# Push and Pull Participant Recruitment System for Vehicular Crowdsensing

*Tzu-Yang Yu*

School of Computer Science
McGill University
Montreal, Canada

December 2020

---

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

# Dedication

*To my wife and parents for their continued support*

## Acknowledgements

Foremost, I would like to express my deepest gratitude to my adviser Prof. Muthucumaru Maheswaran for the continuous support of my Ph.D study and research. His constructive feedback and profound knowledge really helped me to improve the quality of my thesis writing and his unwavering enthusiasm kept me constantly engaged with my research. I am also thankful to the *School of Computer Science, McGill University* for giving me the opportunity to do research and providing invaluable guidance throughout this research.

I thank my fellow lab-mates in *Advanced Network Research Lab*: Xiru Zhu, Tianzi Yang, Fadahunsi Lekan, Ricahrd Ayoola Olaniyan and Richboy Kaycee David for their insightful inputs and stimulating discussions, and for all the fun we have had together in our lab.

Last but not the least, I would like to thank my family for supporting me in every aspect of my life in McGill University. Special thanks to my wife Yi-Tang Chang, her support and encouragement really overcome the barriers along the way during my time at McGill.

# Abstract

Crowdsensing, the use of everyday devices to collect and share data is paving the way for cost-efficient real-time data collection. Every day, information can be quickly sensed and shared publicly using smartphones. Beyond smartphones, modern vehicles have also shown great promise for crowdsensing. In contrast to mobile crowdsensing, vehicles are ideal platforms to collect, store, compute, and share large amounts of sensor data. Vehicles have greater mobility and cover wider sensing area. The mobility patterns of vehicles are predictable due to the prevalence of navigation systems. Most importantly, the abundance of on-board resources and lack of power constraints makes it possible to support complex and long-lasting sensing tasks.

The idea behind vehicular crowdsensing is to leverage vehicles as mobile sensors and computing resources. Vehicles are recruited as sensing participants for large-scale crowdsensing tasks such as urban sensing or traffic condition monitoring. However, existing works often assume that sensing tasks are common tasks where sensing results are shared with the public [1]. Instead of public information, we believe the benefit of the crowdsensing paradigm should be available for personal use. In this, we focus on small-scale sensing tasks, tasks that dynamically change over time and are unlikely to be shared. Thus, sensing information is collected on demand rather than continuously and at all times. We refer to this as personalized vehicular crowdsensing. Furthermore, most works assume the routing of vehicular participants cannot be changed. This further reduces a system's ability to fulfill dynamic sensing needs. Thus, in this thesis, we further explore the possibilities of improving crowdsensing performance by vehicle route planning.

To achieve either public vehicular crowdsensing or personalized vehicular crowdsensing, we must resolve the problem of vehicular participant selection. We first propose two solutions based on push and pull for vehicular crowdsensing. We aim to maximize sensing coverage, improve load balance, error tolerance, and minimize costs. Next, we improve the sensing performance by allowing vehicles to reroute. Instead of greedily generating routes to maximize overall sensing coverage, we

leverage a two-step global and local planning algorithm. Our global algorithm attempts to plan a vehicle's route based on the difference between the distribution of vehicular location and the distribution of task location. Our goal is to send vehicles to areas with a high number of tasks but having few vehicles to service them. The global algorithm does not determine which tasks it should service; this operation is handled by the local algorithm which decides how to optimally leverage a vehicle's sensing ability for a small area. Through the use of SUMO simulation and TAPAS Cologne Large Scale Mobility Dataset, we show that our proposed approaches deliver significant performance improvements compared to traditional approaches.

# Résumé

La crowdsensing, l'utilisation d'appareils de tous les jours pour collecter et partager des données ouvre la voie à une collecte de données en temps réel et rentable. Chaque jour, les informations peuvent être rapidement détectées et partagées publiquement à l'aide de téléphones intelligents. Au-delà des téléphones intelligents, les véhicules modernes sont également très prometteurs en matière de crowdsensing. Contrairement à la crowdsensing avec des appareils mobiles, les véhicules sont des plateformes idéales pour collecter, stocker, calculer et partager de grandes quantités de données de capteurs. Les véhicules sont plus mobiles et couvrent une zone de détection plus large. Les modèles de mobilité des véhicules sont prévisibles en raison de la prédominance des systèmes de navigation. Plus important encore, l'abondance des ressources embarquées et l'absence de contraintes de puissance permettent de prendre en charge des tâches de détection complexes et de longue durée.

L'idée derrière la crowdsensing à bord des véhicules est d'utiliser les véhicules comme des capteurs mobiles et des ressources informatiques. Les véhicules sont recrutés comme participants à la détection pour des tâches de détection de foule à grande échelle telles que la détection urbaine ou la surveillance des conditions de circulation. Cependant, les travaux existants supposent souvent que les tâches de détection sont des tâches courantes dont les résultats sont partagés avec le public. Au lieu d'informations publiques, nous pensons que les avantages du paradigme de la crowdsensing devraient être disponibles pour un usage personnel. Pour cela, nous nous concentrons sur des tâches de détection à petite échelle, des tâches qui changent dynamiquement avec le temps et qui ne sont pas susceptibles d'être partagées. Ainsi, au lieu de collecter des informations continuellement et en tout temps, elles sont collectées sur demande. C'est ce que nous appelons la crowdsensing personnalisée avec des véhicules. En outre, la plupart des travaux partent du principe que l'itinéraire des participants ne peut pas être modifié. Cela réduit encore la capacité d'un système à répondre aux besoins de détection dynamique. Ainsi, dans cette thèse, nous explorons davantage les possibilités d'améliorer les performances de crowdsensing par la planification des itinéraires des véhicules.

Pour parvenir à une crowdsensing publique ou personnalisée, nous devons résoudre le problème de la sélection des participants. Nous proposons d'abord deux solutions basées sur la poussée et la tirée pour la crowdsensing avec des véhicules. Nous visons à maximiser la couverture de la détection, à améliorer l'équilibre de la charge, la tolérance aux erreurs et à minimiser les coûts. Ensuite, nous améliorons les performances de détection en permettant aux véhicules de se rediriger. Au lieu de générer de façon gourmande des itinéraires pour maximiser la couverture globale de détection, nous exploitons un algorithme de planification globale et locale en deux étapes. Notre algorithme global tente de planifier l'itinéraire d'un véhicule en se basant sur la différence entre la distribution de l'emplacement des véhicules et la distribution de l'emplacement des tâches. Notre objectif est d'envoyer des véhicules dans des zones où le nombre de tâches est élevé, mais où il y a peu de véhicules pour les desservir. L'algorithme global ne détermine pas les tâches qu'il doit prendre en charge; cette opération est gérée par l'algorithme local qui décide comment exploiter au mieux la capacité de détection d'un véhicule pour une petite zone. Grâce à l'utilisation de la simulation SUMO et de l'ensemble de données sur la mobilité à grande échelle de TAPAS Cologne, nous montrons que les approches que nous proposons offrent des performances nettement supérieures à celles des approches traditionnelles.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Overview

Sensing data collection is an essential requirement for proactive services [2]. For instance, services can make use of connected sensors to fetch city-wide data and analyze it for better decision-making [3]. Thus, a critical problem is how to efficiently deploy these sensing devices to cover the entire sensing area (e.g. entire city). Luckily, modern vehicles are equipped with increasingly powerful sensors, communication interfaces, and computing resources. As such, vehicles are quickly becoming a new paradigm for data collection [4, 5, 6]. Since the vehicles are owned by different individuals, data collection using the vehicles is referred to as vehicular crowdsensing. In vehicular crowdsensing, vehicles are recruited as sensing participants for large-scale crowdsensing tasks such as urban sensing or traffic condition monitoring. Compared to conventional mobile crowdsensing which relies on hand-held devices [7], vehicles are ideal platforms to collect, store, compute, and share large amounts of sensor data. The advantages are manifold; for instance, vehicles have greater mobility and cover wider sensing area [8]. Furthermore, the mobility patterns of vehicles are predictable due to the prevalence of navigation systems. Most importantly, the abundance of on-board resources and lack of power constraints enable complex and long-running sensing tasks.

As smart vehicles begin to roam the streets, new possibilities will emerge for large-scale data acquisition tasks necessary for proactive smart-city applications [9, 10]. The primary application of modern vehicular crowdsensing focuses on large-scale monitoring such as environment and traffic monitoring, map updating, public safety, urban sensing and so on [11, 12, 13, 14, 15, 16, 17, 18]. As such, collected data are primarily analyzed in a cloud server and results made available for public use [19, 20]. Such information can be reused by multiple applications. Given its nature, large-scale sensing is dominated by enterprises or governments. We believe the benefit of the crowdsensing paradigm should be available for personal use; tasks that are varied but limited in scale. We define this paradigm as Personalized Vehicular crowdsensing (PVC) [21]; it focuses on supporting user-specific sensing tasks.

Unlike generalized crowdsensing tasks, sensing tasks catered towards personalized requests are unlikely to be shared with other users. For instance, different sizes of trucks require varying road width for driving and turning. However, due to construction, snow, events or even bad parking, passable roads may no longer be traversable. Hence, it is necessary to look in real-time for a wide variety of road width for different sizes of trucks. Such a road width requirement depends on the type of truck; the system should allow the user to tune the road width parameter as a sensing task.

One related application, Waze, also attempts to leverage vehicular crowdsensing for everyday users [22]. In Waze, participants form part of a community that gathers information such as police location, traffic or roadblocks location. However, unlike our proposal, Waze does not support customized user inquiries; sensing tasks are predefined by the platform. Besides, Waze requires participants to actively enter information; a participant who sees an accident must manually enter the information while driving. In contrast, PVC does not require participants to be actively engaged. The client generates a customized sensing task as a runnable program and submits the program to a selected vehicle as shown in Fig 1.1. The selected vehicle executes the program, sends the result back to the requester if the task objective is met.

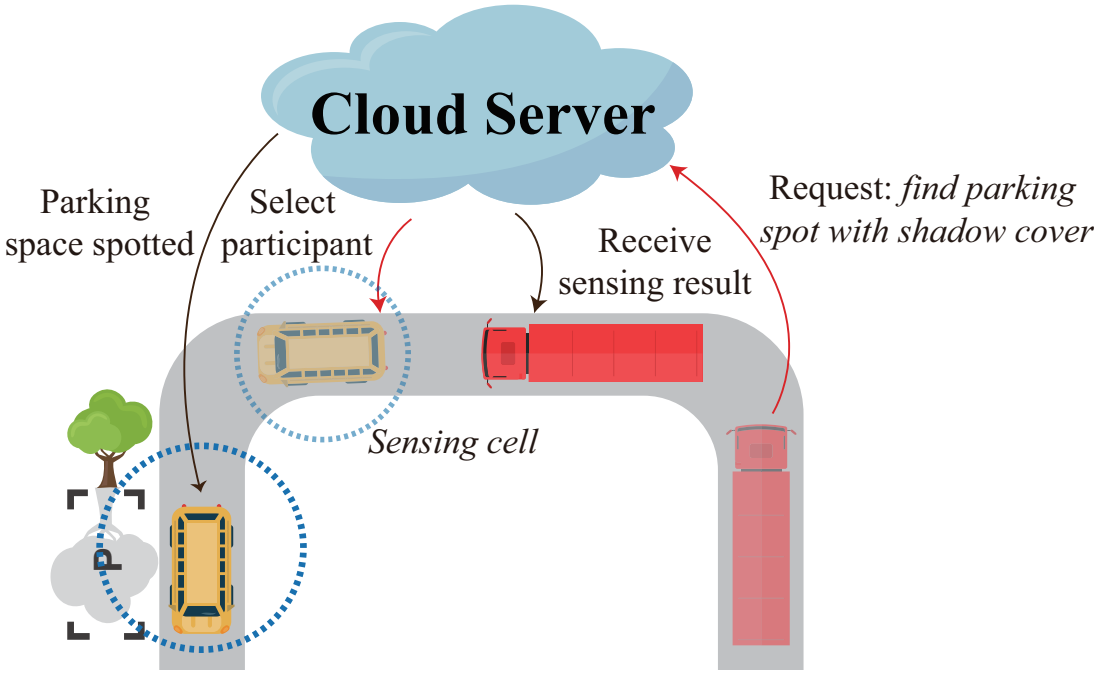


**Fig. 1.1** Example of PVC paradigm: a vehicular client requests sensing information. e.g. finding a parking space with shadow cover ahead of time

Applications like Waze rely on users manually entering sensing results which may not be trustworthy if users intentionally submit faulty sensing results for their benefit. For instance, users who want to have better traffic conditions while driving can report accidents on the road. Thus, vehicles moving in the same direction may be directed to other routes by the system. In contract, our PVC enables the sensing program to implement security policies. The participant running the sensing program must follow the policies; otherwise, the task result will be rejected. Thus, a program-based sensing task not only can support customized sensing tasks but can also reduce trust concerns.

Moreover, unlike conventional cloud computing which consists a number of static servers, a vehicular crowdsensing system comprises a dynamic collection of vehicles (mobile servers). Due to the spatio-temporal nature of moving vehicles, efficiently selecting and leveraging vehicular participants' on-board resources is one of the key challenges for building a vehicular crowdsensing service. A popular

research area in vehicular crowdsensing literature involves selecting a set of participants to cover all sensing areas while satisfying sensing constraints (e.g., budget cost) [23, 24]. Since these sensing tasks are location-dependent, data is valuable to the requester only when it is collected from the targeted sensing area and satisfies the task deadline [25, 26]. Sensing areas are assigned by requesters and are distributed over space and time. Although many participant recruitment algorithms have been recently added to the literature, unique characteristics of PVC introduce several new challenges in designing participant recruitment algorithms.

Furthermore, many existing works assume the participant's current trajectory is fixed and cannot be rerouted for better sensing performance. This limits the overall sensing coverage and performance. For instance, a location of interest where few vehicles are passing-by will suffer from a lack of sensing coverage. Thus, if a vehicular participant is willing to take a detour, performance in areas with less traffic can be vastly improved. However, vehicle detouring will consume extra time and power, and it will increase the chance of missing the scheduled time to the final destination. Therefore the driver must be sufficiently motivated for taking the detour. There are many existing incentive mechanisms or pricing models that were introduced for participatory tasks [27, 28, 29, 30]. For example, Uber, one of the largest ride-sharing company uses dynamic pricing models to incentivize drivers to pick up customers that are located at an unpopular area [31]. Thus, we believe that the driver will be willing to take a detour to collect data if we reward the driver properly.

In this work, we focus on exploring the challenges of providing recruitment algorithms for PVC system as well as propose and evaluate several recruitment algorithms for PVC tasks. We are also looking into possibilities of increasing sensing performance by rerouting vehicles to an unpopular area.

## 1.2 Thesis Contribution

This thesis makes the following six major contributions:

1. We explore a new crowdsensing paradigm in which clients can recruit a set of vehicles to perform personalized vehicular crowdsensing tasks.

2. We formulate the personalized vehicular crowdsensing participant recruitment problem and we prove the problem is NP-complete.

3. We propose push-based and pull-based participant recruitment systems. Furthermore, these systems not only can efficiently recruit necessary vehicular participants but also can reduce overall load balance given a massive number of sensing tasks.

4. We explore a new route planning based system for vehicular crowdsensing, and we also prove that the problem of vehicle route planning for sensing tasks is NP-complete.

5. We introduce two route planning solutions for vehicular crowdsensing: the first solution relies on a centralized route selection algorithm that can efficiently select which participant and which route it should undertake. The second solution leverages a two-step global and local planning algorithm. Our global algorithm plans the route of participants based on the task probabilities and vehicle location probabilities and attempts to spread vehicular agents to minimize the difference between the two distributions. In contrast, local planning looks at its immediate surroundings and attempts to fulfill published tasks.

6. We evaluate our algorithm with the TAPAS Cologne Large Scale Mobility dataset. The result shows that our algorithm is superior to traditional approaches in performance metrics.

## 1.3 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we provide some background research and discussion on vehicular crowdsensing. This includes the crowdsensing architecture, communication protocol, and commonly used incentive mechanisms for participant recruitment in crowdsensing. Chapter 3 describes our motivation for using vehicular crowdsensing for personalized sensing tasks and explores the possibility of using a vehicle route planning mechanism to increase crowdsensing performance. An overview of the existing literature relevant to the participant recruitment problem we tried to address is given in Chapter 4. Chapter 5 introduces our novel push and pull-based participant recruitment solution for vehicular crowdsensing as well as evaluates the proposed approaches and presents detailed performance analysis. In Chapter 6, we further improve sensing performance by using a vehicle planning mechanism. In this chapter, we give the set of proposed algorithms for both global planner and local planner and analyze our proposed algorithms by evaluating simulated results over the TAPAS Cologne dataset. Finally, a summary of the thesis is presented in Chapter 7 and possible future extensions of this research are also briefly indicated in the same chapter.

## 1.4 Co-author Contribution

Major portions of this thesis are already published in the following three publications [21, 32, 33]. In both [21] and [32], Tzu-Yang Yu played the key role of designing and implementing the experiments. Xiru Zhu and Hongji Chen provided help with simulation and problem modelling, respectively. In [33], Xiru Zhu and Shabir Abdul Samadh devised the main conceptual ideas. Tzu-Yang Yu and Xiru Zhu worked out all of the technical details, and all authors carried out the simulations and studied the results.

# Chapter 2

# Background Research for Crowdsensing

## 2.1 From Crowd-sourcing to "Vehicular crowdsensing"

The idea of crowdsensing is inspired from crowd-sourcing. Crowd-sourcing is the idea of relying on the general public to participate and complete a task [34]. Crowd-sourcing tasks are varied and include translation, photo tagging, website testing. article writing and so on [35]. Crowdsensing, on the other hand, is a specific type of crowd-sourcing task that requires a large group of individuals equiped with mobile devices which are capable of sensing and computing to collect sensing data [36].

Traditionally, sensing data are commonly collected from pre-installed fixed sensors such as road side cameras, hygrometers, or thermometers. However, building a large fixed sensor network is expensive for covering an entire city. Rather than building expensive fixed sensors over a sensing region, we can leverage the ubiquitous availability of everyday mobile devices and their mobility. Crowdsensing has come to play a key role in real time information dissemination. Everyday, millions of users share real time information and participate in crowdsensing. In many cases, information shared is mundane such as road potholes, photos of locations, or traffic jams [37, 38]. This aggregated information can be used by applications to improve people's daily lives [39].

Depending on type of sensing devices, crowdsensing can be categorized into mobile crowdsensing and vehicular crowdsensing. For conventional mobile crowdsensing, the sensing device often refers to hand-held devices such as a smart phone, smart watch, etc [40, 41, 42, 43]. With these devices, participants need to actively participate in the sensing task such as entering local information with their smart devices. We refer to such sensing behavior as participatory sensing given it requires the participant to be actively involved in the sensing task [44, 45, 46]. Moreover, these handheld devices contain powerful sensors and can be used for any kind of sensing tasks both indoor or outdoor [47, 48]. However, due to their size, these wearable sensing devices have limited battery and computational power. Hence, they are not suitable for a long lasting and computational heavy sensing task.

In contrast, vehicular crowdsensing relies on smart vehicles which contain powerful sensors and long lasting battery power as seen in Fig 2.1. However, vehicles can only collect sensing data on the road and vehicles must move; a vehicle cannot stop in the middle of the street for long. Thus, vehicles can only gather data passively along their route as they pass through locations of interest. We refer to such automatically sensing as opportunistic sensing because it does not require participants to be actively engaged [49, 50, 51, 52]. To actively involve vehicular participants, we request vehicles to reroute while they are driving to areas which urgently need sensing coverage. Such crowdsensing system should be considered participatory, and is not sufficiently studied in the literature.

Another issue we must consider is privacy where both recruiter information and participants' information must be protected in crowdsensing [53, 54, 55]. On the recruiter side, sending a request may give away the recruiter's intent [56]. On the participant side, sensing gives away current movements and position [57, 58]. After a participant collects data from the surrounding environment and the data is uploaded to the cloud or sent back to the requester, the participant loses control over the collected data. Malicious entities including the service provider, requester, or external hacker can extract various types of personal information from the sensing reports (e.g., location, preferences, health status, and political

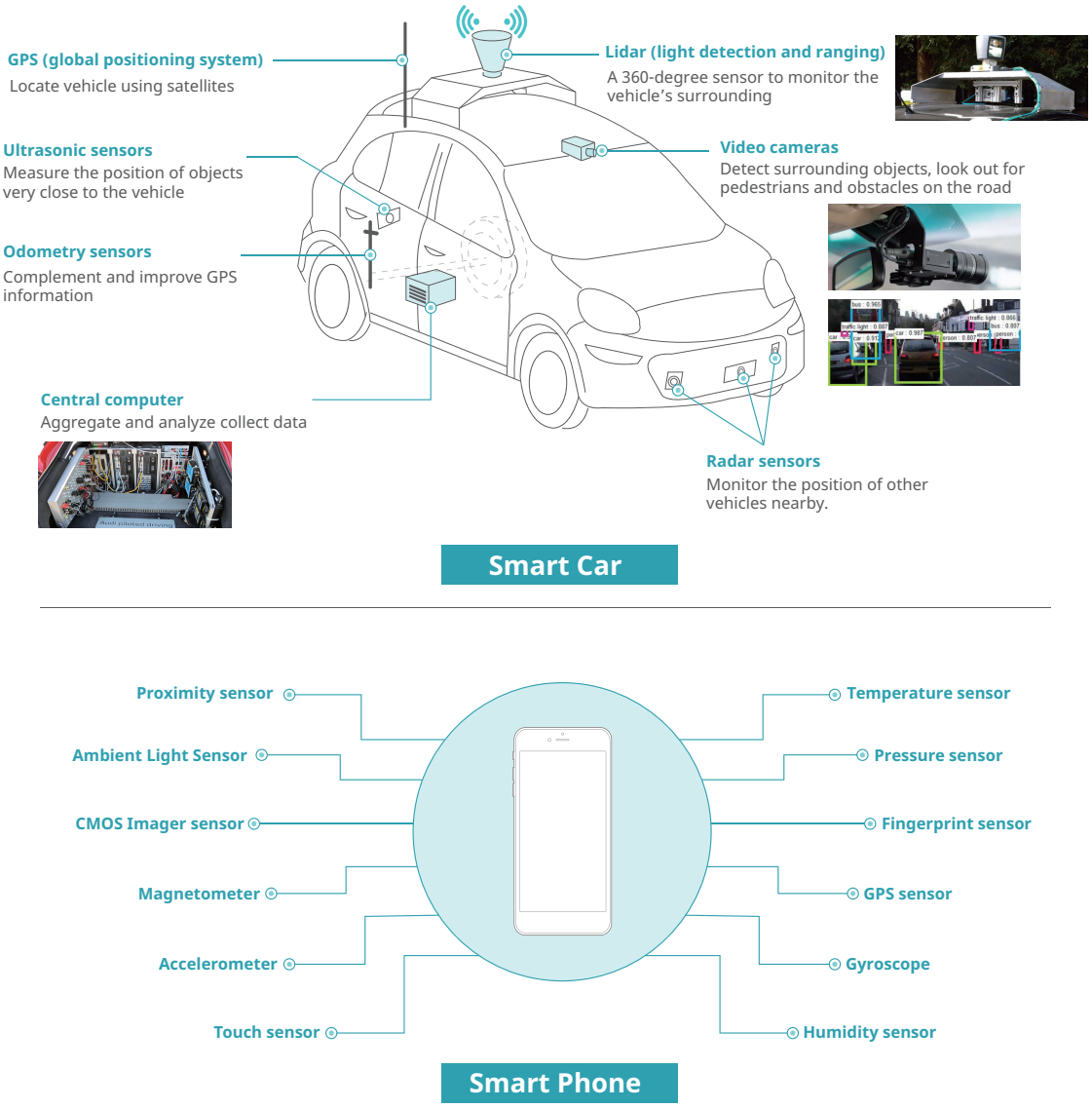


**Fig. 2.1** Sensors comparison

affiliation) [59, 60, 61]. Furthermore, mobile crowdsensing data collection relies on participants to manually enter information. Hence, participants may accidentally or intentionally deliver false reports and as result, the requester may be confused and make poor decisions. For instance, participants may use multiple identities to report false traffic information to gain better benefits. This is in contrast to vehicular crowdsensing where data is collected by vehicles automatically and par-

ticipants are less likely to change the sensing data. However, some professional hackers can still modify the data report if the communication protocol are not secured. Thus the crowdsensing platform should ensure the sources of sensing data are fully trusted and behave honestly by implementing some sort of trust management system into the platform [62, 63, 64, 65, 66, 67].

## 2.2 Crowdsensing Architecture

A typical crowdsensing system mainly consists of three components; the platform, requesters, and participants [68, 69, 70]. The platform provides communication protocols between the requester and participants, security control, and manages the incentive system [71, 72]. Requesters submit sensing tasks to the platform. Participants receive and complete tasks, and transmit the results back to the requester through the platform. A typical crowdsensing architecture consists of three layers: service layer, fog layer and sensing layer which is illustrated in Figure 2.2. The sensing layer consists of various sensors which can be broadly categorized into fixed sensors or mobile sensors. Fixed sensors such as roadside cameras or ombrometers are located in important locations. However, these sensors are often owned by a large organization. On the other hand, mobile sensors such as wearable devices or vehicles are owned by individuals. Thus, the location of mobile sensors is dynamic and can be moved around the city.

A fog layer consists of several fog servers and these servers are located near the sensors to maintain a stable and direct communication link. All requests and collected sensing data must pass through the middle layer of the fog servers. The fog server cleans the data by de-duplicating it or reformatting the data before forwarding the data to the main cloud servers in the service layer [73, 74]. The collected data are analyzed in the cloud servers and are used to provide a useful crowdsensing application for public users [75].

## 2.3 Communication Protocols

There are two principal methods for sensing devices to communicate with the cloud and crowdsensing systems; Wireless Local Area Network (WLAN) and cellular network (e.g. Long-Term Evolution (LTE) or 5G). WLANs are often known by their commercial product name Wi-Fi and they use multiple parts of the IEEE 802 protocol family, and are designed to work seamlessly with the wired Ethernet. In general, for vehicular users, the IEEE 1609 Family of Standards for Wireless Access in Vehicular Environments (WAVE) utilize IEEE 802.11p to enable secure vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) wireless communications [76]. We refer to the vehicular network as vehicular ad-hoc networks (VANETs) [77]. In VANETs, road side units (RSUs), fixed infrastructure installed alongside roads, are used as local servers. Vehicles can access data in RSUs or access the Internet through RSUs using the dedicated short range communication (DSRC) protocol stack [78]. RSUs can provide inexpensive connections to vehicles nearby. However, they suffer from short range; maximum radio range is about 1 km. As such, a sparse deployment of RSUs cannot always guarantee complete network coverage. To fix reliability issues, some works for VANET include LTE to improve reliability [79, 80].

A cellular networks on the other hand, have a wider service range. They are designed for citywide/national/global coverage areas and seamless mobility. Even though modern cellular networks such as LTE-A or 5G can support a bit rate of more than 1 Gb/s [81], dramatic expansion of multimedia data traffic continuously challenges bandwidth availability of the cellular backhaul [82]. Some analyzers also estimate that autonomous cars will generate more than 300 TB of data per year [83]. Hence, existing mobile devices still needs Wi-Fi service as a solution to network congestion.

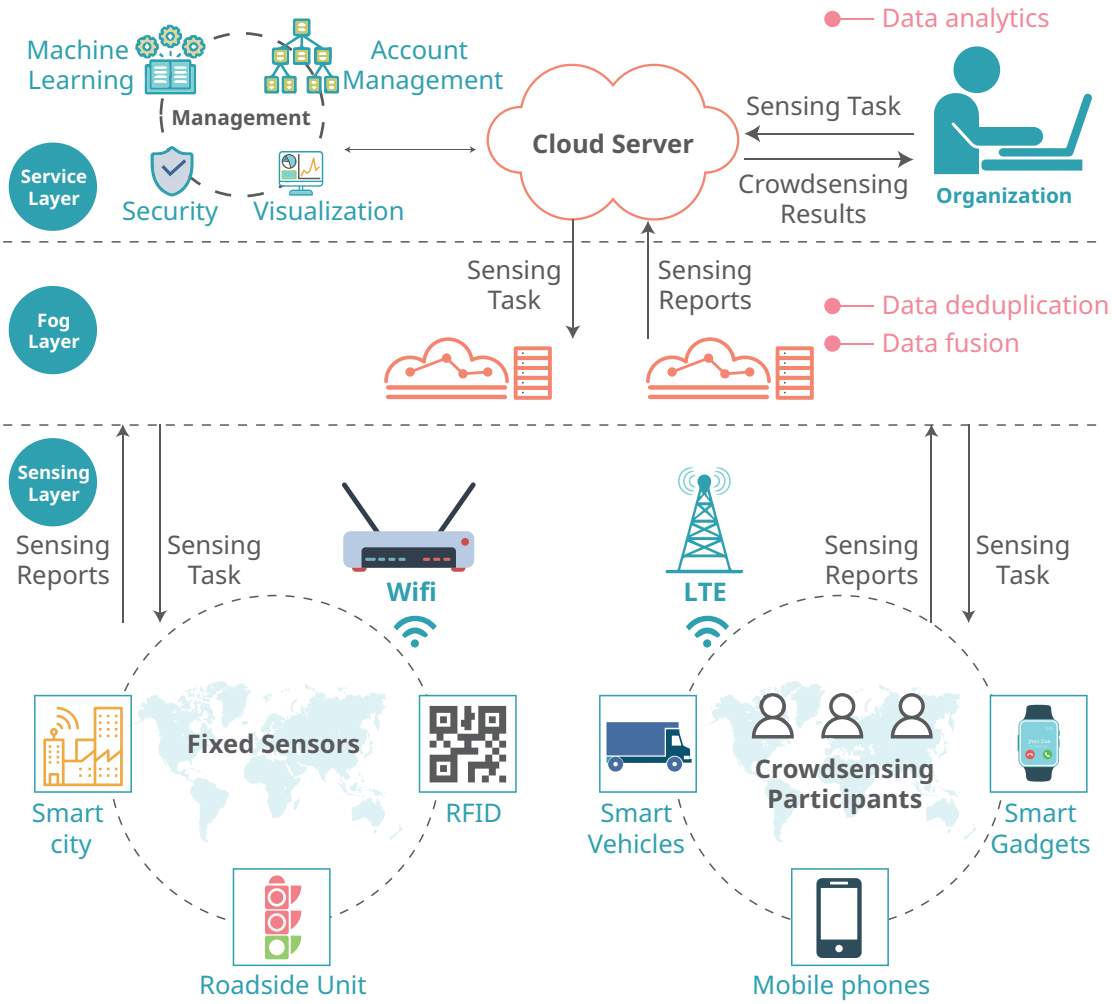


**Fig. 2.2** crowdsensing architecture

## 2.4 Type of Sensing Task

Based on locality of the sensing area, a sensing task can be categorized into two types; static and dynamic. Static sensing tasks consist of urban sensing or environmental sensing where the sensing location is fixed [84, 85, 86, 87]. In such scenarios, the sensing area often covers the entire city or target some specific infrastructure such as buildings or bridges [88, 89]. Recent mobile crowdsensing research focus with small scale sensing tasks because the tasks do not have strict deadline and often requires participant to stay the sensing area for a some amount of time [90, 91]. Static sensing tasks are also heavily studied in vehicular crowd-

sensing literature [92, 93, 94, 95]. They focus more on large scale data collection in urban areas [96, 97, 98, 99]. For instance, updates for Google street view can be simplified using vehicular crowdsensing due to its ability to cover large sensing area by moving along the way. Thus, we could have real time street view from surrounding vehicles.

On the other hand, a task where sensing area shifts over time is dynamic as seen in Figure 2.3. For example, a task would be to find a parking space while driving. In such a personalized request, sensing areas are shifted based on the requester's (the driver's) position. Thus, sensing data needs to be timely to meet the task deadline. Such sensing tasks can only be served by vehicular participants due to its high mobility. The life time of the sensing task could only last for few minutes and could not be completed by the participants with wearable devices. In the literature, however, only few works focus on solving such dynamic sensing tasks.

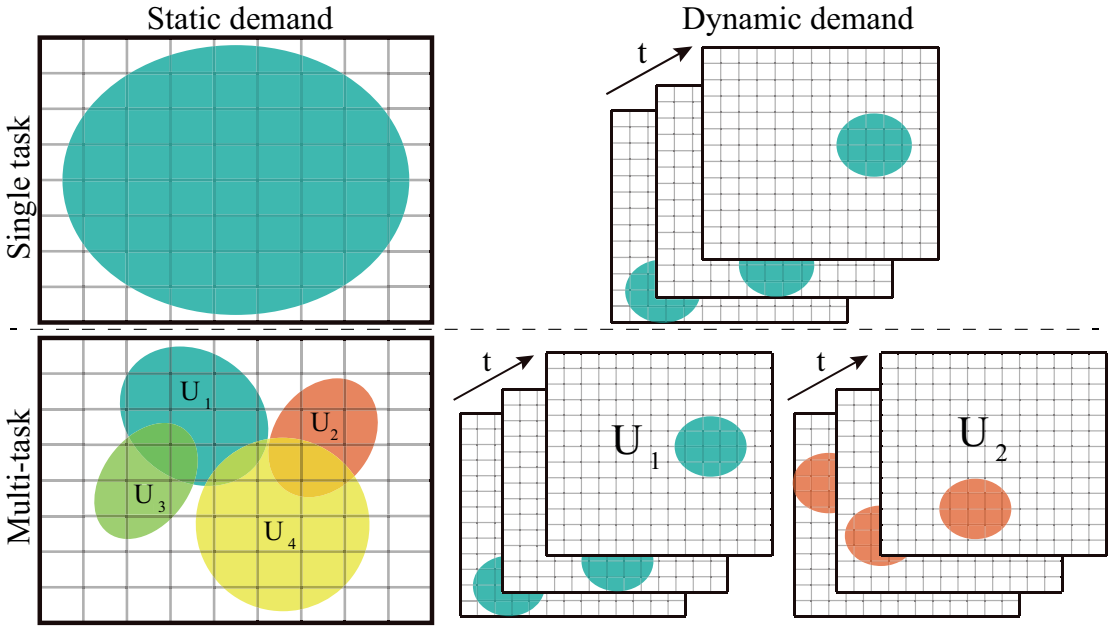


**Fig. 2.3** Classification of different type of sensing demands. $U$=user.

## 2.5 Incentive Mechanism

Crowdsensing applications require a critical number of participants to achieve sufficient service quality [100, 101]. Without sufficient participants, it is impossible to collect adequate sensing coverage to meet user needs. However, ordinary individuals are reluctant to participate and share their sensing capabilities due to insufficient incentives; participating in sensing systems may incur costs and risks. For instance, when a smartphone user participate in a sensing task, the task inevitably consumes the resources of the smartphone, including computation, communication, and energy [102]. Besides, collected data contains personal information making users sensitive to privacy feel uncomfortable [103]. Therefore, it is conceivable that ordinary individuals will not participate in sensing tasks unless they are sufficiently motivated.

