Access Points Planning in Urban Area for Data Dissemination to Drivers

Tan Yan, Wensheng Zhang, Guiling Wang, and Yujun Zhang

***Abstract*—Roadside infrastructure can greatly help disseminate data to drivers. In this paper, we study a fundamental problem, i.e., roadside infrastructure planning. We propose a class of algorithms named *Tailor* to select a minimum number of intersections to install the infrastructure. In the case when the traffic information is not available, we formulate the intersection selection problem, which formally proves its np-completeness, and provide novel heuristics, i.e., the adapted-bipartite-based heuristics (ABS), to solve it, whose worst-case approximation ratio is 4/3. ABS bridges the planar graph and the bipartite graph through topology transformation. With ABS, the approximate solution to all the problems that are NP-hard in a general planar graph but polynomially solvable in a bipartite graph can be efficiently obtained in the planar graph. We also prove that, even with traffic information, the intersection selection problem remains NP-hard. Greedy heuristics is employed to balance the tradeoff between the number of selected intersections and the percentage of reached vehicles.**

***Index Terms*—Graph theory, heuristics, NP-complete, vehicular ad hoc network (VANET).**

I. Introduction

THE UNITED STATES is a nation on wheels. Driving plays an important role in people's daily life. Providing drivers with timely and helpful information can greatly improve road safety and reduce congestion. On the other hand, drivers are an important group of customers. To reach them effectively through advertisement is the very desire of a large number of companies. Therefore, effective and efficient schemes to disseminate data to drivers are in great demand for both safety and commercial applications.

Current research in data dissemination to drivers is generally conducted in vehicular ad hoc networks (VANETs) through epidemic propagation with or without the help of roadside infrastructure [1]–[3]. Epidemic propagation without roadside infrastructure is generally of high overhead and can only reach drivers with the best effort, considering the sporadic disconnection between vehicles. In many situations, without infrastructure, reaching every vehicle with critical information is simply infeasible. The involvement of roadside access points can greatly reduce the overhead of data dissemination, improve its effectiveness, and guarantee that all vehicles can be reached.

With the development and advancement of technology and economy, roadside access points are expected to be widely deployed in the near future, particularly in urban areas. Research on data dissemination with a roadside station generally focuses on how to fully utilize the infrastructure to facilitate the dissemination [4]. In this paper, instead, we address a fundamental problem: How do we provide roadside infrastructures in a specific area so that all the vehicles or a desired percentage of vehicles are guaranteed to be reached when they move around in the area? We observe that, in urban area, a vehicle driving around visits intersections. If information is disseminated at one of the intersections, this vehicle can receive the information when it passes the intersection. To reach all the vehicles in the area, we aim to select multiple intersections so that all the vehicles driving in the area will pass by at least one of the selected intersections.

The research on this paper can be applied to two scenarios. First, a city has no roadside access points, and the city hall or some companies need to determine intersections to install access points. Second, the dissemination area already have many access points provided by coffee shops, cafes, and companies. An agency needs to disseminate data and determines which access points to rent to conduct the dissemination.

We tackle the problem in two steps. First, we determine the intersections to install access points or determine the available access points to rent, so that *it is guaranteed that every vehicle in the area can be reached*. This is for the dissemination of sensitive information, such as safety information. The objective of this paper is to minimize the number of access points to reduce cost. Second, when it is not necessary to reach all the vehicles and there is limited budget, we determine the intersections to install access points to *maximize the percentage of vehicles that are guaranteed to be reached given a budget*. Additionally, a guideline will be provided about the tradeoff between construction or renting cost and the percentage of drivers that can be reached.

In the first step, we consider the most general case in which all the possible paths can be taken by drivers. The objective is to calculate a set of intersections, every possible path taken by the drivers intersects at least one of which. The problem is named topology-aware intersection selection (TIS) problem. We prove that minimizing the number of selected intersections in TIS is NP-complete since the hitting set problem [5], which is a

Manuscript received September 23, 2012; revised January 19, 2013 and April 9, 2013; accepted June 6, 2013. Date of publication July 11, 2013; date of current version January 13, 2014. This work was supported by the National Science Foundation under Grant 1128369. The review of this paper was coordinated by Prof. J. Misic.

T. Yan and G. Wang are with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102-1982 USA (e-mail: ty7@njit.edu; gwang@njit.edu).

