A Vehicular Crowd-sensing Incentive Mechanism for Temporal Coverage

Harish Chintakunta*, Janar Kahr†, Luis Jaimes‡
*Department of Electrical and Computer Engineering,†‡ Department of Computer Science,
Florida Polytechnic University
Email: *hchintakunta@floridapoly.edu, †jkahr4204@floridapoly.edu, ‡ljaimes@floridapoly.edu,

***Abstract*—Vehicular Crowd-sensing (VCS) is a new data collection paradigm that leverages the unique characteristics of vehicular mobility to collect sensing data. One of the main challenges for VCS is how to assign sensing tasks among participant so as to maintain the required Quality of Sensing Data (QoSD), while keeping participation profitable. We tackle this challenge by designing an incentive mechanism that encourages the collection of high QoSD, while improve participant utilities. The proposed mechanism includes a platform which post a set of sensing tasks and the associated rewards, and a set of participant vehicles equipped with sensors, who compete for these rewards. We model this competition as a non-cooperative game in which the set of vehicles are the players, and their trajectories are their strategies. Using open-street maps, SUMO vehicular traffic simulator, and extensive simulations, we show that our algorithm significantly outperforms a greedy approach in terms of QoSD, average vehicle utility, spatial coverage, and road utilization.**

***Index Terms*—**

I. Introduction

Vehicular Crowdsensing (VCS) [4] is a data collection paradigm that extends the concept of Mobile Crowdsensing (MCS) [5] from pedestrians to all kind of mobile vehicles. These include regular and connected vehicles, Autonomous vehicles (AVs), and Unmanned Aerial Vehicles (UAVs) among others. VCS has been extensively used for satellite data validation [6], traffic monitoring [10], infrastructure surveillance [2], and widely used by companies such Google, Here, and Uber for creating and updating maps [9]. A particular characteristic of vehicular mobility is its predictability, which results from setting the traveling route in advance, for instance by using a GPS. The use of this information constitutes an advantage when planning a data collection campaign, because we know in advance the places from where we can get samples, namely those located in the trajectory of participants. However, in this settings, a few natural questions arise, for instance, how collect sensing samples located out of the vehicles trajectories?, how to reach these locations?, and how often to sample from them?. We tackle these issues by proposing an incentive mechanism for VCS which encourage participants to deviate from the pre-planned trajectories to reach these places and collect samples at a given frequency. Figure 1 sketches the main components of the proposed VCS system. It includes a platform (cloud) which sets a set of sensing tasks at different Places of Sensing Interest (PsI) across a given area, and the corresponding rewards. It also includes a set of participant vehicles whose attracted by

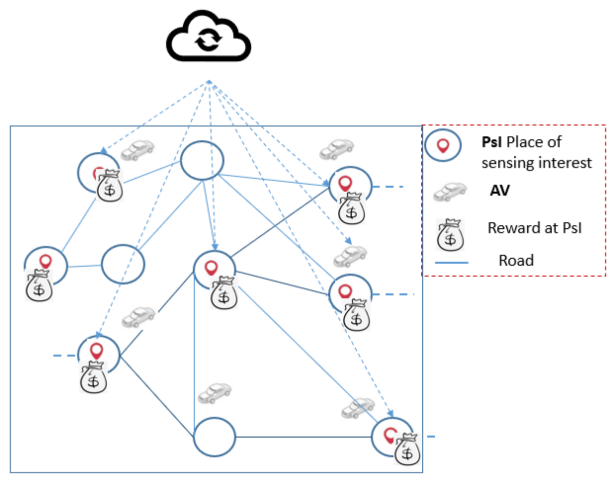


Fig. 1: Vehicular Crowdsensing sketch

the offered rewards are willing to visit and collect samples from PsI located out of their pre-planned trajectories. The success of such a VCS system heavily depends on the quality of data sensed by AVs. Thus, ensuring high Quality of Sensing Data (QoSD) is critical for the successful deployment of such a VCS system. In this paper, we quantify the utility of the platform as a function of the QoSD provided by participants, namely how well spread out is the data gathered from PsIs through a given observation period (i.e., temporal coverage).

Unfortunately, when a participant performs a sensing task, it incurs in costs associated to data collection, they include gas, sensors power consumption and calibration, CPU utilization, time, bandwidth, privacy leakage, etc. Thus, providing high QoSD may result in higher costs for participants, which in turn results in less utility for them. Hence, a rational and self-interested participant would be more inclined to devote low level efforts [8] to provide low-quality services. For that reason, an important functionality of the platform is to provide a good incentive mechanism that encourages high collection standards, and avoids the inefficiency of the undesirable equilibrium that results from a flat payment policy by the platform, and a greedy (low cost) data collection strategy by participants.

