

Article

Distributed Task Offloading in Heterogeneous Vehicular Crowd Sensing

Yazhi Liu ^{1,*}, Wendong Wang ², Yuekun Ma ¹, Zhigang Yang ¹ and Fuxing Yu ¹

¹ College of Information Engineering, North China University of Science and Technology, Tangshan 063009, China; mayuekun@163.com (Y.M.); yzg-lzy@163.com (Z.Y.); yfx626@126.com (F.Y.)

² State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; wdwang@bupt.edu.cn

* Correspondence: liuyazhi@vip.163.com; Tel.: +86-315-2831-680

Academic Editor: Stefan Poslad

Received: 7 March 2016; Accepted: 11 July 2016; Published: 14 July 2016

Abstract: The ability of road vehicles to efficiently execute different sensing tasks varies because of the heterogeneity in their sensing ability and trajectories. Therefore, the data collection sensing task, which requires tempo-spatial sensing data, becomes a serious problem in vehicular sensing systems, particularly those with limited sensing capabilities. A utility-based sensing task decomposition and offloading algorithm is proposed in this paper. The utility function for a task executed by a certain vehicle is built according to the mobility traces and sensing interfaces of the vehicle, as well as the sensing data type and tempo-spatial coverage requirements of the sensing task. Then, the sensing tasks are decomposed and offloaded to neighboring vehicles according to the utilities of the neighboring vehicles to the decomposed sensing tasks. Real trace-driven simulation shows that the proposed task offloading is able to collect much more comprehensive and uniformly distributed sensing data than other algorithms.

Keywords: vehicular crowd sensing; mobile crowd sensing; task offloading

1. Introduction

In mobile crowd sensing systems, ordinary citizens are recruited to collect and share data from their surrounding environments through mobile devices [1,2]. On the basis of collected sensing data, mobile crowd sensing systems are able to provide users with various novel applications [3], such as real-time air quality reports [4] and road traffic monitoring [5,6]. Various sensing devices and mobile communication interfaces are installed in an increasing number of vehicles, which are able to collect and share various types of sensing data in urban environments. The extensive distribution of these vehicles in urban areas has realized vehicular mobile crowd sensing.

In a vehicular mobile sensing system, sensing data are collected by vehicles and then transmitted to a sensing data center via wireless networks. Vehicular mobile crowd sensing can collect a large amount of comprehensive data, which are collaboratively collected by multiple sensing vehicles. To meet the sensing data requirements of vehicular mobile crowd sensing applications, a large amount of sensing tasks should be generated by the sensing system to collect user-required sensing data. Therefore, vehicular mobile crowd sensing systems mainly collect various kinds of sensing data that encompass all the tempo-spatial dimensions required by sensing tasks. However, sensing vehicles are heterogeneous in terms of sensing interfaces and mobility trajectories. Thus, collecting all required data for sensing tasks by one sensing vehicle is impossible.

Sensing data are temporal, and collected sensing data may expire after certain time. For example, atmospheric temperature changes with time in any given day; as a result, collected temperature data will be expired after a few hours. Moreover, different types of sensing data may have different life

cycles. For example, the life cycle of noise data is much shorter than that of temperature data. Thus, vehicular crowd sensing systems should continuously generate sensing tasks and collect sensing data in target sensing areas to ensure relevance of the user-required data.

In vehicular mobile crowd sensing systems, the quality of collected sensing data increases with the increase in number of sensing vehicles used to collect data, which, consequently, equates to high cost [7]. Therefore, continuous collection of tempo-spatial sensing data required for a task with a limited number of heterogeneous sensing vehicles is a critical issue.

This work proposed a utility-based sensing task decomposition and offloading algorithm. A mobility model based on a continuous-time Markov chain is established to forecast the spatial distribution of sensing vehicles. A utility function is designed to estimate the sensing task execution capacity of sensing vehicles on the basis of the spatial distribution and interfaces of sensing vehicles, tempo-spatial coverage of collected data, requirements of the tempo-spatial coverage, and types of sensing data. Then, the sensing tasks are decomposed and offloaded to neighboring vehicles according to the utilities of the neighboring vehicles to the decomposed sensing tasks.

The remainder of this paper is organized as follows. Section 2 reviews the related studies on mobile crowd sensing and vehicle sensing. Section 3 presents the vehicular mobile crowd sensing system model, the tempo-spatial sensing data model, and the mobility model of the sensing vehicles. A sensing utility function of a vehicle to sensing tasks is proposed in Section 4, based on the heterogeneity of the sensing abilities of the sensing vehicles and the sensing task required sensing data tempo-spatial coverage. Then, a sensing Task Offloading algorithm is proposed based on the proposed utility function in Section 4. Section 5 evaluates the performance of the proposed strategy through real trace-driven simulations. Section 6 presents the conclusions.

2. Related Work

Mobile crowd sensing [1] is a promising means of collecting comprehensive sensing data in urban areas. Different from traditional wireless sensor networks [8,9], sensors are equipped in mobile phones or on board units and carried by people or vehicles. In recent years, a large number of novel applications [3] based on mobile crowd sensing have been deployed. For example, The Common Sense project [10] monitored the contents of pollutants in the air by collecting sensing data from handheld air quality monitors. Yu et al. [11] recovered noise situation throughout New York City (NYC) based on mobile crowd sensing and inferred the fine-grained noise situation at different times of the day for each region of NYC with the use of data from the city complaint platform, together with social media, road network data, and points of interests. The Mahali project [12] used GPS signals that penetrate the ionosphere for science rather than positioning; a large number of ground-based sensors fed data into a cloud-based processing environment through mobile devices, thus enabling a tomographic analysis of the global ionosphere at an unprecedented resolution and coverage. Zhou et al. [13] presented a novel bus arrival time prediction system based on crowd sensing and predicted bus arrival times by encouraging passengers to collect generally available and energy-efficient sensing resources, including cell tower signals, movement statuses, and audio recordings. The TomTom City project [14] provides live and historical traffic and travel information services, and the traffic data is collected by mobile vehicles. However, existing mobile sensing applications suffer from poor sensing data qualities owing to the lack of theoretical research on mobile crowd sensing; thus, the commercial deployment of mobile crowd sensing systems is restricted.

Zhang et al. [15] proposed a four-stage life cycle (i.e., task creation, task assignment, individual task execution, and crowd data integration) to characterize the mobile crowd sensing process. Selecting the most efficient sensing nodes to collect sensing data is a commonly used method in task creation and assignment stages to guarantee coverage of sensing data [16]. The basic principle of participant selection is to collect sensing data with the highest coverage ratio by limited number of participants [17]. Reddy et al. [18] developed a selection framework to allow organizers to identify well-suited sensing nodes for data collection on the basis of their geographic and temporal availabilities, as well as their

habits. Tuncay et al. [19] exploited the stability of user behavior and selected sensing nodes according to the fitness of the mobility history profiles of users. Moreover, modeling the mobility of sensing nodes and forecasting the traces of sensing nodes can reduce the required amount of participating sensing nodes [20,21]. When selecting sensing nodes, forecasting the traces of the nodes with their call logs can reduce energy consumption and result in high sensing data coverage ratio [22].

Song et al. [23] satisfied most requirements of sensing tasks by limited incentives to achieve favorable coverage ratio of collected sensing data. Guang et al. [24] constructed contractual arrangements as incentive mechanism for system administrator to encourage participants to help detecting the malicious user. Chien et al. [25] introduced the online task assignment problem, in which heterogeneous tasks are assigned to workers with different unknown skill sets. Furthermore, this method is able to complete more sensing tasks in time because both the target location of the sensing tasks and the movement range of the participants are considered [26].

Frequent encounters between mobile vehicles provides sensing vehicles opportunity to collaborate with each other. Xiang et al. [27] reduced energy consumption of mobile phone sensing by designing a cloud-assisted collaborative sensing method. Bin et al. [28] generated context characteristics according to collaborative sensing. MOSDEN (mobile sensor data engine) [29] operated on smartphones to collect and share sensing data among multiple distributed applications and users.

Collaboration between vehicles in the task execution stage can further improve the coverage of collected sensing data. For example, in the task execution stage, offloading sensing tasks from mobile devices to the environmental embedded sensors can improve quality of collected sensing data and reduce energy consumption of mobile devices [30,31]. In mobile cloud computing, task offloading [32] migrates the heavy computation task from mobile devices to the cloud to reduce load of mobile devices [33]. Mtibaa et al. [34] proposed a mobile device cloud framework, in which the computing task could be offloaded from one device to another upon interaction. Zheng et al. [35] proposed a participatory computing paradigm to harness the computational power on mobile devices. They propose a general randomized task assignment frame work to minimal workload for individual participating devices. In mobile crowd sensing, Ma et al. [36] delivered collected sensing data between mobile devices to improve data transmission efficiency. Thus, we introduce sensing Task Offloading into the task execution stage to improve quality of collected sensing data.