W. Zhang is with the Department of Computer Science, Iowa State University, Ames, IA 50011 USA (e-mail: wzhang@cs.iastate.edu).

Y. Zhang is with the Institute of Computing Technology, Chinese Academy of Science, Beijing 100190, China (e-mail: zhmj@ict.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2013.2272724

well-known NP-complete problem, can be reduced to TIS. We propose novel polynomial-time heuristics, i.e., adapted bipartite-based heuristics (ABS), to tackle this NP-C problem, and to theoretically prove that the approximation ratio of ABS is bounded by 4/3. This means that, if the minimum number of intersections to cover all the possible paths is N, in the worst case, ABS selects at most $4/3N$ intersections to cover the whole area. ABS is motivated by an observation. The vertex cover problem in bipartite graphs is polynomial-time solvable. We prove that if there exists an algorithm that solves the vertex cover problem in polynomial time, this algorithm solves TIS in polynomial time. Therefore, to solve TIS for a general geographic area, we first identify the relationship between the geographic area and a bipartite graph. Then, we design a novel algorithm to adjust the topology of the geographic area and transform it into a bipartite graph. The exact solution for the transformed bipartite graph is calculated using existing polynomial algorithms [6]. Finally, we convert the graph back to its original form and employ a series of rules to adjust the solution and thus solve the problem for the general geographic area. The whole scheme of the first step is named *Tailor-p*.

In the second step, we aim to further reduce the number of necessary installations by taking traffic information into consideration. Among all the theoretical possible paths, vehicles' real trajectories are only a subset. For example, in most cases, it is not likely that a driver drives back and forth in an area or keeps driving in a circle. Therefore, if the traffic information is known, we can further reduce the number of access points but can still reach all the vehicles. We call it the traffic correlation (TC) problem and prove that it is np-complete because the dominating set problem [5], which is a well-known np-complete problem, can be reduced to it. We employ the greedy heuristics, which is recognized as the best heuristics to get an approximation solution to similar problems [7]. Then, we relax the requirement to reach all the vehicles, consider budget constraint, and aim to balance between construction cost and the percentage of vehicles that can be reached. The employment of greedy heuristics can naturally achieve the goal. We call the scheme of the second step *Tailor-f*.

To summarize, the contributions of this paper are three-fold.

1) We formulate the intersection selection problem, which formally proves its NP-completeness, and provide smart heuristics that can achieve an approximation ratio bounded by 4/3.
2) ABS, which is the proposed heuristics, bridges a general planar graph and a bipartite graph with topology transformation. With ABS, the approximate solution to all the problems that are NP-hard in a general planar graph but polynomially solvable in a bipartite graph, e.g., an independent set problem, can be efficiently obtained in the general planar graph.
3) We prove that, given traffic information, the intersection selection problem remains NP-complete.

The remainder of this paper is organized as follows. We present the TIS problem in Section II and the ABS solution in Section III. The analysis of ABS is given in Section IV and its evaluation in Section V. The TC problem and the solution Tailor-f and its evaluation are presented in Section VI. Section VII introduces the related work. Finally, we conclude this paper in Section VIII.

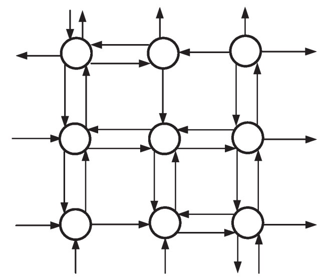


Fig. 1. Example of a geographical area.

II. Topology-Aware Intersection Selection Problem

A vehicle driving inside an area may visit multiple intersections. If a message is disseminated on at least one of these intersections, this vehicle will receive the message when driving in the area. For reaching all the vehicles driving in the area, in Tailor-p, we aim to select a minimum number of intersections of the area, so that the selected intersections intersect all the possible moving trajectories of vehicles within this area. Here, we first introduce the notations used in this paper. Then, we formally define the TIS problem, i.e., the underlying problem of Tailor-p, and prove its NP-completeness.

Throughout this paper, we use (v_i, v_j) to represent an edge connecting vertices v_i and v_j. We use $P(v_1, v_2, \ldots, v_k)$ to denote a path from vertex v_i, $v_2, \ldots,$ to v_k and $F(v_1, v_2, \ldots, v_k)$ to denote a face composed of vertices $v_1, v_2, \ldots, v_k$ and the corresponding edges.

