Optimized Large-Scale Road Sensing Through Crowdsourced Vehicles

Yongxuan Lai, Yifan Xu, Duojian Mai, Yi Fan, and Fan Yang

Abstract—**Modern vehicles are gradually becoming powerful mobile sensing, communication, computing and storage platforms, which bring about the concept of vehicular urban sensing that leverages sensor nodes as an effective and affordable solution for large-scale and fine-grained sensing. And there is a trend to combine the vehicular sensing and outsourcing technologies to solve the large-scale urban road sensing problem. However, how to select appropriate participated vehicles, how to least interrupt the original routes of vehicles, and how to actively maximize the benefit of the sensing remain challenging problems. In this paper, we introduce a Crowdsourced Vehicular Sensing (CVS) framework based on more realistic assumptions of the vehicular sensing, which consists of three steps: vehicle recruitment, candidate path calculation, and path computing. We define a *maximal weighted sensing paths* (MWSP) problem, which is NP-Complete, in vehicular crowdsensing scenario and use heuristic methods to speed up the solving process for large-scale crowdsensing in urban road networks. The MWSP problem is formulated as a maximal satisfiability (MaxSAT) problem, and a *least-interrupted* urban sensing strategy is adopted. So trips are least disrupted when conducting the sensing tasks, which would increase the drivers' willingness to participate in urban sensing. Experiments based on real-world road-network and historical origin-destination datasets verify the effectiveness of the proposed method. The results show that the proposed algorithm outperforms other solutions and it can solve the vehicular crowdsensing problem effectively and efficiently.**

Index Terms—**Vehicular crowdsensing, maximal weighted sensing paths, least-interrupted urban sensing.**

Manuscript received August 25, 2020; revised April 14, 2021 and December 13, 2021; accepted January 19, 2022. This work was supported in part by the Natural Science Foundation of China under Grant 61872154, in part by the Natural Science Foundation of Fujian under Grant 2018J01097, in part by the Natural Science Foundation of Guangdong under Grant 2021A1515011578, in part by the Shenzhen Basic Research Program under Grant JCYJ20190809161603551, in part by the Special Foundation for Talents in Qiannan Normal University for Nationalities in 2019, in part by the Project for Growing Youth Talents of the Educational Department of Guizhou, and in part by the Foundation Project of Science and Technology Plans of Qiannan under Grant 2020XK05ST. The Associate Editor for this article was K. Savla. *(Corresponding authors: Yongxuan Lai; Fan Yang.)*

Yongxuan Lai, Yifan Xu, and Duojian Mai are with the Department of Software, School of Informatics, Xiamen University, Xiamen 361005, China, and also with the Shenzhen Research Institute, Xiamen University, Shenzhen 518000, China (e-mail: laiyx@xmu.edu.cn).

Yi Fan is with the Key Laboratory of Complex Systems and Intelligent Computing, School of Mathematics and Statistics, Qiannan Normal University for Nationalities, Duyun 558000, China (e-mail: 435980444@qq.com).

Fan Yang is with the Department of Automation and the Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision-Marking, Xiamen University, Xiamen 361005, China (e-mail: yang@xmu.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3147211

I. Introduction

With the advancement of mobile sensing and communication technologies, there has been increasing research on the sensing of urban cities, which is also called *urban sensing* [1]. Urban sensing leverages sensor nodes, e.g. fixed road-side sensors, probe vehicles and mobile phones, as an effective and affordable solution for large-scale and fine-grained sensing in road networks or city areas [2]–[4].

Fixed sensors are deployed along the roads to collect traffic data in the traditional urban sensing. But due to the high costs of installation and maintenance, the more cost-effective probe vehicles are used as an alternative for urban sensing [2]. Probe vehicles are equipped with various sensing units, and they could be categorized into two types in the scenario of urban sensing [5]. The "active" probing vehicles are vehicles instrumented and then sent into the field for travel time data collection. They are solely for the tasks of urban sensing, e.g. gathering the traffic data or street view data. They incur cost on the energy and labor, and would increase the traffic flow themselves, which have a risk of deteriorating the traffic states. Conversely, "passive" probing vehicles are vehicles that are already in the traffic stream for purposes other than data collection. This refers to the crowdsourcing model in which an urban sensing task could be divided among participants to achieve a cumulative result. We call this new paradigm "public crowdsensing" [6], which includes generalized and large-scale monitoring such as environment, traffic monitoring, map updating, public safety, noise pollution assessment, etc. The aggregated data are often shared to the public and can be reused by multiple applications [4], [7]. Fig. 1 depicts a scenario of public crowdsensing where vehicles move within the urban area. They schedule trips with origins and destinations as any traditional vehicles. But they are also equipped with sensors, GPS receivers, and wireless communication modules (e.g., 4G and DSRC [8] modules), so they could sense the environment and generate the sensing data (e.g., temperature, CO_2 concentration), similar to the probe vehicles. A central cloud server, or road-side edge nodes, is also deployed as a crowdsourcing server for the urban sensing. Vehicles are selected or recruited by a crowdsourcing system, and they are motivated or paid to conduct the sensing tasks. The server would re-schedule trips for the vehicles to maximize the overall benefit of urban sensing, while at the same time not affecting the vehicles' scheduling too much. In other words, it adopts a "least-interrupted" urban sensing strategy that the

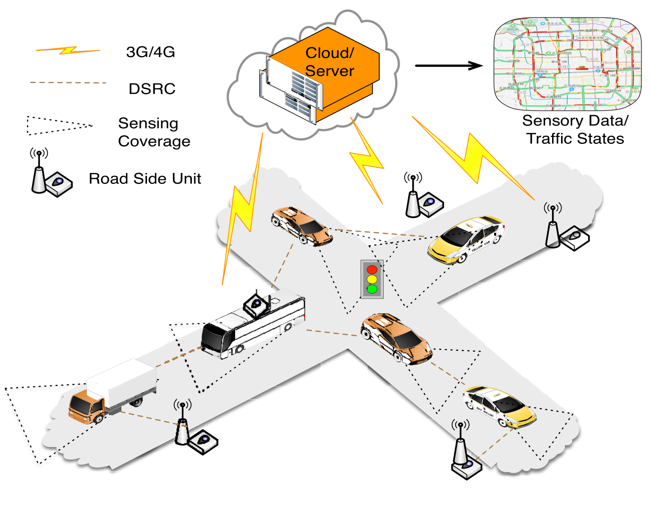


Fig. 1. Scenario of vehicular crowdsensing. Vehicles equipped with sensors move within the urban area. Each vehicle senses the area and generates data that could be collected through wireless communications.

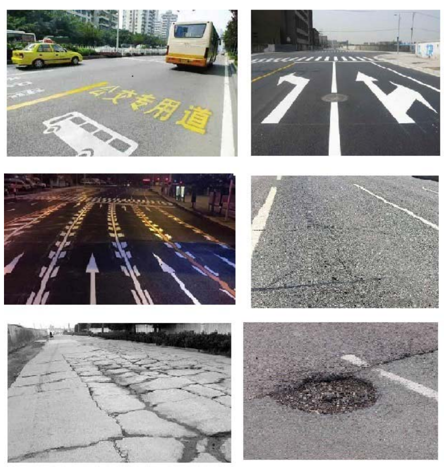


Fig. 2. Example crowdsensing scenario: real-time road surface perception and detection.

detailed paths of trips would be changed a little but still within a user-defined bypass tolerance range.

The vehicular crowdsourced sensing approach has several advantages over the traditional probe vehicle-based solutions. First, it has lower deployment cost because more and more vehicles are equipped with various sensors. It would be easier to recruit vehicles for the sensing tasks, especially with the rapid advancement of self-driving and electrical vehicles. Second, the vehicles are already within the road network, so they do not increase the traffic flow of the roads. And more importantly, their trips are not disrupted when conducting the sensing tasks, which would greatly increase drivers' willingness to participate the urban sensing. Third, it provides optimization opportunities for the urban sensing system, where various algorithms could be adopted to improve the sensing coverage of the urban area and increase the overall sensing benefits. Fig. 2 is the crowdsensing scenario of real-time road surface perception and detection. Vehicles are deployed to gather the data of road surfaces and road conditions through cameras, and computer vision algorithms are then conducted to percept these data. The application is to cover and sense the maximal amount of road segments with the least operation cost.