We tackle this problem by presenting an incentive mechanism for VCS that encourage the collection of high QoSD, while improving participants profits. At the core of the mechanism are the utility functions for both platform and participants, and a smart data collection strategy. On the participant side, the reward for sampling at a particular PsI depends on the time elapsed since the last time any other participant sampled at that location. Thus, sampling at a PsI right after another participant just did it results in a small reward. On the another hand, waiting too long for sensing in the hope of maximizing reward, may be risky for a participant i. In this settings, any other participant j may arrive right before at the PsI, takes i's place, and collect its reward. Under these circumstances, participants compete among each other by deviating from their pre-planned trajectories and selecting to visit and collecting from the PsIs that maximize their utilities. We model this competition as a non-cooperative game in which the set of vehicles are the players, and the trajectories that result from deviating and visiting their selected PoIs, are the participants' strategies. We show that none of the participants can improve their utilities by unilaterally deviating from their current strategies. We also show that the set of participants trajectories increases platform utility (QoSF), spatial coverage, road utilization, and average participant's utility. In this paper, we assume that all vehicles are autonomous, namely passenger decision making is not involved. In this work the terms user, participant, player, contributor, AV, and vehicle are used indistinctly.

The followings items summarize the paper's main contributions:

- We present a VCS market in which time dependable sensing and temporal coverage drive the utilities of contributors and crowdsourcers.
- We formulate the VCS Temporal Coverage Nash equilibrium (NE) problem.
- We design a greedy iterative algorithm to approximate VCS Temporal Coverage NE.
- We evaluate our algorithms using a real-world traffic map, and the state of the art traffic simulator SUMO

II. Related Work

Here, we present a brief review the of most related works to the proposed method and its differentiating properties.

One of the first works on VCS is presented by Hi *et al.* [4]. Here, authors address the problem of recruiting a subset of participants to maximize spatial, and temporal coverage subject to a limited budget k. The authors tackle the problem by taking advantage of the predictable patterns of vehicles trajectories, and thus, they recruit the future combination vehicles that maximize these metrics. This work is similar to our approach in the sense of using VCS to maximize temporal and spatial coverage. However, unlike our work, this approach doesn't provide any mechanism to collect sensing samples located out the vehicles pre-planned trajectories. Similar to Hi's *et al.* [4] work are the approaches in [3, 12]. Here, the authors also look in the future for best combination of vehicles to maximize the aforementioned metrics. However, this time using trajectories segments rather than the whole trajectories. Again, there is not provision for collection out of the planned vehicles trajectories.

Close to our approach is the work of Zhu *et al.* [13]. Here, the platform uses a reverse auction to encourage participants to generate trajectories and bid for them. The platform on the another hand, uses path planing concepts to pre-compute a set of trajectories that maximize coverage, and acquire those which better match its pre-computed set. Although, Similar to our approach in the sense of reaching PoIs out the vehicles pre-planned trajectories, this work doesn't provide details about the incentive mechanism for data acquisition, cost of participation, or any participant behavioral model.

Following a game theoretical approach Xiao *et al.* [11] propose VCS incentive mechanism in the context of vehicular networks. Here, information quality is influenced by the vehicle speed and its radio channel conditions. In this settings, a game between the platform and participants takes place, in which the accuracy of the sensing report is the participant strategy and a payment policy is the platform strategy. This approach is similar to our work in terms of the game design at high level, namely, they look for a free of noise set of samples Nash Equilibrium. However, our focus is on maximizing temporal coverage while reaching PsI located out of the participants trajectories.

III. System Model

This section presents the main components of the proposed incentive mechanism for VCS-QoSD, and how these components relate each other. The platform's goal is to incentivise the participants (AVs) to collect "high quality" samples of measurement variables at a set of points of sensing interest (PsI). The PsIs are typically located out of the pre-planned trajectories for AVs. The quality of the samples is quantified using a platform utility function which is maximized when the sample points are uniformly distributed over observation time (temporal coverage).

The AVs are rewarded with a monetary amount of $R\delta$ per sample, where R is a fixed amount and the multiplier δ is a carefully designed factor which aims to improve the quality of samples. The underlying assumption here is that the AVs are behaving in a selfish and rational way with complete knowledge about the original trajectories of other participating AVs. Each AV receives some reward for collecting a sample, but also incurs some cost for deviating from its pre-planned trajectory. The multiplier δ depends on the sample time of the previous sample, and is designed in such a way that when the AVs choose targets in order to maximize their utility (reward-cost), the resulting samples will be of high quality. Note that since the utility function of an AV is parameterized by the choices of other AVs, the task of AVs choosing a PsI to visit constitutes a competitive game. Any choices made by the AVs should constitute a Nash equilibrium if one exists.

Also, in order to simplify the task at hand, we make the following assumptions: 1) all vehicles start from their source location at the same time, 2) each vehicle deviates from its original trajectory to visit at most one PsI, and 3) the vehicles don't stop at the PsI and the sample time will be equal to the time of arrival at the PsI. Section VI discusses how the proposed mechanism may be modified in case these assumptions are not met.

In what follows, we will formalize and quantify the platform utility function and the utility functions for AVs. We will provide an intuitive justification for why the chosen utility function for the AVs will result in a high utility for the platform. We will also describe an algorithm which can be used by the AVs to choose a PsI to visit such that the choices for PsIs for all AVs constitute a Nash equilibrium.