A. Problem Statement and Proof

Definition 2.1—TIS Problem: The map of the target area can be viewed as a directed graph $G = (V, E)$, as shown in Fig. 1, where set V denotes intersections and the set of directed edges E denotes the set of roads. A directed edge $(u, v) \in E$ if and only if a vehicle can drive from intersection u to intersection v. A path is a sequence of vertices

$$P_i(v_1, v_2, \ldots, v_m) \tag{1}$$

where $(v_j, v_{j+1}) \in E$ for all j, $1 \leq j \leq m - 1$. The problem is identifying a set with a minimum number of vertices $V' \subseteq V$, such that, for each eligible path P_j, at least one intermediate node on the path is inside V'. That is, for every identified P_j, $P_j \cap V' \neq \emptyset$. ■

Note that if a coffee shop providing access points is not at an intersection, we can create a virtual intersection on the graph without affecting the solution.

Theorem 2.1: The TIS problem is NP-complete. ■

Proof of theorem 2.1: To facilitate the proof, we formulate the decision version of the TIS problem as follows: *Given integer $k \leq |V|$, is there a set of vertices $V' \subseteq V$ that solves the problem with $|V'| \leq k$?*

It is obvious to see that TIS $\in$ NP since a nondeterministic algorithm needs to only guess a subset of vertices and to check in polynomial time to determine whether that subset contains at least one intermediate node of every eligible path and is less than or equal to k.

We reduce the hitting set problem [5] to the TIS problem. Let an arbitrary instance of TIS be given by graph $G = (V, E)$. We construct set $A = \{a_1, \ldots, a_n\}$ and a finite collection $B_1, B_2, \ldots, B_m$ of subsets of A and claim that A has hitting set $H \subseteq A$ with $|H| \leq k$, such that $H \cap B_i \neq \emptyset$, $1 \leq i \leq m$, if and only if G has a TIS V' with $|V'| \leq k$.

To prove the claim, based on graph $G = (V, E)$, we do the following constructions:

$$\begin{cases} A = V \\ B_i = P_i, \quad 1 \leq i \leq m \\ V' = H. \end{cases} \tag{2}$$

If there is polynomial solution V' to the TIS problem with $V' \cap P_i \neq \emptyset$ and $|V'| \leq k$, replacing V' with H and P_i by B_i, we obtain solution H for the hitting set problem with $H \cap B_i \neq \emptyset$ and $|H| \leq k$.

Conversely, if $H \subseteq A$ is a polynomial solution to the hitting set problem in A with $|H| \leq k$ and $H \cap B_i \neq \emptyset$ for $1 \leq i \leq m$, by replacing H with V' and B_i with P_i, we obtain $V' \cap P_i \neq \emptyset$ and $|V'| \leq k$, i.e., a polynomial solution to the TIS problem.

During the reduction, all the transformations are polynomial transformation through replacement. By our reduction, graph G has a polynomial solution to the TIS problem if and only if A has a polynomial solution to the hitting set problem. Therefore, the TIS problem is NP-complete. ■

B. Analysis

For a geographic area, if the minimum number of intersections in each eligible path is 1, the solution to TIS becomes trivial, i.e., we have to select all the intersections of the area to intersect every possible path. To avoid this extreme case, in this paper, we assume that the minimum number of intersections in each eligible path is 2. That is, every vehicle driving in the area will pass at least two intersections.

We present Theorem 2.2, which is a foundation of the solution to the TIS problem.

Theorem 2.2: For planar graph $G = (V, E)$, the solution to the TIS problem is the solution to the vertex cover problem. ■

Proof of theorem 2.2: For all the eligible paths in G, we use P_1 to denote the collection of the paths that contain exactly two intersections and P_2 to denote the collection of the paths that have more than two intersections. According to the definition of the TIS problem, it is easy to get $P_1 = E$, and every path in P_2 is composed of the edges. Suppose that V' is a minimum vertex cover of graph G and $|V'| \leq k$. By definition, V' intersects all the edges in G. Then, obviously V' intersects all the paths in P_1 and P_2. Hence, the solution to the vertex cover problem solves the TIS problem with $|V'| \leq k$. ■

From Theorem 2.2, we can see that if there exists an algorithm that solves the vertex cover problem in polynomial time, this algorithm solves TIS in polynomial time. Clearly, for a general geographic area, there is no such polynomial solution for either vertex cover or TIS due to the NP-completeness of both problems. However, for a bipartite graph, the vertex cover problem can be solved in polynomial time [8], which means the exact solution to TIS for a bipartite graph can be easily obtained according to Theorem 2.2.

