and the data collected by the agent. We assume that in our test scenario the wireless technology used is the widespread 802.11g, that operates in ideal conditions at a maximum speed of 54 Mbps. However, a large number of factors can reduce the effective speed of the data transfer, such as the signal attenuation due to the distance and interferences and collisions with other devices operating in the area, as well as the processing overhead due to the packing and unpacking of the mobile agent's code and data (serialization and deserialization before and after transmission, respectively). Therefore, we do not assume a best-case scenario and establish the travel time to one second. Field experiments that we have performed with real smartphones and tablets confirm that this value is reasonable [8].

- 1 • *Data to collect.* Without loss of generality, in the experiments
2 we assume that the interest area is partitioned based on a $3 \times$
3 2 grid, which makes a total of six cells, which the mobile agent
4 will have to visit to recover data from them.
- 5 • *Data collection delay.* When the mobile agent arrives in a spa-
6 tial cell, it reads the corresponding vehicle's sensor, processes
7 the value read, and stores it (if needed), and this action is as-
8 sumed to take 5 seconds, which is a pessimistic estimation.
9 During this interval, the mobile agent will not hop to any
10 other vehicle until the sensor reading, processing, and storage,
11 is complete. Therefore, the mobile agent not only spends time
12 reaching the interest area and returning from it, but also read-
13 ing, processing and storing values measured by sensors in the
14 different cells.
- 15 • *Data collection timeout.* To avoid a potential situation where the
16 data collection task takes too long, a timeout of 180 seconds is
17 set. Reaching this timeout will cause the mobile agent to finish
18 the task and start the process that intends to return the data
19 to the originator vehicle or node where the monitoring pro-
20 cess started. In this case, it might happen that the amount of
21 data collected by the agent is smaller than the desired quan-
22 tity. Therefore, the amount of collected data is a performance
23 metric to consider.
- 24 • *Time limit.* To avoid situations where the monitoring process
25 takes too much time to complete (for example, if an agent
26 finds an unusually hard-to-solve situation), an overall time
27 limit of 20 minutes is set. If this limit is reached, the simu-
28 lation is aborted and its results are ignored, but the failure
29 is counted and used to measure the reliability of the algo-
30 rithm under the conditions that were being simulated. Thus,
31 the time limit represents the maximum overall time that the
32 user is willing to wait to obtain an answer.

33
34 In the experiments presented throughout this section, we use
35 the default values shown in Table 3, using the three city maps,
36 and vary one of the parameters, which will label the X-axis of the
37 corresponding experimental figures. Thus, in Section 4.2.1 we vary
38 the size of the cold area, in Section 4.2.2 the initial distance to the
39 interest area, in Section 4.2.3 and Section 4.4.2 the density of ve-
40 hicles, in Section 4.3 the minimum approaching speed threshold,
41 in Section 4.4.1 the percentage of potential collaborators, and in
42 Section 4.4.3 the minimum stay time in a collaborator. With this
43 setup, we repeat every simulation 50 times (as in [8]), with dif-
44 ferent random starting positions for the vehicles, and compute the
45 average of the results obtained.

47 4.1.2. Metrics evaluated and purpose of the experiments

48 The main metrics considered for evaluation purposes are the
49 following:

- 50 • *Time.* An obvious metric to evaluate is the amount of time
51 needed by the mobile agent to reach the interest area and/or
52 perform the monitoring task.
- 53 • *Amount of collected data.* This metric indicates the number of
54 cells in the interest area that the mobile agent visits and reads
55 data from. In our settings there are 6 cells in the interest area,
56 as we divided the interest area in a 3×2 grid, so we count
57 the number of such cells that the mobile agent visits within
58 the timeout of 3 minutes established for the data gathering
59 phase. If this timeout is reached, then the mobile agent stops
60 the data collection task and returns to the origin point with
61 the data collected until that moment.
- 62 • *Number of hops (bandwidth use).* This metric counts the number
63 of times that the mobile agent transfers itself from one vehi-
64 cle to another one by using the wireless connection (i.e., with
65 the number of hops performed by the agent), and therefore it can

66 be used as a measure of the bandwidth used by the whole
67 process.

- 68 • *Virtual money spent.* This metric represents the payments made
69 by the mobile agent in exchange of help from the collaborating
70 vehicles.

- 71 • *Social cost.* As defined in Section 3.3, the social cost represents
72 the extra time invested by collaborating vehicles in helping the
73 mobile agent to perform its task, as compared to a situation
74 where no help is provided (i.e., all the vehicles follow their
75 intended routes).

76 With the payment scheme considered in this paper (where
77 the agent pays in proportion to the stay time, as explained
78 in Section 3.3), we expect a quite direct relation between the
79 social cost experienced by the collaborating vehicles and the
80 virtual money spent by the mobile agent and paid to the col-
81 laborating vehicles as a compensation. This means that the
82 experimental figures measuring both metrics are expected to
83 show similar trends. Nevertheless, notice that the relation can-
84 not be computed as a direct proportion, as the social cost does
85 not depend on the time spent by the collaborating vehicle car-
86 rying the agent but on the extra time needed to reach its
87 intended destination despite the route deviation performed to
88 help the agent. With other payment schemes (e.g., if the agent
89 needs to pay a minimum fare every time it takes a collabo-
90 rating vehicle), the relation between these two metrics (social
91 cost and virtual money spent) can become blurred. Besides, it
92 might happen that the collaborating vehicle's final trajectory
93 may actually be very similar to the initially-intended one, as
94 explained in Section 3.3: in this case, the virtual money spent
95 by the agent may be higher than the social cost actually in-
96 curred by the collaborating vehicle.

- 97 • *Reliability.* The reliability of the monitoring is defined as the
98 percentage of simulations that ended successfully within the
99 simulation time limit established. This metric is very impor-
100 tant as a complement of all the others. The reason is that all
101 the previous metrics are computed by considering only the
102 results of the successful simulations. Indeed, the metrics for
103 unended simulations are not available, as the corresponding
104 experiment did not end. It should be noted that the alternative
105 of computing partial metrics or default metrics for unended
106 simulations would be confusing; for example, considering a 0%
107 of collected data for an unended simulation is misleading (in
108 that case, it is inaccurate to say that the mobile agent collected
109 0% of the data; e.g., given just a few more seconds of simu-
110 lation the simulation might have ended and the agent might
111 have returned a good percentage of collected data); as another
112 example, the number of hops so far in an unended simulation
113 might be low but this might be only due to the fact that the
114 simulation has not ended (given enough time for the simula-
115 tion to end, the number of hops may actually turn out to be
116 high).

117 To evaluate those metrics we performed two sets of experi-
118 ments, whose purpose and conclusions are summarized in Table 5.
119 The first set of experiments, shown in Section 4.2, is devoted to
120 comparing the proposed spatial crowdsourcing approach with an
121 approach without spatial crowdsourcing to travel to the interest
122 area. The second set of experiments, described in Section 4.3, fo-
123 cuses on determining a suitable value for the minimum approach-
124 ing speed threshold. The third set of experiments, shown in Sec-
125 tion 4.4, evaluates the impact of several parameters, such as the
126 density of vehicles and the number of available collaborators, on
127 the monitoring process.

Table 5

Summary of the experiments performed and their purpose.

Experiment	Section	Parameter	Purpose	Conclusions
Experiments Showing the Interest of Spatial Crowdsourcing to Travel to the Interest Area	4.2		Evaluate the potential benefits of spatial crowdsourcing (SC) to help a mobile agent to reach an <i>interest area</i> .	The use of SC can help the mobile agent particularly in low-traffic scenarios. Besides, even in scenarios with good traffic density, the use of SC does not have a negative impact.
Impact of the Size of a Low-Traffic Area	4.2.1	Size of the cold area	Evaluate the impact of the size of a <i>cold area</i> when SC is used or not.	The use of SC is particularly beneficial in terms of time and bandwidth for large cold areas.
Comparison Varying the Initial Distance to the Interest Area	4.2.2	Initial distance to the interest area	Evaluate the impact of the initial distance to the <i>interest area</i> when SC is used or not.	The use of SC is particularly beneficial for short and medium distances to the target area. The bandwidth usage obviously increases with the distance to the interest area, but the use of SC minimizes it.
Comparison Varying the Density of Vehicles	4.2.3	Density of vehicles	Evaluate the impact of the vehicle density when SC is used or not.	The use of SC is very beneficial when the density of vehicles is low.
Experiments for the Determination of a Suitable Minimum Approaching Speed Threshold	4.3	Minimum approaching speed threshold	Evaluate the impact of the <i>minimum approaching speed</i> threshold, and choose the best value.	A value of the approaching speed threshold as low as 5 km/h, to determine whether SC should be used or not, provides good results. For higher threshold values, the performance is also better than when not using SC.
Experiments Analyzing the Impact of Different Factors on the Monitoring Process	4.4		Analyze the impact of different parameters on the performance of the whole SC monitoring process.	The use of SC can benefit the monitoring process, but a number of parameters must be set properly in order to maximize the performance.
Impact of the Percentage of Potential Collaborating Vehicles	4.4.1	Percentage of potential collaborators	Evaluate the impact of the percentage of collaborating vehicles.	A low percentage of collaborators (about 10%) is enough to benefit from SC.
Impact of the Density of Vehicles	4.4.2	Density of vehicles	Evaluate the impact of the density of vehicles (both collaborating and not collaborating) in the scenario.	Even with a low traffic density (60 vehicles/km ²), the use of SC provides good results.
Impact of the Minimum Stay Time in a Collaborator	4.4.3	Minimum Stay Time in Collaborators	Evaluate the impact of the minimum time that the mobile agent stays in a collaborator.	High staying times do not affect the total time or the reliability, but increase both the social and economic costs.

4.2. Experiments showing the interest of spatial crowdsourcing to travel to the interest area

In this first set of experiments, we evaluate the potential benefits that spatial crowdsourcing can provide. To show this, we compare the proposed spatial crowdsourcing approach with an approach that does not use spatial crowdsourcing, by measuring the performance of the process followed by the mobile agent to reach the interest area. Specifically, in Section 4.2.1 we compare how the performance of both approaches changes depending on the size of an area around the interest area where the density of vehicles may be low, in Section 4.2.2 we compare them by varying the distance to the interest area, and in Section 4.2.3 we compare them in scenarios with different traffic density.

4.2.1. Impact of the size of a low-traffic area

In this first experiment we perform a test in the previously described scenarios to see how useful the use of spatial crowdsourcing might be to the agent for reaching an area with a small amount of vehicles. For this purpose, we consider an area where the traffic density is low, called *cold area*, and we characterize it as a prolongation of the interest area that is extended a certain length around it (see Fig. 8). Specifically, from all the trajectories of vehicles randomly created by the simulator for this experiment, a 15% of them cross the cold area, and thus the traffic inside it is lower than in the rest of the scenario.

In this cold area, which has a shape similar to a belt around the interest area, the density of vehicles is much lower than in the rest of the scenario, and therefore the mobile agent will have a lower probability of finding a vehicle traveling towards the interest area to use it opportunistically to be carried there, or as an intermediate place to hop while looking for another better candidate. In this situation, the use of spatial crowdsourcing can benefit the monitoring process considerably, since the mobile agent can find a collaborator willing to alter its original route in order to carry



Fig. 8. Example of a cold area around the interest area.

the agent straight to the interest area in exchange of an amount of virtual money.

With this setup, we performed a series of simulations where we varied the size of the cold area (i.e., the length of the low-traffic interest area extension) from 0 meters to 1000 meters. In Fig. 9, we can see the total time needed by the mobile agent to complete the travel to the interest area in the three cities. We compare the results when spatial crowdsourcing is used (*Using SC*) with a situation where it is never used (*Without SC*). The larger the cold area, the longer it takes for the mobile agent to reach the interest area, as expected. However, when spatial crowdsourcing help is used, the times are considerably reduced. Next to the time values for the case when spatial crowdsourcing is used, we show the amount of virtual money spent in the process, with the notation "C: amount of virtual money units"; for example, in Fig. 9(c), for a size of the cold area of 1000 meters the agent spends 36 virtual money units, which allows reducing the total time by about 40% (from 227.5 seconds to 136.7 seconds). The differences observed in the total times for the three cities are due to the fact that the

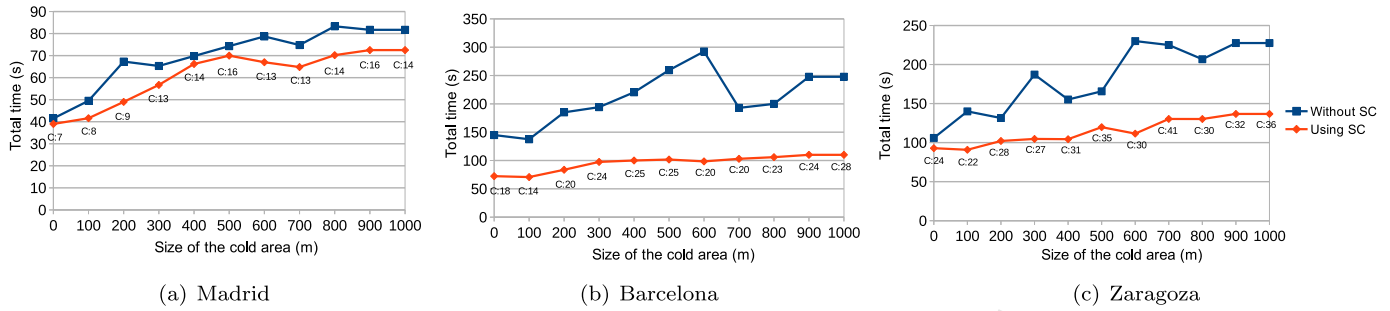


Fig. 9. Traveling to the interest area: comparison of the total time needed by using spatial crowdsourcing (SC) or not, varying the size of the cold area.