However, public vehicular crowdsensing is also a challenging issue given there are a large number of dynamic vehicles in the road network and they are with different pursuit of interests. It brings up issues on how to recruit suitable vehicles and how to properly give incentives for the participated vehicles. On one hand, there is significant sensing overlap existed among vehicles. Selecting all vehicles would result in a large amount of redundant data, so a careful design of public vehicular selection and efficient solving algorithms are needed to maximize the coverage while also limiting the costs. On the other hand, the participated vehicles must be rewarded as there is cost for vehicles to conduct the sensing tasks [9], especially when the trip of data gathering tasks might conflict with the original trips or scheduling of drivers. Existing works focus on the coverage of vehicle recruitment and vehicle crowdsourcing task [10]. Vehicles are either assumed to be "workers" that can go to specific locations as required to conduct the sensing task, or assumed not to purposely divert from their original paths or routes to conduct sensing tasks. And the cost or benefits of sensing operations are assumed to be equal, which is not viable in real-world scenarios where a weighting mechanism of sensing tasks is needed to quantify the sensing tasks and to maximize the overall benefit of public sensing tasks.

In this paper, we introduce a Crowdsourced Vehicular Sensing (CVS) framework based on more realistic assumptions on the constraints and benefits of vehicular sensing. We propose an innovative framework that systematically senses the road in large scale and with low cost, which includes the vehicle recruitment component, the candidate path calculation component, and the path computing component, and the vehicular crowdsensing is optimized through solving combinational problems on the weighted road networks. Our method would extract knowledge to recruit the most promising vehicles based on the historical trajectory datasets, and it is effective because the recruited vehicles could re-schedule their paths to cover the "untouched" road segments to sense the environment. Our method has the least disruption to the transportation when conducting the sensing tasks as there is a threshold for the detour ratio. It trades a balance between the effectiveness and non-interruption of crowdsensing, and would increase the drivers' willingness to participate in the urban sensing. The major contributions of this paper are as follows:

- We introduce the *maximal weighted sensing paths* (MWSP) problem in vehicular crowdsensing scenario. It is based on more realistic assumptions on the benefits of vehicular sensing, where road segments are set different weights of benefit based on the large-scale trajectory datasets, and the objective is to maximize the overall benefits of sensing.
- We propose a general crowdsourced vehicular sensing framework that adopts heuristic methods to speed up the solving process for large-scale crowdsensing in urban road networks. The framework includes the vehicle

recruitment, the candidate path calculation, and the path computing components. And we adopt a *least-interrupted* urban sensing strategy where trips are least disrupted when conducting the sensing tasks, which would increase the drivers' willingness to participate in the urban sensing.
- We conduct experiments on real-world road-network and historical origin-destination datasets to verify the effectiveness of the proposed methods. Experimental results show that the problem could be solved effectively and efficiently, and our scheme outperforms other algorithms.

The remainder of this paper is organized as follows. Section II presents the related work of this paper. Section III gives some preliminaries and formalizes the system model and problem. Section IV presents the detailed crowdsourced vehicular sensing framework, which includes vehicle recruitment, candidate path calculation, and path computing components. Section V presents the experimental studies and analysis. Finally, section VI concludes the paper and presents some future directions.

II. Related Work

In this section we review three categories of related works to position our work in the research community.

A. Mobile Crowdsourcing

The basic idea of crowdsourcing is to leverage the power of crowd to collaboratively complete a complex task, where each worker only completes much easier micro-tasks [11]. Crowdsourcing-based systems are widely used in domains such as software engineering [12], data mining [13], urban sensing [14], etc. It helps to solve industrial, academic, business, and other problems.

In recent years the concept of *spatial crowdsourcing* or *mobile crowdsourcing* becomes a hotspot in the research community [3]. Mobile crowdsourcing involves the utilization of mobile objects, e.g. smartphones or vehicles, to collect the different sensor data at different locations. In general, a typical mobile crowdsensing application does not harness the wisdom or knowledge of the crowd. However, it offers avenues to utilize both the sensor information as well as the knowledge possessed by humans. Reddy *et al.* [15] developed a recruitment framework to enable organizers to identify well-suited participants for data collections based on geographic and temporal availability as well as participation habits. It identifies a group of users who are very suitable for perceptual tasks, so as to enhance users' interest in participation. He *et al.* [16] proposed an efficient local ratio based algorithm (LRBA) to solve the optimal task allocation problem, where sensing tasks are with different requirements of quality of sensing and are associated with specific locations and constrained time budgets. Liu *et al.* [17] proposed the TaskMe framework for multi-task allocation. It transformed the FPMT (few participants, more tasks) problem using the Minimum Cost Maximum Flow (MCMF) theory. And the MPFT (more participants, few tasks) problem is addressed by the multi-objective optimization theory. Xiong *et al.* [18] proposed a generic mobile crowdsensing task allocation framework iCrowd, which operates with the energy-efficient piggyback crowdsensing task model. It optimizes the task allocation with different incentives and k-depth coverage objectives/constraints. Tang *et al.* [19] proposed a crowdsourcing method CLRIC that automatically extracts detailed lane structure of roads based on crowdsourcing. It filters the high-precision GPS data from the raw trajectories and mines the number and locations of traffic lanes through an optimized constrained Gaussian mixture model. Hachem *et al.* [20] proposed a probabilistic registration approach to reduce the number of the involved devices based on a realistic human mobility model. The scheme allowed devices to decide whether or not to register their sensing services depending on the probability of others.

B. Vehicular Sensing

Cooperative vehicular and urban sensing is at the heart of the intelligent and green city traffic management [21].

Lee *et al.* [22] proposed the MobEyes system for proactive urban monitoring. The system exploits the vehicle mobility to opportunistically diffuse concise summaries of the sensed data, and it harvests these summaries and builds a low-cost distributed index of the stored data to support various applications. Hull *et al.* [23] proposed a data management system CarTel for querying and collecting data from mobile vehicles, which enables the application development with data collected. Conceição *et al.* [24] used crowdsensing to sketch the map of on-street parking spaces to help drivers find parking slots. Delot *et al.* [25] proposed a pull-based data gathering strategy called GeoVanet, which adopts a DHT-based (DHT, dynamic hash table) model to identify a fixed geographical location where a mailbox is dedicated to the query. Users are able to send queries to a set of cars and find the desired information in a bounded time. Placzek [26] introduced a method of selective data collection for traffic control applications. The underlying idea is to detect the necessity of data transfers on the basis of uncertainty determination of the traffic control decisions, and sensor data are transmitted from vehicles to the control node only at selected time moments. Lindgren *et al.* [27] presented protocols for traffic-monitoring in vehicular networks. They defined two operation modes, multi-hop forwarding (MF) mode and delay-tolerant mode (DM). During MF mode messages are forwarded through the shortest path to the destination, while in DM mode messages are only forwarded at intersections to keep them inside the shortest path when the current carrier moves away. As the volume of sensed data might be large, there is also some research on reducing the volume of sensed data and the cost of gathering them. Lai *et al.* [21] proposed an efficient continuous event-monitoring framework based on fog nodes in vehicular network, where a two-level threshold strategy is adopted to suppress unnecessary data upload and transmissions.

Most of the above-mentioned research of vehicular sensing focuses on data gathering and communication techniques, whose goal is to efficiently sense the environment using vehicles and to effectively gather the desired data. Different from these works, in this research we focus on crowdsourcing these sensing tasks to selective vehicles and maximizing the benefit of sensing in large-scale road networks.

C. Vehicular Crowdsensing

Vehicular crowdsensing is a trend of research to combine sensing and crowdsourcing through vehicles, which takes advantage of the mobility of vehicles to provide location-based services in large-scale areas [28].