**Incentive mechanism for mobile crowdsensing:**

The design of incentive mechanisms for mobile crowdsensing system has received significant attention in recent years [104, 105, 106]. Mobile crowdsensing requires significant incentives for participation compared to vehicular crowdsensing [107, 108]. For instance, many mobile sensing tasks require smartphone user to manually take out the smart device to sense such as taking pictures or recording sound. Besides, since participants are heterogeneous in sensing capabilities [109, 110]; participants should be treated differently such that efficient users should be encouraged to participate over less efficient users. For instance, if a sensing task is to collect the photos of rare animals, participants with zoological experience would yield better performance compared to those who do not. Hence, the recruiter should hire the best participants to assure overall data quality as well as minimizing expenses on weakly performing participants [111, 112, 113].

In the literature, incentive mechanisms for mobile crowdsensing systems can be broadly classified into two types: requester-centric incentive mechanisms and participant-centric incentive mechanisms [114, 115]. In a requester-centric incentive model, the requester decides on the payment rules. The participants focus their efforts on completing sensing tasks based on the reward provided. Follow-

ing task completion, the requester evaluates performance of the participants and selects high quality results from collected data. In a participant-centric incentive mechanism (also referred to auction-based mechanisms) [116, 117], each participant announces a reserve price before performing the sensing task [118, 119]. The requester can select a subset of participants based on a budget constraint [120, 121, 122]; this approach cannot guarantee the quality of sensing. For a task which is quality sensitive or requires a specific type of participant such as a participant with expertise in a specific field, contract-based mechanisms are used [123]. In contract-based incentive mechanisms, the platform will publish quality payment bundles to participants, and the participants receive payment only when the specified qualities are met [105, 124, 125]. Not all incentive mechanisms involve monetary payment. Some information sharing applications such as Waze [22] require users to participate sensing to access pooled sensing data. That is, users are both participants and consumers of sensing information.

**Incentive mechanism for vehicular crowdsensing:**

Current research on incentive mechanisms for vehicular crowdsensing applications is limited. Most of the incentive mechanisms used for vehicular crowdsensing are price based [126, 127]. In general, applications which use vehicles as sensor do not heavily rely on human participation; vehicles simply gather data as they move [12]. Besides, the vehicle's position is trackable due to embedded GPS systems. Furthermore, a vehicle's sensors and computational capacity are known due to standardized manufacture. Hence, the sensing performance of a vehicle is more predictable compared to mobile phone sensing. Since these moving vehicles are mobile computational devices, the recruiter only needs to select a subset of the vehicular participants to achieve the sensing objectives given a limited budget.

The price-based mechanism described above is suitable for selecting vehicular participants based on their current trajectory. It assumes the participant's trajectory is fixed and cannot be modified for improved sensing performance. We refer to such participants as passive. Running sensing tasks relying only on passive participants cannot guarantee overall sensing quality. For instance, a place of interest

with few vehicles passing by will not be completely sensed. However, if a vehicular participant is willing to take a detour, we can increase the coverage area. We refer to such vehicles as active participants [33]. However, rerouting consumes a participant's time and requires direct participation. Thus a reward system must be designed to incentivize users to tolerate rerouting while minimizing the recruiting costs. Hu et al. proposed an incentive mechanism based on reverse auction to select active participants [128]. Their approaches require a recruiter to announce the sensing task containing places of interest (POI) and a deadline to participants. Participants reply with bids that may cover partial POIs by their possible trajectories and the corresponding costs. The recruiter decides which bids should be selected to minimize the recruiting cost while ensuring the sensing quality. Thus, participants who are willing to take more detour toward the announced POI, will be selected.

## 2.6 Participant Selection Problem

To provide effective crowdsensing service, effective recruitment schemes are necessary to minimize the number of participants required; otherwise, we would run out of participants. Selection criteria are diverse based on application requirements; these include the expertise of a participant [129], requester budget constraints [130, 131], and spatio-temporal coverage of places of interest [132, 133]. For selecting expert participants, a background check is required before the participant is selected. In mobile crowdsensing, recruiters attempting to select hand-held device participants may check their historical performance or experience [134]. For vehicular crowdsensing, the capabilities of on-board sensors can be directly verified before selection. To reduce recruitment cost, eliminating participants who might produce redundant data can be considered [135]. For example in Times Squares, we would have multiple participants willing to provide similar information on the New Year's eve event. Selecting all participants for a sensing task will give redundant data which not only increases the cost but also increases the burden on data transmission and data management[136]. As seen in Figure 2.4, a proper participant selection mechanism can reduce the sensing cost while still guaranteeing the sensing coverage.

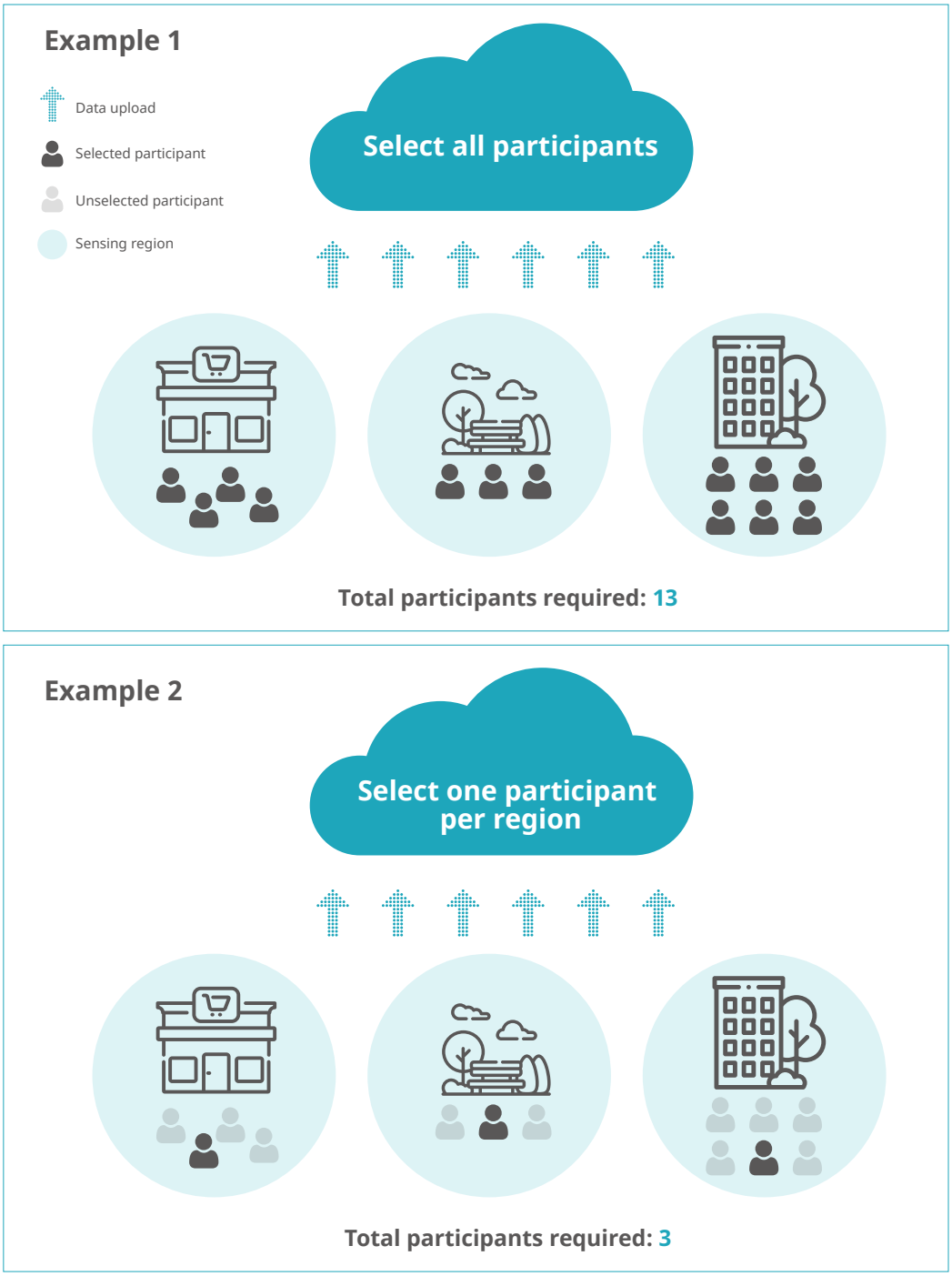


**Fig. 2.4** We can reduce number of participants by eliminating participants who might produce redundant data.

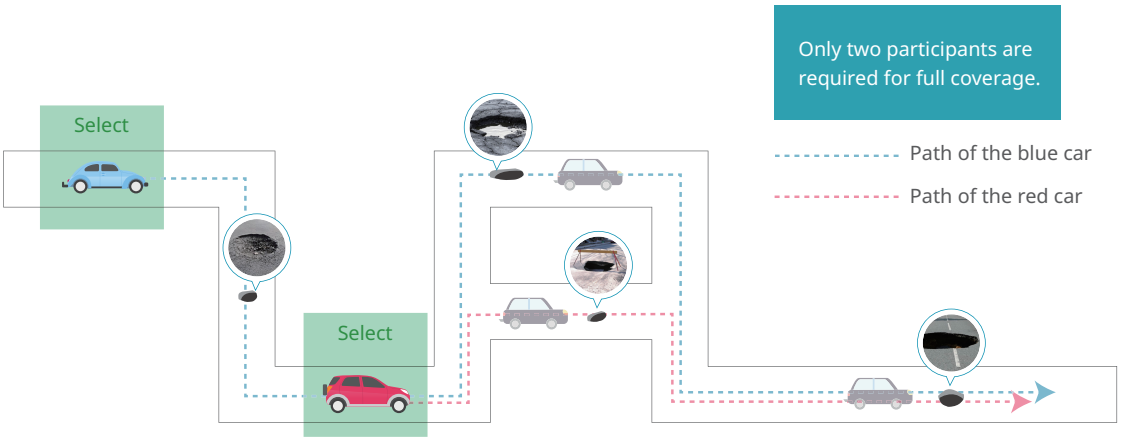


**Fig. 2.5** Example: only two vehicular participants are required for full coverage.

To reduce the number of participants, crowdsensing research focuses on selecting participants which can maximize sensing coverage. Figure 2.5 shows that selecting participants who are traveling through multiple uncovered sensing regions can reduce duplicated sensing data. To enquire whether the required monitoring areas are fully sensed, the predicted participant route is often used to meet a spatio-temporal coverage requirement [137]. In mobile crowdsensing, participants with hand-held devices can move freely around the city [138]. However, participants walk on the streets unpredictably from one location to another. Due to such behaviors, most of the predicted routes are obtained based on data-driven prediction via their historical call and location traces [139, 140, 141, 142, 143]. In vehicular crowdsensing, on the other hand, most vehicles can only run on roads. Additionally, the mobility pattern of vehicles is predictable due to the prevalence of GPS-based car navigation systems [144, 21, 5]. Specifically for self-driving vehicles, we know that the navigation system will be always on.

| | Handheld device | Vehicle |
|---|---|---|
| **Mobility pattern** | Unpredictable | Predictable (navigation system) |
| **Power supply** | High power constraint (battery) | Low power constraint |
| **Computational resources** | Limit | abundant |
| **Sensing characteristics** | participatory | opportunistic |
| **Data quality** | Very diverse (depends on participant performance) | Not very diverse (automatically collect from sensors) |
| **Incentive mechanism** | Price based, auction house based, contract based..etc. | Price based (similar to renting computational resources in cloud) |
| **Security issue** | High, malicious users can send faulty information for their own benefits | median |

**Fig. 2.6** Characteristic between handheld device and vehicle

## 2.7 Summary and Discussion

The key differences between vehicles and handheld devices are summarized in Figure 2.6. First, vehicles carry extremely powerful sensors such as 360-degree cameras and other sensing devices as opposed to mobile devices have sensors that are more limited due to size. Self-driving cars must contain powerful sensors to properly function. Second, vehicles can cover a larger sensing area due to their high mobility; covering a large area with a similar number of mobile devices may prove to be difficult. This also means when the number of participants is sparse, vehicles are advantageous. For example, at 2 in the morning, covering an avenue with the few vehicles passing by is not difficult. In contrast, the number of mobile devices required for sensing would leave most of the street without sensing information. Third, vehicles have more predictable routes; we can obtain vehicle GPS routing, unlike a person strolling on the streets. This allows for more effective planning. Fourth to maintain participation, handheld device participants require complex and fine-grained incentive mechanisms [104, 145, 146]; it requires participants to use their mobile phones and actively gather sensing data from their local environment, whereas vehicles do not require active engagement to complete tasks. Indeed, the quality of sensing results depends on the onboard sensors rather than the human factor. The participant only lends the on-board resources to the

recruiter, which can be equivalent to buying computational resources from cloud servers. Thus, typical monetary schemes used in cloud computing can be applied for vehicular crowdsensing. Finally, vehicles do not suffer from significant battery life issues which can be an issue for mobile phones working longer time periods. Evidently, vehicular crowdsensing cannot be deployed everywhere. Unlike mobile devices, vehicles cannot sense in areas outside of roads; beyond crashing vehicles in buildings, sensing in those areas should remain the prerogative of mobile crowdsensing [147]. In addition, the dwell time of vehicles in an area is often shorter than the mobile device participant. Thus the vehicle cannot do a long processing task in a small area.

Due to these unique characteristics of handheld devices and vehicles, participant selection algorithms for mobile crowdsensing cannot be directly applied to vehicular crowdsensing. In this thesis, we explore a variety of existing algorithms for the vehicular crowdsensing participant selection problem. We also provide our own solution to improve selection performance.

# Chapter 3

# Motivation

## 3.1 Personalized Vehicular Crowdsensing

With more and more self-driving cars hitting the roads, there is an opportunity to create a sensing cloud that can support a variety of sensing tasks leveraging the sensors already built into the autonomous cars. Analysts expect that the autonomous vehicles market is expected to reach $20 billion by 2024; growing at a CAGR (Compound Annual Growth Rate) of 25.7% from 2016 to 2024 as shown in Figure 3.1. In addition, The Boston Consulting Group predicts autonomous vehicles will represent 25% of the global market by 2035 as shown in Figure 3.2. As a result, 21 million autonomous vehicles will reach the roads globally by 2035, and traffic management systems will free up commute time 1.9 trillion minutes for passengers [148]. Thus, the future crowdsensing system should support personalized crowdsensing due to the popularity of using smart vehicles. Personalized crowdsensing focuses on supporting user-defined sensing tasks. These tasks are dynamic and are catered towards a specific user's requests. Unlike sensing such as air pollution [149], personalized sensing requests are unlikely to be requested by other users. For instance, a user is attempting trying to pass a vehicle in a two-lane road and this can be dangerous when the line of sight is blocked by vehicles ahead. Thus, a vehicle could request for traffic information on the opposing lane by vehicles ahead of it. Such information should be gathered on demand rather than everywhere given that we do not often need it.

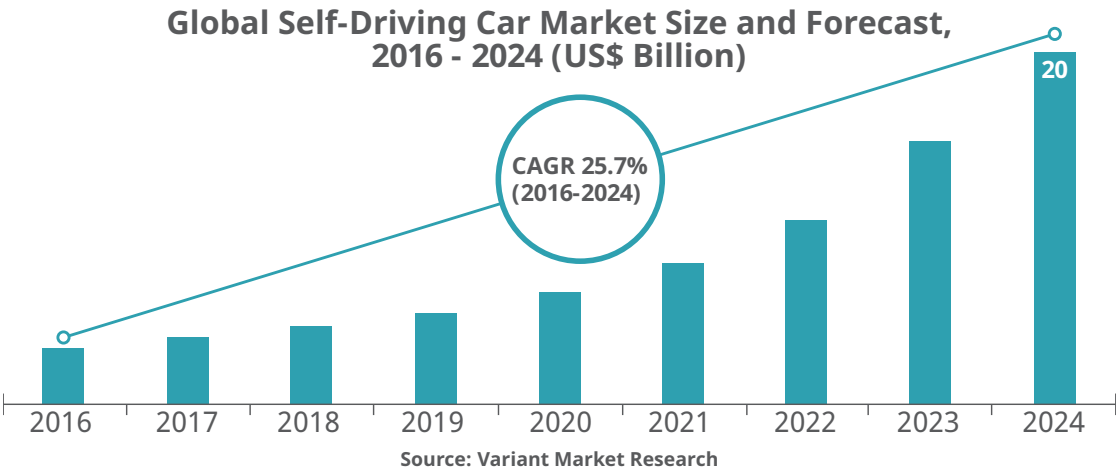


**Fig. 3.1** Market size

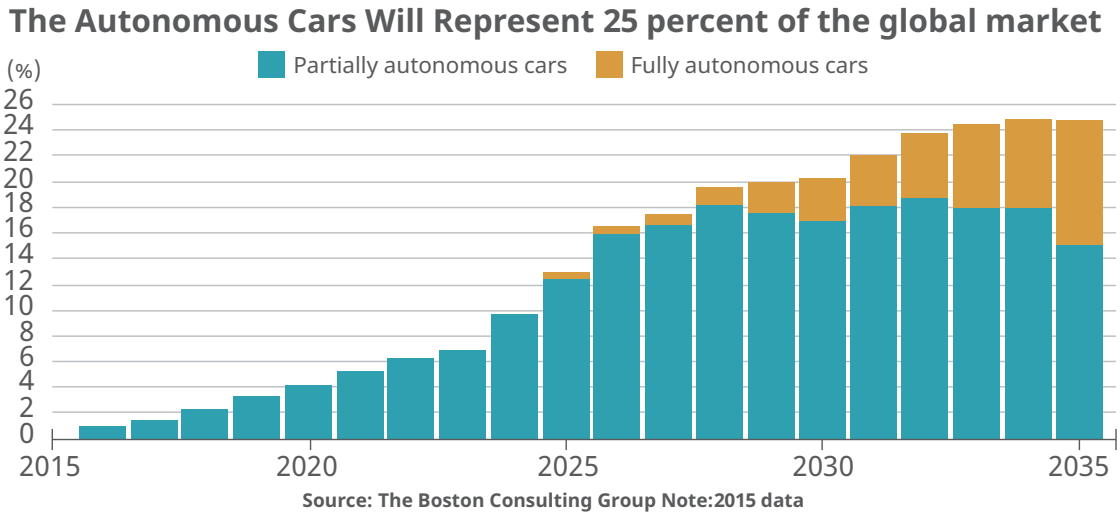


**Fig. 3.2** Autonomous cars global market

### 3.1.1 Future Applications

Beyond public sensing that focuses on large-scale common data gathering, personalized vehicular crowdsensing focuses on serving small scale, time-sensitive, customized sensing tasks. Therefore, its potential applications diverge greatly from traditional sensing tasks. For example, a driver may seek information about an exit on the freeway to determine when they can change lane can hire a set of vehicles ahead to monitoring the traffic condition. Such sensing tasks are highly time-sensitive because the tasks must be completed before the driver reaches the exit. Furthermore, such sensing tasks require lane-level traffic monitoring which remains a very popular research topic in nowadays literature [150].

Moreover, personalized vehicular crowdsensing enables user-customized requirement for their sensing tasks. For example, because of the various sizes of cars, the driver can search for a parking space that is suitable for their car. The size of parking space, such as for downtown street parking, is dynamic as it depends on the other vehicles' parking position. Some other parking requirements such as a location in the shade, flat surface, or no snow blocking the area can also be added to the sensing requests.

Although the applications above were specific to drivers, nondriving users can also take advantage of personalized vehicular crowdsensing to obtain timely sensing information. For instance, a user who lost their cat or dog can hire a set of vehicles to monitor potential areas where it might be. The sensing task finishes when the missing animal is found. Hence, the lifetime of the sensing task depends on the time of task completion. Other requests such as verifying whether a vendor is open during the holiday or counting the length of the line at the grocery store during peak hours can also prove to be useful applications of personalized vehicular crowdsensing.

### 3.1.2 Challenges

Unlike conventional large scale sensing tasks, the unique characteristics of personalized vehicular crowdsensing have introduced several new challenges in designing participant recruitment algorithms. These include moving monitoring area, load balancing, and inexact predicted positions. In the following subsections, we describe and explain these challenges in detail by providing examples as well as experimental results.

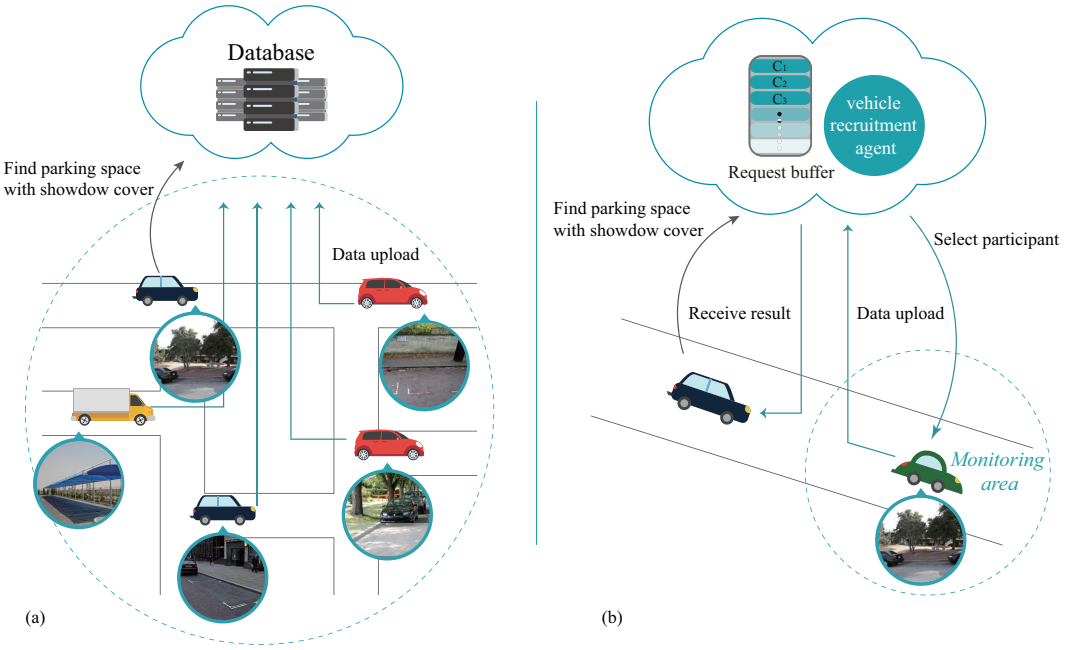


**Fig. 3.3** Given a request to find a parking space with shadow cover, public vehicular crowdsensing would gather information over the entire sensing region for parking and shadow. In contrast, personalized vehicular crowdsensing requires sensing for a specified area at specific time points, limiting the amount of sensing information required.

#### Moving monitoring area

Because the requester is a moving entity (driver), we require timely sensing data; sensing information within a time constraint. For instance, suppose a requester is interested in finding a parking slot near her destination. The requester's future route consist of 3 unique regions $R = \{r_1, r_2, r_3\}$, each region requiring 10 minutes to traverse. In such tasks, the requester does not want to know the parking status

at region $r_3$ immediately while the requester is still in the region $r_1$. This is because the parking space might be taken away before the requester arrives at $r_3$ which requires a total 20 minutes driving time. Thus the system should recruit participants for monitoring $r_3$ only when the requester is at $r_2$. Figure 3.3 shows an example to describe the difference between traditional public crowdsensing and personalized vehicular crowdsensing in terms of participant selection. In such an example, participant recruitment approaches for public crowdsensing suffers from over recruitment. This is very inefficient for PVC tasks since we only need to maintain partial coverage at indicated locations rather than at all times and locations.

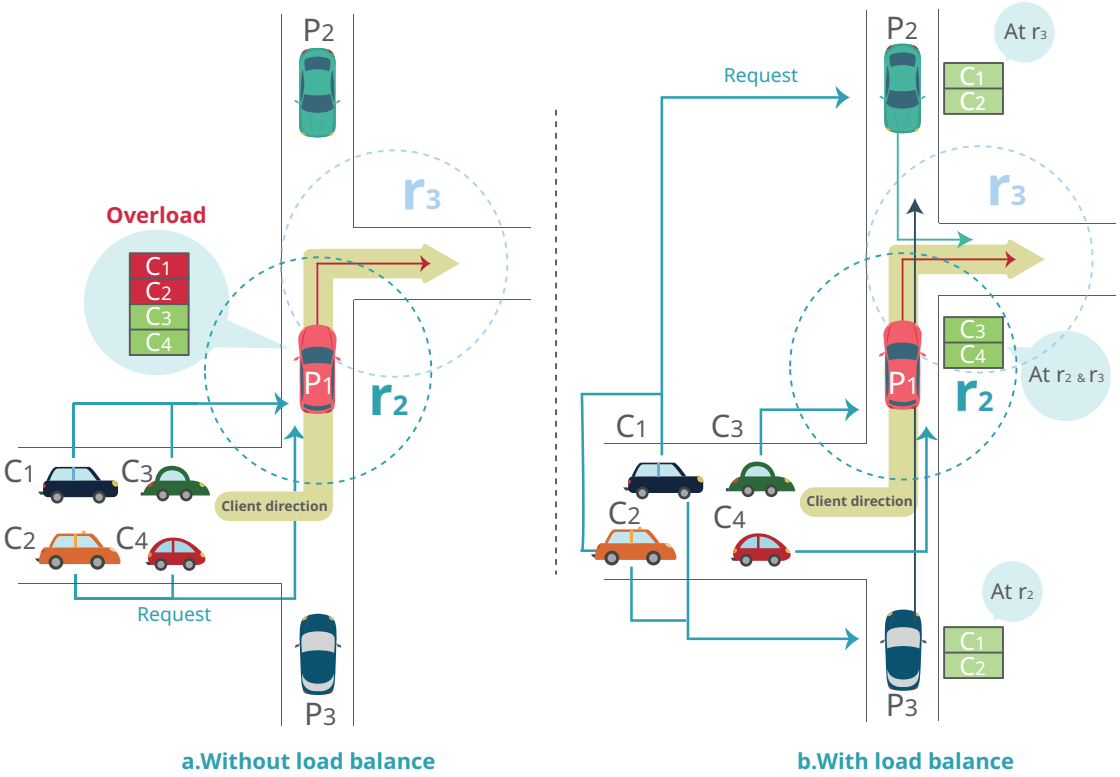


**Fig. 3.4** We can see the effects of overloading where the requester's workload arrives at p1 which can only process 2 tasks. In contrast, using load balancing, we can increase the number of tasks successfully completed.

### Load balancing

In the PVC system, clients can submit multiple requests as needed. In addition, participants will receive sensing tasks from different clients along this journey. Given our goal of minimizing participant recruitment costs, vehicles with significant overlapping regions for sensing tasks are frequently picked. However, due to limited on-board resources on each vehicle, a single vehicular participant may not be able to process all requests. For instance, in Figure 3.4, four clients $\{c_1, c_2, c_3, c_4\}$ and a participant $p_1$ are heading towards the same direction. Suppose the four clients are able to assign sensing request to $p_1$ for monitoring the sensing regions $r_2$ and $r_3$ as shown in Figure 3.4a. The cost of participant recruitment for each client given the sensing task can be completed by hiring only one participant. Suppose $p_1$ can only serve 2 requests concurrently; sending all 4 requests to $p_1$ will overload the participant, leading to potential task failures. Thus, sensing tasks from clients all $c_1$ and $c_2$ should be assigned to $p_2$ and $p_3$, even though the recruitment cost is increased as shown in Figure. 3.4b. When a large number of sensing requests are submitted to the server, we must prevent overloading selecting participants for the tasks. Otherwise, an overloaded participant may not execute all tasks within the time constraints.

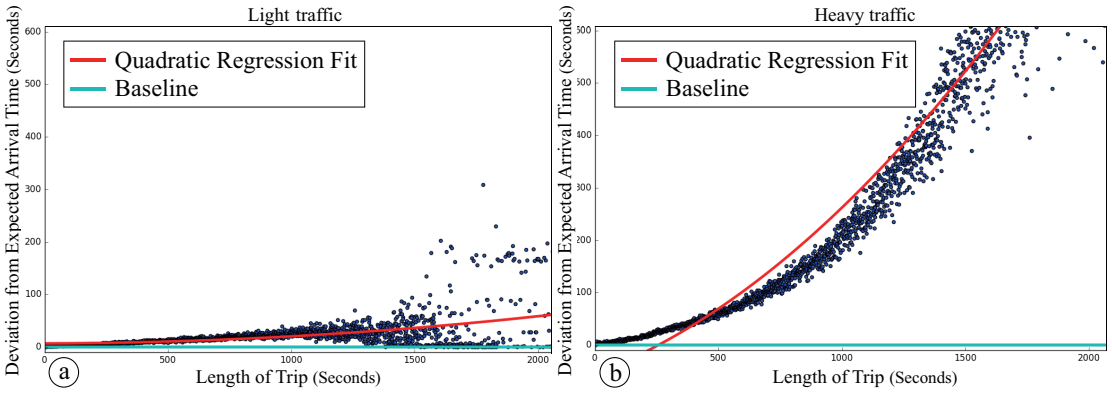


**Fig. 3.5** The y-axis measures the difference in time from very light traffic expected arrival time. We can see the deviation from arrival time increases at a different rate based on traffic.

#### Inaccurate future position

Because participant selection depends on the predicted time and route of participant vehicles, large route changes could force task rescheduling. We hypothesized that as the length of a trip increases, the error in predicted locations of participant vehicles will increase. To test this hypothesis, we evaluate the predicted trip error based on different traffic levels using the TAPAS Cologne simulated vehicle trace. As shown in Figure 3.5a, as the length of the trip increase, the expected deviation from the predicted arrival times increase. Further as shown in Figure 3.5b, we can see that as traffic increases, the expected deviation also increases. Overall, this would mean scheduling participants far ahead in time would lead to rescheduling when participants no longer meet Spatio-temporal constraints. Rescheduling would occur more frequently when faced with longer trips. Furthermore, research has shown that the travel time of each road segment can be affected by several attributes such as speed limit, the number of lanes, bus stops, weather, time of day, etc [151]. Frequent rescheduling not only wastes computation a resources but also requires the selection of more participants to complete the task, increasing cost. Thus, participant scheduling should not consider an entire sensing route. Instead, a window-based scheduling approach could be advantageous. For example in Figure 3.5, the deviation from expected arrival time is small within a trip length of 300 seconds. Hence, we may only need to schedule all requests which fall within a window base of 300 seconds to reduce the rescheduling risk.

## 3.2 Vehicular Route Planning For Vehicular Crowdsensing

Existing works for participant recruitment focus on selecting a set of vehicles to maximize sensing coverage given a constrained budget. Nonetheless, these works assume the participant's current trajectory is fixed and cannot be rerouted for better sensing performance. That is, participants only gather data passively along their planned route. Although a participant's route and path may change, the vehicular crowdsensing system itself cannot change the participant's path. While experimenting with these systems, we have come to the realization that data collected by vehicles is severely imbalanced. Certain areas have very high coverage

while other regions receive little coverage because very few vehicles pass by. Running sensing tasks by relying on these passive participants cannot guarantee overall sensing quality. For instance, a location of interest where few vehicles are passing-by will suffer from a lack of sensing coverage. Thus, if a vehicular participant is willing to take a detour, performance in areas of low traffic can be vastly improved. For instance in Fig 3.6 (a), suppose we need to cover all the road segments, however, the original paths of the three vehicles based on the shortest traveling time manner cannot cover all the roads. Thus, we need to reroute some of the vehicles to cover all regions while enabling participants to arrive at their desired destination on time as shown in Fig 3.6 (b).