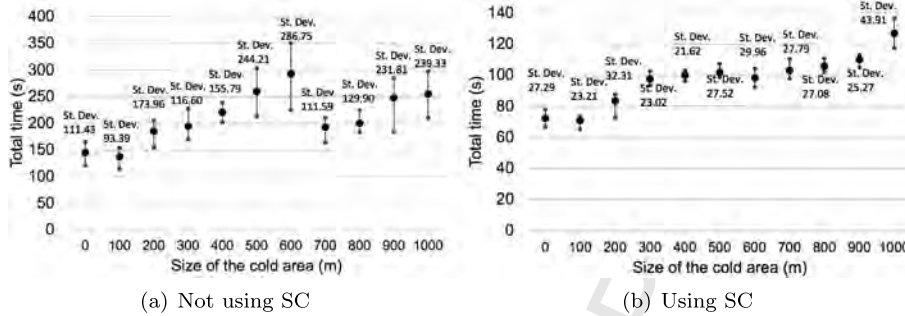


Fig. 10. Confidence intervals for Fig. 9(b) – city of Barcelona.

trajectories of the vehicles and the communications among them vary with the different topologies of the streets.

The difference in the total times shown in Fig. 9(b) for cold areas of size 600 and 700 meters when no spatial crowdsourcing is not used may attract the attention of the reader; this is because, in that case, the mobile agent finds more difficulties with a size of the cold area of 600 meters than with a size of the cold area of 700 meters, and therefore we observe a drop in the total time when the size of the cold area goes from 600 meters to 700 meters. Although this might seem counter-intuitive, as one may expect that a larger cold area would imply higher difficulties for the agent, we also have to take the topology of the streets into account. Indeed, it is not only the size of the cold area that matters but also the number and shape of the streets close to the boundaries of the cold area. So, by analyzing the results in detail, we have noticed that with a size of the cold area of 600 meters, in some simulations the agent tries to keep itself for a considerable amount of time in a narrow area close to the cold area until it finds a suitable vehicle to go towards the interest area, rather than exploring other alternatives, because it takes local decisions that lead to a local optimum (exploring other streets in those scenarios would initially lead the agent farther from the interest area). These problematic situations are avoided when the agent uses spatial crowdsourcing, as the agent would detect little progress towards the interest area (the approaching speed would be below the minimum approaching speed threshold) and ask the help of potential collaborating vehicles.

When spatial crowdsourcing is not used, we also observe a higher variability in the total times of different executions, as the approach with no spatial crowdsourcing is more sensitive to changes in the scenario. For illustration, in Fig. 10(a), we show the 95% confidence intervals of the average time values shown in Fig. 9(b) for the case without spatial crowdsourcing (the thicker circles represent the average values and the extremes the higher and lower values defining the interval), along with labels to indicate the standard deviations (*St. Dev.*) computed from the 50 executions for each value of the size of the cold area. Similarly, in Fig. 10(b), we show the confidence intervals and standard de-

viations for the case of using spatial crowdsourcing. Notice the different scale in the Y-axis of Fig. 10(a) and Fig. 10(b).

In Fig. 11, the total number of hops is shown. When spatial crowdsourcing is used, the number of times that the mobile agent hops (and so the bandwidth needed) is smaller: when the agent finds a collaborating vehicle, it is directly carried towards the interest area and does not need to use any other vehicle to hop to, unless the agent's approaching speed decreases below the minimum speed threshold and there is some promising neighboring vehicle; in this experiment, this is unlikely because the carrying vehicle is usually moving through the cold area, which is a low-traffic zone. On the contrary, when no collaborators are used, the mobile agent is constantly looking for a carrier better than the one it is currently traveling on. In this case, the current carrier may follow unpredictable routes, not only carrying the agent nearer the interest area; indeed, it might take a route that travels farther from the interest area. The size of the cold area seems to have little influence on the number of hops. There are two reasons for that. When using spatial crowdsourcing help, once a collaborator is found the mobile agent does not usually need to hop to other vehicles and stays in the same vehicle moving towards the interest area. When no spatial crowdsourcing is used, the number of hops varies with the size of the cold area in a more irregular way. This is due to the unpredictability of the routes and locations of the vehicles: the mobile agent must constantly evaluate its environment to decide the most suitable vehicle.

We also evaluated the social cost of using spatial crowdsourcing. In Fig. 12, we show the *social cost* (in minutes) in the different cities, when the mobile agent uses spatial crowdsourcing (when spatial crowdsourcing is not used, there is no social cost). Regarding the size of the cold area, it has little effect in the social cost, although a slight increase with its size is observed. This is an expected consequence, since the larger the cold area is, the longer it takes a collaborator to transport the agent to the interest area, and the longer he/she must postpone following his/her originally-intended route.

To summarize the conclusions regarding the influence of the size of the cold area, when it is larger the total time needed by

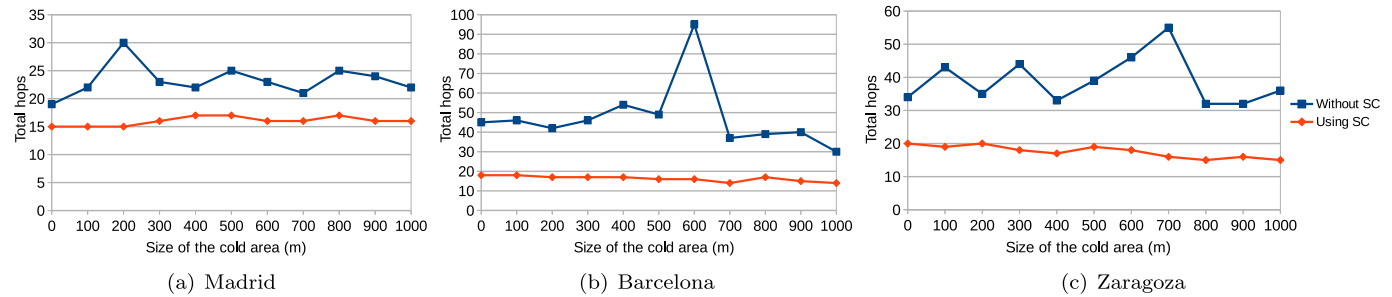


Fig. 11. Traveling to the interest area: comparison of the total number of hops needed by using spatial crowdsourcing (SC) or not, varying the size of the cold area.

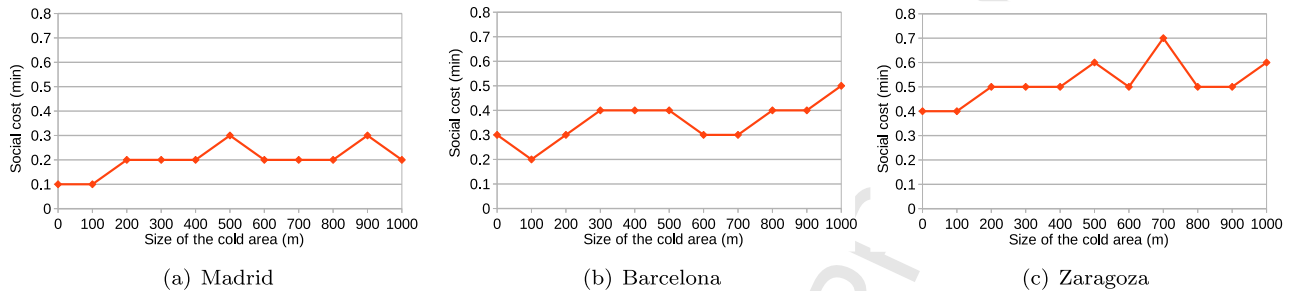


Fig. 12. Traveling to the interest area: collaborators' social cost.

the mobile agent to reach the interest area grows, although it is smaller when spatial crowdsourcing is used. The bandwidth usage remains similar, but when no spatial crowdsourcing is used it is higher and more irregular. Finally, the social cost, which only applies when using spatial crowdsourcing, is only slightly influenced by the size of the cold area.

4.2.2. Comparison varying the initial distance to the interest area

The previous experiment indicates that spatial crowdsourcing can be very useful in scenarios with areas where the density of vehicles is not high. Now, we will perform another experiment with higher traffic density in the whole scenario, in order to see if the proposed approach also behaves well when the number of potential agent carriers increases.

In this experiment, there is no cold area and there is a uniform density of vehicles in the scenario, which is set to 100 vehicles/km² (a medium density), as the default values in Table 3 indicate. We vary the initial distance from the point where the mobile agent starts its execution to the limit of the interest area, and we evaluate the potential benefits of spatial crowdsourcing for different values of such a parameter.

In Fig. 13, we can see how the total time to reach the interest area varies with the initial distance. In general, the times are slightly higher when no spatial crowdsourcing is used for the smaller to medium initial distances, whereas for higher distances the differences are near 0 and practically the same. A minor exception can be found in Fig. 13(c), for the city of Zaragoza, in a couple of cases where for some distance values (e.g., 1750 and 2250 meters) the times are slightly higher when spatial crowdsourcing is used, but this difference is quite small (about 3 seconds, from a total of around 67 seconds and 80 seconds, respectively); this difference is very small and can be due to a single sub-optimal choice made by the agent (e.g., a decision about being transported rather than actively jumping to a promising vehicle).

Regarding the number of hops that the mobile agent performs to reach the interest area, shown in Fig. 14, the figures for all the cities have a similar shape: the number of hops is slightly higher when no spatial crowdsourcing is used, and it grows linearly with the distance to the interest area.

As a summary, the time needed to reach the interest area grows with the distance. For short and medium distances, the spatial crowdsourcing option is better, but for longer distances the difference between both options is smaller. Regarding the total number of hops needed, the initial distance has a linear effect and in all the cities the crowdsourcing option takes fewer hops than the one without spatial crowdsourcing.

4.2.3. Comparison varying the density of vehicles

The density of vehicles is very important for the performance of the agent, since it can be difficult to reach the interest area if the number of vehicles that can be used as a physical transport or as an intermediate relay is low. In this experiment, we want to know in which circumstances the use of spatial crowdsourcing helps the mobile agent to reach the interest area. In this scenario, there is no cold area and in all the zone there is a uniform density of vehicles, which varies from 10 vehicles/km² (a very low density) to 175 vehicles/km² (a high density).

In Fig. 15, we can see how the total time to reach the interest area varies with the density of vehicles. In all the cities, the time is lower when spatial crowdsourcing is used than when it is not. The differences are higher with low vehicle density values, especially for the lowest value of 10 vehicles/km². For density values higher than 30 to 45 vehicles/km² the difference between using spatial crowdsourcing or not is practically negligible and in some cases (e.g., Fig. 15(c)) the total time might even increase very slightly when using spatial crowdsourcing. The reason is that, when the traffic density is high and in the absence of a cold area, the mobile agent can find suitable vehicles to hop to in order to transport itself via wireless communications. When spatial crowdsourcing is not used, the mobile agent must go to the interest area by hopping from one vehicle to another that seems more promising to reach the target sooner, and finding candidates is more difficult when the density of vehicles (and therefore, its total amount) is lower.

Fig. 16 shows that, as the vehicular density grows, the number of hops that the mobile agent needs to perform to reach the interest area decreases when spatial crowdsourcing is not used. On the other hand, when it is used, the number of hops increases with the density of traffic until around 45 vehicles/km², when the num-

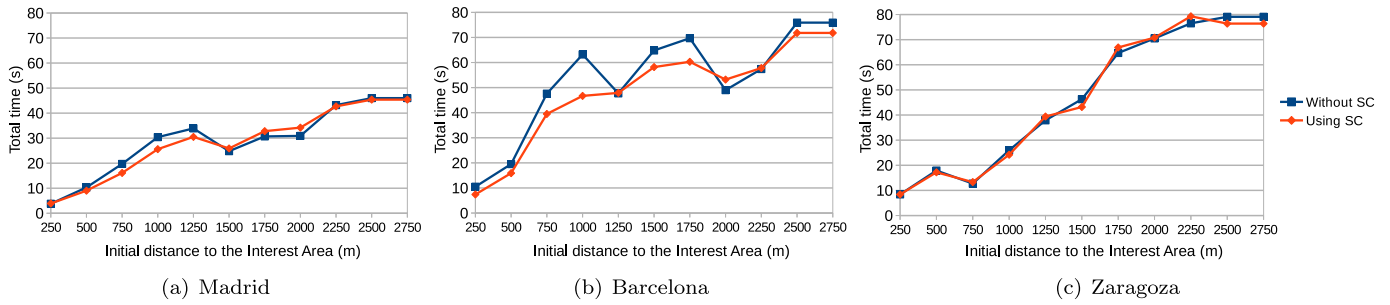


Fig. 13. Traveling to the interest area: comparison of the total time needed by using spatial crowdsourcing (SC) or not, varying the initial distance (no cold area).