In early vehicle-based environmental sensing applications, vehicles were used to monitor locations [37], air quality [38], rainy weather [39], or a street view map [40]. However, sensing data are not shared among vehicles in these applications. On the other hand, the emerging Vehicular Ad Hoc Networks (VANETs) reduces the cellular traffic for in-vehicle data services in a cost effective way [41]. Therefore, vehicles are able to exchange information with each other by VANETs [42]. In a previous study [43], multiple probe vehicles were utilized to estimate large-scale traffic in an urban environment. Our previous work [44] proposed a collaborative sensing strategy to collect top quality sensing data with limited sensing vehicles. Vehicular collaborative sensing can also be utilized for safe driving [45]. Collecting all sensing data required by applications is the foremost criterion of vehicular mobile crowd sensing systems. However, none of the previous studies considered decomposition and offloading sensing tasks between sensing vehicles to improve efficiency of sensing task execution.

3. System Model

3.1. System Architecture

A vehicular mobile crowd sensing system is composed of sensing data servers, wireless networks, and vehicles (i.e., sensing nodes), as shown in Figure 1. A sensing node is a vehicle installed with multiple types of sensing devices and wireless communication interfaces. All sensing nodes should be willing to participate in the collection of sensing data. The sensing data center is in charge of a vehicular sensing system, and collected sensing data are stored in the sensing data center. Users of the

sensing system can make queries on the sensing data of interest from the data server. Then, the data requirements are delivered to sensing vehicles in the form of sensing tasks.

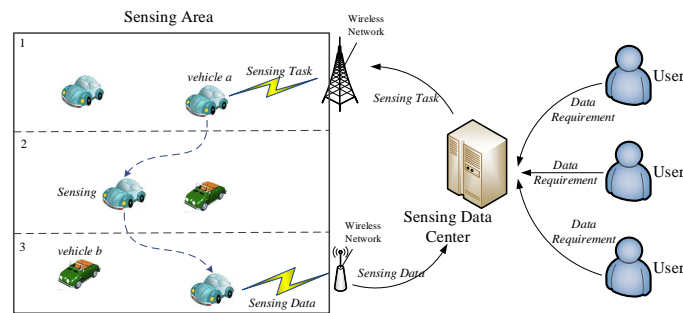


Figure 1. Architecture of vehicular mobile crowd sensing systems.

The sensing data center generates sensing tasks according to the data requirements from the system users. Then, the sensing tasks are delivered to the sensing vehicles to collect matching sensing data. Sensing task-based data collection can collect data with limited sensing resources. In practical scenarios with a significantly large number of sensing tasks and available heterogeneous sensing vehicles, the execution of a sensing task by an appropriate sensing vehicle becomes an arduous problem. This work presents a sensing task decomposition and offloading strategy (Figure 2) that can execute as many as possible sensing tasks with a limited number of vehicles.

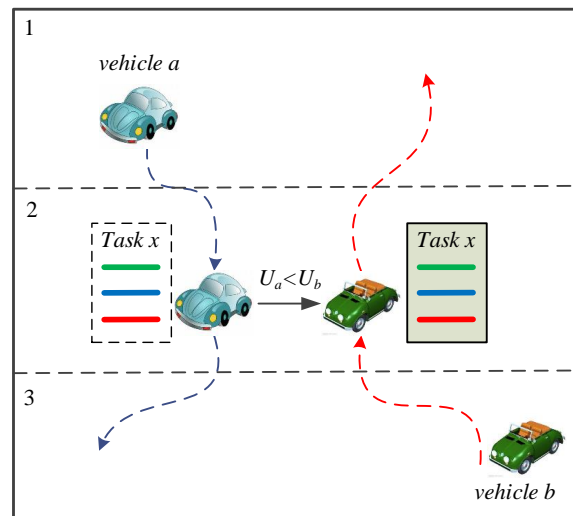


Figure 2. Sensing task offloading.

3.2. Sensing Data Model

The geographic area where the vehicular crowd sensing system is going to collect sensing data is called target sensing area (or sensing area). The sensing area is divided into lattice cells based on the geographical location (Figure 2), and each lattice cell is called a subarea. A subarea is a unit area of the sensing system; the assumption is that sensing data sampled at any point in the subarea can indicate the sensing value of the subarea. The entire sensing area is divided into N subareas in the form of grids based on geographical coordinates (Figure 2). Let $I = \{1, 2, \dots, N\}$ denote the set of subareas in the sensing area. Let i ($i \in I, 1 \leq i \leq N$) denote one of the subareas. Table 1 lists notations used in this paper.

Table 1. List of notations.

Notation	Description
I	The set of subareas
$i, j \in I, 1 \leq i \leq N$	One subarea
w_l	Life cycle of the l th type of sensing data
α, β	Sensing vehicles
$Y_\alpha = (y_1^\alpha, y_2^\alpha, \dots, y_M^\alpha)$	Sensing interfaces of vehicle α
ζ_i^α	The time that vehicle α arrives at subarea i
θ_i	Residence time of vehicle in subarea i
$Z(t)$	Valid sensing data in the sensing data center at time t
$\zeta_{i,l}(t)$	Valid function of the sensing data of type l in subarea i at time t
$X = \{X(t), t \geq 0\}$	Mobility procedure of a vehicle
$P_{ij}(t)$	The transition probability of a vehicle from sub area i to sub area j after time duration t
T_k	Sensing task k
$u_\alpha(T_k)$	Utility of sensing vehicle α to task T_k

Given that sensing data are usually sensitive to time (e.g., air temperature data collected an hour ago are no longer accurate), sensing data should be sampled periodically in each unit sensing area. We suppose that there are M types of sensing interfaces embedded in the vehicle sensing nodes. Let w_l denote the validity period of the l th type of sensing data. Let $Y_\alpha = (y_1^\alpha, y_2^\alpha, \dots, y_M^\alpha)$ denote the sensing ability of the vehicular sensing node α ,

$$y_l^\alpha = \begin{cases} 0 \\ 1 \end{cases} \quad (1)$$

where $y_l^\alpha = 1$ indicated that α can collect the sensing data of type l , and $y_l^\alpha = 0$ indicates that α cannot collect the data of type l .

Vehicles collect sensing data of a specific subarea when it is travelling in the subarea. Let ζ_i^α denote the moment when vehicle α travels into subarea i , and θ_i^α denote the duration vehicle α stay in subarea i . Then, the valid function of the sensing data of type l collected by α when it travels in subarea i could be denoted as follows:

$$h_\alpha(i, l, t) = \begin{cases} y_l^\alpha & \zeta_i^\alpha \leq t \leq \zeta_i^\alpha + \theta_i^\alpha + w_l \\ 0 & \text{other} \end{cases} \quad (2)$$

$Z(t)$ is the valid sensing data in the sensing data center at time t .

$$Z(t) = \begin{bmatrix} \zeta_{1,1}(t) & \zeta_{1,2}(t) & \cdots & \zeta_{1,M}(t) \\ \zeta_{2,1}(t) & \zeta_{2,2}(t) & & \zeta_{2,M}(t) \\ \vdots & & & \vdots \\ \zeta_{N,1} & \zeta_{N,2}(t) & \cdots & \zeta_{N,M}(t) \end{bmatrix} \quad (3)$$

where $\zeta_{i,l}(t)$ is the valid function of the sensing data of type l in subarea i at time t .

$$\zeta_{i,l}(t) \begin{cases} 1 & \text{the sensing data is effective at } t \\ 0 & \text{the sensing data is ineffective at } t \end{cases} \quad (4)$$

3.3. Mobility Model

In the vehicular crowd sensing system, vehicles move in and transfer between the subareas. Let $X = \{X(t), t \geq 0\}$ denote the mobility procedure, where $X(t)$ is the state of the vehicle at time t (i.e., the subarea in which the vehicle stays at time t). Normally, the state of the vehicle is only related

to its state at the last moment. Thus, for any $0 \leq t_0 < t_1 < \dots < t_n < t_{n+1}$, and $i_k \in I, 0 \leq k \leq n+1$, we have

$$\begin{aligned} P\{X(t_{n+1}) = i_{n+1} | X(t_0) = i_0, X(t_1) = i_1, \dots, X(t_n) = i_n\} \\ = P\{X(t_{n+1}) = i_{n+1} | X(t_n) = i_n\} \end{aligned} \quad (5)$$

Equation (5) indicates that the vehicle mobility procedure $X = \{X(t), t \geq 0\}$ is a time-continuous Markov chain. For any $s, t \geq 0$, and $i, j \in I$, we have

$$\begin{aligned} P\{X(s+t) = j | X(s) = i\} \\ = P\{X(t) = j | X(0) = i\} \\ = P_{ij}(t) \end{aligned} \quad (6)$$

where $P_{ij}(t)$ is the transition probability of a vehicle from subarea i to subarea j after time duration t . Let $\mathbf{P}(t) = (P_{ij}(t)), i, j \in I$ denote the matrix of transition probability.

We define the transfer rate matrix of the mobility procedure $X = \{X(t), t \geq 0\}$ as $\mathbf{Q} = (q_{ij})$, which is also called the Q matrix. For any $i \in I$, we define

$$q_{ii} = \lim_{\Delta t \rightarrow 0} \frac{1 - P_{ii}(\Delta t)}{\Delta t} \quad (7)$$

For any $i, j \in I, j \neq i$

$$q_{ij} = \lim_{\Delta t \rightarrow 0} \frac{P_{ij}(\Delta t)}{\Delta t} \quad (8)$$

where q_{ij} is the transition intensity of the vehicle from subarea i to subarea j . Since the number of the subareas in the sensing area is limited, for $\forall i \in I, 0 < \sum_{j \neq i} q_{ij} = q_i < \infty$. The average length of the duration that the vehicle stay in state $X(0) = i$ is determined by q_i . Thus, the value of q_i can be estimated by the historical mobility traces of the vehicles.

