Journal of Parallel and Distributed Computing 127 (2019) 224–237

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Traffic flow monitoring systems in smart cities: Coverage and distinguishability among vehicles

Huanyang Zheng [a,*], Wei Chang [b], Jie Wu [a]

[a] Computer and Information Sciences, Temple University, Philadelphia, PA, 19122, United States
[b] Computer Science, Saint Joseph's University, Philadelphia, PA, 19131, United States

HIGHLIGHTS

- Placement problem is deeply explored for traffic flow monitoring systems in smart cities. Interesting properties are presented.
- Approximation algorithms and extensive experiments for the placement problem are presented.

ARTICLE INFO

Article history:
Received 17 June 2017
Received in revised form 5 March 2018
Accepted 1 July 2018
Available online 6 August 2018

Keywords:
Traffic flow tracking systems
RSU placement
Coverage and distinguishability
Smart city
Location proof

ABSTRACT

Traffic flow monitoring systems aim to measure and monitor vehicle trajectories in smart cities. Their critical applications include vehicle theft prevention, vehicle localization, and traffic congestion solution. This paper studies an RoadSide Unit (RSU) placement problem in traffic flow monitoring systems, in order to secure vehicles through location proofs. Given some traffic flows on streets, the objective is to place a minimum number of RSUs to cover and distinguish all traffic flows. A traffic flow is covered and distinguishable, if the set of its passing RSUs is non-empty and unique among all traffic flows. The RSU placement problem is NP-hard, monotonic, and non-submodular. It is a non-trivial extension of the traditional set cover problem that is submodular. Three bounded RSU placement algorithms are proposed with respect to the number of given traffic flows. To further reduce the number of deployed RSUs, this paper extends a credential propagation mechanism via vehicle-to-vehicle communications, which essentially enlarges the coverage of an RSU. Extensive real data-driven experiments demonstrate the efficiency and effectiveness of the proposed algorithms.

1. Introduction

Recent breakthroughs on Traffic Flow Monitoring Systems (TFMSs) have enabled accurate measurements and monitors of vehicle trajectories in smart cities. Applications of TFMSs include: (i) vehicle theft preventions by trajectory monitoring [24,17], (ii) vehicle localizations by trajectory analysis and prediction [10], and (iii) traffic congestion solutions by traffic flow management [33]. Due to the growing popularity of location-based vehicle services, the measurement and monitoring capacities of TFMSs would further benefit intelligent transportation systems [24]. Most TFMSs are implemented through WiFi technologies [27], Bluetooth low energy radios [11], or GSMs [18]. These TFMS implementations will deploy RoadSide Units (RSUs) as the infrastructure for measuring and monitoring passing traffic flows. Since RSUs are expensive, the manufacturing cost of a TFMS depends heavily on the placement (or deployment) of the RSU.

This paper considers a scenario where the TFMS measures and monitors vehicles through *location proofs* [36,21,13]. The location proof is a means for a vehicle to demonstrate that it was indeed in a specific traffic flow. RSUs are deployed on streets and broadcast their unique RSU IDs via RSU-messages to passing vehicles. A location proof for a given vehicular trajectory is created based on the collected RSU IDs along the moving path of the vehicle. When a vehicle claims to be in a specific vehicle flow, we must be able to verify this claim by comparing its collected RSU IDs against a known database of every RSU's geographic information. All malicious vehicles, such as thieved vehicles that were in other traffic flows but not in the claimed one, should be unable to obtain the correct set of RSU IDs.

This paper studies an RSU placement problem to reduce the manufacturing cost of the TFMS in a smart city. An example is shown in Fig. 1, which involves multiple streets and intersections. On streets, there exist some given traffic flows, which are composed of moving vehicles. The TFMS should measure and monitor these given traffic flows. An RSU can be placed on a street to broadcast its unique RSU ID to passing vehicles. A traffic flow is said

* Corresponding author.
E-mail address: huanyang.zheng@temple.edu (H. Zheng).

https://doi.org/10.1016/j.jpdc.2018.07.008
0743-7315/

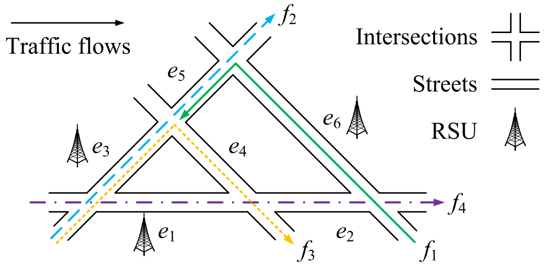


Fig. 1. An illustration of the RSU placement scenario.

to be *covered*, if it goes through at least one RSU. Clearly, all given traffic flows should be covered in the TFMS. Otherwise, some given traffic flows may not be monitored. However, even if all traffic flows are covered, the TFMS cannot measure and monitor different traffic flows. The reason for this is that an RSU cannot distinguish its passing vehicles if they belong to different traffic flows. A covered traffic flow is said to be *distinguishable*, if the set of its passing RSUs is unique among all traffic flows. To measure and monitor traffic flows through secure location proofs, the TFMS should be able to cover and distinguish all given traffic flows in the smart city.

To satisfy the coverage and distinguishability requirements, we can simply place an RSU on each street that is passed by each given traffic flow. However, this placement strategy is impractical, since RSUs are expensive. We should minimize the number of placed RSUs to reduce the manufacturing cost of the TFMS. The objective of this paper is to minimize the number of placed RSUs, and the constraint is that all given traffic flows are covered and distinguishable. Challenges come from the difference between coverage and distinguishability: some given traffic flows can be indistinguishable, even if all given traffic flows are covered. An example is shown Fig. 1, which involves six streets (e_1 to e_6) and four given traffic flows (f_1 to f_4). As an RSU placement strategy, three RSUs are placed on e_1, e_3, and e_6, respectively. Clearly, all given traffic flows are covered, since each traffic flow goes through one RSU. However, f_2 and f_3 are indistinguishable, since they go through the same set of placed RSUs (i.e., the RSU placed on e_3). Consequently, we should place one more RSU on e_4 or e_5 to distinguish f_2 and f_3. The coverage and distinguishability requirements pose unique challenges for our problem.

The RSU placement problem is NP-hard, monotonic, and non-submodular [9,28]. It is a non-trivial extension of the traditional set cover problem that is submodular [3]. Let f and f' denote an arbitrary pair of traffic flows (in terms of their sets of passing streets). We demonstrate that, to cover and distinguish f and f', two RSUs are necessary and sufficient to be placed on the streets from two different subsets of $f \setminus f'$, $f' \setminus f$, and $f \cap f'$. Three RSU placement algorithms are proposed. They are bounded with respect to the number of given traffic flows (denoted by n). The first algorithm iteratively places a pair of streets to cover and distinguish maximum pairs of traffic flows, resulting in a ratio of $n \ln \frac{n(n-1)}{2}$ that belongs to $O(n \ln n)$. The second algorithm places redundant RSUs on streets from each subset of $f \setminus f'$, $f' \setminus f$, and $f \cap f'$. Its approximation ratio is $\frac{n+1}{2} \ln \frac{3n(n-1)}{2}$, which also belongs to $O(n \ln n)$. However, it has a lower time complexity than the first algorithm. The third algorithm has the lowest time complexity, as well as the best ratio of $\ln \frac{n(n+1)}{2}$ that belongs to $O(\ln n)$. It avoids redundant RSU placements by subtly redefining subsets.

The remainder of this paper is organized as follows. Section 2 surveys related works. Section 3 describes the model and formulates the problem. Section 4 analyzes the problem. Section 5 proposes bounded algorithms. Section 6 extends a credential propagation mechanism. Section 7 includes the experiments. Section 8 concludes the paper.

2. Related work

2.1. Existing traffic flow monitoring systems and applications

In the past decade, TFMSs have brought multiple promising and emerging applications to pedestrians and vehicles [24]. One application is vehicle theft prevention through trajectory monitoring. Lee et al. [17] designed a vehicle tracking system using GPS/GSM/GPRS technologies and smartphone applications. Perera et al. [20] monitored traffic flows based on vehicle trajectory predictions. Autowitness [8] can track stolen properties (e.g., vehicles) with robust tolerances of GPS outages. TFMSs can be applied to localize passing pedestrians and vehicles [10]. Jin et al. [15] explored a pedestrian tracking system with sparse infrastructure supports. Sivaraman et al. [25] surveyed recent vehicle detection and localization technologies through RSUs and cameras. Kyun queue technology [23] monitored and localized road traffic queues to manage traffic congestion. Janecek et al. [12] estimated the bus travel time based on the cellular data and the vehicular traffic theory.

2.2. Location-based TFMSs

Popular TFMSs usually measure and monitor vehicles through location proofs [16], which are widely used in the mobile computing community. Location proofs are used to securely demonstrate that a claimer has indeed appeared at a specific location at a specific time. To verify the spatiotemporal claims, different types of schemes are designed. Using the distance-bounding protocols [1] as a common approach, we can measure the physical times/distances for messages to transmit between a claimer and its verifiers, and estimate the claimer's real physical location based on these physical times. However, the accuracy of the distance-bounding approaches relies on the deploying density of the verifiers and their trustworthiness [29]. Recently, people have begun to consider using unique impacts of environmental factors on surrounding objects to create evidence for location verification [32]. Consequently, this paper is motivated by indoor-tracking technologies [10], where a set of collected WiFi data is used to associate identities with different moving objects in surveillance videos. More specifically, we verify the presence of a vehicular trajectory by providing spatiotemporal-bounded messages on some crucial road stretches, the combination of which can cover and uniquely distinguish one trajectory from others.

2.3. Location-based TFMSs with RSU placement

This paper studied the RSU placement problem, where the TFMS places RSUs on streets to monitor passing traffic flows. A similar scenario includes Xu's work [30], which places RSUs for vehicle communications. Randomized and bounded algorithms were introduced to optimize the RSU placement. Zheng and Wu [35] use RSUs in smart cities to disseminate advertisements to drivers on passing vehicles. Reis et al. [22] placed RSUs as intermediate relays, which can improve communications in sparse vehicular networks. This paper differs from classic placement problems [37] in terms of the coverage and distinguishability requirements. Our RSU placement problem extends the traditional set cover problem [3] in terms of the coverage and distinguishability requirements. Given some elements and a collection of sets of elements, the traditional set cover problem aims to select minimum sets to cover all given elements [2]. Elements in a set are covered, if this set is selected. In contrast to the existing literature, our RSU placement problem is not submodular, and is a non-trivial extension of the traditional set cover problem that is submodular [4] (see Table 1).

Table 1
Comparison to existing work.

Comparison	Objective	Problem property
Khan [16]	Vehicle localization	Non-optimization
Xu [30]	Vehicle communication	Optimization, submodular
Zheng [35]	Advertisement dissemination	Optimization, submodular
Reis [22]	Vehicle communication	Optimization, submodular
Our problem	Traffic flow measurement	Optimization, non-submodular

2.4. Submodularity technique

Our problem brings more unique challenges, since it involves submodular techniques and non-submodular techniques. Note that the problem of non-submodular function maximization [6] has not been perfectly solved in the literature [5]. This is because certain properties of the objective function are required to design approximation algorithms. Although the problem of supermodular function maximization can be optimally solved by the minimum-norm-point algorithm [7], non-submodular functions are not the same. The latest approach is based on the curvature [26], which typically assumes that the marginal gain of the non-submodular function varies within a given curvature. This approach is based on modifications of the continuous greedy algorithm and non-oblivious local search, and allows us to approximately maximize the sum of a nonnegative, nondecreasing submodular function and a (possibly negative) linear function. Meanwhile, it has been proved that these approximation results are the best possible in the value oracle model, even in the case of a cardinality constraint.