A. Notations

This section describes the notations used to denote various entities. VCS-QoSD includes multiple PsIs and AVs (participants). The set of M PsIs is denoted by $T = \{t_1, t_2, \ldots, t_M\}$, and the set of N AVs is denoted by $V = \{v_1, v_2, \ldots, N\}$. The sets of source and destination locations for AVs are denoted by $S = \{s_1, s_1, \ldots, s_N\}$ and $D = \{d_1, d_2, \ldots, d_N\}$ respectively.

B. Platform utility function

The quality of the samples in the sense of temporal coverage can be determined by how close the samples are to uniform sampling in time, that is, the ideal sample points will be when the time periods two consecutive samples are equal. We will use the notion of entropy to quantify the quality of the samples.

If there are n samples at a given PsI, these divide the observation interval of length T_o into $n+1$ intervals. Denote these intervals as $l_i, i = 1, \ldots n+1$. Note that we have $\sum_1^{n+1} l_i = T_o$. We can then normalize these intervals with respect to the total observation time T_o as $p_i = \frac{l_i}{T_o}$. Note the values p_i lie in the interval $[0, 1]$ and $\sum_i p_i = 1$, therefore, we can interpret the set of these values as a probability distribution P.

In case of uniform sampling (the ideal case), we will obtain a uniform probability distribution. The utility function should achieve its maximum value in this case. The worst case in terms of temporal coverage is when all the samples are located at the beginning or at the end of the observation period. In this case, the probability distribution will have one of the probability as 1 and the remaining probabilities will be zero. The utility function should achieve its smallest value in this case and increase as the sample points are transitioned from the worst case to the ideal case. The entropy function given as

$$E(P) = \sum_i p_i \ln \frac{1}{p_i} \tag{1}$$

has all the desired properties mentioned above. It can be shown that $\sum_{i=1}^{n} p_i \ln(1/p_i i) \leq \ln(n)$, and the maximum value is achieved when all p_is are equal, i.e., all inter-sample periods are equal [1]. Furthermore, by defining $\lim_{p \to 0} p \ln 1/p = 0$ (to make $p \ln 1/p$ continuous), $E(P) = 0$ when one of the probability is equal to 1.

The quality of the PsI t_j is quantified by $\mathcal{U}_j$ (utility of t_j) which is calculated using the sample times at the PsI t_j. The utility of the platform is then given as

$$\mathcal{U} = \frac{1}{M} \sum_{j=1}^{M} \mathcal{U}_j, \tag{2}$$

the average of the utilities at all PsIs.

C. AV utility function

Each AV v_i has a pre-planned trajectory $s_i \to d_i$ from its source to the destination. The goal of the platform is to incentivize the AVs to deviate from their pre-planned trajectories to collect good quality samples at the PsIs. The platform does so by assigning a reward $R\delta$ for every sample, where R is a fixed maximum reward, and δ is a multiplier in the interval $[0, 1]$ which changes from sample to sample. The utility function for the AV v_i for collecting a sample at PsI t_k is then given as

$$u_i(t_k) = R\delta - \alpha\left(|s_i t_k d_i| - |s_i t_i|\right), \tag{3}$$

where $|s_i t_k d_i|$ is the length of the trajectory $s_i \to t_k \to d_i$, and $|s_i t_i|$ is the length of the trajectory $s_i \to d_i$. The regulatory parameter α is used to compare the cost of extra distance travelled to the monetary reward obtained by collecting the sample.

We will now turn our attention to the multiplier δ. As the primary goal of the platform is to analyze temporal variations of a sensing variables over an observation period, any sample obtained immediately after an another offers almost no additional value[1]. The multiplier δ should be chosen so as to incentivise the AVs to obtain samples which are "spread out".

Let a_{ik} be the arrival/sample time[2] of the AV v_i at the PsI t_k, and a_{ik}^- be the sample time of the most recent sample (other than that by v_i) at t_k. If v_i is the first AV to arrive at t_k, a_{ik}^- is set to $-\infty$. Then, δ should have the following properties: 1) As $a_{ik} \to a_{ik}^-$, we should have $\delta \to 0$, 2) as $a_{ik} - a_{ik}^- \to \infty$, we should have $\delta \to 1$, and 3) δ should be a concave function of $a_{ik} - a_{ik}^-$ to reflect diminishing rate of return as the sample interval increases. Note that the value of δ depends on choice/allocation of PsIs of other vehicles due to the use of the term a_{ik}^-. Let v_j^* denote the PsI chosen by the AV v_j to visit. Therefore, δ is parameterized by $a_{ik} - a_{ik}^-$ and as a consequence, the choices $v_j^*, j \neq i$ of all other AVs. One such function for δ which satisfies all the above properties is

$$\delta\left(a_{ik} - a_{ik}^-, \{v_j^*, j \neq i\}\right) = 1 - e^{-\frac{a_{ik} - a_{ik}^-}{\tau}}, \tag{4}$$

where τ is acting as a controlling parameter setting the desired interval between two samples. Using equations (3) and (4), we can write the utility function for AVs as :

$$u_i\left(t_k, \{v_j^*, j \neq i\}\right) = R\left(1 - e^{-\frac{a_{ik} - a_{ik}^-}{\tau}}\right) - \alpha\left(|s_i t_k d_i| - |s_i t_i|\right) \tag{5}$$

A rational AV v_i will chose to visit the PsI v_i^* which will maximize its utility as

[1]Assuming the sensors on the AVs are reasonably accurate

[2]We assume the arrival time can be estimated with reasonable accuracy.

$$v_i^* = \arg\max_{t_k} u_i\left(t_k, \{v_j^*, j \neq i\}\right), \tag{6}$$

as long as the utility for visiting v_i^* is positive. Note that if equation 6 is satisfied for all i, such an assignment constitutes a Nash equilibrium.

