

# Towards Uniform Urban Map Coverage in Vehicular Crowd-Sensing: A Decentralized Incentivization Solution

SERGIO DI MARTINO<sup>ID</sup> AND LUIGI LIBERO LUCIO STARACE<sup>ID</sup>

Department of Information Technology and Electrical Engineering, Università degli Studi di Napoli Federico II, 80125 Naples, Italy

CORRESPONDING AUTHOR: L. L. L. STARACE (e-mail: luigiliberolucio.starace@unina.it)

**ABSTRACT** Vehicular Crowd-Sensing (VCS) is a well-known data collection approach leveraging sensors of connected vehicles to efficiently gather contextual information in urban environments. High-mileage vehicles such as taxis are often regarded as effective VCS platforms, due to their pervasiveness in modern cities, even though the road network coverage achievable by these vehicles is still an open issue. Indeed, their drivers generally follow the most-efficient route to destination, leading to major roads being frequently visited, while others are often neglected. To address this issue, many centralized *incentivization* solutions have been proposed to recruit/reward drivers accepting minor detours towards roads with higher sensing demand. However, these works mostly focus on assigning specific sensing tasks to drivers, rather than achieving an overall better-balanced urban sensing coverage, which is nonetheless required for many use cases, such as air quality monitoring. To fill this gap, we present ROU<sub>TR</sub>, an incentivization budget-aware routing solution designed to achieve more uniform coverage in VCS without requiring central coordination, thus significantly reducing back-end infrastructure costs. We empirically evaluated the proposal using taxi traces collected in the City of San Francisco. Results highlighted that, even with small incentivization budgets, our proposal leads to significantly more uniform urban road network coverage.

**INDEX TERMS** Internet of Vehicles, vehicular crowd-sensing, routing, smart cities, intelligent transportation systems.

## I. INTRODUCTION

IN THE era of the Internet of Things (IoT), a significant fraction of devices is expected to consist of connected vehicles, giving rise to the concept of the Internet of Vehicles (IoV). In the IoV, each vehicle is considered as a moving sensing entity, equipped with on-board environmental sensors (e.g., radars, cameras, accelerometers, sun/rain intensity, air quality monitoring) and computing facilities/storage, and connected to a remote back-end by means of mobile communication technologies [38]. Many new solutions can be developed on top of the IoV paradigm, including Vehicular Crowd-Sensing (VCS), a special case of Mobile Crowd-Sensing, in which the data collection is performed by vehicles acting as probes, sensing information

from the environment around them in an *opportunistic* fashion, i.e., without requiring the driver to explicitly trigger the sensing [12], [48]. When data sensed from swarms of vehicles is aggregated on a back-end, it can be used to generate an unprecedented amount of spatio-temporal information/knowledge, which could be used to enable many new exciting and valuable use cases, such as more accurate traffic predictions [40] and real-time on-street parking availability [5], air quality [11] or weather monitoring [33], better surveillance of urban scenarios [27], and so on. Hence, it is not surprising that the value of IoV data is estimated to be worth between US\$11.6 billion and US\$92.6 billion for the US market alone [39].

Most of the VCS-related investigations in the literature suggest to exploit high-mileage vehicles fleets, such as ridesharing vehicles operating for taxi services or

The review of this article was arranged by Associate Editor Jiwon Kim.

Transportation Network Companies (TNCs) like Uber or Lyft, to crowd-sense data, due to their long operational times and pervasive presence in modern cities [9]. However, since the priority of the drivers of ridesharing vehicles is to find customers and deliver them to their destination as efficiently as possible, the resulting collected data might be insufficient in some parts of the urban road network [44]. To clarify, if all the drivers were routed by a shortest-path algorithm, each trajectory going, for example, from the train station to the business district in a modern city, would mostly cover the same major thoroughfares, possibly neglecting minor, adjacent roads.

Many prior works (e.g., [7], [10], [16], [44], [49]) have proposed suitable incentive mechanisms to re-route these vehicles towards areas/streets in which there is a sensing need, offering some sort of monetary and/or social incentivization to compensate drivers willing to accept the re-routing. Nevertheless, to the best of our knowledge, most of these works focus on an *ad-hoc* sensing scenario, in which a central system, aware of sensing demands over the entire map, assigns incentivized *tasks* (e.g., monitoring traffic congestion around a stadium before an event) to a group of potential contributors, trying to satisfy the sensing demands under budgetary constraints. On the other hand, very little research has been aimed at obtaining, in a cost-effective way, an overall broader and more uniform sensing coverage (e.g., [1], [24]), even if this is required for a multitude of use cases, such as air quality monitoring in a urban scenario. Moreover, most incentivization mechanisms presented in the literature require a centralized, sensing demand-aware computing infrastructure to decide which vehicles should be recruited for sensing [47], to compute their new trajectories and to allocate the given incentivization budget. A major drawback of these architectures is that such a centralized, mission-critical computing infrastructure can be very costly to deploy and operate, especially at a nation- or region-wide scale, limiting the budget available for incentivization [2].

In this paper we propose a VCS incentivization solution geared towards achieving a more uniform urban road network coverage, while reducing the overall infrastructural and operational costs. The rationale behind the proposal is that each vehicle computes its own trajectories by randomly generating a possibly sub-optimal route compatible with a given incentivization budget. This way, when different vehicles compute a route from the same given origin to the same given destination, each of them is likely to generate a slightly different route, thus achieving an overall more uniform distribution over the urban road network.

More in detail, the core of the proposal is ROUTr, a specifically-designed routing algorithm, intended a semi-admissible evolution of  $A^*$  [19], capable of taking into account incentivization budget constraints. With ROUTr, the burden of computing routes is shifted from the centralized back-end towards the ‘edges’ of the sensing architecture, intended as either on-board navigation devices, or

tablet/smartphones. Such a distributed computing architecture, consisting of ubiquitously connected heterogeneous devices at the edge of the network can be considered an instance of the *fog computing* paradigm [46]. To obtain a more uniform urban sensing distribution without a central coordination, the algorithm leverages a random factor to compute routes in a probabilistic way, guaranteeing that a given per-ride incentivization budget constraint is satisfied.

We formalize the set of admissible routes satisfying the incentivization budget constraint by leveraging the concept of Potential Path Area (PPA) [31], often applied in many fields to quantitatively analyze and describe the spatial behaviour of people. Let us note that the proposed solution is based on some intuitions presented in [2]. That work, however, only reported the feasibility of a decentralized, randomized approach to improve the spatio-temporal sensing coverage of probe vehicles, not taking into account the key issues of incentivization and budgetary constraints, which are crucial to deploy VCS solutions.

We performed an empirical evaluation of the effectiveness of the proposed approach on a publicly available massive dataset of more than 400 000 real taxi trajectories, collected in San Francisco (USA). In particular we computed, at a fine-grained street-segment level, the potential spatio-temporal coverage improvements achievable by using our solution, with eight different incentivization budgets, w.r.t. to a shortest path algorithm, such as  $A^*$  [19]. The results highlight that, even with an incentivization budget of just \$ 500 per three-weeks for a fleet of hundreds of taxis, the proposed solution can lead to higher spatio-temporal coverage, with significant improvements especially for minor roads.

The main contributions of this paper can be summarized as follows:

- 1) A novel decentralized solution is presented, specifically designed to incentivize a more uniform urban road network coverage for VCS leveraging high-mileage vehicles.
- 2) An empirical evaluation of the achievable spatio-temporal sensing coverage has been conducted, leveraging real-world trajectory data, with 8 different budget constraints, at a road segment granularity.
- 3) Valuable insights on the achievable road-network coverage and incentivization costs (and their trade-offs) are provided to Decision Makers investigating the feasibility of leveraging high-mileage vehicle fleets to crowd-sense information in urban environments.

The remainder of the paper is organized as follows. In Section II the key concepts of IoV and VCS are described and an overview on the state-of-the-art on incentivization approaches is given. Section III presents the proposed solution, while Section IV describes the empirical study that was put in place to assess its effectiveness. In Section V, the results of this empirical analysis are reported and discussed. Conclusions and final remarks are given in Section VI.

## II. RELATED WORKS

VCS aims at exploiting on-board sensors installed in modern connected vehicles to opportunistically *crowd-sense* contextual information (e.g., [5], [13], [15], [29]). Such pervasive, real-time data gathered through VCS enables many novel and interesting use cases, ranging from the monitoring of spatio-temporal phenomena of interest to the creation of smarter *Intelligent Transportation Systems* [41]. In practice, however, vehicles are not uniformly distributed over the road-network [4], limiting the feasibility of many VCS-based use cases [49]. To overcome this limitation, many researchers have proposed ad-hoc solutions to help achieve an adequate distribution of the collected data to support VCS use cases.

Broadly speaking, two main categories of solutions have been proposed in the literature to support different VCS scenarios: those designed to efficiently assign ad-hoc sensing tasks to selected vehicles, and those aiming at achieving an overall broader and/or more uniform sensing distribution over the entire road-network. Solutions belonging to the former category typically focus on selecting which vehicles should participate in the crowd-sensing activity (e.g., [42]) and/or on reassigning vehicles to urban areas in which there is a sensing need (e.g., [44]), offering some form of monetary or social reward to drivers willing to accept the re-assignment.