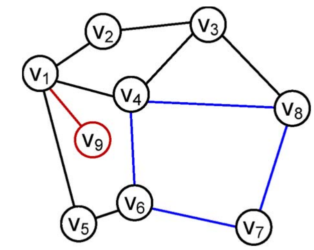


Fig. 2. Planar graph with four faces and one dead end.

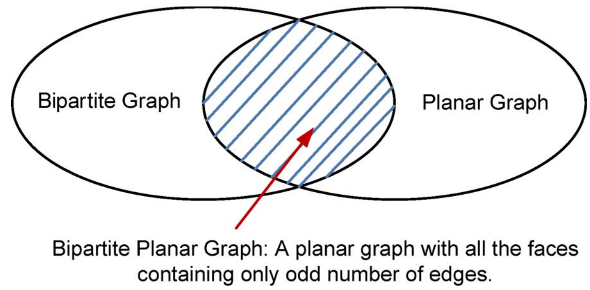


Fig. 3. Relationship between a planar graph and a bipartite graph.

III. Adapted Bipartite-Based Heuristics

According to Theorem 2.2, TIS for a bipartite graph can be solved in polynomial time, which motivates us to propose our solution, i.e., ABS. ABS transforms a geographic area to a bipartite graph. We are able to get the exact solution in the bipartite graph by applying existing algorithms. Then, we convert the graph back to the original planar graph and adjust the solution accordingly to solve the problem for the general geographic area.

Here, we first introduce preliminaries of the planar graph and the bipartite graph. Then, we present how to do the graph transformation and the intersection selection.

A. Preliminaries

A planar graph is defined as [8] a graph where all the edges intersect only at their common vertices. Inside the planar graph, a face is a circle that does not contain a subcircle. Moreover, if a vertex inside the region formed by a face and it connects to only one vertex of the face, this vertex is a "dead end" of the face. When counting the number of edges in a face, the edge of the dead end needs to be counted twice. For example, in Fig. 2, (v_4, v_6, v_7, v_8) is a face, whereas $(v_3, v_4, v_6, v_7, v_8)$ is not. In face $F(v_9, v_1, v_5, v_6, v_4)$, edge (v_1, v_9) is a dead end, and the number of edges in this face is 6, instead of 5.

A bipartite graph [8] is a graph whose vertices can be divided into two disjoint sets U and V, such that every edge connects a vertex in U to one in V. The relationship between the planar graph and the bipartite graph is shown in Fig. 3. A graph can

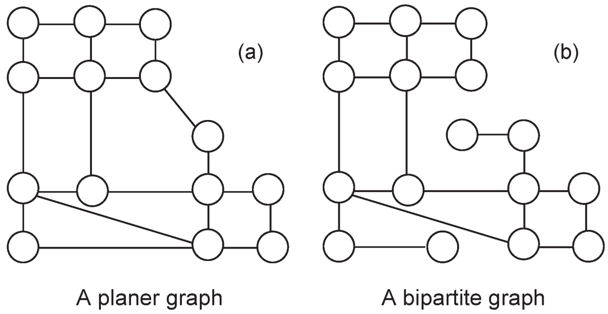


Fig. 4. Example of a planar graph and a bipartite graph.

be both a planar graph and a bipartite graph at the same time. *If all the faces of a planar graph only have an even number of edges, the graph is a bipartite graph* [8]. For example, Fig. 4(b) is a bipartite planar graph, whereas Fig. 4(a) is not because it has faces with an odd number of edges. The vertex cover problem for a bipartite graph is polynomial-time solvable by transforming to a maximum matching problem using König's theorem [8].

B. Philosophy of Topology Transformation

A geographic area consisting of roads and intersections without overpasses is a planar graph because any two intersecting roads must cross at an intersection. A bipartite planar graph is both a bipartite graph and a planar graph. The vertex cover problem in a bipartite planar graph can be solved in polynomial time. To seek for a solution to the vertex cover problem in a general planar graph, we propose the idea of transforming a planar graph to a bipartite planar graph. Our solution is dictated by such a theorem: *If all the faces of a planar graph only has an even number of edges, it is a bipartite planar graph* [8] Furthermore, if we compare the difference between the planar graph and the bipartite planar graph, as shown in Fig. 4, we can see that Fig. 4(b) is almost the same as Fig. 4(a), except for some detached edges and vertices. This inspires us that, to discover in a planar graph, if we perform modifications to transform all the faces with an odd number of edges, we can transform a planar graph to a bipartite planar graph that "looks similar" to it.