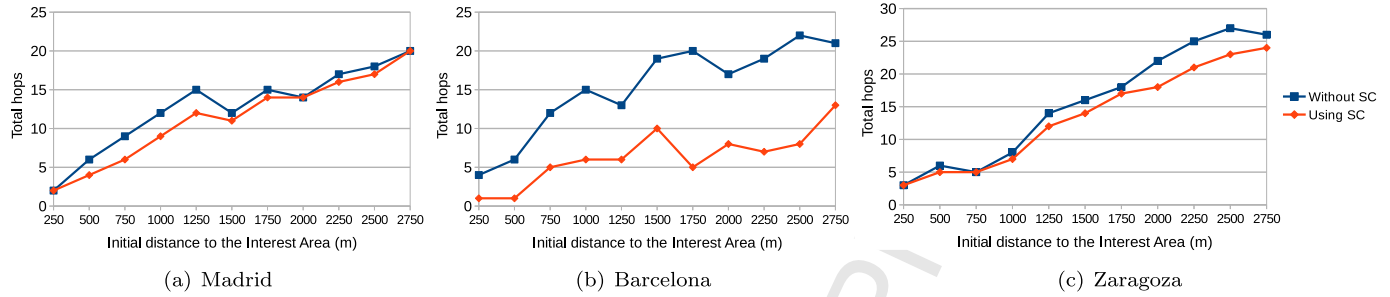


Fig. 14. Traveling to the interest area: comparison of the total number of hops needed by using spatial crowdsourcing (SC) or not, varying the initial distance (no cold area).

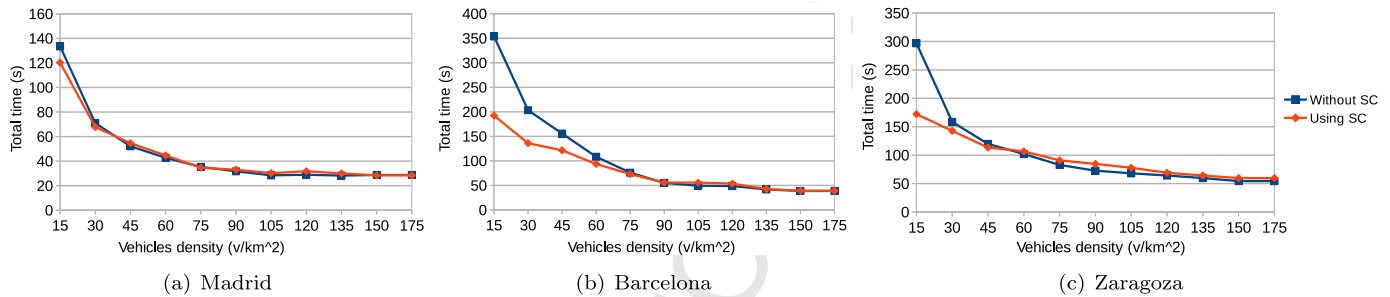


Fig. 15. Traveling to the interest area: comparison of the total time needed by using spatial crowdsourcing (SC) or not, varying the vehicle density (no cold area).

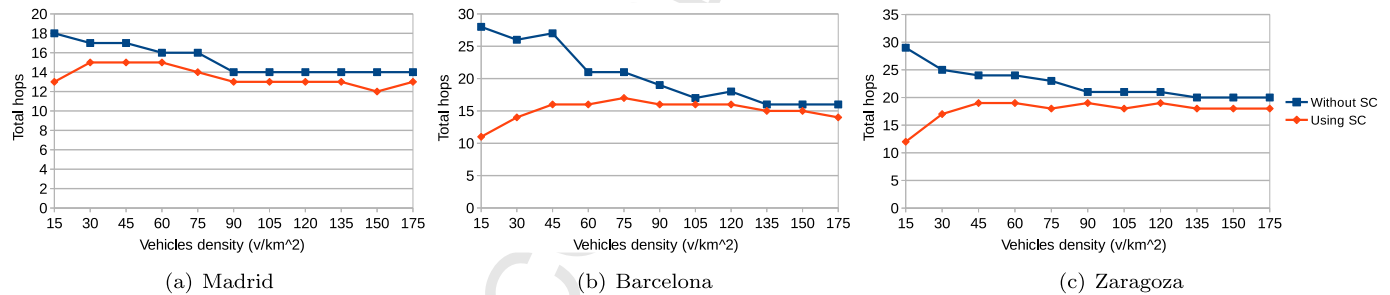


Fig. 16. Traveling to the interest area: comparison of the total number of hops needed by using spatial crowdsourcing (SC) or not, varying the vehicle density (no cold area).

ber of hops remains stable with small variations for higher density values. In all the cases, the number of hops when spatial crowdsourcing is used is smaller than when it is not used, although the difference decreases as the vehicular density grows. This is due to the fact that, once the mobile agent takes a collaborator, it stays in there and does not leave it until the interest area is reached, being unnecessary to perform more hops to other vehicles.

Summing up, we conclude that the spatial crowdsourcing approach is particularly beneficial in terms of time and bandwidth usage in scenarios with a low density of vehicles. Moreover, when the traffic density is higher, the use of spatial crowdsourcing does not harm either the performance. Indeed, based on the minimum

speed threshold, the mobile agent can choose to hop among the vehicles when it considers that paying a collaborator is not needed.

4.3. Experiments for the determination of a suitable minimum approaching speed threshold

In the second set of experiments, we focus on the determination of a suitable value for the minimum approaching speed threshold parameter used by a mobile agent to decide when collaborators should be used. As explained in Section 3.3, the travel to the target area, as well as the return to the origin with the results, is performed by the mobile agent thanks to a combination

of two techniques: initially, the mobile agent starts hopping from one vehicle to another one that it considers to be a more promising carrier to reach the target area and, additionally, the agent may also use spatial crowdsourcing to be physically carried nearer the target area by a collaborating vehicle. To decide whether using one technique or the other at any time, a minimum speed threshold is set: if the mobile agent approaches the destination area with a speed smaller than this threshold value, it will ask for help to nearby collaborating vehicles to be carried nearer the destination area faster, until the approaching speed raises again above the threshold. Otherwise, the mobile agent will keep hopping from one vehicle to another, as described before. It is important to establish the minimum speed threshold carefully. If it is too high, the mobile agent will likely look for collaborators too early, and therefore it will spend too much virtual money needlessly. On the other hand, if the value is too low, then the agent will ask for help later, but then it may be already too far from the interest area and thus take too much time to reach the destination.

In Section 3.3.4, we describe two approaches regarding the use of spatial crowdsourcing during the data collection phase (SCCA and PHCA), which we now compare experimentally for varying values of the minimum approaching speed threshold (in the range from 0 km/h to 50 km/h, which is the usual maximum speed within a city). In the rest of this section, the results obtained by using SCCA are shown in Section 4.3.1, the results obtained by using PHCA are shown in Section 4.3.2, and a comparison between the two is presented in Section 4.3.3. As the default parameter values in Table 3 show, the ratio of collaborating vehicles is set to 50%, taken as an intermediate value, and there is no cold area. Besides, as we want to see the benefits of using spatial crowdsourcing also within the interest area, the probability that the trajectory of a given vehicle will travel inside the interest area is 10%, as this implies a low-traffic situation where the difference between using spatial crowdsourcing or not using it during the data collection phase can be more relevant.

4.3.1. Monitoring process: results with SCCA

Fig. 17(a) shows the time required for completing the monitoring process in the three cities, by using the SCCA variant of the mobile agent, as described before. The extreme case when the minimum speed threshold equals 0 km/h means that the condition for using spatial crowdsourcing is *never* met, and therefore the agent must reach the target area only by hopping from one vehicle to another (spatial crowdsourcing is not used at all). However, once the target area is reached, the SCCA mobile agent *always* uses spatial crowdsourcing to travel to all the cells, with independence of the minimum approaching speed threshold (the agent is already within the target area, and so the concept of approaching speed does not apply). As a result, in general, in all the maps, the total time required is higher with a value of 0 km/h than with other higher values, and the variations for higher minimum speed thresholds are quite small.

In Fig. 17(b), we can observe that the number of hops needed by the mobile agent to complete the process is much higher for the minimum speed threshold value of 0 km/h (when no spatial crowdsourcing is used), and for values higher than 5 km/h the number of hops decreases. The reason is that when no spatial crowdsourcing is used the mobile agent needs to constantly hop from one vehicle to another until it reaches the interest area.

In Fig. 17(c), the percentage of collected data is shown. The result for all the minimum speed threshold values tested are 100% (i.e., all the requested data are collected), but it must be taken into account that for the value of 0 km/h (i.e., when no spatial crowdsourcing is used) the reliability is below 90%, so the collected data percentage shown in Fig. 17(c) refers only to the simulations ended within the simulation time limit.

In Fig. 17(d), the amount of virtual money spent by the mobile agent is shown. As explained in Section 3.3, in our experiments, we assume that the agent pays one unit of virtual money for every second it is carried by a collaborating vehicle. When the minimum speed threshold value is 0 km/h, the amount paid is very low, since the collaborators are only used when the mobile agent collects data when traveling to the target area cells within the interest area, but not for reaching the area when returning to it when the agent leaves it unintentionally, or for returning to the monitoring origin. For higher values of the minimum speed threshold, the virtual money spent increases, being the best overall value around 5 km/h.

Regarding the social cost, shown in Fig. 17(e), the SCCA agent exhibit a similar behavior in the three cities: when the minimum speed threshold value is 0 km/h, the cost is very low because spatial crowdsourcing is used only when the mobile agent is inside the target area collecting data. For higher speed threshold values, the social cost also increases, since the agent makes use of spatial crowdsourcing more frequently. Note that this graphic shows a trend very similar to the one of the virtual money spent (though not exactly the same), which means that the established mechanism to compensate the collaborators (paying them according to their invested time) is quite fair, since the compensation paid is related to the social costs incurred by the collaborators.

Finally, Fig. 17(f) shows the number of collaborating vehicles that the SCCA agent needs to complete the process. For the minimum speed threshold value of 0 km/h, the number of collaborators is very low, because they are only used in the data collection phase, but not for traveling to/from the target area. For the next minimum speed threshold value (5 km/h), the number of collaborators grows to between 11 and 15, but then it decreases as the minimum speed threshold value increases, until 6 to 10 collaborators for 50 km/h. This is interesting, as compared to the results of the total time, which has a low variability for minimum approaching speed threshold values in the range of 5 to 50 km/h (see Fig. 17(a)). When the minimum approaching speed is set to a low value (e.g., 5 km/h), the mobile agent uses more collaborators but for a shorter time. On the other hand, when the minimum approaching speed is set to high values (e.g., 45 km/h), fewer collaborators are used by the mobile agent, but it travels physically aboard them for a longer time. The reason is the following: when the agent uses spatial crowdsourcing it will stay in the collaborating vehicle until the approaching speed exceeds the minimum speed threshold, and shortly after that moment the agent will leave the collaborator if it finds a more promising vehicle, and will continue approaching the target area, hopping from one vehicle to another by using the established hop strategy. For low values of the minimum speed threshold, it will be easier to reach the threshold value, and the agent will leave the collaborator sooner than for high values. As the minimum approaching speed threshold grows, the mobile agent will travel longer in a collaborator, and will approach nearer the target area, so it will also need a smaller number of collaborators to reach it.

It should be noted that the performance results of unended simulations are not available, and so another parameter tested in the experiment is the reliability of the monitoring. In Table 6, we can observe that the reliability of the process is between 56% and 90% when then minimum speed threshold equals 0 km/h (i.e., when no spatial crowdsourcing is used). For higher values of the minimum speed threshold, the reliability reaches 100% (with one single exception for the minimum speed threshold value of 40 km/h and the city of Zaragoza, with a reliability of 98%), meaning that the simulations end within the time limit. That is, the use of spatial crowdsourcing for values above or equal to 5 km/h increases the reliability of the whole monitoring process considerably.