Solutions belonging to the latter category, on the other hand, typically involve custom routing algorithms that aim at distributing vehicles more uniformly on the road network, and do not consider the problem of incentivizing drivers to accept possibly sub-optimal routes (e.g., [2], [23], [24]). Thus, the solution we propose is not directly comparable with any of the above-mentioned approaches, as it combines the idea of a custom routing algorithm designed to achieve a more uniform sensing coverage with the key concepts of incentivization and budgetary constraints that are typically used only in solutions supporting ad-hoc sensing tasks. In the following we summarize the state of the art for both the above-mentioned categories.

### A. SOLUTIONS SUPPORTING AD-HOC SENSING TASKS

This class of incentive systems deal with scenarios in which a number of specific sensing tasks (e.g., visiting a particular street at a given time) need to be carried out, often framing the challenge of finding the most cost-effective budget allocation as an optimization problem. In [42], for example, the authors focus on deciding which vehicles to recruit to fulfill minimum coverage requirements while minimizing the incentivization costs, by leveraging the predictability of vehicular traffic. Similarly, in [20] the problem of participant selection leveraging trajectory prediction is addressed, but with the goal of maximizing coverage under some budgetary constraints. In that work, two solutions, based on a greedy and a genetic algorithm, respectively, are proposed to solve the participant recruitment problem, and their performance is evaluated using real traffic trace datasets. Reference [48]

considers a more general scenario with two kinds of sensing tasks: general sensing (e.g., the need to cover a given area as much as possible) and location-based query tasks (e.g., the need to specifically cover a given road segment). The authors define a multi-objective optimization approach that tries to maximize the utility of each recruited sensing vehicle by making it simultaneously complete both kind of tasks. In [44], the authors propose an incentivization under budgetary-constraints approach in which a sensing vehicle (e.g., a taxi waiting for a client) is incentivized to move to a different zone of the city to carry out a sensing task. To make a better use of the available budget, the authors devised a new way of computing rewards for drivers, offering a combination of monetary incentives with an increased probability of potential taxi riding requests at the destination. Reference [16] investigates a different formulation of the problem, contemplating the possibility that a vehicle might fail in performing a sensing task, for example due to sensor errors. In this *non-deterministic* VCS context, a sensing task might be performed by multiple vehicles in order to maximize the probability of success. In that work, the authors propose a reverse-auction-based incentive mechanism that includes an approximation algorithm to select winning bids and a payment algorithm to determine payments for all participants. Reference [21] addresses the problem of recruiting sensing vehicles in a way that satisfies given sensing quality, redundancy and quality requirements. That approach is based on a distributed vehicle-ranking scheme, in which each connected vehicle classifies itself as relevant w.r.t. a certain sensing task, and on a centralized recruitment back-end which uses a game-theoretical approach to select the best vehicles to recruit. In [49], the authors investigate the impact of including vehicular social networks effect and intrinsic rewards into incentive mechanisms design. In that work, the authors envision that vehicles benefiting from the data sensed by other participants (e.g., in real-time traffic information use cases), will be more willing to participate in the crowd-sensing due to such an intrinsic reward, even without explicit monetary payoff. The authors then propose a social-aware incentive mechanism by deep reinforcement learning to derive the optimal long term sensing strategy for all vehicles.

Even though these ad-hoc sensing solutions could be used to support VCS-based use cases requiring an overall broader and more uniform distribution, for example by uniformly generating sensing tasks, all of these related works suffer from two major limitations. Firstly, they consider sensing coverage at a very coarse-grained scale of city areas, with each of such areas being usually much bigger than one city block. Our work, on the other hand, considers sensing coverage at road segment-level, relying on the real topology of the road network, achieving a detail that is crucial when dealing with VCS-based use cases such as on-street parking availability or pothole monitoring. Secondly, all these approaches (e.g., [44], [48]) require, to some extent, a centralized component which might be difficult to deploy and expensive to operate at metropolis or region-level scale. Asprone et al.,

in [2], quantified these costs: in the Municipality of San Francisco (USA) the Transportation Network Companies (TNCs), like Uber or Lyft, served on average 170 000 trips per day [36]. Having their routes calculated in the cloud with Amazon Web Services would cost roughly 72 000\$ per year, just for the computational resources, without considering other costs such as data access, load balancing, network transfers, etc., and other factors, such as the need of guaranteeing short response times and high reliability. When scaled on a nation-wide level, these costs might rise to millions of dollars per year, which might significantly reduce the amount of economic resources available for the incentivization [2].

### B. SOLUTIONS AIMING AT UNIFORM SENSING DISTRIBUTION

To date, less work (e.g., [2], [23], [24]) has been directed towards incentive systems specifically geared towards achieving a more uniform sensing distribution in urban scenarios. In [23], [24], for example, Masutani investigates the adoption of suitably-designed centralized routing strategies to achieve a more uniform sensing coverage and reduce traffic congestion. In those works, the routes for the probe vehicles are computed by a centralized system, which continuously updates the costs associated with the traversal of each road segment by lowering the costs for segments for which there is a sensing need, and increasing the costs for segments that have been recently visited by a probe vehicle. Thanks to such dynamic weighting of the road segments, the centralized system can achieve a more uniform sensing distribution by simply computing routes for the probe vehicles using shortest-path algorithms. Although being among the first works to investigate routing solutions to achieve a generally broader sensing distribution in VCS, the approach presented in those works does not consider the key issues of incentivization and budget constraints and is affected by the same limitations as ones supporting ad-hoc sensing tasks. They rely on a centralized component, which might be expensive to operate when deployed at large scale. Moreover, the empirical evaluation of those solutions is based on traffic simulations and not on real-world data, and considers coverage at a very coarse-grained scale of city areas, not providing useful insights for many VCS-based applications.

The first decentralized approach to achieve a broader and more uniform sensing coverage was presented in [2]. In that work, the authors describe  $RA_{\epsilon}^*$ , a probabilistic routing algorithm designed to increase spatio-temporal road-network coverage by computing slightly sub-optimal routes, featuring small detours from the shortest route to destination. The solution has been evaluated using real-world taxi trajectories, and a fine-grained analysis of the achieved coverage conducted at road-segment level highlighted that decentralized approaches could achieve promising results. However, that work does not take into consideration two key factors: (I) incentivization, which is necessary for drivers to accept the proposed sub-optimal routes, and (II) budgetary

constraints, which are crucial to Decision Makers aiming at deploying VCS-based solutions.

### III. THE PROPOSED VCS INCENTIVIZATION SOLUTION

This section is aimed at presenting the decentralized, incentivization budget-aware routing solution we propose. Firstly, we describe the rationale behind the proposed approach, and its novelty w.r.t. other incentivization mechanisms. Then, we provide some preliminary notions and definitions for routing algorithms and VCS incentivization, necessary to formalize the proposed routing solution. Finally, we describe in detail the proposed solution.

#### A. RATIONALE AND NOVELTY

When dealing with VCS in urban environments, privately-owned vehicles are generally not considered an effective solution, since they are parked most of the time and are active mostly at rush hours [22]. Ride-sharing vehicles fleets (e.g., taxis and vehicles operating for Transportation Network Companies such as Uber or Lyft), on the other hand, are often considered as a most effective platform, due to the long operational times and pervasiveness of their vehicles in modern urban environments [9]. Drivers of ridesharing vehicles, however, typically choose the most efficient route from their origin to destination, to save money and time. As a consequence, without any incentivization mechanism in place, drivers potentially involved in a VCS activity would mostly drive through the main city thoroughfares, which as a result will be sensed very frequently, whereas minor, adjacent streets would be rarely (if at all) visited.

For a Smart City willing to implement use cases requiring a more pervasive sensing distribution, it is necessary to re-route the vehicles out of these main thoroughfares, in exchange for an explicit reward for the driver/passengers willing to accept the re-routing towards a potentially less efficient route (e.g., [20], [42]).

Most of the related works on VCS incentivization are based on a centralized, sensing demand-aware back-end, which knows the position of the involved vehicles and of the sensing demands. Using this knowledge, the back-end can compute paths in such a way that vehicles are routed through areas that require a higher sensing level (e.g., [10], [44]).

The downsides of this strategy are related to the provisioning and operation costs for such a complex, mission-critical and centralized back-end infrastructure. In [2], a quantification of these costs is provided: for the San Francisco (USA) urban area, it would cost roughly \$ 72 000 per year, just for the computing needs, to compute routes for ride-sharing vehicles in the cloud with Amazon Web Services. When scaled on a nation-wide level, these costs might rise to millions of dollars per year, significantly reducing the potential monetary resources available for the incentivization itself [2].

Thus, in [2], the authors proposed a first decentralized approach, computing the routes on-board, in a *edge computing* fashion [28], without the need for any central



coordination among vehicles. In that work, which did not take into account incentivization and budget constraints, the authors showed that the introduction of a probabilistic component in the route calculation, and the selection of slightly sub-optimal routes in place of the optimal ones, could help achieve a significantly more uniform sensing distribution at the cost of slightly increasing the overall distance travelled by vehicles.

In this paper, an evolution of the prototype by Asprone et al. [2] is presented, by introducing the crucial concepts of incentivization and budgetary constraints. In particular, the proposed algorithm, rather than computing the most efficient path between a given origin and destination, returns a slightly sub-optimal route, randomly selected among all the routes whose incentivization cost does not exceed a given, per-ride allocated budget. This way, when computing multiple routes between the same given origin and destination, potentially different routes are produced each time, resulting in an overall more uniform road-network coverage.