Li *et al.* [29] formalized the vehicle selection problem as cost control problems, and introduced 5 cost control methods: pruning, task selection, answer production, sampling, and miscellaneous. Xiong *et al.* [30] proposed the Crowdrecruiter platform which utilizes the vehicle's historical trajectory. It attempts to select a subset from the recruitment vehicles to meet the probability coverage constraints of multiple sensing cycles. Xiao *et al.* [28] formulated the interactions between a crowdsensing server and vehicles equipped with sensors in the area of interest as a vehicular crowdsensing game. They proposed a Q-learning based MCS payment strategy and sensing strategy for the dynamic vehicular crowdsensing game. Han *et al.* [31] proposed a trajectory-based vehicle election scheme in vehicular crowdsensing where offline and online model based algorithms are proposed for the recruitment strategy. Osamu [32] proposed a reservation based proactive route search. The route reservation cost is combined with a distance cost. If there are many reservations on a link, other cars avoid to choose the link. Xu *et al.* [9] designed a scheme to efficiently incentivize the vehicle agents to match the sensing distribution of the sampled data to the desired target distribution with a limited budget. They formulate the incentivizing problem as a new type of non-linear multiple-choice knapsack problem, with the dissimilarity between the collected data distribution and the desired distribution as the objective function, and the incentive is customized by combining monetary incentives and potential task (ride) requests at the destination. He *et al.* [10] proposed a participant recruitment strategy for vehicle-based crowdsourcing based on the predicted trajectory. They defined spatial and temporal coverage as two metrics for crowdsourcing quality to design greedy and genetic approximation algorithm, but assumed a single vehicle is sufficient to cover a geographical region at a specific time.

Most of the above works focus on the coverage of vehicle recruitment and vehicle crowdsourcing task. As there are a large number of vehicles on the road network, one key research direction of vehicular crowdsensing is how to select the appropriate number of users to complete the collection task, so as to achieve high coverage of perception tasks. In most cases, vehicles are either assumed to be "workers" that can go to specific locations as required to conduct the sensing task, or assumed not to purposely divert from their original paths or routes to conduct sensing tasks. Different from these assumptions, in this research we consider a scenario of large-scale road network sensing with limited budget of cost. Vehicles with sensing tasks still follow their origin and destination routes, but can make small adjustments on the original paths, so as to maximize the overall weight of benefit within the budget. The most similar work to us is [33], which proposed an evolution of the standard A* algorithm to enhance vehicular crowd-sensing coverage. But the route is chosen in a probabilistic way, among all those satisfying a constraint on the total length of the path, and they do not consider the problem of vehicular recruiting. Our approach is based on more realistic assumptions on the benefits of vehicular sensing, where road segments are set different weights of benefit by extracting knowledge from the historical trajectory dataset. The "detour ratio" is set as an important constraint for the optimization, and the objective is to maximize the overall benefits of sensing. In this way, on the premise of minimizing the interference of vehicular routes, the perceived coverage ratio of urban areas and the overall perceived benefit of sensing can be improved.

III. System Model and Problem Formulation

A. System Model

The road network is represented by a directed graph $G = (V, E)$, where V is a set of vertices and $E \subseteq V \times V$ is a set of ordered pairs of vertices, with a weight function $w(E) \rightarrow \mathbb{R}$ mapping edges to valued weights. The weight of an edge $e(u, v) \in E$ is $w(e)$, which represents the benefit of visiting the edge. A path is denoted by $l(o, d)$, where o is the origin and d is the destination, and it consists of a set of sequential edges, i.e. $C_l = \{(o, x_1), (x_1, x_2), \ldots, (x_k, d)\}$. Edge e is said to be covered by path l if $e \in C_l$. Given a set of paths, e.g. L, the set of its covered edges is defined as: $C_L = C_{l_1} \cup C_{l_2} \ldots \cup C_{l_{|L|}}$. Also, a path l is covered by C_L if $\forall e \in l, e \in C_L$.

A route is denoted by $r(o, d)$, where o is the origin and d is the destination. A path l is said to be feasible to a route r if they have the same origins and destinations, i.e. $l.o = r.o, l.d = r.d$. A route usually has one shortest path and multiple feasible paths. We define the set of τ-detour paths of r, i.e. P_r^τ, as

$$\{l \mid cost(l) \leq (1 + \tau) \cdot cost(l^*) \text{ and } r \text{ is covered by } l\},$$

where $cost(l)$ is the cost of path l, l^* is the shortest path of r, and $\tau \geq 0$ is the detour ratio which indicates the extra cost of path compared with the minimal cost.

We view the urban vehicular crowdsourced sensing problem as to find a set of paths that are covered by a set of vehicles and their corresponding routes so that the benefit of covering or sensing these paths is maximized. So given a set of routes R, a threshold budget, and a detour ratio r, the *crowdsourced sensing problem* could be defined as a *maximal weighted sensing paths* (MWSP) problem, which is to find a set of paths L so that the edges covered by them have the maximal weight under the budget. We are to maximize:

$$\sum_{e \text{ is covered by } L} w(e) \tag{1}$$

where $e \in C_L$ means e is covered by L. The τ-detour paths are sets of edges which could be overlapped. The constraints are described as: 1) the overall cost of paths in L is under some threshold θ, i.e. $\sum_{l \in L} f(l) \leq \theta$. Here f is a function that defines the cost of a path; 2) for every route in R, at most one

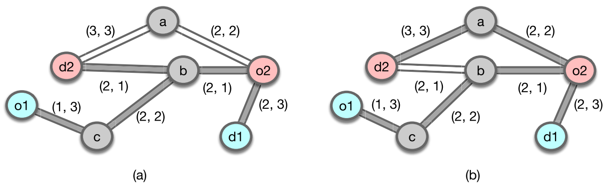


Fig. 3. Illustration of a road network that selects the maximal weighted sensing paths. Colored nodes o_1, d_1, o_2, d_2 denote the origin and destination points of routes, the lines in solid gray denote the covered paths, and (c, w) on the edge denotes the cost traveling the edge is c and the weight covering the edge is w. (a) The set of covered edges have a total cost of 11 and a total weight of 10; (b) the set of covered edges have a total cost of 12 and a total weight of 14.

path is selected to the path set L, i.e. $\forall r \in R$, $|P_r^\tau \cap L| \leq 1$, where P_r^τ is the set of paths for route r given τ.

Example 1: Fig. 3 illustrates the selection of maximal weighted sensing paths. Colored nodes o_1, d_1, o_2, d_2 denote the origin and destination points of routes, and the paths in solid gray denote the covered paths. (c, w) on the edge denotes the cost traveling the edge is c and the weight covering the edge is w. In Fig. 3(a), the shortest path for (o_1, d_1) is $l_1 = o_1 \rightarrow c \rightarrow b \rightarrow o_2 \rightarrow d_1$. Its total cost $cost(l_1)$ is 7, and the total weight $w(l_1)$ is 9. The shortest path for (o_2, d_2) is $l_2 = o_2 \rightarrow b \rightarrow d_2$. Its total cost $cost(l_2)$ is 4, and the total weight $cost(l_2)$ is 2. The set of covered edges, i.e. C_L, is denoted by the solid lines, which have a total cost of 11 $(cost(l_1) + cost(l_2))$ and a total weight of 10. In Fig. 3(b), the path chosen for (o_1, d_1) is the same as in (a), but the path chosen for (o_2, d_2) is $l_2 = o_2 \rightarrow a \rightarrow d_2$. Its total cost $cost(l_2)$ is 5, and the total weight $cost(l_2)$ is 5. The detour ratio of new path for l_2 is 0.25 $(= \frac{5}{4} - 1)$. The set of covered edges by l_1 and l_2 have a total cost of 12 and a total weight of 14. So the maximal weighted sensing paths C_L are the set of paths in solid gray in (b). Without the loss of generality the weight w could be defined according to the sensing applications. For example, the weight of the edge could be set inversely proportional to the number of vehicles passing the road segments, so vehicles are encouraged and motivated to choose less "popular" edges in the crowdsourced sensing tasks.

B. Problem Formulation

The MWSP problem could be formulated as an integer linear programming problem, and Table I summarizes the notations used in this paper. $\{t_1, \ldots, t_{|E|}\}$ are a list of variables, s.t. $t_i = 1$ if e_i is covered by C_L. Given a route r, also an origin-destination pair, let $\{l_1^r, \ldots, l_{|P_r^\tau|}^r\}$ be the set of τ-detour paths of r. Also there is a list of variables

$$p_1^r, \ldots, p_{|P_r^\tau|}^r \tag{2}$$

s.t. $p_k^r = 1$ if $l_k^r \in C_L$ and $p_k^r = 0$ otherwise. Let $e_1, \ldots, e_{|E|}$ be the set of directed edges in G. Given a path l_k^r, we introduce a list of propositions

$$q_{1,k}^r, \ldots, q_{|E|,k}^r, \tag{3}$$

s.t. $q_{i,k}^r = 1$ if edge e_i is covered by the k^{th} path of route r, i.e. $e_i \in l_k^r$.