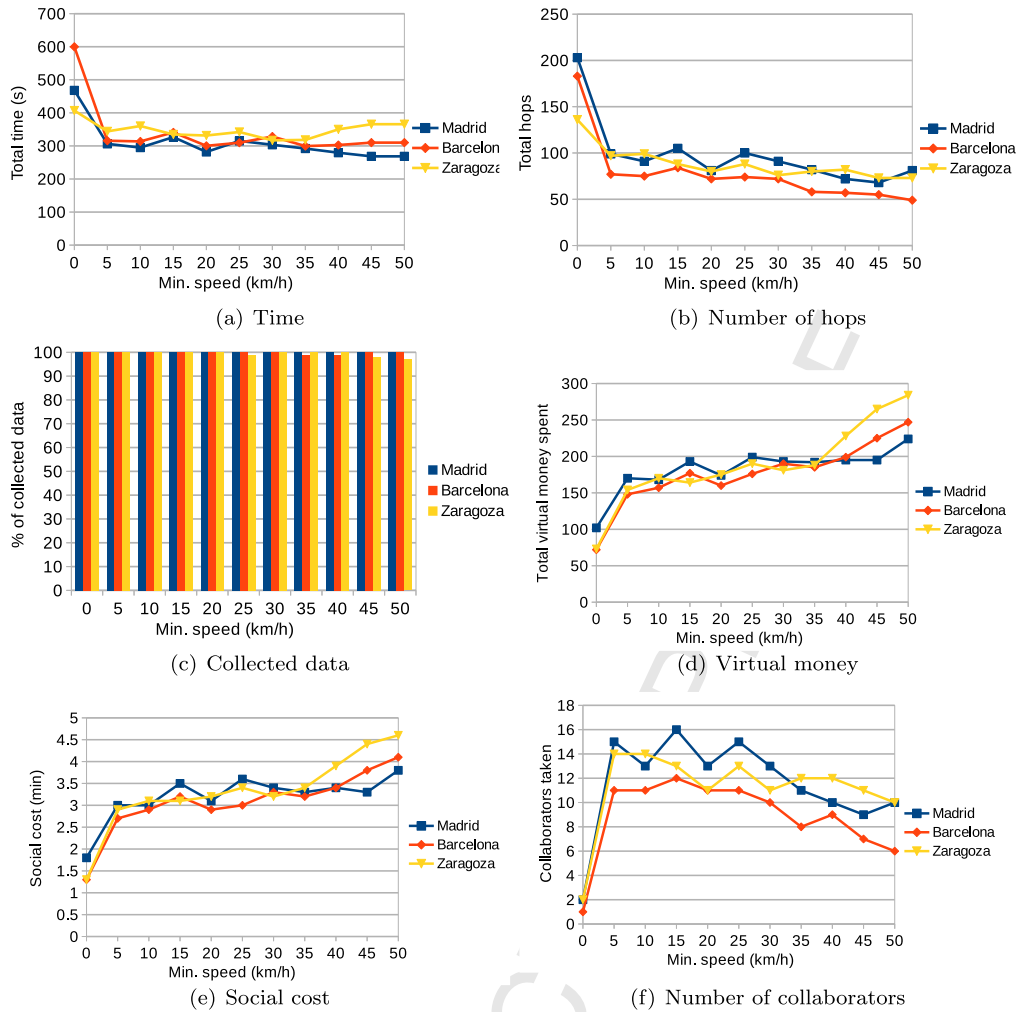


Fig. 17. Monitoring process: results with SCCA.

Table 6
Monitoring process: reliability with SCCA.

minspeed (km/h)	0	5	10	15	20	25	30	35	40	45	50
Madrid (% of ended simulations)	56	100	100	100	100	100	100	100	100	100	100
Barcelona (% of ended simulations)	90	100	100	100	100	100	100	100	100	100	100
Zaragoza (% of ended simulations)	68	100	100	100	100	100	100	100	98	100	100

4.3.2. Monitoring process: results with PHCA

If we test the same minimum speed threshold values with the variant PHCA of the mobile agent, we obtain another set of results, that are shown in Fig. 18. This version of the agent never uses spatial crowdsourcing when it is inside the target area, nor when the agent leaves it accidentally and must return to resume the monitoring task. The mobile agent only hops from car to car by using the MAP hop strategy to estimate which car in range will reach the target area earlier.

In Fig. 18(a), the total time is shown and the best results are, in general, for minimum speed threshold values around 5 km/h, whereas for higher values the total time increases very slightly.

The total number of hops is shown in Fig. 18(b), which exhibits a behavior similar to SCCA. For the extreme value of 0 km/h, the agent must only hop among the vehicles without using spatial crowdsourcing, and therefore the number of hops is much higher. For higher values of the minimum speed threshold, the number of hops decreases very slightly due to the use of spatial crowdsourcing to reach the target area and return to the origin.

The results of the amount of collected data, shown in Fig. 18(c), are noteworthy: all the results are only around 60%, which is much lower than with SCCA. The reason is that, once the agent reaches the target area, it must visit the cells through only car-to-car hops, without using spatial crowdsourcing, which is harder to accomplish, as the number of vehicles inside the area is low and their trajectories unknown. Due to this, the agent needs more time to perform the task and in many cases it reaches the data collection timeout of 3 minutes even before it is completed.

In Fig. 18(d), the virtual money spent by the agent is shown. For the minimum speed threshold value of 0 km/h, the amount of money spent is 0, since the agent never uses spatial crowdsourcing inside the target area, and neither to reach or leave it. For higher values, the best results are obtained for 5 km/h and the worst results for values higher than 30 km/h.

Regarding the social cost shown in Fig. 18(e), we can observe the same behavior as with the SCCA version and again a very similar trend to the one shown in Fig. 18(d) for the virtual money

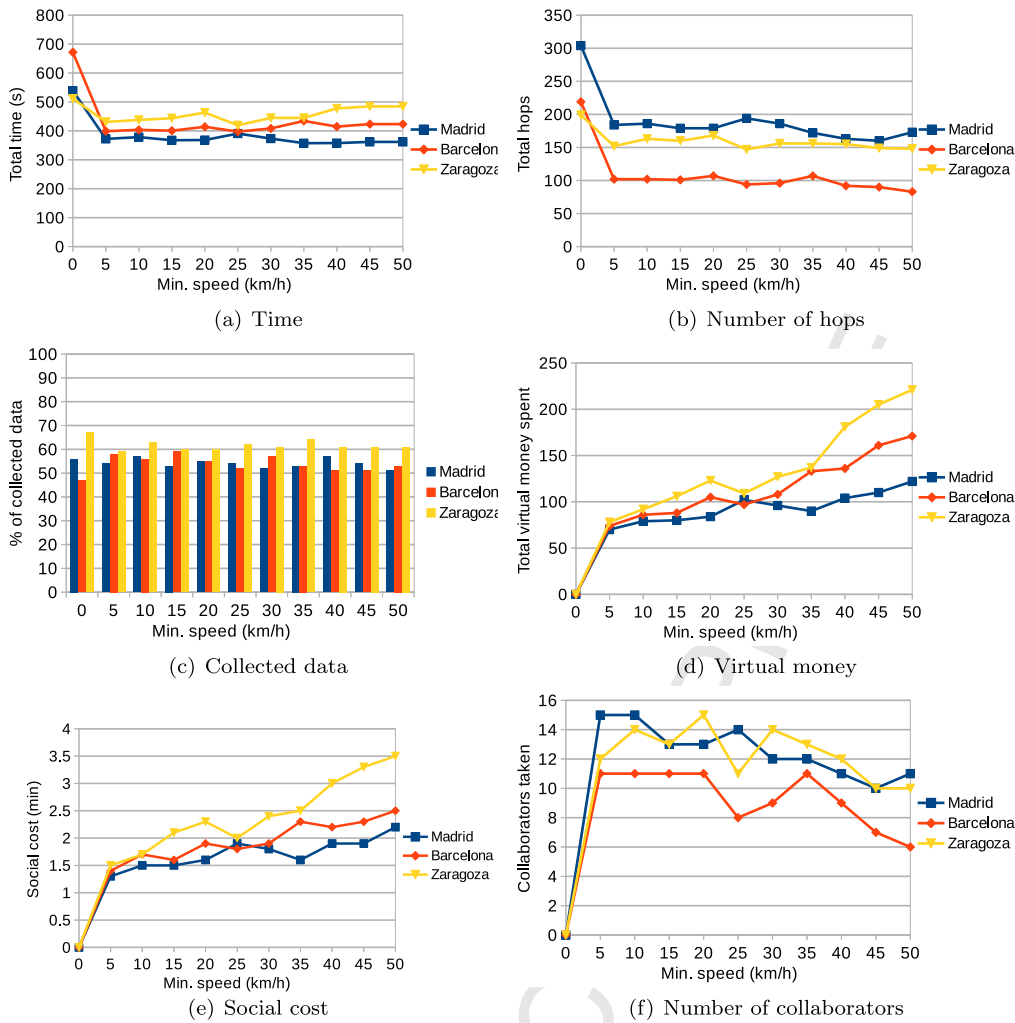


Fig. 18. Monitoring process: results with PHCA.

Table 7
Monitoring process: reliability with PHCA.

minspeed (km/h)	0	5	10	15	20	25	30	35	40	45	50
Madrid (% of ended simulations)	68	98	100	100	100	100	100	100	100	100	100
Barcelona (% of ended simulations)	82	100	100	100	100	100	100	100	100	100	98
Zaragoza (% of ended simulations)	80	98	100	98	100	100	100	100	100	100	100

spent by the agent, and thus the collaborators are compensated according to their actual cost in quite a fair way.

Finally, in Fig. 18(f), the number of collaborators taken by the agent is shown. This amount is exactly 0 for the minimum speed threshold value of 0 km/h, since spatial crowdsourcing is never used for traveling to/from the target area, nor for collecting data inside it when it is reached. For minimum speed threshold values of 5 km/h and above, the collaborators are used in a similar way to the SCCA variant of the agent: as the minimum speed threshold grows, the number of collaborators decreases. Since the total time remains more or less constant, this means that for low values of the minimum speed threshold, a collaborator is usually taken for a smaller time than for higher threshold values.

Regarding the reliability of the whole process (shown in Table 7) the best results are for minimum speed threshold values between 10 km/h and 45 km/h, with reliabilities of 100%. On the other hand, the worst results are for the value of 0 km/h, when no spatial crowdsourcing is used, but there are also some cases when the reliability does not reach 100% and stays in 98%.

4.3.3. Monitoring process: comparison between SCCA and PHCA

Comparing the results of these experiments (regarding the determination of the minimum approaching speed threshold), we can conclude that:

- The variant SCCA of the agent takes less time than PHCA to complete the monitoring process, due to the advantage of using spatial crowdsourcing inside the target area (on average, around 300 s in the first case and slightly more than 400 s in the second).
- Regarding the reliability, both approaches yield similar results, being not smaller than 98% for minimum approaching speeds of at least 5 km/h.
- Considering the total number of hops, SCCA is better (less than 100 hops, versus a value between 100 and 200 hops with PHCA) since the use of spatial crowdsourcing reduces the necessity to keep looking for potentially better vehicles to hop to.

- Regarding the data collection rate, again SCCA outperforms PHCA, being the first one around 100% and the second one around 60%.
- Concerning the virtual money cost, however, PHCA is better than the SCCA (with an average cost of 93 versus 126 units of virtual money), due to the fact that the former never uses spatial crowdsourcing once the mobile agent reaches the target area, which saves money.
- For both approaches, the use of spatial crowdsourcing with a minimum approaching speed threshold is always better than not using it, in terms of time, reliability, bandwidth usage and amount of collected data, as the results for the minimum speed threshold value of 0 km/h show.
- For both variants, a minimum approaching speed threshold around 5 km/h is appropriate, if we take into account all the results from these experiments.

Therefore, we can establish that the best algorithm in all terms (except in the virtual money cost) is SCCA, and we choose for the minimum approaching speed threshold 5 km/h, which is quite a low value. This value, as well as the SCCA strategy, are used by default in the experiments, as shown in Table 3.

4.4. Experiments analyzing the impact of different factors on the monitoring process

In the third set of experiments, we analyze the impact of different parameters on the performance of the whole monitoring process, such as the percentage of collaborators (Section 4.4.1), the existing traffic density (Section 4.4.2), and the required minimum stay time of mobile agents in the vehicles (Section 4.4.3). In these experiments, the mobile agent follows the complete monitoring process described in Section 3: it first travels to the interest area and, once the interest area is reached, it visits every cell into which it is divided to read the required data by using the available sensors in its carrying vehicle. When the mobile agent has read all the sensors required (or the maximum timeout expires), it returns to its origin carrying with it the data collected. As in Section 4.3, the probability that the trajectory of a given vehicle will travel inside the interest area is 10%, as in this case the use of spatial crowdsourcing for the data collection stage (SCCA approach) can be more relevant.

4.4.1. Impact of the percentage of potential collaborating vehicles

Fig. 19(a) shows how the total time needed to complete the whole monitoring process varies with the ratio of vehicles willing to act as collaborators in spatial crowdsourcing tasks. The required time is around 500 to 700 seconds for 0% of potential collaborators, that is, when no spatial crowdsourcing can be used and the mobile agent can only rely on hopping from one vehicle to another to reach the destination and perform the monitoring task inside the target area. The time decreases to around 400 seconds when the ratio of potential collaborators increases to 10%, and then decreases slightly until it reaches around between 300 and 350 seconds for 100% collaborators (i.e., all the vehicles in the scenario can act as collaborators). The reason is that once the mobile agent finds a suitable collaborator it is not necessary to keep looking for another one, and thus having more potential collaborators to choose from is useless.

Figs. 19(b), 19(c) and 19(e) show the number of hops performed by the mobile agent, the amount of data collected, and the social cost, respectively. In all these figures, we can see the same pattern: when spatial crowdsourcing is not used (because the ratio of potential collaborators is 0%), then the corresponding metric measured is significantly worse than when it is used.

The results improve dramatically with only about 10% collaborators. However, when the ratio of potential collaborators is higher than this value, the measured parameters do not show further improvement (the mobile agent does not significantly benefit from a higher number of potential collaborators). The exception to this pattern is the virtual money spent, shown in Fig. 19(d), which is obviously 0 when no collaborators are used for performing spatial crowdsourcing. Similarly to what was observed in previous experiments, as expected, Figs. 19(e) and 19(d) show a similar pattern.

Regarding the number of collaborators used by the agent, shown in Fig. 19(f), it is obviously 0 when the ratio of potential collaborators is 0%. For higher ratios, the number of effective collaborators used by the agent varies more or less randomly, but for all the cases its number is within the range of 10 to 21 collaborators. When the mobile agent commutes from spatial crowdsourcing to hopping to other vehicles and back again to spatial crowdsourcing, it seems to find a new collaborator easily, as the total time for the monitoring (shown in Fig. 19(a)) does not increase significantly.