More in detail, in our vision, drivers of high-mileage vehicles that are willing to participate in the crowd-sensing activity can negotiate a per-kilometer reimbursement rate with the entity collecting data (e.g., a Smart City, or the TNC itself), and accept to use the ROUTr algorithm, for example with a custom routing app. Then, every time the need arises, the driver uses ROUTr to compute a possibly sub-optimal route between its current position and the destination, and receives a monetary reimbursement based on the additional distance travelled because of the sub-optimal route, covering for its expenses and efforts. Upon verifying that the driver actually followed the route produced by the algorithm, the data-collecting entity reimburses the driver. Since ROUTr allows for the definition of a maximum per-ride reimbursement amount, the data-collecting entity can control the total budget allocated for reimbursements.

## B. PRELIMINARY DEFINITIONS

A road network can intuitively be represented as a directed graph in which each edge can be mapped to a road segment. In such a representation, each road segment is typically characterized by some non-negative associated cost, e.g., length or travel time. Formally, we introduce the graph representation of a road network as follows.

**Definition 1 (Road network graph):** A road network graph is an ordered tuple  $M = \langle N, S, c \rangle$ , where  $N$  is a set of nodes,  $S$  is a set of directed road segments defined as  $S \subseteq \{(x, y) \mid x, y \in N \text{ and } x \neq y\}$ , and  $c : S \rightarrow \mathbb{R}^{\geq 0}$  is a function associating to each road segment a non-negative cost.

A route from an origin node  $o$  to a destination node  $d$  in  $M$  is a sequence of road segments joining a sequence of nodes starting at  $o$  and ending in  $d$ . More in detail, we define a route as follows.

**Definition 2 (Route):** Given a road network graph  $M = \langle N, S, c \rangle$ , an origin node  $o$  and a destination node  $d$ , with

$o, d \in N$ , a sequence of segments  $\rho = (s_1, s_2, \dots, s_{k-1})$  is a route from  $o$  to  $d$  in  $M$  if there exists a sequence of vertices  $(v_1, v_2, \dots, v_k)$  such that  $s_i = (v_i, v_{i+1})$ , for all  $i \in [1, k-1]$ , and  $v_1 = o$  and  $v_k = d$ .

**Definition 3 (Cost of a route):** We define the cost associated with a route as the sum of the costs of its segments. Formally, the cost of a route  $\rho = (s_1, \dots, s_k)$  in a road network graph  $M = \langle N, S, c \rangle$  is defined as  $cost(\rho) = \sum_{i=1}^k c(s_i)$ .

A route  $\rho$  from  $o$  to  $d$  is said to be *optimal* if there exists no other route  $\pi$  from  $o$  to  $d$  such that  $cost(\pi) < cost(\rho)$ . For the sake of clarity, in the following we will consider the length in kilometers of the route as its cost. Nevertheless, all the considerations in the remainder of the paper can be also applied to other types of costs, such as for example the travel time.

On-board and hand-held navigation systems help drivers reach their destination efficiently by using specialized algorithms, such as A\* [19] or the well-known Dijkstra's one [14], to compute an *optimal* route, i.e., a route minimizing the considered associated cost. In VCS scenarios involving ride-sharing fleets, however, if all the drivers always follow the optimal route for their rides, the main city thoroughfares, usually included in the most efficient routes, would be sensed very frequently, whereas minor, adjacent streets would be rarely (if at all) visited. Thus, some incentivization mechanism should be put in place to reward a ridesharing vehicle driver to accept a sub-optimal route, for the sake of increasing the sensing coverage. The incentivization strategy we considered consists in rewarding drivers based on the additional cost (e.g., travelled distance or time) the sub-optimal routes require, paying a given reimbursement rate per additional cost unit. More formally, the incentivization cost associated with a route  $\rho$  going from  $o$  to  $d$  is defined as follows.

**Definition 4 (Incentivization cost):** Let  $M = \langle N, S, c \rangle$  be a road network graph and let  $\rho$  be a route in  $M$  going from the origin node  $o$  to the destination node  $d$ . The incentivization cost of  $\rho$  is defined as  $IncCost(\rho) = (cost(\rho) - cost(opt)) \cdot r$ , where  $opt$  is the optimal route from  $o$  to  $d$  and  $r$  is a given reimbursement rate.

Note that, from Definition 4, it follows that a route is optimal if and only if its incentivization cost is zero. In real-world applications, systems designed to improve VCS coverage by re-routing vehicles are typically subject to incentivization budget constraints [10], [43], [44]. Indeed, for any VCS-based system to be profitable, it is crucial for a decision maker to make sure that the incentivization budget does not exceed the value of the collected data. In our decentralized scenario, we assume that a per-ride incentivization budget is given. Moreover, we formalize the concept of admissible (i.e., incentivization budget compatible) routes by leveraging the notion of Potential Path Area (PPA), which refers to the spatial extent of where individuals can participate in activities subject to time and other (e.g., travelled distance, or incentivization) constraints [31]. In particular, we formalize

**Algorithm 1: ROUTr**


---

```

1 ROUTr (Origin, Destination, PerRideIncentBudget, ReimburseRate) :
2   optimalRoute  $\leftarrow$  findOptimalRoute(Origin, Destination);
3   if (optimalRoute = nil) then
4     return nil ;
5   MaxRouteCostIncr  $\leftarrow$  PerRideIncentBudget / ReimburseRate ;
6   MaxRouteCost  $\leftarrow$  cost(optimalRoute) + MaxRouteCostIncr ;
7   return
   RandomizedPpaRouting (optimalRoute, maxRouteCost) ;

```

---

the idea of Incentivization Potential Path Area (iPPA) as follows.

*Definition 5 [Incentivization Potential Path Area (iPPA)]:* Given an origin  $o$  and a destination  $d$  in  $M$ , and a non-negative per-ride incentivization budget  $B \in \mathbb{R}^{\geq 0}$ , the Incentivization Potential Path Area (iPPA) is the set of all routes  $\rho$  from  $o$  to  $d$  in  $M$  whose incentivization cost is not greater than  $B$ , i.e.,  $IncCost(\rho) \leq B$ .

Notice that the iPPA from  $o$  to  $d$  induced by a per-ride budget  $B$  is empty when  $o$  and  $d$  are disconnected, i.e., when there is no route going from  $o$  to  $d$ . Otherwise, if there exists at least one route from  $o$  to  $d$ , the iPPA is guaranteed to contain at least the optimal route.

**C. THE ROUTr ALGORITHM**

The algorithm we propose is detailed in Algorithm 1 and described as follows.

ROUTr takes as inputs an *Origin* and a *Destination* node in the current road-network  $M$ , which we assume to be globally defined for the sake of simplicity, as well as a per-ride incentivization budget and a reimbursement rate, and returns a randomly-generated admissible route from *Origin* to *Destination* belonging to the iPPA induced by the given incentivization budget and reimbursement rate. Notice that both the per-ride budget and the reimbursement rate are parameters for the ROUTr algorithm, meaning that each vehicle/driver willing to participate in the crowd-sensing could negotiate their own budget and rates with the entity collecting data (e.g., a Smart City).

Firstly (see Line 2), ROUTr computes (by means of any optimal routing algorithm, such as Dijkstra's one [14]), the optimal route between *Origin* and *Destination* in  $M$ . If *Origin* and *Destination* are disconnected, the algorithm returns *nil* (see Lines 3-4). Otherwise, ROUTr computes, based on the given reimbursement rate, the maximum admissible route cost compatible with the per-ride incentivization budget (see Lines 5-6).

Lastly, the actual route to be returned is generated by the helper function `RandomizedPpaRouting`, that we devised as an ad-hoc variant of a semi-admissible version [32] of the well-known A\* algorithm [19]. This helper function, which is detailed in Algorithm 2, takes as input the *optimalRoute* between the current *Origin* and *Destination* points and the maximum admissible route cost compatible with the per-ride incentivization budget, and returns a route

**Algorithm 2: randomizedPpaRouting**


---

```

1 RandomizedPpaRouting (optimalRoute, MaxRouteCost) :
2    $\triangleright$  Initializations
3   toleranceThreshold  $\leftarrow$  MaxRouteCost / cost(optimalRoute);
4   Origin  $\leftarrow$  optimalRoute.Origin;
5   Destination  $\leftarrow$  optimalRoute.Destination;
6   forall ( $v \in$  Map.Nodes) do
7      $v$ .Predecessor  $\leftarrow$  nil;
8      $v$ .RouteCost  $\leftarrow$   $+\infty$ ;
9   Origin.Predecessor  $\leftarrow$  Origin;
10  Origin.RouteCost  $\leftarrow$  0;
11  OpenList  $\leftarrow$  {Origin};
12   $\triangleright$  Main loop
13  while (OpenList  $\neq$   $\emptyset$ ) do
14     $m \leftarrow$  estimateMinRouteCostToDestination(OpenList);
15    nextNodeToVisit  $\leftarrow$  selectRandomly( $\{n \in$  OpenList |
16      ( $n$ .RouteCost +  $h(n)$ )  $\leq$   $m \cdot$  toleranceThreshold});
17    expandNode(nextNodeToVisit);
18    if (Destination.RouteCost  $\leq$  maxRouteCost) then
19      return buildPath(Destination);
20  expandNode (node) :
21  forall ( $s \in$  node.OutgoingSegments) do
22    tentativeRouteCost  $\leftarrow$  node.RouteCost +  $s$ .RouteCost;
23    if (tentativeCost <  $s$ .Target.RouteCost) then
24       $s$ .Target.RouteCost  $\leftarrow$  tentativeRouteCost;
25       $s$ .Target.Predecessor  $\leftarrow$  node;
26      if tentativeRouteCost  $\leq$  maxRouteCost then
27        OpenList.addIfNotExists( $s$ .Target);
28  OpenList.remove(node);

```

---

between *Origin* and *Destination*, randomly-selected among those that have a cost not exceeding that upper threshold, i.e., that belong to the iPPA induced by the per-ride incentivization budget and reimbursement rate.