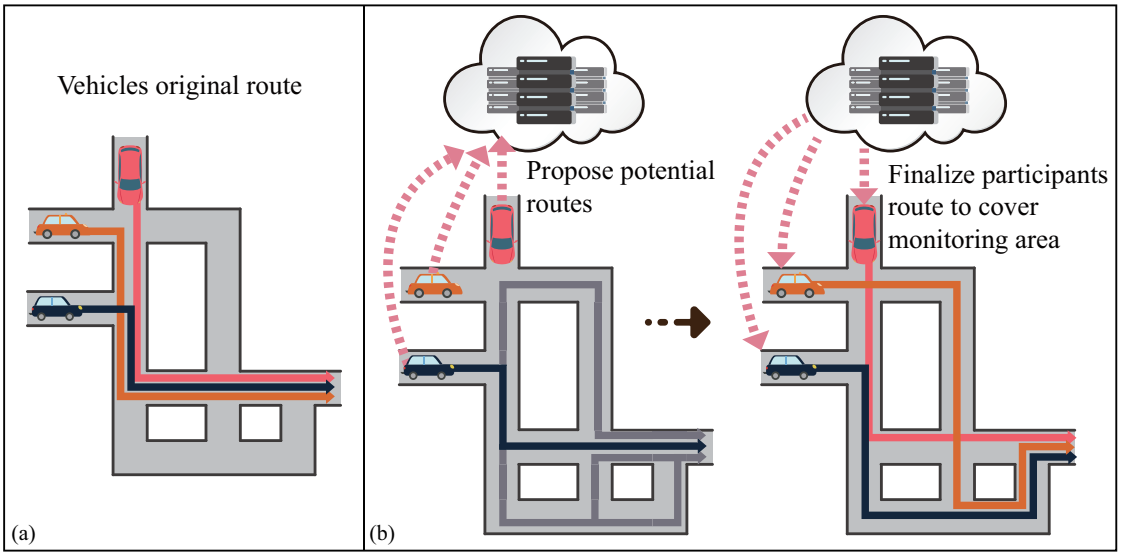


**Fig. 3.6** In this example, three participants share overlapping sensing routes. Instead of collecting three times the same information, we can reroute to improve overall sensing coverage. Each participant generates new potential routes and proposes them to the server. The server decides which routes and participants to use to maximize coverage.

Rerouting vehicles require a method for generating and route planning in addition to participant selection. Routing planning has been widely studied in multi-robot planning research. Thus, we explored various multi-robot planning algorithms for sensing data collection. For instance, Portugal et al. proposed a multi-robot patrol system that emphasizes minimizing the average time between visits for all nodes. In this work, each robot generates its next step by computing

the value of visiting each neighboring node and selecting the maximum. This value is computed based on a Bayesian learning strategy based on the last visited time. To prevent overlap, communication between robots is assumed each time a path movement is decided [152]. Unfortunately for vehicular crowd sensing with thousands of vehicles, having each communicate about its next path independently is too complex to optimize.

Furthermore, most of the solutions for multi-robot planning cannot be directly applied to vehicular crowdsensing; we must obey road constraints when rerouting vehicles. Each road has a defined location where it can transition to other roads. Furthermore, we must respect road direction restrictions and road size. Finally, beyond the distance traveled, we must also consider the problem of speed. Unlike in the robotic system where speed is assumed to be uniform, speeds on roads can vary widely.

# Chapter 4

# Related Works

With vehicular crowdsensing, one major problem is to reduce redundant data [153]. Unlike fixed sensors, vehicles move around and it's likely that sensing coverage overlap with others. This means selecting all vehicles for a sensing task would result in redundant data. For instance, sensing air quality using every vehicle in the same location would result in processing a large quantity of similar information; selecting only a few would have been sufficient [154]. Processing large quantities of data requires large data centers that add costs. Thus, vehicular crowdsensing requires a proper participant selection mechanism to select participants which can maximize coverage while also limiting costs.

In the literature, a public vehicular crowdsensing system is comprised of a set of participant vehicles with a wireless transceiver and sensor devices. Participants are either smart cars or buses [155]; both can carry onboard computing resources and sensing devices. Many studies assume that we cannot control the participant's movements [156]; participants do not actively participate in sensing tasks. Selected participants only gather data passively along their route as they pass the location of interest.

To store accumulated data, cloud servers are necessary. Cloud servers receive and aggregate all data collected. Then, applications can request sensing information from the data center. We assume the cloud servers know the participants'

current location; participants could beacon periodically. Many works in the literature assume knowledge of participants predicted route; this can be information from a GPS for instance. However, some work in the literature only assumes knowledge of past routing information. At regular intervals, the cloud servers can schedule a subset of available participants to collect sensing data.

When a vehicle participates in crowdsensing, it must be rewarded and most recent studies rely on monetary incentive systems to maintain participation. A typical cost for a vehicular participant, $v$, is based upon a fixed cost plus a variable cost.

$$C(v) \rightarrow C_{fixed} + C_{variable} \tag{4.1}$$

The variable cost is an extra reward for the vehicle if it satisfies certain criteria. For instance, a vehicle can get more payment if it covers more regions, if it spendes more time on the sensing task, or if it generates high-quality data. Such cost variables are various and depend on the focus of the research. Given a cost function, the goal for the participant recruitment problem is to select a set of vehicular participants that maximize the sensing coverage while minimizing the cost.

## 4.1 Participant Selection Algorithms for Public Vehicular Crowdsensing

In this section we will explore a variety of algorithms proposed for public vehicular crowdsensing participant selection. We define notations to be used when describing participant selection algorithms for this section (See Table 4.2). We define a sensing region $R = \{r_1, r_2, ..., r_m\}$, composed of smaller areas. The exact definition and size of each area $r_i$ differs with each approach. For instance, in [157], each area consist of graph nodes whereas in [158], they consist of 1 meter by 1 meter square areas. We define the set of vehicles $V = \{v_1, v_2, ..., v_n\}$ . We assume each $v_i$ has sensors and is willing to collect information. We define vehicles selected as participants as $P = \{p_1, p_2, ..., p_m\}$ where $P \subseteq V$. We define the time window as $T = \{t_1, t_2, ..., t_q\}$ where each $t_j$ is a time unit. Let $r_{v_i,t_j}$ be the location of vehicle $v_i$ at time $t_j$ where $r_{v_i,t_j} \in R$. We define $C$ as a function which when given a vehicle

| Notation | Description |
|---|---|
| $T$ | Sensing time window. |
| $V$ | Set of all vehicles. |
| $P$ | Set of vehicular participants selected for sensing $P \in V$. |
| $R$ | Set of areas for sensing. |
| $t_i$ | time i within time window T. |
| $v_i$ | vehicle i in V. |
| $p_i$ | participant i in P. |
| $r_i$ | area i in R. |
| $r_{v_i,t_j}$ | Location of vehicle $v_i$ at time $t_j$. |
| $C(v_i)$ | Function which returns the cost of participant $v_i$. |
| $B$ | The total budget of sensing task. |

**Table 4.2** List of notations for Section 3.2

returns the cost of recruiting such vehicle. We define $B$ as the budget constraint which limits the number of participants we can select. Finally, we define *coverage* as a function which takes in a participant set and sensing region and returns the coverage measure of selecting such participant set.

Data quality is often considered in mobile crowdsensing research [159, 160, 144, 161, 162, 163, 164, 165]. For instance, Hamid et al. first proposed the idea of utilizing vehicular trajectory to best select vehicular participants [127]. In this paper, each participant has a reputation based on its past sensing results, participant commitment and quality of the information provided. Participant commitment is the likelihood a participant will follow its provided trajectory. The quality of information is based on past sensing quality of the participant. Hence the reputation of a participant $v$ is given by the following equation where $\alpha$ and $\beta$ are weights.

$$reputation(v) = \alpha * commit_v + \beta * quality_v \tag{4.2}$$

The reward for each participant, $cost_v$ is based upon a fixed cost plus a variable cost based on coverage distance $d_v$ and reputation.

$$C(v) = C_{fixed} + d_v * reputation(v) \tag{4.3}$$

The problem is formulated as two-step integer linear programming, one for maximizing the number of the region covered and the second for minimizing the cost of participants which achieves maximum coverage. In the first step, Hamid et al. maximize the regions covered while remaining within budget $B$.

$$Maximize \sum_{t_j \in T} |\bigcup_{p_i \in P} r_{p_i,t_j}| \; subject\ to \sum_{v \in P} C(v) < B \tag{4.4}$$

In the second step, the distance covered by selected vehicles is minimized while maintaining the same level of coverage in step one. This will minimize the total cost.

$$Minimize \sum_{p_i \in P} p_i * d_{p_i} \tag{4.5}$$

Han et al. present two participant recruitment algorithms based on predicted trajectory [6]. They used the same objective function as described in Equation 4.4. Note that, in their work, the cost, $C(v_i)$ for selecting $v_i$ is assumed to be 1 $\forall v_i \in V$. The cost of selecting a vehicle is the same as selecting any other vehicle. Thus, the budget constraint $B$ is the number of participants selected.

The first algorithm in [6], referred to as the offline algorithm, assumes full knowledge of all vehicles and their trajectory within a time window $T$. It selects vehicles that maximize coverage. The algorithm then iterates until $B$ vehicles have been selected. The algorithm's time complexity is $O(B * |V| * \log(|V|))$. In contrast, the second algorithm, referred as the online algorithm, assumes no prior knowledge of a vehicle before it joins the crowdsensing system. Since the system is unaware of the vehicle and its trajectory before it enters into range, this may be a more realistic assumption. The algorithm decides whether to select a vehicle $v_i$ when it joins the system by comparing the gain in spatio-temporal coverage of adding $v_i$ with a dynamic threshold. The dynamic threshold is computed based on the number of participants already selected. The online algorithm's time complexity is $O(B * |V|))$.

Similarly, He et al. also proposed two participant recruitment algorithm based on vehicular trajectory [157]. Both algorithms assume full knowledge of vehicles and their trajectory within a time window $T$. The crowdsensing cost $C$ is generated according to a normal distribution. To evaluate the solution, a traffic trace dataset was obtained from TAPAS-Cologne [166], a 24 hour-generated vehicular trace of the city of Cologne in Germany simulated using SUMO [167]. The first algorithm consists of a greedy approximation that aims to recruit a set of participants to maximize their cost effectiveness function. The cost-effectiveness function is defined as the difference in spatio coverage divided by its cost.

$$cost\_effectiveness(P, R) = \frac{\sum_{t_j \in T} |\bigcup_{p_i \in P}(r_{p_i,t_j}) - r_{p_k,t_j}|}{C(v_k)},\ k \neq i \tag{4.6}$$

This algorithm adds the most cost-effective participant and iterates until the budget constraint is met, and the algorithm has a time complexity of $O(|V|^2|T|)$. This is beneficial with a small number of participants sparsely deployed. In contrast, the second algorithm proposed is a genetic algorithm meant for a large number of densely deployed participants. The genetic algorithm encodes vehicle selection outcomes as a binary string. Thus, with 5 possible participants, if we only select the first participant, the encoding would be "10000". Initially, a large number of solutions are randomly generated. At each generation, the coverage is computed and only a top percent of the population survives. Furthermore, mutation and crossover operation occur as well to mimic evolutionary processes. Crossover combines two solutions to obtain a hybrid of the two. Mutation randomly changes part of the selection outcome. Finally, solutions which violate cost constraint are trimmed to fit. The algorithm runs until the time limit is reached or until the theoretical upper bound is reached. The main advantage of this algorithm is that it can be capped in terms of run-time which allows selection for a large number of participants; on the downside, it may result in weaker solutions.

In contrast to utilizing personal vehicles for public crowdsensing approaches, Gao et al. propose an air monitoring vehicular crowdsensing system using buses [168]. Unlike vehicles, buses have regular routes. Thus, their first base solution selects entire bus routes to achieve spatial coverage. However, such an approach

suffers from high costs given the need to install sensors on every bus. To expand on the first algorithm, the second algorithm reduces selected bus routes by introducing Points of Interests (POI) which are high priority sensing locations. These high priority areas could be schools, hospital or other public spaces. The priority between POI is considered to be equal. The sensing region R is split into 100m by 100m areas. Each area has an importance value $\delta$ attached, based on its distance to the nearest POI. Hence, an area close to the POI has a higher $\delta$ compared to an area far from any POIs. The importance value is shown in the following equation:

$$local(r_k, R, POI) = \begin{cases} \delta(r_k, R, POI) & > 2\, routes\, passing\, thru\, r_k \\ 0.75 * \delta(r_k, R, POI) & 1\, or\, 2\, routes\, passing\, thru\, r_k \\ 0.5 * \delta(r_k, R, POI) & >= 1\, route\, 1\, areas\, away \\ 0.25 * \delta(r_k, R, POI) & >= 1\, route\, 2\, areas\, away \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

Since air quality can be inferred from nearby measurements, coverage can be defined as the number of the routes passing within or near the area which is shown as the following:

$$coverage(V, R, POI) = \sum_{r_i \in R} local(r_i, V, POI) \tag{4.8}$$

Given the coverage function described above, the authors proposed an algorithm for finding the best bus route which is similar to the greedy algorithms mentioned in previous papers. The algorithm attempts to select the bus with the highest coverage and adds it as part of the solution. It iterates until the budget $B$ is reached. The algorithm's time complexity is $O(B * |V| * |R|^{0.5})$ and takes about 10 seconds to run with 1415 buses. To evaluate their algorithm, data was collected from a 2.9 x 3.1 $km^2$ city area in China. It consisted of 282 bus routes and 1415 bus schedule with 72 POI consisting of school and hospital locations. Their crowdsensing system works well for air quality monitoring but would be less appropriate when crowdsensing for other types of sensing data such as public safety monitoring.

One weakness of the current vehicular recruitment strategy is that once selected, a participant must continue sensing for a fixed duration. However, inefficiency exists; the participant selected may be only truly cost-effective for part of the time window $T$ selected. Furthermore, the trajectories provided by participants can be error-prone in reality.

To deal with uncertainty, Hu et al. propose a probabilistic method for estimating the location of a vehicle given a trajectory. The probability item was obtained from realistic vehicular traces. Let $prob(r_k, t_j, v_i)$ be the probability of vehicle $v_i$ to be at region $r_k$ at time $t_j$, thus $\sum_{k=1}^{|R|} prob(r_k, t_j, v_i) = 1$, which denotes that $v_i$ is located at one of the sensing region $r$ at any timestamp. Furthermore, let $V_j$ denote the set of recruited vehicles at timestamp $j$ and let $req_{r_k}$ denote the required number of vehicles for the $k$th sensing region, their spatial coverage is defined as the following:

$$coverage(r_k, t_j, P) = \sum_{x=req_{r_k}}^{|P|} \sum_{y=1}^{|x|} \prod_{p_i \in x} prob(r_k, t_j, p_i) \prod_{(p_i \in x) \cap P} [1 - prob(r_k, t_j, p_i)] \tag{4.9}$$

Where x represents the number of possible observation cases with x vehicles covering at the kth ROI. Thus, for all the target sensing regions and time points we have total spatial coverage as the follow:

$$SC(R, T) = \sum_{k \in R} \sum_{j \in T} coverage(r_k, t_j, P) \tag{4.10}$$

Moreover, the authors assume the true cost for each vehicle is available and define a cost matrix, where a cost is associated with each timestamp for each vehicle at a sensing region $r_k$. Thus, the cost function for a participant $v_i$ is defined as follows:

$$C(v_i) = \sum_{t_j \in T} C_{v_i, t_j, r_k} \tag{4.11}$$

The objective of the algorithm is to select a set of vehicles at a different time point which maximizes the coverage function while staying within budget $B$. Given

that the variable duration participant selection problem is a subset of standard vehicular participant selection, the problem is NP-hard. Thus, the first step is a pruning algorithm to remove nonviable participants while the second step is a greedy algorithm to finally select the vehicle and time for sensing.

The pruning step computes a Pearson correlation matrix to find vehicles with significant trajectory overlap. Vehicles with significant overlap are grouped together. The vehicle with the lowest average costs within the group will be preferred given they cover similar areas at similar times. In such, the algorithm can prune to only $|V'|$ participants. The step has a time complexity of $\mathrm{O}(|V'||V|)$.

In the second step, vehicles must still be selected at specific time. The algorithm seeks to iteratively recruit the best participant $v_k$ for each area $r_k$ at each time $t_j$. To find such a best vehicle, Hu et al. define the following profit enhancement efficiency function to rank participants.

$$efficient(R,T) = \frac{SC(R,T)^{(\ell)} - SC(R,T)^{(\ell-1)}}{C_{p_i,t_j}} \tag{4.12}$$

where, $SC(R,T)^{(\ell)}$ denotes the recruitment profit obtained in the $\ell$-th iteration.

To evaluate their algorithms, Hu et al. utilized a trace dataset from taxis in Rome over an area of 64 $km^2$ [169]. The duration of the dataset is over 30 days. Thus, the model proposed attempts to select participants at each $t_j$ during the time window. This tradeoff improves coverage metrics but suffers from higher computational costs. Furthermore, sensing data collected may lead to fragmented data from multiple sources; for instance, a single time window can only have at most certain vehicles covering in each area. This is a problem because of a lack of continuity and varying sensor quality. Compared to single continuous data, having multiple fragments of data increases the level of noise and reduces sensing quality.

### 4.1.1 Summary and Discussion

In this section, we provided an overview of existing participant recruitment solutions for large scale public vehicular crowdsensing. These works relied on full knowledge of vehicles' trajectory, which we considered not realistic. To deal with the trajectory uncertainty, one solution is using a real-time recruitment approach which suffers high cost. Other approaches rely on predicting vehicular position based on vehicular traces, and are more sound. For recruitment cost, existing work relies on monetary incentive mechanisms [170, 171]. They often assume each vehicle has a fixed cost plus additional coverage cost. Some studies used the auction procedure to determine the cost [172, 173]. However, such a procedure requires a significant level of data exchange between the server and vehicular participants. This is not only complex but lacks adaptivity to dynamically changing vehicular scenarios. For simplicity, most crowdsensing leverages a known posted pricing model, where the task owner can choose to recruit the vehicle or not based on the known cost information [128, 174, 175]. Finally, for the participant recruitment algorithm, these works use approximation greedy algorithms, since the public vehicular crowdsensing recruitment problem has been proved to be NP-hard.

## 4.2 Vehicular Participant Route Planning for Crowdsensing

Rerouting vehicles may increase travel time and consume extra energy [176]. A key problem is minimizing rerouting costs while maximizing the task completion rate by planning vehicular routes. Studies on route planning can be found in various research topics such as modern game AI, automated transportation system, or multi-robot planning [177, 178]. Route planning algorithms can be classified into single-agent path planning and multi-agent path planning [179]. Single-agent path planning has been broadly studied since the mid-twentieth century. Algorithms such as Dijkstra's algorithm and A* algorithm [180] are widely used in path-finding or graph traversal problems in modern research. Although these algorithms can route a single agent to its destination, they cannot be directly applied to the multi-agent path-finding problem because each agent may conflict with each other when

they move from their start positions to their unique goals [181].

The conflict issue has been solved in two ways. One solution is to let each moving unit decide its path independently but enable each unit to communicate to avoid a collision. [152] proposed a multi-robot patrol system that aims to minimize the average robot traveling time for all nodes. The system uses the Bayesian learning strategy to select the next visiting node for each robot, and communication between robots is assumed to prevent collisions. The second solution is to plan all robot moves in a centralized manner. However, finding an optimal solution for the multi-robot path planning problem is NP-hard [182]. Other approaches solve global planning by using prioritized planning. Prioritized planners compute each robot's path based on their priority. That is, high priority robots are scheduled first and are considered to move without obstacles from low priority robots [183, 184]. In [185], they considered reducing search space-time complexity by decomposing the entire road-map into a grid-world. [186] takes advantage of both priority planning and windowing the search space by partitioning the robot route into a set of partial routes.

One approach for multi-agent path-finding is to find the shortest path solution for every agent while avoiding collisions. However, our objective is maximizing sensing while constrained by the need that agents must arrive at their final destination within an accepted time range. Furthermore, in our multi-vehicle planning approach, we cannot stop a vehicle in the middle of the street to gather all data necessary, unlike other multi-agent exploration or planning work. This leads to an important planning problem we need to solve; a single sensing task may require multiple agents to complete given each agent can only spend limited time at a sensing location. Furthermore, each road has driving directions and speed limits which must be respected. Unlike in the robotic system where speed can be assumed to be uniform, speeds on roads can vary widely from 25 km/h to 120 km/h. These requirements are not addressed in traditional multi-agent path-finding research.

To the best of our knowledge, only a few works have studied the route planning problem for vehicular crowdsensing. For instance, [128] proposed an auction-like route planning solution to complete the sensing task. In this solution, each vehicle generates multiple routes as a bid while a central server selects which route each vehicle should undertake to maximize the global sensing task completion rate. In their algorithm, vehicles are rerouted when there is a better path nearby. However, vehicles in some constrained roads such as the highway are difficult to reroute. Thus we need to predict where sensing needs will occur before sending sufficient participants to the location. The main weakness of this paper is that rerouting is only locally greedy; vehicles only reroute to a position which immediately requires sensing. Furthermore, vehicles do not look beyond a short time window for a location to reroute to; regions which require sensing but are distant are ignored. This can lead to a problem where when requests appear and not enough agents are available for sensing.

# Chapter 5

# Opportunistic Participant Recruitment

In this chapter, we will introduce our solutions for personalized vehicular crowd-sensing participant selection problem. Compared to other works, PVC handles dynamic sensing tasks; that is, tasks whose sensing area dynamically changes over time. Similar to other works, we assume the participants cannot be rerouted for better sensing; that is a participant can only gather data passively along to its original route. In the following sub-section, we show an overview of the system model and our algorithms in detail. The notations used, specifically in this chapter, can be found in the Table 5.1

## 5.1 System Model and Problem Statement

A PVC system is comprised of cloud servers and a massive number of smart vehicles. We assume they are equipped with sensors and communication devices such as Wi-Fi and cellular interface. Once a vehicle begins its journey, basic information such as unique vehicle ID and predicted trajectory from the navigation system are continuously uploaded via wireless networks and stored within a PVC server. Clients can make a query for sensing data of interest from the PVC server. The query specifies the task sensing regions and it's duration. The server selects other vehicles as sensing participants within the targeted monitoring regions and time constraints. On the participant's side, the participant receives the sensing request

| Notation | Description |
|---|---|
| $R$ | Set of road segments. |
| $V$ | Set of vehicles. |
| $C$ | Set of clients, $C \in V$. |
| $P$ | Set of vehicular participants, $P \in V$. |
| $R_c$ | The query route segments of the client $c \in C$. |
| $R_p$ | The predicted future road segments of the participant $p \in P$. |
| $\mathcal{T}(r, R)$ | The function to obtain the timestamp of a road segment $r$, given a set of road segments. |
| $R_{c,p}^{com}$ | The set of common road segments, obtained from $R_c \cap R_p$. |
| $\delta$ | Lower bound of moving monitoring window (MMW). |
| $\varepsilon$ | Upper bound of MMW. |
| $\Gamma(R_c, R_p, r)$ | Value function returns 1 if the data timeliness constraints are satisfied and 0 otherwise, given a road segment $r$. |
| $single\_cover(R_c, p)$ | The function returns set of road segments such that $\forall r\ \Gamma(R_c, R_p, r) = 1$. |
| $\ell oad_p^{max}$ | The maximum workload the participant $p$ will have for entire journey. |
| $\beta_c$ | Maximum number of vehicles that the client $c$ can hire for doing the sensing task. |
| $cap_p$ | Soft threshold of maximum number of tasks that $p$ is willing to serve simultaneously. |
| $SW$ | scheduling window for window based algorithm. |

**Table 5.1** List of Notations for chapter 5

ahead of time, and the task is stored in memory in pause mode. When the vehicle reaches the sensing location, the sensing task switches to active mode until the vehicle leaves the sensing location or timeouts. The detailed flow for a task in our PVC system is depicted in Figure 5.1.

Sensing tasks consume multiple resources including computation, communication, and energy of the vehicle. Furthermore, collected data can contain location information, which may lead to significant privacy and security issues [187, 59, 188, 189, 190]. We assume that participants are willing to contribute to sensing tasks because they are sufficiently motivated. We suggest a price-based incentive mechanism due to its popularity in crowdsensing research [114].

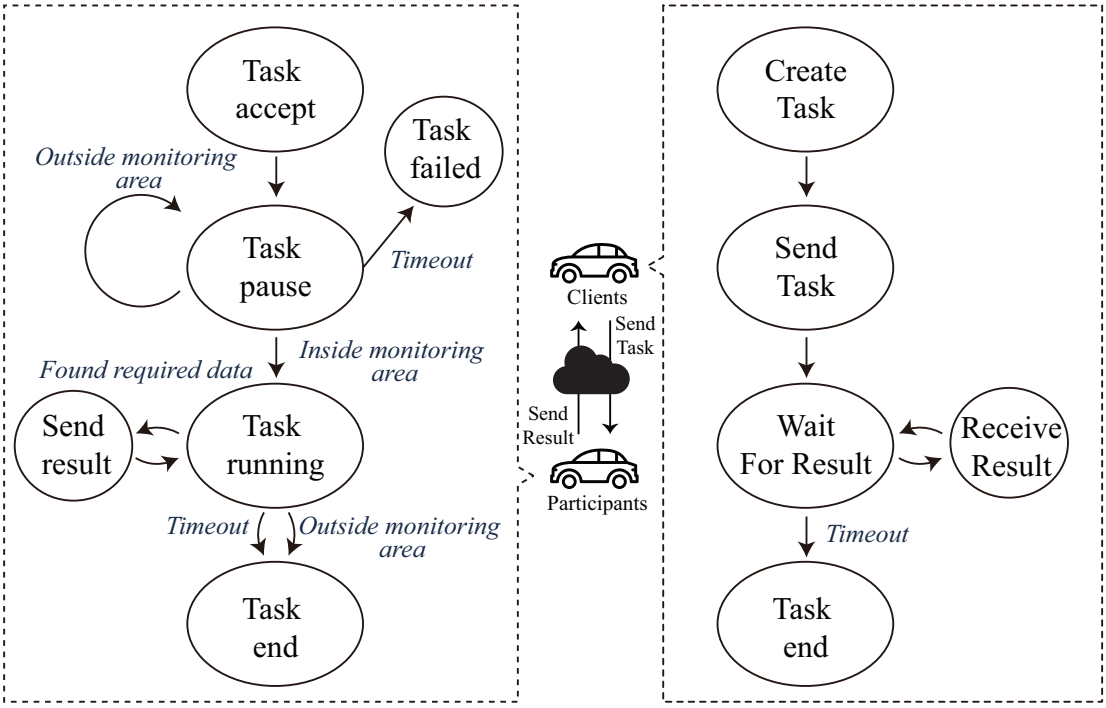


**Fig. 5.1** Task flow

We define an area of interest to be divided into several road segments $R := \{r_1, r_2 ... r_m\}$. Each road segment has a single traffic direction as shown in Figure 5.2a. Roads that have opposite traffic directions are considered two different road segments. The area also contains a set of vehicles $V$. Let $C \subseteq V$ be the set of clients and $P \subseteq V$ be the set of participants whereby clients can also be participants.

The client sensing route and participant projected route can be constructed into a trajectory graph as shown in Figure 5.2b. In this graph, a route is composed of a sequence of road segments. Each road segment contains a timestamp specifying arrival time of the vehicle. Let $R_c$ be the set of query route segments of the client $c$ and let $R_p$ be the set of predicted future road segments of a given participant $p$. For each vehicle $v$, the arrival timestamp of a given road segment can be derived from the following function: $\mathcal{T} : r \times R_v \rightarrow \mathbb{Z}^+$

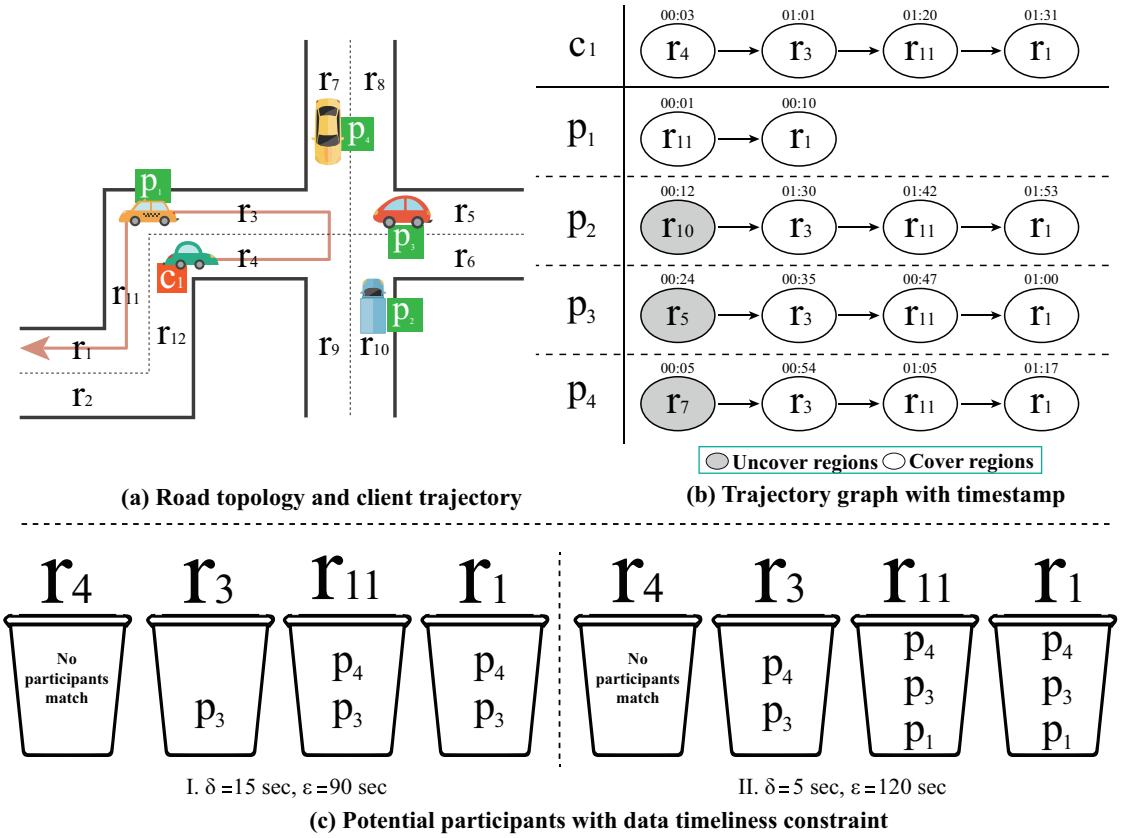


**Fig. 5.2** This figure shows a task with a total of 4 vehicular participants on the map. The timestamp in b) for each road segment signifies the arrival time of the vehicle on the road segment. In c) $\delta$ and $\varepsilon$ are the threshold for the timely participant; Filtering potential participants under specific monitoring time window.

To select valid and suitable participants for a specific client, we use the following definitions:

**Definition 1.** ***Data timeliness:*** *A sensing task must be completed at the specified location before a deadline to be useful to the client. Therefore, data that arrive too early or too late is worthless. Thus, participants need to be within a sensing window to provide meaningful sensing data. We refer to such a window as moving Monitoring Window (MW) which is depicted in Figure 5.3. Let $\delta$ and $\varepsilon$ be the lower bound and upper bound for the window, respectively. The value of a*

*participant at route r is represented as follow:*

$$\phi(R_c, R_p, r) = \begin{cases} 1 & \delta \leq \mathcal{T}(r,c) - \mathcal{T}(r,p) \leq \varepsilon, \\ & r \in R_c \cap R_p \\ 0 & otherwise \end{cases} \tag{5.1}$$

For instance, in Figure 5.2cI, participant $p_4$ cannot become a potential participant for road segment $r_3$ for sensing task because $\mathcal{T}(r_3, R_{c_1}) - \mathcal{T}(r_3, R_{p_4}) = 7$; less than the threshold $\delta = 15$. Participant $p_1$ also cannot be a participant for road segment $r_{11}$ and $r_1$ as the time difference between the client sensing route and participant route is greater than the threshold $\varepsilon = 90$. However, $p_4$ could become a potential participant for $r_3$ when the value of $\delta$ changes to 5 seconds as shown in Figure 5.2cII. Similarly, $p_1$ could become a participant candidate for both $r_{11}$ and $r_1$ when the value of $\varepsilon$ changes to 120 seconds. Thus, the number of potential participants may increase when the size of MW is increased.

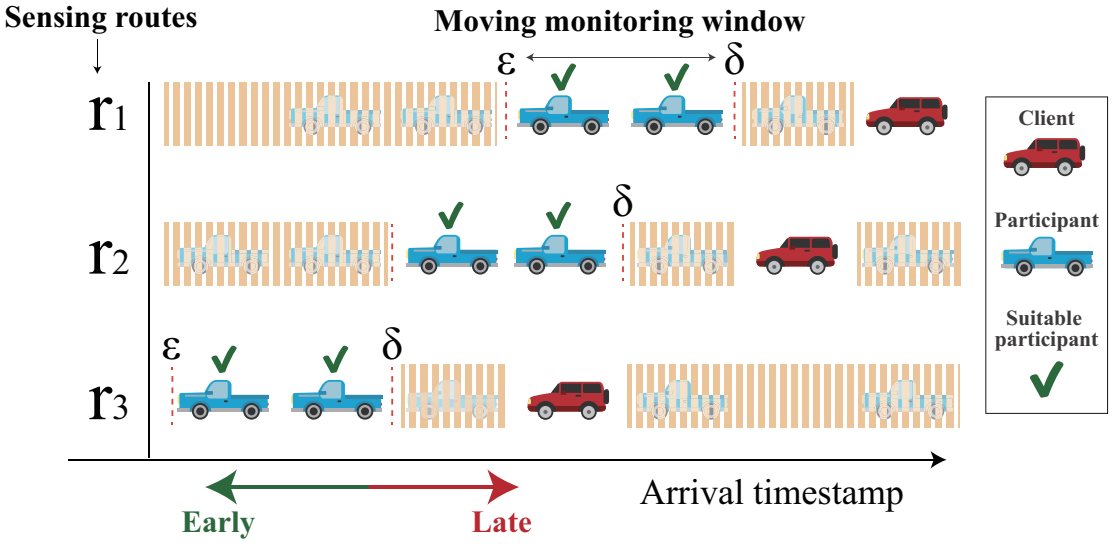


**Fig. 5.3** The moving monitoring window is a window for selecting useful sensing participants. For example, a client requests sensing information at least one minute ahead of its current path, but not more than 5 minutes ahead $\varepsilon = 300$ *second*.