In Table 8, we can see how the *reliability* is between 68% and 80% (depending on the city map) when the ratio of potential collaborators is 0%. For 10% of potential collaborators, it increases to between 96% and 100%, and for higher values the reliability is always 100%, with the exception of a few cases where it is 98%. Therefore, not being able to use spatial crowdsourcing (i.e., 0% of potential collaborators) makes the process more unreliable than when using it.

As a conclusion for this experiment, we can say that the benefits of using spatial crowdsourcing are remarkable even with a ratio of potential collaborators as low as 10%. Therefore, we do not need a high number of drivers willing to modify their trajectories to benefit from the spatial crowdsourcing approach and increase the performance of the process.

4.4.2. Impact of the density of vehicles

In this experiment, we test the influence of the density of vehicles on the performance of the monitoring process. In the previous experiments, where the density was set to 100 vehicles/km² (a medium value), the results showed that even with a small number of potential collaborators the performance improves dramatically. However, we also want to test it in a more challenging scenario with low and very low vehicle density, to find out the limits of spatial crowdsourcing. The chosen density values and their denomination (low density, medium density, etc.) are inspired by those used in proposals such as [37,38], and for this experiment we vary them from 10 vehicles/km² (very low density) to 100 vehicles/km² (medium density). Based on the results of the experiment presented in Section 4.4.1, in this experiment the percentage of potential collaborating vehicles is set to 10%.

In Fig. 20(a), the total time to complete the monitoring process is shown. As the number of vehicles increases, the time needed decreases, with little variance when the density reaches around between 70 and 80 vehicles/km². The reason is that, with a high number of vehicles, the mobile agent has less difficulties to find a car to be carried nearer the target area, whether it uses spatial crowdsourcing or just hops from one vehicle to another.

Regarding the number of hops performed by the mobile agent, shown in Fig. 20(b), its number increases with the vehicle density, since the agent is constantly looking for a better option to reach its destination sooner: if there exist more vehicles, then the number of opportunities is also higher and the agent will likely hop to another car more frequently.

The amount of data collected within the timeout of 3 minutes, shown in Fig. 20(c), increases with the vehicle density, although there is little variation and all the values are higher than 90%.

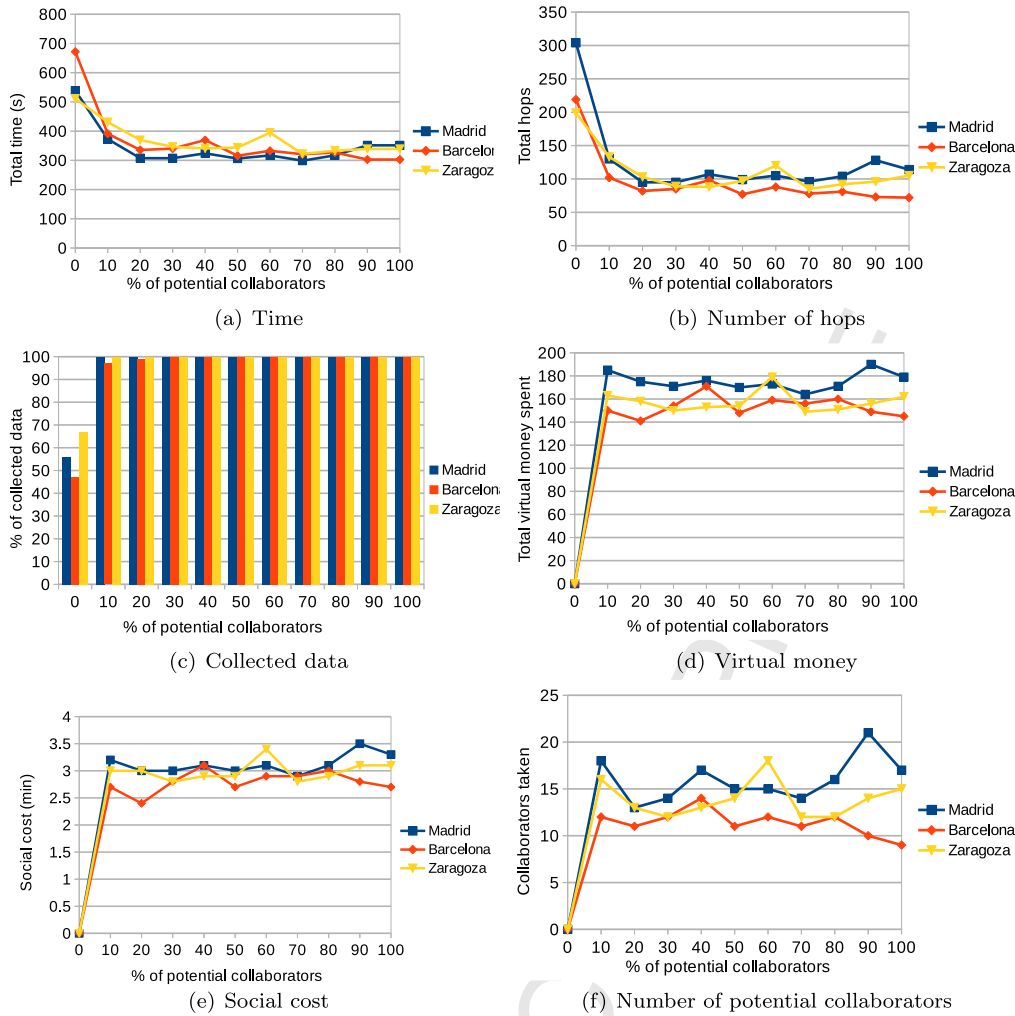


Fig. 19. Monitoring process: results varying the percentage of potential collaborating vehicles.

Table 8
Monitoring process: reliability varying the percentage of collaborating vehicles.

% of collaborators	0	10	20	30	40	50	60	70	80	90	100
Madrid (% of ended simulations)	68	100	100	100	98	100	100	98	100	98	100
Barcelona (% of ended simulations)	82	100	100	100	100	100	100	100	100	100	100
Zaragoza (% of ended simulations)	80	96	100	98	100	100	100	98	100	100	100

Figs. 20(d) and 20(e) show the virtual money paid by the mobile agent to the collaborators and the social cost, respectively. As in the previous experiments, these figures are closely related, showing the relation between the social cost and the virtual money spent by the mobile agent, and both values increase slightly with the density of vehicles. With a higher number of vehicles there is also a higher number of potential collaborators, and the agent can obtain help more easily, and therefore it could end up spending more virtual money. On the other hand, with less vehicles there is a smaller number of potential collaborators, so it is more likely that the agent cannot find one of them when needed and, consequently, it might spend less virtual money.

Finally, the number of collaborators taken by the mobile agent, shown in Fig. 20(f), increases significantly with the density of vehicles, for the same reasons: since the number of potential collaborators is higher, the agent has more opportunities to take advantage of them and they are used more frequently.

Table 9 shows the reliability of the monitoring process for the different traffic density values. As the vehicle density increases, so

does the reliability, and starting with 50 vehicles/km² there is little variance. This result is related to the total time required for the monitoring, as the sooner the mobile agent completes all the phases of the process, the less likely the time limit will be reached.

As a conclusion for this experiment, we can say that, regarding the influence of the vehicle density in spatial crowdsourcing, the best results in terms of time, reliability and amount of collected data are obtained for values around 60 vehicles/km² (a low density) and higher. For lower densities, the results are worse, but not so bad for the spatial crowdsourcing to be considered useless under those circumstances: for example, for the extremely low density value of 10 vehicles/km² the amount of collected data is higher than 90% and the reliability is near 75% in the worst case, which may be enough for monitoring tasks with soft requirements.

4.4.3. Impact of the minimum stay time in a collaborator

In this experiment, we evaluate the impact of the minimum stay time required in each collaborating vehicle used. Specifically, we vary this value from 0 to 50 seconds in increments of

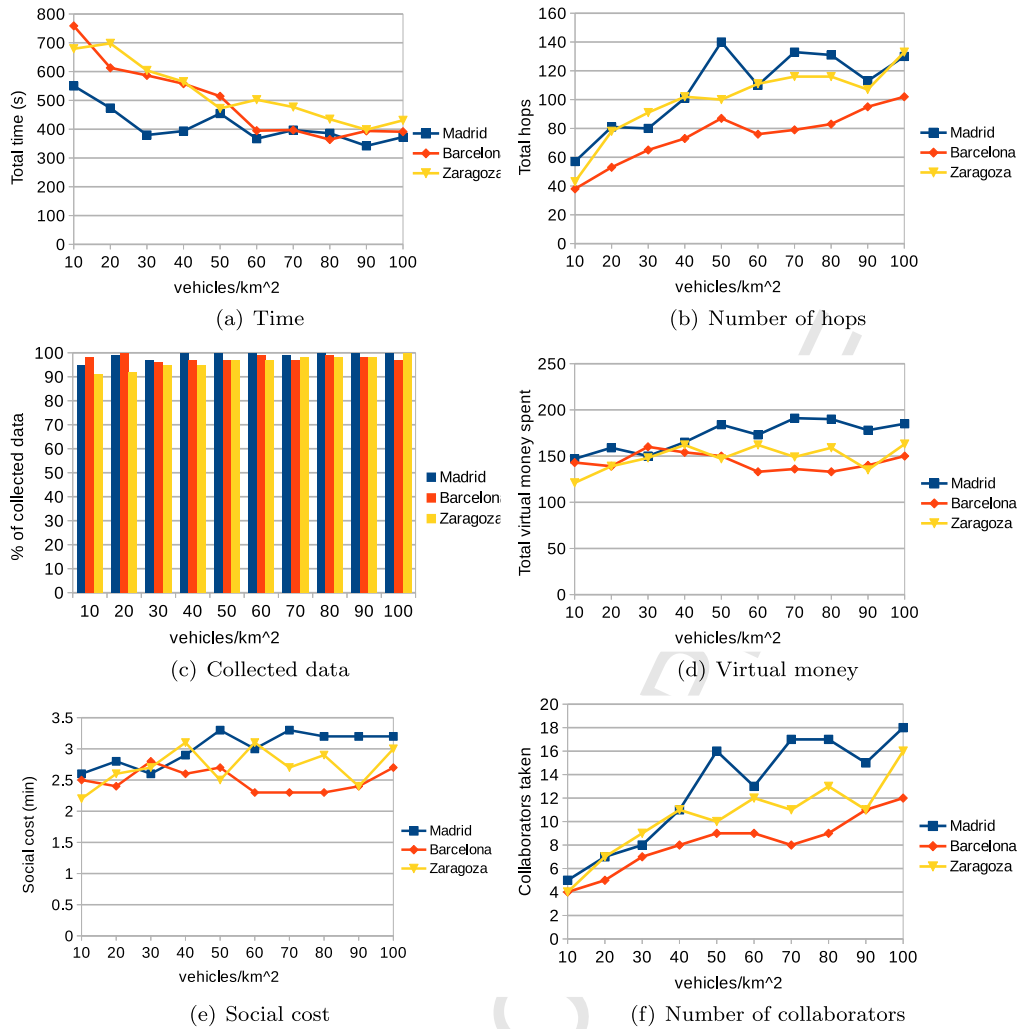


Fig. 20. Monitoring process: results varying the density of vehicles.

Table 9

Monitoring process: reliability varying the density of vehicles.

Density (vehicles/km ²)	10	20	30	40	50	60	70	80	90	100
Madrid (% of ended simulations)	86	98	98	100	98	100	98	100	98	100
Barcelona (% of ended simulations)	74	86	100	92	96	98	100	100	100	100
Zaragoza (% of ended simulations)	80	82	96	88	98	100	100	100	100	96

5 seconds, to evaluate its impact. Besides, based on the results of the experiment presented in Section 4.4.1, in this experiment the percentage of potential collaborating vehicles is set to 10%. In Fig. 21(a), we can see how the total time to complete the process decreases slightly as the minimum stay time increases. The reason is that the likelihood of wasting time decreases when the agent stays in a collaborator longer, as the collaborator commits itself to bring the agent to its destination. Besides, the number of hops also decreases, as shown in Fig. 21(b).

Regarding the amount of collected data, shown in Fig. 21(c), it remains more or less constant as the minimum stay time increases, and it is always above 96%. The mobile agent takes into account the minimum stay time only in the phases of the process where it approaches the target area and returns to its origin, but not in the data gathering phase, so it has no influence during the data collection step.

The invested virtual money and the social cost, shown in Figs. 21(d) and 21(e), respectively, exhibit a similar behavior, like

in the previous experiments, in this case both growing with the minimum stay time. The difference between the values observed for the lowest minimum stay value (0 seconds) and the highest minimum stay value (50 seconds) is about 25%, and the reason is that the mobile agent wastes less time in hops to other vehicles and stays longer in the collaborating vehicles, so these collaborators receive more compensations in the form of virtual money. This fact is confirmed by the number of collaborating vehicles taken by the agent, shown in Fig. 21(f). As the minimum stay time grows, the number of collaborators decreases strongly. Along with the low variability of the total time, this means that the mobile agent takes a smaller number of collaborators but stays longer in them.