`RandomizedPpaRouting` firstly determines a tolerance threshold (see Line 2), computed as the ratio between the maximum admissible route cost in the iPPA, and the cost of the optimal path. Such a tolerance threshold will necessarily be greater than or equal to 1. Then, in lines 3 to 10, the necessary data structures are initialized. In particular, the procedure maintains, for each node in the road network graph, information about its predecessor in the graph exploration, and a tentative route cost for reaching that node from the *Origin*. Initially, all the nodes in *Map* except *Origin* are assigned no predecessor and infinite cost, as no route from *Origin* reaching them has yet been found. As for *Origin*, its predecessor is set by convention to the node itself, and the cost to reach itself is set to zero. `RandomizedPpaRouting` also maintains a list (*OpenList*) of nodes to be explored. Initially, this list contains only the *Origin* node, as it will be the starting point of the road-network exploration.

The main loop of `RandomizedPpaRouting`, in lines 11 to 16, iterates until the *OpenList* is empty. At each iteration, the procedure selects the next node to visit among the candidates that are currently in the *OpenList*. To do so, a first step consists in computing the minimum estimated

route cost to reach *Destination* passing from one of the nodes in the *OpenList*. Such an estimated minimum cost is computed by the *estimateMinRouteCostToDestination* procedure, which takes as input the current *OpenList* and proceeds as follows. For each node  $n$  in the *OpenList*, the procedure computes an estimate cost for a path from *Origin* to *Destination* passing through  $n$  by adding the minimum known cost for reaching node  $n$ , which is stored in  $n.RouteCost$ , with an estimated distance from  $n$  to *Destination* computed by a monotone heuristic distance-estimation function  $h$ . In our experiments, we used the widely-adopted *Great Circle Distance* [35] as heuristic function  $h$ , which produces accurate distance estimation between two points on the earth surface [17, Ch. 1]. More formally, the *estimateMinRouteCostToDestination* procedure returns a cost  $m$  defined as

$$m \triangleq \min_{n \in OpenList} (n.RouteCost + h(n)).$$

Subsequently, the next node to visit (see Line 13) is selected by randomly choosing a candidate in the *OpenList* among those whose minimum estimated route cost to destination does not exceed  $m \cdot toleranceThreshold$ . Selecting the next node to visit in this way guarantees that, as proved by Pearl and Kim [32], the cost of the final route will not exceed that of the optimal one multiplied by the tolerance threshold, and so that *RandomizedPpaRouting* returns a route in the considered iPPA, satisfying the budget constraint.

After selecting the next node to visit, the algorithm proceeds as other standard routing algorithms by *expanding* the selected node, i.e., by removing the node from the *OpenList* and by updating the predecessor and tentative route cost information of the adjacent nodes, possibly discovering new nodes that will be added to the *OpenList* to be explored in the next iterations. This is done by the *expandNode* procedure, detailed in lines 17 to 25. The procedure to expand a given *node* consists in iterating over all of its outgoing segments. For each of these segments  $s$ , a tentative cost to reach the target node  $s.Target$  passing through  $s$  is computed as the sum of the cost of reaching *node* from the *Origin* and the cost of traversing  $s$  (see Line 19). If such a tentative cost is smaller than the best currently known one, stored in  $s.Target.RouteCost$ , then the best known cost for  $s.Target$  is updated accordingly, and the predecessor for  $s.Target$  is set to *node*, since the best known path to  $s.Target$  from *Origin* passes through *node*. Moreover, if the tentative cost does not exceed the maximum admissible cost,  $s.Target$  is added to the *OpenList*. After the expansion of the current node, *RandomizedPpaRouting* checks whether an admissible path to *Destination* has been found (see Line 15). If that is the case, i.e., if *Destination.RouteCost* is set to an admissible value smaller than the maximum admissible cost, the procedure returns the current path from *Origin* to *Destination* obtained by starting at the latter and navigating the predecessors until *Origin* is reached. Otherwise, the procedure continues with the next iteration. Notice that the stopping

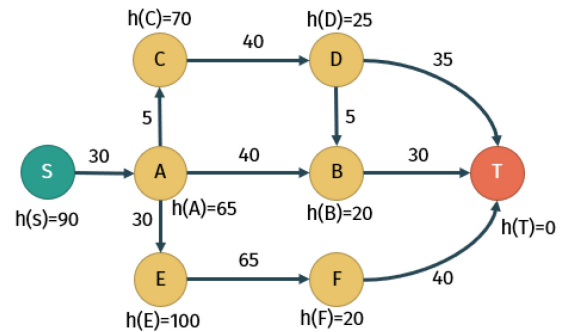


FIGURE 1. The simple road-network graph *Map*, annotated with a heuristic function  $h$ , estimating the route cost from any node to node T.

condition described above is guaranteed to be eventually satisfied, since, for the *RandomizedPpaRouting* procedure to be invoked, there exists at least one route (the optimal one) from *Origin* to *Destination*.

#### D. EXAMPLES

To better explain how ROUTr operates, in this section its application on a small example road-network graph is described in a step-by-step fashion. Then, to contextualize the intuitions behind ROUTr on a *real*, complex, urban road-network, an additional example based on the City of San Francisco is provided.

Consider the road-network graph *Map* depicted in Figure 1. In the figure, each node is decorated with its name and a consistent distance-estimation heuristic  $h$ , representing an estimation of the cost to reach T from said node. Each edge, representing a road segment connecting two nodes, is decorated with its corresponding traversal cost. Suppose to run ROUTr to find a path from node S to T, with an incentivization budget of 20 and a unitary reimbursement rate. This means that a driver receives a reimbursement of 1 for each additional unit of cost in the sub-optimal route w.r.t. the optimal one, and that all routes with an incentivization cost not exceeding 20 are admissible.

Notice that the step-by-step example described hereafter is also available as a short video animation at the doi: <https://doi.org/10.5281/zenodo.5171686>. Firstly, (see Line 2 in Algorithm 1), ROUTr computes the optimal route from S to T in *Map*. Such a route is  $\pi = S \rightsquigarrow A \rightsquigarrow B \rightsquigarrow T$ , and it is easy to see that  $cost(\pi) = 30 + 40 + 30 = 100$ . Then, the maximum admissible route cost increment with the given incentivization budget of 20 is computed (see Line 5) as  $20/1 = 20$ , inducing a maximum admissible route cost (see Line 6) of 120, and the helper function *RandomizedPpaRouting* is called (Line 8). After the initialization (lines 2 to 10 in Algorithm 2), *toleranceThreshold* is computed as  $120/100 = 1.2$ .

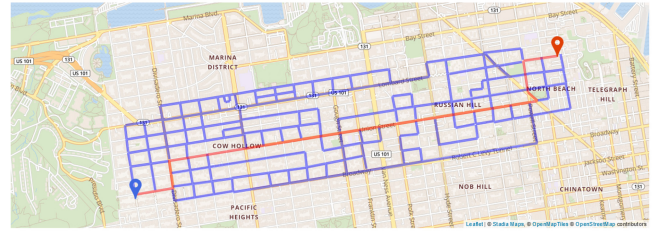
At the beginning of the first iteration of the main loop (lines 11 to 16 in Algorithm 2, *OpenList* contains only S. The minimum estimated cost to destination for S is computed as  $m = S.RouteCost + h(S) = 0 + 90 = 90$ . S is trivially a candidate node for expansion, since  $S.RouteCost + h(S) \leq m \cdot$

$toleranceThreshold = 90 \cdot 1.2 = 108$ . Hence, S is selected for expansion and the `expandNode` procedure is called. During the expansion of S, a tentative cost to reach node A from node S is computed as  $S.RouteCost + cost(S \rightarrow A) = 30$ . Since such a tentative cost is smaller than the currently known cost to reach A (which, after the initialization, is  $\infty$ ),  $A.RouteCost$  is updated and set to 30, and  $A.Predecessor$  is set to S. S is then removed from the *OpenList* and, since  $A.RouteCost$  is smaller than the maximum admissible route cost, A is added to the *OpenList*.

At the beginning of the second iteration of the main loop, *OpenList* contains only A, which is selected for expansion. During its expansion, three adjacent nodes B, C, and E need to be analyzed. The tentative cost to reach node B from node S is computed as  $A.RouteCost + cost(A \rightarrow B) = 30 + 40 = 70$ . Since such a tentative cost is smaller than the currently known cost to reach B (which is  $\infty$ ),  $B.RouteCost$  is set to 70, and  $B.Predecessor$  is set to A. Since  $B.RouteCost$  is smaller than the maximum admissible cost, B is added to *OpenList*. Continuing with the expansion of A, the next node to analyze is C. The tentative cost to reach C from node S is computed as  $A.RouteCost + cost(A \rightarrow C) = 30 + 5 = 35$ . Similarly to node B,  $C.RouteCost$  is set to 35, and  $C.Predecessor$  is set to A, and C is added to *OpenList* as well. The last adjacent node to analyze is E, whose tentative cost is computed as  $A.RouteCost + cost(A \rightarrow E) = 30 + 30 = 60$ . As with B and C,  $E.RouteCost$  and  $E.Predecessor$  are updated accordingly, and E is added to the *OpenList*. A is then removed from the *OpenList*, completing its expansion.