According to the Kolmogorov backward differential equation

$$\mathbf{P}'(t) = \mathbf{Q}\mathbf{P}(t) \quad (9)$$

Let $\mathbf{P}(0) = \mathbf{I}$. Then, the transition probability of the Markov chain X could be derived as follows:

$$\mathbf{P}(t) = e^{\mathbf{Q}t} \quad (10)$$

4. Sensing Task Offloading

In this section, a utility function is designed to estimate the sensing task execution capacity of sensing vehicles on the basis of the spatial distribution and interfaces of sensing vehicles, tempo-spatial coverage of collected data, requirements of the tempo-spatial coverage, and types of sensing data. Then, the sensing tasks are decomposed and offloaded to neighboring vehicles according to the utilities of the neighboring vehicles to the decomposed sensing tasks. By decomposing and offloading the sensing tasks, each part of the sensing tasks can be executed more efficiently. Therefore, the sensing efficiency of the whole sensing system will be improved.

4.1. Sensing Task Decomposition

Let $T_k = \{I_k, l_k, \tilde{t}_k\}$ denote a sensing task, where I_k is a set of target subareas of the sensing task, l_k is the required sensing data types of the sensing task, and \tilde{t}_k is the temporal coverage duration of required sensing data. We define the utility of sensing vehicle α to task T_k as follows:

$$u_\alpha(T_k) = \sum_{i \in I_k} \sum_{l \in l_k} \int_{t \in \tilde{t}_k} u_\alpha(i, l, t) dt \quad (11)$$

where $u_\alpha(i, l, t)$ is the utility of sensing vehicle α to the sensing data of type l in subarea i at time t .

$$u_\alpha(i, l, t) = h_\alpha(i, l, t) - h_\alpha(i, l, t) \zeta(i, l, t) \quad (12)$$

where $h_\alpha(i, l, t)$ is the effective duration length of the sensing data of type l collected by α in subarea i . The calculation of the utility of a node will consume resources of the sensing node, and the node should predict its location in finite time. Thus, a system duty cycle t_u is set to reduce the cost of utility calculation. The vehicle trace prediction is within one system duty cycle t_u . The utility of the vehicle sensing nodes are also updated on a t_u cycle. If α arrives the subarea i at time t , and then stay in the subarea i for time τ , (i.e., $\zeta_i^\alpha = t$, $\theta_i^\alpha = \tau$), then

$$\int_{t=0}^{t_u} h_\alpha(i, l, t) dt = \tau + w_l \quad (13)$$

Thus, for any ζ_i^α and θ_i^α , we have

$$\begin{aligned} & \int_{t=0}^{t_u} h_\alpha(i, l, t) dt \\ &= \iint_{t \in (0, t_u), \tau \in (0, t_u - t)} \{P(\zeta_i^\alpha = t) \cdot P(\theta_i^\alpha = \tau) \cdot (\tau + w_l)\} \cdot d\tau dt \end{aligned} \quad (14)$$

where $P(\zeta_i^\alpha = t)$ is the probability density of the time that vehicle α arrives at subarea i . $P(\theta_i^\alpha = \tau)$ is the probability density of the duration that vehicle α stays in subarea i .

The sensing task T_k is decomposed into n subtasks $T_{k1} \cdots T_{kn}$ based on the expressions in Equations (15)–(17)

$$\bigcap_{\alpha=1}^n T_{k\alpha} = \emptyset \quad (15)$$

$$\bigcup_{\alpha=1}^n T_{k\alpha} = T_k \quad (16)$$

$$u_\alpha(i, l, t) \geq u_\beta(i, l, t), \forall (i, l, t) \in T_{k\alpha}, \alpha, \beta \in \Omega. \quad (17)$$

Equations (15) and (16) are the restrictions to sensing task decomposition. It means that the decomposed tasks are not overlapped and the union of all the subtasks equals to the origin sensing task. Equation (17) decomposes the sensing task according to the utility function of the neighbor nodes to the sensing task.

4.2. Distribution of Residence Time

If the sensing node arrives in subarea i at time 0, then $X(0) = i$. If θ_i is the residence time of a vehicle in subarea i , the residence time θ_i can be defined as

$$\theta_i = \inf \{t : t > 0, X(t) \neq X(0), X(0) = i\} \quad (18)$$

then

$$P(\theta_i > t | X(0) = i) = P[X(u) = i, 0 \leq u \leq t | X(0) = i] \quad (19)$$

First, the uncountable events are translated into countable events, as shown in Equation (20).

$$B = \{Y : X(u) = i, 0 \leq u \leq t\} = \bigcap_{0 \leq u \leq t} \{Y : X(u) = i\} \quad (20)$$

where B indicates that the sensing node stays in subarea i in time duration $[0, t]$. Then, the interval $[0, t]$ is uniformly divided into 2^n subintervals as follows:

$$\begin{aligned}
 A_n &= \left\{ Y : X \left(\frac{k}{2^n} t \right) = i, k = 0, 1, 2, \dots, 2^n \right\} \\
 &= \bigcap_{k=0}^{2^n} \left\{ Y : X \left(\frac{k}{2^n} t \right) = i \right\}.
 \end{aligned} \tag{21}$$

A_n is the event where the sensing node is in subarea i at time $\frac{k}{2^n}t$, where $k = 0, 1, 2, \dots, 2^n$. Given that: $A_{n+1} \subset A_n$, it could be denoted as follows:

$$A = \bigcap_{n=1}^{\infty} A_n = \lim_{n \rightarrow \infty} A_n \tag{22}$$

A also indicates the event where the sensing node stays in subarea i from time 0 to time t . Obviously, $B = A$. Therefore,

$$\begin{aligned}
 &P \{ \theta_i > t | X(0) = i \} \\
 &= \lim_{n \rightarrow \infty} P \{ A_n | X(0) = i \} \\
 &= \lim_{n \rightarrow \infty} \exp \left\{ 2^n \ln P_{ii} \left(\frac{t}{2^n} \right) \right\} \\
 &= \lim_{n \rightarrow \infty} \exp \left\{ \frac{\ln [1 - q_i \left(\frac{t}{2^n} \right) + o \left(\frac{t}{2^n} \right)]}{-q_i t / 2^n} (-q_i t) \right\} \\
 &= \exp (-q_i t)
 \end{aligned} \tag{23}$$

Thus, the residence time of a vehicle in subarea i is subject to the exponential distribution with parameter q_i . The average residence time in the subarea is denoted as λ_i , and $\lambda_i = q_i$, which means that the parameter of the exponential distribution can be estimated by the average residence time of the sensing vehicles in the subarea.

Consequently, θ_i obeys the exponential distribution with parameter λ_i , which is the average residence time of the vehicle in subarea i .

$$P \{ \theta_i \leq t | X(0) = i \} = 1 - \exp (-q_i t) \tag{24}$$

where $q_i = \lambda_i$.

4.3. Distribution of Arrival Time

The arrival time of a vehicle in subarea j from subarea i (i.e., $t = 0$) is denoted as ξ_{ij} and defined as

$$\xi_{ij} = \inf \{ t : t > 0, X(t) = j | X(0) = i \} \tag{25}$$

where $\xi_{ij} = 0$ if $X(0) = j$.

$F_{ij}(t)$ is defined as the distribution function of ξ_{ij} and is expressed as

$$F_{ij}(t) = P \{ \xi_{ij} \leq t \}, i, j \in I, i \neq j, t \geq 0. \tag{26}$$

If $\vartheta_1 = \inf \{ t : t > 0, X(t) \neq X(0) \}$ is the moment that the vehicle leaves its initial state and

$$G_i(x) = P(\vartheta_1 \leq x | X(0) = i) = (1 - \exp(-q_i x)) \tag{27}$$

then

$$F_{ij}(t) = q_i^{-1} q_{ij} G_i(t) + q_i^{-1} \sum_{k \neq i, k \neq j, k \in I} q_{ik} \int_0^t F_{kj}(t-u) dG_i(u) \tag{28}$$

If $\vartheta' = \inf \{t : t > 0, X(\vartheta_1 + t) = j\}$ and $X'(t) = X(\vartheta_1 + t)$, then $\vartheta = \vartheta_1 + \vartheta'$. Based on the total probability formula and the strong Markov property, ϑ_1 and $X(\vartheta_1)$ are conditionally independent when $X(0) = i$; thus, Equation (28) is true.

4.4. Distributed Task Offloading

Utility of a sensing vehicle to a sensing task can be derived based on the historical traces of the vehicles, sensing interfaces of the participant vehicles, and task requirements in terms of sensing data types and tempo-spatial coverage. We divide time into duty for convenient calculation. Therefore, each task consists of unit tasks, and each unit sensing task covers a unit subarea, sensing data type, and unit duty cycle.