As shown in Table 10, the reliability parameter is 100% for almost all the minimum stay times (with only one exception, with 96%, for the city of Zaragoza and a minimum stay time of 0 seconds). The reliability is related to the total time of the process (and whether it takes longer than the timeout), and given the

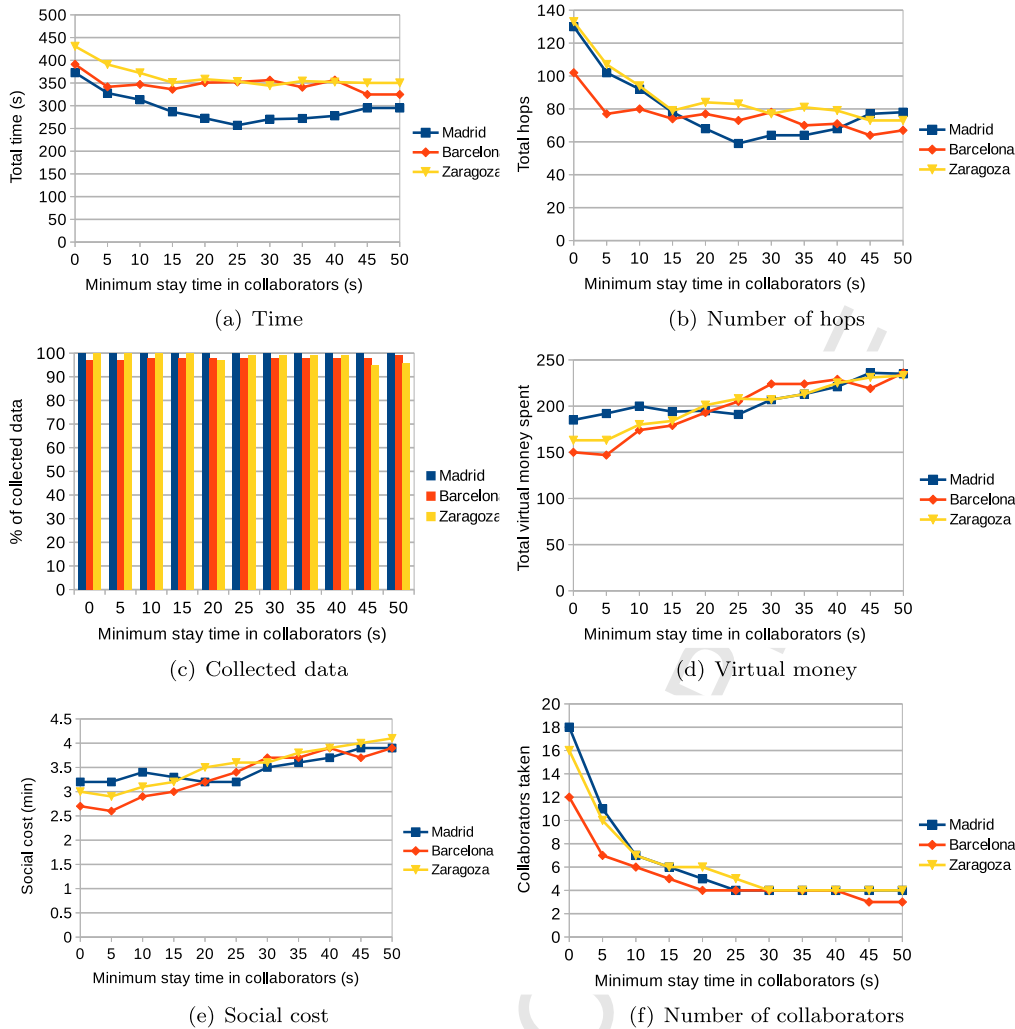


Fig. 21. Monitoring process: results varying the minimum stay time of the mobile agent in collaborators.

Table 10 Reliability varying the minimum stay time of the mobile agent in collaborators.

Minimum stay time (s)	0	5	10	15	20	25	30	35	40	45	50
Madrid (% of ended simulations)	100	100	100	100	100	100	100	100	100	100	100
Barcelona (% of ended simulations)	100	100	100	100	100	100	100	100	100	100	100
Zaragoza (% of ended simulations)	96	100	100	100	100	100	100	100	100	100	100

small variance of the total time, the minimum stay time has no appreciable effect on the reliability.

As a conclusion, the use of a minimum stay time in a collaborator has a certain influence on the number of hops that the agent needs to take, decreasing them as the minimum stay time grows. However, this reduction has no influence on the reliability or the amount of collected data, and increases the social cost of the collaborators and the amount of virtual money spent by the agent.

5. Related work

Spatial crowdsourcing is a hot topic nowadays. There exist some proposals related to monitoring and sensing in urban areas by using vehicles and networks but, up to the authors' knowledge, our work is the first proposal that applies the idea of spatial crowdsourcing to vehicular networks, with the exception of our previous seminal work that was published as a short paper [9]. Moreover, as opposed to other works focusing on spatial crowdsourcing in general, our proposal is completely distributed and opportunistic,

as it does not rely on a central entity with global knowledge about the potential participants and benefits from the direct interactions among nearby vehicles. By opportunistically considering neighbor vehicles as potential collaborators, the disadvantages of centralized approaches are avoided. In this section, we compare some related work and highlight the major differences and enhancements of our proposal. We first consider approaches related to spatial crowdsourcing in general and then approaches in the specific context of vehicles.

5.1. Related work on spatial crowdsourcing and crowdsensing

The topic of *mobile crowdsensing* (also called *collaborative sensing*, *mobile sensing*, or *participatory sensing*) has attracted significant research attention (e.g., see [39–45]).

One common problem of spatial crowdsourcing is known as the *Maximum Task Assignment (MTA)* problem, which focuses on how to allocate the available *workers* (in our context, collaborating vehicles) to spatial tasks, to maximize the global performance. There

1 exist many proposals to tackle this problem (e.g., see [46–51])
2 under different circumstances and with different constraints. How-
3 ever, most of them need a central server which centralizes all the
4 tasks that need to be assigned and which constantly receives the
5 updated location of the potential collaborators, in order to allo-
6 cate them the most appropriate task according to several factors
7 (e.g., their position, travel time to the destination, etc.). These so-
8 lutions have the advantage of optimizing the performance of the
9 crowdsourcing process, but they also have some drawbacks, such
10 as privacy concerns (the central server needs to know the position
11 of all the collaborators) and the need of direct communications
12 with the central server, which can have an economic cost (in case
13 of using 3G/4G technologies) or may need a previously-existing in-
14 frastructure (e.g., cellular phone towers or Wi-Fi access points). In
15 our approach, we consider a pure ad hoc solution where a mo-
16 bile agent directly asks for help to the neighbor vehicles. Thus, we
17 avoid the disadvantages of the centralized approaches. However,
18 the assignments performed in our approach are not necessarily
19 globally optimal, as they are based on local decisions taken with-
20 out having a global vision of the scenario.

21 In [52], the authors propose a method for recruiting collabo-
22 rators for crowdsensing tasks, by using semi-Markov models to
23 predict the trajectory that the potential collaborators will follow
24 and to estimate if recruiting them may be interesting or not ac-
25 cording to the existing sensing needs. In our approach, there is no
26 explicit recruiting process based on the trajectories of the vehicles,
27 which we assume that can be unknown (e.g., due to privacy rea-
28 sons): only vehicles in the neighborhood (within the communica-
29 tion range) of the vehicle carrying the mobile agent can participate
30 in the spatial crowdsourcing, and they may accept or not a poten-
31 tial collaboration when the agent asks for their help. Nevertheless,
32 the proposal in [52] could be used as a complementary technique
33 if access to those trajectories is enabled.

34 In [41], the authors present a sensing engine called *MoST* (*Mo-
35 bile Sensing Technology*), that can be used as a framework for devel-
36 oping location-aware applications for Android mobile phones. The
37 data read by the mobile phone sensors are transferred to a cen-
38 tral server by using the connection available on the phone (Wi-Fi,
39 3G/4G, etc.) and processed in the server. This work is not related
40 specifically to VANETs, but it illustrates the interest of sensing and
41 crowdsourcing applications.

42 In [53], the authors tackle the problem of recruiting vehicles
43 for public sensing in such a way that the selected vehicles cover
44 an area as large as possible at the lowest cost. To achieve this,
45 they present a *reputation-aware, trajectory-based recruitment (RTR)*
46 framework that handles the recruitment of vehicles for public
47 sensing. The framework considers the spatiotemporal availability
48 of participants along with their reputation to select vehicles that
49 should achieve the desired coverage of an area of interest with
50 a cost within the budget limitations. The framework consists of
51 a reputation assessment scheme, a pricing model, and a selection
52 scheme, that are combined to accomplish the objective of maxi-
53 mizing the coverage with the minimum cost. The authors propose
54 greedy heuristic solutions targeting the selection problem in real-
55 time. The RTR framework generalizes the basic selection problem
56 to handle some practical scenarios, including vehicles that leave
57 the area that needs to be monitored, and varying redundancy re-
58 quirements. An extensive performance evaluation shows that the
59 proposed greedy heuristics are able to achieve results close to
60 those previously obtained by optimal benchmarks under differ-
61 ent scenarios, and that the framework succeeds in achieving high
62 levels of coverage even when the vehicles do not stick to their an-
63 nounced trajectories. Another work related to the recruitment of
64 collaborators is [54], where the authors propose a model based on
65 genetic algorithms that selects the most appropriate group of par-
66 ticipants using parameters related to the users, the devices, and

67 the interest area. In our spatial crowdsourcing approach, a mobile
68 agent takes decisions locally based on information regarding the
69 neighbor vehicles, and therefore the collaboration takes place op-
70 portunistically rather than through an explicit recruitment process.

71 It might also be possible to consider more sophisticated crowd-
72 sourcing strategies with the goal to minimize the social cost
73 while at the same time ensuring a minimum compensation for
74 the driver. For example, in the context of ride-sharing for pas-
75 sengers, [55] provides a method to select a driver to maximize
76 the *overall shared route percentage (SRP)* subject to a minimum re-
77 quired value of this parameter for each driver (*expectation rate
78 of the driver*). This technique could be applied to locate vehicles
79 that, even if they are not at that moment within communication
80 range, could pick up the mobile agent and carry it to the intended
81 destination. However, applying this method in a vehicular ad hoc
82 network, where there is no global view of the environment and
83 the vehicles are usually constantly moving, is challenging.

84 Other works related to crowdsensing, but further from the
85 specifics of the spatial crowdsourcing approach described in this
86 paper, are [56,57]. In [56], the authors propose GENIUS-C, a frame-
87 work to support the development of spatial crowdsourcing plat-
88 forms. It is based on a generic architecture to reduce the gap be-
89 tween academy and industry, and is meant to decrease the devel-
90 opment cost and effort and increase the overall quality of spatial
91 crowdsourcing platforms. A case study is created using GENIUS-C
92 to demonstrate its benefits and how it can be used in the develop-
93 ment of spatial crowdsourcing platforms. In [57], the authors de-
94 scribe an algorithm to detect the accuracy of data that come from
95 crowdsensing sources such as sensors aboard vehicles (car sen-
96 sors), sensors built into mobile devices (phone sensors) carried by
97 one or more occupants, or both. They demonstrate, through eval-
98 uation in real environments (using infrastructure that captures car
99 and phone sensor readings and uploads them to a cloud database),
100 that their detection algorithms can achieve high accuracy for some
101 tasks related to the driver's behavior and the environment (e.g.,
102 higher than 90% for lane change determination) and that crowd-
103 sensing plays an indispensable role in improving the detection
104 performance (e.g., improving recall by 35% for lane change deter-
105 mination on curves). These are complementary works to ours, as
106 they focus on different issues related to crowdsensing.

107 Besides, although it does not specifically propose a crowdsen-
108 sing method, another relevant related work to mention is [58],
109 where the authors describe a hierarchical Bayesian non-parametric
110 approach for efficient and scalable route prediction, that can har-
111 ness *the wisdom of crowds* of route planning agents by aggregat-
112 ing their sequential routes of possibly-varying lengths and origin-
113 destination pairs. This approach has the advantages that it does
114 not require a Markov assumption and that it generalizes well with
115 sparse data, thus resulting in an improved prediction accuracy, as
116 the authors demonstrate empirically using real-world data about
117 trajectories of taxis.

118 Finally, regarding the use of incentives to encourage coopera-
119 tion, *OPPay* [59] is a payment system for opportunistic data ser-
120 vices (such as Wi-Fi sharing, content-based file sharing, and oppor-
121 tunistic networking), which implements a micropayment commu-
122 nication protocol for mobile devices to perform data transactions
123 and make payments using bitcoins. *OPPay* is intended to operate
124 using incremental payments that are resilient to interruptions in
125 the communications, and therefore it may be of interest in the
126 context of vehicular networks. Another interesting work regarding
127 this topic is presented in [60], where the authors propose a re-
128 warding mechanism for mobile crowdsourcing applications based
129 on auctions and biddings of the collected data, that focuses on
130 preserving the privacy of the collaborators in order to encourage
131 their participation. In our approach, we propose the use of vir-
132 tual money to pay collaborating vehicles, but the specific details

1 about how those payments can be made is out of the scope of this
2 paper. Other economic and incentive models have been proposed
3 for mobile P2P networks (e.g., [61,62]) and collaborative sensing in
4 general (e.g., [63–66]).