At the beginning of the third iteration, *OpenList* contains nodes B, C, and E. To select the next node to expand, the minimum estimated cost to destination is computed for the nodes in the *OpenList*. In particular:  $B.RouteCost + h(B) = 70 + 20 = 90$ ;  $C.RouteCost + h(C) = 35 + 70 = 105$ ;  $E.RouteCost + h(E) = 60 + 100 = 160$ . Hence, the minimum of the estimated costs to destination for the nodes in *OpenList* is  $m = 90$ . Of the three nodes in *OpenList*, only B and C are candidate to be randomly selected for expansion, as  $E.RouteCost + h(E) = 160$  which is greater than the tolerance threshold, which is 108. In this example, C is selected for expansion in the third iteration. During its expansion,  $D.Cost$  is set to  $C.RouteCost + cost(C \rightarrow D) = 35 + 40 = 75$ ,  $D.Predecessor$  is set to C, D is added to the *OpenList* and C is removed from it.

At the beginning of the fourth iteration, *OpenList* contains B, E, and D. As during the previous iteration, the minimum estimated cost to destination is computed for the nodes in *OpenList*. In particular:  $B.RouteCost + h(B) = 70 + 20 = 90$ ;  $E.RouteCost + h(E) = 60 + 100 = 160$ ;  $D.RouteCost + h(D) = 75 + 25 = 100$ . Again, the minimum of the estimated costs to destination is  $m = 90$ , and only B and D are candidates for selection. In this example, node D is randomly selected for expansion in the fourth iteration. To complete the expansion of D, two adjacent nodes B and T need to be analyzed. The tentative cost for



**FIGURE 2.** Representation of the shortest route (trajectory in red) between origin (blue marker) and destination (red marker), and cumulative representation (trajectories in blue) of 64 routes from the iPPA induced by a \$ 0.05 incentivization budget and \$ 0.01 reimbursement rate.

reaching node B from S passing through node D is computed as  $D.RouteCost + cost(D \rightarrow B) = 75 + 5 = 80$ . Since such a tentative cost is greater than the best currently known cost to reach B (which is 70),  $B.RouteCost$  and  $B.Predecessor$  remain unchanged. As for T, its tentative cost is computed as  $D.RouteCost + cost(D \rightarrow T) = 75 + 35 = 110$ , that is smaller than the best currently known cost to reach T ( $\infty$ ). Thus  $T.RouteCost$  is updated and set to 110, and  $T.Predecessor$  is set to D. Since  $T.RouteCost$  is smaller than the maximum admissible cost, T is added to *OpenList*. Lastly, D is removed from the *OpenList*, and its expansion is completed.

After the expansion of node D, the stopping condition (see Line 15 in Algorithm 2) is satisfied, as a path from S to T with an admissible cost has been found ( $T.RouteCost = 110$  which is smaller than 120, the maximum admissible cost). At this point, the `buildPath` procedure can build the path to return by recursively navigating the stored predecessors from T to S, resulting in the path  $S \rightsquigarrow A \rightsquigarrow C \rightsquigarrow D \rightsquigarrow T$ , whose incentivization cost is 10. In this example, ROUTr computed a route which is slightly longer than the shortest one, using part of the allowed incentivization budget. Notice that, had node B been selected for expansion in the third or fourth iteration, the algorithm would have returned the shortest path.

To better highlight the potential of ROUTr on a *real* road-network, in which the iPPA induced by a given budget is likely to contain many routes, a preliminary analysis on the road network of the City of San Francisco was conducted. In this analysis, a per-ride incentivization budget of just \$ 0.05 was selected, along with a \$ 0.1 per km reimbursement rate, thus allowing detours up to 0.5 km. The results are shown in Figure 2. After selecting a pair of origin and destination points on the San Francisco map, the shortest route between them (in red) was computed, as well as 64 routes (in blue) belonging to the iPPA induced by the considered budget and incentivization rates. Intuitively, if many vehicles runs from the given origin and destination were routed using a shortest-path algorithm, all the trajectories would cover the same road segments (in red). On the other hand, if the same vehicles would have been routed using ROUTr, each run would randomly compute a route among those in the iPPA (trajectories in blue), resulting in a higher road-network coverage.



#### IV. EMPIRICAL EVALUATION

The goal of the empirical evaluation is to understand whether the use of ROUTR can lead to a more uniform urban road network coverage w.r.t. a shortest path algorithm. To this end, we aim at quantifying how many additional road segments are traversed by the involved vehicle fleet using ROUTR, as well as the consequent implications on the visit frequencies for the entire urban map. More precisely, we experimentally assessed the spatio-temporal road-network coverage achievable by a swarm of vehicles if they were routed by ROUTR, under different incentivization budgets, with respect to the coverage achievable by the same vehicles if their routes were computed by a shortest path algorithm. To this end, we employed a massive dataset of real-world taxi trajectories, recorded over a three-weeks period in the San Francisco Bay Area. For each of these taxi runs between an Origin  $O$  and a Destination  $D$ , we computed the shortest path from  $O$  to  $D$  using the  $A^*$  shortest-path algorithm. We then computed a route from the same  $O$  and  $D$  using ROUTR, with different incentivization budgets. On top of these routes, we computed some standard road network coverage metrics, widely used in similar works.

In what follows, the experimental protocol is presented by describing in detail the employed dataset and data preparation steps, the considered reimbursement rate and incentivization budgets, and the employed road-network coverage metrics.

##### A. THE DATASET

The empirical evaluation is based on a publicly-available dataset of real taxi trajectories collected within the *Cabspotting* project [34]. This dataset consists in 11,219,955 timestamped GPS coordinates, collected in the San Francisco Bay Area from more than 500 vehicles of the *Yellow Cab* company, over 25 days, from 2008/05/17 to 2008/06/10.

As for the logical representation of the road network on which the routes and the achieved spatio-temporal coverage are computed, as done in other similar works (e.g., [2]), we leveraged open data from the *OpenStreetMap* (OSM) project, whose quality is generally considered to be comparable to the one of authoritative datasets in urbanized areas [18], [26].

The taxi dataset required some pre-processing tasks before being in a state useful for the experiments. More in details, the original dataset contains, for each taxi, a single sequence of timestamped GPS positions spanning over 25-days, enriched with information on the vehicle occupation status (i.e., whether there are passengers on board or not). A first data preparation step for the empirical evaluation consisted in splitting this single stream of data into a set of independent trajectories, where each of them represents an independent taxi movement from a given origin to a destination.

Firstly, we split each taxi's data stream whenever there was a change in the occupancy status. It is worth noting that, differently from similar works (e.g., [8]), we also considered taxi trajectories in which the vehicle is not occupied, since

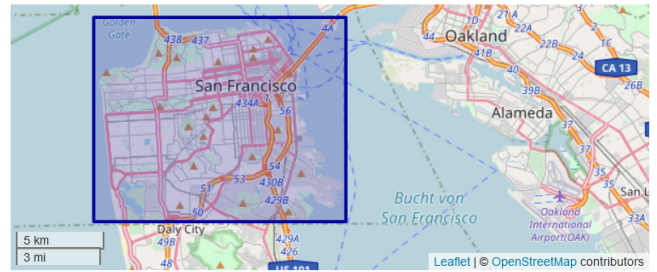


FIGURE 3. The considered urban area of San Francisco.

it can act as a probe in these cases as well. Furthermore, we also split the sequence every time there was a time gap between subsequent GPS points greater than 3 minutes, assuming that the taxi was not operating in that time frame [5].

For each of the resulting trajectories, the GPS coordinates for the origin and destination points, and the initial timestamp were extracted. These pairs of points were then map-matched to the respective OSM maps, by selecting the closest routable, non-highway, OSM road segment within a 30 meters radius, under the assumption that taxi cannot start a run on a highway. All the trajectories for which this map-matching step failed were discarded. Similarly, all the trajectories for which there was no vehicle-routable path between the map-matched source and destination in the considered OSM map were removed. Furthermore, since most of the potential use cases employing VCS involve urban environments, we restricted our analysis to the urban area of San Francisco, whose boundaries, according to the Nominatim OSM service,<sup>1</sup> are shown in Figure 3.

Lastly, to avoid the introduction of biases due to weekly fluctuations in traffic dynamics, we temporally restricted our analysis by considering trajectories recorded over a three weeks period. In particular, the three contiguous weeks from 2008/05/18 to 2008/06/07 were considered. After these filtering steps, we retained 405 599 trajectories from 534 taxis, accounting for a total of about 1.1 million kilometers over the three weeks.

##### B. REIMBURSEMENT RATES AND INCENTIVIZATION BUDGETS

An accepted rule of the thumb for incentive design is to ensure that the value of the incentive is not less than the cost that the driver has to face to fulfill a sensing assignment [44]. Thus, the vehicle operating costs per kilometer, including fuel costs, maintenance, repair and tires consumption, can be considered as a lower bound reimbursement rate. In San Francisco, most of the taxis are hybrid vehicles [37], [45]. As indicated by the American Automobile Association [3], the operating cost for this kind of vehicles in 2019 amounts to \$ 0.082 per kilometer, as detailed in Table 1. Hence, in our experiments, we considered this operating cost as a reimbursement rate for all the taxis.