3. Model and problem formulation

3.1. Model and traffic flow analysis

The RSU placement scenario is based on a directed graph $G = (V, E)$, where V is a set of nodes (i.e., street intersections), and $E \subseteq V^2$ is a set of directed edges (i.e., one-way and two-way streets). We use e_i to denote the ith edge. The graph G includes n given traffic flows of $F = \{f_1, f_2, \ldots, f_n\}$ on the streets. Each given traffic flow is represented as a walk, which is a sequence of edges, i.e., $f = (e_1, e_2, \ldots)$. An example is shown in Fig. 1, where $f_1 = (e_6, e_5)$, $f_2 = (e_3, e_5)$, $f_3 = (e_3, e_4)$, and $f_4 = (e_1, e_2)$. Both nodes and edges can be repeated in a walk. All given traffic flows are unique, i.e., we have $f \neq f'$ for $\forall f, f' \in F$. A given traffic flow is composed of moving vehicles that need to be monitored by the TFMS. Applicable scenarios include vehicle theft prevention, vehicle localization, and traffic congestion management in smart cities.

To obtain the traffic flow information of F, we rely on the GPS approach. GPS-enabled devices are widely spread between drivers making the collection of GPS data more accessible. So there is an opportunity to infer useful traffic flow patterns. Emilian [19] applied a statistical analysis on 10000 vehicle GPS traces, from around 3600 drivers which are mined to extract the outlier traffic pattern. The urban area can be divided into a grid and organizing the road infrastructure as segments in a graph. The relationship between the time (e.g., morning/afternoon or weekdays/weekends) and the traffic flow can be figured out using the GPS trace data. As a result, a state-space analysis can be obtained with respect to the vehicle flow-time pattern. The vehicle flow in this paper refers to all vehicle flows that are aggregated over time in the state-space analysis. Note that, in practice, vehicles move in an arbitrary manner on the road depending on the traffic density and the pathway ahead. Therefore, it is impossible to track the full trace of each vehicle. To resolve this problem, we can just focus on the major roads with in the city, such as expressways and highways. In other words, we can mainly consider the vehicle flows on major roads instead of all roads.

3.2. Problem formulation

The TFMS places RSUs on streets (i.e., edges) to monitor and measure passing vehicles through location proofs. Let S denote an RSU placement strategy, which is our variable. S is the set of edges with placed RSUs. For example, in Fig. 1, we have $S = \{e_1, e_3, e_6\}$. Let $S(f)$ denote the subset of S, the edges in which f goes through. In Fig. 1, we have $S(f_1) = \{e_6\}$, $S(f_2) = S(f_3) = \{e_3\}$, and $S(f_4) = \{e_1\}$. A traffic flow is said to be covered, if it goes through at least one RSU. To monitor all given traffic flows in F, each given traffic flow should be covered, meaning that $S(f) \neq \emptyset$ for $\forall f \in F$. However, the coverage requirement is insufficient to distinguish different traffic flows. A covered traffic flow is said to be distinguishable, if the set of its passing RSUs is unique among all traffic flows. To accurately monitor the traffic flow, a covered traffic flow should be distinguishable, meaning that $S(f) \neq S(f')$ for $\forall f, f' \in F, f \neq f'$.

Since RSUs are expensive, the manufacturing cost of a TFMS depends on the placement of the RSU. To reduce the manufacturing cost, our objective is to minimize the number of placed RSUs, while all given traffic flows must be covered and distinguishable for location proofs. Let $|S|$ denote the set cardinality of S. Our problem is formulated as follows:

$$\begin{aligned} \text{minimize} \quad & |S| \\ \text{subject to} \quad & S(f) \neq \emptyset \text{ for } \forall f \in F \\ & S(f) \neq S(f') \text{ for } \forall f, f' \in F, f \neq f' \end{aligned} \tag{1}$$

4. Problem analysis

This section shows that our problem is NP-hard, monotonic, and non-submodular. The key theorem is also presented.

4.1. Problem hardness

Theorem 1. *The RSU placement problem is NP-hard.*

The proof of Theorem 1 is available in [34]. The key idea is that the coverage and distinguishability requirements can be unified under a special assumption. This proof also indicates that our problem is a non-trivial extension of the traditional set cover problem.

4.2. Monotonicity and non-submodularity

This subsection presents two basic properties of the RSU placement problem: monotonicity and non-submodularity. Let $N(S)$ denote the number of covered and distinguishable traffic flows, under the RSU placement strategy of S. By definition, $0 \leq N(S) \leq n$. Let T denote a superset of S, i.e., $S \subseteq T$. The monotonicity is stated as follows:

Theorem 2. *The RSU placement problem is monotonic, meaning that $N(S) \leq N(T)$ for $\forall S \subseteq T, T \subseteq E$.*

The proof of Theorem 2 is available in [34]. Theorem 2 shows that more RSUs can always cover and distinguish no fewer traffic flows. Since the RSU placement problem is monotonic, it can be

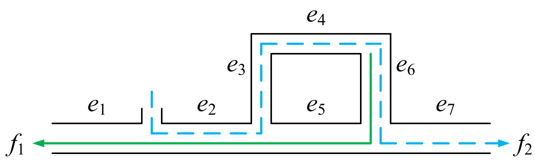


Fig. 2. An example to illustrate Theorems 4 and 8.

solved by greedy algorithms. However, the monotonicity is insufficient to obtain bounded solutions. In general, the submodularity [4] is desired. $N(S)$ is submodular, if it satisfies

$$N(S \cup \{e\}) - N(S) \geq N(T \cup \{e\}) - N(T) \tag{2}$$

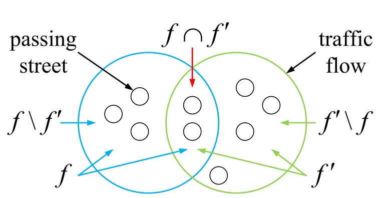
for $\forall e \in E, S \subseteq T, T \subseteq E$. Here, e denotes an arbitrary edge (street to place an RSU). The submodularity means that the marginal gain of $N(S)$ decreases with respect to the size of S. It is also known as the diminishing return property [4]. Unfortunately, the RSU placement problem is proven to be non-submodular in the following theorem:

Theorem 3. *The RSU placement problem is not submodular, meaning that* $N(S \cup \{e\}) - N(S) < N(T \cup \{e\}) - N(T)$ *for* $\exists e \in E, S \subseteq T, T \subseteq E$.

The proof of Theorem 3 is available in [34]. Due to the non-submodularity, simple greedy algorithms are not bounded [2]. Non-submodularity clearly differentiates our RSU placement problem from the traditional set cover problem that is submodular [4]. The coverage and distinguishability requirements pose unique challenges for our problem, which is a non-trivial extension of the traditional set cover problem. Therefore, further observations are needed to obtain approximation results.

4.3. Key observation and trivial bound

This paper minimizes the number of placed RSUs under the coverage and distinguishability requirements. For a traffic flow (say f), $S(f)$ should be non-empty and unique. Note that $S(f)$ is unique, if and only if $S(f) \neq S(f')$ for $\forall f, f' \in F, f \neq f'$. The distinguishability requirement should be analyzed in a *pairwise* manner. The key observation is that two RSUs are necessary and sufficient to cover and distinguish an arbitrary pair of given traffic flows (say f and f'). In the following paper, we slightly abuse the notation, in which f can also denote the set of streets (edges) it goes through. Then, we can divide the set of $f \cup f'$ into three disjoint subsets of $f \setminus f', f' \setminus f$, and $f \cap f'$. These subsets are depicted in the following:



The key observation is formally presented as follows:

Theorem 4. *To cover and distinguish a given pair of traffic flows (f and f'), two RSUs should be placed on streets from two different subsets of* $f \setminus f', f' \setminus f$, *and* $f \cap f'$.

The proof of Theorem 4 can be done by checking all the combinational possibilities. RSUs, which are not placed on streets in $f \cup f'$, will not cover or distinguish f and f'. An example is shown in Fig. 2, where we have:

Algorithm 1 Pair-Based Greedy (PBG)

Input: A graph, G, and a set of traffic flows, F.
Output: A RSU placement strategy, S.

1: Initialize $S = \emptyset$.
2: Initialize F^2 as the set of all pairs of traffic flows.
3: **for** each pair of streets, $e \in E$ and $e' \in E$ **do**
4: Initialize a counter of $C_{ee'} = 0$.
5: **while** $F^2 \neq \emptyset$ **do**
6: **for** each pair of traffic flows, f and f', in F^2 **do**
7: **for** a pair of streets, e and e', in $f \cup f'$ **do**
8: **if** ($e \notin S$ or $e' \notin S$) and (e and e' are in two different subsets of $f \setminus f', f' \setminus f$, and $f \cap f'$) **then**
9: Update $C_{ee'} = C_{ee'} + 1$.
10: Update $S = S \cup \{\arg\max_{ee'} C_{ee'}\}$.
11: Remove f and f' for $\arg\max_{ee'} C_{ee'}$ from F^2.
12: Reset $C_{ee'} = 0$ for each pair of streets, e and e'.
13: **return** S as the RSU placement strategy.

Three disjoint subsets for $f_1 \cup f_2$	$f_1 \setminus f_2$	$f_2 \setminus f_1$	$f_1 \cap f_2$
Corresponding streets (edges)	e_1, e_5	e_3, e_4, e_7	e_2, e_6

To satisfy $S(f_1) \neq \emptyset$, $S(f_2) \neq \emptyset$, and $S(f_1) \neq S(f_2)$, we can have $S = \{e_1, e_3\}$, $S = \{e_2, e_4\}$, or $S = \{e_5, e_6\}$. In contrast, we cannot have $S = \{e_1, e_5\}$, $S = \{e_3, e_4\}$, or $S = \{e_2, e_6\}$. Theorem 4 results in a trivial bound as follows:

Theorem 5. *The minimum number of placed RSUs, which can cover and distinguish all n given traffic flows, should be no smaller than* $\lceil \log_2 n \rceil$ *and no larger than* $n(n-1)$.

The proof of Theorem 5 is available in [34].

5. Algorithmic design

5.1. Pair-based greedy

This subsection presents a bounded greedy algorithm based on Theorem 4. Two RSUs are necessary and sufficient to cover and distinguish an arbitrary pair of given traffic flows. The key idea is to place a pair of RSUs in each greedy iteration. Such pairwise placements convert our problem to be submodular, and thus, have a bounded performance.

Algorithm 1 is proposed to pairwisely place RSUs. In lines 1 and 2, it initializes S as an empty set and F^2 as the set of all traffic flow pairs. A counter is maintained for each pair of streets (lines 3 and 4). Algorithm 1 iteratively updates a pair of RSUs to the current S though a greedy placement (lines 5 to 12). The iteration terminates, when all pairs of given traffic flows are covered and distinguishable ($F^2 \neq \emptyset$ in line 5). In each iteration (lines 6 to 9), Algorithm 1 calculates $C_{ee'}$ for each pair of streets that are *not both* in S (i.e., the streets e or e' may already be in S, but not both of them are in S). $C_{ee'}$ is the number of covered and distinguishable pairs of traffic flows, if two RSUs are placed on the pair of streets e and e'. Based on Theorem 4, f and f' are covered and distinguishable, if e and e' are in two different subsets of $f \setminus f', f' \setminus f$, and $f \cap f'$ (line 8). e and e' may cover and distinguish multiple pairs of traffic flows. The pair of streets, which maximize $C_{ee'}$, are greedily added to S as the RSU placement (line 10). The corresponding pairs of traffic flows are removed from F^2 (line 11). We reset $C_{ee'} = 0$ for the next iteration (line 12). Finally, S is returned when the iteration terminates (line 13).