TABLE I

NOTATIONS AND THEIR MEANINGS

Notation	Meaning	Notation	Meaning
$e_i, w(e_i)$	edge, and weight of edge	t_i	$\in \{0, 1\}$, whether e_i is covered
$r(o, d)$	route from o to d	τ	≥ 0, threshold of the detour ratio
δ	≥ 0, penalty factor for shortest paths calculation	$\hbar$	≥ 0, similarity constraint of paths
P_r^τ	set of paths for route r given τ	l_k^r	the k^{th} path of route r
C_x	set of edges covered by x	p_k^r	$\in \{0, 1\}$, whether path l_k^r is selected
θ	overall budget of cost	$q_{i,k}^r$	$\in \{0, 1\}$, whether e_i is covered by l_k^r

Having these notations defined, the MWSP problem is formulated as a 0-1 Integer Linear Programming (0-1 ILP) problem:

$$Maximize: \sum_{1 \leq i \leq |E|} w(e_i) * t_i \tag{4}$$

$$Constraints: \sum_{1 \leq k \leq |P_r^\tau|} p_k^r \leq 1, \quad \forall r \in R \tag{5}$$

$$t_i = max(q_{i,k}^r), \quad r \in R, 1 \leq k \leq |P_r^\tau|, \quad 1 \leq i \leq |E| \tag{6}$$

$$\sum_{r \in R} \sum_{1 \leq k \leq |P_r^\tau|} f(l_k^r) * p_k^r \leq \theta \tag{7}$$

$$t_i \in \{0, 1\}, 1 \leq i \leq |E| \tag{8}$$

$$p_k^r \in \{0, 1\}, \quad 1 \leq k \leq |P_r^\tau|, \ r \in R \tag{9}$$

Constraint 5 implies for each route, there exists at most one path covered by the final path set. Constraint 6 is set by the definition of t_i and $q_{i,k}^r$. Constraint 7 implies the overall cost of the covered paths should be within the cost budget, where f is a predefined cost function that could be tailored as required. When $f(l_k^r) = 1$, the constraint is reduced to a cardinality constraint that means there are at most θ sensing paths in the final set. Constraint 8 and 9 imply the value range of the variables as it is a 0-1 integer programming problem. Note that only t_i and p_k^r are variables to be solved, while $q_{i,k}^r$ could be computed before formulating the formula.

C. Problem Complexity

As it could be formulated as a 0-1 ILP problem, the MWSP is NP-Complete [34]. Consequently, using 0-1 ILP for the optimal solution of the MWSP problem becomes impractical especially when there are a larger number of OD routes and paths. Actually, in a large road network graph, each route has a large number of paths even when with a relatively small detour ratio constraint.

Also, as the crowdsourced sensing tasks are usually executed in a continuous batched approach, e.g. dispatching tasks every 5 minutes, the crowdsourced sensing system should be executed timely and efficiently. So in the next section, we further discuss methods that scale down the problem and speed up the solving process.

IV. Crowdsourced Vehicular Sensing Framework

In this section, we propose a Crowdsourced Vehicular Sensing (CVS) framework that adopts heuristic methods to scale down the crowdsensing problem and speed up the solving process.

The key idea of the framework is to prune redundant and similar trajectories on the large-scale weighted road networks, given that there is a large overlap of paths among the routes of vehicles. The CVS system would first recruit a subset of participated vehicles by scanning the set of routes and selectively choosing a subset of the OD pairs, then these pairs are set as the input of the solver to calculate the final detailed sensing paths. The CVS system consists of three components or steps:

1) *Vehicle Recruitment*: the system recruits a set of vehicles to participate the sensing while given a cost budget. So a subset of routes, e.g. $R' \subseteq R$, are selected from the set of routes.
2) *Candidate Paths Calculation*: for every route, only part of the paths are selected from all the set of feasible paths as candidate paths given a detour ratio. Redundant and similar paths are pruned to reduce the problem size;
3) *Paths Computing*: given the selected routes and their feasible candidate paths, the solver computes a solution with the final paths to maximize the overall sensing benefits.

Heuristic approaches are adopted for the paths selection and calculation. The detour ratio of paths plays an important role as it controls the number of candidate paths and hence determines the scale of the problem. Also, diversified paths are preferred when selecting paths as the input of the solver.

A. Vehicle Recruitment

Crowdsourcing systems usually adopt incentive mechanisms to encourage users' involvement and participation. In this research, we assume the CVS system has a budget of cost or incentives, and it can maximally pay m vehicles for each sensing epoch. We also assume the origins and destinations of vehicles are known to the system by active reporting or deduction, and the travelling paths are generated by the navigation systems based on historical datasets.

Suppose the number of vehicles participating in the sensing is n ($n \gg m$), each vehicle has an OD route and a plan path p (e.g. shortest path), then at the vehicle recruitment phase the goal is to find a set of paths L, s.t. the edges covered by L have the maximal weight under the budget, that is to maximize:

$$\sum_{e \text{ is covered by } L} w(e) \tag{10}$$

It is an ILP problem similar to that described in III-A, except that each route has only one path and this path is already known and defined as the plan path. So it is can be solved easily and efficiently.

B. Candidate Paths Calculation

The urban road network is abstracted as a directed connected graph. Each route has multiple feasible paths from its origin to the destination, so the path searching algorithms could be adopted to calculate a set of paths for each route. Here we also introduce the concept of *least-interrupted* vehicular urban sensing given a detour ratio constraint, hence the trips are least disrupted when conducting the sensing tasks.

The classic A* algorithm [35] only returns a single shortest path, and the k shortest path (KSP) routing algorithm [36] usually returns a set of paths whose similarity is too high. So we borrow the idea from reference [37] and introduce a penalty factor to diversify the feasible paths. The basic idea is to increase the traffic cost of each edge on the optimized path when searching the next optimized path. Also, the detour ratio τ and the similarity $\hbar$ are introduced to terminate the searching, i.e. the cost of path is no more than $1+\tau$ times of that of the shortest path, and the similarity between the paths does not exceed $\hbar$.

Algorithm 1 presents the pseudocode of the detailed algorithm. The algorithm consists of three steps: 1) Initialize the set of L and PS, where PS is a pool set for feasible paths (line 1). Also the shortest path from o to d is denoted by p^* (line 2); 2) Calculate an optimized shortest path p with A* algorithm in graph G, add path p to the set PS, and add the penalty to graph G based the penalty factor (lines 4-6). Function $add\_penalty(G, p, \delta)$ would multiply the weight cost of each edge on path p by $(1+\delta)$. Repeat these operations until there are enough feasible paths in PS. For example, set max_pool_size as $2K$, so there are $2K$ paths in PS; 3) For each path p in PS, calculate its detour ratio and similarity with other paths (lines 8-9). If the detour ratio is less than $1+\tau$ and the similarity does not exceed $\hbar$, the path is added to L (line 11). The algorithm terminates once there are enough paths in set L (line 12). In this way, the algorithm selects K paths that meet the detour ratio and similarity constraints in the pool set PS. Here we assume the impact of re-routing is equal for all the drivers, but our algorithm could be easily extended to handle the scenarios that each driver has a different cost/negative impact. We can change the detour ratio τ to τ_i in line 10 in Algorithm 1, where i indicates the index of vehicle. Note that the number of paths obtained may be less than K as all paths in L should meet the similarity constraint $\hbar$. The candidate paths are diversified and the problem is scaled down.

C. Path Computing

After vehicle recruitment and path selection, redundant paths are pruned and the problem is scaled down. Through careful design, we just formulate the MWSP problem as a maximal satisfiability (MaxSAT) problem, which could be solved efficiently by solvers.