In Algorithm 1, each utility of the neighbor nodes to each unit sensing task is firstly calculated. Then, the unit tasks are assigned to different neighbors according to the utilities of the neighbors to the unit tasks. We suppose the task set of sensing vehicle α is \mathbf{T} ; the Task Offloading algorithm is shown as Algorithm 1. In Algorithm 1, \mathbf{T} is the task set of α . $T_k \in \mathbf{T}$ is a task in \mathbf{T} . Firstly, α identify its neighbor set \mathbf{N} . Then, each $T_k \in \mathbf{T}$ is divided into unit tasks $\{T_u\}$. For each T_u , find the neighbor β who has the highest efficiency to execute unit task T_u . Then, insert T_u into task set \mathbf{T}_β , which will be offloaded to β after all tasks $T_k \in \mathbf{T}$ is processed.

Algorithm 1: Distributed Sensing Task Offloading Algorithm

Input: Sensing task set \mathbf{T}

- 1 Find neighbor node set \mathbf{N} ;
- 2 **for each** $T_k \in \mathbf{T}$ **do**
- 3 Decompose T_k to $\{T_u\}$ where T_u is a unit task;
- 4 **for each** $T_u \in \{T_u\}$ **do**
- 5 $\beta \leftarrow \arg \max_{\beta \in \mathbf{N}} u_\beta(T_u)$;
- 6 Add T_u to \mathbf{T}_β ;
- 7 **end**
- 8 **end**
- 9 **for each** $\beta \in \mathbf{N}$ **do**
- 10 Offloading \mathbf{T}_β to β ;
- 11 **end**

5. Experimental Section

T-Drive trajectory [46,47] is utilized to simulate and evaluate the proposed Task Offloading strategy. The T-Drive trajectory includes one-week trajectories of 10357 taxis. The total number of GPS points in this data set is approximately 15 million, and the total distance of the trajectories is 9 million km. The time frequency of location data sampling in T-Drive is set to 5 s. Taxi drivers can usually find an optimal route to a destination based on their experience. Therefore, the T-Drive project attempts to improve the efficiency of the navigation software according to the experience of taxi drivers. The traces of taxis can reflect the characters of passengers. The mobility of taxis is also predictable given that the mobility of passengers is not completely random. For example, Huang et al. [48] designed a vehicle mobility model on the basis of the regular patterns derived from traces of 4000 taxis in Shanghai.

Based on the vehicular traces, we constructed our simulation program using Python language and Numpy package. In the program, we designed a trace based mobility module, where the traces of the 1000 vehicle candidates are preloaded into the memory. Before collecting sensing data and sensing task offloading between sensing vehicles, the vehicles train their time continuous Markov process based mobility model for 24 h. After mobility model training, sensing vehicles periodically collect sensing data and offload sensing tasks according to Algorithm 1.

In our simulations, a rectangular region around the Fourth Ring Road of Beijing is adopted as the target sensing area, between 39.840018° N and 39.993971° N, and between 116.276206° E and 116.494243° E. The traces in the rectangular area are used in the simulations. Although the traces are not limited in the target sensing area, we abstract the area out of the target sensing area as subarea 0, and the subareas in the target sensing area as 1, 2, 3, \dots . The 1000 vehicles with the longest traces in the rectangular region are employed as sensing vehicle candidates, and the traces are used to calculate the arrival and residence time distribution among the subareas.

The default parameters of the experiments are set as follows. The default number of participant sensing vehicles is 60, which are selected randomly from the 1000 sensing vehicle candidates. The total time of the experiment is 533,315 s (>6 days). In reality, the value of most types of sensing data can be modeled as a continuous tempo-spatial functions (e.g., temperature or humidity). According to the function, the best sampling tempo-spatial resolution could be estimated based on the Nyquist theorem. In our simulation, the sensing area is a part of the downtown area of Beijing. The rectangular region is uniformly divided into 100 (10×10) subareas. This resolution is supposed to be high enough for the sensing system. The system duty cycle is set to 1000 s. Ten types of sensors are employed to collect data for the sensing system, and each vehicle has 50% possibility to be equipped with each sensor. The default upper bound of each type of sensing data life cycle is 10 system duty cycles. The simulation parameters and their default values are shown in Table 2.

Table 2. Simulation parameters.

Parameter	Default Value
Simulation time	533,315 s
Number of participant vehicles	60
System duty cycle	1000 s
Sensor types	10
Sensing data life cycle	10 system duty cycles
Area partition	100 (10×10)

We configured number of sensing vehicles, mean of task durations and task generation rate in our simulation, where number of sensing vehicles ranges from 5 to 100; mean of task duration ranges from 1 system duty cycle to 19 system duty cycle; task generation rate ranges from 10 per system duty cycle to 100 per system duty cycle. We did not configure the mobility model parameters during the simulation procedure because of the mobility of the vehicles is determined by the real traces, which are not configurable.

We compared the Task Offloading algorithm with two other algorithms, namely, Random and Dynamic Participant Selection (DPS) [23]. In the Random algorithm, participatory sensing vehicles are selected randomly from all the participants at the beginning of the simulation. In the DPS algorithm, sensing vehicles are selected according to the utilities of candidate sensing vehicles at the beginning of the simulation.

5.1. Results

5.1.1. Impact of the Number of Selected Vehicles

Figure 3 shows the relationship between the number of vehicles that participated in the sensing data collection and the sensing data coverage ratio. The sensing data coverage ratio r_s is the ratio between the temporal coverage of collected sensing data and the product of simulation duration, amount of subareas and number of sensing data types, as shown in Equation (29):

$$r_s = C_s / (N \times M \times T_{total}) \quad (29)$$

where C_s is cumulative temporal coverage of collected sensing data, N is the number of subareas, M is the number of sensing data types, and T_{total} is the duration of simulation.

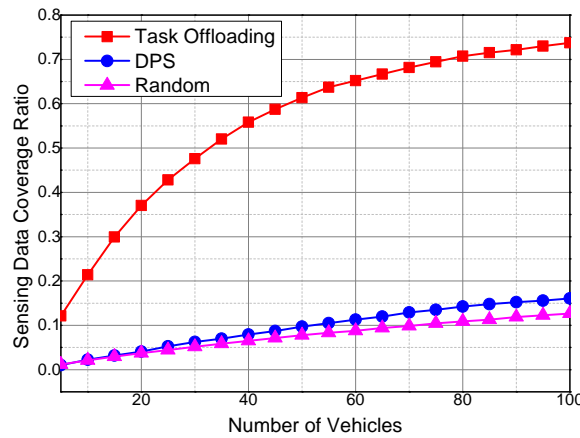


Figure 3. Impact of the number of vehicles on the sensing data coverage ratio.

A sensing record is generated when sampling data, which include sensing time, area, data type, and value, of a sensing node are recorded. Furthermore, the length of the validity period of sensing data is determined by the type of sensing data. Therefore, the validity period is also determined after the sensing time is recorded.

We define a 2-D array S to count the effective coverage of the collected sensing data. $S[i, j]$ is the effective coverage of sensing data with type j in subarea i . $S[i, j]$ is a list of ordered time pairs (t_s, t_e) , where $t_s < t_e$, which denotes a continuous time period. For $\forall p_x, p_y \in S[i, j]$, if $x < y$, then t_e of p_x should be smaller than t_s of p_y . In calculating the cumulative sensing data coverage, we iterate all the collected sensing records to insert effective coverage period pairs into S . When inserting one sensing record into S , its effective sensing period pair is inserted into S according to the type of the sensing data and the subarea where it was collected. The cumulative coverage of collected sensing data should get rid of the impact of overlapping coverage to the same time, subarea and data type. The algorithm for inserting an effective sensing data is shown as Algorithm 2. The input L in Algorithm 2 is a list of ordered time pairs and the input p is the ordered time pair to be inserted into L .

In time pair list L , time pairs are sequential and non-overlapping as shown in Figure 4. After a sensing data are collected, its temporal coverage is denoted as a time pair. Algorithm 2 inserts the collected time pair p into time pair list L , as shown in Figure Algorithm 2. In line 20 of Algorithm 2, a target temp time pair p' is found in L . The start time of p is between the end time of p' and its precursor in L . If the end time of p is smaller than the end time of p' , p will not be processed. If the start time of p is earlier than the start time of p' , then the new start time is set to the start time of p , else the new start time will be set to the start time of p' (line 26–31). Next, the algorithm find the last target pair p'' who is overlapping with p (line 32–42). The new end time is set to the relatively late one in the end time of p and p'' (line 43–47). At last, the new time pair is inserted into L , and the covered temp time pairs are deleted from L (line 50–57).

Then, the cumulative sensing data coverage C_s can be calculated as the sum of all the period lengths in S .

In Figure 3, the sensing data coverage ratios of the three algorithms increase with the increase in the number of participating vehicles. The sensing data coverage ratio of the proposed Task Offloading scheme is approximately four times higher than those of the other two schemes. With task partition and offloading, the tasks are offloaded to the vehicles with higher task utilities, and the vehicles with higher utilities are more likely to complete the corresponding task. In the proposed Task Offloading scheme, the heterogeneous sensing data types and temporal cover ages are considered, and one sampling of the sensing data may cover more than one system duty cycle. Both the Random and DPS algorithms do

not consider the sensing data lifetime, and each sampling of the sensing data covers only one system duty cycle.