An example is shown in Fig. 3 to illustrate Algorithm 1. Each traffic flow pair can be covered and distinguished by placing RSUs on the following pairs of streets:

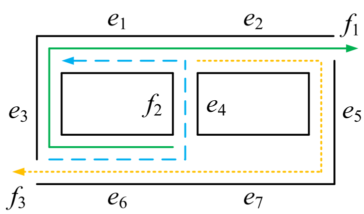


Fig. 3. An example to illustrate Algorithms 1, 2, and 3.

f and f'	Pairs of streets that can cover and distinguish f and f'
f_1 and f_2	$\{e_1, e_2\}$ $\{e_1, e_3\}$ $\{e_1, e_4\}$ $\{e_2, e_4\}$ $\{e_2, e_6\}$ $\{e_3, e_4\}$ $\{e_3, e_6\}$ $\{e_4, e_6\}$
f_1 and f_3	$\{e_1, e_2\}$ $\{e_1, e_5\}$ $\{e_1, e_6\}$ $\{e_1, e_7\}$ $\{e_2, e_3\}$ $\{e_2, e_5\}$ $\{e_2, e_7\}$ $\{e_3, e_5\}$ $\{e_3, e_6\}$ $\{e_3, e_7\}$ $\{e_5, e_6\}$ $\{e_6, e_7\}$
f_2 and f_3	$\{e_1, e_2\}$ $\{e_1, e_5\}$ $\{e_1, e_6\}$ $\{e_1, e_7\}$ $\{e_2, e_4\}$ $\{e_2, e_6\}$ $\{e_4, e_5\}$ $\{e_4, e_6\}$ $\{e_4, e_7\}$ $\{e_5, e_6\}$ $\{e_6, e_7\}$

Algorithm 1 initializes F^2 to include three traffic flow pairs. In the first iteration (lines 5 to 12), we have $\max_{ee'} C_{ee'} = 3$ for e_1 and e_2, since they can cover and distinguish three traffic flow pairs (f_1 and f_2, f_1 and f_3, f_2 and f_3). Hence, e_1 and e_2 are added to S, and the corresponding three traffic flow pairs are removed from F^2. After the first iteration, F^2 becomes empty and the iteration terminates. Algorithm 1 returns $S = \{e_1, e_2\}$, which is the optimal RSU placement for this example. To satisfy the coverage and distinguishability requirements, we have $S(f_1) = \{e_1, e_2\}$, $S(f_2) = \{e_1\}$, and $S(f_3) = \{e_2\}$. For each f, $S(f)$ is non-empty and unique.

The time complexity of Algorithm 1 is $O(n^2|E|^3)$, resulting from $O(|E|)$ iterations. This is because Algorithm 1 adds at least one new street to S in each iteration, while we have at most $|E|$ streets. Then, each iteration takes $O(n^2|E|^2)$ to go through each pair of traffic flows to compute $C_{ee'}$ for each pair of streets. In total, we have $O(n^2)$ pairs of traffic flows and $O(|E|^2)$ pairs of streets. Algorithm 1 has a high time complexity, because it computes $C_{ee'}$ for each pair of streets. We claim that Algorithm 1 is bounded:

Theorem 6. *Algorithm 1 achieves a ratio of $n \ln \frac{n(n-1)}{2}$ to the optimal algorithm for the number of placed RSUs.*

The proof of Theorem 6 is described in the Appendix. $n \ln \frac{n(n-1)}{2}$ belongs to $\Theta(n \ln n)$. The next subsection will present another greedy algorithm, which has a similar bound but a lower time complexity than Algorithm 1.

5.2. Subset-based greedy

This subsection describes another greedy algorithm. Theorem 4 states that, to cover and distinguish f and f', two RSUs are placed on streets from two different subsets of $f \setminus f'$, $f' \setminus f$, and $f \cap f'$. As a relaxation, our idea is to place an RSU on a street from each of $f \setminus f'$, $f' \setminus f$, and $f \cap f'$. In other words, three RSUs are placed for each pair of traffic flows. Such a relaxation converts our problem to be submodular by using redundant placements. Hence, it is bounded.

Algorithm 2 is proposed. After the initialization (line 1), it decomposes each pair of traffic flows into three subsets (lines 2 and 3). These subsets are added to $F^\dagger$. A counter is maintained for each street (lines 4 and 5). Then, Algorithm 2 iteratively updates an RSU to the current S though a greedy placement (lines 6 to 12). The iteration terminates, when all pairs of given traffic flows are covered and distinguishable ($F^\dagger \neq \emptyset$ in line 6). In each iteration, Algorithm 1 calculates C_e for each street (lines 7 to 9). C_e represents the number of included subsets in $F^\dagger$, if an RSU is placed on the street of e. An RSU is placed on a street from each of the three subsets of each traffic flow pair. However, a street, e, may include multiple subsets from different traffic flow pairs. The street, which maximize C_e, is greedily added to S (line 10). The corresponding subsets in $F^\dagger$ are removed (line 11). Algorithm 2 resets $C_e = 0$ for the next iteration (line 12). Finally, S is returned (line 13).

Algorithm 2 Subset-Based Greedy (SBG)

Input: A graph, G, and a set of traffic flows, F.
Output: A RSU placement strategy, S.

1: Initialize $S = \emptyset$ and $F^\dagger = \emptyset$.
2: **for** each pair of traffic flows, f and f' **do**
3: Add three subsets of $f \setminus f'$, $f' \setminus f$, and $f \cap f'$ to $F^\dagger$.
4: **for** each street, $e \in E$ **do**
5: Initialize a counter of $C_e = 0$.
6: **while** $F^\dagger \neq \emptyset$ **do**
7: **for** each subset, $f^\dagger \in F^\dagger$ **do**
8: **for** $e \in f^\dagger$ and $e \in E \setminus S$ **do**
9: Update $C_e = C_e + 1$.
10: Update $S = S \cup \{\arg\max_e C_e\}$.
11: Remove $f^\dagger$ for $\arg\max_e C_e$ from $F^\dagger$.
12: Reset $C_e = 0$ for each street, e.
13: **return** S as the RSU placement strategy.

The same example in Fig. 3 is used to illustrate Algorithm 2. The subsets corresponding to each traffic flow pair are shown as follows (nine subsets for three traffic flow pairs):

f and f'	$f \setminus f'$	$f' \setminus f$	$f \cap f'$
f_1 and f_2	$\{e_2, e_3\}$	$\{e_4\}$	$\{e_1, e_6\}$
f_1 and f_3	$\{e_1, e_3\}$	$\{e_5, e_7\}$	$\{e_2, e_6\}$
f_2 and f_3	$\{e_1, e_4\}$	$\{e_2, e_5, e_7\}$	$\{e_6\}$

These subsets are added to $F^\dagger$ by Algorithm 2 (lines 1 to 3). In the first iteration (lines 6 to 12), we have $\max_e C_e = 3$ for e_6, since e_6 appears in three subsets of $\{e_1, e_6\}$, $\{e_2, e_6\}$, and $\{e_6\}$. Hence, e_6 is added to S, and the corresponding three subsets are removed from $F^\dagger$. In the following iterations, e_3, e_4, and e_5 are added to S according to the same principle. A random street can be selected in a tie. The iteration terminates when $F^\dagger = \emptyset$. Algorithm 2 returns $S = \{e_3, e_4, e_5, e_6\}$, where $S(f_1) = \{e_3, e_6\}$, $S(f_2) = \{e_4, e_6\}$, and $S(f_3) = \{e_5, e_6\}$. The coverage and distinguishability requirements are satisfied, since $S(f)$ is non-empty and unique.

The time complexity of Algorithm 2 is $O(n^2|E|^2)$, since it has $O(|E|)$ iterations, while each iteration takes $O(n^2|E|)$. Each iteration of Algorithm 2 scans each pair of traffic flows to compute C_e. Algorithm 2 has a lower time complexity than Algorithm 1, since it scans streets rather than pairs of streets (computes C_e rather than $C_{ee'}$). The insight is that Algorithm 2 uses redundant placements to reduce the problem complexity. As a trade-off, Algorithm 2 has a bound that is slightly worse than Algorithm 1:

Theorem 7. *Algorithm 2 achieves a ratio of $\frac{n+1}{2} \ln \frac{3n(n-1)}{2}$ to the optimal algorithm for the number of placed RSUs.*

The proof of Theorem 7 is shown in the Appendix.

5.3. Improved subset-based greedy

This subsection improves the ratio of Algorithm 2 through a subtle change. Algorithm 2 is based on Theorem 4, which places two RSUs on streets from two different subsets of $f \setminus f'$, $f' \setminus f$, and $f \cap f'$. Let $f \triangle f' = (f \setminus f') \cup (f' \setminus f)$, we find that Theorem 4 can be rephrased as follows:

Algorithm 3 Improved Subset-Based Greedy (ISBG)

Input: A graph, G, and a set of traffic flows, F.
Output: A RSU placement strategy, S.

1: Same as Algorithm 2, except the subtle change in line 3: Add three subsets of f, f', and $f \triangle f'$ to $F^{\dagger}$.

Theorem 8. *To cover and distinguish an arbitrary pair of traffic flows (f and f'), f, f', and $f \triangle f'$ should all include a street with a placed RSU.*

Proof. Theorem 8 is not obvious, but can be easily proven by checking all the combinational possibilities. We have three cases in total, based on Theorem 4. In the first case, two RSUs are placed on two streets from $f \setminus f'$ and $f' \setminus f$, respectively. Then, Theorem 8 validates, since $f \setminus f' \subseteq f, f' \setminus f \subseteq f'$, and $f \setminus f' \subseteq f \triangle f'$. In the second case, two RSUs are placed on two streets from $f \setminus f'$ and $f \cap f'$, respectively. Theorem 8 also validates, since $f \setminus f' \subseteq f, f \cap f' \subseteq f'$, and $f \setminus f' \subseteq f \triangle f'$. In the third case, two RSUs are placed on two streets from $f' \setminus f$ and $f \cap f'$, respectively. Theorem 8 remains valid, since $f \cap f' \subseteq f, f \setminus f' \subseteq f'$, and $f \setminus f' \subseteq f \triangle f'$. By checking all the possibilities, the proof completes. ■

The insight of Theorem 8 is that f (also f') should include a street with a placed RSU for the coverage requirement, while $f \triangle f'$ should include a street with a placed RSU for the distinguishability requirement. Note that $f = (f \setminus f') \cup (f \cap f'), f' = (f' \setminus f) \cup (f \cap f')$, and $f \triangle f' = (f \setminus f') \cup (f' \setminus f)$. Each of f, f', and $f \triangle f'$ is an union of two different subsets of $f \setminus f', f' \setminus f$, and $f \cap f'$. Therefore, Theorem 8 validates according to the pigeonhole principle. If we go back to the example in Fig. 2, we have the following subsets for Theorem 8:

Subsets	f_1	f_2	$f_1 \triangle f_2$
Streets (edges)	e_1, e_2, e_5, e_6	e_2, e_3, e_4, e_6, e_7	e_1, e_3, e_4, e_5, e_7

To satisfy Theorem 8, we have $S = \{e_1, e_3\}$, $S = \{e_2, e_4\}$, or $S = \{e_5, e_6\}$. In contrast, we cannot have $S = \{e_1, e_5\}$, $S = \{e_3, e_4\}$, or $S = \{e_2, e_6\}$. It can be seen that, the result for Theorem 8 is the same as the result for Theorem 4.