Given a route r, also an origin-destination pair, let $\{l_1^r, \ldots, l_{|P_r^\tau|}^r\}$ be the set of τ-detour paths of r. We introduce a list of propositions

$$p_1^r, \ldots, p_{|P_r^\tau|}^r \tag{11}$$

Algorithm 1 *K* Shortest and Diversified Path Routing Algorithm

Input : graph $G = (V, E)$, route $r(o, d)$, number of paths K, detour ratio τ, similarity constraint $\hbar$, penalty factor δ

Ouput: set of feasible paths L for r

```
L ← ∅,   PS ← ∅;
p* = A_star(G, o, d);
while size(PS) < max_pool_size do
    p ← A_star(G, o, d);
    PS.add(p);
    G ← add_penalty(G, p, δ);
for each path p in PS do
    p.detour = cost(p) / cost(p*);
    p.sim = max{sim(p, s)|s ∈ L};
    if p.detour ≤ 1 + τ and p.sim < ħ then
        L.add(p);
    if size(L) > K then
        break;
return L;
```

s.t. $p_k^r = 1$ if $l_k^r \in C_L$ and $p_k^r = 0$ otherwise. Therefore we have

$$p_1^r \vee \ldots \vee p_{|P_r^\tau|}^r \tag{12}$$

for all $r \in R$, and

$$p_j^r \rightarrow \bigwedge_{1 \leq i \leq |P_r^\tau| \text{ and } i \neq j} \neg p_i^r \tag{13}$$

for all $r \in R$ and $1 \leq j \leq |P_r^\tau|$. Thus we have $|P_r^\tau| \cdot (|P_r^\tau| - 1)$ binary clauses here. Constraint 12 implies there is at least one path of route r covered in the set C_L. In constraint 13, $p_j^r \in 0, 1$ is the variable that indicates whether the k^{th} path of route r is selected. If any path, e.g. p_j^r, is selected, then all the other paths should not be selected. So constraint 13 implies there is only one path for route r covered in the set C_L.

Let $e_1, \ldots, e_m$ be the set of directed edges in G. Given a path l_k^r, we introduce a list of propositions

$$q_{1,k}^r, \ldots, q_{m,k}^r, \tag{14}$$

s.t. $q_{i,k}^r = 1$ if edge e_i is covered by the k^{th} path of route r, i.e. $e_i \in l_k^r$. Based on this, we introduce a list of propositions $t_1, \ldots, t_m$ s.t. $t_i = 1$ if e_i is covered by C_L. Thus we have

$$t_i \leftrightarrow \bigvee_{r \in R} \bigvee_{1 \leq k \leq |P_\tau^r|} (p_k^r \wedge q_{i,k}^r) \tag{15}$$

for $1 \leq i \leq m$. Constraint 15 implies that if edge e_i is covered, at least one route with a feasible path exists, which contains e_i and the path is covered by set C_L. Also, if there is any route with a feasible path covered by set C_L, and the path contains e_i, then edge e_i should be covered by C_L. Note that $q_{i,k}^r$ can be computed before transforming this formula.

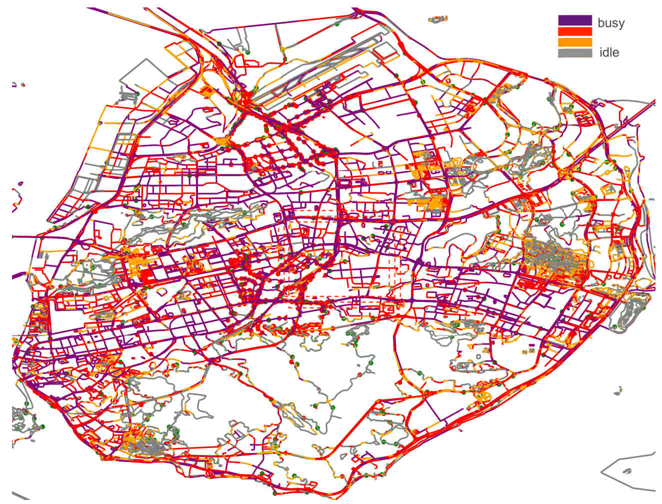


Fig. 4. The map of taxi trajectory coverage in Xiamen City based on the Xiamen Taxi Dataset. The color of each road segment is determined by the number of times covered by taxis.

Finally we are to maximize

$$\max \sum_{1 \leq i \leq m} w(e_i) * t_i. \tag{16}$$

where $t_1, \ldots, t_m$ are m soft clauses with weights $w(e_1), \ldots, w(e_m)$. In this way, the MWSP problem is formulated as a weighted MaxSAT problem.

There are many solvers for MaxSAT problem with varied performances. In this research we just choose Open-WBO MaxSAT Solver [38] to solve this problem, which won 2 gold medals and 1 silver medal in the MaxSAT Evaluation 2017.

V. Experimental Study

A. Environment Setting

We conduct experiments to study and verify the performance of the proposed algorithms on vehicular crowdsourcing. The experiments are based on the real road network data of Xiamen City, China. We obtained the Xiamen map from OpenStreetMap (longitude range: 118.066° ∼ 118.197°, latitude range: 24.424° ∼ 24.561°), and adopted the NetworkX framework[1] to model road network as a graph and simulate the experiment. There are 52479 road segments/edges and 49773 intersections/vertices. The cost weight of edges are given based on real-world datasets. The Xiamen Taxi Dataset [39] is used for the simulation, which consists of one-month trajectory data of about 5,000 taxicabs in Xiamen City, China during July 2014, where there are about 220 million GPS position records and 8 million live trips. The average number of times, e.g. $|c|$, covered by the taxis for every road segment hourly is first calculated, then the weight of the edge is set inversely proportional to $|c|$. Hence vehicles are encouraged and motivated to choose less "popular" edges for sensing in the crowdsourced sensing tasks. The weight setting is straightforward, but other customized methods, e.g. based on historical datasets or predictive methods, would be adopted.

[1] https://networkx.github.io/

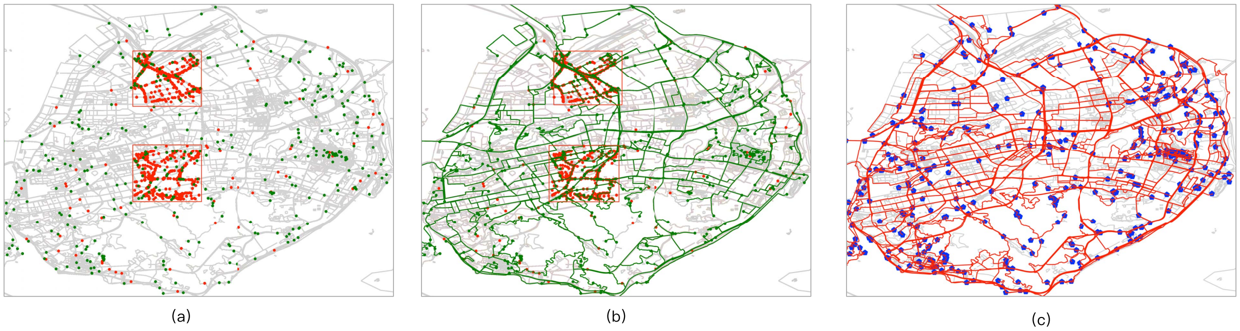


Fig. 5. (a) Regions in red rectangle are the main area of origins and destinations. 50% of the origin and destination points (in red) of the routes are located within main regions, the remaining OD points (in green) are randomly distributed in the map. (b) The edges in green color are road segments that covered by the 200 selected vehicles. Paths are diversified to maximize the weight. (c) Points in blue are the 400 OD points of the selected vehicles, edges in red are covered road segments. The benefit of weight increases to 465,474, and achieves 96.62% of the benefit sensed by the 1000 vehicles.

By default there are 1000 vehicles in the road network, yet only 200 vehicles are to be recruited for the crowdsourced sensing tasks due to limited budget. The 1000 vehicles are randomly selected from the dataset and they move according to their trajectories in the simulation. If a vehicle is recruited in the crowdsensing tasks, it would follow the instructions of the CVS system to take a detour. The detour ratio is set 0.3, the similar factor is set 0.7, the penalty factor is set 2, and the number of feasible paths for a route, K, is set 10. The optimal travelling paths are assumed to be generated by the navigation systems based on historical datasets (e.g. by the Google Map. For the heuristic and local search algorithms the maximal cut time is 200 seconds.