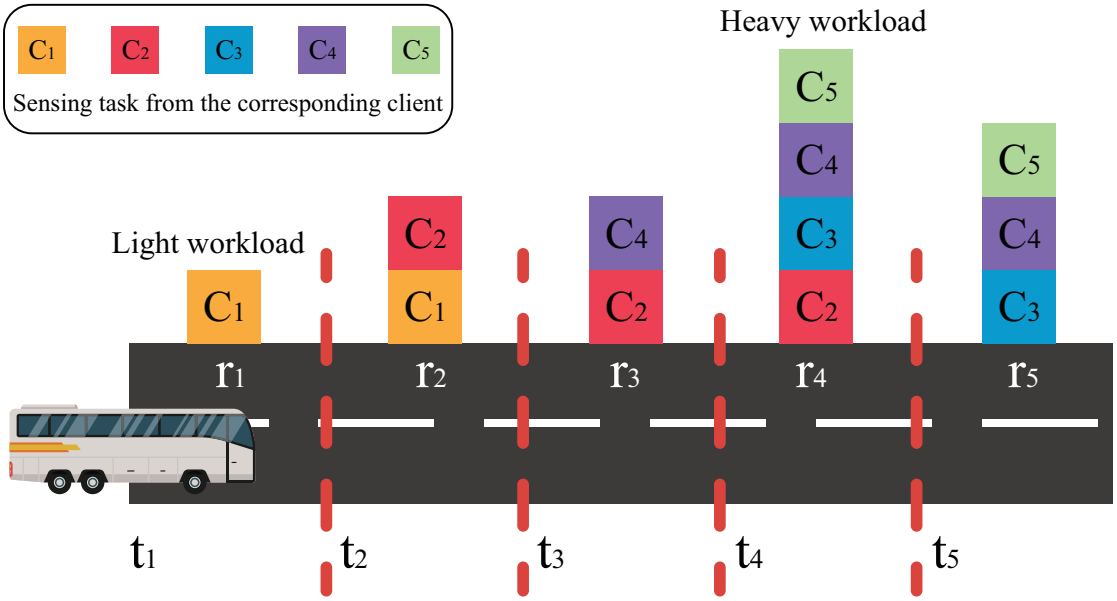


**Fig. 5.4** A sample workload for a single participant with 5 clients is shown; depending on the client's location of interest, the workload is spatio-temporally assigned.

**Definition 2.** ***Query route coverage:*** *We define query route coverage as the set of road segments a set of selected participants can cover for a single client within the associated timeliness constraint. Given a client, $c \in C$, and a participant, $p \in P$, a single participant coverage for a single client can be defined as the following $\gamma$ function:*

$$\gamma(c, p) = \{r \in R_p | \phi(R_c, R_p, r) = 1\} \tag{5.2}$$

*Thus, given a set of participants $S \subseteq P$, the function $\zeta$ for a query route coverage for a given client $c$ can be formally defined as the following:*

$$\zeta(S) = \bigcup_{p' \in S} \gamma(c, p'), \text{ where } |\zeta(S)| \leq |R_c| \tag{5.3}$$

**Definition 3.** ***Participant maximum load:*** *Participants will be assigned sensing tasks by different clients along its journey. While the number of requests increases, the peak workload increases drastically. We define the peak workload as the maximum workload of a participant along its sensing route. Figure 5.4 shows the peak workload* $\ell oad_p^{max} = |\{c_2, c_3, c_4, c_5\}|$ *of a single participant. This occurs when a vehicle is traveling through a sensing area requested by multiple clients where it could be selected for multiple simultaneous tasks. Thus, a massive amount of sensing requests continuously assigned to such participants would overload it and result in failed tasks.*

**Definition 4.** ***Objective function:*** *The objective function of the PVC participant recruitment problem is to recruit a subset of vehicles* $S' \subseteq P$ *maximizing coverage while subject to a budget constraint* $\beta_c$ *and maximum load cap of the participant* $cap_{p'}$*.*

***Objective:*** $S' \leftarrow \arg\max_S \ \zeta(S)$

***Subject to:*** $S' < \beta_c; \qquad \forall p' \in P, \ \ell oad_{p'}^{max} \leq cap_{p'};$

## 5.2 Problem Hardness

In this section, we show that the well-known NP-hard Set Cover problem can be reduced to the Personalized Vehicular Crowdsensing Participant Recruitment (PVC-PR) problem, ie. Set-Cover $\leq_z p$ PVC-PR. We follow the steps from [191] in proving a problem to be NP-complete.

**Theorem 1.** ***PVC-PR problem is NP Complete:*** *The decision version of PVC-PR problem is described as given a client query routes* $R_c$ *with the budget limit* $\beta_c \in \mathbb{Z}^+$*, and list of vehicular participants* $P$ *and their future routes, does there exist a recruitment* $P' \subseteq P$ *such that* $\bigcup P'$ *is equal to specific coverage of* $R_c$*, while all PVC-PR constraints are satisfied?*

*Proof.* To prove PVC-PR problem is NP: Let $P' \subseteq P$ be the solution for the PVC-PR problem, we can verify the correctness of it in polynomial time in which the complexity is $O(|P'| \times |R'|)$ where $R'$ is the corresponding future route segments. □

*Proof.* To prove PVC-PR problem is NP-complete: The Set Cover problem is a NP-hard problem which is described as: *given a universe* $U = \{u_1, u_2, ..u_n\}$*, a collection of* $S = \{S_1, S_2, ..S_m\}$*, where each* $S_i$ *consists of elements of* $U$ *and* $\bigcup S = U$*, and a positive integer* $k \leq m$*, does there exist a subset* $\hat{S} \subseteq S$*, such that* $\bigcup \hat{S} = U$ *and* $|\hat{S}| \leq k$ *?*

The polynomial time reduction step of constructing a PVC-PR instance is described as follows. We first make sure the list of participants is able to cover all query route segments, $\bigcup R_p = R_c$. With the data timeliness constraint described in definition 10, there exists a possibility a temporal route segment cannot be covered by any vehicle. Thus, we construct a new query set, $R'_c$, where $\left\{R'_c \subseteq R_c | \ \exists_{v \in P} \wedge \exists_{r \in R_p} \ \phi(R_c, R_v, r) = 1\right\}$. The process of filtering is done in polynomial time in which the complexity is $O(|R_c| \times |R_p|)$. Let each vehicular participant $R_{p'} \in R_P$ correspond to a subset $\hat{S} \in S$, each road segment $r$ in $R'_c$ corresponds to the $u \in U$, and the limited budget $\beta_c$ corresponds to $k$. □

Let both $\beta_c$ and $k$ be the same positive integer number, we claim that the Set Cover problem has solution, $\bigcup \hat{S} = U$, if and only if the instance of PVC-PR has a solution $\bigcup R_{\hat{P}} = R'_c$.

## 5.3 Push Based Solution for PVC-PR problem

We proved that PVC-PR is NP-complete by reducing the set cover problem to the PVC-PR. As such, our work GoSense, used the Longest Cover First (LCF) greedy approach for selecting participants [21]. In it, we continuously choose the participant $p^{best}$ that contains the largest number of uncovered road segments until no road segment is left uncovered.

**Definition 5.** ***Longest Cover First (LCF) heuristic function:*** *We select the participant $p^{best}$ that can cover the longest continuous task segment under the specified data timeliness constraint.*

$$p^{best} \leftarrow \arg\max_{p' \in P} \sum \phi(R_c, R_{p'}, r) \tag{5.4}$$

*We continue recruiting vehicular participant $p^{best}$ until all segments of the query routes are covered. If there exists no satisfiable participant who can cover the remaining query route, the recruiting process will terminate and return a list of selected participants so far. A detailed pseudocode is shown in Algorithm 1.*

**Algorithm 1** Longest Cover First Algorithm (LCF)

**Input** c, P
**Output** *Participants*
1: $Participants \leftarrow \emptyset$
2: $c' \leftarrow c$
3: **while** $R_{c'}$ is not empty **do**
4: $p^{best} \leftarrow \arg\max_{p' \in P} \sum \phi(R_{c'}, R_{p'}, r)$
5: **if** $p^{best}$ is empty **then**
6: No valid participants, break the loop
7: **end if**
8: $R_{c'} \leftarrow R_{c'} \setminus R_{p^{best}}$
9: $Participants \leftarrow Participants \cup \{p^{best}\}$
10: **end while**

LCF is a polynomial-time algorithm; the worst case occurs when each participant only covers one road segment of the querying road. This cannot be greater than $R_c$, thus $O(|R_c|)$. Furthermore, in each round we must search at most $P$ participants, requiring the complexity $O(|P|)$. Thus, the coverage for a sensing task

yields an overall time complexity of $O(|R_c| \times |P|)$. LCF can select participants to cover the sensing route in polynomial time; however, it suffers from load balancing problems. Given the massive amount of sensing tasks, we must consider the participants' workload which can lead to task delay and failure. Thus, to ensure our solution can maximize coverage for requested sensing routes as well as spread workload, we use the following score function for participant selection.

**Definition 6.** ***Workload score function:*** *To select participants to meet load balance, our workload score function $\omega : \ell oad_p^{max} \rightarrow \mathbb{R}$, which considers participant's current maximum workload, is defined as the follow:*

$$\omega(\ell oad_p^{max}) \leftarrow \frac{cap_p}{\ell oad_p^{max}}; \quad \forall\ \ell oad_p^{max}, cap_p \in \mathbb{Z}^+ \tag{5.5}$$

*Let $\ell oad_p^{max}$ indicate the maximum workload the participant $p$ will have for the entire journey. Let $cap_p$ be the soft threshold of maximum number of tasks that $p$ is willing to serve simultaneously. The motivation behind this score function is to enforce a soft penalty upon participants to spread workload even if a vehicle is below its task capacity. This ensures fairness for allocation tasks such that all vehicles have a chance of participating and receiving rewards in contrast to a winner take all approach where only the best participants receive all tasks.*

The pseudocode detailed in Algorithm 2 shows how our score function is used for recruiting participants while considering load balance. In the WB algorithm, a participant is selected in each round, where the number of iterations is capped by the size of the query routes $|R_c|$. The vehicle selection decision is based on the vehicle's weight $w$. The weight of the participant $p'$ is calculated by its workload score times the size of its route coverage for the client's query route: $w \leftarrow \omega(\ell oad_{p'}^{max}) \times |\gamma(c, p')|$.

We select a vehicle with the maximum weight $w^{max}$ as shown in algorithm 2 from line 6 to line 13. In terms of run time complexity, we require extra computation for handling the workload update which requires total complexity $O\left(|R_c| \times (|P| \times |R_p| + |R_p|)\right)$. However, since $|R_p| < |P| \times |R_p|$, the complexity of WB algorithm can be simplified to $O(|R_c||P||R_p|)$.

**Algorithm 2** The Weight Based Greedy Algorithm (WB)

**Input** c, P
**Output** *Participants*

$Participants \leftarrow \emptyset$
$c' \leftarrow c$
**while** $R_{c'}$ is not empty **do**
  $p^{best} \leftarrow \emptyset$
  $w^{max} \leftarrow INT\_MIN$
  **for** $p\prime \in P$ **do**
    $\hat{R} \leftarrow \gamma(c', p')$
    $w' \leftarrow \omega(\ell oad_{p'}^{max}) \times |\hat{R}|$
    **if** $w' > w^{max}$ **then**
      $w^{max} \leftarrow w'$
      $p^{best} \leftarrow p'$
    **end if**
  **end for**
  **if** $p^{best}$ is empty **then**
    No valid participant, break the loop
  **end if**
  $R_{c'} \leftarrow R_{c'} \setminus R_{p^{best}}$
  Update workload of $p^{best}$
  $Participants \leftarrow Participants \cup \{p^{best}\}$
**end while**

WB solution assumes the availability of flawless predictions of vehicles' paths. Such an assumption is not realistic given high prediction errors. As discussed in the previous section, the prediction error is dependent on traffic conditions. In light traffic, the prediction error increases slowly as the length in time of the trip increases, whereas prediction error grows quickly in heavy traffic. We solve this issue by proposing a windowed based scheduling approach where we only schedule participants for set time windows instead of whole trips. We set the initial scheduling window, $SW_{min}$, to 300 seconds for each sensing task, since errors for predicted arrival times within trip length of 300 seconds are acceptable in both light and heavy traffic as shown in Figure 3.5. However, a static window size suffers from too frequent scheduling, resulting in computation resources waste and increased delays. For instance, in light traffic, a 1200 second window may be a

better choice for the scheduling window instead of 300 seconds. Thus, we proposed a Dynamic-Window Based (DWB) solution.

**Algorithm 3** Dynamic Window Based Scheduling (DWB)

1: $\mathcal{P} \leftarrow$ List of participant candidates
2: $\mathcal{C} \leftarrow$ List of the client's requests
3: **for each** $c \in \mathcal{C}$ **do**
4: $R_c \leftarrow \{R' \mid \forall_r \mathcal{T}(r, c) < t^{current} + SW_c\}$
5: ▷ SW is calculated based on Equation 5.6
6: ▷ The $t^{current}$ means current time
7: $S \leftarrow WB(R_c,\ \mathcal{P})$ ▷ Based on algorithm 2
8: $Expected\_cover_c \leftarrow |\zeta(S)|$
9: Notify the client $c$ and each $p \in S$
10: **end for**

As shown in algorithm 3, DWB only schedules participants for serving a client's sensing task within a specific scheduling window $SW$. The size of $SW$ is calculated as the following:

$$SW_c \leftarrow \begin{cases} cover\_rate < 1 & \text{MAX} \begin{cases} SW_c \times cover\_rate_c \\ SW_{min} \end{cases} \\ \text{otherwise} & SW_c \times \theta \quad \text{where } \theta \in \mathbb{R}^+ \end{cases} \tag{5.6}$$

$\theta$ is a positive multiplier that controls how fast $SW_c$ grows. Note that to predict the next $SW$ size, we need to know the coverage performance of the previous $SW$. In such case, we assume that the client calculated the coverage rate before submitting the next rescheduling request, where the coverage rate is calculated as follows:

$$cover\_rate_c \leftarrow \frac{Actual\_cover_c}{Expected\_cover_c} \tag{5.7}$$

Each client has a personalized scheduling window due to varying driving behaviors. The scheduler continuously adjusts the size of $SW$ until the corresponding sensing routes are fully scheduled. The flow of DWB scheduling is shown in Figure 5.5.

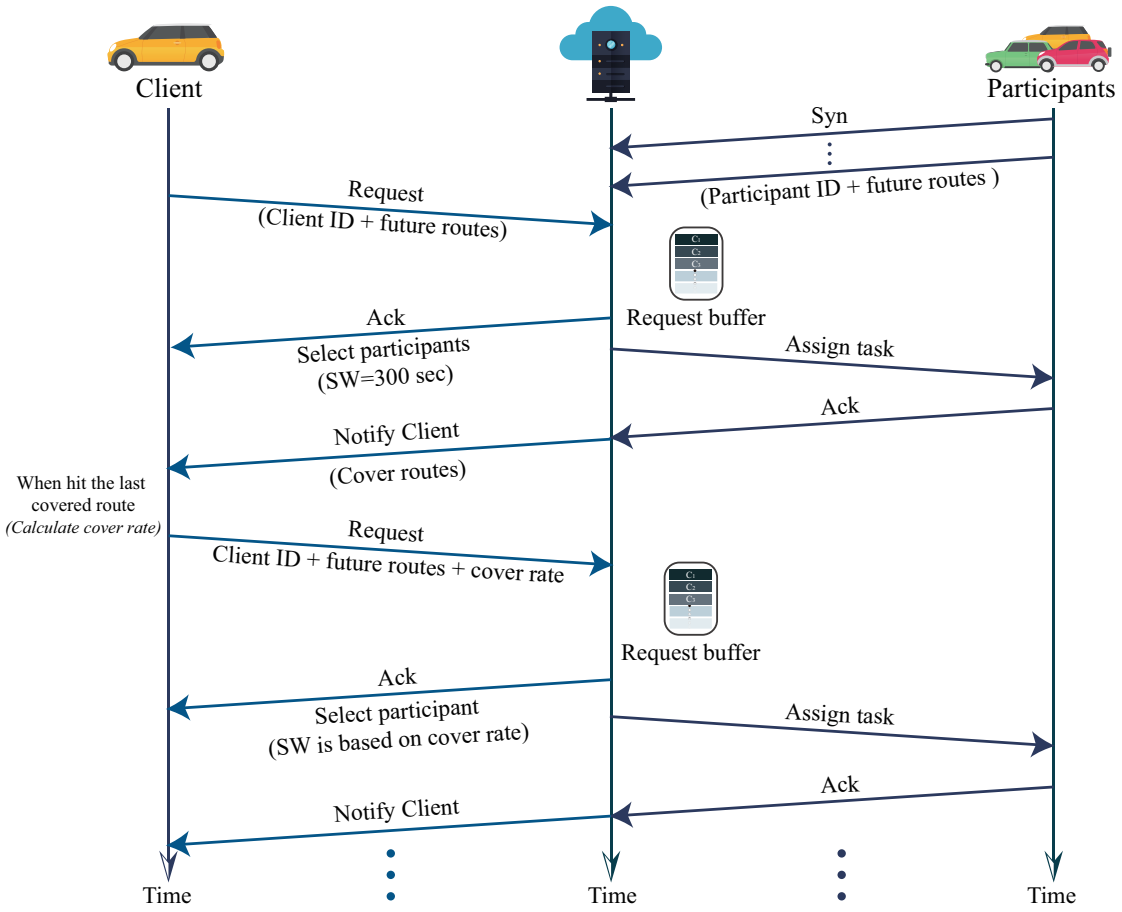


**Fig. 5.5** Dynamic window based scheduling workflow.

## 5.4 Pull based solution for PVC-PR problem

Push based participants recruitment approaches are popular in vehicular crowdsensing research. This is primarily because the trajectory of vehicles is predictable due to the ubiquitous use of a navigation system while driving. As the estimated positions of vehicles are known, the system can select vehicles whose future route maximizes the overlap with the client's sensing path. However, traffic conditions are often mercurial due to incidents such as road accidents, roadblocks, bad weather, or change of plans. We define two types of prediction errors. The first type of error occurs when participants reach the expected location but not on time. The second type of error occurs due to route adjustments; participants will not

arrive at the expected location. In push-based approaches, we solve such errors by window-based scheduling as we discussed in the previous section. However, deciding a size of the scheduling window is a crucial problem. Small windows will increase scheduling overhead, whereas large windows will increase subtask failure rates. Besides, canceling tasks in a push-based solution requires sending commands to each assigned participants which require extra processing overhead.

Besides the push-based approach, we believe that a pull-based system can preempt the issues above; participants can pull sensing subtasks in real-time which can respond to dynamic traffic conditions. One problem we face is that this may lead to a large number of participants per task. This not only increases the recruitment cost but may also lead to lower sensing quality due to noise and differences in sensors between participants. Hence, a key challenge to the design of a pull-based recruitment approach is how to minimize participant changes required for a sensing task. We solve this challenge by introducing a new pull system, Best Offer First (BOF), which does not only rely on First Come First Pull (FCFP) basis. Instead, participants can submit sensing offers for the same task to a server during a given time window. The server then selects the best participant from the participant candidate pool according to the following rules:

- A participant with the longest route overlap with the request will be selected.
- Participants who previously serve the same client will be prioritized over those who were not previously selected.

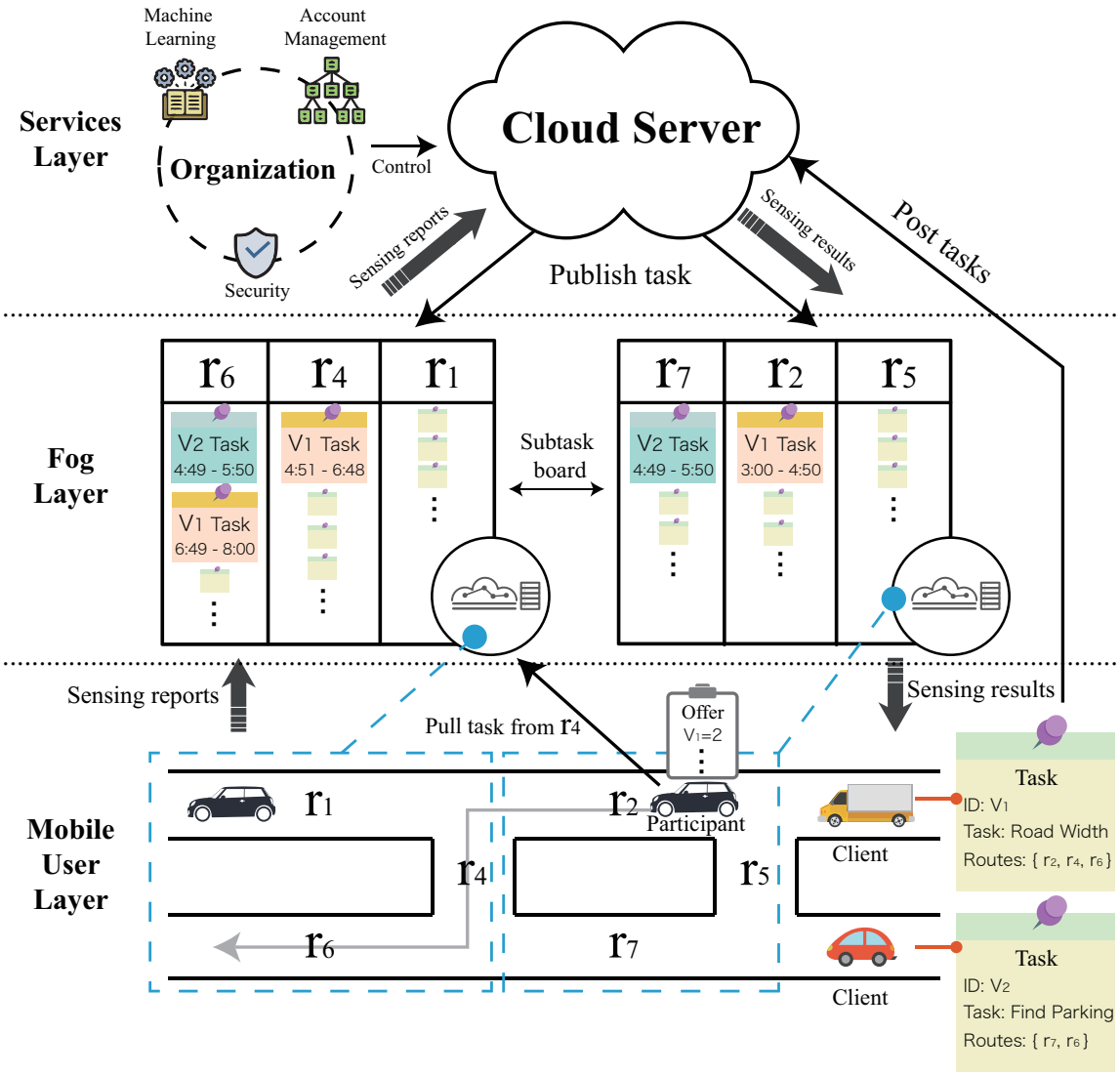


**Fig. 5.6** Architecture of Personalized Vehicular crowdsensing.

## 5.5 Assumption and System Design

In our Pull Based Participant Recruitment System (PBPRS), we assume road networks are covered by distributed fog servers. Each fog server acts like a cluster head, managing sensing tasks for a group of road segments. Furthermore, unlike the push-based system, a participant in PBPRS does not need to transmit its future route $R_p$ to the server. Participants interested to sense for a particular road segment can pull the tasks themselves from the corresponding fog server via the wireless network. We use the same price-based incentive mechanism, that is, the participant receives payment only when the assigned sensing task is completed.

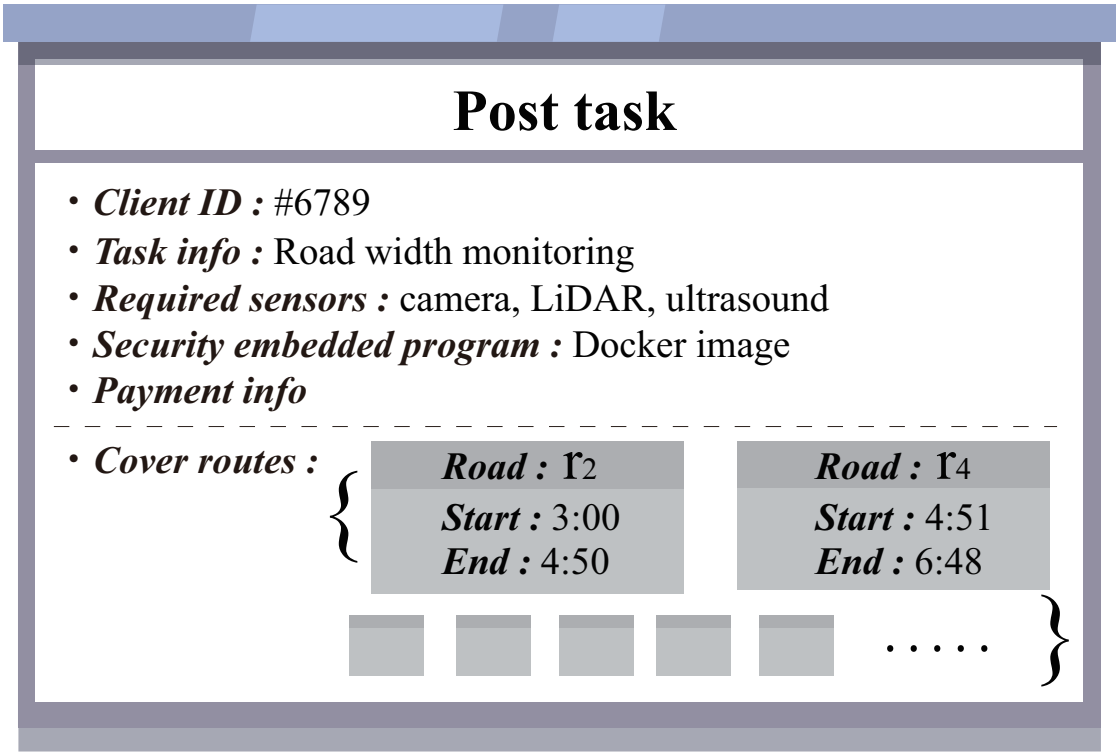


**Fig. 5.7** Sample Task; each task includes requirements, program and road segment requested.

The architecture of PBPRS is shown in Figure 5.6. When a vehicular client submits sensing tasks to the central cloud server, the task contains client vehicle ID and task program. Additionally, the query needs to specify the future sensing route, $R_c$, and duration as shown in Figure 5.7. After receiving the sensing tasks,

the central server publishes a subtask for each route segment in $R_c$ for each task to the distributed fogs. Each subtask has two values that denote subtask sensing starting time and end time respectively. These subtasks are kept on the subtask board in fogs nodes. The subtask will be removed if the sensing end time is reached, if the subtask has been pulled by a participant, or if the client cancels it due to a change of plans. Participants pull sensing subtasks according to their interest and capacity. Furthermore, each subtask has its hiring window, $\lambda \in \mathbb{Z}^+$ ahead of the subtask starting time within which participants are allowed to pull subtasks. Note that subtasks are revealed to participants only during the subtask hiring period.

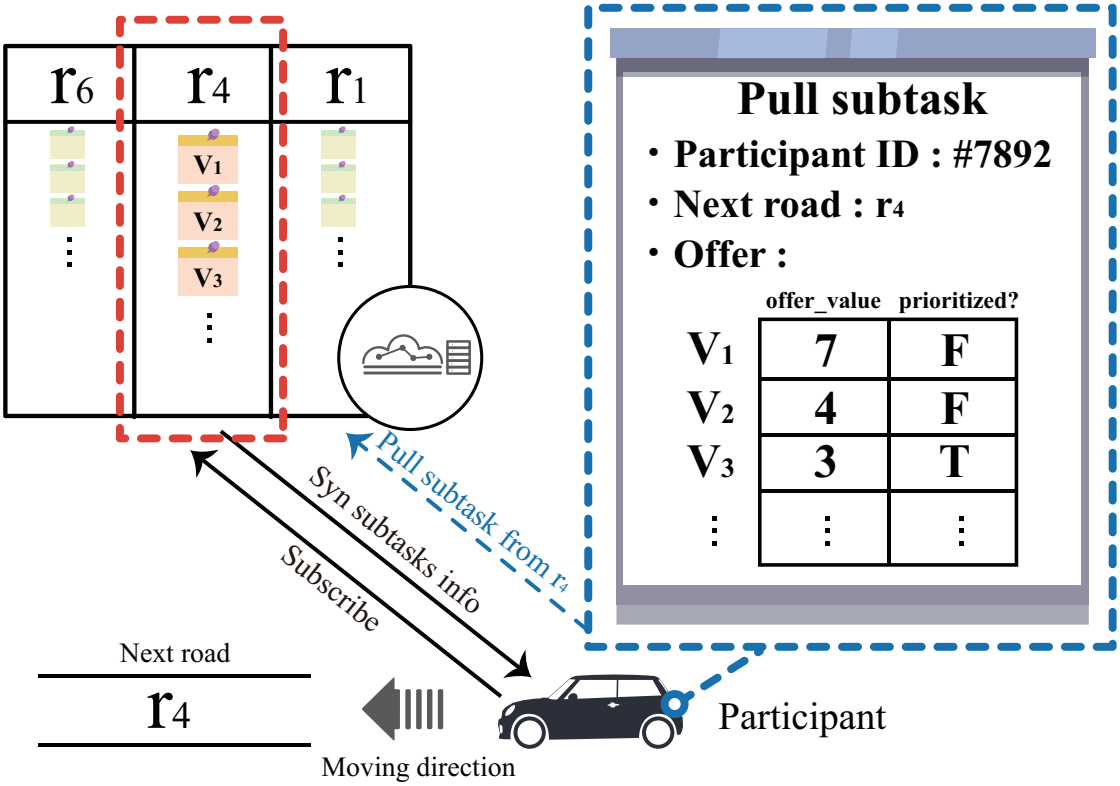


**Fig. 5.8** A participant pulls from the local subtasks for the next road segment it will reach. If selected, it will perform the sensing task only for the next road segment.

During the subtask hiring period, any participants may pull for the same task. Since we only select one participant per subtask, a participant selection rule is required. Participants who wish to serve a specific client need to provide in the offer how many future road segments the participant could cover for the client as seen in Figure 5.8. Each subtask on the subtask board contains information about the remaining uncovered road segments. Let $\bar{R_c} \in R_c$ be the remaining uncovered road segments of the client $c$, the $offer\_value$ is calculated as follows:

$$offer\_value(R_p, \bar{R_c}) = |\left\{r \in R_p | \phi(\bar{R_c}, R_p, r) = 1\right\}|$$

Each offer value is calculated locally on the participant. The value has its own expiration time which depends on when the participant enters the next road segment.

In BOF, when the subtask deadline is reached, the server will select the participant with the highest valid offer value. The participant can provide multiple offers for different subtasks to maximize the chance of being selected. Once the participant is selected, a confirmation notice will be sent to the participant and it will make the final decision to accept the subtask. If the participant rejects the subtask, the participant next in line will be offered and this repeats until a participant is selected or no participants are available.

Besides selecting a participant according to their offer value, the system also prioritizes participants who have previously served the same task. When the prioritized participant submits the offer for a subsequent subtask, he will be immediately selected as the participant regardless of its offer value. Participants will lose priority if they do not continuously serve the same client. The selection flow is depicted in Figure 5.9.

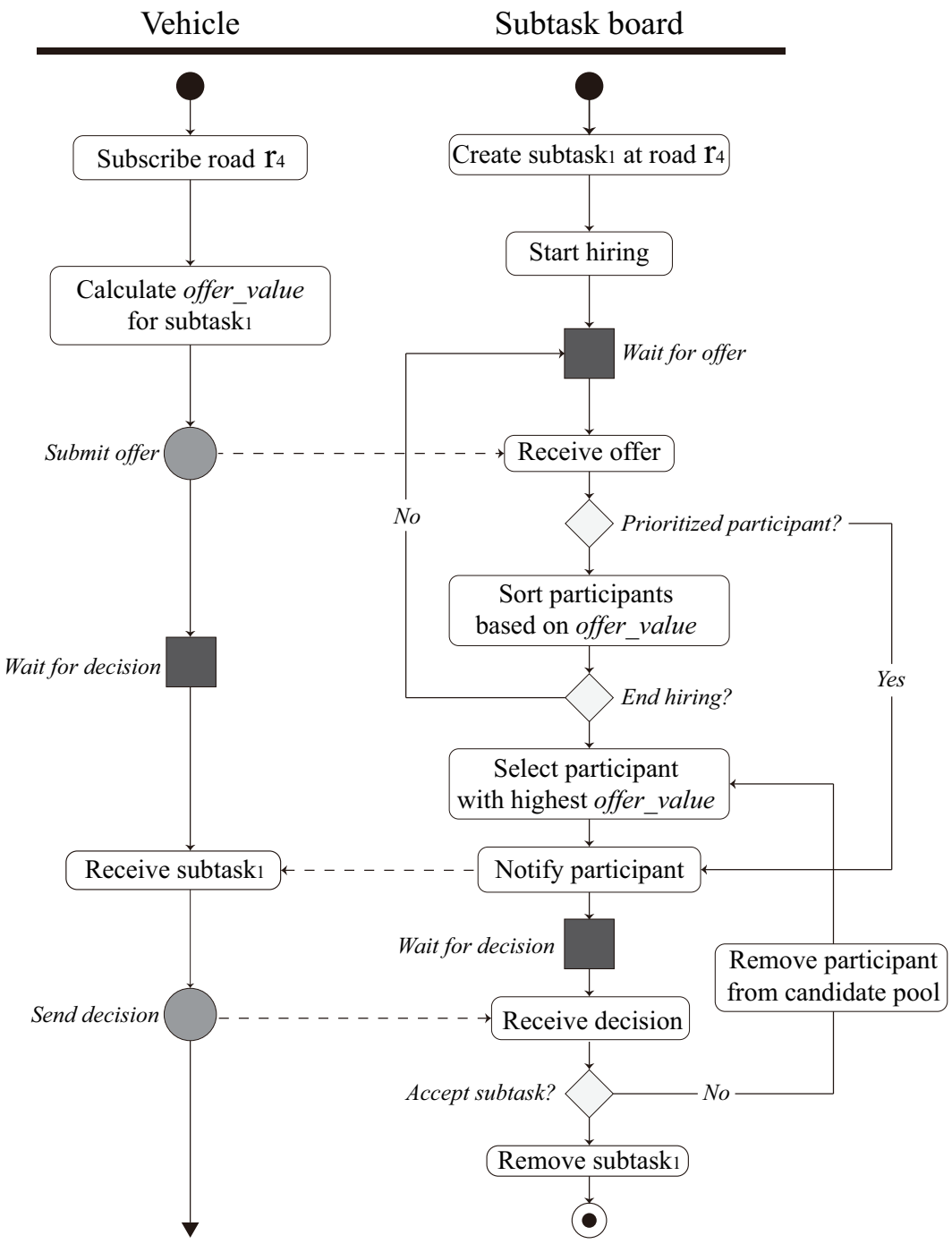


**Fig. 5.9** Pull subtask flow chart.

For fault tolerance, we assume that subtask boards are well replicated in the cloud. Thus failures on fog nodes would still enable us to retrieve tasks information.