Algorithm 2: Time Pair Insertion Algorithm

```

Input: time pair list  $L$ , time pair  $p = (t_s, t_e)$ 
Output: updated time pair list  $L$ 
1  StartTime =  $t_s$ , EndTime =  $t_e$ ;
2  if  $\text{len}(L) == 0$  then
3    append  $p$  to  $L$ ;
4    return;
5  end
6   $i = 0$ ;
7  while  $i < \text{len}(L)$  do
8    TempPair =  $L(i)$ ;
9    TempStartTime = TempPair(0), TempEndTime = TempPair(1);
10   if  $\text{StartTime} \geq \text{TempEndTime}$  then
11      $i++$ ;
12     if  $i \geq \text{len}(L)$  then
13       append  $p$  to  $L$ ;
14       break;
15     end
16   else
17     continue;
18   end
19 end
20 else if  $\text{StartTime} < \text{TempEndTime}$  then
21   if  $\text{EndTime} < \text{TempEndTime}$  then
22     break;
23   end
24   else
25      $j = i$ ;
26     if  $\text{StartTime} < \text{TempStartTime}$  then
27       NewStartTime = StartTime;
28     end
29     else
30       NewStartTime = TempStartTime;
31     end
32     while  $\text{EndTime} \geq \text{TempEndTime}$  do
33       TempPairStack.push(TempPair);
34        $j++$ ;
35       if  $j < \text{len}(L)$  then
36         TempPair =  $L(j)$ ;
37         TempStartTime = TempPair(0), TempEndTime = TempPair(1);
38       end
39       else
40         break;
41       end
42     end
43     if  $\text{EndTime} \geq \text{TempEndTime}$  then
44       NewEndTime = EndTime;
45     end
46     else if  $\text{EndTime} \geq \text{TempStartTime}$  then
47       NewEndTime = TempEndTime;
48       TempPairStack.push(TempPair);
49     end
50     else
51       NewEndTime = EndTime;
52     end
53     NewPair = (NewStartTime, NewEndTime);
54     for  $k$  from 0 to  $\text{len}(\text{TempPairStack})$  do
55       delete  $L(i)$  from  $L$ ;
56     end
57     insert NewPair into  $L$  at  $i$ ; break;
58   end
59 end
60 end

```

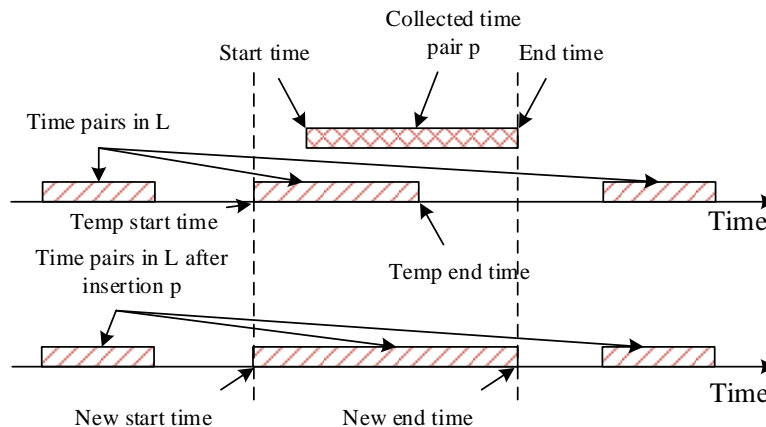


Figure 4. An illustrative diagram to time pair insertion.

Furthermore, the slope of the Task Offloading curve is also steeper than the other two algorithms. In other words, the sensing data coverage ratio in the Task Offloading algorithm increases faster than that of other two algorithms. If more vehicles are recruited into the system, then more opportunities for collaboration between sensing vehicles are generated, resulting in more chances of offloading one sensing task to a sensing vehicle with higher sensing utility. As a result, the sensing data coverage ratio increases much faster than the other two algorithms.

We also compared the sensing data coverage ratio of the Task Offloading algorithm and DPS with period selection of sensing participants (P-DPS). The duty cycle of the both algorithms is set to 1000 s. For convenience, we take temporal coverage of the sensing data into consideration in P-DPS, and the coverage ratios are shown in Figure 5.

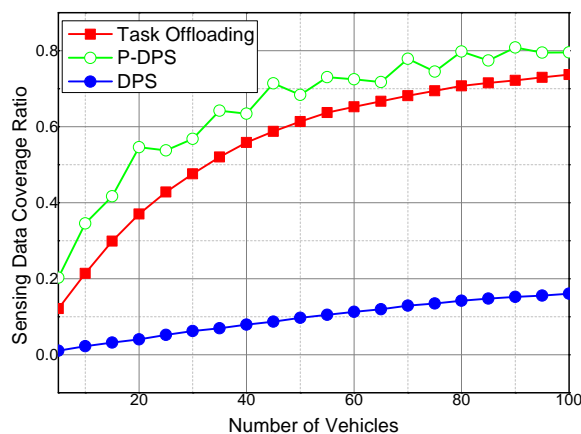


Figure 5. Sensing data coverage ratio of P-DPS, DPS, and Task Offloading.

The sensing server can get the utilities of all the sensing vehicles given that a centralized participant selection method is used in the P-DPS algorithm. Then, the P-DPS algorithm is able to select the best participants out of all the candidates to collect sensing data in the following duty cycle. However, P-DPS cannot maintain high performance in a distributed sensing scenario, due to its requirements to global node information. Task Offloading exchanges sensing task between neighboring vehicles in a distributed way to improve quality of collected sensing data. In the proposed Task Offloading algorithm, the location prediction and task decomposition and offloading are utilized to improve the quality of the collected sensing data. Although the sensing data coverage ratio of Task Offloading is lower than that of P-DPS, the performance of Task Offloading is better than that of DPS in the distributed scenarios.

Figure 6 shows the impact of sensing vehicle amount to the sensing data coverage ratio when the number of sensing vehicles are more than 100. In the first experiment, we suppose that vehicles in a same subarea are neighbors, and they can offload tasks to each other. In the other two experiments, the communication range of the sensing vehicles are set to 500 m and 200 m. Vehicles are neighbors if they are within the communication range of each other. As shown in Figure 6, the coverage ratio raises with the increase of the number of sensing vehicles. But, the increase rate of the coverage ratio decreases along with the number of sensing vehicles. On the other hand, if the communication range is small, the opportunity to offload sensing tasks between the vehicles will be low. As a result, the coverage ratio decreases with the communication range of sensing vehicles.

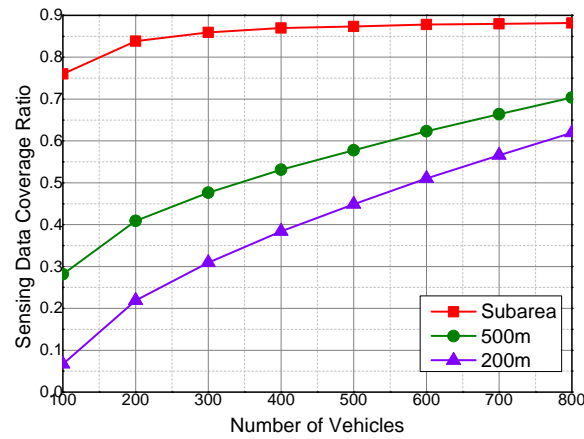


Figure 6. Sensing data coverage ratio with more sensing vehicles.

Figure 7 shows the impact of the number of vehicles on the task coverage ratio. The task coverage ratio r_t is calculated as Equation (30),

$$r_t = C_s / C_t \quad (30)$$

where C_s is cumulative sensing data coverage, and C_t is cumulative task requirements. Both C_s and C_t can be calculated according to Algorithm 2.

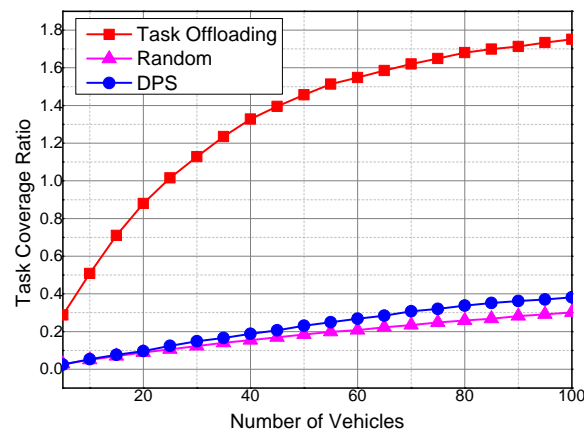


Figure 7. Impact of the number of vehicles on the task coverage ratio.

The task coverage ratios of the three algorithms exhibit the same variation trend with the sensing coverage ratio under the impact of the number of vehicles. However, the task coverage ratios are higher than that of the sensing data coverage ratios with the same number of sensing vehicles. In our task offloading algorithm, since sensing tasks are decomposed and offloaded, each part of the sensing

tasks is offloaded to the most efficient sensing vehicle. As a result, the task offloading algorithm outperforms DPS and Random algorithm, where the sensing tasks are not decomposed and offloaded.