B. Metrics and Compared Algorithms

The following metrics are adopted for algorithm evaluation:

- Total benefit: the maximal benefit of the sensing paths as defined in Eq. 1 as the cost weight of all covered edges;
- Coverage ratio: the ratio of edge coverage, which is calculated by dividing the number of all covered edges by the total number of edges in the graph.
- Time efficiency: the running time that an algorithm consumes (or the cut off time) to get the results.

The *maximal weighted sensing paths* (MWSP) problem is NP-complete, which usually adopts heuristic approaches to get the solutions. Besides the proposed algorithm, we also conduct the following algorithms in the *path computing* phase for comparison:

1) Random: a random sensing algorithm that 200 vehicles are randomly selected out of the 1000 vehicles for the sensing.
2) Opportunistic: a sensing algorithm that all the vehicles opportunistically sense the road segments with a probability of 20%.
3) OTB [31]: an off-line trajectory based greedy algorithm which chooses a near-optimal vehicle set. Depending on the trajectory information of all the vehicles, the algorithm selects the best K vehicles from the set of participating vehicles, which are expected to achieve the biggest union coverage.
4) AStar [33]: an evolution of the standard A* algorithm to enhance vehicular crowd-sensing coverage. The route is chosen in a probabilistic way among those satisfying a detour constraint on the total length of the path, but it does not consider the problem of vehicular recruiting.
5) Hill Climbing: an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by making an incremental change to the solution [40]. It is a greedy algorithm that travels each OD path set and selects a path from the OD path set, and joins the path to current solution to maximize the current weight.
6) Simulated Annealing: a stochastic optimization algorithm based on Monte-Carlo iterative solution strategy [40]. It models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy. The local optimal solution jumps probabilistically and eventually tends to global optimal.
7) Genetic Algorithm [41]: a computation model that simulates the natural evolution of Darwin's biological evolution theory and the biological evolution process of genetic mechanism. It generates high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection.

C. Overall Results

The OD pairs are selected from the real-world dataset. As illustrated in Fig. 5(a), two regions (in red rectangle) are defined as the main area of origins and destinations. 50% of the origin and destination points (in red) of the routes are located within these regions when 1000 vehicles participate in the crowdsourced sensing tasks. The remaining OD points (in green) are randomly distributed on the map. The concentration of the OD points increases the trajectory redundancy and helps to test the effectiveness of our vehicle recruitment strategy.

Our crowdsensing framework combines the vehicle recruitment phase and the path computing phase, where the former is to prune redundant and similar trajectories on the large-scale

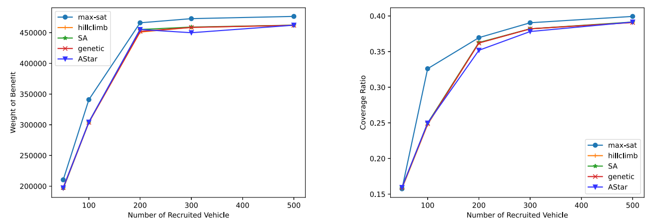
TABLE II
COMPARISON OF ALGORITHMS UNDER DEFAULT PARAMETERS

Algorithms /Metrics	Benefit	Coverage Ratio	Cutoff Time
Random	291,267	0.2423	-
Opportunistic	112,802	0.0967	-
AStar	450,804	0.3476	-
OTB	316,238	0.2546	7
Hill Climbing	450,970	0.3621	46
Genetic Algorithm	450,970	0.3621	200
Simulated Annealing	451,343	0.3610	200
max-sat	**465,474**	**0.3663**	**10**

weighted road networks. The vehicle recruitment problem is converted into an integer linear programming (ILP) problem, and solved by classic optimization engines, e.g. CPlex.[2] 200 out of the 1000 vehicles are selected by the solver to participate in the final sensing tasks. As shown in Fig. 5(b), the road segments in green color are paths covered by these 200 vehicles and they are diversely dispersed in the road network. And the total benefit of weight of the 1000 vehicles to sense according to their original route is 486,754, and while that of the 200 recruited vehicles is 426,234. The total benefit of weight decreases by 12.43%, yet 80% of the crowdsourced budget, i.e. the recruited vehicles, are saved. And by further processing within our CVS framework, the final sensing tasks and covered road segments are illustrated in Fig. 5(c). 400 OD points of the 200 vehicles are in blue color, and the covered road segments are in red. The benefit of weight increases to 465,474, and achieves 96.62% of the benefit sensed by the 1000 vehicles.

Table II shows the overall results of the compared algorithms given that the 200 vehicles have been recruited. Our algorithm has the maximal weight of benefit at 465,474 which has the best performance of all schemes. The proposed algorithm has more than 3 percent of benefit advantage than Hill Climbing, Genetic algorithm and Simulated Annealing algorithm. The proposed algorithm also has the highest coverage ratio at 0.3663. Paths in the Random scheme are not optimized, which leads to high overlapped ratio among road segments, resulting in low coverage ratio. In the Opportunistic scheme, vehicles randomly sense the road segments, which leads to unstable sense results. The ratio of overlapped road segments is also high, resulting in low coverage ratio, which is 0.0967. The Hill Climbing algorithm is a greedy algorithm, it terminates at 46 seconds and achieves benefit weight at 450,970 as it is easy to fall into the local optimum. The Genetic algorithm and the Simulated Annealing algorithm achieve similar results as the Hill Climbing algorithm when the cutoff time is set to 200 seconds. The AStar algorithm extends the paths in a probabilistic way, and the path computing is based on the simulated annealing, so its performance is similar

[2]https://www.ibm.com/analytics/cplex-optimizer



(a) Number of Vehicles Vs. Weight of Benefit (b) Number of Vehicles Vs. Coverage Ratio

Fig. 6. The effect of the number of recruited vehicles on the weight of benefit and coverage ratio.

to that of SA. The OTB scheme has the shortest running time, but it achieves a poor benefit weight at 316,238, which is about 32% lower than the proposed algorithm. Section V-D.5 and Fig. 11 further illustrate the impact of the running time on the algorithms.

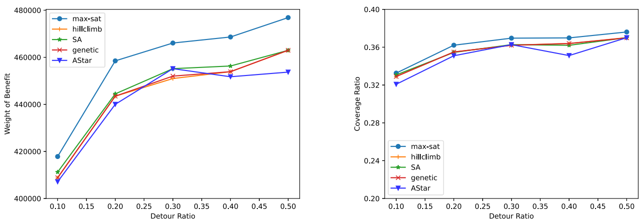
D. Analysis of Impact Factors

We also vary the parameters to study the impact of factors for the solvers to calculate the final paths. The number of crowdsourced vehicles, the detour ratio, the number of maximal paths for a route, the similarity threshold among feasible paths, and the penalty factor are controlled in the experiments to study and analyze their impact in this section.

1) Number of Recruited Vehicles: In this research, the number of recruited vehicles is set as the main constraint of budget in the crowdsourced sensing tasks. Fig. 6 (a) shows that the weight of benefit increases with the number of recruited vehicles. The weight of the proposed algorithm, denoted as *max-sat*, goes from 21,873 to 465,474 when the budget of recruited vehicles increases from 50 to 200. A larger budget means there are more vehicles available for doing the sensing tasks, hence more road segments would be covered and the weight of benefit would increase accordingly. Yet when the number increases further, e.g. to 300, the weight of benefit increases less than 1%. This is because when more vehicles participate in the sensing tasks, some vehicles would cover the same road segments. This leads to redundancy of sensing and does not increase the weight of benefit. Fig. 6 (b) shows the impact of recruited vehicles on the coverage ratio. The coverage ratio increases with the number of recruited vehicles. The ratio goes from 0.156 to 0.366 for *max-sat* when the number of recruited vehicles goes from 50 to 200. Yet when the number of vehicles is larger than 200, the trend of increment on the coverage ratio decreases, similar to the impact on the cost of weight.