1. <https://nominatim.openstreetmap.org/>

**TABLE 1.** Considered taxi operating costs in San Francisco.

Operating cost type	Cost (\$/km)
Fuel	\$ 0.034
Maintenance, wear and tear, and other expenses	\$ 0.048
<b>Total</b>	<b>\$ 0.082</b>

As for the incentivization budgets, we considered eight values, which we believe can adequately represent different VCS scenarios, given the value of the collected data. In particular, we considered the values \$ 500, \$ 1000, \$ 1500, \$ 2000, \$ 2500, \$ 5000, \$ 7500, and \$ 10 000, intended as the overall incentivization budget that a Smart City would allocate for the whole fleet of 534 taxis, during the three-weeks timespan.

### C. PERFORMANCE METRICS

To assess the effectiveness of our proposal, we computed, for both the shortest path algorithm and ROUTr, and for each considered incentivization budget, the following metrics:

- 1) The number of road segments visited at least once by a vehicle in considered three weeks period. This metric is a key spatial-coverage indicator for decision makers;
- 2) The total travelled distance (in kilometers) for the entire fleet of vehicles. This metric is an indicator of the efficiency of a coverage-improving solution: techniques that achieve improvements in spatial coverage with limited increases in travelled distance are more cost-effective. Moreover, this metric also gives insights on the additional vehicular traffic generated by a vehicular crowd-sensing solution;
- 3) The average timegap between subsequent visits in a road-segment. This metric (also used in other works, such as [6], [25]) is a key temporal coverage quality indicator, showing how frequently a road-segment is sensed by a vehicle. In particular, if a road segments is visited by  $n$  vehicles at times  $t_1, \dots, t_n$ , with  $t_i \leq t_{i+1}$  for all  $i \in [1, \dots, n - 1]$ , the average timegap for that segment is defined as

$$\frac{\sum_{i=2}^n (t_i - t_{i-1})}{n - 1}.$$

To gain additional insight on the coverage dynamics at road segment granularity level, these metrics were computed both for the entire considered map, and for each of the main road types defined in the OSM standard (see Table 2). Let us note that OSM defines also additional types of roads [30], but they were excluded from the analysis, as they are either not routable by public vehicles, or their presence in the considered part of the map is negligible. Table 2 reports, for the considered road segment types, also the corresponding number in the employed OSM map of San Francisco.

Finally, to account for statistical fluctuations due to the randomness in the proposed solution, the experiments were repeated 10 times for each configuration. In Section V we report the average and the standard deviation of these results.

## V. RESULTS AND DISCUSSION

In Table 3, the spatial coverage results, computed over the entire road-network of San Francisco, for both the shortest-path algorithm and ROUTr, with the considered eight incentivization budgets, are reported. In particular, for each algorithm/budget configuration, the corresponding per-ride incentivization budget is reported, as well as the number of distinct segments visited at least once during the three-weeks period, and the cumulative distance in kilometers travelled by the taxis during the considered timespan.

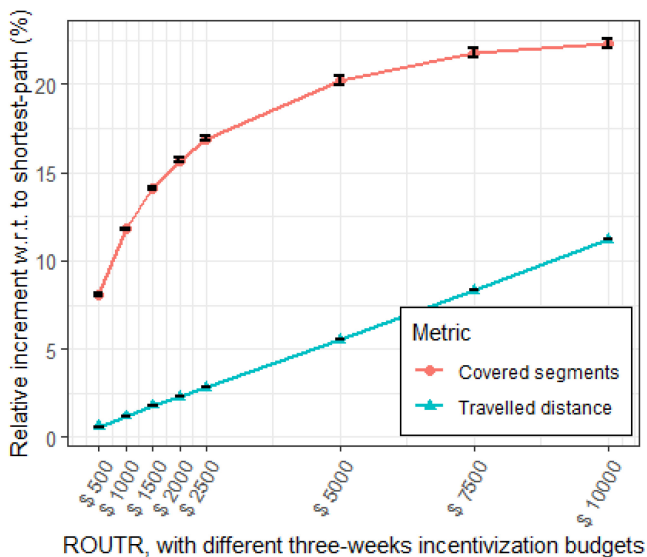
These results show that, even with an incentivization budget for the entire fleet as low as \$ 500 (i.e., less than \$1 per driver over the three-weeks), ROUTr can achieve a significant 8% improvement in the number of covered segments, with an overall increase in the travelled distance of just 0.6%. Notice that, even though \$ 1 per driver for three weeks may seem unrealistic, in this configuration each driver drove on average less than 20 additional meters for each trajectory. Increasing the budget up to \$ 10 000 leads to a 22% rise in the number of distinct covered segments, managing to visit more than 86% of the road segments of the considered urban map at least once over the three-weeks timespan. This improvement in coverage comes at the cost of an increment of 11% in the travelled distance. These trends are graphically reported in Figure 4, which shows the changes in distinct covered segments (red line) and travelled distance (blue line) w.r.t. the considered ROUTr incentivization budget, as well as error bars indicating the variability of the results over the 10 repetitions of the experiments. From the figure we can notice that, while the overall travelled distance increases more or less linearly w.r.t. the budget, the same does not hold for the number of distinct covered segments. Indeed, the latter increases way more rapidly in the initial part of the plot, going from 8% to 20% when increasing the budget from \$ 500 to \$ 5000, and more slowly thereafter, going from 20% to 22% when increasing the budget from \$ 5000 to \$ 10 000, with an almost flat trend at higher budgets. This evidence suggests that, when using decentralized, random-based routing approaches to improve road-network coverage, increasing the budget above certain levels might not be cost-effective w.r.t. the achieved spatial coverage. In particular, in the considered scenario of about 500 taxis in San Francisco, the best trade-off between spatial coverage improvement and costs seems to fall between \$ 2500 and \$ 5000. As for the fluctuations due to the randomness in ROUTr, the results show that there is very little variability among the different repetitions of the experiments, especially for the overall travelled distance, for which the standard deviation was always smaller than 0.1%. The number of covered segments exhibited a slightly higher variability, especially

**TABLE 2.** Considered types of roads, and number of respective road segments in the considered part of the San Francisco OSM map.

Road type	Description	Count
Motorway	restricted access major highways, typically with multiple lanes.	1718
Motorway link	roads leading to and from a motorway.	1807
Primary	the most important arterial roads, usually linking different big cities.	6488
Secondary	smaller roads which do not qualify as primary.	8022
Tertiary	roads connecting local centres within larger cities.	14 021
Residential	roads which serve as an access to housing, often lined with houses.	55 741
Service	access roads to, or within, industrial estates, alleys, etc.	32 411
Unclassified	roads not falling in any other category, or not yet classified.	3463
<b>Total</b>		<b>123 671</b>

**TABLE 3.** Spatial coverage results for both A\* and ROUTr, with the considered three-weeks incentivization budgets.

Three-weeks budget	A*		ROUTr						
	\$0	\$500	\$1000	\$1500	\$2000	\$2500	\$5000	\$7500	\$10 000
Per-ride budget	\$0	\$0.001	\$0.003	\$0.004	\$0.005	\$0.006	\$0.012	\$0.018	\$0.025
Distinct covered segments (count)	87 335	94 424	97 662	99 684	101 060	102 111	104 969	106 405	106 813
Distinct covered segments (% of total)	70.6	76.4	79.0	80.6	81.7	82.6	84.9	86.0	86.4
Increase in covered segments w.r.t. A* (%)	n/a	8.1	11.8	14.1	15.7	16.9	20.2	21.8	22.3
Increase in travelled distance w.r.t. A* (%)	n/a	0.6	1.2	1.8	2.3	2.8	5.5	8.3	11.2



**FIGURE 4.** Relative increments in the number of covered segments and overall travelled distance achieved by ROUTr with the considered incentivization budgets w.r.t. A\*.

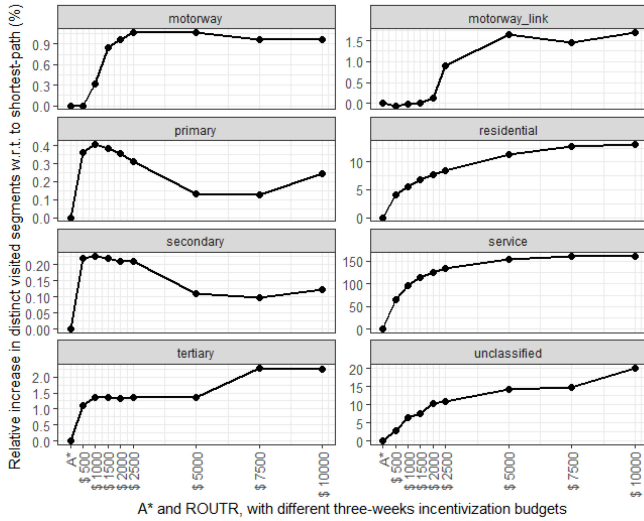
with greater budgets. Still, the standard deviation remained always smaller than 0.3%.