5.2. Related work on monitoring tasks with vehicles

8 The monitoring of certain parameters (e.g., polluting gases and
9 particles, pollen, or solar radiation) has been performed tradition-
10 ally using fixed sensing stations, that have a number of disad-
11 vantages, such as their lack of flexibility to perform a monitoring
12 task different from the one considered initially (e.g., monitor a dif-
13 ferent area), and the requirement of a certain infrastructure to
14 provide the energy and communication support required, which
15 also make them costly to build and maintain. As an alternative,
16 vehicles can be used as mobile monitoring platforms, since they
17 can be equipped with the necessary sensors and can travel to any
18 place (as long as a nearby road exists) to monitor the required
19 parameters. The sensors installed can be replaced by others if it
20 is required to monitor different parameters. Besides, the sensors
21 do not need a dedicated power or communication infrastructure,
22 since they can use the power generated by the vehicle's engine and
23 the wireless communication devices of the on-board computer. For
24 these reasons, the use of vehicles for performing monitoring tasks
25 has attracted the attention of the research community, and the
26 term *vehicular sensor network* has emerged [67,68]. In the follow-
27 ing, we present some representative works related to this topic, in
28 order to illustrate a variety of monitoring tasks that have attracted
29 research attention.

30 In [69], the authors assume the existence of a vehicular net-
31 work in which the vehicles carry sensors, and they pose the prob-
32 lem of determining how much information should be received
33 from the sensors and where should they be located to achieve
34 an optimum monitoring process. If the sensors send information
35 very frequently, then their readings would likely be similar, which
36 implies that bandwidth would be wasted. On the other hand, if
37 the sensors send a little amount of information, then the mea-
38 surements may be not accurate enough. That work is centered on
39 the mathematical aspect of such a method. It is considered that
40 sensors send their data to a central server by using cellular con-
41 nections instead of ad hoc communications. As stated before, and
42 also in studies such as [4], this has some disadvantages, such as
43 the existence of economic cost for the user.

44 In [70], the authors use information about the traffic status to
45 compute the best route to recharging points for electrical vehicles,
46 taking also into account the amount of energy available on board.
47 The traffic status is received from the surrounding cars, that send
48 information about their position and speed to estimate if the traf-
49 fic is light or dense. Ad hoc communications and a simple flooding
50 model for data dissemination is used. However, the proposal fo-
51 cuses only on sharing traffic data. In our proposal, instead, the data
52 sent by the vehicles can be of any type (not only the speed and po-
53 sition of vehicles) and the sensed data are carried out by mobile
54 agents, which can behave more efficiently than flooding algorithms
55 that continuously send all the data to all the neighbors. So, a mo-
56 bile agent can encapsulate any desired dissemination approach and
57 even commute from one approach to another depending on con-
58 text conditions (e.g., the density of vehicles in the area). With the
59 approach presented in this paper, only a limited amount of data is
60 sent, every time a mobile agent hops from one vehicle to another,
61 instead of sending multiple copies of the data to multiple receivers,
62 as it occurs with flooding algorithms. Besides, the proposal in [70]
63 does not exploit any spatial crowdsourcing approach.

64 A working prototype is presented in [71] that consists of a
65 monitoring VANET where the vehicles are equipped with sensors
66 and wireless communication devices that constantly look for Wi-

Fi access points to opportunistically send the collected data to a
67 central server. By contrast, in our approach the data are stored
68 in the vehicles instead of being all sent to a central server, and
69 they are processed distributively by mobile agents searching for
70 the most relevant data to perform the monitoring task. The system
71 presented in [71] was deployed in a real city and the vehicles par-
72 ticipating in the VANET belonged to the local authority, instead of
73 being open to any citizen, which has both advantages and draw-
74 backs.

75 In [72], the authors propose a method for route planning in a
76 city by using information collected opportunistically by the mov-
77 ing vehicles. The system operates in a decentralized way and it
78 exploits a technique based on the ant optimization problem, which
79 leads to routes that are not strictly optimal (as opposed to classic
80 shortest path search algorithms) but good enough to both reach
81 the intended destination and distribute the traffic flows along dif-
82 ferent streets so that they do not become congested. While the
83 approach is interesting, its application is limited to the problem of
84 route planning and its performance in low-traffic density scenarios
85 is not clear.

86 In [73], a smart parking system for locating available parking
87 slots is proposed. In this system, end users are not the only sub-
88 jects that collaborate by sharing information. Additionally, both the
89 parking controllers and the city administrators also participate by
90 providing information in a fast and integrated way. However, the
91 system operates in a centralized way, and a mobile 3G/4G connec-
92 tion is required to operate it.

93 In [74], a probabilistic vehicle routing algorithm is proposed,
94 based on a variant of the A* algorithm, to improve the sensing
95 coverage of vehicles. The idea is to reduce the optimality of A*
96 regarding the computation of shortest paths in exchange for a
97 better sensing coverage. As opposed to our proposal, special ve-
98 hicles (called probe vehicles) are considered to perform the spatial
99 crowdsourcing (in particular, taxis) rather than conventional cars.
100 Besides, the approach presented in [74] is not based on mobile
101 agents and has to be predefined in advance. Finally, it requires a
102 centralized entity to compute the paths, as global knowledge of
103 the sensing area covered and the trajectories is required.

104 In [75], the authors present a unified delay analysis framework
105 for opportunistic data collection, that integrates the *sensing and*
106 *transmission delays*, that are usually analyzed separately. A third
107 delay metric (that they call *data collection delay*) is added to the
108 analysis that can be performed by the framework, and thus the
109 QoS of opportunistic data collection applications can be measured
110 more comprehensively. The theoretical analysis of the framework is
111 validated by the authors by performing a number of simulations.
112 This work is complementary to ours, as it focuses on the analysis
113 of performance metrics.

114 Also related to delay aspects, in [76], the authors tackle the
115 problem of the sensor's reading delay, and they propose a solution
116 to mitigate it. Certain types of sensors (for example, some gas sen-
117 sors) take a relatively long time (about 20 to 50 seconds) to reach
118 a stable state and obtain a valid sample measurement. If these sen-
119 sors are carried by moving vehicles, they may have moved several
120 hundred meters in that lapse, thus obtaining data from a place
121 which is distant from the intended target. If this effect is not taken
122 into account, the measured parameters may be treated incorrectly
123 and they could lead to wrong conclusions. To avoid this, the au-
124 thors propose a method to calibrate the sensor readings by using
125 a filter on the raw data provided by the sensors, which also has as
126 a benefit the reduction of the time needed to obtain valid sensors
127 readings.

128 Finally, a recent work [77] analyzes crowdsourcing in Intelligent
129 Transportation Systems (ITS), but it does not focus on vehicular
130 networks nor spatial crowdsourcing. As another example, [78] pro-
131 poses a privacy-preserving vehicular urban sensing platform; how-

Table 11
Summary of advantages and disadvantages of several spatial crowdsourcing (SC) strategies.

SC strategy	Description	Advantages	Disadvantages
SC/PHCA	Pure hopper collecting agent	It needs less collaborators	The collecting data task in low-density traffic areas can be challenging
SC/SCCA	Spatial crowdsourcing collecting agent	It can collect more data in low-density traffic areas	It needs more collaborators
Without SC	It does not use SC	It does not need to compensate drivers	The whole monitoring process in low-density traffic areas can be challenging

ever, it relies on a cellular network, rather than when using ad hoc communications, and focuses on privacy issues and not on the use of spatial crowdsourcing.

6. Conclusions and future work

In this paper, we have presented a method that uses mobile agents for monitoring data in vehicular networks using spatial crowdsourcing. Specifically, in order to enhance the efficiency of the mobile agents when they perform their tasks, an agent has the possibility of requesting other collaborating users to modify their vehicles' routes to transport the agent to a more convenient place (i.e., closer to the target area) when its approaching speed is too low. We have performed an extensive set of tests by using a realistic simulator of vehicular networks and real road maps, and the results obtained show that the use of the proposed spatial crowdsourcing technique can increase the performance of the monitoring process, allowing the agent to reach the interest area faster, with a more efficient use of the wireless communications bandwidth and a higher data collection rate than when spatial crowdsourcing is not used. We quantify the results and precisely determine the extent of the improvement, its costs, and under which conditions those benefits are obtained. According to the experimental results presented, the proposed strategy is suitable even in conditions with a good traffic of vehicles, as it adapts itself to the existing conditions in order to make an efficient use of collaborators, commuting between using collaborators and hopping as needed. Therefore, using the spatial crowdsourcing approach presented is highly beneficial in low-traffic scenarios and it does not affect adversely the time performance of the monitoring process in any situation. Nevertheless, the experimental results show that several parameters could be adjusted depending on the existing constraints and preferred metrics. For example, if in the application scenario the agent is needed to spend as little virtual money as possible, then the minimum approaching speed threshold and the minimum stay time should be set to low values, although the amount of data collected and the reliability of the process will likely be negatively affected. Table 11 shows a summary with the main advantages and disadvantages of the different spatial crowdsourcing strategies considered.

As future work, we would like to analyze the use of possible variants and extensions of our proposed spatial crowdsourcing techniques and monitoring strategies and evaluate them in a wide variety of scenarios. Besides, it is interesting to study in more detail different payment methods that could be considered and their impact, considering desired qualities regarding the reliability, security, privacy, fairness and trustfulness of the system. The promising results that we have obtained with this approach encourages us to continue this research line and explore these and other possibilities.

Acknowledgements

This research has been supported by the projects TIN2016-78011-C4-3-R (AEI/FEDER, UE) and DGA-FSE (COS2MOS research group). We also thank the support of the TRAFIR project

(2017-EU-IA-0167), co-financed by the Connecting Europe Facility of the European Union. The contents of this publication are the sole responsibility of its authors and do not necessarily reflect the opinion of the European Union. The maps of the cities used in the experimental evaluation are derived from data obtained from OpenStreetMap. Last but not least, we thank the reviewers for their useful and insightful comments, that have helped to improve this paper.

References

- [1] J.J. Blum, A. Eskandarian, L.J. Hoffman, Challenges of intervehicle ad hoc networks, *IEEE Trans. Intell. Transp. Syst.* 5 (4) (2004) 347–351.
- [2] H. Hartenstein, L.P. Laberteaux, A tutorial survey on vehicular ad hoc networks, *IEEE Commun. Mag.* 46 (6) (2008) 164–171.
- [3] S. Olariu, M.C. Weigle, *Vehicular Networks: From Theory to Practice*, 1st edition, Chapman & Hall/CRC, 2009.
- [4] S. Ilarri, T. Delot, R. Trillo-Lado, A data management perspective on vehicular networks, *IEEE Commun. Surv. Tutor.* 17 (4) (2015) 2420–2460.
- [5] B. Fleming, Sensors – a forecast [automotive electronics], *IEEE Veh. Technol. Mag.* 8 (3) (2013) 4–12.
- [6] ABI Research, Sensor data crowdsourcing to transform connected car services, <https://www.abiresearch.com/press/sensor-data-crowdsourcing-transform-connected-car-services/>, May 2018, last access on February 6, 2019.
- [7] ABI Research, Transformative connected car services built on data crowdsourcing, <https://www.abiresearch.com/market-research/product/1027945-transformative-connected-car-services-buil/>, 2019, last access on February 6.
- [8] O. Urria, S. Ilarri, R. Trillo-Lado, An approach driven by mobile agents for data management in vehicular networks, *Inf. Sci.* 381 (2017) 55–77.
- [9] O. Urria, S. Ilarri, Towards spatial crowdsourcing in vehicular networks using mobile agents, in: 20th East-European Conference on Advances in Databases and Information Systems (ADBIS), vol. 637, Springer, 2016, pp. 88–95.
- [10] L.-W. Chen, Y.-C. Tseng, K.-Z. Syue, Surveillance on-the-road: vehicular tracking and reporting by V2V communications, *Comput. Netw.* 67 (2014) 154–163.
- [11] J. Zuo, H. Xia, S. Liu, Y. Qiao, Mapping urban environmental noise using smartphones, *Sensors* 16 (10) (2016) 1692.
- [12] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, W. Trappe, ParkNet: drive-by sensing of road-side parking statistics, in: Eighth International Conference on Mobile Systems, Applications, and Services, MobiSys, ACM, 2010, pp. 123–136.
- [13] Inria, SoundCity: a mobile application for understanding your exposure to noise pollution, <https://mimove.inria.fr/soundcity>, 2019, last access on February 6.
- [14] G. Dimitrakopoulos, P. Demestichas, Intelligent transportation systems, *IEEE Veh. Technol. Mag.* 5 (1) (2010) 77–84.
- [15] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: a survey, *IEEE Trans. Intell. Transp. Syst.* 12 (4) (2011) 1624–1639.
- [16] S. Al-Sultan, M.M. Al-Doori, A.H. Al-Bayatti, H. Zedan, A comprehensive survey on vehicular ad hoc network, *J. Netw. Comput. Appl.* 37 (2014) 380–392.
- [17] D. Jiang, L. Delgrossi, IEEE 802.11p: towards an international standard for wireless access in vehicular environments, in: IEEE Vehicular Technology Conference, VTC Spring, IEEE Computer Society, 2008, pp. 2036–2040.
- [18] R.A. Uzcategui, A.J.D. Sucre, G. Acosta-Marum, WAVE: a tutorial, *IEEE Commun. Mag.* 47 (5) (2009) 126–133.
- [19] B.T. Sheref, R.A. Alsaqour, M. Ismail, Vehicular communication ad hoc routing protocols: a survey, *J. Netw. Comput. Appl.* 40 (2014) 363–396.
- [20] D. Milojicic, F. Douglis, R. Wheeler, *Mobility: Processes, Computers, and Agents*, ACM, 1999.
- [21] R. Trillo, S. Ilarri, E. Mena, Comparison and performance evaluation of mobile agent platforms, in: Third International Conference on Autonomic and Autonomous Systems, ICAS, IEEE Computer Society, 2007, pp. 41–46.
- [22] O. Urria, S. Ilarri, E. Mena, Agents jumping in the air: dream or reality?, in: 10th International Work-Conference on Artificial Neural Networks (IWANN), Special Session on Practical Applications of Agents and Multi-Agent Systems, Springer, 2009, pp. 627–634.