D. PsI allocation algorithm

The optimization in equation (6) constitutes a game $\langle V, T, \{u_i\}\rangle$, where the players are the AVs (V), the actions available to each AV are the choices of PsIs (T), and each AV v_i has a utility function u_i which determines the utility of choosing a PsI v_i^* given the choices of all other AVs $v_j^*, j \neq i$. The allocation choice according to equation (6) represents a Nash equilibrium, which if exists, will be chosen by rational players.

Note that since each vehicle has $M+1$ possible allocations (visting one of $M+1$ PsIs, or vising none of them), there are a total of $(M+1)^N$ possible PsI allocations to vehicles. An exhaustive search for the best Nash equilibrium over all possible allocations is therefore not feasible. We therefore present a heuristic iterative algorithm (1). We call the allocation of PsIs to AVs resulting from this algorithm, the *Smart PsI Allocation*, denoted as **SPA** (pseudocode in table 1).

*1) Overview of **SPA**:* We now provide a description of **SPA**. We refer to the variables used in the pseudocode as we describe the algorithm. Initially, all vehicles are assigned PsIs randomly. We maintain a set S of vehicles whose optimal PsI allocation is yet to be computed. Note that initially, this set contains all vehicles (line 7). Then, we remove one vehicle v at random from the set S and compute the best PsI t_k given the current allocation of other vehicles according to equation (6). If the current PsI t_1 assigned to v is different from t_k, then t_k is assigned to v.

When the PsI allocation for v is changed as above, there are only two other vehicles whose utilities will be modified. The utility of the vehicle arriving at $t1$ immediately after v will increase, and the utility of the vehicle arriving at t_k immediately after v will decrease. We refer to the latter as successor of v, denoted as sc. This means that the best PsI allocation for sc should be recomputed and sc is added to S at this point. Note that this step will have no effect if sc is already in S because S is implemented as a set.

2) Stopping criteria: If a Nash equilibrium is reached, then by definition, the PsI allocation for each vehicle is the best given the allocations of all other vehicles. We now justify that when a Nash equilibrium is reached, the outer while-loop will terminate. Initially, S is initialized with all the vehicles (line 7). Note that for any vehicle v removed from S, the current assignment t_1 will be the best assignment. This means the assignment for v will not change, and no new vehicle will be added to S. Eventually, the set S will get empty without changing any assignment leaving the *count* variable at 0, which is the condition for the termination of the outer loop.

There could be cases where a Nash equilibrium does not exist. In these cases, the iterative algorithm given in table (1) will not converge. In what follows, we will introduce some formalization of the iterative algorithm to clarify this phenomenon, and provide a solution for the convergence problem.

*3) **SPA** formalization:* We will formalize the iterative process in **SPA** as a forward propagation on a directed graph. Define the state space $\Phi = \{(v_1^*, v_2^*, \ldots, v_N^*)\}$ as the collection of ordered tuples of PsI allocations. Note that since each AV can be assigned any PsI (or not assigned any), the total number of states in Φ is equal to $(M+1)^N$. Then, consider a directed graph $G = (\Phi, E)$, where $(\phi_a, \phi_b) \in E$ is an edge if the tuples ϕ_a and ϕ_b differ only in one location, say i^{th} location, and ϕ_{bi} is the best PsI for AV v_i (according to equation (6)) given all $\{v_j^*, j \neq i\}$ in ϕ_a and ϕ_b.

***Example** 1:* Consider a case where $N = 3$ and $M = 2$, with the set of vehicles $\{v_1, v_2, v_3\}$ and the set of PsIs $\{t_1, t_2\}$. Then an example state could be $\phi_a = (t_1, t_2, t_1)$ implying that the vehicles v_1, v_2, and v_3 are assigned the PsIs t_1, t_2 and t_1 respectively. Now, given these assignments for v_1 and v_2, let the best assignment for v_3 be t_2. Consider the state $\phi_b = (t_1, t_2, t_2)$. Note that ϕ_a and ϕ_b differ in only one location (3^{rd}), and $\phi_{b3} = t_2$ is the best PsI assignments for v_3 given the assignments for other vehicles. In this case, (ϕ_a, ϕ_b) will be an edge in the graph.