In addition, PBPRS allows clients to cancel their request any time during their journey. That is, if clients require adjustment due to roadblocks, delay, or accident then the task will be canceled and new subtasks will be posted based on the new route. Such delete operations occur on the subtask board which avoids the necessity of deleting tasks from participants. A problem we may face is that $offer\_value$ provided by participants is calculated based on the future position of participants (the $R_p$). Prediction error on the $R_p$ will cause the participant to fail to serve the client. Thus, we only allow participants to pull tasks for the next road segment in their route. For subsequent road segments, the participant needs to repeat the pull process even if it is a prioritized participant. Thus, a vehicle with errors in its predicted route will be unable to pull the following request. Even though it may take more vehicles, a task can always be completed.

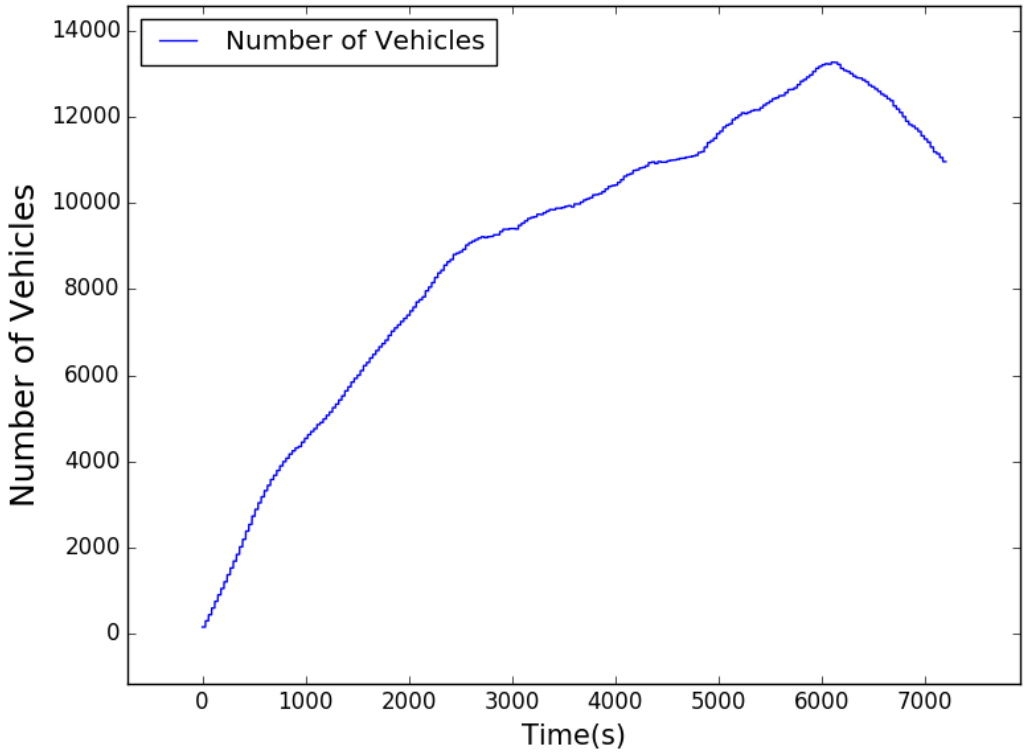


**Fig. 5.10** Number of vehicles active in TAPAS Cologne simulation from 6 am to 8 am, in seconds(s)

## 5.6 Evaluation

To showcase the performance of our approaches, we used TAPAS Cologne, one of the largest realistic traffic simulation datasets. It consists of data simulated over 24 hours for the city of Cologne, Germany. It covers over 900 square kilometers [166]. To reduce simulation time, we restricted ourselves to a data subset which consists of traces of 7200 seconds from 6 AM to 8 AM. It should be noted that between 6 to 8 am more vehicles are on the roads than during other time periods; this can be seen as a morning commute. The number of vehicles on the road can be seen in Figure 5.10. During this period about 34000 unique vehicles are part of the simulation. For simplicity, we assume all PVC clients are drivers, and we utilize uniform random distribution to select clients. We obtain the trace of vehicular positions using the SUMO traffic simulator [167], and we build our system on top of this simulator. SUMO allows users to shut down roads, forcing vehicles to reroute and allowing us to test our system for fault tolerance. In addition, SUMO updates the average speed of each road segment. Thus, we calculate the future position of the vehicle based on the average speed of roads. Note that this prediction is not the actual arrival time as prediction error increases when predicting longer routes.

When clients wish to submit their requests, they send their future sensing route and sensing objective to the server. In the push-based solution, the system will schedule participants to cover clients' sensing routes in a first-in-first-serve manner. A vehicular client cannot be a participant for its own task but can be a participant for other tasks. For parameter settings, we set the budget to be constrained by $\beta_c = 40$, the soft parallel task constraint for each vehicle to be $cap = 10$, the monitoring window which is defined as $MW \leftarrow \varepsilon - \delta$ to be 900, and we set the hiring window for the pull-based solution to be $\lambda = 60$ *second*. The experiment flow is shown in Figure 5.11.

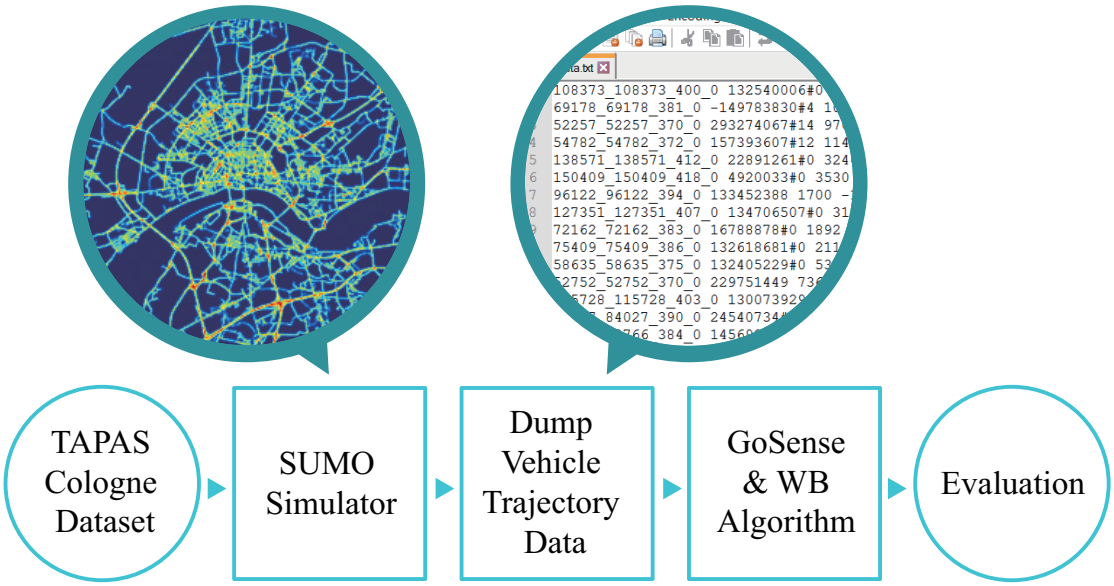


**Fig. 5.11** Experiment workflow

### 5.6.1 Load balance

We evaluate the performance of our solutions in terms of load balance by comparing against the PR algorithm in GoSense [21], Least Load First, Hard Cap, and FCFP. Least Load First refers to allocating a task to a participant with fewest tasks and Hard Cap refers to allocating tasks to the best participant who is not overloaded (above its capacity). Note that we use $LCF$ heuristic, described in definition 5 for selection in Hard Cap. As for the performance metric, we compare the maximum peak workload among all participants. We define the peak workload as:

$$Peak\ workload \leftarrow \arg\max_{p' \in P}\ \ell oad_{p'}^{max}.$$

As shown in Figure 5.12, WB outperforms GoSense, Hard Cap, BOF, FCFP, and is similar in performance to Least Load First for peak workload. Note that the workload cap is 10, and hence the peak workload is at most 10. Peak workload grew the fastest for FCFP and BOF second because vehicles can pull requests greedily from servers until they reached the cap.

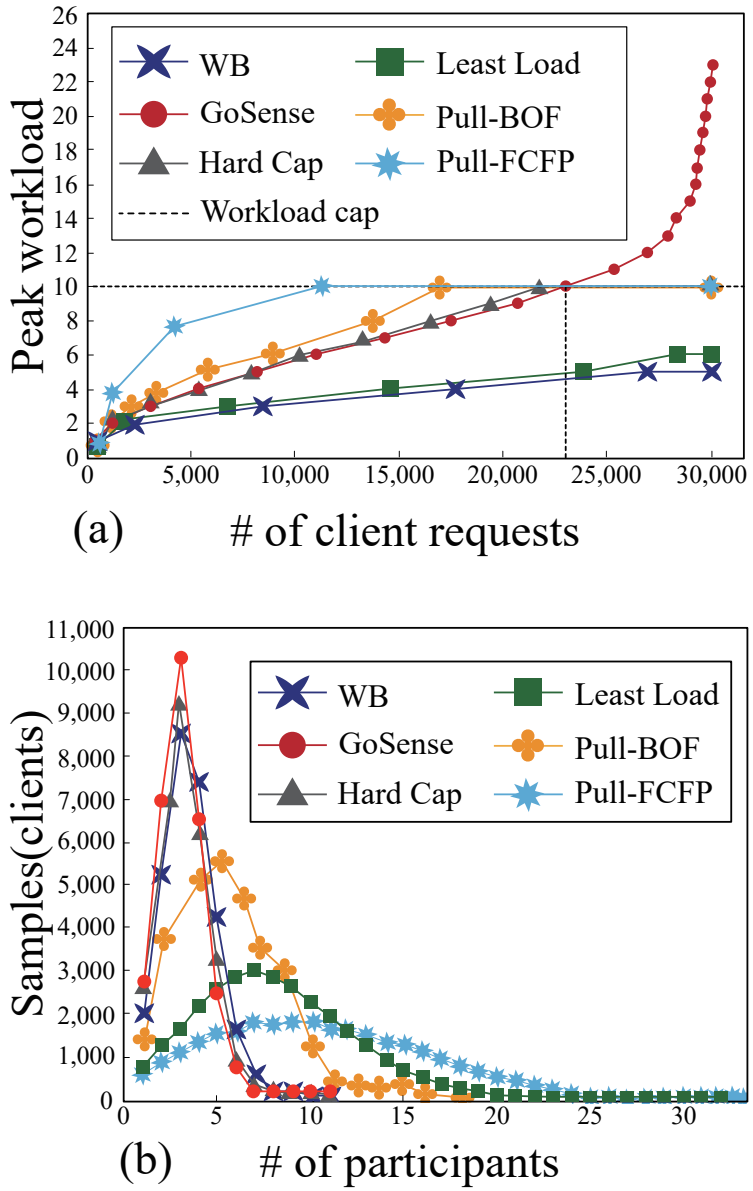


**Fig. 5.12** Comparison of WB algorithm, Least Load First, Hard Cap, GoSense, FCFP, and BOF performance.

### 5.6.2 The participants number

We further contrast the number of participants required for each algorithm. The number of participants refers to the average number of vehicles required for sensing per task which should be minimized. We find GoSense, Hard Cap, and WB are similar in performance while BOF requires slightly more participants. Finally, Least Load First performed far worse as shown in Figure 5.12 and FCFP performed even worse. We also observed that GoSense requires the least number of participants but also causes significantly heavier peak workload in comparison to WB. Thus, this represents an acceptable tradeoff between the number of participants and the load balance.

### 5.6.3 Window Based Scheduling (Static v.s Dynamic)

In this section, we explore the improved WB algorithm using window-based scheduling. For this evaluation, we utilize a simple position prediction algorithm using average route speed. Hence, significant errors exist when predicting for longer periods. Figure 5.13 shows the performance in terms of subtask failure rate in the Cumulative Distribution Function (CDF) graph format. We define subtask failure rate as $Subtask\ failure\ rate_c \leftarrow 1 - cover\_rate_c$. Subtask failure rate refers to the average percent of road segments task without sensing coverage. For instance, a task with a 30% subtask failure rate signifies at least 30% of its sensing road segments could not be completed within the allotted time.

We see that scheduling for whole routes led to the worst performance. Using 80% of vehicles as a comparison point, scheduling for whole routes result in 20% and 40% subtask failure rate compared to 5% and 20% for static window scheduling and 10% and 20% for dynamic window scheduling. There exists a trade-off; static window scheduling requires frequent scheduling and increased computational overhead. Thus, for reducing the frequency of scheduling, dynamic window scheduling can reduce the scheduling count as seen in Figure 5.14. Furthermore, the performance of dynamic window scheduling remains within reach of static scheduling. Thus, we have shown dynamic window scheduling can be a competitive option when considering overhead for rescheduling all vehicles.

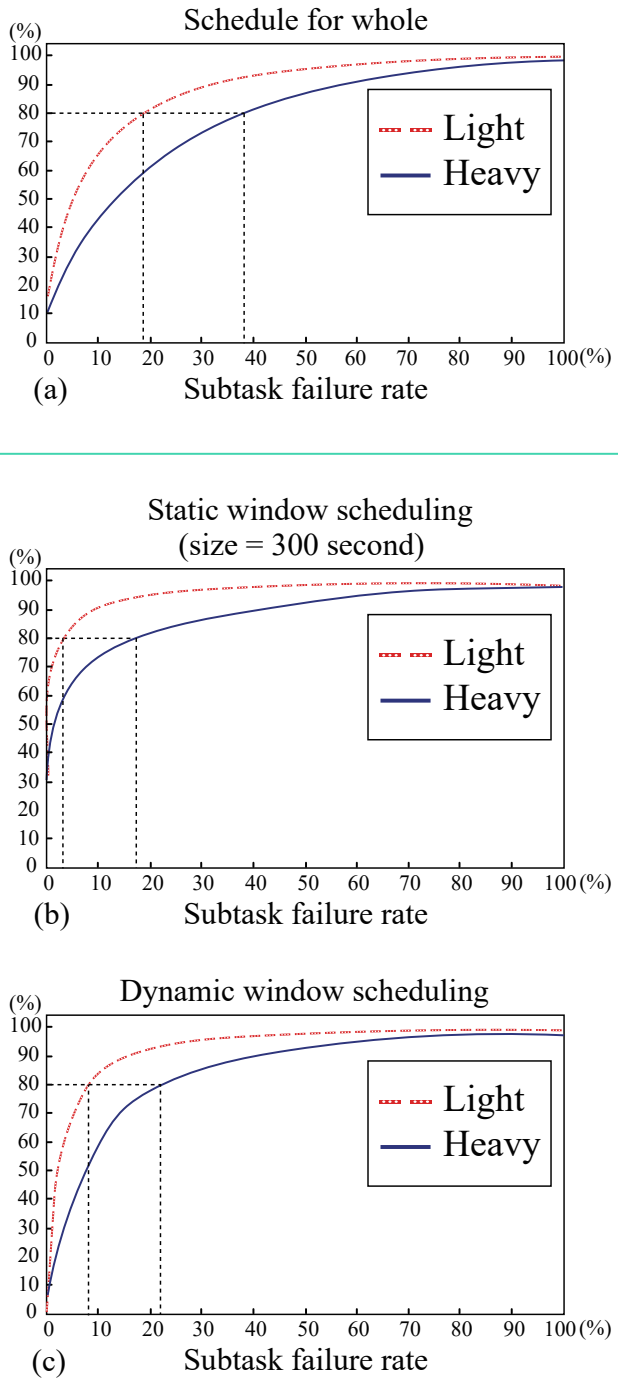


**Fig. 5.13** Evaluation results are shown in the Cumulative Distribution Function (CDF) graph for random monitoring window. The random number is generated using the same seed. We define subtask failure rate as the number of route segments without sensing coverage $R_c^{miss}$ divided by the total query route segments $R_c$. For calculating $SW$, we set $\theta = 2$ and $SW_{min} = 300$ for our dynamic window scheduling.

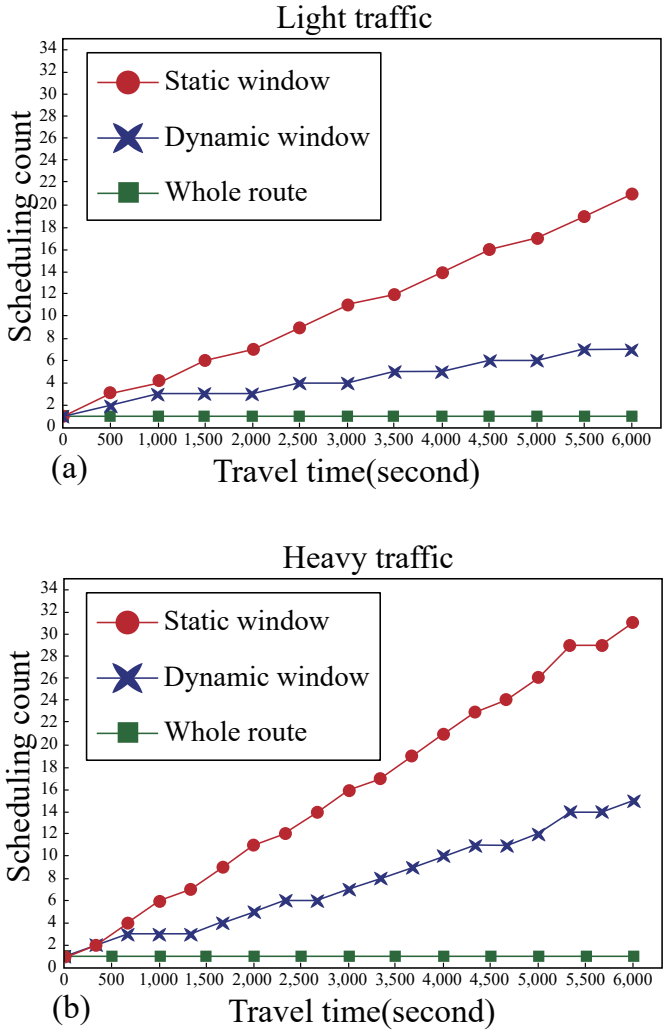


**Fig. 5.14** Scheduling count difference under light and heavy traffic.

### 5.6.4 Pull Based Solution v.s Push Based Solution

In this section, we contrast the performance of our pull-based solution to push-based solution: dynamic window and GoSense. We evaluate the solutions under two types of simulation conditions: base simulation and simulation with rerouting.

The base simulation did not include unforeseen events such as accidents and route changes. In the real world, drivers change route based on real-time conditions. With the base simulation, there exists error for predicting when a vehicle will arrive but its route does not change. Thus, we implemented a rerouting error which allows for vehicles to change their current route. We implemented this in SUMO by adding random slowdown on-road segments. We selected from a uniform distribution 0.4% of road segments (out of 70000) and set the maximum speed of these routes to on average 1 m/s and repeated once every 120 seconds as to vary roads segment slowed. Thus, vehicles set to travel on those routes segment will be slowed. Furthermore, vehicles would check their route once every 60 seconds and reroute if there exists a faster alternative to destination. This in combination with the road segment slowdown increased the prediction error rate. We refer to this error as a rerouting error.

As shown in Figure 5.15, using the subtask failure rate as the performance metric, we observe that having rerouting error decreased performance for all algorithms to varying degrees. The performance of the push-based solutions is worse; the subtask failure rate is high. For both the base simulation and rerouting simulation, GoSense has the highest subtask failure rate. This is expected given each task is only scheduled once leading to high estimation errors. Furthermore, we note that GoSense performance with rerouting error significantly worsens; this is also expected given its inability to change participants. In contrast, push with scheduling window performs better for both the base and rerouting simulation. The improvements in performance are likely due to frequent rescheduling which can take into account estimation errors instead of allowing them to accumulate. Finally, using 80% of vehicles as a comparison point, Pull-BOF outperforms push-based solutions with result about 8% and 15% subtask failure rate in both the base simulation and rerouting simulation, respectively. It's interesting to note that the gap between Pull-BOF and window based push increased with rerouting error. This can be explained given Pull-BOF selects participants when they are about to reach the sensing location, leading to small estimation errors. When rerouting happens, participants' sensing may become useless if it continues to sense. Thus, Pull-BOF has shown it can consistently outperform push in terms of task failure rate especially in the presence of rerouting errors.

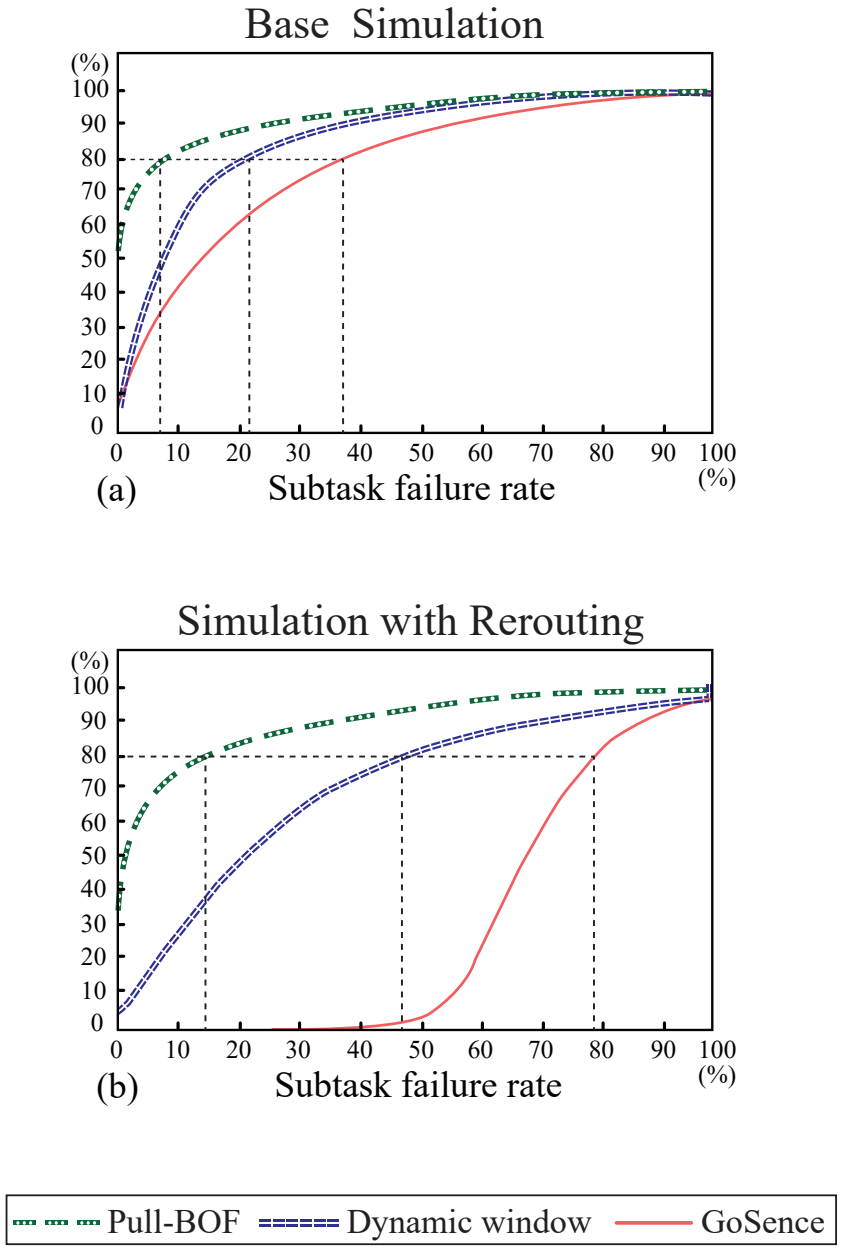


**Fig. 5.15** CDF of tasks versus task failure rate.

# Chapter 6

# Vehicular Route Planning For Vehicular Crowdsensing

In this chapter, we introduce our vehicle rerouting algorithm to enhance sensing coverage performance. Since the public vehicular crowdsensing and personalized vehicular crowdsensing target different types of sensing tasks, we will provide solutions accordingly and they will be explained in two different sections. In addition, the notations used in this chapter are summarized in Table 6.1.

| Notation | Description |
|---|---|
| $R$ | Set of road segments. |
| $V$ | Set of vehicular participants (vehicles). |
| $V^{new}$ | Set of new added vehicles. |
| $T$ | Set of tasks. |
| $C$ | Set of tasks owner |
| $\mathbb{T}$ | Sensing time window. |
| $G$ | A city map is decomposed into set of grids. |
| $g^R$ | A grid contains a set of road segments. |
| $g_{ij}$ | Represent an uncertainty value in the map. The value are initially set to 1. |
| $b$ | The label of start. |
| $d$ | The label of end. |
| $p$ | A single path (route). |
| $p^{b,d}$ | A path starting at road segment $s^b$ to a destination road segment $s^d$. |
| $tasks(r)$ | The number of available tasks a single road segment currently has. |
| $r^d$ | The vehicle's destination road segment. |
| $r^{d'}$ | The global planning destination road segment. |
| $v^{cap}$ | Vehicle sensing capacity. |
| $v_i^{t,r}$ | The set of tasks $t$ that vehicle $v_i$ at road segment $r$ can serve. |
| $t^p$ | The completion probability of the task $t$. |
| $\mathcal{A}(v, r)$ | The predicted arrival time of vehicle $v$ at road $r$. |
| $\tau(r)$ | Traveling time of the road $r$. |
| $dist(r^b, r^d)$ | Total traveling time given $r^b$ and $r^d$. |
| $D(g_1, g_2)$ | The distance between the two grids. |
| $\zeta(p_i)$ | The total expected tasks completion given a path $p$ of the vehicle $i$. |
| $A^*(r_i^b, r_i^d)$ | The shortest path from $r_i^b$ to $r_i^d$ in terms of travelling time using $A^*$ algorithm. |
| $\mathcal{E}(r_i^d)$ | The extra time the vehicle $v_i$ is willing to reroute given the destination $r^d$. |
| $\phi(r, r_i^d)$ | The cost function combine both travelling time and tasks taken for a single road. |
| $\theta_i$ | The current timestamp of the $i$th vehicle at $r_i^b$. |
| $static\_t(r)$ | Function to generate a static sensing task. |
| $dynamic\_t(r_i)$ | Function to generate a dynamic sensing task given an $i$th vehicle's route. |

**Table 6.1** List of the major notations for chapter 6

## 6.1 Route Planning for Public Vehicular Crowdsensing

In this section, we introduce our vehicles routing algorithms for public vehicular crowdsensing. Inspired from the work [128], our proposed route planning solution requires a vehicular participant to bid for picking a sensing task. Since this is an auction-based system, participants have the right to choose whether to take a detour or to stay on the same route. A participant who wishes to participate but stay on the same route could propose its original route for bidding.

Our system is composed of a set of participants who propose multiple bids (with possible routes) to the server; each participant seeks to maximize its chances of being selected subject to its constraint. The server seeks to minimize the number of participants selected while maximizing coverage. Similar to the conventional crowdsensing system our system consists of a centralized cloud server and a set of smart vehicle participants. The server acts as the data storage and access point for sensing data. The server contains information about all participants such as a participant's sensing capability and its position.

At every time window, the server attempts to select a set of participants for crowdsensing. As part of our auction-based system, each participant generates multiple potential routes it is willing to take. They propose routes that maximize their chances of getting selected. The routes a vehicle generates depend on its willingness to reroute. For instance, a participant late to work would propose new routes close to its current route if any at all. In contrast, a participant on a joyride might generate routes that stray far from its intended route.

Once the server receives proposals from participants, it must select the set of vehicles and proposed routes that can maximize coverage. The system can evaluate the value of a proposed route by estimating the predicted position of the vehicle within the time window. Note that given it is an estimate, errors frequently occur. Once the selection is complete, the server notifies the selected vehicles to begin sensing for the specified time window. This process is continuously repeated given that we seek to cover a sensing region over an indefinite amount of time.

For the assumptions, we assume that all vehicles are self-driving cars; regular vehicles may not possess sufficient sensors. Participants only need to specify a time frame for arrival time and let the self-driving car freely plan its route. A participant who urgently needs to arrive at his destination can lower available rerouting time whereas the participant with time to spare can increase rerouting time. In addition, we assume that participants are sufficiently incentivized for rerouting such as receiving extra payment when they reroute. We further assume that once a vehicle proposes a potential route and is selected, it will always be willing to take that route; i.e. a vehicle would not propose any route it is unwilling to drive by. Furthermore, we assume that once a vehicle is selected, it will actually follow the route it proposed for the duration of the time window. Note, that we make no assumption about a participant's speed. A participant can be stuck in an unforeseen traffic jam and may not arrive by the estimated times.

### 6.1.1 System Model and Problem Statement

For explaining our system model and problem statement, we use the following definitions.

**Definition 7.** ***Information Level:*** *We define our information level in relation to the overall level of information for the sensing region. The sensing map ($G$) is viewed as a grid of discrete spaces addressable as $g_{i,j}$. Each position in the grid-map holds an* ***uncertainty value*** *which indicates the information level of that area. The uncertainty value varies between* 0 *to* 1 *as shown in Figure 6.1. An uncertainty value of 1 denotes a lack of sensing information at that area and 0 denotes a high information level in that region. The $g_{i,j}$ values, the uncertainty values, for all $i, j \in G$ are initialized to be* 1 *at the beginning of our simulation. Thus, we initialize the map to have no coverage at the beginning of the simulation. The overall information level of the map at any instance is given by the Average Uncertainty Level ($AUL$) calculated across all grid positions.*

$$AverageUncertaintyLevel(AUL) = \frac{\sum_{\forall i,j \in G} g_{i,j}}{|G|} \tag{6.1}$$

*Where* $|G|$ *=total number of grid points in the map. The route selection algorithm selects the set of routes that minimizes* $AUL$*. We also use the* $AUL$ *as our evaluation metric for the algorithm(s). As participants move past these grid cells, their* $g_{i,j}$ *values are updated to be reduced by a factor of the participant's sensing quality. In our simulation we initialize the sensing quality* $(SQ)$ *per participant randomly between the range of* $0.2 - 0.8$*. This is because each participant performs its sensing task differently which results in varying quality of sensing information; thus each has a different impact on the uncertainty level. In addition to* $SQ$*, the participants are also assigned a random metric for sensing range* $(SR)$*. The* $SR$ *denotes the effective sensing radius that can affect the nearby grid. The effectiveness of the sensing range depends on the distance between the neighbor grid and its current position. Inspired from the work [168], how both* $SQ$ *and* $SR$ *updates the uncertainty level is described as the follow:*

$$updated\_UL \leftarrow \begin{cases} SQ & D(here, g_{ij}) < 0.25 \times SR \\ 0.75 \times SQ & D(here, g_{ij}) < 0.50 \times SR \\ 0.50 \times SQ & D(here, g_{ij}) < 0.75 \times SR \\ 0.25 \times SQ & D(here, g_{ij}) < SR \\ 0 & otherwise \end{cases} \tag{6.2}$$

*Where* $D(g_1, g_2)$ *returns the distance between the two grids.*

*The* $SR$ *along with the* $SQ$ *of the participants contributes to updating the* $g_{i,j}$ *values of grid-cells surrounding a participant. Whilst these updates reduce the* $g_{i,j}$ *per grid-cell, a constant* ***decay rate*** *increase* $g_{i,j}$ *per simulation time-step. We set this constant to* $0.0005$*, which means the uncertainty level of each grid cell takes* $\approx 2000secs$ *to reach* $1$ *from* $0$*. The decay introduced at every simulation-step and the uncertainty level value reduced by the moving participants modifies the* $AUL$ *throughout the simulation. Thus, our route selection algorithm attempts to pick the best combination of proposed routes such that no grid cell is left unvisited for a long period.*

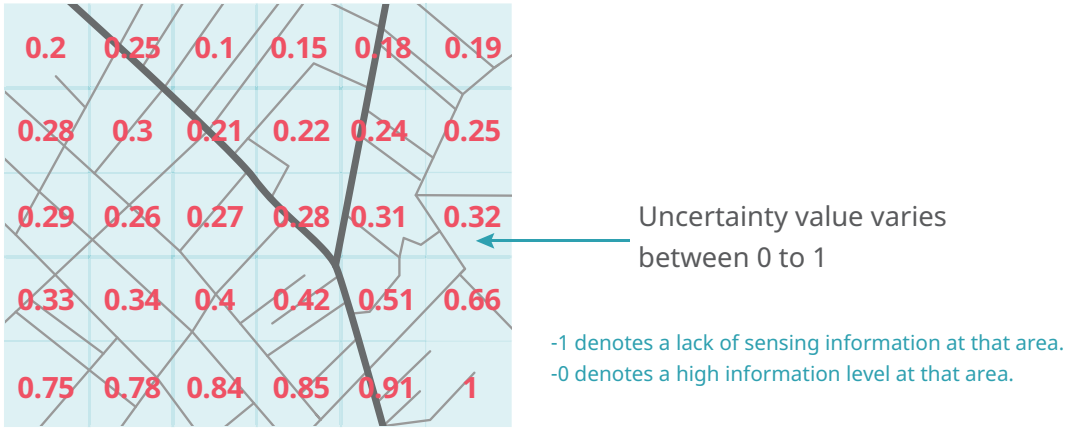


**Fig. 6.1** Information level

**Definition 8.** ***Route Information Gain (IG):***

*Since each vehicle can propose multiple routes, we calculate the information gain specific to each proposed route. Let a single route, p, contain a sequence of location grids and* $P_v = \{p_1,\ p_2,\ ...p_n\}$ *denotes all the proposed routes by vehicle* $v$*. We also let function* $ESQ(g_{ij}), \forall g_{ij} \in G$ *gives the Effective Sensing Quality based on Equation 6.2, given the current location* $g_{ij}$ *and the sensing radius. Thus, for every route proposed by vehicle* $v$*, we can derive the IG value as*

$$IG(p) \leftarrow \sum_{g \in p} ESQ(g) \tag{6.3}$$

*Thus, the maximum IG value amongst all routes is calculated as:*

$$MaxIG(P_v) \leftarrow maximize\ IG(p),\ \forall p \in P_v. \tag{6.4}$$

**Definition 9.** ***Objective Function:*** *Given vehicle set* $V$*, cost function* $Cost()$*, and Budget* $B$*, find a set of sensing routes and vehicular participants which minimizes* $AUL$*; subject to* $\sum_{v_i \in V} Cost(v_i) <= B$.