Table 4 presents the details of the spatial coverage results by road segment type. In particular, for each of the considered OSM road classes and for each algorithm/incentivization budget combination, the table reports

the percentage of segments of the given type that were visited at least once during the considered three-weeks timespan. These results show that, in the scenario where vehicles are routed by the A\* algorithm, the swarm of taxis can cover the 97-99% of the segments belonging to the main types, namely motorways, primary, secondary, and tertiary segments, during the three weeks. On the other hand, minor road segments such as residential, unclassified and service ones, are not as thoroughly sensed, achieving only 78.4%, 64.3% and 23.8% coverage. This is the type of unevenness that can compromise many VCS use cases requiring a uniform road-network coverage. In the scenario where the ROUTr algorithm is used, a more uniform road-network coverage is obtained, as highlighted also by Figure 5. Indeed, without sacrificing the coverage rates for the main road segments, ROUTr significantly improves the coverage rates for minor segments. With an incentivization budget of just \$ 500, ROUTr achieves a 4% coverage rate improvement for residential segments, a 3% improvement for unclassified ones, and a 65% improvement for service ones. This means ~1700 new residential road segments, ~65 new unclassified segments, and ~5000 new service segments would have been visited, if the taxis routes were computed by our proposal. When increasing the incentivization budget up to \$ 10 000, ROUTr achieves a 13% coverage rate improvement for residential segments, a 20% improvement for unclassified ones, and a 160% improvement for service ones. These improvements

**TABLE 4.** Percentage of covered segments by type, for both A\* and ROUTr, with the considered three-weeks incentivization budgets.

Budget	A*		ROUTr						
	\$0	\$500	\$1000	\$1500	\$2000	\$2500	\$5000	\$7500	\$10 000
Motorway	98.5%	98.5%	98.8%	99.3%	99.4%	99.5%	99.5%	99.4%	99.4%
Motorway link	93.6%	93.5%	93.6%	93.6%	93.7%	94.4%	95.1%	94.9%	95.2%
Primary	99.2%	99.6%	99.6%	99.6%	99.6%	99.5%	99.4%	99.4%	99.5%
Secondary	99.6%	99.8%	99.8%	99.8%	99.8%	99.8%	99.7%	99.7%	99.7%
Tertiary	96.7%	97.8%	98.0%	98.0%	98.0%	98.0%	98.0%	98.9%	98.9%
Residential	78.4%	81.6%	82.8%	83.8%	84.5%	85.1%	87.3%	88.5%	88.7%
Service	23.8%	39.3%	46.8%	51.1%	53.9%	55.9%	60.3%	62.1%	62.1%
Unclassified	64.3%	66.2%	68.4%	69.1%	71.0%	71.3%	73.5%	73.8%	77.3%



**FIGURE 5.** Coverage increments by segment type, w.r.t. A\*, achieved by ROUTr with the considered incentivization budgets.

correspond to ~5700 new residential segments, ~450 new unclassified segments, and ~12 400 new service segments being visited during the considered three-weeks timespan. As for the fluctuations of these coverage results among the 10 repetitions of the experiment, also in this case there was very little variability (less than 0.1% standard deviation), so we omitted the error bars in Figure 5 for the sake of clarity.

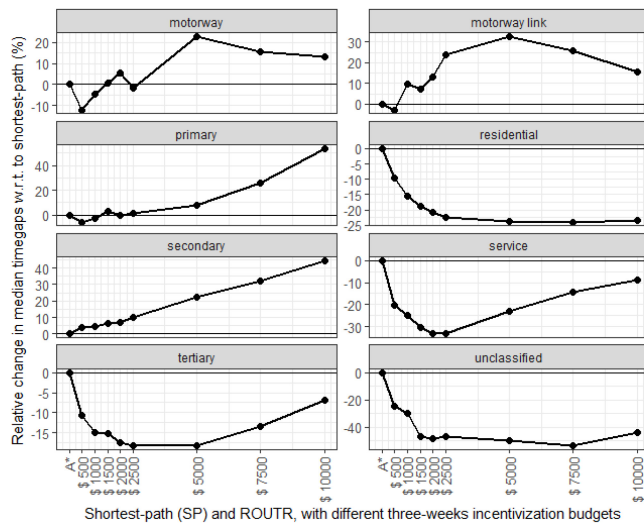
As for the temporal coverage results, Table 5 reports, for each considered road segment type, the median timegap (in hours) between subsequent visits of a probe vehicle during the three-weeks timespan. These data show that, as for the spatial coverage, in the scenario in which all the vehicles are routed by A\*, there are significant differences in the visit frequency between main and minor road segment types. Indeed, segments belonging to motorway, primary, secondary and tertiary classes are visited way more frequently (with median timegaps ranging from less than an hour for primary segments, to six hours for tertiary ones). On the other hand, minor road segment such as residential, unclassified, and service ones are visited more rarely, with median timegaps ranging from about one day for residential segments, to about two and a half days for service ones.

**TABLE 5.** Median timegaps between subsequent visits in the three-weeks period per segment type (in hours), for both the A\* and ROUTr, with the considered incentivization budgets.

Budget	A*		ROUTr						
	\$0	\$500	\$1000	\$1500	\$2000	\$2500	\$5000	\$7500	\$10 000
Motorway	4.9	4.3	4.7	4.9	5.1	4.8	6.0	5.6	5.5
Motorway link	6.7	6.5	7.4	7.2	7.6	8.4	8.9	8.5	7.8
Primary	0.7	0.7	0.7	0.7	0.7	0.7	0.8	0.9	1.1
Secondary	2.4	2.5	2.5	2.5	2.5	2.6	2.9	3.1	3.4
Tertiary	5.9	5.3	5.0	5.0	4.9	4.8	4.8	5.1	5.5
Residential	24.0	21.7	20.2	19.5	19.0	18.6	18.3	18.2	18.4
Service	58.5	46.7	43.8	40.8	39.0	39.2	45.0	50.1	53.3
Unclassified	34.2	25.7	24.0	18.2	17.6	18.2	17.2	15.9	19.3

Just like with the spatial coverage results, these figures show that ROUTr helps achieve a more uniform temporal coverage among road classes, as well. Indeed, by re-routing taxis from major city thoroughfares towards minor roads, ROUTr reduces the timegap between subsequent visits for the latter, at the cost of slightly increasing timegaps for the main road segment types. These trends are visible in Figure 6, which shows the relative change in median timegaps achieved by ROUTr with the considered incentivization budgets, w.r.t. the shortest-path algorithm. As for the fluctuations among the 10 repetitions of the experiments, also in this case there was a standard deviation smaller than 0.1%, so the error bars in Figure 6 have been omitted. The figure also shows that, at higher budgets, the median timegaps generally increase w.r.t. medium-low budgets. This is probably due to the fact that, as previously discussed, higher budgets are associated with a higher number of sensed segments, but the number of vehicles and trajectories remains the same, thus leading to less frequent visits for each of the covered road segments. Another interesting insight emerging from Figure 6 is that, in the considered case study, a budget of \$ 2500 seems to be the most cost-effective in terms of temporal coverage. Increasing the budget over that threshold helps achieve greater spatial coverage, but at the cost of generally decreasing the frequency of visits. This findings show that, for a Decision Maker of a Smart City investigating the feasibility of using a swarm of vehicles to crowd-sense information, it is crucial to carry-out preliminary studies and simulations, such





**FIGURE 6.** Relative change in median timegaps by segment type, w.r.t. A\*, achieved by ROUTr with the considered incentivization budgets.

as the case study we conducted, to determine these dynamics and select the best trade-off between incentivization budget and coverage.

## VI. CONCLUSION

Leveraging swarms of ridesharing vehicles for crowd-sensing is considered a promising and cost-effective solution, especially in urban scenarios [10]. Still, the achievable sensing coverage of a fleet of vehicles can be inadequate to support use cases requiring a more uniform and pervasive sensing on the urban road network [15], as the drivers typically prefer more efficient routes passing through the main urban arterial roads.

Many solutions have been proposed in the literature to obtain aimed sensing distributions in VCS, but most of these approaches rely on centralized components that might be expensive to operate at large scale, and based their coverage analyses at a coarse-grained scale of city areas, which can hardly provide sufficient insights for many use cases.

To address this issue, we have presented ROUTr, to the best of our knowledge the first decentralized, incentivization budget-aware routing solution, specifically designed to support VCS in achieving a broader and more-uniform road-network coverage in a fog computing fashion, without the need for any costly central coordination components. More in detail, given an origin, a destination, and an allowed budget, ROUTr generates a route whose incentivization cost is guaranteed not to exceed the budget. Thanks to the introduction of a random component, each sensing vehicle, given the same origin and destination, might compute a different route, thus increasing the overall number of monitored streets.

The proposed solution has been empirically evaluated by simulating its application in a real-world scenario, with different incentivization budgets. In particular, the assessment is based on a real-world mobility dataset of trajectories from about 400 taxis in the City of San Francisco (USA), and on open data from the OpenStreetMap project. The results of

this empirical assessment have highlighted that, even with incentivization budgets below \$2500 per three-weeks for the considered fleet of taxis, ROUTr can achieve significantly more uniform urban road network coverage w.r.t. a shortest-path algorithm, possibly enabling many additional VCS use cases. Moreover, our investigation also provided valuable insights on the achievable map coverage (measured at a fine-grained scale of single road-segment) and incentivization costs, as well as the existing trade-offs between these two factors, which are crucial to Decision Makers investigating the feasibility of leveraging high-mileage vehicle fleets to crowd-sense information in urban environments.