Referring to algorithm (1), the first for-loop represents picking a random state in the graph. Lines 10 through 12 describe picking a AV v at random (picking a random location in the tuple) and determining the best PsI v^*. Lines 13 through 15 pair v with v^*. This new paring changes the assignment of a single vehicle to the best PsI while keeping all other assignments constant. Therefore, this change of state corresponds to traversing G by one edge in the forward direction.

4) Nash equilibrium: Section III-D2 provides an intuitive explanation of why the algorithm will stop at (and only at) a Nash equilibrium. We will now provide a formal proof. Recall that given a directed graph $G = (\Phi, E)$, the *in-degree* $d_i(\phi)$ of a vertex $\phi \in \Phi$ is the number of edges pointing into ϕ, i.e., the number of edges where ϕ is the second vertex. Similarly, the *out-degree* $d_o(\phi)$ of ϕ is the number of edges pointing out of ϕ. We can now state the following lemma:

***Lemma** 1:* A state ϕ_e will constitute a Nash equilibrium if and only if $d_o(\phi_e) = 0$.

Proof: Assume ϕ_e is a Nash equilibrium, and that $d_o(\phi_e) > 0$. Then $\exists$ and edge $e \in E$ such that $e = (\phi_e, \phi_f)$. According to the definition of the edges in the graph, ϕ_e and ϕ_f differ in exactly one location, say i, and that

$$u_i(\phi_{fi}, \{\phi_{ej}, j \neq i\}) > u_i(\phi_{ei}, \{\phi_{ej}, j \neq i\})$$

.

On the other hand, according to the definition of the Nash equilibrium, we have

$$\phi_{ei} = \arg\max_t u_i(t, \{\phi_{ej}, j \neq i\}),$$

leading to a contradiction. This means that if ϕ_e is a Nash equilibrium, then $d_o(\phi_e) = 0$.

Now, for any vertex $\phi_e \in \Phi$, assume that $d_o(\phi_e) = 0$. This means that for each i, ϕ_{ei} is the best assignment for v_i given all other assignments. Otherwise, there would have been an edge starting at ϕ_e making its out-degree positive. This means that ϕ_e is a Nash equilibrium. □

If the algorithm reaches a state with no outward edges, this would mean all AVs are assigned the best PsI, the *if* statement in line 13 will never evaluate to being true, and the algorithm will stop.

If a Nash equilibrium does not exist, G must have a cycle since G is a finite graph and every vertex must have at least one outward edge. In these cases, the algorithm **SPA** will get stuck in such a cycle. **SPA** therefore detects such a cycle, and uses a subroutine to break it. This is done by reducing the reward R to remove one of the edges in the cycle. Lines 8,19 and 22 in **SPA** are dedicated to detecting a cycle.

Breaking a cycle: Say we detect a cycle $c = (e_1, e_2, \ldots, e_n)$, where $e_\rho = (\phi_{\rho 1}, \phi_{\rho 2})$ is an edge in the cycle. For each edge e_ρ, let v_ρ be the corresponding AV whose PsI allocation changed from t_k to t_l. This means, given the allocation of all other vehicles, t_l was the best PsI for v_ρ according to equation (6). In order to break this edge, we need to reduce the value of R such that t_l is no longer the best PsI. We will use the short hand notation $u_{\rho k} = u_\rho(t_k, \{v_j^*, j \neq \rho\})$ and $\delta_{\rho k} = \delta\left(a_{\rho k} - a^-_{\rho k}, \{v_j^*, j \neq \rho\}\right)$. Arrange the utilities in descending order as $u_{\rho l} \geq u_{\rho j_1} \geq \cdots \geq u_{rhoj_v} \geq u_{\rho k}$. Observing equation (5), since the costs of deviating to different PsIs are fixed, and given the choices of other vehicles, the δs are fixed. Therefore, the utilities $u_{\rho k}$ are affine functions of R. Let R_ρ be the value of R such that $u_{\rho l} = u_{\rho j_1}$. Then, if we set $R < R_\rho$, t_{j_1} will be the best PsI for v_ρ instead of t_l, thus remove the edge e_ρ from G. We can do this for each edge in the cycle c and set R value to maximum (minus a small number ϵ) of these R_ρs so that we remove exactly one edge from the cycle c, thus breaking it.

To summarize, this section describes an iterative algorithm **SPA** which finds the Nash equilibrium allocation according to equation (6). It does so by traversing a directing graph until it reaches a vertex with zero out-degree. If it encounters a cycles, it reduces the R to break the cycle.

E. Selection of maximum reward

The reward received by an AV for making a measurement at a PsI is given by $R\delta$, where $\delta \in [0, 1]$. Therefore, the R is the maximum reward any AV can get for a sample. As shown in section IV, the average utility $\mathcal{U}$ of the platform as measured by equation (2) increases as the value the R increases, and reaches a maximum value at a certain R_{max}. If the average utility is the primary concern for the platform, then it would set $R = R_{max}$. However, the total reward awarded by the platform also increases with R, and the rate of return measured as $\mathcal{U}/R$ decreases. If the rate of return is the primary concern for the platform then it would set R according to a threshold. Also note that there may not exist a Nash equilibrium according to utilities in equation (5) for large values of R. Existing of Nash equilibrium is necessary to ensure compliance from selfish, rational players (AVs), therefore, this will also be an important consideration when setting the value of R. Note that the reward R is set by the platform, where as the algorithm **SPA** is being executed by the AVs. Therefore, this indicates that there is a certain level of cooperation between the two parties in setting R value.