### 6.1.2 Algorithms

#### Route Generation

Since we assume each participant only knows the location of nearby participants, they do not know what others will propose. Therefore, we allow each participant to submit multiple potential routes. Participants only submit routes which they consider to be promising. Hence, our route generation algorithm is fairly straight-forward; it attempts to generate routes step by step by selecting greedily the best valid step based on a heuristic function. To generate multiple different routes, sensed locations have their heuristic value weighted by the number of times visited. Hence, this can create diverse new routes without much overhead. The algorithm stops when the participation-willingness or the time-window is reached. Thus, vehicles unwilling to reroute will not generate any routes beyond its original route. The run time of the base route generation algorithm is $O(NumPath * TimeWindow * |R|)$ * $O(heuristic)$. Algorithm 4 describes this process in detail.

**Algorithm 4** Path Generation Algorithm

```
1: Input: v_i, 𝕋, num_path, heuristic
2: VisitedGrids ← ∅
3: MaxReroutingTime ← min(𝕋, getRerouteWillingness(v_i))
4: generatedPaths = ∅
5: for j ← 0; j < num_path; j++ do
6:     result ← ∅
7:     CT ← time(v_i)                                  ▷ Current Time
8:     CG ← g_{v_i}                                    ▷ Current Grid
9:     while CT < MaxReroutingTime do
10:        bestGridValue ← 0
11:        for neighboringGrid in CG do                ▷ Base on sensing range
12:            if heuristic(neighboringGrid) > bestGridValue then
13:                nextGrid = neighboringGrid
14:                bestGridValue = heuristic(neighboringGrid)
15:            end if
16:        end for
17:        CT ← currentTime + nextGrids.avgTime
18:        CG = nextGrid
19:        result ← result + nextGrid
20:    end while
21:    generatedPaths ← generatedPaths + result
22: end for
23: return generatedPaths
```

We explored three heuristics for path generation. The first is the baseline and is a random heuristic where the chance of visiting different nodes is uniform. The second algorithm is based on greedy grid values, whereby we select the set of grids with the highest expected information gain. Finally, the third heuristic only takes into account the distance of the grids from other participants. Hence, a participant seeks to visit grids that other participants are farther away from and are less likely to be visited. To adjust for grids of different length, the value is multiplied by the grid-length to reward longer grids. The idea is to generate routes that cover grids

unlikely to be covered by other participants. We denote this heuristic as *unlikely heuristic*. This heuristic only considers close neighbors since participants far away are less likely to compete for covering the same areas.

#### Route selection

Based on the model that we have established above we experimented with two different route selection heuristics to evaluate how well the map could be covered.

- MAX-IG: In the first heuristic, we directly leverage $IG(P_v)$ of the proposed routes to pick the best set of routes. We sort the proposed routes by $IG(P_v)$ and select the top $B$ routes. Note that once a vehicle is selected all other routes it proposed are removed from the solution.
- MAX-WIG: Given that $IG(P_v)$ is computed independently, selecting $B$ sorely based on this measure leads to overlaps. Thus, we propose weighted information gain ($WIG$) which accounts for areas already covered by already selected routes. Here, we simply weight the $IG(P_v)$ of grid positions by the number of times they have already been covered. Let vehicle $v_1$ propose a route covering $\{g_1, g_2, g_3\}$ with no overlap and as such its $IG(P_v)$ remains unchanged. However, if vehicle $v_2$ proposes route $\{g_5, g_8, g_3, g_4\}$, its $ESQ(g)$ value will be penalized for the overlap resulting in a lower $IG$. Since the overlapping grid is $g_3$ and it only appears twice, the $ESQ(g_3)$ is divided by 2. Thus, the penalized weight function is described as follows:

$$WIG(P_v) \leftarrow \sum_{g \in p} \frac{ESQ(g)}{Overlap\_Count} \tag{6.5}$$

### 6.1.3 Walk Through Example

We show the inner workings of our route generation and selection with the example in Figure 6.2. In this example, let $V^{new}$ be the set of newly added vehicles. Each newly added vehicle follows its original route that is generated by the built-in navigation system. For serving sensing tasks, each vehicle $v \in V^{new}$ needs to propose a set of new routes that includes its original route. Let $v_1 \in V^{new}$ and $v_2 \in$

$V^{new}$ be the two new added vehicles. Additionally, let $P_{v_1}$ be the proposed routes of $v_1$ and $P_{v_2}$ be the propose routes of $v_2$ using one of the route generation algorithm introduced in the section 6.1.2. From the server-side, the platform continuously collects proposed routes and begins selecting routes after a set time interval using the route section algorithm introduced in section 6.1.2. Even though a single vehicle can propose multiple routes, only one route can be selected, discarding other proposed routes. Once the selection is complete, the server notifies $v_1$ and $v_2$. The selected vehicle $v_1$ will begin sensing for a specified time window. The unfulfilled vehicle $v_2$, on the other hand, need to propose new routes in the next hiring time window. $v_2$ will continuously propose new routes until the proposal gets accepted or when it arrives at its destination.

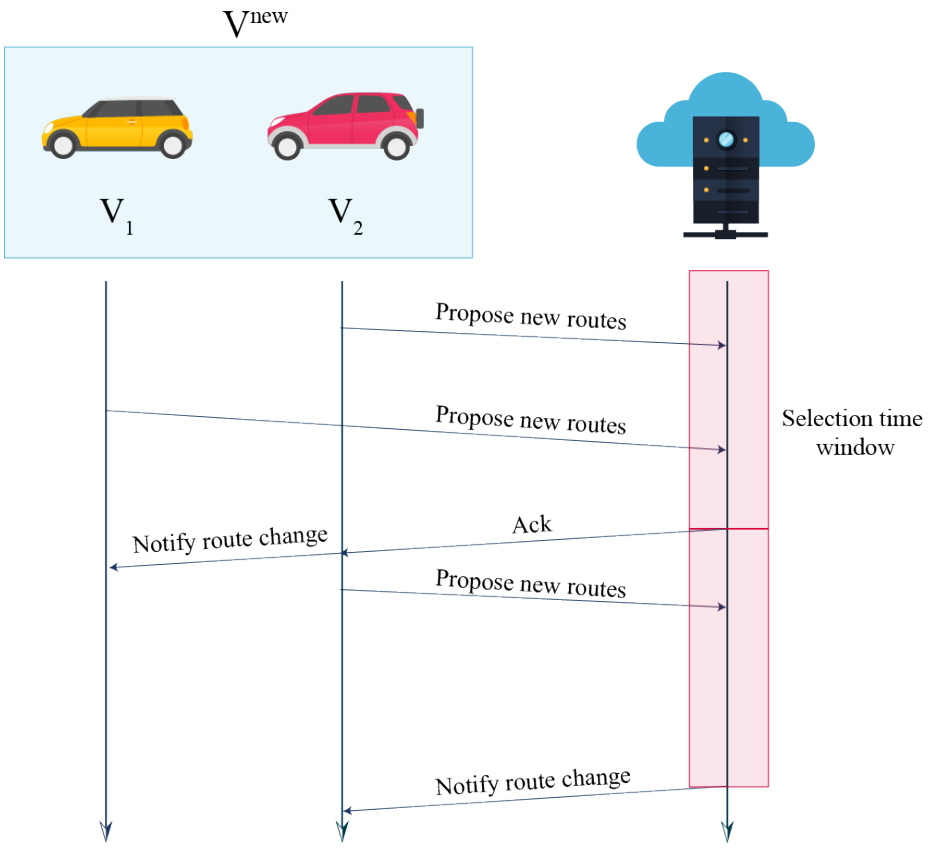


**Fig. 6.2** Route planning flow.

### 6.1.4 Algorithm Evaluation

In this section, we explore the combined results of our proposed route selection and route generation algorithms. Given longer execution-times, some evaluations were run for slightly longer than others. First, we can see that with rerouting, coverage metrics improved by 15 to 20 percent compared to the performance without any rerouting. Thus, there is an advantage for vehicles to reroute for sensing. Furthermore, we can see that $WIG$ always outperforms $IG$ regardless of the route generation algorithm employed (see Figure 6.3 and Figure 6.4). This makes sense given that it takes into account previously selected areas and attempts to avoid selecting routes with too much overlap.

In contrast, the results were less clear for route generation. Each participant proposed 21 routes with the rerouting approach; 1 consisting of its original route and 20 generated routes which it is willing to reroute to. Thus, the original route is always a possibility. Rerouting heuristics comprised of Random, Greedy and Unlikely. The baseline approach only proposed its original route. As we can see from both Figure 6.3 and Figure 6.4, all rerouting based approaches greatly outperformed the baseline approach by about 20 percent on average. However, different path generation heuristics yielded negligible differences. We suspect this is because of high participant density and the high number of proposed routes. High participant density means all possible areas are covered and the high number of proposed routes meant the utility of heuristic is reduced given most possible routes are included. Due to these factors, it's likely that even if the routes proposed were random, there are enough proposals and participants to cover all necessary sensing locations. We did attempt a limited test with only 2 routes proposed with the original and greedy solutions that consistently showed better performance.

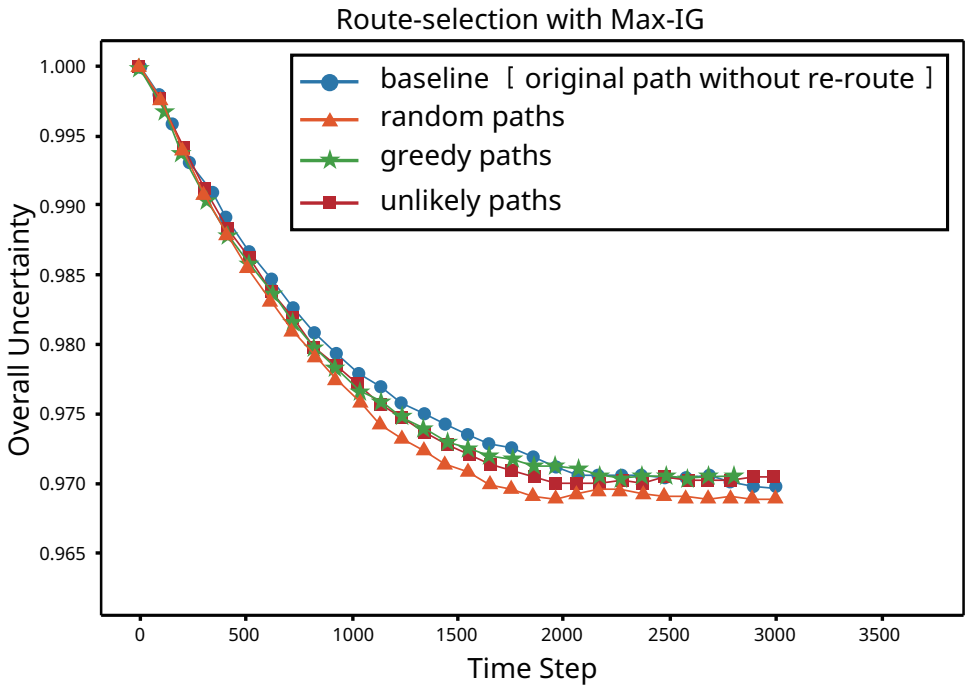


**Fig. 6.3** Max-IG selection

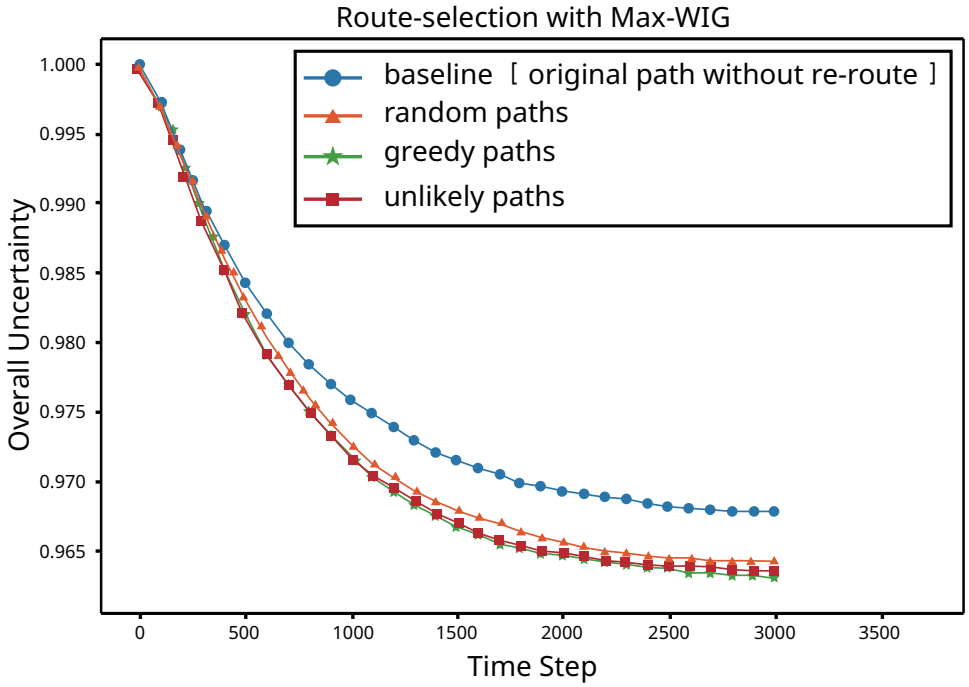


**Fig. 6.4** Max-WIG selection

## 6.2 Route Planning for Personalized Vehicular Crowdsensing

In this section, we introduce our solutions with personalized sensing tasks into consideration; that is, the solutions should consider dynamic sensing tasks. Similar to most vehicular crowdsensing systems, our system consists of three entities: crowdsensing service platform, sensing task provider, and vehicular participants. The crowdsensing service platform is the service provider that includes a central cloud server and multiple fog nodes that are deployed at different geographical locations as shown in Figure 6.5. Each vehicle can have a reliable platform connection with minimal response delay. The sensing task provider is the task owner and can be individuals (e.g. drivers) or organizations. Vehicular participants are the mobile participants who perform the crowdsensing task and earn rewards from the task provider. The task providers who want to collect data should publish their sensing tasks to the service platform. The platform can then select a set of vehicles willing to collect the required data and send it back to the task provider through the platform. The sensing tasks are location and time-sensitive; a task must be completed at a specific location before its deadline to be meaningful to the task provider.

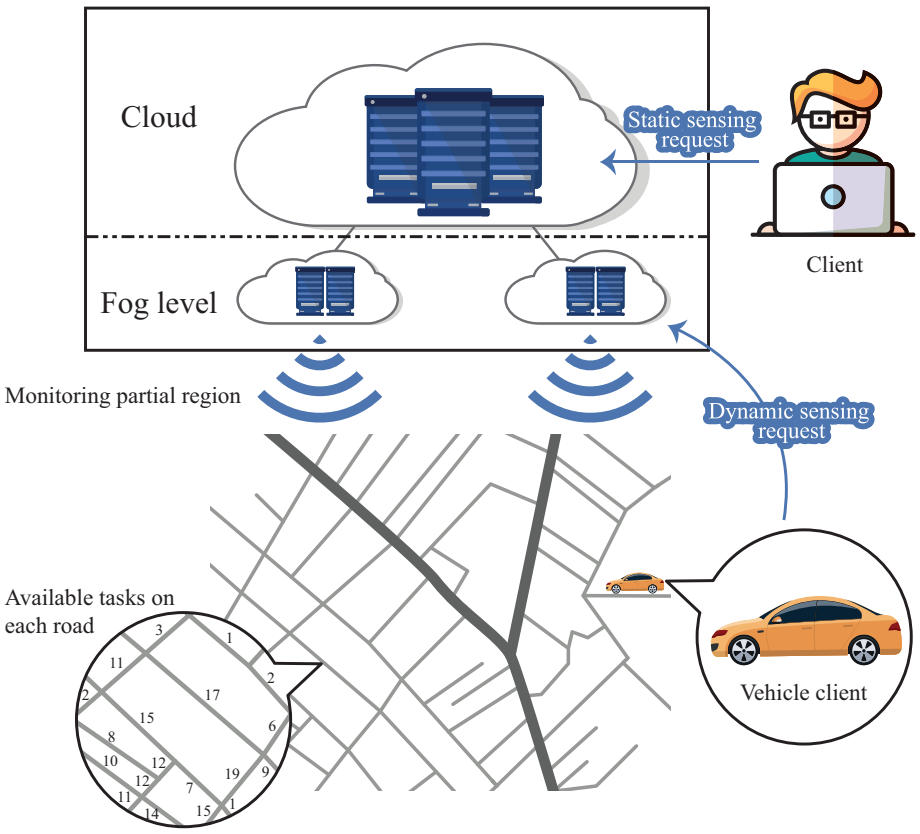


**Fig. 6.5** The vehicular crowdsensing architecture

A vehicle can join the crowdsensing system at the beginning of its journey and leave the system any time when its destination is reached. We also assume that the cloud service can cover the entire map: vehicles can communicate to our platform at all times. Also, we assume that there exists a trust management system implemented on our platform. This means that both participants and task providers must follow certain rules and policies to achieve their objective of receiving data and rewards respectively. Furthermore, we assume that all task providers generated sensing requests follow the Zipf distribution. The decision is according to the studies in [192]. Finally, since our solution requires vehicles to take detours, the amount of extra time and effort these vehicles are willing to provide is based on the incentive models. Thus, we assume that there is a reward function $\gamma : v \rightarrow \mathbb{Z}^+$ where a vehicle $v$ is willing to offer to perform extra sensing tasks in exchange for further rewards.

### 6.2.1 Problem Definition

In this section, we introduce the notations we used when describing rerouting algorithms. All the notations are summarized in Table 6.1. Our system contains $|V|$ vehicles which are denoted as $V \rightarrow \{v_1, ..., v_i\}$. Each vehicle has a fixed capacity of $v^{cap}$ which specifies the maximum number of tasks the vehicle is willing to serve simultaneously. In addition, the system contains a road map which consists of a set of $R$ road segments $R \rightarrow \{r_1, r_2, ..., r_j\}, j \in \mathbb{Z}^+$. Road segments consist of a section of a road between two intersections. Each road segment has a single traffic direction. Roads that have opposite traffic directions are considered to be two different road segments.

Let the function $\tau : r \rightarrow \mathbb{Z}^+$ be the expected traveling time given a road segment $r$ and let $p^{b,d} = \{r^b, r_1, ...r^d\}$ $\forall r \in R$ denote a path starting at road segment $r^b$ to a destination road segment $r^d$. The total traveling time given a path $p^{b,d}$ can be calculated as the follow:

$$dist(r^b, r^d) = \sum_{r_i \in p^{b,d}} \tau(r_i) \tag{6.6}$$

Note that the return value of function $\tau(r_i)$ is a a constant for a short time duration.

Let $r_i^d$ be the destination of the $i$th vehicle, the shortest path from $r_i^b$ to $r_i^d$ in terms of travelling time using $A^*$ algorithm can be found from:

$$A^*(r_i^b, r_i^d) \rightarrow \arg\min_{r^{b,d}} \ dist(r_i^b, r_i^d) \tag{6.7}$$

**Definition 10.** ***Rerouting Time:*** *We define rerouting time as the extra time a vehicular participant is willing to reroute. The willingness of each vehicle to take detour varies depending on the incentive mechanism. Since designing a mechanism is not the main focus on this paper, we use a price-based incentive mechanism due to its popularity in crowdsensing research [114]. As shown in prior works, drivers' rerouting behaviour [193] and participant sensing performance given an incentive mechanism are distributed following a normal distribution [194]. Thus we define an expected arrival time $\mathcal{E}(s_i^d)$ which includes the willingness of rerouting time for*

*each vehicle as follows:*

$$\mathcal{E}(r_i^d) \rightarrow \theta_i + A^*(r_i^b, r_i^d) + \gamma(v_i) \tag{6.8}$$

*where $\gamma(v_i)$ is the extra time the $i$th vehicle is willing to reroute and follows a normal distribution and $\theta_i$ is the current timestamp of the $i$th vehicle at $r_i^b$. Since a vehicle can have more than one potential path to its destination, a set of valid routes $R_i$ for a vehicle $v_i$ is defined as:*

$$R_i \rightarrow \{r | \theta_i + dist(r_i^b, r_i^d) < \mathcal{E}(r_i^d)\}$$

*Note that we prohibit repeated road segments.*

Furthermore, the server receives a set of tasks from a set of clients which can be denoted as $\mathcal{T} \rightarrow \{T_1, T_2, ..., T_c\}$, where $c$ is the index of the associated client. Each client's sensing task contains set of subtasks which is denoted as $T_c \rightarrow \{t_1, t_2, ..., t_n\}, n <= |R|$. Each subtask specifies the sensing road segment $t^s$, and the sensing start time $t^b$ and the end time $t^d$ of the task. Besides, each subtask has a value $t^{cost}$ which determines task processing time. After the server receives the sensing task, the central server distributes subtasks according to the associated road segment. Thus each road segment $r$ contains a set of tasks which can be obtained from the function: $tasks(r) \in \mathbb{Z}^+$.

**Definition 11.** ***Tasks Generation:*** *Our system accepts both static and dynamic task requests as shown in Figure 2.3. Multiple prior studies suggest that web data usage follows a Zipf distribution [195], thus we generate both static and dynamic demands following the Zipf distribution. The probability density function of Zipf distribution is: $Zipf(x; \alpha, R) = \frac{1}{x^\alpha \sum_{s=1}^{|R|}(\frac{1}{s^\alpha})}$, where $\alpha$ is the control parameter for the shape of the distribution, $|R|$ is the total number of road segments and $x \in \mathbb{Z}^+$ is the road's rank. Each road's rank is based on the distance to the downtown area. The roads which are close to the downtown area have the highest rank whereas the roads in the suburbs have the lowest rank.*

*When a client selects a sensing road segment, the nearby road segments also get selected. This is because, in reality, a sensing request usually covers a specific area rather than a single road segment. Thus, given a function* $near(r, radius)$ *to obtain the nearby road segments with a central road segment* $r$ *and a radius, a static sensing demand can be obtained from the following:*

$$static\_t(r) \rightarrow \{t|near(r, radius), r \sim Zipf(\alpha, R)\} \tag{6.9}$$

*For dynamic demands, tasks are generated by vehicles, and their sensing subtasks are shifted based on their positions. Thus, a dynamic sensing demand from* $i$*th vehicle is obtained from:*

$$dynamic\_t(r_i) \rightarrow \bigcup_{r \in r_i} static\_t(r) \tag{6.10}$$

Let $\mathcal{A} : v \times r \rightarrow \mathbb{Z}^+$ be a function for the predicted arrival time of vehicle $v$ at the road $r$. The predicted arrival time can be obtained from the built-in navigation system in the vehicle. Thus, a completed task at a road segment is defined as:

$$task\_completed(v, t, r) \rightarrow \begin{cases} 1 & t^b < \mathcal{A}(v, r) < t^d - t^{cost} \wedge v^{cap} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.11}$$

Given the Equation 6.11 and a road segment, the number of tasks a vehicle $v_i$ can take is calculated as:

$$\sum_{t \in tasks(r)} task\_completed(v_i, t, r) <= v_i^{cap}$$

Thus, given a single path of $i^{th}$ vehicle, the total expected tasks completion can be obtained from the following function:

$$\zeta(p_i) \rightarrow \sum_{r \in path_i} \sum_{t \in tasks(r)} task\_completed(v_i, t, r) \tag{6.12}$$

Thus, given a set of vehicles and a set of tasks continuously added on the map, our goal for the vehicular crowdsensing path planing (VCPP) problem is to find

an optimal route:

$$p^{best} \rightarrow \arg\max_{p_i \in P_i} \ \zeta(p_i)$$

### 6.2.2 Problem Hardness

In this section, we prove that VCPP is NP-complete by reducing the well-known 0/1 Knapsack problem to the VCPP problem.

**Theorem 2.** ***VCPP problem is NP Complete:*** *The decision version of the VCPP problem is described as: Suppose we have a set of road segments $R$. Each road segment has an average traveling time $\tau(r), r \in R$, and has a set of tasks which vehicular participants can complete. Given a destination, expected arrival time to the destination $\mathcal{E}(r^d)$, and vehicle sensing capacity $v^{cap}$, is there a subset of road segments with the total traveling time that meets the $\mathcal{E}(r^d)$ constraint, such that the corresponding task assignments meet the constraint of $v^{cap}$?*

*Proof.* To prove VCPP problem is NP: Let route $p^{best}$ be the solution for the VCPP problem. We can verify the correctness of a route in polynomial time where the complexity is $O(|p^{best}|)$. □

*Proof.* To prove VCPP problem is NP-complete: The 0/1 Knapsack problem is a NP-hard problem which is described as: *Given a non-negative weights $\hat{W} = \{\hat{w}_1, \hat{w}_2, ..\hat{w}_m\}$, $\hat{B}$ and values $\hat{V} = \{\hat{v}_1, \hat{v}_2, ..\hat{v}_n\}$, $\hat{K}$. Is there a subset of weights with total weight at most $\hat{B}$, such that the corresponding profit is equal to $\hat{K}$?* The polynomial time reduction step of constructing VCPP instance is described as follows: □

Let the traveling time for each road segment $\tau(r),\ r \in R$ corresponds to $\hat{w} \in \hat{W}$, and let $\mathcal{E}(r^d)$ corresponds to $\hat{B}$. In addition, let the number of tasks a vehicle $v_i$ can take at a road segment, $T'_{i,r} \in \zeta(p_i)$, correspond to $\hat{v} \in \hat{V}$, and let $V_i^{cap} \rightarrow \sum_{r \in p_i} v_i^{cap}$ correspond to $\hat{K}$. Finally, let $\mathcal{E}(s^d) = \hat{B}$ and $V^{cap} = \hat{K}$, we claim that the Knapsack problem has a solution if and only if the instance of VCPP has a solution.

### 6.2.3 Algorithm Design

Unlike a typical vehicular crowdsensing recruitment problem which recruits a set of vehicles to fulfill published sensing tasks, the VCPP problem needs to plan routes for each vehicle to complete more sensing tasks under the constraint that vehicles must still arrive at their destination within a time range. However, according to our observations, planning an entire trip for crowdsensing tasks for each vehicle not only increase computation cost but is also not feasible due to sensing tasks which could be published or have expired in the system at any time. Besides, the vehicle's future position cannot be fully predicted. Another challenge we found during our experiments is that vehicles tend to follow the shortest path to its destination and vehicles stuck in some road segments are difficult to reroute. For instance, vehicles that enter the highway are difficult to redirect to a local street. To solve the challenges mentioned, we divide our solution into two stages: global planning and local planning. The global planning looks into task probabilities and vehicle location probabilities and attempts to spread vehicular agents to minimize the difference between the two distributions. In contrast, local planning looks at its immediate surroundings and attempts to fulfill published tasks. Thus, the global planner looks from a macro level which region requires sensing. The local planner, on the other hand, decides which tasks an agent should undertake immediately.

#### 6.2.3.1 Global Planning

With global planning, we aim to reroute vehicles to popular sensing areas such that popular sensing area can have sufficient sensing participants. To do this, we first uniformly decomposed the city map into set of grids $G \rightarrow \{g_1, ..g_k\}$. Each grid has size $\beta$, where the $\beta$ is a positive number. Furthermore, each grid contains a set of road segments which is denoted as $g^R$. Moreover, each grid has a heat value $g^{heat}$ denoting tasks likelihood within the grid region. The heat values of a grid is calculated by:

$$g^{heat} \rightarrow \frac{\sum_{r \in g^R} |tasks(r)|}{T^{total}} \in [0, 1]$$

where $T^{total}$ is the current total available tasks on the map.

The goal of global planning is to reroute every newly added vehicle to an intermediate point as the following:

$$\textbf{Objective: } min \sum_{g \in G} g^{heat} \textbf{ subject to:}$$

$$\mathbb{G}(v_i) \rightarrow \begin{cases} true & A^*(r_i, r_i^{d'}) + A^*(r_i^{d'}, r_i^d)) \leq \mathcal{E}(r_i^d), \forall v_i \in V^{new} \\ false & \text{otherwise} \end{cases}$$

where $r^{d'}$ is the global planning destination which is a road segment in the grid $g \in G$. The constraints ensure vehicles can still arrive at their final destination on time even when they are rerouted to another location $r^{d'}$ during their journey. Given a set of recently added vehicles $V^{new}$, the algorithm for finding the rerouted destination $r^{d'}$ for each vehicle is shown in algorithm 5.

**Algorithm 5** Global Planning Algorithm (GPA)

**Input** $V^{new}$ and $G$

**for** $v \in V^{new}$ **do**
  $selected\_g \leftarrow \emptyset$
  **for** $g \in G$ **do**
    **if** $\mathbb{G}(v) == true$ and $g^{heat} > selected\_g^{heat}$ **then**
      $selected\_g = g$
    **end if**
  **end for**
  **if** $selected\_g$ is not empty **then**
    $r_v^{d'} \leftarrow r \in selected\_g^R$
    $path_v \leftarrow A^*(r_v^b, r_v^{d'})$
  **end if**
**end for**

#### 6.2.3.2 Local Planning

The local planning algorithm is triggered when vehicle $v_i$ is almost at the global planning destination $r_i^{d'}$ which is calculated by our first stage solution. The local planing algorithm aims to find the best path $p_i^{best}$ such that the vehicle $v_i$ can

reach its final destination $r_i^d$ before it's expected arrival time while maximizing the number of sensing tasks completed. Since we proved that finding the optimal solution for $p^{best}$ is NP-Hard, we present a greedy approach to approximate the solution of our local planning problem.

To explain our solutions, we use the following definitions:

**Definition 12.** ***Road Cost:*** *Since our goal is to schedule a path for a vehicle to not only arrive at its destination but also maximize the number of sensing tasks completed, our cost function combines both travelling time and tasks taken for a single road is described as the follow:*

$$\phi(r^{next}, r_i^d) \rightarrow \frac{\tau(r^{next})}{\mathcal{E}(r_i^d) - (\theta_i + dist(r_i^b, r^{next}))} \times \mathbb{N}(|tasks(r^{next})|)^{-1} \tag{6.13}$$

where, the function $\mathbb{N} \rightarrow [0, 1]$ is our normalization function.

The motivation behind this cost function is to enforce a soft penalty upon road segments which has high average traveling time. Vehicular participants which are more willing to reroute have greater tolerance to a road segment which requires more travelling time.

**Definition 13.** ***Task Selection*** *When a vehicular participant enters a road segment, the vehicle will try to pick as many tasks as possible to reach its maximum capacity $v^{cap}$. Vehicles pick tasks following the earliest deadline first manner. Thus, the set of tasks $t$ that vehicle $v_i$ at road segment $s$ can select is define by:*

$$v_i^{t,s} \rightarrow \{t | \forall t \in tasks(r), minimum\ t^d\}\ subject\ to\ |v_i^{t,s}| \leq |v^{cap}|$$

The Local Planning Algorithm (LPA), which is described in algorithm 6, shows how our cost function is leveraged for finding the $p^{best}$ while taking into consideration total traveling time and number of tasks taken. In each round, LPA picks the next road segment to extend its route along the route to vehicle $v_i$'s destination $r_i^d$. Specifically, LPA continuously selects the next road segment which minimizes $f(r^{next}) \leftarrow \Phi(r^{next}, r_i^d) + h(r^{next})$, where $r^{next}$ is the next road segment, $\Phi(r^{next}, r_i^d) \rightarrow \sum_{r_r^b} \phi(r, r_i^d)$ is the cost of the route from the initial road segment to

**Algorithm 6** Local Planning Algorithm (LPA)

**Input** $r^b$, $r^d$, and $v_i$
**Output** $p^{best}$

1: $Open \leftarrow \emptyset$
2: $Close \leftarrow \emptyset$
3: $Open.add(r^b)$
4: **while** $Open$ is not empty **do**
5: $r' \leftarrow$ the lowest f value in $Open$
6: $Open.remove(r')$
7: $Close.add(r')$
8: $R^{children} \leftarrow getAllChildren(r')$
9: **for** $r^{child} \in R^{children}$ **do**
10: **if** $r^{child} == r^d$ **then**
11: $p^{best} \leftarrow constructPath(r^{child})$
12: **return** $p^{best}$
13: **end if**
14: $f(r^{child}) \leftarrow \Phi(r^{child}, r_i^d) + h(r^{child})$
15: **if** $r^{child} \cap Close == \emptyset$ **then**
16: $Open.add(r^{child})$
17: **end if**
18: **end for**
19: **end while**
20: **return** $\emptyset$

the next road segment. $h(r^{next})$ is a heuristic function which estimates the cost of route from $r^{next}$ to the destination $r^d$. We use Manhattan distance for our heuristic function. Lastly, LPA terminates when reaching the $r_i^d$ or if there are no available next road segments.