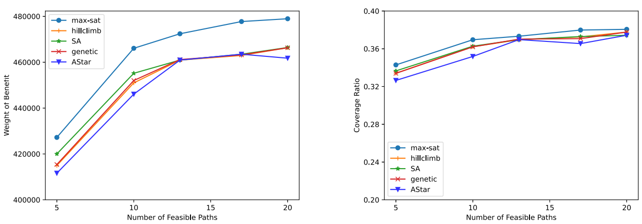
2) Detour Ratio τ: Vehicles are allowed to travel extra paths to the crowdsourced sensing tasks. The detour ratio of a path $l(o, d)$ is calculated as dividing the length of $l(o, d)$ by the length of the shortest path between o and d.

Fig. 7 (a) and Fig. 7 (b) show that as the detour ratio increases, the weight of benefit and coverage ratio for each algorithm increase. A larger detour ratio means more options of road segments could be chosen when finding feasible paths,



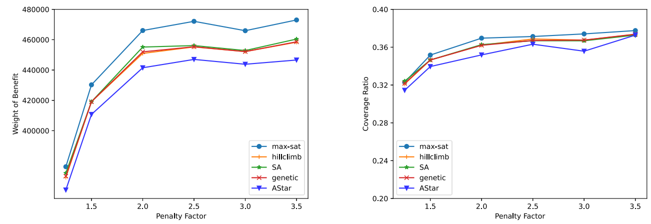
(a) Detour Ratio Vs. Weight of Benefit (b) Detour Ratio Vs. Coverage Ratio

Fig. 7. The effect of detour ratio on the weight of benefit and coverage ratio.



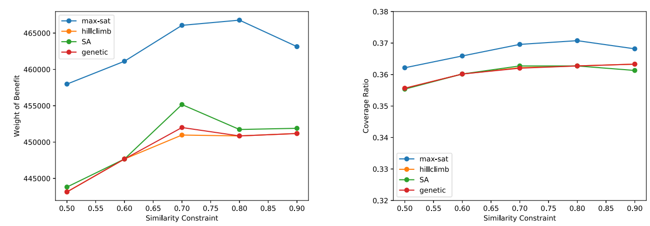
(a) Number of Feasible Paths Vs. Weight of Benefit (b) Number of Feasible Paths Vs. Coverage Ratio

Fig. 8. The effect of penalty factor on the weight of benefit and coverage ratio.



(a) Penalty Factor Vs. Weight of Benefit (b) Penalty Factor Vs. Coverage Ratio

Fig. 9. The effect of the number of feasible paths on the weight of benefit and coverage ratio.



(a) Number of Feasible Paths Vs. Coverage Ratio (b) Similarity Constraint Vs. Coverage Ratio

Fig. 10. The effect of similarity constraint and cutoff time on the weight of benefit and coverage ratio.

hence road segments with larger weight of cost could be added to the final path set. *max-sat* has the best performance both on the weight of benefit and coverage ratio. When the detour ratio goes from 0.10 to 0.50, the weight of benefit increases from 417821 to 476867, and the coverage ratio increases from 0.3326 to 0.3761. The other compared algorithms have similar performance. Their weight of benefit goes from around 41000 to around 46000, and their coverage ratio increases from around 0.330 to 0.369.

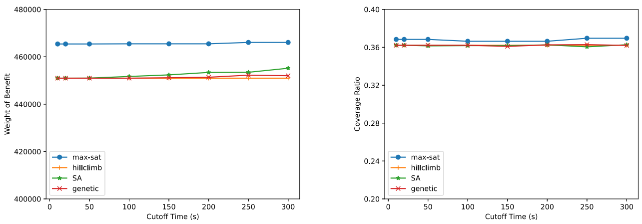
3) Number of Feasible Paths K: For recruited vehicles, more optional paths mean more choices for the path selection process. We also varied the number of feasible paths, K, to study its impact on the performance. Fig. 8 (a) and Fig. 8 (b) show that the weight of benefit and coverage ratio for all algorithms increase with the number of feasible paths. When K increases from 5 to 20, the weight of benefit increases from around 42000 to around 46000, and the coverage ratio increases from 0.345 to 0.374 for the *hillclimb*, *SA* and $genetic$ algorithms. But when the number of feasible paths for a route is bigger than 13, the increment on the two metrics is relatively small. One reason is that the similarity and detour thresholds are added to select the feasible paths, and there might not be any paths added to the candidate set of feasible paths even when K increases to a relatively large number. Furthermore, the *max-sat* algorithm has the best performance, which achieves a weight of benefit at 478948 and a coverage ratio of 0.3806. The AStar algorithm extends the feasible path in a probabilistic way, so K is not applicable for the performance analysis.

4) Similarity Constraint and Penalty Factor: In the proposed framework, diversified feasible paths are preferred in the final path selection process, and we use two parameters to add this diversity. The first parameter is the similarity constraint $\hbar$ that prunes out similar paths under the threshold, and the second parameter is the penalty factor δ that adds the cost of weight to the already selected paths when calculating the shortest paths.

Fig. 9 (a) and Fig. 9 (b) show that the weight of benefit and coverage ratio increase as the similarity threshold goes up. This is because the larger the threshold, the more feasible paths could be added to the candidate set, which adds diversity to the result set. But when $\hbar$ is relatively large, e.g. 0.8, the performance goes down for the *max-sat* algorithm. The weight of benefit goes down to 463134 and the coverage ratio goes down to 0.3681. This is because too many similar paths are added to the path set, and this would inversely harm the overall performance.

By setting a penalty factor δ, we increase the traffic cost of each edge on the optimized path when searching the next optimized path. So the penalty factor has similar impact as the similarity constraint, where larger penalty factor means more diversified paths to choose from. The weight of benefit and coverage ratio becomes larger as the penalty factor increases, as illustrated at Fig. 10 (a) and Fig. 10 (b).

5) Cutoff Time: The MWSP problem is NP-complete and we adopt some local search algorithms to solve it. So we also control the cutoff time to study its impact on the algorithms. As illustrated in Fig. 11 (a) and Fig. 11 (b) the cutoff time is set as $\{10, 20, 50, 100, 150, 200, 250, 300\}$. For the SA and $genetic$ algorithm, the weight of benefit increases as the cutoff time increases. This is because more running time is allowed for the searching. But for the $hillclimb$ algorithm, the program terminates before the cutoff time, so the performance is not improved even when more running time is allowed. For the



(a) Cut Off Time Vs. Weight of Benefit (b) Cut Off Time Vs. Coverage Ratio

Fig. 11. The effect of similarity constraint and cutoff time on the weight of benefit and coverage ratio.

max-sat algorithm, 10 seconds running time is enough to have a weight of benefit at 465409, though the value would increase to 466064 when the cutoff time is set larger than 250 seconds. The coverage ratio is the ratio of sensed road segments over all the road segments. All the algorithms would select similar number of road segments or paths into the final result set, so the ratio is relatively stable regarding the cutoff time.

VI. CONCLUSION

Recently there is a trend to combine the vehicular sensing and outsourcing technologies to solve the large-scale urban road sensing problem, which leverages sensor nodes as an effective and affordable solution for large-scale and fine-grained sensing. In this paper, we have presented a crowdsourced vehicular sensing framework that consists of three steps: vehicle recruitment, candidate path calculation, and path computing. We defined the MWSP problem in vehicular crowdsensing scenario and formulated it as a maximal satisfiability problem. A *least-interrupted* urban sensing strategy was adopted and heuristic methods were used to scale down the problem and speed up the solving process for large-scale crowdsensing in urban road networks. Experiments based on real-world road-network and historical origin-destination datasets verify the effectiveness of the proposed methods. The results show that the proposed algorithm outperforms other solutions and it can solve the vehicular crowdsensing problem effectively and efficiently.

For the future work, we will further verify and improve our algorithm based on more real-world datasets and consider factors such as traffic state, time selection and cost of transportation in the crowdsensing process. We will also study the incentive mechanisms of crowdsourced vehicles in urban sensing. Only when vehicles or drivers have the willingness to participate in the sensing, would the vehicular crowdsensing platform be successful and gather enough data for the system of smart city.