Algorithm 3 is proposed as a simple but subtle variation of Algorithm 2. The only difference is that Algorithm 3 uses $f \setminus f', f' \setminus f$, and $f \cap f'$ rather than $f \setminus f', f' \setminus f$, and $f \cap f'$. The same example in Fig. 3 is used to illustrate Algorithm 3. Algorithm 3 includes six subsets in $F^{\dagger}$ as follows:

Subsets	f_1	f_2	f_3
Streets	e_1, e_2, e_3, e_6	e_1, e_4, e_6	e_2, e_5, e_6, e_7
Subsets	$f_1 \triangle f_2$	$f_1 \triangle f_3$	$f_2 \triangle f_3$
Streets	e_2, e_3, e_4	e_1, e_3, e_5, e_7	e_1, e_2, e_4, e_5, e_7

Algorithm 3 iteratively selects the street that is included in the most subsets. In the first round, we have $C_e = 4$ for e_1 and e_2, which appear in the most subsets. Suppose that the first iteration adds e_1 into S, and then, the corresponding subsets are removed ($f_1, f_2, f_1 \triangle f_3$, and $f_2 \triangle f_3$ are removed). The second iteration adds e_2 into S, since it appears in all remaining subsets of f_3 and $f_1 \triangle f_2$. The iteration terminates, since $F^{\dagger} = \emptyset$. Algorithm 3 returns $S = \{e_1, e_2\}$, which is also the optimal RSU placement strategy for this example. We have $S(f_1) = \{e_1, e_2\}$, $S(f_2) = \{e_1\}$, and $S(f_3) = \{e_2\}$, i.e., $S(f)$ is non-empty and unique.

The time complexities of Algorithms 2 and 3 are the same, i.e., $O(n^2|E|^2)$. This is because their only difference is the definitions for the subsets. Algorithms 2 and 3 have lower time complexities than Algorithm 1, since they scan streets rather than pairs of streets. Algorithm 2 uses redundant placements to reduce the problem complexity. It has a bound that is similar to Algorithm 1. In contrast, Algorithm 3 does not use redundant placements, and thus, has the best approximation ratio:

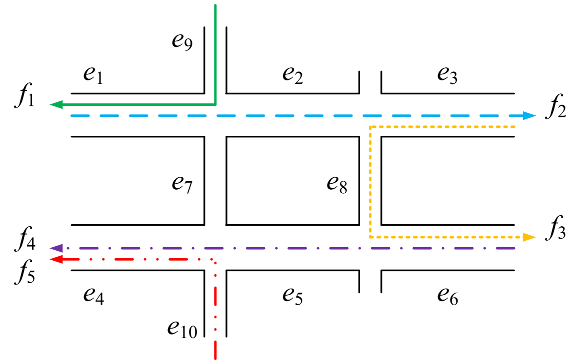


Fig. 4. Credential propagations through V2V communications.

Theorem 9. *Algorithm 3 achieves a ratio of* $\ln \frac{n(n+1)}{2}$ *to the optimal algorithm for the number of placed RSUs.*

The proof of Theorem 9 is shown in the Appendix.

6. Problem extensions with priority levels and vehicle-to-vehicle communications

In practice, traffic flows have different importances, and thus, many TFMSs do not need to monitor all traffic flows in the most secure way. Representative scenarios include location proofs for vehicular trajectories, traffic congestion management, and vehicle theft prevention, where busy or highly sensitive streets always get primary monitoring. As a result, this section extends our problem using priority levels and Vehicle-To-Vehicle (V2V) communications [31].

6.1. Priority levels and V2V Communications

In order to further reduce the number of deployed RSUs while maintaining the coverage and distinguishability for given traffic flows, we define a new concept, *priority level*, with respect to traffic flows. The priority level of a traffic flow indicates its importance. We cut off some RSUs that mainly monitor low priority traffic flows by exploring V2V communications. V2V communications enable the location proofs generated by the RSUs to propagate to other nearby traffic flows. The key idea is that this credential propagation mechanism can increase the coverage of an RSU.

Without loss of generality, let l_i denote the priority level of the traffic flow f_i, where l_i is an integer ranging from 0 to δ. The lower the value of l_i is, the higher priority the traffic flow has. The priority level controls the maximum hops that V2V communications could contribute to the corresponding traffic flow's secure distinguishability. From the consideration of system management costs, the credential's losing probability, and the computing complexity, we generally set δ as a small integer constant, such as 1 or 2.

We add a new dimension, *propagation hop*, to the received RSU tags. For the ease of description, we use e_i^k to represent a k-hop propagated credential from the RSU on edge e_i. For a vehicle that directly passes an RSU on street e_i, it will possess a tag (e_i^0) from the RSU. Whenever a vehicle received a tag e_i^k, the vehicle immediately creates a new message, which contains a new tag e_i^{k+1}, and keeps broadcasting the message to all passing vehicles. We assume that the tag propagation process is secure and probabilistic. The propagation terminates when the hop counter reaches δ. For instance, if an RSU is placed at street e_9 of Fig. 4 and $\delta = 1$, vehicles in flow f_1 will obtain e_9^0 directly from the RSU and e_9^1 from other vehicles in the same flow. Since f_1 and f_2 share a common street (e_1), vehicles in f_2 will also get e_9^1 from the vehicles in f_1 when they pass each other on e_1. However, flows f_3 to f_5 will not have any tags related to e_9 since the maximum propagation hop is $\delta = 1$.

The V2V-based RSU tags can provide secure distinguishability among the given traffic flows. An example is shown in Fig. 4. According to the coverage and distinguishability requirements in the previous section, at least 4 RSUs are needed. One possible RSU placement strategy is $S = \{e_1, e_3, e_4, e_6\}$. However, if V2V communication is allowed with $l_1 = l_3 = l_5 = 0, l_2 = l_4 = 1$, and $\delta = 1$, placing only 3 RSUs is sufficient, where $S' = \{e_8, e_9, e_{10}\}$. Under strategy S', the received tag sets of flows f_1 to f_5 are $\{e_9^0, e_9^1\}$, $\{e_8^1, e_9^1\}$, $\{e_8^0, e_8^1\}$, $\{e_8^1, e_{10}^1\}$, and $\{e_{10}^0, e_{10}^1\}$, respectively. Based on the tags, flows can be securely distinguished from each other through propagated RSU tags.

Our approach further establishes a priority level-based requirement. The coverage and distinguishability among traffic flows with priority level l_i must be provided by the RSU tags within l_i hops. In Fig. 4, the secure distinguishability among flows f_1, f_3, and f_5 can be achieved by purely using the tags directly from the RSUs, i.e., only using the tags $\{e^0\}$. In contrast, the distinguishability of flows f_1 to f_5 can be achieved by using the tags $\{e^0\}$ and $\{e^1\}$. The main idea is that the distinguishability between flows with a higher priority should be provided by more direct, reliable, and credible evidence (i.e. RSU tags).

6.2. Extended problem formulation

Realistically, a credential (i.e. e_i^k) from a nearby RSU may not be always available since the tag could be lost or there were not enough V2V communications among encountered vehicles in different traffic flows. Since V2V communications are probabilistic, let $p(f_i, e^k)$ be the probability that f_i receives k-hop propagated tags from the RSU on street e. If $e \in f_i$, then $p(f_i, e^k) = 1$. Let $\mathbb{P}\{\cdot\}$ denote the probability of an event. Let $T_{ij} \in [0, 1]$ denote a predefined threshold to distinguish any two flows with priority levels l_i and l_j. T_{ij} is symmetric and non-increasing: (i) $T_{ij} = T_{ji}$ and (ii) $T_{ij} \geq T_{ij'}$ for $\forall j < j'$. The distinguishing threshold of a higher priority flow is no less than that of a lower one.

The extended objective is to deploy a minimum number of RSUs such that the probability for securely distinguishing any pair of flows is no less than a predefined threshold, which is in turn determined by the flows' priority levels. The RSU placement problem with the help of V2V communications is formulated as follows:

$$\begin{aligned} &\text{minimize} \quad |S| \\ &\text{subject to} \quad S(f_i) \neq \emptyset \text{ for } \forall f_i \in F \\ &\qquad \mathbb{P}\{S^l(f_i) \neq S^l(f_j)\} \geq T_{ij} \\ &\qquad\qquad \text{for } \forall f_i, f_j \in F \text{ and } l = \max(l_i, l_j) \end{aligned} \tag{3}$$

Here, $S^l(f)$ denotes a set of received tags that vehicles in flow f can obtain within an l-hop V2V credential propagation under a given RSU placement strategy S. When RSUs are placed on $\{e_8, e_9, e_{10}\}$ of Fig. 4, we have $S^0(f_1) = \{e_9^0\}$, $S^0(f_2) = \emptyset$, $S^0(f_3) = \{e_8^0\}$, $S^0(f_4) = \emptyset$, and $S^0(f_5) = \{e_{10}^0\}$. We also have $S^1(f_1) = \{e_9^0, e_9^1\}$, $S^1(f_2) = \{e_8^1, e_9^1\}$, $S^1(f_3) = \{e_8^0, e_8^1\}$, $S^1(f_4) = \{e_8^1, e_{10}^1\}$, and $S^1(f_5) = \{e_{10}^0, e_{10}^1\}$. The priority levels are $l_1 = l_3 = l_5 = 0$ and $l_2 = l_4 = 1$.

The extended problem formulation in Eq. (3) is similar to the original problem formulation in Eq. (1). The coverage requirement remains the same. The distinguishability requirement is extended to use propagated RSU tags within priority levels. Note that the distinguishability requirement is a probabilistic one, since propagations of RSU tags among vehicles in different traffic flows are probabilistic. Clearly, the original problem formulation belongs to a special case of the extended one, in which $T_{ij} = 1$ and $l_i = l_j = 0$ for $\forall f_i, f_j \in F$. Note that the extended problem formulation is practical. For example, most traffic surveillance cameras are placed on main roads or accident-prone sections. When an accident occurs, related information is directly captured by these cameras, while on less-busy road stretches, such information is usually obtained by witness testimony.

6.3. Extended problem analysis

Using V2V communication essentially increases the coverage of an RSU such that traffic flows are more distinguishable. For analysis simplicity, we temporarily ignore the fact that V2V communications are probabilistic, and assume that V2V communications are always successful. We have:

Theorem 10. *If f_i and f_j could be securely distinguished by $S^l(f_i) \neq S^l(f_j)$ with $l = \max(l_i, l_j)$, then this distinguishability is preserved when using RSU tags from more than l hop, i.e., $S^{l'}(f_i) \neq S^{l'}(f_j)$ for $\forall l' > l$.*

Proof. By definition, for $\forall l' > l$, we have $S^l(f_i) \subseteq S^{l'}(f_i)$ and $S^l(f_j) \subseteq S^{l'}(f_j)$. In addition, any tags in $S^l(f_j)$ will not belong to $S^{l'}(f_i) \setminus S^l(f_i)$. This is because $S^l(f_j)$ only includes RSU tags within l hops, while $S^{l'}(f_i) \setminus S^l(f_i)$ only includes RSU tags that are not within l hops. Let e be the RSU tag that leads to $S^l(f_i) \neq S^l(f_j)$. Without loss of generality, we assume that $e \in S^l(f_j)$ and $e \notin S^l(f_i)$. Since any tags in $S^l(f_j)$ will not belong to $S^{l'}(f_i) \setminus S^l(f_i)$, we have $e \notin S^{l'}(f_i) \setminus S^l(f_i)$. Since $e \notin S^l(f_i)$ and $e \notin S^{l'}(f_i) \setminus S^l(f_i)$, we can conclude $e \notin S^{l'}(f_i)$. Since $e \in S^l(f_j) \subseteq S^{l'}(f_j)$, we have $S^{l'}(f_i) \neq S^{l'}(f_j)$ and the proof completes. ■

The idea of Theorem 10 is that, if two traffic flows are distinguishable via l-hop V2V communications, then they remain distinguishable via l'-hop V2V communications for $\forall l' > l$. This is because the RSU tags, which are not within l hops, do not change the distinguishability within l hops. Theorem 10 indicates that the RSU placement can consider traffic flows separately in terms of different priority levels, since RSU tag propagations do not have a negative impact.