LPA can find a path for individual vehicles to maximize the number of sensing tasks while constrained by the arrival deadline. However, with multiple vehicles, a future sensing task might be taken away by other vehicles before vehicle $v_i$ arrives at the sensing location. One solution is to apply some rescheduling mechanism where rescheduling is triggered when the difference between expected and taken tasks are below a certain threshold. Rescheduling is triggered by:

$\frac{T'_{i,r}-T^{actual}_{i,r}}{\zeta(r_i)} \leq \lambda \in [0,1]$, where $T'_{i,r}$ are the expected tasks received at road segment $r$, $T^{actual}_{i,r}$ are the exact tasks vehicle $v_i$ received at the road segment $r$, and $\lambda \in [0,1]$ as the threshold.

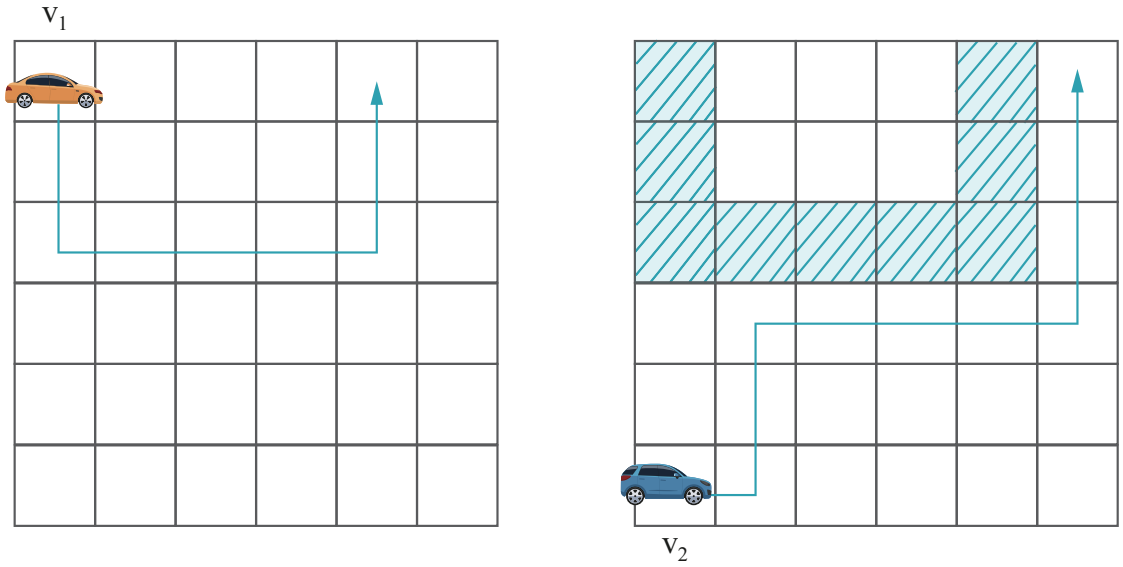


**Fig. 6.6** Cooperative $A^*$ algorithm by David Silver: vehicles reserve their path a head of time. The reservations are managed in a first come first serve manner.

Another solution is to apply some task reservation mechanisms to the system to ensure the task can be assigned to the scheduled participant. Inspired from the Cooperative $A^*$ algorithm by David Silver [186], given using a reservation table to store path for all vehicles as shown in Figure 6.6; we introduce a task reservation mechanism that allows a sensing participant to reserve sensing tasks ahead of time as shown in Figure 6.7. However, since we cannot guarantee vehicles will always arrive on time, reserving a set of tasks for a specific vehicle may fail all the tasks if the vehicle cannot arrive on time. Thus, instead of using hard tasks reservation, we introduce a soft tasks reservation which leverages probabilistic reservation.

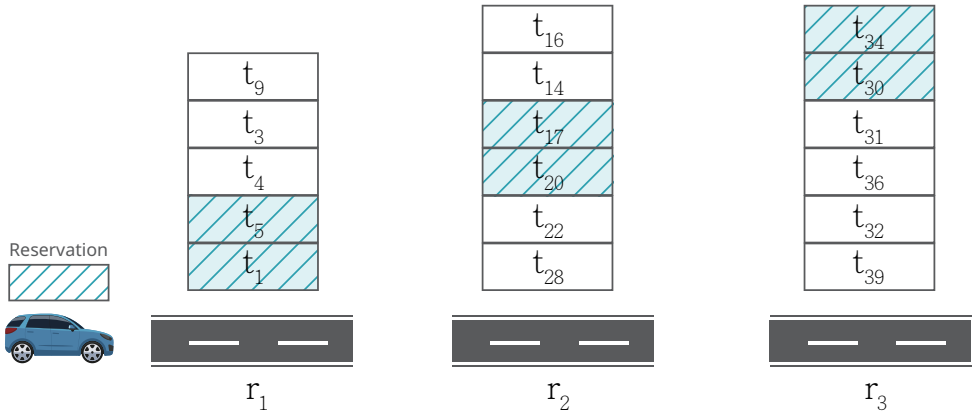


**Fig. 6.7** Task reservation mechanism to ensure the sensing task can be assigned to the scheduled vehicles

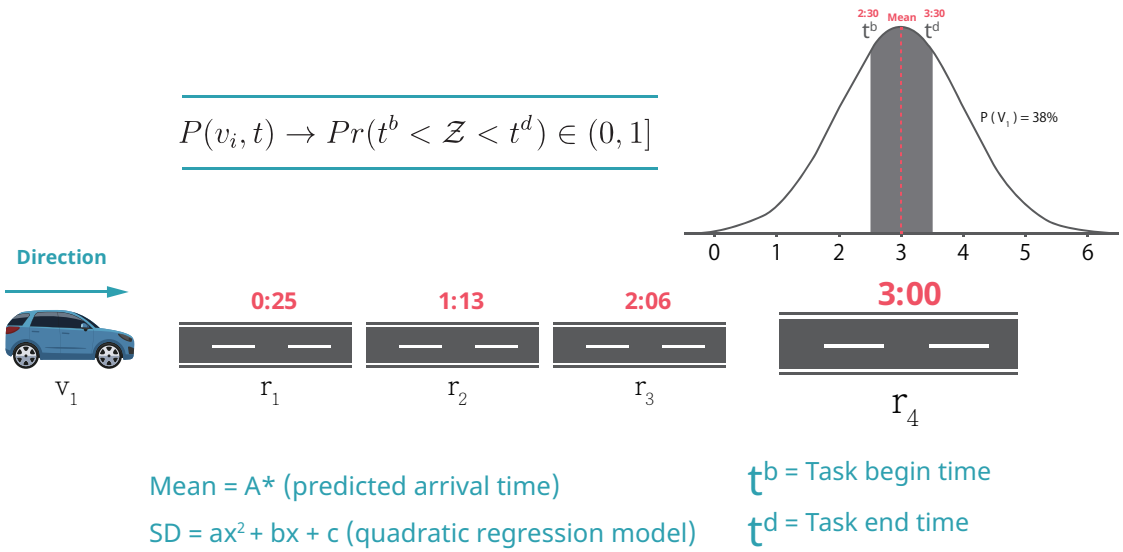


**Fig. 6.8** We use our vehicle arrival time estimation to update the task completion probability

**Definition 14.** ***Task completion probability:*** *We define task completion probability as the probability a task is completed on time by a set of vehicles. Thus, a vehicular participant which cannot arrive on time before a task expires can be replaced by another vehicle. Task completion rate is dependent on the accuracy of predicted arrival time. A study has shown that vehicle arrival times follows a normal distribution [196]. As such, the probability of a vehicle $v_i$ arriving at a road segment can be obtained from the following function using the z table: $\mathcal{Z} = \frac{X - \mathcal{A}(v_i, r)}{\sigma}$, where $\sigma$ is the standard deviation. For selecting the value $\sigma$, we evaluated predicted trip error using the TAPAS Cologne vehicular trace. As shown in Figure 3.5, as the trip length increases, the expected deviation from predicted arrival time increases quadratically. Thus, our $\sigma$ increases based on trip time. Figure 6.8 shows more detail how we calculate the task completion rate.*

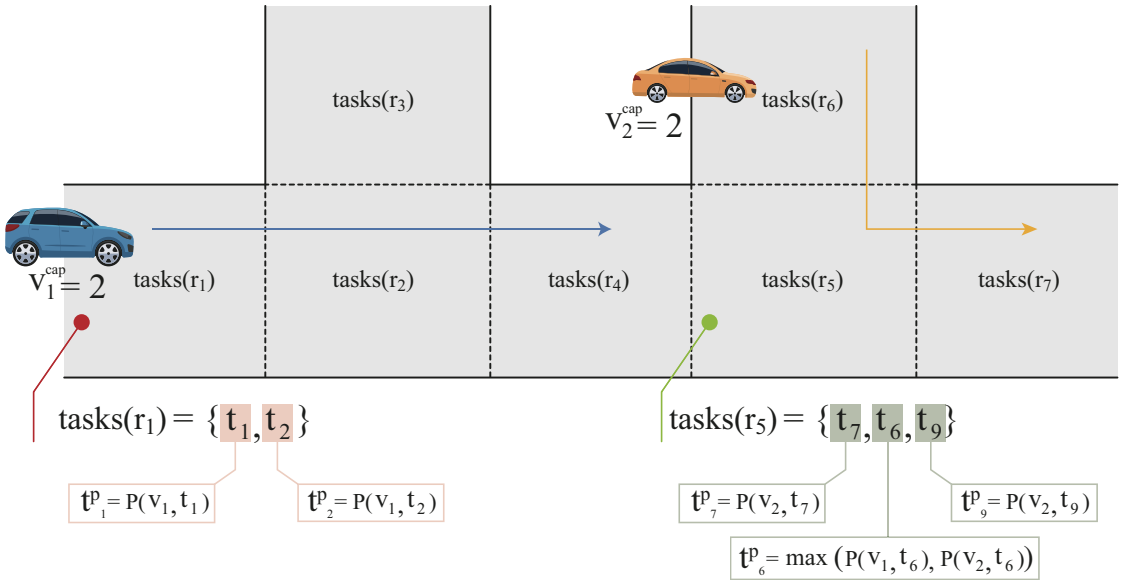


**Fig. 6.9** The task completion probability is updated based on the length of route.

*When the path $p_i^{best}$ is scheduled, we must update all associated completion probabilities for each task on the route. Since a road segment can have many sensing tasks, we can only select a subset of tasks $\check{T} \in tasks(r)$, where $\check{T} \leq v^{cap}$ and $r \in p_i^{best}$. Let $t^p$ denote the completion probability of the task $t$. The selection of $\check{T}$ is based on:*

$$\check{T} \rightarrow \{t | minimum\ t^p, \forall t \in tasks(r)\}$$

*Tasks which have the lowest completion probability have higher priority for selection. When a task is selected by a vehicle, its completion probability $t^p$ must be increased. Thus the probability of vehicle $v_i$ arriving at the sensing location before the deadline of the sensing task $t \in \check{T}_i$ is the following:*

$$P(v_i, t) \rightarrow Pr(t^b < \mathcal{Z} < t^d - t^{cost}) \in (0, 1] \tag{6.14}$$

*However, a single task might be selected by multiple vehicular participants, thus we update the task probability by the following function:*

$$t^p \leftarrow max(P(v_1, t), P(v_2, t), ...P(v_i, t))$$

The update function of task probability is shown in Figure 6.9

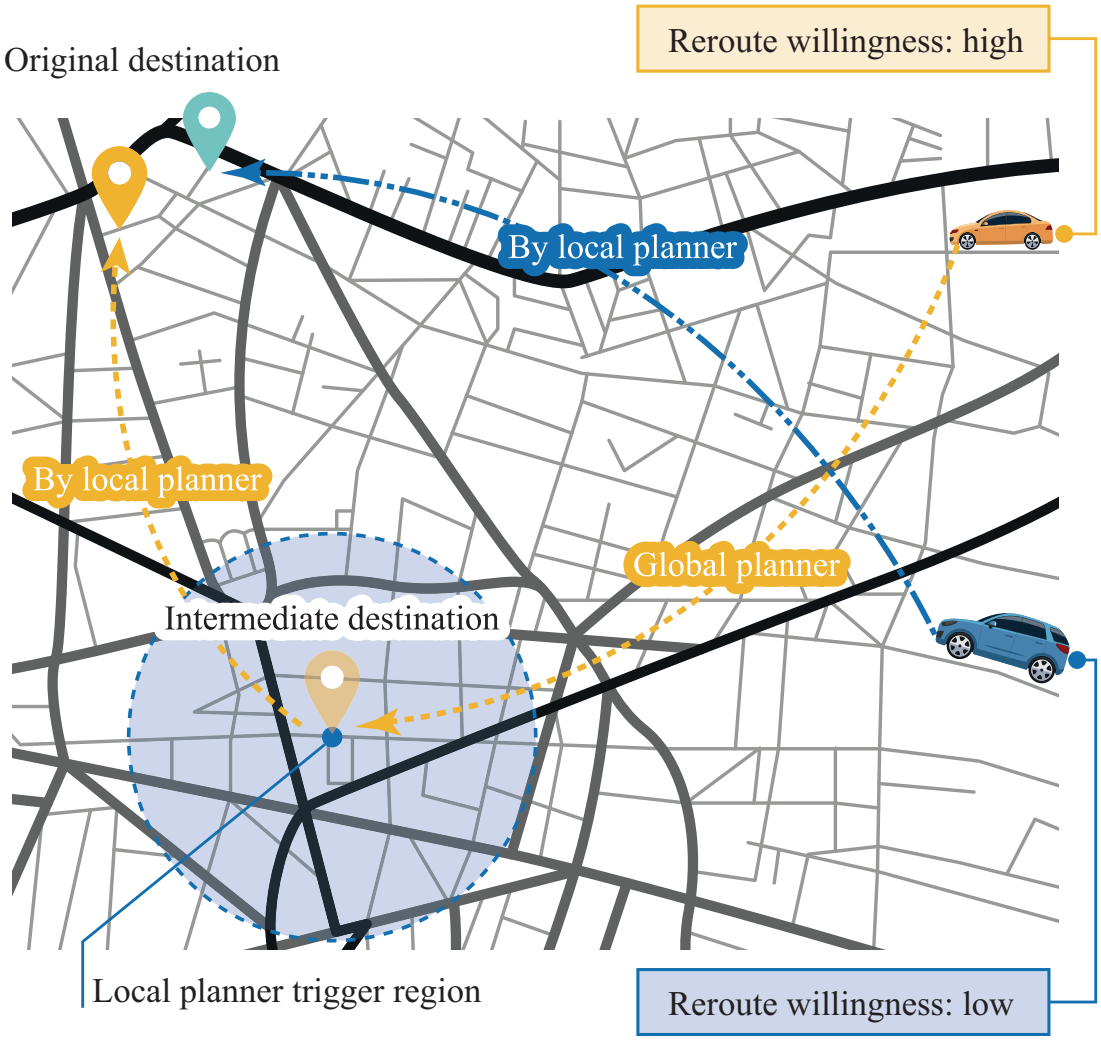


**Fig. 6.10** Global Planning first reroutes the vehicle to an intermediate destination. Then, Local Planning will be triggered to find a route to the original destination after the vehicle near the intermediate destination.

**Definition 15.** ***Road Cost With Task Completion Probability:*** *The proposed road cost considers the task completion probability. This means that a road segment which has low average task completion probability is more likely to be selected by the vehicular participant. Thus the road cost* $\widehat{\phi}(r^{next}, r_i^d)$ *is shown as the follow:*

$$\widehat{\phi}(r^{next}, r_i^d) \rightarrow \frac{\tau(r^{next})}{\mathcal{E}(r_i^d) - (\theta_i + dist(r_i^b, r_i^{next}))} \times \frac{\sum_{t \in tasks(r^{next})} t^p}{|tasks(r^{next})|}$$

Given road cost function $\widehat{\phi}(r, r_i^d)$, we can change our route selection function to the following: $f(r^{next}) \leftarrow \widehat{\Phi}(r^{next}, r_i^d) + h(r^{next})$ and replace the function in the algorithm 6 on line 14.

#### 6.2.3.3 Global Planning + Local Planning

In this section, we describe how our global planning and local planning performs in symbiosis. As shown in Figure 6.10, when a vehicle enters the system, it will be scheduled by the global planner. If the vehicle cannot be rerouted by the global planner due to limited rerouting time $\gamma(v)$, the local planner will be triggered immediately. Otherwise, the vehicle will be rerouted to a destination $r^{d'}$ chosen by the global planner. When the vehicle is within a certain threshold to $r^{d'}$, the local planner will be triggered. Moreover, vehicles moving on its path will greedily pull the sensing tasks on each road segment it passes. Thus, for the task reservation, the global planner further updates the associated task probability $t^p$ when a route is scheduled. Figure 6.11 is the flowchart of our global and local planner system.

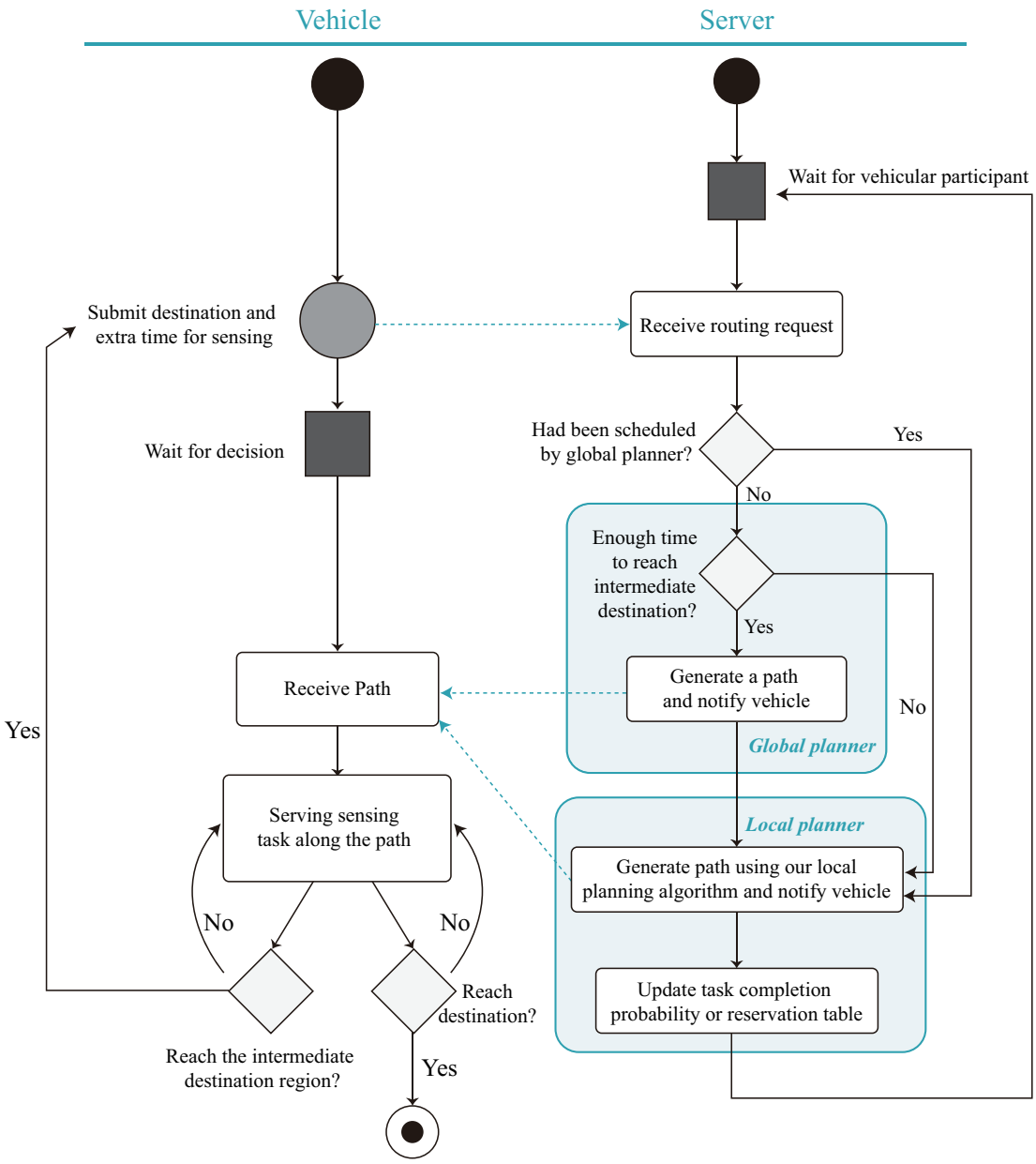


**Fig. 6.11** Global planning and Local planning

### 6.2.4 Evaluation

In this section, we evaluate the performance of our proposed solution by utilizing TAPAS Cologne, one of the largest traffic simulation datasets. It consists of data simulated over 24 hours for the city of Cologne, Germany. It covers over 400 square kilometers [197]. To reduce simulation time, we restricted the dataset to a data subset which consists of 7200-time steps from 6 AM to 8 AM. During this period, about 34000 unique vehicles were part of the simulation. To obtain the trace of vehicle positions as a predicted path, we utilized the SUMO traffic simulator [167], and we built our system on top of this simulator. SUMO updates the average speed of each road segment. Thus, we calculated the future position of a vehicle based on the average speed of roads.

#### 6.2.4.1 Simulation Setup

We generate tasks based on the two demands types: static and dynamic tasks. The generation functions are described in definition 11. For dynamic tasks, we randomly select a subset of vehicles as clients from the 34000 unique vehicles. We utilize uniform random distribution to select 10000 clients where each client's sensing route ranges from 1-125 road segments. When clients begin their journey, clients send their future sensing route and sensing objective to the server. The system then schedules participants to cover clients' sensing routes in a first-in-first-serve manner. A vehicular client cannot be the participant for its own task but can be a participant for other tasks.

For parameter settings, we set the task deadline and the task execution cost interval to range from 300 to 1800 seconds which is based on the average traveling time of different road segments. We also set $v^{cap}$ to range from 2 to 8 sampled uniformly. We evaluated the performance of our algorithms by comparing with $A^*$ baseline which computes the fastest route to the destination without considering the task completion rate, and $CA^\star$ (Cooperative $A^*$) which is a reservation-based multi-agent route planning algorithm [186].

#### 6.2.4.2 Result: Task Completion Rate

Here, we demonstrate the performance of our algorithms compared to existing solutions. We utilize task completion rate as the metric to evaluate the performance. The task completion rate is defined as follows:

$$completion\ rate = \frac{T^{complete}}{T}, T^{complete} \subseteq T$$

where $T$ is the total number of tasks on the map since the start of the simulation.

As shown in Figure 6.12, $CA^{\star}$ performs worst out of all algorithms, for 80% of its tasks, the completion rate is at most 50%, compared to *G+L Task Probability* and *G+L Task-Reschedule* which performs best with only 40% of its tasks reaching at most 50% completion rate. This means that 60% of its tasks have a completion rate higher than 50%. In contrast, our baseline has 70% of its tasks having at most 50% completion rate and only 30% of its tasks with higher than 50% completion rate.

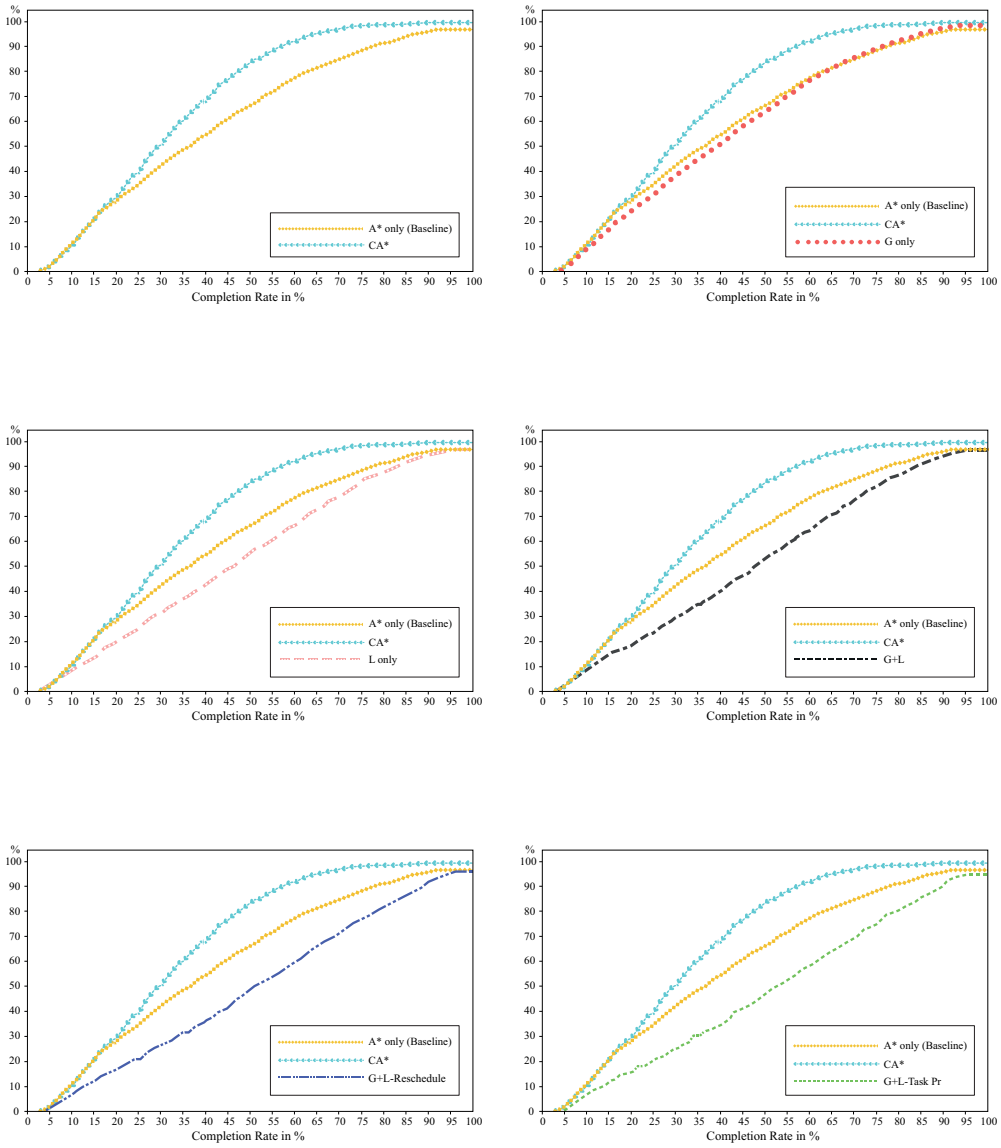


**Fig. 6.12** Completion rate results are shown in Cumulative Distribution Function (CDF) graph. $G = GPA$, $L = LPA$, and $completion\ rate = \frac{T^{complete}}{T}, T^{complete} \subseteq T$

$CA^{\star}$ performs worse than the baseline because it reserves tasks ahead of time, but due to arrival time estimation errors, those tasks cannot be completed and in turn, fail. However, the *G only* does slightly better than the baseline because vehicles are rerouted to the region that has most tasks. This will increase the chance of utilizing the available vehicles' sensing capacities.

Furthermore, our local approach greatly boosts performance. This is because of the availability of Spatio-temporal tasks that are the key selection criteria for the local route planning [198]. Vehicles are greedily finding a path that has the potential to serve more tasks. However, having local planning only without global planning will quickly hit a plateau after tasks in a region have been taken by too many vehicles. As seen in Figure 6.12, the combined planner shows an improved performance compared to only the local planner. This is because the global planner spreads vehicular participants to minimize the difference between task probabilities and vehicle availability. Thereby making the local planner able to find sufficient tasks nearby once arrived.

Comparing all the *G+L* approaches, *G+L reschedule* and *G+L Task Probability* outperforms the *G+L* only solution. This is because the predicted arrival time for future positions can be very error-prone especially when we have a lengthy path. Furthermore, with a multi-vehicle system, the future sensing task might be taken away by other vehicles before the vehicle arrives at the sensing location. Thus, having *G+L* only without considering the above issue will end up with a performance bottleneck. Finally, our *G+L reschedule* and *G+L Task Probability* perform at similar levels. The reschedule approach performs well as it readjusts to the current situation but suffers from heavy rescheduling overheads. In contrast, *G+L Task Probability* is an estimate-based approach of rescheduling with probability and performs at similar levels but with much lower overheads.

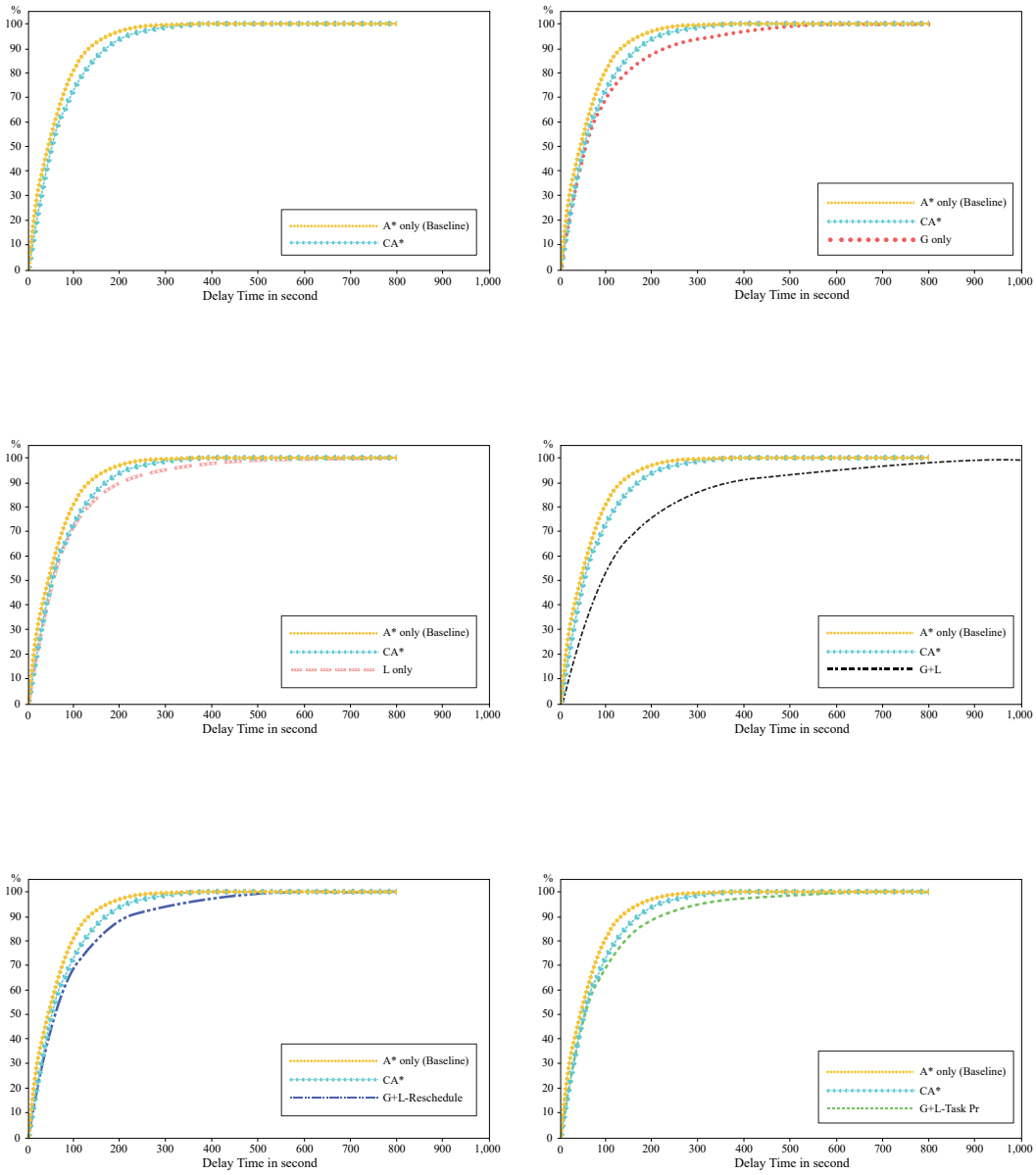


**Fig. 6.13** CDF of arrival time delay to destination in seconds.

#### 6.2.4.3 Result: Arrival Delay

Next, we consider the arrival delay time of all algorithms under the same conditions. We compare the amount of arrival delay suffered by participants as a CDF graph as shown in Figure 6.13. As expected, the baseline performs best as it finds the faster route without rerouting using $A^*$, followed by $CA^\star$ which is slightly worse. The solution with the global planner adds little arrival delay to vehicles compared to the other solutions which do not have. The only algorithm which performs poorly is the *G+L* caused by a combination of global and local rerouting without task reservation. This is because when a task is not reserved, both the global and local planners will continuously send vehicles to the sensing region even when the region had been scheduled and tasks had been reserved by other vehicles. This leads to an increase in traffic congestion and results in a delay to the arrival time at the original destination. Finally, for task completion rate and arrival delay time, *G+L Task Probability* performs best in terms of overall task completion, arrival delay, and low overheads compared to the other approaches.

# Chapter 7

# Conclusion And Future Work

## 7.1 Conclusion

In this thesis, we explore crowdsensing for data acquisition. As part of this we propose a user-defined crowdsensing paradigm for vehicles. Existing approaches only focus on large scale sensing unsuitable for personalized sensing tasks. We believe our solution, given its focus on efficiency and low budget requirements, can open crowdsensing for the individual. Rather than focus on sensing coverage on a massive scale, users can choose exactly what they need. Each sensing task runs on the computing resources of the participants, reducing the need for server computation. The server only would need to schedule participants and pass information between vehicles. However, since participants have limited computing resources, we included load balancing so no single vehicle is swamped with requests.