Algorithm 1 SPA: Smart PsI Allocation

1: **for** $i \leftarrow 1\ to\ N$ **do** ▷ PsI initialization
2: $PsIs[i] \leftarrow rand(1, M)$ ▷ Random allocation
3: **end for**
4: $count \leftarrow N$
5: **while** $count > 0$ **do**
6: $count \leftarrow 0$
7: $S \leftarrow 1, \ldots, N$ ▷ Initialization
8: $ts \leftarrow [PsIs]$ ▷ Transition sequence
9: **while** S is not empty **do**
10: $v \leftarrow$ pop random AV from S
11: $t1 \leftarrow PsIs[v]$
12: $tk \leftarrow getBestPsI(v)$
13: **if** $t1 \neq tk$ **then**
14: $count \leftarrow count + 1$
15: $PsIs[v] \leftarrow tk$
16: $sc \leftarrow successor(v)$
17: **if** sc **then**
18: $S.add(sc)$
19: **end if**
20: $ts.append(PsIs)$
21: **end if**
22: **end while**
23: **if** $ts[0] = ts[end]$ **then** ▷ Cycle detected
24: $R \leftarrow breakCycle(transitionSequence)$
25: **end if**
26: **end while**

IV. Performance Evaluation

This section presents experimental validation of the effectiveness of the proposed VCS-QoSD program. Several experiments were designed to evaluate our proposed approach based on the following metrics: temporal coverage, spatial coverage, road utilization, and average participant utility. We performed experiments on a unit grid and on a real map. Both tor the unit grid and the real map, the source and destination locations were generated randomly as two clusters, and the PsI locations were scattered randomly between the source and destination clusters.

In the unit grid, the trajectory between a source destination pair is a straight line. This simplified environment is used as a proof of concept and as way to illustrate the concepts in a simple way. The second and more realistic simulation environment uses a dense Open Street Map (OSM) from Cologne Germany with drivable roads to allow variation in routing possibilities. Then, the OSM and the participant traces are linked and imported into the traffic simulator SUMO [7].

SUMO provides realistic vehicle movements and routing algorithms on the imported map.

The proposed VCS-QoSD program involves two parties with different motives. On one side, we have the platform whose goal is to incentivise the AVs to deviate from their original trajectories and deliver high quality samples. On the other side, we have the AVs whose goal is to maximize their utility. For the program to be effective, we have to show that 1) **Temporal coverage:** VCS-QoSD indeed provides a way for the platform to obtain high quality samples, and a way to increase the quality by spending more money, and 2) **Particpant utility** the AVs are appropriately incentivised to use the proposed **SPA** algorithm to maximize their utility, and more importantly 3) **Systemic incentives** that the incentives for the platform and AVs are compatible, that is, the AVs acting selfishly to improve their own utility also improve the platform utility as a by-product.

In order to aid in demonstration of the effectiveness of VCS-QoSD, we introduce a naive reward mechanism (**N**) for the platform, and a greedy algorithm (**G**) for the AVs. The reward mechanism in VCS-QoSD is denoted by **V**. Under the **N** reward mechanism, the platform provides a reward of R for any sample taken by the AVs irrespective of the sample time. In the **G** algorithm, the vehicles simply select the PsI with the lowest cost and a positive utility. Note that the algorithm **G** produces a Nash equilibrium for the AVs under **N** reward mechanism.

A. Temporal Coverage

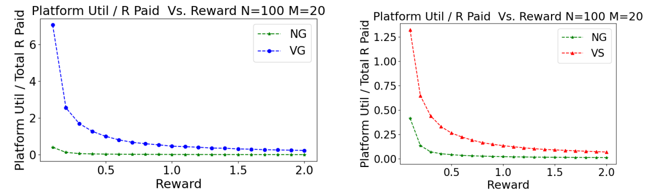
We compare the proposed reward mechanism **V** with **N** reward mechanism. Figure 2a shows the comparison of average platform utility $\mathcal{U}$ per unit reward amount paid out (y-axis) between these two reward mechanisms. The x-axis represents the R value. As shown in the figure, the **V** reward mechanism produces a higher rate of return on the money spent. In these simulations, the AVs are using a the G algorithm for PsI allocation. Figure 3 on the other hand illustrates the ability of the platform to improve the quality of samples by spending more money using the **V** reward mechanism. Note that $\mathcal{U}$ is an increasing function of R. Also, as seen the figure, such a control on the quality of samples is not possible using the **N** reward mechanism.

B. Participant Utility

In section IV-A, we show that the platform benefits from using a **V** reward mechanism if the AVs are using the **G** algorithm. Note that transitioning from the **G** algorithm to **SPA** algorithm for the AVs requires them to share their source and destination locations with the platform and with all other AVs, and dedicate computational resources to the **SPA** algorithm. The AVs have to be appropriately incentivised to do so. Figure 4 compares the average vehicle utilities (y-axis) in the case of **G** and the **SPA** algorithm. The x-axis represents the R value. Note that in both the unit grid and the map cases, the average AV utility is significantly improved when using the **SPA** algorithm, and the incentive for the AVs to swith from **G** to **SPA** increases with increasing R.