6.4. Priority level-based RSU placement

This subsection will solve the extended RSU placement problem. Theorem 10 already indicates that RSUs can be placed separately for traffic flows with different priority levels. Intuitively, one may start with the traffic flows which are in the highest priority level (i.e. $l_i = 0$), and then, gradually include more traffic flows according to the decreasing order of their levels. However, this strategy may perform poorly, since the number of high priority traffic flows is generally much smaller than that of low priority traffic flows in practical. Since high priority traffic flows are few, RSUs placed for high priority traffic flows are also few, and thus, are not likely to provide distinguishability for low priority traffic flows. Consequently, we start with low priority traffic flows and gradually consider high priority traffic flows.

The details are presented in Algorithms 4 and 5. Algorithm 4 is a wrapper function. In line 1 of Algorithm 4, S is initialized to be $\emptyset$. Lines 2 and 3 are greedy iterations for traffic flows from the lowest priority levels (i.e., δ) to the highest priority levels (i.e., 0). In each greedy iteration, Algorithm 4 calls Algorithm 5 (i.e., RPLK) to place RSUs that are able to distinguish all traffic flows of the current priority level. Line 3 adds these RSUs to S. After the greedy iteration l, traffic flows with priority levels from δ to l are distinguishable (but not necessarily covered). After the greedy iterations in lines 2 and 3, all traffic flows are distinguishable. To satisfy the coverage constraint, line 4 iteratively places an RSU on the street that covers the maximum number of uncovered traffic

Algorithm 4 Priority Level-based RSU Placement (PLRP)

Input: A graph, G, and a set of traffic flows, F.
Output: A RSU placement strategy, S.

1: Initialize $S = \emptyset$.
2: **for** each priority level $l = \delta, \ldots, 1, 0$ **do**
3:   Update $S = S \cup \text{RPLK}(G, F, l, S)$.
4: Update S by iteratively placing an RSU on the street that covers a maximum number of uncovered traffic flows, until all traffic flows are covered.
5: **return** S as the RSU placement strategy.

Algorithm 5 RSU Placement within Level (RPL)

Input: A graph, G, a set of traffic flows, F, priority level, l, and RSU placement strategy, S.
Output: A RSU placement strategy, S'.

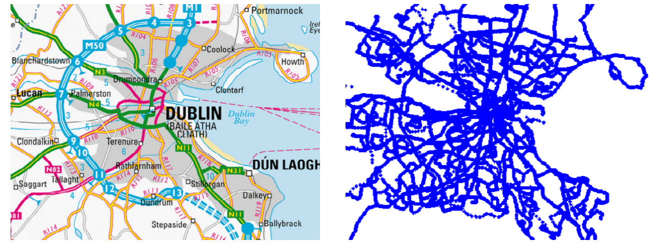
1: Initialize traffic flow set as $F' = \{f_i \mid f_i \in f, l_i \leq l\}$.
2: Initialize distinguish set as $D = \emptyset$.
3: **for** each ordered pair of traffic flows $f_i, f_j \in F'$ **do**
4:   **if** $\mathbb{P}\{S^l(f_i)\backslash S^l(f_j) \neq \emptyset\} < T_{ij}$ under S **then**
5:     Initialize $d_{ij} = \emptyset$.
6:     **for** each $e^k \in S^l_*(f_i)\backslash S^l_*(f_j)$ **do**
7:       $w = 1 - \mathbb{P}\{S^l(f_i)\backslash S^l(f_j) = \emptyset\} \cdot (1 - p(f_i, e^k))$.
8:       Update $d_{ij} = d_{ij} \cup \{(e^k, w)\}$.
9:     Update $D = D \cup \{d_{ij}\}$.
10: Pruning D by removing all supersets within D: remove d'_{ij} for any $d_{ij}, d'_{ij} \in D$ and $d_{ij} \subset d'_{ij}$. Initialize $S' = \emptyset$.
11: **for** each street, $e \in E\backslash S$ **do**
12:   Initialize a counter of $C_e = 0$.
13: **while** $D \neq \emptyset$ **do**
14:   **for** each subset, $d_{ij} \in D$ **do**
15:     **for** each $e \in E\backslash S$ and $e \notin S'$ **do**
16:       **if** $\exists (e^k, w) \in d_{ij}$ **then**
17:         Update $C_e = C_e + w$.
18:   Update $S' = S' \cup \{\arg\max_e C_e\}$.
19:   Update D for $\arg\max_e C_e$.
20:   Reset $C_e = 0$ for each street, e.
21: **return** S' as the RSU placement strategy.

flows, until all traffic flows are covered. When both coverage and distinguishability constraints are satisfied, Algorithm 4 returns S as the RSU placement strategy in line 5.

Algorithm 5 places RSUs by using a V2V-based credential propagation. It involves a new notation, $S^l_*(f_i)$. Here, $S^l_*(f_i)$ denotes a set of RSU tags, which could be received by the vehicles in the traffic flow f_i through at most l-hop V2V communications, if RSUs were placed on all edges (i.e., $S = E$). $S^l_*(f_i)$ indicates the maximum possible set of RSU tags received by f_i. Let us take f_3 in Fig. 4 as an example. We have $S^0_*(f_3) = \{e^0_3, e^0_6, e^0_8\}$. This is because the vehicles in f_3 can only receive tags directly from the RSUs on edges e_3, e_6, and e_8 through 0-hop V2V communications. Similarly, we have $S^1_*(f_3) = \{e^0_3, e^0_6, e^0_8, e^1_1, e^1_2, e^1_3, e^1_6, e^1_8\}$, since a vehicle in f_3 is able to receive additional tags from other vehicles in f_2 and f_3 through 1-hop V2V communications.

Algorithm 5 only solves the distinguishability constraint. It first constructs a set of traffic flows within priority level l in line 1, i.e., $F' = \{f_i \mid f_i \in f, l_i \leq l\}$. The distinguish set, D, is initialized in line 2, and is computed from lines 3 to 9. D is a collection of subsets of streets. The subset d_{ij} includes streets, in which an RSU deployment can distinguish f_i from f_j, but may not distinguish f_j from f_i. Line 3 is a loop through each ordered pair of traffic flows $f_i, f_j \in F'$. If the probability to distinguish f_i from f_j under the current RSU placement strategy S does not meet the threshold, d_{ij} must be computed (line 4). Line 5 initializes $d_{ij} = \emptyset$ and lines 6 to 8 computes d_{ij} through each $e^k \in S^l_*(f_i) \setminus S^l_*(f_j)$. Note that $S^l_*(f_i) \setminus S^l_*(f_j)$ is the set of all streets that can distinguish f_i from f_j. Line 7 computes the corresponding probability if an RSU is placed. Line 8 records the probability in d_{ij}, and line 9 records d_{ij} in D. Line 10 is a pruning procedure to remove redundancies in D. Line 10 also initializes $S' = \emptyset$. From line 11 to line 20, Algorithm 5 greedily selects the street with the largest weight, in terms of distinguishability recorded in D. In lines 11 and 12, Algorithm 5 initializes a counter for each street $e \in E \setminus S$. Lines 13 to 20 include a loop to iteratively select streets for the RSU placement. This loop terminates until D becomes an empty set. In lines 14 to 17, the total weights of a street e in D are aggregated and recorded in its counter C_e. Line 18 greedily places an RSU to the street with maximum weight. Line 19 updates D after the RSU placement, and line 20 resets the counter for the next iteration. Finally, S' is returned in line 21.

The time complexity of Algorithm 5 is $O(\delta n^2|E|^3)$. It takes $O(\delta n^2|E|)$ to compute the distinguish set, since we need to go through each ordered pair of traffic flows with each edge and each priority level in lines 1 to 9. Meanwhile, the cardinality of the distinguish set is also $O(\delta n^2|E|)$. The greedy iterations in lines 11 to 20 take $O(\delta n^2|E|^3)$. This is because it has at most $O(|E|)$ iterations, each of which goes through each street for each d_{ij} in the distinguish set.



(a) The Dublin map. (b) The vehicle trace.

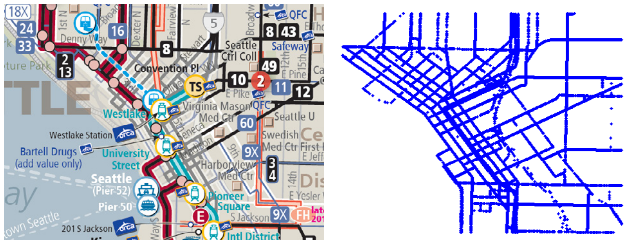
Fig. 5. The map and vehicle trace for Dublin's central area.

7. Experiments

7.1. Real trace-driven datasets

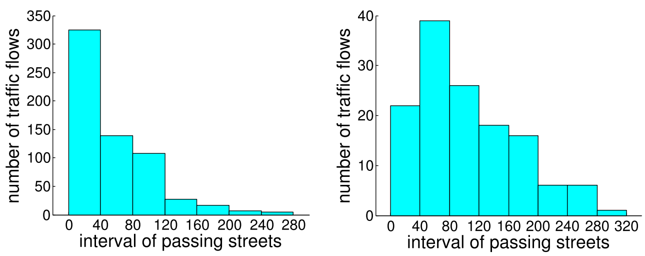
This section conducts experiments based on two real traces, the Dublin vehicle trace [35] and the Seattle bus trace [14]. For the Dublin vehicle trace, we focus on the part within Dublin's central area, which is an 80,000 × 80,000 square foot area, as shown in Fig. 5. The Dublin vehicle trace includes longitude, latitude, and vehicle journey ID. The vehicle journey is a given run on a journey pattern, which corresponds to our concept of the given traffic flow. The Dublin vehicle trace includes 628 given traffic flows on 3657 streets. For the Seattle bus trace, we also focus on the part within Seattle's central area, which is a 10,000 × 10,000 square foot area, as shown in Fig. 6. The Seattle bus trace includes the x-coordinate, y-coordinate, and bus route ID. Each bus route is a given traffic flow. The Seattle bus trace includes 135 given traffic flows on 2283 streets.

The distributions of the Dublin vehicle trace and the Seattle bus trace are analyzed. Fig. 7 shows the distribution of the number of passing streets for a traffic flow. In both traces, a traffic flow can go through as many as about 300 streets. In the Dublin vehicle trace, most traffic flows go through less than 40 streets. In contrast, in the Seattle bus trace, most traffic flows go through 40 to 80 streets. Traffic flows in the Seattle bus trace, on average, go through more



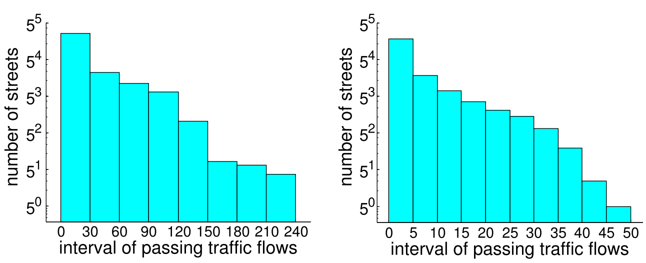
(a) The Seattle map. (b) The bus trace.

Fig. 6. The map and bus trace for Seattle's central area.



(a) Dublin vehicle trace. (b) Seattle bus trace.