REFERENCES

[1] D. Zhao, H. Ma, L. Liu, and X.-Y. Li, “Opportunistic coverage for urban vehicular sensing,” *Comput. Commun.*, vol. 60, pp. 71–85, Apr. 2015.
[2] Y. Zhu, X. Liu, and Y. Wang, “Pervasive urban sensing with large-scale mobile probe vehicles,” *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 2, Feb. 2013, Art. no. 762503.
[3] S. R. B. Gummidi, X. Xie, and T. B. Pedersen, “A survey of spatial crowdsourcing,” *ACM Trans. Database Syst.*, vol. 44, no. 2, pp. 1–46, Jun. 2019.
[4] F. Zheng and H. Van Zuylen, “Urban link travel time estimation based on sparse probe vehicle data,” *Transp. Res. C, Emerg. Technol.*, vol. 31, pp. 145–157, Jun. 2013.
[5] I. Espada and P. Bennett, “Probe data and its application in traffic studies,” in *Proc. Inst. Public Works Eng. Australasia*, 2015, pp. 1–9.
[6] Y. Liu, J. Niu, and X. Liu, “Comprehensive tempo-spatial data collection in crowd sensing using a heterogeneous sensing vehicle selection method,” *Pers. Ubiquitous Comput.*, vol. 20, no. 3, pp. 397–411, 2016.
[7] T.-Y. Yu, X. Zhu, and M. Maheswaran, “Vehicular crowdsensing for smart cities,” in *Handbook Smart Cities*. Cham, Switzerland: Springer, 2018, pp. 175–204.
[8] J. B. Kenney, “Dedicated short-range communications (DSRC) standards in the United States,” *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
[9] S. Xu, X. Chen, X. Pi, C. Joe-Wong, P. Zhang, and H. Y. Noh, “ILOCuS: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 8, pp. 1831–1847, Aug. 2020.
[10] Z. He, J. Cao, and X. Liu, “High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2015, pp. 2542–2550.
[11] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, “Crowdsourced data management: A survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2296–2319, Sep. 2016.
[12] K. Mao, L. Capra, M. Harman, and Y. Jia, “A survey of the use of crowdsourcing in software engineering,” *J. Syst. Softw.*, vol. 126, pp. 57–84, Apr. 2017.
[13] G. Xintong, W. Hongzhi, Y. Song, and G. Hong, “Brief survey of crowdsourcing for data mining,” *Expert Syst. Appl.*, vol. 41, no. 17, pp. 7987–7994, Dec. 2014.
[14] M. Ferreira *et al.*, “Vehicular sensing: Emergence of a massive urban scanner,” in *Proc. Int. Conf. Sensor Syst. Softw.* Lisbon, Portugal: Springer, 2012, pp. 1–14.
[15] S. Reddy, D. Estrin, and M. Srivastava, “Recruitment framework for participatory sensing data collections,” in *Proc. Int. Conf. Pervasive Comput.*, Helsinki, Finland, May 2010, pp. 138–155.
[16] S. He, D. H. Shin, J. Zhang, and J. Chen, “Toward optimal allocation of location dependent tasks in crowdsensing,” in *Proc. Conf. Comput. Commun.*, 2014, pp. 745–753.
[17] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, “TaskMe: Multi-task allocation in mobile crowd sensing,” in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2016, pp. 403–414.
[18] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, “ICrowd: Near-optimal task allocation for piggyback crowdsensing,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2010–2022, Aug. 2016.
[19] L. Tang, X. Yang, Z. Dong, and Q. Li, “CLRIC: Collecting lane-based road information via crowdsourcing,” *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 9, pp. 2552–2562, Sep. 2016.
[20] S. Hachem, A. Pathak, and V. Issarny, “Probabilistic registration for large-scale mobile participatory sensing,” in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2013, pp. 132–140.
[21] Y. Lai *et al.*, “Fog-based two-phase event monitoring and data gathering in vehicular sensor networks,” *Sensors*, vol. 18, no. 2, p. 82, Dec. 2017.
[22] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, “Mobeyes: Smart mobs for urban monitoring with a vehicular sensor network,” *IEEE Wireless Commun.*, vol. 13, no. 5, pp. 52–57, Oct. 2006.
[23] B. Hull *et al.*, “CarTel: A distributed mobile sensor computing system,” in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, 2006, pp. 125–138.
[24] H. Conceiç ao, L. Damas, M. Ferreira, and J. Barros, “Large-scale simulation of v2v environments,” in *Proc. ACM Symp. Appl. Comput.*, 2008, pp. 28–33.
[25] T. Delot, N. Mitton, S. Ilarri, and T. Hien, “Decentralized pull-based information gathering in vehicular networks using GeoVanet,” in *Proc. IEEE 12th Int. Conf. Mobile Data Manage.*, Jun. 2011, pp. 174–183.
[26] B. Płaczek, “Selective data collection in vehicular networks for traffic control applications,” *Transp. Res. C, Emerg. Technol.*, vol. 23, pp. 14–28, Aug. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X1100180X
[27] A. Skordylis and N. Trigoni, “Efficient data propagation in traffic-monitoring vehicular networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 680–694, Sep. 2011.
[28] L. Xiao, T. Chen, C. Xie, H. Dai, and H. V. Poor, “Mobile crowdsensing games in vehicular networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1535–1545, Feb. 2018.

[29] X. Fan, J. Liu, Z. Wang, Y. Jiang, and X. Liu, "Crowdsourced road navigation: Concept, design, and implementation," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 126–128, Jun. 2017.

[30] D. Zhang, H. Xiong, L. Wang, and G. Chen, "CrowdRecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2014, pp. 703–714.

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

[32] O. Masutani, "A sensing coverage analysis of a route control method for vehicular crowd sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, Mar. 2015, pp. 396–401.

[33] S. Di Martino, P. Festa, and D. Asprone, "Adapting the a* algorithm to increase vehicular crowd-sensing coverage," in *Proc. Int. Conf. Comput. Logistics*. Vietri sul Mare, Italy: Springer, 2018, pp. 331–343.

[34] S. A. Cook, "The complexity of theorem-proving procedures," in *Proc. 3rd Annu. ACM Symp. Theory Comput.*, 1971, pp. 151–158.

[35] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[36] D. Eppstein, "Finding the K shortest paths," *SIAM J. Comput.*, vol. 28, no. 2, pp. 652–673, 1999.

[37] B. Liu, X. Chen, and Z. Cheng, "A dynamic multipath planning algorithm based on a* algorithm," *Microcomputer Appl.*, vol. 35, no. 4, pp. 17–19, 2016.

[38] R. Martins, V. Manquinho, and I. Lynce, "Open-WBO: A modular MaxSAT solver," in *Proc. Int. Conf. Theory Appl. Satisfiability Test.* Cham, Switzerland: Springer, 2014, pp. 438–445.

[39] Y. Lai, Z. Lv, K.-C. Li, and M. Liao, "Urban traffic Coulomb's law: A new approach for taxi route recommendation," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3024–3037, Aug. 2019.

[40] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. London, U.K.: Pearson, 2020.

[41] J. Sadeghi, S. Sadeghi, and S. T. A. Niaki, "Optimizing a hybrid vendor-managed inventory and transportation problem with fuzzy demand: An improved particle swarm optimization algorithm," *Inf. Sci.*, vol. 272, pp. 126–144, Jul. 2014.

Yongxuan Lai received the Ph.D. degree in computer science from the Renmin University of China in 2009. From September 2017 to September 2018, he was a Visiting Scholar with The University of Queensland, Australia. He is currently a Professor with the Software Engineering Department, School of Informatics, Xiamen University, China. His research interests include network data management, intelligent transportation systems, big data management, and analysis.

Yifan Xu received the B.S. degree from the Department of Software Engineering, Xiamen University, in 2017, where he is currently pursuing the master's degree with the School of Informatics. His research interest is intelligent transportation systems.

Duojian Mai is currently pursuing the bachelor's degree with the School of Informatics, Xiamen University. His research interest is intelligent transportation systems.

Yi Fan received the Ph.D. degree in artificial intelligence from Griffith University, Australia. His is currently an Associate Professor with the School of Mathematics and Statistics, Qiannan Normal University for Nationalities. His research interests include heuristics in NP-hard problems, deep learning, and graph theory.

Fan Yang received the Ph.D. degree in control theory and control engineering from Xiamen University in 2009. He is currently an Associate Professor with the Department of Automation, Xiamen University. His research interests include feature selection, ensemble learning, and intelligent transportation systems.