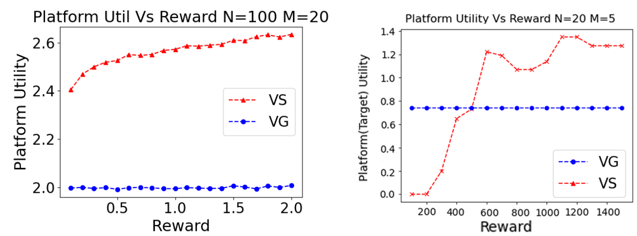
C. Systemic Incentives

In this section, we put together the results from sections IV-A and IV-B to show how the goals of the platform and the goals of the AVs are aligned. This is pictorially represented in figure 5. In the figure, rows represent the choice of the platform (between **V** reward or **N** reward) and the columns represent the choice of the AVs (between **G** and the **SPA** (**S**) algorithms). When the platform is using **N** reward mechanism,



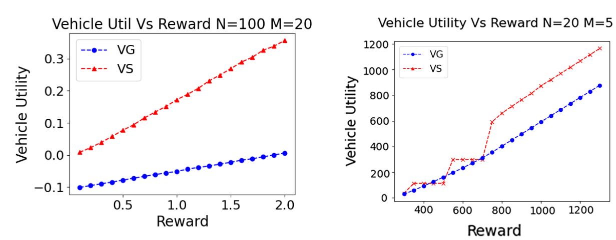
(a) VG vs NG (b) VS vs NG

Fig. 2: (a)Comparison of VCS-QoSD (**V**) rewarding mechanism with a naive (**N**) rewarding mechanism. Figures shows that even when the AVs shift from **G** algorithm to the **SPA** algorithm, the return on investment for platform is superior when **V** rewards are used compared to **N**.



(a) Unit grid (b) Map

Fig. 3: Figure shows the utility of the platform $\mathcal{U}$ increases with R demonstrating the ability of the platform to improve the quality of samples by spending more money in the **V** reward mechanism. The blue line corresponds to **N** reward mechanism.



(a) Participant Utility Grid (b) Participant Utility Map

Fig. 4: AV Utility. Comparison between the greedy (**G**) algorithm, and the **SPA** algorithm. The platform is using **V** reward mechanism in both cases.

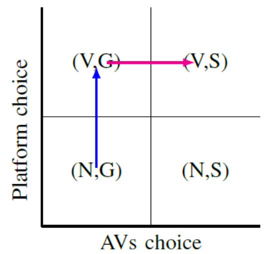
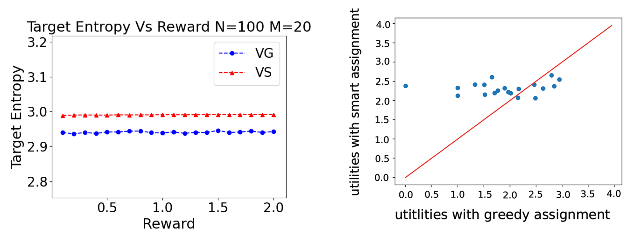


Fig. 5: Figure shows the incentive drives for the players involved. The platform is incentivised to used a well designed reward mechanism to improve return on investment, and the AVs are incentivised to share information and use the target allocation algorithm.



(a) (b)

Fig. 6: Spatial coverage: figure (a) shows that the sample entropy of PsI utilities increases when AVs switch from **G** algorithm to **SPA** algorithm. Figure (b) a scatter plot of PsI utilities in the case of **G** algorithm (x-axis) vs **SPA** algorithm (y-axis). Note that the spread in the PsI utilities is much less in the **SPA** case.

both **G** and **S** algorithms lead to the same result.

As shown in section IV-A, platform will get a better return on investment if it switches to **V** reward mechanism. Also, as seen in section IV-B, when the platform is using **V** reward mechanism, the utility of the AVs is improved when they switch from **G** algorithm to **S** algorithm. Therefore, starting from the base case of (**N**,**G**), the system ends up in the state of (**V**,**S**) as both the platform and the AVs behave selfishly. Finally, figure 2b shows that the return on investment is improved for the platform when the system switches from the base case (**N**,**G**) to the equilibrium case of (**V**,**S**).

As an interesting by-product of the **SPA** algorithm, we also observe that sample entropy of the PsI utilities increases (illustrated in figure 6) when the AVs switch from **G** algorithm to **S** algorithm. This leads to **spatial coverage** as the spread in sample quality at different PsIs goes down, resulting in another aspect where the goals of the platform and the AVs are aligned and both benefit while acting selfishly. The underlying reason behind this is that the **V** reward mechanism disincentivises AVs from taking sample close together in time, and therefore the AVs are forced to seek out distant PsIs to avoid sample crowding.