Fig. 7. The distribution of the number of passing streets for a traffic flow.



(a) Dublin vehicle trace. (b) Seattle bus trace.

Fig. 8. The distribution of the number of passing traffic flows for a street.

streets than those in the Dublin vehicle trace. On the other hand, Fig. 8 shows the distribution of the number of passing traffic flows for a street. A street in the Dublin vehicle trace can have up to 240 passing traffic flows, while a street in the Seattle bus trace has no more than 50 passing traffic flows. In other words, traffic flows are more dense on a street in the Dublin vehicle trace.

Practical TFMS applications in the Dublin vehicle trace include the traffic congestion solution by managing the traffic flows captured by the TFMS. Since our RSU placement can cover and distinguish all given traffic flows, the rate of each traffic flow can be collected by the TFMS for vehicle redirections. Practical TFMS applications in the Seattle bus trace can include the dynamic bus arrival time estimation through TFMS's trajectory predictions, under the assumption of a fixed bus speed. They are applicable in smart cities.

7.2. Evaluations for original problem

This subsection describes experiments for the original problem in Eq. (1).

7.2.1. Experimental settings

Our experiments mainly focus on the relationship between the number of placed RSUs and the percentage of traffic flows, under nine different scenarios that are defined by three different flow locations and three different flow lengths. Streets are classified into downtown and suburb, depending on the number of passing traffic flows. If a traffic flow goes through more downtown streets than suburb streets, then it is in downtown. Otherwise, it is in suburb. We have three different flow locations of downtown, suburb, and both of them (i.e., all locations). After determining the flow location, we filter traffic flows by their lengths. The length of a traffic flow is defined as the number of its passing streets. We have three different flow lengths of the top half, bottom half, and both of them (i.e., all lengths). Once the scenario is decided, a given percentage of traffic flows are uniform-randomly selected for the RSU placement. The results are averaged over 1000 times for smoothness.

7.2.2. Comparison algorithms

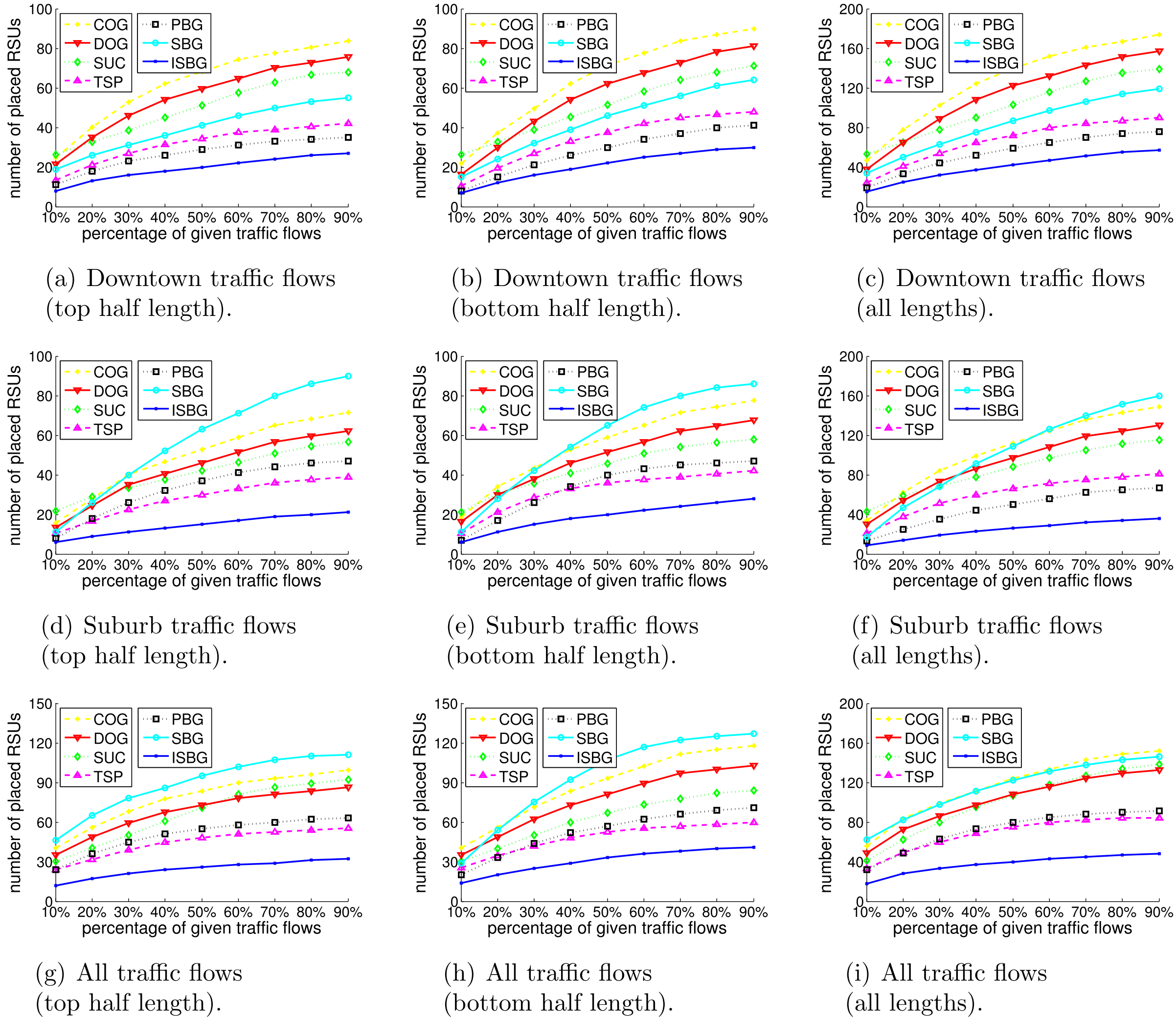
Algorithms 1 to 3 are evaluated in the experiments. They are denoted as PBG, SBG, and ISBG, respectively. In addition to the proposed algorithms, four baseline algorithms are used according to different ideas:

- Coverage-Oriented Greedy (COG). It just iteratively places an RSU on the street that covers maximum uncovered traffic flows. The iteration terminates when both the coverage and distinguishability are satisfied.
- Distinguishability-Oriented Greedy (DOG). For each pair of f and f', it iteratively places an RSU on the street that covers the maximum number of subsets created by $f \triangle f'$. The iteration terminates when both the coverage and distinguishability are satisfied.
- Select Unique Coverage (SUC). It iteratively places an RSU on a street that uniquely covers a traffic flow. If such a street is not found, it performs an exhaustive search to optimally place remaining RSUs.
- Two Stage Placement (TSP). It has two stages. In the first stage, it greedily places RSUs to cover all traffic flows. In the second stage, it greedily places RSUs to distinguish all traffic flows.

7.2.3. Evaluation results in Dublin vehicle trace

The evaluation results of the Dublin vehicle trace are shown in Fig. 9, which has three rows and three columns of subfigures. Rows are scenarios with different flow locations of downtown (first row), suburb (second row), and all locations (third row). Columns are scenarios with different flow lengths of top half (first column), bottom half (second column), and all lengths (third column). Experiments focus on the performances with respect to different percentages of randomly-selected traffic flows in nine scenarios. A smaller number of placed RSUs means a better performance. Note that, because of the monotonicity, more RSUs are needed to cover and distinguish more traffic flows.

Fig. 9 shows that, in all scenarios, a larger percentage of given traffic flows always brings a larger number of placed RSUs. ISBG significantly outperforms all the others among all nine scenarios. This is because ISBG avoids redundant RSU placements, based on Theorem 8. TSP and PBG have the second-best performances. TSP fails to jointly consider the coverage and distinguishability requirements. PBG has redundant RSUs due to its pairwise placement. PBG is better and worse than TSP for downtown and suburb traffic flows, respectively. This is because PBG has redundant RSUs when traffic flows are densely overlapped on streets (i.e., downtown traffic flows). COG, DOG, and SUC do not have good performances, since (i) COG ignores the distinguishability requirement, (ii) DOG ignores the coverage requirement, and (iii) SUC does not utilize traffic flow overlaps to minimize the number of placed RSUs. SBG also performs poorly, especially for suburb traffic flows. This is because it may place more redundant RSUs for each traffic flow



(a) Downtown traffic flows (top half length).
(b) Downtown traffic flows (bottom half length).
(c) Downtown traffic flows (all lengths).
(d) Suburb traffic flows (top half length).
(e) Suburb traffic flows (bottom half length).
(f) Suburb traffic flows (all lengths).
(g) All traffic flows (top half length).
(h) All traffic flows (bottom half length).
(i) All traffic flows (all lengths).

Fig. 9. Results in the Dublin vehicle trace (nine different scenarios defined by three different flow locations and three different flow lengths).

pair. Another notable point is that different flow locations and different flow lengths have some impacts on the number of placed RSUs. For ISBG, slightly more RSUs should be placed for downtown short-length traffic flows in Fig. 9(b) than suburb long-length traffic flows in Fig. 9(d). COG and DOG have the worst performances for downtown traffic flows in Fig. 9(a) and (b), while SBG has the worst performance for suburb traffic flows in Fig. 9(d) and (e). This is because SBG has many redundant placements that are unnecessary for sparse traffic flows in suburb.

7.2.4. Evaluation results in Seattle bus trace

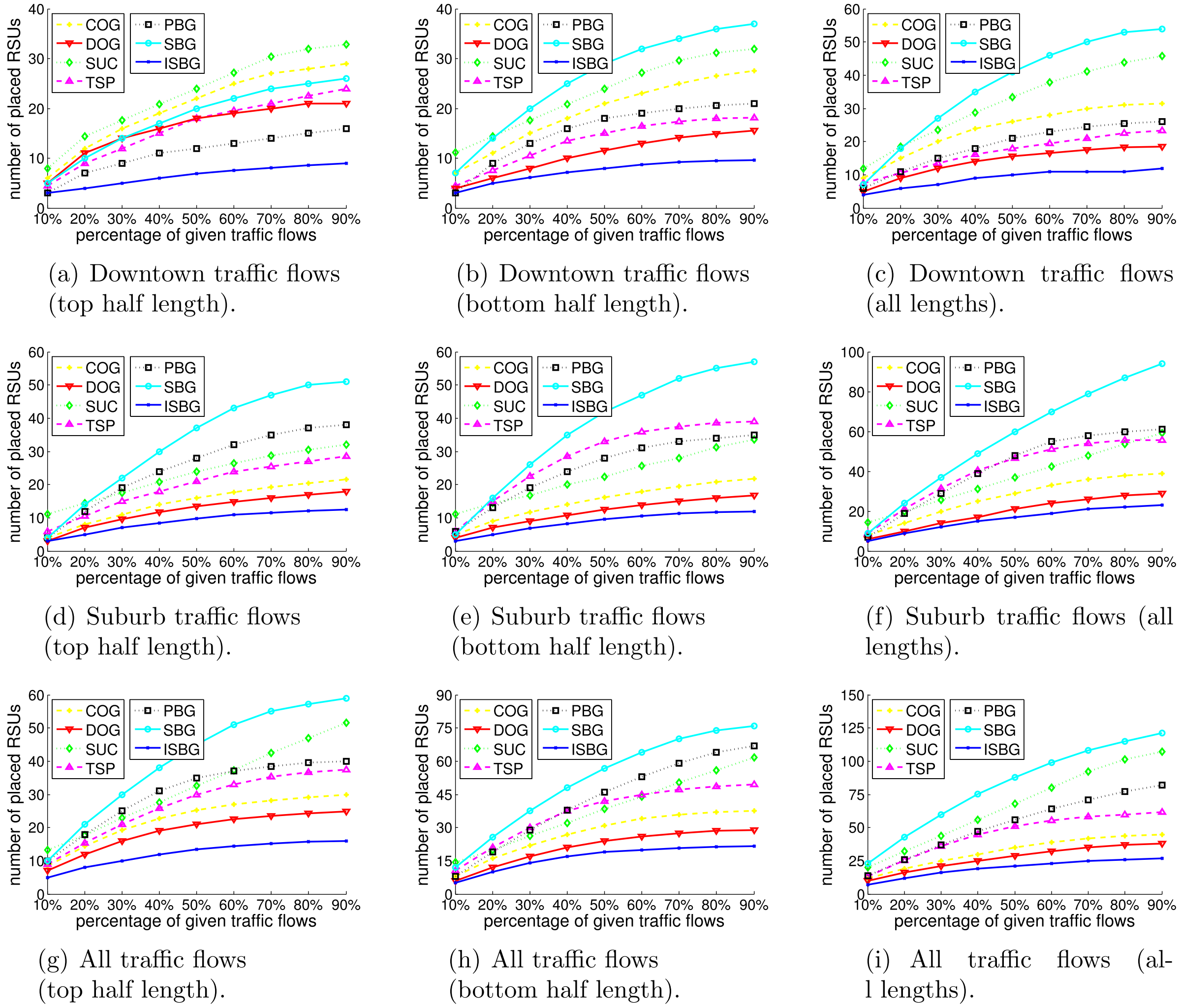
The evaluation results of the Seattle bus trace are shown in Fig. 10, which has the same settings as Fig. 9. The Seattle bus trace has fewer and sparser traffic flows than the Dublin bus trace. While ISBG keeps to have the best performance, SBG has the worst performance, except for SUC in Fig. 10(a). Such a performance gap results from Theorem 8, which can avoid redundant RSU placements. Note that COG and DOG outperform TSP and PBG, since the traffic flows in the Seattle bus trace have longer lengths. This differs from the result in the Dublin bus trace. We also find that more RSUs should be placed for downtown short-length traffic flows in Fig. 10(b) than suburb long-length traffic flows in Fig. 10(d). Areas with denser traffic flows need more RSUs to satisfy the coverage and distinguishability.