The task coverage ratio is higher than 1 when there are more than 25 sensing vehicles in the Task Offloading algorithm. This phenomenon is unlikely because the task coverage ratio is the ratio between the coverage of collected sensing data and the cumulative coverage of the sensing task requirements. In our simulation, the vehicles continuously collect sensing data if there are sensing task requirements, and the temporal coverage of the collected sensing data may exceed the required range of the sensing task. Therefore, the coverage of the collected sensing data may be larger than the cumulative requirements of the sensing tasks.

Figure 8 shows the spatial entropy with different numbers of sensing vehicles. Spatial entropy is able to reflect the degree of uniformity of the collected sensing data. The spatial entropy of the collected sensing data is large when the sensing data spatial distribution degree is high. The spatial entropy of the three algorithms increase with the increase in the number of sensing vehicles, which means that more sensing vehicles can collect more uniformly distributed sensing data. Given that the Random algorithm chooses the sensing vehicles randomly at the beginning of the simulation, the repetition rate of the collected sensing data in the Random algorithm is higher than that of the other two algorithms. As a result, the spatial entropy of Random is smaller than Task Offloading and DPS under the same number of sensing vehicles.

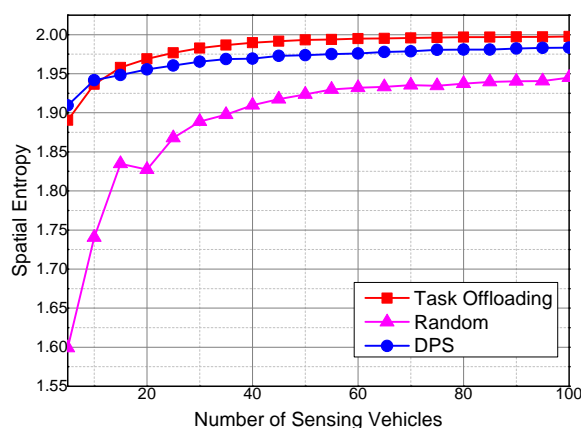


Figure 8. Spatial entropy with different numbers of sensing vehicles.

By offloading the tasks between the sensing vehicles, the Task Offloading algorithm executes the sensing tasks by the vehicles with high utilities. DPS selects the vehicles with higher utilities at the beginning of the simulation. Thus, both Task Offloading and DPS are able to improve the spatial distribution of the collected sensing data. As shown in Figure 8, the Task Offloading algorithm has bigger spatial entropies than the DPS algorithm, because the Task Offloading algorithm further optimizes the task execution efficiency by decomposing and offloading the sensing tasks to the vehicles with higher utilities. However, when the number of selected vehicles is too small (i.e., <10), the chance for the sensing vehicles to collaborate with each other is too small. Thus, the spatial entropy of the Task Offloading algorithm becomes lower than that of the DPS algorithm.

Temporal entropy reveals the stability of the sensing system on collecting the sensing data continuously. As shown in Figure 9, the temporal entropies of the collected sensing data by the three algorithms show different variation trends with the increase in number of sensing vehicles. The temporal entropy of Task Offloading increases with the increase in number of vehicles collecting the sensing data. The growth rate of the temporal entropy of the Task Offloading algorithm decreased with the increase in number of sensing vehicles. The temporal entropy of DPS slightly decreases along with the increase in number of sensing vehicles. The temporal entropies of Task Offloading and

DPS are both higher than that of Random, which means that Task Offloading and DPS are better at continuously collecting sensing data.

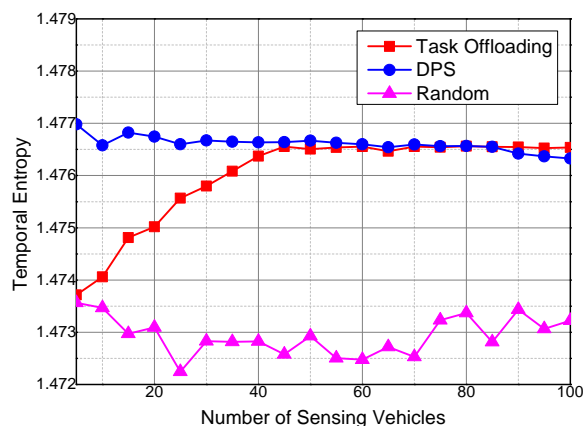


Figure 9. Temporal entropy with different numbers of sensing vehicles.

The temporal entropy of the Random algorithm fluctuates around a small value. The Random algorithm selects sensing vehicles randomly from the candidates; as a result, the Random algorithm experiences difficulties in collecting sensing data continuously. For the same reason, the fluctuation degree of the Random algorithm is also higher than that of the other two algorithms. When the number of sensing vehicles is small, the temporal entropy of Task Offloading is lower than DPS, given that DPS selects vehicles according to their utilities, whereas Task Offloading distributes sensing task randomly to the sensing vehicles at the beginning of the simulation. When the number of sensing vehicles is small, there are less chances of offloading sensing tasks among sensing vehicles. Therefore, the temporal entropy of the Task Offloading algorithm is smaller than that of DPS and higher than that of Random. More opportunities to offload sensing tasks among sensing vehicles are generated with the increase in number of sensing vehicles. Thus, the temporal entropy of Task Offloading exceeds that of DPS at high number of sensing vehicles.

5.1.2. Mean Value of Task Durations

The performance of algorithms under different temporal coverage expectations of the sensing tasks is evaluated. The task coverage ratio is the ratio between the cumulative tempo-spatial coverage of the collected sensing data and the cumulative tempo-spatial coverage of the sensing tasks.

Figure 10 shows the impact of the task duration for the required sensing data temporal coverage on task coverage ratio. The task coverage ratio decreases with the increase in duration of the generated sensing tasks. The increase in temporal coverage duration of a single task would increase the cumulative task duration for the required sensing data temporal coverage, thereby decreasing the task coverage.

In our simulations, each vehicle collects sensing data according to the tempo-spatial coverage requirements of the sensing tasks. Therefore, more sensing task tempo-spatial coverage requirements lead to more sensing data collected by the vehicles. However, the increase rate of the amount of collected sensing data is slower than that of the required tempo-spatial coverage of a task. As a result, the task coverage ratio exponentially decreases with the increase in sensing task duration for the required temporal coverage. The task coverage ratio of Task Offloading is higher than that of the other two algorithms, because sensing tasks are executed by the sensing vehicles with higher sensing utilities.

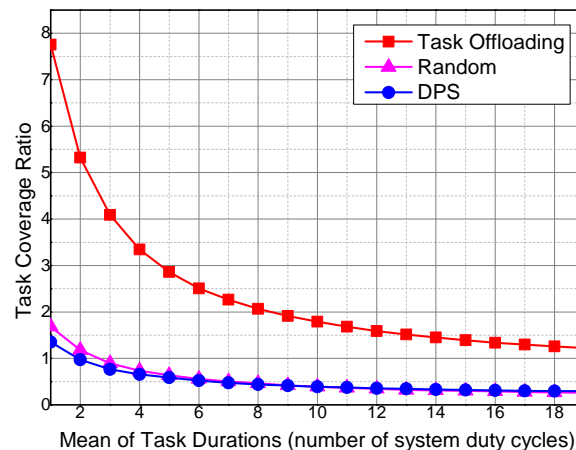


Figure 10. Impact of task durations on task coverage ratio.

The sensing data coverage ratio of Task Offloading is much higher than that of DPS and Random, because Task Offloading considers the lifetime of the collected sensing data, whereas the collected sensing data in Random and DPS cover only one system duty cycle. Furthermore, Task Offloading is able to offload sensing tasks to sensing vehicles with the highest utilities for a specific sensing task, and the sensing data collected by the task performer offer the highest coverage ratio.

Figure 11 reveals the variations of spatial entropy under different sensing task requirements on the temporal coverage duration of the collected sensing data. The required temporal coverage duration of a sensing task has little influence on the spatial entropy of the three algorithms. The spatial entropy of DPS decreases slightly with the increase of sensing task requirements of the temporal coverage duration. In particular, the spatial entropies of Task Offloading and Random are barely affected by the sensing task requirements. The Random algorithm has the lowest spatial entropy, because vehicles are selected randomly and the overlapping ratio of the collected sensing data is high. The DPS and Task Offloading algorithms select sensing vehicles based on their utilities, and thus the collected sensing data are uniformly distributed. As shown in Figure 11, the Task Offloading is able to obtain higher spatial uniformity than DPS in terms of task offloading among sensing vehicles.

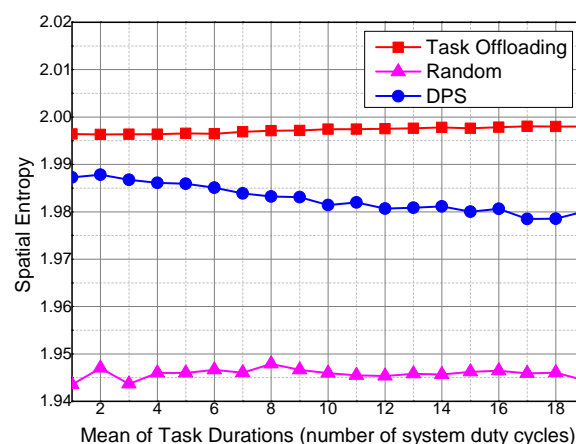


Figure 11. Spatial entropy under different mean values of task durations.