In future research, we aim at replicating this study in different cities, featuring possibly different traffic dynamics and road-network topologies to better evaluate the generalizability of these results. Moreover, in future works, we also plan on releasing an open source tool, based on the well-known open source KNIME Analytics Platform,<sup>2</sup> to allow Decision Makers to effortlessly carry out simulations like the one we described in this study. Such a tool could prove to be very valuable to Decision Makers investigating the feasibility of a VCS use case, as it would allow them to derive useful insights on the case at hand.

## REFERENCES

- [1] D. Asprone, S. Di Martino, and P. Festa, "Improving sensing coverage of probe vehicles with probabilistic routing," in *Proc. Int. Symp. Web Wireless Geograph. Inf. Syst.*, 2018, pp. 1–10.
- [2] D. Asprone, S. Di Martino, P. Festa, and L. L. Starace, "Vehicular crowd-sensing: A parametric routing algorithm to increase spatio-temporal road network coverage," *Int. J. Geograph. Inf. Sci.*, vol. 35, no. 9, pp. 1876–1904, 2021.
- [3] "AAA American automobile association, your driving costs—How much are you really paying to drive?" Feb. 2021. [Online]. Available: <https://exchange.aaa.com/wp-content/uploads/2019/09/AA-A-Your-Driving-Costs-2019.pdf>
- [4] F. Bock, Y. Attanasio, and S. Di Martino, "Spatio-temporal road coverage of probe vehicles: A case study on crowd-sensing of parking availability with taxis," in *Societal Geo-Innovation (Lecture Notes in Geoinformation and Cartography)*, A. Bregt, T. Sarjakoski, R. van Lammeren, and F. Rip, Eds. Cham, Switzerland: Springer Int., 2017, pp. 165–184.
- [5] F. Bock, S. Di Martino, and A. Origlia, "Smart parking: Using a crowd of taxis to sense on-street parking space availability," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 496–508, Feb. 2020.
- [6] F. Bock, S. Di Martino, and M. Sester, "What are the potentialities of crowdsourcing for dynamic maps of on-street parking spaces?" in *Proc. 9th ACM SIGSPATIAL Int. Workshop Comput. Transp. Sci.*, 2016, pp. 19–24.
- [7] A. Chakeri, X. Wang, Q. Goss, M. I. Akbas, and L. G. Jaimes, "A platform-based incentive mechanism for autonomous vehicle crowd-sensing," *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 13–23, 2021.
- [8] C. Chen et al., "Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1478–1496, Jun. 2017.
- [9] X. Chen et al., "ASC: Actuation system for city-wide crowdsensing with ride-sharing vehicular platform," in *Proc. 4th Workshop Int. Sci. Smart City Oper. Platforms Eng.*, 2019, pp. 19–24.
- [10] X. Chen et al., "PAS: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3719–3734, May 2020.

2. <https://www.knime.org/>

- [11] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," in *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput.*, 2013, p. 15.
- [12] S. Di Martino and S. Rossi, "An architecture for a mobility recommender system in smart cities," *Procedia Comput. Sci.*, vol. 98, pp. 425–430, Jan. 2016.
- [13] S. Di Martino and L. L. Starace, "Vehicular crowd-sensing on complex urban road networks: A case study in the city of porto," *Transp. Res. Procedia*, vol. 62, pp. 350–357, Jan. 2022.
- [14] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [15] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
- [16] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2982–2997, Dec. 2018.
- [17] Great Britain Ministry of Defence, *The Admiralty Manual of Navigation: General Navigation, Coastal Navigation and Pilotage*, vol. 1, TSO, London, U.K., 1997.
- [18] M. Haklay, "How good is volunteered geographical information? A comparative study of openstreetmap and ordnance survey datasets," *Environ. Plan. B, Plan. Des.*, vol. 37, no. 4, pp. 682–703, 2010.
- [19] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.
- [20] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 2542–2550.
- [21] J. A. Khan and Y. Ghamri-Doudane, "ROVERS: Incentive-based recruitment of connected vehicles for urban big data collection," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5281–5294, Jun. 2019.
- [22] R. E. Knack, "Pay as you park," *Planning*, vol. 71, no. 5, pp. 4–8, 2005.
- [23] O. Masutani, "A proactive route search method for an efficient city surveillance," in *Proc. 21st World Congr. ITS*, 2014, pp. 1–18.
- [24] O. Masutani, "A sensing coverage analysis of a route control method for vehicular crowd sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2015, pp. 396–401.
- [25] S. Mathur et al., "ParkNet: Drive-by sensing of road-side parking statistics," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Serv.*, New York, NY, USA, 2010, pp. 123–136.
- [26] M. Minghini and F. Frassinelli, "OpenStreetMap history for intrinsic quality assessment: Is OSM up-to-date?" *Open Geospatial Data Softw. Stand.*, vol. 4, no. 1, p. 9, 2019.
- [27] F. Morselli, F. Zabini, and A. Conti, "Environmental monitoring via vehicular crowdsensing," in *Proc. IEEE 29th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2018, pp. 1382–1387.
- [28] A. Ndikumana, N. H. Tran, D. H. Kim, K. T. Kim, and C. S. Hong, "Deep learning based caching for self-driving cars in multi-access edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2862–2877, May 2021.
- [29] Z. Ning, F. Xia, N. Ullah, X. Kong, and X. Hu, "Vehicular social networks: Enabling smart mobility," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 16–55, May 2017.
- [30] *Key:Highway—Openstreetmap Wiki*, OSM, San Francisco, CA, USA, Feb. 2021.
- [31] Z. Patterson and S. Farber, "Potential path areas and activity spaces in application: A review," *Transp. Rev.*, vol. 35, no. 6, pp. 679–700, 2015.
- [32] J. Pearl and J. H. Kim, "Studies in semi-admissible heuristics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 4, pp. 392–399, Jul. 1982.
- [33] K. Petty and W. Mahoney, III, "Enhancing road weather information through vehicle infrastructure integration," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2015, no. 1, pp. 132–140, 2007.
- [34] M. Piorowski, N. Sarafijanovic-Djukic, and M. Grossglauser, Feb. 2009, "CRAWDAD Dataset EPFL/Mobility (v. 2009-02-24)," CRAWDAD. [Online]. Available: <http://crawdad.org/epfl/mobility/20090224>
- [35] E. Porcu, M. Bevilacqua, and M. G. Genton, "Spatio-temporal covariance and cross-covariance functions of the great circle distance on a sphere," *J. Amer. Stat. Assoc.*, vol. 111, no. 514, pp. 888–898, 2016.
- [36] "San Francisco mobility trends report 2018." SFMTA. 2019. [Online]. Available: <https://www.sfmta.com/reports/san-francisco-mobility-trends-report-2018>
- [37] "Essential trip card guide." SFMTA. 2020. [Online]. Available: [https://www.sfmta.com/sites/default/files/reports-and-documents/2020/04/essential\\_trip\\_card\\_rules.pdf](https://www.sfmta.com/sites/default/files/reports-and-documents/2020/04/essential_trip_card_rules.pdf)
- [38] X. Shen, R. Fantacci, and S. Chen, "Internet of Vehicles [scanning the issue]," *Proc. IEEE*, vol. 108, no. 2, pp. 242–245, Feb. 2020.
- [39] A. M. Soley, J. E. Siegel, D. Suo, and S. E. Sarma, "Value in vehicles: Economic assessment of automotive data," *Digit. Policy Regulation Governance*, vol. 20, pp. 513–527, Oct. 2018.
- [40] J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in Internet of Vehicles," *Sensors*, vol. 16, no. 1, p. 88, 2016.
- [41] X. Wang, X. Zheng, Q. Zhang, T. Wang, and D. Shen, "Crowdsourcing in ITS: The state of the work and the networking," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1596–1605, Jun. 2016.
- [42] X. Wang, W. Wu, and D. Qi, "Mobility-aware participant recruitment for vehicle-based mobile crowdsensing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4415–4426, May 2018.
- [43] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial Internet of Things: A cyber-physical systems perspective," *IEEE Access*, vol. 6, pp. 78238–78259, 2018.
- [44] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh, "iLOCuS: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1831–1847, Aug. 2020.
- [45] "Yellow cab of San Francisco—Website." Yellow Cab of San Francisco. 2021. [Online]. Available: <https://yellowcabsf.com/>
- [46] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, 2015, pp. 73–78.
- [47] T.-Y. Yu, X. Zhu, and H. Chen, "GoSense: Efficient vehicle selection for user defined vehicular crowdsensing," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, 2017, pp. 1–5.
- [48] X. Zhang, Z. Yang, and Y. Liu, "Vehicle-based bi-objective crowdsourcing," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3420–3428, Oct. 2018.
- [49] Y. Zhao and C. H. Liu, "Social-aware incentive mechanism for vehicular crowdsensing by deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2314–2325, Apr. 2021.



**SERGIO DI MARTINO** is a Full Professor with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Italy, where he is also the Co-Chair of the Knowledge Management and Engineering Lab. He has published more than 100 papers in journals, conference proceedings and books. His research interests focus on knowledge discovery and exploitation from complex spatio-temporal datasets, especially in the field of mobility.



**LUIGI LIBERO LUCIO STARACE** received the M.Sc. degree (*cum laude*) in computer science from the University of Naples Federico II, Italy, in 2018, where he is currently pursuing the Ph.D. degree in information technology and electrical engineering. His research interests include software engineering, design and evaluation of end-to-end testing techniques, formal methods, and applications of computer science to intelligent transportation systems and smart cities.