7.2.5. Evaluation results for RSU distribution

To further understand the behavior of the RSU placement, this subsection studies the location of placed RSUs in the Dublin trace. The intersections are evenly divided into high-traffic intersections and low-traffic intersections, according to the number of passing vehicles. For each algorithm, we compute the percentage of high-traffic intersections and low-traffic intersections for placed RSUs. The evaluation results are shown in Table 2. Note that COG places 83% RSUs on high-traffic intersections, since it iteratively places an RSU on the street that covers maximum uncovered traffic flows. Although high-traffic intersections can cover traffic flows, they cannot distinguish these traffic flows. Therefore, COG has a bad performance. TSP has the same problem as COG for the same reason. In contrast, DOG and SUC prefers low-traffic intersections to distinguish traffic flows. Meanwhile, our algorithms (especially for ISBG) significantly favor low-traffic intersections during the RSU placement, since our algorithms prioritize distinguishability. Once flows are distinguishable, they are likely to be covered. As a result, we can conclude that distinguishability favors to place RSUs on low-traffic intersections.

7.3. Evaluations for extended problem

This subsection describes experiments for the extended problem in Eq. (3).



(a) Downtown traffic flows (top half length).
(b) Downtown traffic flows (bottom half length).
(c) Downtown traffic flows (all lengths).
(d) Suburb traffic flows (top half length).
(e) Suburb traffic flows (bottom half length).
(f) Suburb traffic flows (all lengths).
(g) All traffic flows (top half length).
(h) All traffic flows (bottom half length).
(i) All traffic flows (al-l lengths).

Fig. 10. Results in the Seattle bus trace (nine different scenarios defined by three different flow locations and three different flow lengths).

Table 2
RSU distribution in the Dublin trace.

Algorithm	High-traffic intersection	Low-traffic intersection
COG	83%	17%
DOG	68%	32%
SUC	61%	39%
TSP	74%	26%
PBG	73%	27%
SBG	57%	43%
ISBG	49%	51%

7.3.1. Experimental settings and comparison algorithms

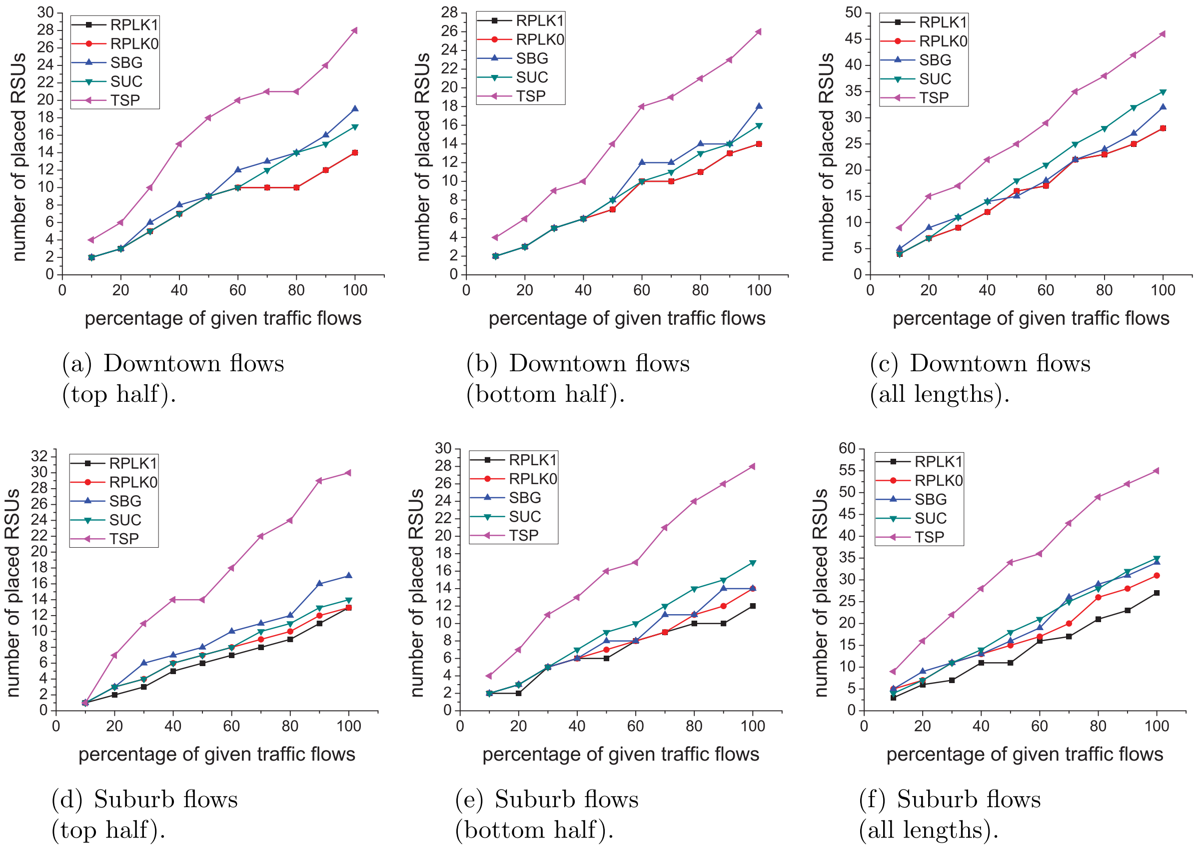
Our experiments continue to study the relationship between the number of deployed RSUs and the percentage of traffic flows, under different scenarios defined in the previous subsection. In addition, the probability for successful V2V communications between vehicles in the same street is set to be 0.5. Note that vehicles in the same street can come from different traffic flows. The predefined threshold, T_{ij}, is uniform-randomly selected from 0 to 1. The priority level of each traffic flow is set to be 1.

Algorithms 2 (SBG) and 4 (PLRP) are evaluated in our experiments. PLRP are further divided into two kinds, RPLK0 and RPLK1, based on the flow priority levels. RPLK0 and RPLK1 assume that the priority level of each traffic flow is 0 or 1, respectively. RPLK1 is expected to use fewer RSUs than RPLK0, since RPLK1 is less restricted by the priority level. SBG is used as a baseline rather than ISBG, since PLRP is essentially a probabilistic variation of SBG. SUC and TSP in the previous subsection are also used for comparisons with some modifications: early termination is used if the probabilistic distinguishability requirement is satisfied.

7.3.2. Evaluation results

The evaluation results of the Dublin vehicle trace and the Seattle bus trace are shown in Figs. 11 and 12, respectively. Similar to the previous subsection, the results are composed of two rows (different flow locations of the downtown and suburb) and three columns (different flow lengths of the top half, bottom half, and all lengths) of subfigures. A smaller number of placed RSUs means a better performance.

We have several interesting observations. The curves in Figs. 11 and 12 are no longer as smooth as Figs. 9 and 10. This is because V2V communications are probabilistic, leading to more fluctuations on the number of RSUs. RPLK1 uses fewer RSUs than RPLK0, since RPLK has less restrictions on the priority levels of traffic flows. By leveraging the priority levels of traffic flows, RPLK1 uses the smallest number of RSUs to cover and distinguish all traffic flows. We find that the number of RSUs needed in the extended problem is less than that in the original problem, since V2V communications can essentially enlarge the propagation of RSU tags. Flow locations and flow lengths have very limited impacts on the number of RSUs for the extended problem. This is also because the credential propagation mechanism increases the coverage of an RSU, such



(a) Downtown flows (top half).

(b) Downtown flows (bottom half).

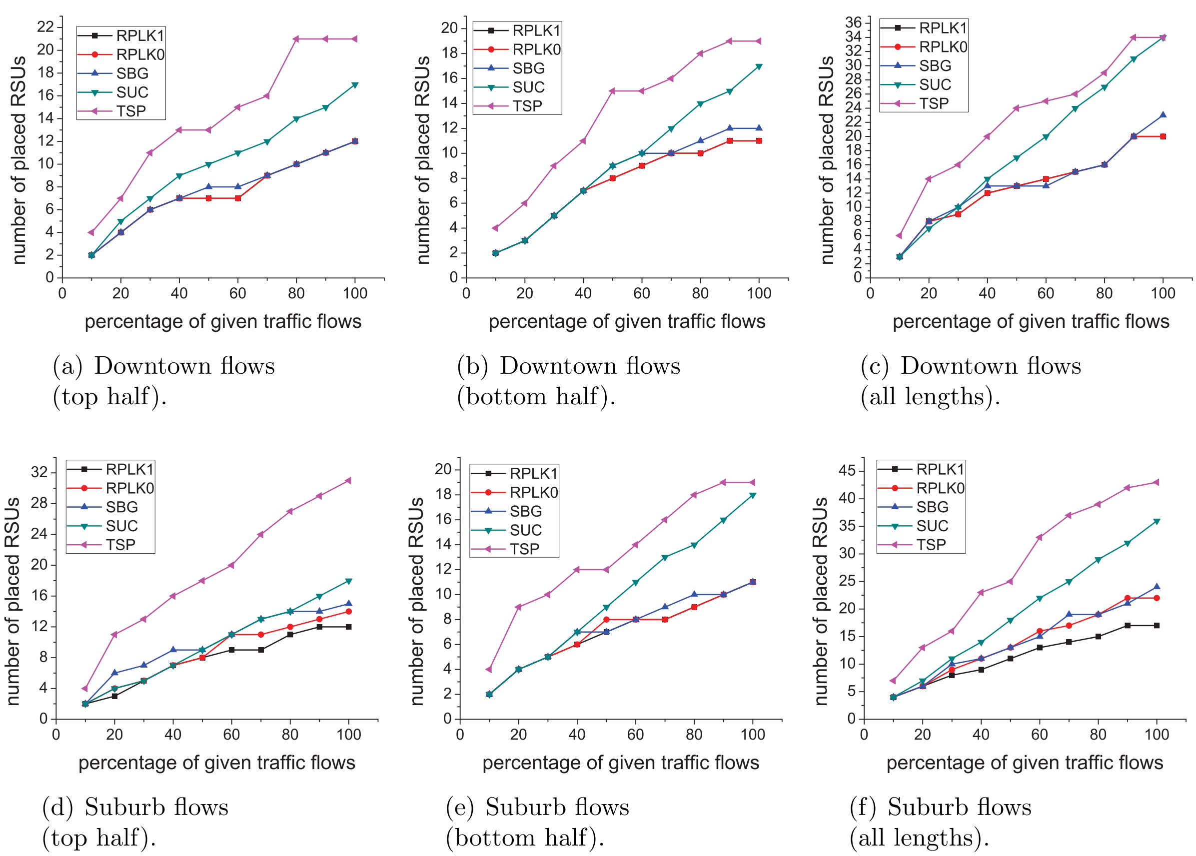
(c) Downtown flows (all lengths).