For participant requirements solutions, we first design and evaluate two participant recruitment systems, push and pull, for vehicular crowdsensing. In our push system, the central server schedules sensing tasks based on predicted routes of participants. One problem we faced in a push-based system is position prediction error, which can lead to high sub-task failure rate. Thus, we proposed window-based scheduling to improve the sub-task failure rate. Furthermore, we proposed a dynamic window-based scheduling algorithm where the size of the window is adjusted based on the sub-task coverage rate to adjust the size of the scheduling window. This allows for reducing the scheduling frequency, and the overhead of

the scheduler. In contrast, our pull-based system allocates sensing tasks into sub-tasks and sends them to fog nodes. Participants can then pull tasks when they are close to the sensing location.

Contrasting the performance of both algorithms, we find push required fewer participants per sensing task. However, it suffered from a greater sub-task failure rate. This is because the pull-based approach has greater tolerance to rerouting errors. After all, participants are only allowed to pull the sub-task when they are close by. Besides, the pull-based solution enables the participant to choose what they sense. The decision to participate depends on the drivers; drivers who do not want to sense can refuse to pick any task unlike in push. However, the pull-based solution requires responsive servers for managing and updating the sub-tasks. Thus, for an area with unstable network connection, participants will fail to pull sub-task even when they are prioritized participants, thus increasing the sub-task failure rates.

Additionally, we proposed a route planning based approach to improve vehicular crowdsensing coverage. We have shown that vehicles with path planning in the crowdsensing system's improved in performance. We also show that with a combined global and local planner, we can obtain reasonable coverage performance with limited overheads compared to existing planning approaches. We believe that having such a solution, given the focus on serving more sensing tasks, can open crowdsensing for more user-specific use. Rather than focus on sensing coverage on a massive scale, users can choose exactly what they need. Each sensing task runs on the computing resources of the participants, reducing the need for server computation. The server only needs to schedule participants and pass limited information between vehicles.

## 7.2 Future Work

As future work, we would like to test our approach with a small fleet of vehicles. A real implementation of our system tests its robustness and reveals practical problems with our design. Furthermore, we would like to explore a reward scheme specifically created for personalized sensing tasks. That is, an improved reward system to help increase the sensing participation rate. Furthermore, we believe communication delays between vehicles and servers should be tested as tasks require lightning-fast response times. Thus, having a real implementation that considers both Wi-Fi and cellular network in our system is also a future goal.

To improve vehicular path planning, we also would like to explore various path finding heuristics to boost the performance of the solution. A road network has unique constraints such as traffic conditions, road lanes, traffic direction, speed, etc. We may need to have a specific pathfinding heuristic for such a network. Additionally, we are also interested in exploring vehicular travel time prediction algorithms to have more accurate estimated arrival times, which can improve the performance of our system.

# References

[1] B. Guo, H. Chen, Z. Yu, X. Xie, S. Huangfu, and D. Zhang, "Fliermeet: A mobile crowdsensing system for cross-space public information reposting, tagging, and sharing," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2020–2033, 2015.

[2] R. Pryss, M. Reichert, B. Langguth, and W. Schlee, "Mobile crowd sensing services for tinnitus assessment, therapy, and research," in *2015 IEEE International Conference on Mobile Services*, 2015, pp. 352–359.

[3] M. Pouryazdan, B. Kantarci, T. Soyata, and H. Song, "Anchor-assisted and vote-based trustworthiness assurance in smart city crowdsensing," *IEEE Access*, vol. 4, pp. 529–541, 2016.

[4] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 2542–2550.

[5] K. Yi, R. Du, L. Liu, Q. Chen, and K. Gao, "Fast participant recruitment algorithm for large-scale vehicle-based mobile crowd sensing," *Pervasive and Mobile Computing*, vol. 38, Part 1, pp. 188 – 199, 2017.

[6] K. Han, C. Chen, Q. Zhao, and X. Guan, "Trajectory-based node selection scheme in vehicular crowdsensing," in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, Nov 2015, pp. 1–6.

[7] X. Zhang, Z. Yang, Y. Liu, J. Li, and Z. Ming, "Toward efficient mechanisms for mobile crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1760–1771, 2017.

[8] X. Wang, W. Wu, and D. Qi, "Mobility-aware participant recruitment for vehicle-based mobile crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4415–4426, May 2018.

[9] T. Ludwig, C. Reuter, T. Siebigteroth, and V. Pipek, "Crowdmonitor: Mobile crowd sensing for assessing physical and digital activities of citizens during emergencies," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 4083–4092. [Online]. Available: https://doi.org/10.1145/2702123.2702265

[10] F. J. Villanueva, D. Villa, M. J. Santofimia, J. Barba, and J. C. López, "Crowdsensing smart city parking monitoring," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 751–756.

[11] D. Zhao, H. Ma, L. Liu, and X.-Y. Li, "Opportunistic coverage for urban vehicular sensing," *Computer Communications*, 2015.

[12] L. Shao, C. Wang, Z. Li, and C. Jiang, "Traffic condition estimation using vehicular crowdsensing data," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, Dec 2015, pp. 1–8.

[13] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772–782, June 2017.

[14] C. Wang, Z. Zhang, L. Shao, and M. Zhou, "Estimating travel speed via sparse vehicular crowdsensing data," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec 2016, pp. 643–648.

[15] J. Ballesteros, M. Rahman, B. Carbunar, and N. Rishe, "Safe cities. a participatory sensing approach," in *37th Annual IEEE Conference on Local Computer Networks*, Oct 2012, pp. 626–634.

[16] N. Maisonneuve, M. Stevens, M. E. Niessen, e. I. N. Steels, Luc", A. E. Rizzoli, P. A. Mitkas, and J. M. Gómez, *NoiseTube: Measuring and mapping*

*noise pollution with mobile phones*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 215–228.

[17] M. Elhamshary, M. Youssef, A. Uchiyama, H. Yamaguchi, and T. Higashino, "Transitlabel: A crowd-sensing system for automatic labeling of transit stations semantics," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 193–206. [Online]. Available: https://doi.org/10.1145/2906388.2906395

[18] H. Aly, A. Basalamah, and M. Youssef, "Map++: A crowd-sensing system for automatic map semantics identification," in *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2014, pp. 546–554.

[19] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 183–196. [Online]. Available: https://doi.org/10.1145/2809695.2809719

[20] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1535–1545, 2018.

[21] T. Y. Yu, X. Zhu, and H. Chen, "Gosense: Efficient vehicle selection for user defined vehicular crowdsensing," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, June 2017, pp. 1–5.

[22] Google, "Waze mobile," 2017, https://www.waze.com/.

[23] K. Han, C. Zhang, and J. Luo, "Taming the uncertainty: Budget limited robust crowdsensing through online learning," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1462–1475, 2016.

[24] L. Wang, D. Zhang, Z. Yan, H. Xiong, and B. Xie, "effsense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 12, pp. 1549–1563, 2015.

[25] M. Xiao, J. Wu, H. Huang, L. Huang, and C. Hu, "Deadline-sensitive user recruitment for mobile crowdsensing with probabilistic collaboration," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, Nov 2016, pp. 1–10.

[26] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 392–403, 2017.

[27] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *2012 IEEE International Conference on Pervasive Computing and Communications*, March 2012, pp. 103–108.

[28] H. Amintoosi, S. S. Kanhere, and M. N. Torshiz, "A socially-aware incentive scheme for social participatory sensing," in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, April 2015, pp. 1–6.

[29] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 2016, pp. 344–353.

[30] K. Ota, M. Dong, J. Gui, and A. Liu, "Quoin: Incentive mechanisms for crowd sensing networks," *IEEE Network*, vol. 32, no. 2, pp. 114–119, 2018.

[31] M. K. Chen, "Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform," 2016.

[32] T. Yu, X. Zhu, and M. Maheswaran, "Push vs pull participant recruitment system for personalized vehicular crowdsensing," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[33] X. Zhu, S. A. Samadh, and T. Yu, "Large scale active vehicular crowdsensing," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug 2018, pp. 1–5.

[34] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspectives on psychological science*, vol. 6, no. 1, pp. 3–5, 2011.

[35] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. S. Yilmaz, and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using twitter," in *2010 IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, June 2010.

[36] B. Guo, Z. Yu, D. Zhang, and X. Zhou, "From participatory sensing to mobile crowd sensing," *CoRR*, vol. abs/1401.3090, 2014. [Online]. Available: http://arxiv.org/abs/1401.3090

[37] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: A crowdsourced traffic regulator detection and identification system," in *Information Processing in Sensor Networks (IPSN), 2013 ACM/IEEE International Conference on*, April 2013, pp. 331–332.

[38] E. Aubry, T. Silverston, A. Lahmadi, and O. Festor, "Crowdout: A mobile crowdsourcing service for road safety in digital cities," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, March 2014, pp. 86–91.

[39] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proceedings of the 2Nd Annual International Workshop on Wireless Internet*, ser. WICON '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1234161.1234179

[40] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, no. 5, pp. 36–44, Sept 2012.

[41] X. Hu, X. Li, E. C. . Ngai, V. C. M. Leung, and P. Kruchten, "Multidimensional context-aware social network architecture for mobile crowdsensing," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 78–87, 2014.

[42] G. Merlino, S. Arkoulis, S. Distefano, C. Papagianni, A. Puliafito, and S. Papavassiliou, "Mobile crowdsensing as a service: A platform for applications on top of sensing clouds," *Future Generation Computer Systems*, vol. 56, pp. 623 – 639, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X15002976

[43] J. An, X. Gui, Z. Wang, J. Yang, and X. He, "A crowdsourcing assignment model based on mobile crowd sensing in the internet of things," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 358–369, 2015.

[44] K. Han, C. Zhang, J. Luo, M. Hu, and B. Veeravalli, "Truthful scheduling mechanisms for powering mobile crowdsensing," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 294–307, 2016.

[45] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," *Center for Embedded Network Sensing*, 2006.

[46] B. Guo, C. Chen, D. Zhang, Z. Yu, and A. Chin, "Mobile crowd sensing and computing: when participatory sensing meets participatory social media," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 131–137, 2016.

[47] C. Zhang, K. P. Subbu, J. Luo, and J. Wu, "Groping: Geomagnetism and crowdsensing powered indoor navigation," *IEEE Transactions on Mobile Computing*, vol. 14, no. 2, pp. 387–400, 2015.

[48] G. Cardone, A. Cirri, A. Corradi, and L. Foschini, "The participact mobile crowd sensing living lab: The testbed for smart cities," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 78–85, 2014.

[49] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 29–35, 2014.

[50] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, 2016.

[51] H. Li, K. Ota, M. Dong, and M. Guo, "Mobile crowdsensing in software defined opportunistic networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 140–145, 2017.

[52] V. Petrov, A. Samuylov, V. Begishev, D. Moltchanov, S. Andreev, K. Samouylov, and Y. Koucheryavy, "Vehicle-based relay assistance for opportunistic crowdsensing over narrowband iot (nb-iot)," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3710–3723, 2018.

[53] I. J. Vergara-Laurens, L. G. Jaimes, and M. A. Labrador, "Privacy-preserving mechanisms for crowdsensing: Survey and research challenges," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 855–869, 2017.

[54] J. Fan, Q. Li, and G. Cao, "Privacy-aware and trustworthy data aggregation in mobile sensing," in *2015 IEEE Conference on Communications and Network Security (CNS)*, Sept 2015, pp. 31–39.

[55] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, "Participant privacy in mobile crowd sensing task management: A survey of methods and challenges," *SIGMOD Rec.*, vol. 44, no. 4, p. 23–34, May 2016. [Online]. Available: https://doi.org/10.1145/2935694.2935700

[56] H. Xiong, D. Zhang, L. Wang, J. P. Gibson, and J. Zhu, "Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, Apr. 2015. [Online]. Available: https://doi.org/10.1145/2644827

[57] L. Kong, L. He, X. Liu, Y. Gu, M. Wu, and X. Liu, "Privacy-preserving compressive sensing for crowdsensing based trajectory recovery," in *2015 IEEE*

*35th International Conference on Distributed Computing Systems*, 2015, pp. 31–40.

[58] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16. New York, NY, USA: ACM, 2016, pp. 341–350. [Online]. Available: http://doi.acm.org/10.1145/2942358.2942375

[59] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, 2017.

[60] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2958–2970, 2018.

[61] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proceedings of the 26th International Conference on World Wide Web*, ser. WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 627–636. [Online]. Available: https://doi.org/10.1145/3038912.3052696

[62] Y. Yao, L. T. Yang, and N. N. Xiong, "Anonymity-based privacy-preserving data reporting for participatory sensing," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 381–390, Oct 2015.

[63] Y. Zhang, Q. Chen, and S. Zhong, "Privacy-preserving data aggregation in mobile phone sensing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 980–992, May 2016.

[64] E. De Cristofaro and C. Soriente, "Extended capabilities for a privacy-enhanced participatory sensing infrastructure (pepsi)," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2021–2033, Dec 2013.

[65] E. De Cristofaro and R. Pietro, "Adversaries and countermeasures in privacy-enhanced urban sensing systems," *IEEE Systems Journal, Special Issue on Security and Privacy of Complex Systems*, vol. 7, 01 2012.

[66] Q. Li, G. Cao, and T. F. L. Porta, "Efficient and privacy-aware data aggregation in mobile sensing," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 2, pp. 115–129, March 2014.

[67] L. Zhang, X. Wang, J. Lu, P. Li, and Z. Cai, "An efficient privacy preserving data aggregation approach for mobile sensing," *Security and Communication Networks*, vol. 9, no. 16, pp. 3844–3853, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1546

[68] G. Han, L. Liu, S. Chan, R. Yu, and Y. Yang, "Hysense: A hybrid mobile crowdsensing framework for sensing opportunities compensation under dynamic coverage constraint," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 93–99, 2017.

[69] H. Xiong, Y. Huang, L. E. Barnes, and M. S. Gerber, "Sensus: A cross-platform, general-purpose system for mobile crowdsensing in human-subject studies," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 415–426. [Online]. Available: https://doi.org/10.1145/2971648.2971711

[70] S. K. Datta, R. P. Ferreira da Costa, C. Bonnet, and J. Härri, "onem2m architecture based iot framework for mobile crowd sensing in smart cities," in *2016 European Conference on Networks and Communications (EuCNC)*, 2016, pp. 168–173.

[71] B. Guo, Q. Han, H. Chen, L. Shangguan, Z. Zhou, and Z. Yu, "The emergence of visual crowdsensing: Challenges and opportunities," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2526–2543, 2017.

[72] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Network*, vol. 32, no. 4, pp. 54–60, 2018.

[73] M. Marjanović, A. Antonić, and I. P. Žarko, "Edge computing architecture for mobile crowdsensing," *IEEE Access*, vol. 6, pp. 10 662–10 674, 2018.

[74] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 581–594, 2020.

[75] A. Antonic, K. Roankovic, M. Marjanovic, K. Pripuic, and I. P. arko, "A mobile crowdsensing ecosystem enabled by a cloud-based publish/subscribe middleware," in *2014 International Conference on Future Internet of Things and Cloud*, 2014, pp. 107–114.

[76] S. Eichler, "Performance evaluation of the ieee 802.11p wave communication standard," in *Vehicular Technology Conference*, 2007.

[77] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of network and computer applications*, vol. 37, pp. 380–392, 2014.

[78] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of vanet clustering techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 657–681, 2017.

[79] T. Yu, X. Zhu, H. Chen, and M. Maheswaran, "Hetcast: Cooperative data delivery on cellular and road side network," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–6.

[80] S. Ucar, S. C. Ergen, and O. Ozkasap, "Vmasc: Vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 2381–2386.

[81] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "Lte for vehicular networking: a survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, May 2013.

[82] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec 2013.

[83] S. Dmitriev, "Autonomous cars will generate more than 300 tb of data per year," 2020, https://www.tuxera.com/blog/autonomous-cars-300-tb-of-data-per-year/.

[84] D. Zhao, H. Ma, and L. Liu, "Energy-efficient opportunistic coverage for people-centric urban sensing," *Wirel. Netw.*, vol. 20, no. 6, pp. 1461–1476, Aug. 2014. [Online]. Available: http://dx.doi.org/10.1007/s11276-014-0687-0

[85] D. Zhao, H. Ma, L. Liu, and J. Zhao, "On opportunistic coverage for urban sensing," in *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, Oct 2013, pp. 231–239.

[86] G. Cardone, L. Foschini, P. Bellavista, A. Corradi, C. Borcea, M. Talasila, and R. Curtmola, "Fostering participaction in smart cities: a geo-social crowdsensing platform," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 112–119, June 2013.

[87] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause, "The next big one: Detecting earthquakes and other rare events from community-based sensors," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, April 2011, pp. 13–24.

[88] W. Zhang, B. Zhu, L. Zhang, J. Yuan, and I. You, "Exploring urban dynamics based on pervasive sensing: Correlation analysis of traffic density and air quality," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, July 2012, pp. 9–16.

[89] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *2014 IEEE 15th International Conference on Mobile Data Management*, vol. 1, 2014, pp. 73–82.

[90] S. He, D. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 745–753.

[91] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4w1h in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 42–48, 2014.

[92] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, Sept 2010.

[93] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 403–414. [Online]. Available: https://doi.org/10.1145/2971648.2971709

[94] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "Ccs-ta: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 683–694. [Online]. Available: https://doi.org/10.1145/2750858.2807513

[95] X. Duan, C. Zhao, S. He, P. Cheng, and J. Zhang, "Distributed algorithms to compute walrasian equilibrium in mobile crowdsensing," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4048–4057, 2017.

[96] G. Cardone, A. Cirri, A. Corradi, L. Foschini, R. Ianniello, and R. Montanari, "Crowdsensing in urban areas for city-scale mass gathering management: Geofencing and activity recognition," *IEEE Sensors Journal*, vol. 14, no. 12, pp. 4185–4195, 2014.

[97] G. Cardone, A. Corradi, L. Foschini, and R. Ianniello, "Participact: A large-scale crowdsensing platform," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 1, pp. 21–32, 2016.

[98] A. Antonić, V. Bilas, M. Marjanović, M. Matijašević, D. Oletić, M. Pavelić, I. P. Žarko, K. Pripužić, and L. Skorin-Kapov, "Urban crowd sensing demonstrator: Sense the zagreb air," in *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2014, pp. 423–424.

[99] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. New York, NY, USA: ACM, 2008.

[100] H. Jin, L. Su, and K. Nahrstedt, "Centurion: Incentivizing multi-requester mobile crowd sensing," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[101] J. Li, Z. Cai, J. Wang, M. Han, and Y. Li, "Truthful incentive mechanisms for geographical position conflicting mobile crowdsensing systems," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 2, pp. 324–334, 2018.

[102] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K. Leung, "Energy-aware participant selection for smartphone-enabled mobile crowd sensing," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1435–1446, Sep. 2017.

[103] J. Chen, H. Ma, D. S. L. Wei, and D. Zhao, "Participant-density-aware privacy-preserving aggregate statistics for mobile crowd-sensing," in *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, Dec 2015, pp. 140–147.

[104] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue, "Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, June 2016, pp. 354–363.

[105] M. Li, J. Lin, D. Yang, G. Xue, and J. Tang, "Quac: Quality-aware contract-based incentive mechanisms for crowdsensing," in *2017 IEEE 14th Interna-*

*tional Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2017, pp. 72–80.

[106] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, "Free market of crowdsourcing: Incentive mechanism design for mobile sensing," *IEEE transactions on parallel and distributed systems*, vol. 25, no. 12, pp. 3190–3200, 2014.

[107] L. Huang, Y. Zhu, J. Yu, and M. Y. Wu, "Group buying based incentive mechanism for mobile crowd sensing," in *2016 IEEE International Conference on Sensing, Communication, and Networking (SECON)*.

[108] J. Wang, M. Li, Y. He, H. Li, K. Xiao, and C. Wang, "A blockchain based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 6, pp. 17 545–17 556, 2018.

[109] M. Xiao, J. Wu, L. Huang, Y. Wang, and C. Liu, "Multi-task assignment for crowdsensing in mobile social networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2227–2235.

[110] Y. Liu, J. Niu, and X. Liu, "Comprehensive tempo-spatial data collection in crowd sensing using a heterogeneous sensing vehicle selection method," *Personal Ubiquitous Comp.*, vol. 20, no. 3, pp. 397–411, 2016.

[111] D. Lin, Q. Wang, P. Yang, and Z. Zhang, "A multidimensional reputation evaluation model for mobile crowd sensing," in *15th International Wireless Communications Mobile Computing Conference (IWCMC)*, June 2019, pp. 2070–2073.

[112] S. Liu, Z. Zheng, F. Wu, S. Tang, and G. Chen, "Context-aware data quality estimation in mobile crowdsensing," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[113] M. Pouryazdan and B. Kantarci, "The smart citizen factor in trustworthy smart city crowdsensing," *IT Professional*, vol. 18, no. 4, pp. 26–33, 2016.

[114] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1732–1744, June 2016.

[115] L. G. Jaimes, I. J. Vergara-Laurens, and A. Raij, "A survey of incentive techniques for mobile crowd sensing," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 370–380, 2015.

[116] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203–4214, 2015.

[117] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 635–644.

[118] J. Lin, D. Yang, M. Li, J. Xu, and G. Xue, "Bidguard: A framework for privacy-preserving crowdsensing incentive mechanisms," in *IEEE Conference on Communications and Network Security (CNS)*, Oct 2016.

[119] Y. Liu, X. Xu, J. Pan, J. Zhang, and G. Zhao, "A truthful auction mechanism for mobile crowd sensing with budget constraint," *IEEE Access*, vol. 7, pp. 43 933–43 947, 2019.

[120] J. Xu, Z. Rao, L. Xu, D. Yang, and T. Li, "Incentive mechanism for multiple cooperative tasks with compatible users in mobile crowd sensing via online communities," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[121] S. Mondal, S. Ghosh, S. Khatua, R. Das, and U. Biswas, "Cost effective algorithms for participant selection problem in mobile crowd sensing environment," in *2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Dec 2018, pp. 453–458.

[122] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "Emc3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Transactions on Mobile Computing*, vol. 14, no. 7, pp. 1355–1368, 2015.

[123] M. Dai, Z. Su, Y. Wang, and Q. Xu, "Contract theory based incentive scheme for mobile crowd sensing networks," in *2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, June 2018, pp. 1–5.

[124] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Motivating smartphone collaboration in data acquisition and distributed computing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2320–2333, Oct 2014.

[125] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, Oct 2018.

[126] S. A. Hamid, G. Takahara, and H. S. Hassanein, "On the recruitment of smart vehicles for urban sensing," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 36–41.

[127] S. A. Hamid, H. Abouzeid, H. S. Hassanein, and G. Takahara, "Optimal recruitment of smart vehicles for reputation-aware public sensing," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2014, pp. 3160–3165.

[128] C. Hu, M. Xiao, L. Huang, and G. Gao, "Truthful incentive mechanism for vehicle-based nondeterministic crowdsensing," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, June 2016, pp. 1–10.

[129] H. Li, T. Li, F. Li, S. Yang, and Y. Wang, "Multi-expertise aware participant selection in mobile crowd sensing via online learning," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Oct 2018, pp. 433–441.

[130] H. Xiong, D. Zhang, G. Chen, L. Wang, and V. Gauthier, "Crowdtasker: Maximizing coverage quality in piggyback crowdsensing under budget constraint," in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2015, pp. 55–62.

[131] Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, and G. Chen, "A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2392–2407, 2017.

[132] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proceedings of the 8th International Conference on Pervasive Computing*, ser. Pervasive'10. Berlin, Heidelberg: Springer Verlag, 2010, pp. 138–155.

[133] S. He, D. Shin, J. Zhang, and J. Chen, "Near-optimal allocation algorithms for location-dependent tasks in crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3392–3405, 2017.

[134] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2306–2320, 2017.

[135] M. Davari and H. Amintoosi, "A survey on participant recruitment in crowdsensing systems," in *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*, Oct 2016, pp. 286–291.

[136] J. Liu, H. Shen, H. S. Narman, W. Chung, and Z. Lin, "A survey of mobile crowdsensing techniques: A critical component for the internet of things," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 3, Jun. 2018. [Online]. Available: https://doi.org/10.1145/3185504

[137] M. Hu, Z. Zhong, Y. Niu, and M. Ni, "Duration-variable participant recruitment for urban crowdsourcing with indeterministic trajectories," *IEEE Transactions on Vehicular Technology*, 2017.

[138] S. Chessa, A. Corradi, L. Foschini, and M. Girolami, "Empowering mobile crowdsensing through social and ad hoc networking," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 108–114, 2016.

[139] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14. New York, NY, USA: ACM, 2014, pp. 703–714. [Online]. Available: http://doi.acm.org/10.1145/2632048.2632059

[140] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2254–2262.

[141] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, Oct 2015, pp. 136–144.

[142] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[143] A. Bazzi and A. Zanella, "Position based routing in crowd sensing vehicular networks," *Ad Hoc Networks*, vol. 36, pp. 409–424, 2016.

[144] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Reputation-aware, trajectory-based recruitment of smart vehicles for public sensing," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–14, 2017.

[145] G. Yang, S. He, Z. Shi, and J. Chen, "Promoting cooperation by the social incentive mechanism in mobile crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 86–92, 2017.

[146] H. Jin, L. Su, and K. Nahrstedt, "Theseus: Incentivizing truth discovery in mobile crowd sensing systems," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. Mobihoc '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3084041.3084063

[147] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, 2011.

[148] A. Kearney, "How automakers can survive the self-driving era," 2017, www.atkearney.com/automotive/.

[149] X. Yu, W. Zhang, L. Zhang, V. O. Li, J. Yuan, and I. You, "Understanding urban dynamics based on pervasive sensing: An experimental study on traffic density and air pollution," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1328–1339, 2013.

[150] Y. Gu, L.-T. Hsu, and S. Kamijo, "Towards lane-level traffic monitoring in urban environment using precise probe vehicle data derived from three-dimensional map aided differential gnss," *IATSS Research*, vol. 42, no. 4, pp. 248 – 258, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0386111217300894

[151] E. Jenelius and H. N. Koutsopoulos, "Travel time estimation for urban road networks using low frequency probe vehicle data," *Transportation Research Part B: Methodological*, no. Supplement C, pp. 64 – 81, 2013.

[152] D. Portugal and R. P. Rocha, "Cooperative multi-robot patrol with bayesian learning," *Auton. Robots*, vol. 40, no. 5, pp. 929–953, Jun. 2016. [Online]. Available: http://dx.doi.org/10.1007/s10514-015-9503-7

[153] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 7:1–7:31, Aug. 2015.

[154] C. Leonardi, A. Cappellotto, M. Caraviello, B. Lepri, and F. Antonelli, "Secondnose: An air quality mobile crowdsensing system," in *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, ser. NordiCHI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 1051–1054. [Online]. Available: https://doi.org/10.1145/2639189.2670273

[155] M. Wu, D. Ye, S. Tang, and R. Yu, "Collaborative vehicle sensing in bus networks: A stackelberg game approach," in *Communications in China (ICCC), 2016 IEEE/CIC International Conference on*. IEEE, 2016, pp. 1–6.

[156] M. Báguena, C. T. Calafate, J.-C. Cano, and P. Manzoni, "An adaptive anycasting solution for crowd sensing in vehicular environments," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7911–7919, 2015.

[157] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 2542–2550.

[158] K. Yi, R. Du, L. Liu, Q. Chen, and K. Gao, "Fast participant recruitment algorithm for large-scale vehicle-based mobile crowd sensing," *Pervasive and Mobile Computing*, 2017.

[159] F. Restuccia, N. Ghosh, S. Bhattacharjee, S. K. Das, and T. Melodia, "Quality of information in mobile crowdsensing: Survey and research challenges," *ACM Trans. Sen. Netw.*, vol. 13, no. 4, Nov. 2017. [Online]. Available: https://doi.org/10.1145/3139256

[160] J. Yang, P. Li, and H. Wang, "Participant reputation aware data collecting mechanism for mobile crowd sensing," in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, Oct 2017, pp. 1–6.

[161] M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, and H. Song, "Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing," *IEEE Access*, vol. 5, pp. 1382–1397, 2017.

[162] M. Zhang, P. Yang, C. Tian, S. Tang, X. Gao, B. Wang, and F. Xiao, "Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7698–7707, Sept 2016.

[163] B. Guo, H. Chen, Q. Han, Z. Yu, D. Zhang, and Y. Wang, "Worker-contributed data utility measurement for visual crowdsensing systems," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2379–2391, 2017.

[164] T. Luo, S. S. Kanhere, J. Huang, S. K. Das, and F. Wu, "Sustainable incentives for mobile crowdsensing: Auctions, lotteries, and trust and reputation systems," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 68–74, 2017.

[165] D. Peng, F. Wu, and G. Chen, "Data quality guided incentive mechanism design for crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 307–319, 2018.

[166] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, 2014.

[167] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[168] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, and C. Chen, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016.

[169] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD dataset roma/taxi (v. 2014-07-17)," Downloaded from https://crawdad.org/roma/taxi/20140717, Jul. 2014.

[170] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[171] J. Xu, J. Xiang, and D. Yang, "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353–6364, 2015.

[172] "Movement-based incentive for crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7223–7233, 8 2017.

[173] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," *2012 Proceedings IEEE INFOCOM*, pp. 2140–2148, 2011.

[174] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," 05 2013, pp. 1167–1178.

[175] K. Han, H. Huang, and J. Luo, "Posted pricing for robust crowdsensing," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 261–270. [Online]. Available: https://doi.org/10.1145/2942358.2942385

[176] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets uav: Energy-efficient task assignment and route planning," *IEEE Transactions on Communications*, vol. 66, no. 11, pp. 5526–5538, 2018.

[177] I. Rekleitis, A. P. New, E. S. Rankin, and H. Choset, "Efficient boustrophedon multi-robot coverage: an algorithmic approach," *Annals of Mathematics and Artificial Intelligence*, vol. 52, pp. 109–14c2, 2008.

[178] P. Janoušek and J. Faigl, "Speeding up coverage queries in 3d multi-goal path planning," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5082–5087.

[179] T. Pereira, A. P. G. Moreira, and M. Veloso, "Multi-robot planning for perception of multiple regions of interest," in *Iberian Robotics conference*. Springer, 2017, pp. 275–286.

[180] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[181] V. Karpov, A. Migalev, A. Moscowsky, M. Rovbo, and V. Vorobiev, "Multi-robot exploration and mapping based on the subdefinite models," in *International Conference on Interactive Collaborative Robotics*. Springer, 2016, pp. 143–152.

[182] J. Yu and S. M. LaValle, "Structure and intractability of optimal multi-robot path planning on graphs," in *AAAI*, 2013.

[183] J. P. van den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 430–435.

[184] M. Čáp, P. Novák, A. Kleiner, and M. Selecký, "Prioritized planning algorithms for trajectory coordination of multiple mobile robots," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 835–849, July 2015.

[185] M. R. K. Ryan, "Exploiting subgraph structure in multi-robot path planning," *CoRR*, vol. abs/1111.0053, 2011. [Online]. Available: http://arxiv.org/abs/1111.0053

[186] D. Silver, "Cooperative pathfinding," in *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, ser. AIIDE'05. AAAI Press, 2010, pp. 117–122.

[187] L. Xiao, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 35–47, 2018.

[188] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 1257–1262.

[189] M. A. Alsheikh, Y. Jiao, D. Niyato, P. Wang, D. Leong, and Z. Han, "The accuracy-privacy trade-off of mobile crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 132–139, 2017.

[190] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2019.

[191] S. Arora and B. Barak, "Computational complexity - a modern approach," 2009.

[192] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: evidence and implications," in *IEEE INFOCOM '99. Conference on Computer Communications.*, March 1999.

[193] W. Genders and S. N. Razavi, "Impact of connected vehicle on work zone network safety through dynamic route guidance," *Journal of Computing in Civil Engineering*, vol. 30, no. 2, p. 04015020, 2016.

[194] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15. ACM, 2015, pp. 177–186.

[195] H. Liu, V. Ramasubramanian, and E. G. Sirer, "Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '05. Berkeley, CA, USA: USENIX Association, 2005, pp. 3–3. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251086.1251089

[196] R. H. Jeong, "The prediction of bus arrival time using automatic vehicle location systems data," *Doctoral dissertation, Texas A&M University. Texas A&M University.*, 2004.

[197] S. Uppoor, "Tapascologne," http://tapioca.sfaye.com/TAPASCologne/.

[198] D. Hasenfratz, O. Saukh, C. Walser, C. Hueglin, M. Fierz, and L. Thiele, "Pushing the spatio-temporal resolution limit of urban air pollution maps," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2014, pp. 69–77.

[199] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "icrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 2010–2022, 2016.

[200] L. Pournajaf, L. Xiong, and V. Sunderam, "Dynamic data driven crowd sensing task assignment," *Procedia Computer Science*, vol. 29, pp. 1314 – 1323, 2014, 2014 International Conference on Computational Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914002956

[201] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.

[202] L. Shu, Y. Chen, Z. Huo, N. Bergmann, and L. Wang, "When mobile crowd sensing meets traditional industry," *IEEE Access*, vol. 5, pp. 15 300–15 307, 2017.

[203] D. He, S. Chan, and M. Guizani, "User privacy and data trustworthiness in mobile crowd sensing," *IEEE Wireless Communications*, 2015.

[204] C. Fiandrino, A. Capponi, G. Cacciatore, D. Kliazovich, U. Sorger, P. Bouvry, B. Kantarci, F. Granelli, and S. Giordano, "Crowdsensim: a simulation platform for mobile crowdsensing in realistic urban environments," *IEEE Access*, vol. 5, pp. 3490–3503, 2017.