Figure 12 shows the variation trends of temporal entropy of the three algorithms under different sensing task requirements on the temporal coverage duration. The temporal entropy of Task Offloading and DPS is barely affected by the required temporal coverage duration of a sensing task. The temporal entropy of the Random algorithm increases with the increase in required temporal coverage duration of the collected sensing data for a sensing task. The required temporal coverage duration of a sensing

task affects the amount of collected sensing data; more sensing data entail more requirements on the temporal coverage of the collected sensing data. The Task Offloading algorithm optimizes the execution of sensing task by decomposing and offloading the sensing tasks to the sensing vehicles with higher utilities to collect sensing data with less temporal overlapping ratio. Therefore, the temporal entropy of Task Offloading is higher than that of DPS and Random.

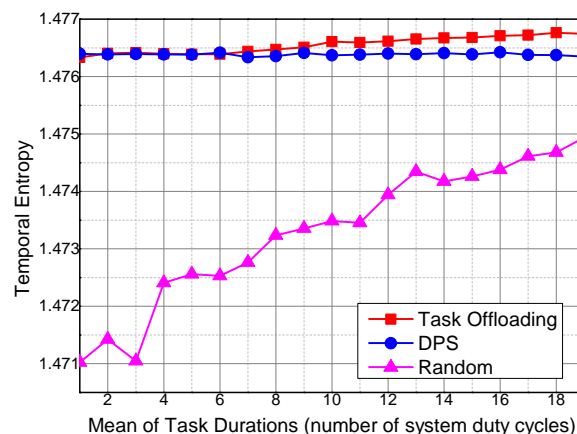


Figure 12. Temporal entropy under different mean values of task durations.

5.1.3. Impact of Task Generation Rate

The impact of task generation rate on the sensing task coverage ratio and tempo-spatial entropy of the collected sensing data is evaluated. The task generation rate denotes the amount of generated sensing task in a system duty cycle. Figure 13 shows the impact of the sensing task generation rate on the task coverage ratio. The task coverage ratio of the three algorithms decreases with the increase in sensing task generation rate. Although generating more sensing tasks can increase the requirements on the tempo-spatial coverage of collected sensing data, and then more sensing data will be collected by the sensing vehicles, the increase rate of the cumulative coverage of the collected sensing data is slower than the cumulative required tempo-spatial coverage duration of sensing data. Thus, the sensing task coverage ratio decreases with the increase in sensing task generation rate.

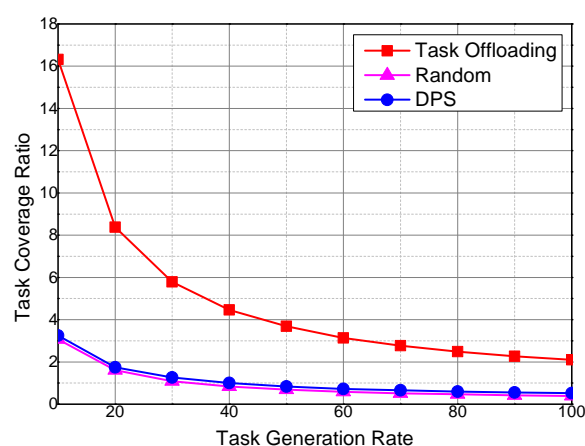


Figure 13. Impact of task generation rate on task coverage ratio.

The task coverage ratio of Task Offloading is higher than that of DPS and Random because the sensing tasks are offloaded to the sensing vehicles with higher utilities. The impact of task generation rate on the task coverage ratio is similar to the impact of required temporal coverage duration of the

sensing data for a sensing task on the task coverage ratio, because sensing task generation rate is increased to increase the tempo-spatial coverage requirements of the sensing tasks.

For the same reason, spatial and temporal entropies have similar variation trends under different task generation rates as that under different temporal coverage of sensing tasks, as shown in Figures 14 and 15.

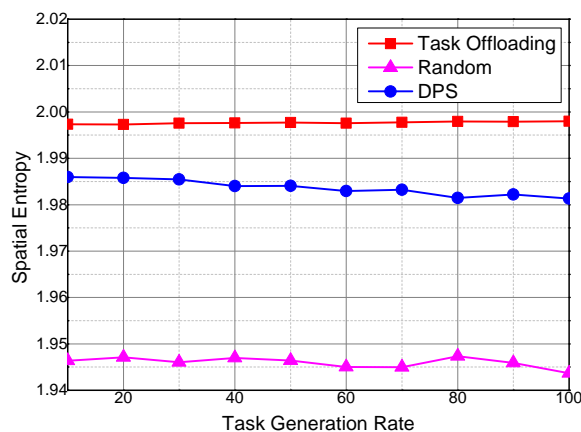


Figure 14. Spatial entropy under different task generation rates.

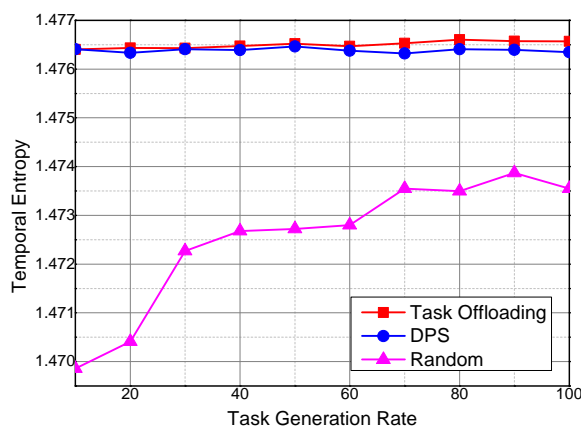


Figure 15. Temporal entropy under different task generation rates.

6. Conclusions

Task Offloading algorithm is proposed for vehicular sensing systems to collect comprehensive covered sensing data. In the proposed Task Offloading algorithm, the mobility of the vehicles is modeled with a time-continuous Markov chain, and the utility function of a vehicle to a sensing task is established according to the required tempo-spatial coverage of the sensing data, required sensing data types, sensing interfaces of the vehicle, and mobility trace of the sensing vehicle. The sensing tasks are decomposed and offloaded to other participants according to the utilities of the sensing vehicles. Real trace-driven simulations show that the proposed Task Offloading algorithm is able to obtain considerably more uniformly-distributed tempo-spatial sensing data and higher task coverage ratio than the other algorithms, namely, Random and DPS.

Acknowledgments: This research was supported by the National Natural Science Foundation of China under Grant No. 61370197.

Author Contributions: Yazhi Liu and Wendong Wang conceived and designed the experiments; Yuekun Ma performed the experiments; Zhigang Yang analyzed the data; Yazhi Liu wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GPS	Global Position System
MOSDEN	MOBILE Sensor Data ENgine
DPS	Dynamic Participant Selection

References

1. Ganti, R.K.; Ye, F.; Lei, H. Mobile Crowdsensing: Current State and Future Challenges. *IEEE Commun. Mag.* **2011**, *49*, 32–39.
2. Guo, B.; Yu, Z.; Zhou, X.; Zhang, D. From participatory sensing to mobile crowd sensing. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM 2014 Workshops), Budapest, Hungary, 24–28 March 2014; pp. 593–598.
3. Khan, W.Z.; Xiang, Y.; Aalsalem, M.Y.; Arshad, Q. Mobile phone sensing systems: a survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 402–427.
4. Devarakonda, S.; Sevusu, P.; Liu, H.; Liu, R.; Iftode, L.; Nath, B. Real-Time Air Quality Monitoring through Mobile Sensing in Metropolitan Areas. In Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing (2013), Chicago, IL, USA, 11 August 2013; pp. 1–8.
5. Zhou, P.; Cheng, Z.; Li, M. Smart traffic monitoring with participatory sensing. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys’13), Rome, Italy, 11–14 November 2013; pp. 1–2.
6. Banerjee, R.; Sinha, A.; Saha, A. Participatory sensing based Traffic Condition Monitoring using Horn Detection. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 567–569.
7. Liu, Y.; Li, X. Heterogeneous Participant Recruitment for Comprehensive Vehicle Sensing. *PLoS ONE* **2015**, *10*, 1–20.
8. Xie, S.; Wang, Y. Construction of Tree Network with Limited Delivery Latency in Homogeneous Wireless Sensor Networks. *Wirel. Person. Commun.* **2014**, *78*, 231–246.
9. Shen, J.; Tan, H.; Wang, J.; Wang, J.; Lee, S. A Novel Routing Protocol Providing Good Transmission Reliability in Underwater Sensor Networks. *J. Internet Technol.* **2015**, *16*, 171–178.
10. Dutta, P.; Aoki, P.M.; Kumar, N.; Mainwaring, A.; Myers, C.; Willet, W.; Woodruff, A. Common sense: Participatory urban sensing using a network of handheld air quality monitors. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys), Berkeley, CA, USA, 4–6 November 2009; pp. 349–350.
11. Zheng, Y.; Liu, T.; Wang, Y.; Zhu, Y.; Liu, Y.; Chang, E. Diagnosing New York City’s Noises with Ubiquitous Data. In Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014), Seattle, WA, USA, 13–17 September 2014; pp. 715–725.
12. Pankrati, V.; Lind, F.; Coster, A.; Erickson, P.; Semeter, J. Mobile Crowd Sensing in Space Weather Monitoring: The Mahali Project. *IEEE Commun. Mag.* **2014**, *52*, 22–28.
13. Zhou, P.; Zheng, Y.; Li, M. How Long to Wait? Predicting Bus Arrival Time With Mobile Phone Based Participatory Sensing. *IEEE Trans. Mob. Comput.* **2014**, *13*, 1228–1241.
14. TomTom. TomTom City: Optimizing Traffic by Linking Road Users, Connected Vehicles and Traffic Management. Available online: https://www.tomtom.com/en_gb/traffic-news/about (accessed on 2 July 2016).
15. Zhang, D.; Wang, L.; Xiong, H.; Guo, B. 4W1H in Mobile Crowd Sensing. *IEEE Commun. Mag.* **2014**, *52*, 42–48.
16. Chon, Y.; Lane, N.D.; Kim, Y.; Zhao, F.; Cha, H. Understanding the coverage and scalability of place-centric crowd sensing. In Proceedings of the 2013 ACM International Joint Conference On Pervasive and Ubiquitous Computing (UbiComp 2013), Zurich, Switzerland, 8–12 September 2013; pp. 3–12.
17. Cardone, G.; Foschini, L.; Bellavista, P.; Corradi, A.; Borcea, C.; Talasila, M.; Curtmola, R. Fostering participation in smart cities: A geo-social crowdsensing platform. *IEEE Commun. Mag.* **2013**, *51*, 112–119.