(d) Suburb flows (top half).

(e) Suburb flows (bottom half).

(f) Suburb flows (all lengths).

Fig. 11. Results in the Dublin vehicle trace for the extended problem.



(a) Downtown flows (top half).

(b) Downtown flows (bottom half).

(c) Downtown flows (all lengths).

(d) Suburb flows (top half).

(e) Suburb flows (bottom half).

(f) Suburb flows (all lengths).

Fig. 12. Results in the Seattle bus trace for the extended problem.

that the differences among flow locations and flow lengths are relatively ignorable. Finally, the number of deployed RSUs in the Seattle bus trace is smaller than that in the Dublin vehicle trace, since the Seattle bus trace is smaller and denser.

8. Conclusion

This paper studies an RSU placement problem for the TFMS. Given some traffic flows on streets, the objective is to place a minimum number of RSUs to cover and distinguish all traffic flows. The coverage and distinguishability requirements are that, for each traffic flow, the set of its passing RSUs should be non-empty and unique. Our problem is NP-hard, monotonic, and non-submodular. Three approximation algorithms are proposed to place RSUs with different insights. Extensive real data-driven experiments demonstrate the efficiency and effectiveness of the proposed algorithms. Our future work will further analyze spatio-temporal nature of vehicle flows (since this paper uses the averaged one over time). The role of time will be identified with respect to the coverage and distinguishability. The state-space graph with more parameters (such as time, location, weather) can be used to model vehicle traffic flows.

Acknowledgments

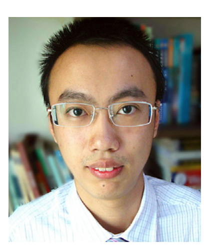
This research was supported in part by NSF grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, and CNS 1439672.

Appendix A. Supplementary data

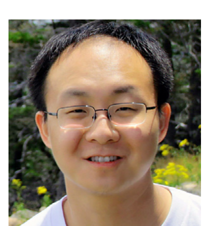
Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jpdc.2018.07.008.

References

[1] O. Abumansoor, A. Boukerche, A secure cooperative approach for nonline-of-sight location verification in VANET, IEEE Trans. Veh. Technol. 61 (1) (2012) 275–285.
[2] M. Cardei, M.T. Thai, Y. Li, W. Wu, Energy-efficient target coverage in wireless sensor networks, in: IEEE INFOCOM, 2005, pp. 1976–1984.
[3] A. Deshpande, L. Hellerstein, D. Kletenik, Approximation algorithms for stochastic boolean function evaluation and stochastic submodular set cover, in: ACM-SIAM SODA, 2014, pp. 1453–1467.
[4] A. Deshpande, L. Hellerstein, D. Kletenik, Approximation algorithms for stochastic submodular set cover with applications to boolean function evaluation and min-knapsack, ACM Trans. Algorithms 12 (3) (2016) 42.
[5] S. Dughmi, Algorithmic information structure design: a survey, ACM SIGecom Exch. 15 (2) (2017) 2–24.
[6] M. Feldman, R. Izsak, Constrained monotone function maximization and the supermodular degree, in: ACM-SIAM SODA, 2014, pp. 1–23.
[7] S. Fujishige, S. Isotani, A submodular function minimization algorithm based on the minimum-norm base, Pac. J. Optim. 7 (1) (2011) 3–17.
[8] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, S. Kumar, Autowitness: locating and tracking stolen property while tolerating gps and radio outages, ACM Trans. Sens. Netw. 8 (4) (2012) 31.
[9] T. He, N. Bartolini, H. Khamfroush, I. Kim, L. Ma, T. La Porta, Service placement for detecting and localizing failures using end-to-end observations, in: IEEE ICDCS, 2016, pp. 560–569.
[10] T. Higuchi, P. Martin, S. Chakraborty, M. Srivastava, AnonyCast: privacy-preserving location distribution for anonymous crowd tracking systems, in: ACM UbiComp, 2015, pp. 1119–1130.
[11] R.M. Ishtiaq Roufa, H. Mustafaa, S.O. Travis Taylora, W. Xua, M. Gruteserb, W. Trappeb, I. Seskarb, Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study, in: USENIX Security, 2010, pp. 11–13.
[12] A. Janecek, K.A. Hummel, D. Valerio, F. Ricciato, H. Hlavacs, Cellular data meet vehicular traffic theory: location area updates and cell transitions for travel time estimation, in: ACM UbiComp, 2012, pp. 361–370.
[13] C. Javali, G. Revadigar, K.B. Rasmussen, W. Hu, S. Jha, I am alice, i was in wonderland: Secure location proof generation and verification protocol, in: IEEE LCN, 2016, pp. 477–485.
[14] J. Jetcheva, Y. Hu, S. PalChaudhuri, A. Saha, D. Johnson, Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture, in: WMCSA, 2003, pp. 32–43.
[15] Y. Jin, W.-S. Soh, M. Motani, W.-C. Wong, A robust indoor pedestrian tracking system with sparse infrastructure support, IEEE Trans. Mob. Comput. 12 (7) (2013) 1392–1403.
[16] R. Khan, S. Zawoad, M.M. Haque, R. Hasan, OTIT: Towards secure provenance modeling for location proofs, in: ACM ASIACCS, 2014, pp. 87–98.
[17] S. Lee, G. Tewolde, J. Kwon, Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application, in: IEEE WF-IoT, 2014, pp. 353–358.
[18] K. Maurya, M. Singh, N. Jain, Real time vehicle tracking system using GSM and GPS technology-an anti-theft tracking system, IJSE (2012) 1956–2277.
[19] E. Necula, Analyzing traffic patterns on street segments based on GPS data using R, Trans. Res. Procedia 10 (2015) 276–285.
[20] L.P. Perera, P. Oliveira, C. Guedes Soares, Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction, IEEE Trans. Intell. Transp. Syst. 13 (3) (2012) 1188–1200.
[21] A. Pham, K. Huguenin, I. Bilogrevic, I. Dacosta, J.-P. Hubaux, Securerun: Cheat-proof and private summaries for location-based activities, IEEE Trans. Mob. Comput. 15 (8) (2016) 2109–2123.
[22] A. Reis, S. Sargento, F. Neves, O. Tonguz, Deploying roadside units in sparse vehicular networks: what really works and what does not, IEEE Trans. Veh. Technol. 63 (6) (2014) 2794–2806.
[23] R. Sen, A. Maurya, B. Raman, R. Mehta, R. Kalyanaraman, N. Vankadhara, S. Roy, P. Sharma, Kyun queue: a sensor network system to monitor road traffic queues, in: ACM SenSys, 2012, pp. 127–140.
[24] S. Sivaraman, M.M. Trivedi, Integrated lane and vehicle detection, localization, and tracking: A synergistic approach, IEEE Trans. Intell. Transp. Syst. 14 (2) (2013) 906–917.
[25] S. Sivaraman, M.M. Trivedi, Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis, IEEE Trans. Intell. Transp. Syst. 14 (4) (2013) 1773–1795.
[26] M. Sviridenko, J. Vondrák, J. Ward, Optimal approximation for submodular and supermodular optimization with bounded curvature, in: ACM-SIAM SODA, 2015, pp. 1134–1148.
[27] A. Thiagarajan, J. Biagioni, T. Gerlich, J. Eriksson, Cooperative transit tracking using smart-phones, in: ACM SenSys, 2010, pp. 85–98.
[28] V. Tzoumas, M.A. Rahimian, G.J. Pappas, A. Jadbabaie, Minimal actuator placement with bounds on control effort, IEEE Trans. Control Netw. Syst. 3 (1) (2016) 67–78.
[29] X. Wang, A. Pande, J. Zhu, P. Mohapatra, STAMP: enabling privacy-preserving location proofs for mobile users, IEEE/ACM Trans. Netw. 24 (6) (2016) 3276–3289.
[30] L. Xu, C. Huang, P. Li, J. Zhu, A randomized algorithm for roadside units placement in vehicular ad hoc network, in: IEEE MSN, 2013, pp. 193–197.
[31] Y. Yao, X. Chen, L. Rao, X. Liu, X. Zhou, LORA: Loss differentiation rate adaptation scheme for vehicle-to-vehicle safety communications, IEEE Trans. Veh. Technol. 66 (3) (2017) 2499–2512.
[32] Y. Zhang, C.C. Tan, F. Xu, H. Han, Q. Li, VProof: Lightweight privacy-preserving vehicle location proofs, IEEE Trans. Veh. Technol. 64 (1) (2015) 378–385.
[33] M. Zhao, T. Ye, R. Gao, F. Ye, Y. Wang, G. Luo, VeTrack: Real time vehicle tracking in uninstrumented indoor environments, in: ACM SenSys, 2015, pp. 99–112.
[34] H. Zheng, W. Chang, J. Wu, Coverage and distinguishability requirements for traffic flow monitoring systems, in: IEEE/ACM IWQoS, 2016, pp. 1–10.
[35] H. Zheng, J. Wu, Optimizing roadside advertisement dissemination in vehicular cyber-physical systems, in: IEEE ICDCS, 2015, pp. 41–50.
[36] Z. Zhu, G. Cao, Toward privacy preserving and collusion resistance in a location proof updating system, IEEE Trans. Mob. Comput. 12 (1) (2013) 51–64.
[37] C. Zhu, C. Zheng, L. Shu, G. Han, A survey on coverage and connectivity issues in wireless sensor networks, J. Netw. Comput. Appl. 35 (2) (2012) 619–632.



Huanyang Zheng received his B.Eng. degree in Telecommunication Engineering from Beijing University of Posts and Telecommunications, China, in 2012. He is currently a Ph.D. candidate in the Department of Computer and Information Sciences, Temple University, USA. His research focuses on wireless and mobile networks, social networks and structures, and cloud systems.



Wei Chang is an Assistant Professor at Computer Science Department, Saint Joseph's University, Pennsylvania, USA. He received Ph.D. in Computer Science from Temple University in 2016, and B.S. from Beijing University of Posts and Telecommunications in 2009. His research interests include social information-assisted system design, and its related security and privacy issues. He serves as the coordinator of Cybersecurity (CSEC) Program at Department of Computer Science, Saint Joseph's University.

Jie Wu is the Associate Vice Provost for International Affairs at Temple University. He also serves as the Chair and Laura H. Carnell professor in the Department of Computer and Information Sciences. Prior to joining Tempe University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. He was general co-chair/chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). He is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.