D. Road Utilization

The goal of this experiment is to explore the effects of reward on road network utilization. Figure 7 visualizes the set of trajectories that result from using our proposed **SPA** algorithm versus the **G** algorithm. Here, different trajectories are overlaid on the map of Cologne Germany. Only a part of the simulated map is shown here for the sake of clarity. It can be seen that the a greater portion of the roads are utilized in case of the **SPA** algorithm.

(a) Greedy (b) Smart

Fig. 7: Road Utilization

Figure 8 shows the effect of reward on road utilization. Here, the values of road utilization for **SPA** and **G** are normalized by baseline. Interestingly **SPA** and **G** have the same road utilization for reward values between 100, and 700. This is explained by the fact that before 700, cost is the dominant factor driving the decision of whether or not visiting a PsI. However, this change for reward values greater than 700, here, PsIs start to attract participants using **PSA** otherwise constrained by cost. On the other hand, **G** is based on maximizing profit by selecting visiting PsI at the lowest cost. Thus, after 700 there is not motivation for visiting other PsIs for those using G.

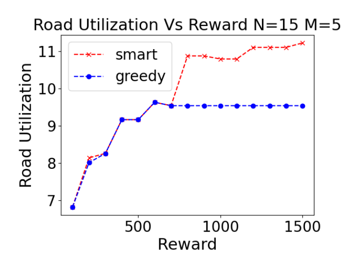


Fig. 8: Road utilization SPA vs G on the Map

V. Acknowledgements

This material is partially supported by the National Science Foundation (NSF) under NSF Award Number 1739409. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s), and do not necessarily reflect those of the NSF.

VI. Conclusion

In this paper, we presented the design and evaluation of an incentive mechanism for VCS that encourages the collection of High Quality Sensing Data (QoSD). We modeled the incentive mechanism as a non-cooperative game where the AVs are the players and their trajectories are their strategies. We formulated the target allocation problem, present utility functions for both participants, and platform, and proposed an iterative algorithm to compute a Nash equilibrium. Using open-street maps, SUMO vehicular traffic simulator, and extensive simulations, we show how our algorithm significantly outperforms a greedy approach in terms of QoSD, average vehicle utility, spatial coverage, and road utilization. We demonstrate that the goals for selfish players are aligned and that the equilibrium state is beneficial for both the platform and the AVs.

References

1 Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

2 Amr S El-Wakeel, Jin Li, Aboelmagd Noureldin, Hossam S Hassanein, and Nizar Zorba. Towards a practical crowdsensing system for road surface conditions monitoring. *IEEE Internet of Things Journal*, 5(6):4672–4685, 2018.

3 Kang Han, Cailian Chen, Qianli Zhao, and Xinping Guan. Trajectory-based node selection scheme in vehicular crowdsensing. In *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1–6. IEEE, 2015.

4 Zongjian He, Jiannong Cao, and Xuefeng Liu. High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2542–2550. IEEE, 2015.

5 Luis G Jaimes, Idalides J Vergara-Laurens, and Andrew Raij. A survey of incentive techniques for mobile crowd sensing. *IEEE Internet of Things Journal*, 2(5):370–380, 2015.

6 Margaret Kosmala, Alycia Crall, Rebecca Cheng, Koen Hufkens, Sandra Henderson, and Andrew D Richardson. Season spotter: Using citizen science to validate and scale plant phenology from near-surface remote sensing. *Remote Sensing*, 8(9):726, 2016.

7 Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

8 Jianfeng Lu, Yun Xin, Zhao Zhang, Xinwang Liu, and Kenli Li. Game-theoretic design of optimal two-sided rating protocols for service exchange dilemma in crowdsourcing. *IEEE Transactions on Information Forensics and Security*, 13(11):2801–2815, 2018.

9 Shuja Jamil Sheikh, Anas Basalamah, Heba Aly, and Moustafa Youssef. Demonstrating map++: A crowdsensing system for automatic map semantics identification. In *Annual International Conference on Sensing, Communication, and Networking (SECON)*, pages 152–154. IEEE, 2014.

10 Cheng Wang, Zhenzhen Zhang, Lu Shao, and MengChu Zhou. Estimating travel speed via sparse vehicular crowdsensing data. In *3rd World Forum on Internet of Things (WF-IoT)*, pages 643–648. IEEE, 2016.

11 Liang Xiao, Tianhua Chen, Caixia Xie, Huaiyu Dai, and H Vincent Poor. Mobile crowdsensing games in vehicular networks. *IEEE Transactions on Vehicular Technology*, 67(2):1535–1545, 2018.

12 Susu Xu, Xinlei Chen, Xidong Pi, Carlee Joe-Wong, Pei Zhang, and Hae Young Noh. Incentivizing vehicular crowdsensing system for large scale smart city applications. In *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019*, volume 10970, page 109701C. International Society for Optics and Photonics, 2019.

13 Xiru Zhu, Shabir Abdul Samadh, and Tzu-Yang Yu. Large scale active vehicular crowdsensing. In *IEEE Vehicular Technology Conference VTC*, pages 1–5. IEEE, 2018.