18. Reddy, S.; Estrin, D.; Srivastava, M. Recruitment framework for participatory sensing data collections. In *Pervasive Computing*; Springer-Verlag: Berlin, Germany; Heidelberg, Germany, 2010; pp. 138–155.
19. Tuncay, G.; Benincasa, G.; Helmy, A. Autonomous and distributed recruitment and data collection framework for opportunistic sensing. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 407–410.
20. Ahmed, A.; Yasumoto, K.; Yamauchi, Y.; Ito, M. Distance and time based node selection for probabilistic coverage in people-centric sensing. In Proceedings of the 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2011), Salt Lake City, UT, USA, 27–30 June 2011; pp. 134–142.
21. Hachem, S.; Pathak, A.; Issarny, V. Probabilistic registration for large-scale mobile participatory sensing. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom 2013), San Diego, CA, USA, 18–22 March 2013; pp. 132–140.
22. Zhang, D.; Xiong, H.; Wang, L.; Chen, G. CrowdRecruiter: Selecting Participants for Piggyback Crowdsensing under Probabilistic Coverage Constraint. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, WA, USA, 13–17 September 2014; pp. 703–714.
23. Song, Z.; Liu, C.H.; Wu, J.; Ma, J.; Wang, W. QoI-Aware Multitask-Oriented Dynamic Participant Selection With Budget Constraints. *IEEE Trans. Veh. Technol.* **2014**, *63*, 4618–4632.
24. Yang, G.; He, S.; Shi, Z. Leveraging Crowdsourcing for Efficient Malicious Users Detection in Large-Scale Social Networks. *IEEE Internet Things J.* **2016**, doi:10.1109/JIOT.2016.2560518.
25. Ho, C.J.; Vaughan, J.W. Online Task Assignment in Crowdsourcing Markets. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; Volume 12, pp. 45–51.
26. He, S.; Shin, D.H.; Zhang, J.; Chen, J. Toward Optimal Allocation of Location Dependent Tasks in Crowdsensing. In Proceedings of the 33rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'14), Toronto, ON, Canada, 27 April–2 May 2014; pp. 745–753.
27. Sheng, X.; Tang, J.; Zhang, W. Energy-efficient collaborative sensing with mobile phones. In Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM'12), Orlando, FL, USA, 25–30 March 2012; pp. 1916–1924.
28. Liu, B.; Jiang, Y.; Sha, F.; Govindan, R. Cloud-enabled privacy-preserving collaborative learning for mobile sensing. In Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, Toronto, ON, Canada, 6–9 November 2012; pp. 57–70.
29. Jayaraman, P.P.; Perera, C.; Georgakopoulos, D.; Zaslavsky, A. Efficient opportunistic sensing using mobile collaborative platform MOSDEN. In Proceedings of the 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (2013), Austin, TX, USA, 20–23 October 2013; pp. 1–10.
30. Rachuri, K.K.; Efstratiou, C.; Leontiadis, I.; Mascolo, C.; Rentfrow, P.J. Smartphone sensing offloading for efficiently supporting social sensing applications. *Pervasive Mob. Comput.* **2014**, *15*, 3–21.
31. Wen, X.; Shao, L.; Xue, Y.; Fang, W. A rapid learning algorithm for vehicle classification. *Inform. Sci.* **2015**, *295*, 395–406.
32. Fernando, N.; Loke, S.W.; Rahayu, W. Mobile cloud computing: A survey. *Future Gener. Comput. Syst.* **2013**, *29*, 84–106.
33. Shi, C.; Lakafosis, V.; Ammar, M.H.; Zegura, E.W. Serendipity: Enabling Remote Computing among Intermittently Connected Mobile Devices. In Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hilton Head Island, SC, USA, 11–14 June 2012; pp. 145–154.
34. Mtibaa, A.; Fahim, A.; Harras, K.A.; Ammar, M.H. Towards Resource Sharing in Mobile Device Clouds: Power Balancing Across Mobile Devices. In Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing, Hong Kong, China, 12 August 2013; pp. 51–56.
35. Dong, Z.; Kong, L.; Cheng, P.; He, L.; Gu, Y.; Fang, L.; Zhu, T.; Liu, C. REPC: Reliable and Efficient Participatory Computing for Mobile Devices. In Proceedings of the Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Singapore, 30 June–3 July 2014; pp. 257–265.
36. Ma, H.; Zhao, D.; Yuan, P. Opportunities in Mobile Crowd Sensing. *IEEE Commun. Mag.* **2014**, *52*, 29–35.

37. Hull, B.; Bychkovsky, V.; Zhang, Y.; Chen, K.; Goraczko, M.; Miu, A.; Shih, E.; Balakrishnan, H.; Madden, S. CarTel: A Distributed Mobile Sensor Computing System. In Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys'06), Boulder, CO, USA, 31 October–3 November 2006; pp. 125–138.
38. Hu, S.C.; Wang, Y.C.; Huang, C.Y.; Tseng, Y.C.; Kuo, L.C.; Chen, C.Y. Vehicular sensing system for CO₂ monitoring applications. In Proceedings of the IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS'09), Shanghai, China, 8 October 2009; pp. 168–171.
39. Görner, S.; Park, S.B.; Egbert, P. Vision-based rain sensing with an in-vehicle camera. In Proceedings of the IEEE Intelligent Vehicles Symposium (2009), Xi'an, China, 3–5 June 2009; pp. 279–284.
40. Hyypä, J.; Jaakkola, A.; Hyypä, H.; Kaartinen, H.; Kukko, A.; Holopainen, M.; Zhu, L.; Vastaranta, M.; Kaasalainen, S.; Krooks, A.; et al. Map updating and change detection using vehicle-based laser scanning. In Proceedings of the IEEE Joint Urban Remote Sensing Event, Shanghai, China, 20–22 May 2009; pp. 1–6.
41. Lee, M.; Song, J.; Jeong, J.P.; Kwon, T.T. DOVE: Data Offloading through Spatio-temporal Rendezvous in Vehicular Networks. In Proceedings of the 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, USA, 3–6 August 2015; pp. 1–8.
42. Bazzi, A.; Masini, B.M.; Zanella, A.; Pasolini, G. IEEE 802.11p for cellular offloading in vehicular sensor networks. *Comput. Commun.* **2015**, *60*, 97–108.
43. Zhu, Y.; Li, Z.; Zhu, H.; Li, M.; Zhang, Q. A Compressive Sensing Approach to Urban Traffic Estimation with Probe Vehicles. *IEEE Trans. Mob. Comput.* **2013**, *12*, 2289–2302.
44. Liu, Y.; Li, X. heterogeneous participant recruitment for comprehensive vehicle sensing. *PLoS ONE* **2015**, *10*, 1–19.
45. Jang, J.; Baik, N. Diagnosing Hazardous Road Surface Conditions Through Probe Vehicle as Mobile Sensing Platform. In Proceedings of the International Conference on Winter Maintenance and Surface Transportation Weather, Coralville, IA, USA, 30 April–3 May 2012; pp. 54–60.
46. Yuan, J.; Zheng, Y.; Xie, X.; Sun, G. Driving with knowledge from the physical world. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11), San Diego, CA, USA, 21–24 August 2011; pp. 316–324.
47. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; Huang, Y. T-drive: Driving directions based on taxi trajectories. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 99–108.
48. Huang, H.; Zhang, D.; Zhu, Y.; Li, M.; Wu, M.Y. A Metropolitan Taxi Mobility Model from Real GPS Traces. *J. Univers. Comput. Sci.* **2012**, *18*, 1072–1092.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).