- [23] S. Ilarri, R. Trillo-Lado, E. Mena, SPRINGS: a scalable platform for highly mobile agents in distributed computing environments, in: Fourth International WoW-MoM 2006 Workshop on Mobile Distributed Computing, MDC, IEEE Computer Society, 2006, pp. 633–637.
- [24] S. Ilarri, P. Roig, R. Trillo-Lado, GeoSPRINGS: towards a location-aware mobile agent platform, in: 16th International Symposium on Web and Wireless Geographical Information Systems, W2GIS, in: Lecture Notes in Computer Science (LNCS), vol. 10819, Springer, 2018, pp. 51–60.
- [25] O. Urra, S. Ilarri, Using mobile agents in vehicular networks for data processing, in: 14th International Conference on Mobile Data Management (MDM), Ph.D. Forum, vol. 2, IEEE Computer Society, 2013, pp. 11–14.
- [26] A. Boubrima, W. Bechkit, H. Rivano, Optimal WSN deployment models for air pollution monitoring, *IEEE Trans. Wirel. Commun.* 16 (5) (2017) 2723–2735.
- [27] M. Bacco, F. Delmastro, E. Ferro, A. Gotta, Environmental monitoring for smart cities, *IEEE Sens. J.* 17 (23) (2017) 7767–7774.
- [28] C.T. Calafate, K. Cicenía, O. Alvear, J.C. Cano, P. Manzoni, Estimating rainfall intensity by using vehicles as sensors, in: *Wireless Days*, IEEE Computer Society, 2017, pp. 21–26.
- [29] C. Lochert, B. Scheuermann, M. Caliskan, M. Mauve, The feasibility of information dissemination in vehicular ad-hoc networks, in: Fourth Annual Conference on Wireless on Demand Network Systems and Services, WONS, 2007, pp. 92–99.
- [30] T. Delot, N. Mitton, S. Ilarri, T. Hien, GeoVanet: a routing protocol for query processing in vehicular networks, *Mob. Inf. Syst.* 7 (4) (2011) 329–359.
- [31] O. Urra, S. Ilarri, MAVSIM: testing VANET applications based on mobile agents, in: CRC Taylor and Francis Group, 2016, pp. 199–224, Ch. 10.
- [32] Y. Zhao, Q. Han, Spatial crowdsourcing: current state and future directions, *IEEE Commun. Mag.* 54 (7) (2016) 102–107.
- [33] OpenStreetMap Foundation (OSMF), Openstreetmap, <http://openstreetmap.org>, 2019, last access on February 6.
- [34] M. Ruiz, L. Velasco, Background, 1st edition, Wiley Publishing, 2017, pp. 15–42, Ch. 2.
- [35] T. Darwish, K.A. Bakar, Traffic density estimation in vehicular ad hoc networks: a review, *Ad Hoc Netw.* 24 (2015) 337–351.
- [36] B.S. Kerner, Introduction to Modern Traffic Flow Theory and Control, Springer, 2009.
- [37] F.J. Martinez, M. Fogue, C.K. Toh, J.-C. Cano, C.T. Calafate, P. Manzoni, Computer simulations of VANETs using realistic city topologies, *Wirel. Pers. Commun.* 69 (2) (2012) 639–663.
- [38] J. Barrachina, P. Garrido, M. Fogue, F.J. Martinez, J.-C. Cano, C.T. Calafate, P. Manzoni, Road side unit deployment: a density-based approach, *IEEE Trans. Intell. Transp. Syst.* 5 (3) (2013) 30–39.
- [39] R.K. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges, *IEEE Commun. Mag.* 49 (11) (2011) 32–39.
- [40] X. Sheng, J. Tang, W. Zhang, Energy-efficient collaborative sensing with mobile phones, in: 31st Annual IEEE International Conference on Computer Communications, INFOCOM, 2012, pp. 1916–1924.
- [41] G. Cardone, A. Cirri, A. Corradi, L. Foschini, R. Ianniello, R. Montanari, Crowdsensing in urban areas for city-scale mass gathering management: geofencing and activity recognition, *IEEE Sens. J.* 14 (12) (2014) 4185–4195.
- [42] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C.C. Cao, Y. Tong, C.J. Zhang, gMission: a general spatial crowdsourcing platform, *Proc. VLDB Endow.* 7 (13) (2014) 1629–1632.
- [43] H. To, G. Ghinita, C. Shahabi, A framework for protecting worker location privacy in spatial crowdsourcing, *Proc. VLDB Endow.* 7 (10) (2014) 919–930.
- [44] B. Guo, Z. Wang, Z. Yu, Y. Wang, N.Y. Yen, R. Huang, X. Zhou, Mobile crowd sensing and computing: the review of an emerging human-powered sensing paradigm, *ACM Comput. Surv.* 48 (1) (2015) 7.
- [45] Y. Tong, L. Chen, C. Shahabi, Spatial crowdsourcing: challenges, techniques, and applications, *Proc. VLDB Endow.* 10 (12) (2017) 1988–1991.
- [46] Y. Tong, J. She, B. Ding, L. Wang, L. Chen, Online mobile micro-task allocation in spatial crowdsourcing, in: 32nd International Conference on Data Engineering, ICDE, IEEE Computer Society, 2016, pp. 49–60.
- [47] H. To, C. Shahabi, L. Kazemi, A server-assigned spatial crowdsourcing framework, *ACM Trans. Spat. Algorithms Syst.* 1 (1) (2015) 2.
- [48] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, J. Zhao, Reliable diversity-based spatial crowdsourcing by moving workers, *Proc. VLDB Endow.* 8 (10) (2015) 1022–1033.
- [49] U. ul Hassan, E. Curry, Efficient task assignment for spatial crowdsourcing: a combinatorial fractional optimization approach with semi-bandit learning, *Expert Syst. Appl.* 58 (2016) 36–56.
- [50] H. To, Task assignment in spatial crowdsourcing: challenges and approaches, in: Third ACM SIGSPATIAL PhD Symposium, ACM, 2016, 1.
- [51] Y. Wang, H. Li, T. Li, Participant selection for data collection through device-to-device communications in mobile sensing, *Pers. Ubiquitous Comput.* 21 (1) (2017) 31–41.
- [52] E. Wang, Y. Yang, J. Wu, W. Liu, X. Wang, An efficient prediction-based user recruitment for mobile crowdsensing, *IEEE Trans. Mob. Comput.* 17 (1) (2018) 16–28.
- [53] S. Abdelhamid, H.S. Hassanein, G. Takahara, Reputation-aware, trajectory-based recruitment of smart vehicles for public sensing, *IEEE Trans. Intell. Transp. Syst.* 19 (5) (2018) 1387–1400.
- [54] R. Azzam, R. Mizouni, H. Otrok, A. Ouali, S. Singh, GRS: a group-based recruitment system for mobile crowd sensing, *J. Netw. Comput. Appl.* 72 (2016) 38–50.
- [55] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma, Z. Gong, An efficient ride-sharing framework for maximizing shared route, *IEEE Trans. Knowl. Data Eng.* 30 (2) (2018) 219–233.
- [56] A.S. Fonteles, S. Bouveret, J. Gensel, A programming framework for spatial crowdsourcing, in: 15th International Conference on Advances in Mobile Computing & Multimedia, MoMM, ACM, 2017, pp. 131–140.
- [57] H. Qiu, J. Chen, S. Jain, Y. Jiang, M. McCartney, G. Kar, F. Bai, D.K. Grimm, M. Gruteser, R. Govindan, Towards robust vehicular context sensing, *IEEE Trans. Veh. Technol.* 67 (3) (2018) 1909–1922.
- [58] J. Yu, K.H. Low, A. Oran, P. Jaillet, Hierarchical Bayesian nonparametric approach to modeling and learning the wisdom of crowds of urban traffic route planning agents, in: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 2, 2012, pp. 478–485.
- [59] F. Shi, Z. Qin, J.A. McCann, OPPay: design and implementation of a payment system for opportunistic data services, in: 37th International Conference on Distributed Computing Systems, ICDCS, IEEE Computer Society, 2017, pp. 1618–1628.
- [60] T. Dimitriou, I. Krontiris, Privacy-respecting auctions and rewarding mechanisms in mobile crowd-sensing applications, *J. Netw. Comput. Appl.* 100 (2017) 24–34.
- [61] O. Wolfson, B. Xu, P. Sistla, An economic model for resource exchange in mobile peer to peer networks, in: 16th International Conference on Scientific and Statistical Database Management, SSDBM, IEEE Computer Society, 2004, pp. 235–244.
- [62] L.-Y. Yeh, Y.-C. Lin, A proxy-based authentication and billing scheme with incentive-aware multihop forwarding for vehicular networks, *IEEE Trans. Intell. Transp. Syst.* 15 (4) (2014) 1607–1621.
- [63] L.G. Jaimes, I.J. Vergara-Laurens, A. Raji, A survey of incentive techniques for mobile crowd sensing, *IEEE Int. Things J.* 2 (5) (2015) 370–380.
- [64] H. Gao, C.H. Liu, W. Wang, J. Zhao, Z. Song, X. Su, J. Crowcroft, K.K. Leung, A survey of incentive mechanisms for participatory sensing, *IEEE Commun. Surv. Tutor.* 17 (2) (2015) 918–943.
- [65] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, X. Mao, Incentives for mobile crowd sensing: a survey, *IEEE Commun. Surv. Tutor.* 18 (1) (2016) 54–67.
- [66] B. Di, T. Wang, L. Song, Z. Han, Collaborative smartphone sensing using overlapping coalition formation games, *IEEE Trans. Mob. Comput.* 16 (1) (2017) 30–43.
- [67] U. Lee, M. Gerla, A survey of urban vehicular sensing platforms, *Comput. Netw.* 54 (4) (2010) 527–544.
- [68] J.R. Vacca, Handbook of Sensor Networking: Advanced Technologies and Applications, CRC Press, 2015.
- [69] L. Liu, W. Wei, D. Zhao, H. Ma, Urban resolution: new metric for measuring the quality of urban sensing, *IEEE Trans. Mob. Comput.* 14 (12) (2015) 2560–2575.
- [70] Y. Wang, J. Jiang, T. Mu, Context-aware and energy-driven route optimization for fully electric vehicles via crowdsourcing, *IEEE Trans. Intell. Transp. Syst.* 14 (3) (2013) 1331–1345.
- [71] P.M. Santos, T. Calçada, D. Guimarães, T. Condeixa, S. Sargento, A. Aguiar, J.A. Barros, Demo: platform for collecting data from urban sensors using vehicular networking, in: 21st Annual International Conference on Mobile Computing and Networking, MobiCom, ACM, 2015, pp. 167–169.
- [72] D. Cerotti, S. Distefano, G. Merlino, A. Puliafito, A crowd-cooperative approach for intelligent transportation systems, *IEEE Trans. Intell. Transp. Syst.* 18 (6) (2016) 1529–1539.
- [73] A. Grazioli, M. Picone, F. Zanichelli, M. Amoretti, Collaborative mobile application and advanced services for smart parking, in: 14th International Conference on Mobile Data Management (MDM), vol. 2, IEEE Computer Society, 2013, pp. 39–44.
- [74] D. Asprone, S. Di Martino, P. Festa, Improving sensing coverage of probe vehicles with probabilistic routing, in: M.R. Luaces, F. Karimipour (Eds.), 16th International Symposium on Web and Wireless Geographical Information Systems, W2GIS, Springer, 2018, pp. 1–10.
- [75] D. Zhao, H. Ma, Q. Li, S. Tang, A unified delay analysis framework for opportunistic data collection, *Wirel. Netw.* 24 (4) (2018) 1313–1325.
- [76] X. Liu, X. Chen, X. Xu, E. Mai, H.Y. Noh, P. Zhang, L. Zhang, Delay effect in mobile sensing system for urban air pollution monitoring, in: 15th ACM Conference on Embedded Network Sensor Systems, SenSys, ACM, 2017, 73.
- [77] X. Wang, X. Zheng, Q. Zhang, T. Wang, D. Shen, Crowdsourcing in ITS: the state of the work and the networking, *IEEE Trans. Intell. Transp. Syst.* 17 (6) (2016) 1596–1605.
- [78] C. Zuo, K. Liang, Z.L. Jiang, J. Shao, J. Fang, Cost-effective privacy-preserving vehicular urban sensing system, *Pers. Ubiquitous Comput.* 21 (5) (2017) 893–901.