To transform a face with an odd number of edges to a face with an even number of edges, one intuitive method is to delete one of its edges. However, the solution obtained on the transformed graph is no longer a correct solution on the original graph. For example, Fig. 5(a) is a face with an odd number of edges. All the eligible paths in the original topology are listed in Fig. 5(a). To intersect every path, the selected intersections are $\{v_2, v_4, v_5\}$. After deleting edge (v_3, v_4), Fig. 5(b) is no longer a face. In this transformed graph, to intersect every path, the solution is $\{v_2, v_5\}$. Hence, vertex v_4 is not selected because edge (v_3, v_4) is no longer an eligible path in Fig. 5(b). This example shows that removing an edge invalidate the correctness of the solution because it removes all the valid paths that the edge is a part of. Thus, vertices connecting to the removed paths are not selected, although they should.

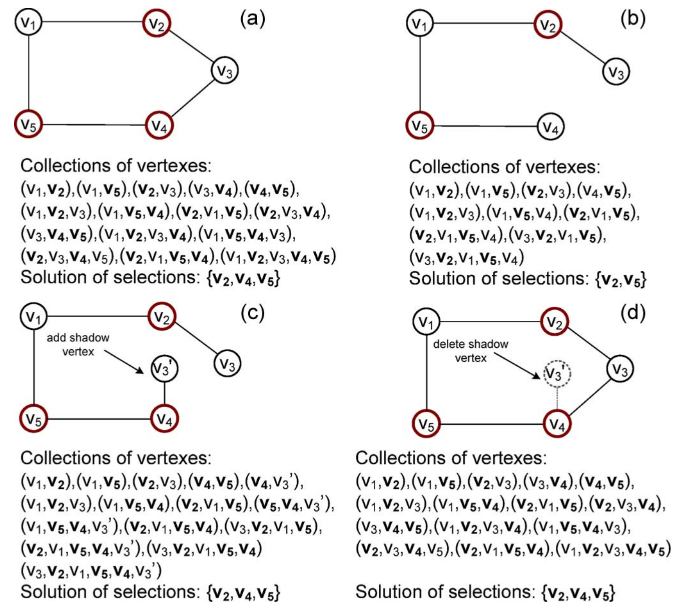


Fig. 5. Philosophy of topology transformation in ABS.

To address this problem, during the topology transformation, if an edge is removed from its two endpoints, we always create a shadow vertex, connect it to one of the endpoints, and generate a new edge. In this way, a correct solution can be obtained. For example, Fig. 5(c) shows a transformed graph, where v_3' is the added shadow vertex. The TIS solution for this graph is shown in red dots. We can see that vertex v_4 is in the selection because edge (v_4, v_3') is an eligible path in Fig. 5(c). The calculated selection $\{v_2, v_4, v_5\}$ based on Fig. 5(c) is exactly the solution for Fig. 5(a). After the selection, we convert the transformed graph back to the original one, as shown in Fig. 5(d). In some cases, if shadow vertices are selected, some adjustment is needed after transforming back to the original graph. The details will be presented in the following.

The exact solution for the transformed bipartite graph may not be the exact solution for the original graph. It may have redundancy and contain more vertices. That is, the existence of some vertices are for covering the fake edges created during the topology transformation. This redundancy cannot be completely removed; otherwise, we are able to obtain an exact solution to the NP-C problem. To reduce the redundancy, when detaching an edge from one of its two endpoints, we always detach it from the vertex with a relatively lower edge degree. This way, the edge degree of the vertex becomes even one degree lower, and the other vertex, which connects to the created fake edge, keeps its higher edge degree. In the exact vertex cover solution to the transformed bipartite graph, the vertex with a higher edge degree is more likely to be selected. When the vertex is selected, the fake edge is covered. Therefore, we do not need to select an extra vertex to cover this fake edge. Thus, the redundancy in this instance can be eliminated with our best effort.

For example, in Fig. 6(a), $\{v_2, v_3\}$ is the minimum vertex cover solution to the original graph. In Fig. 6(b), if we detach edge (v_1, v_3) from v_3, then $\{v_1, v_2, v_3\}$ is the minimum vertex cover for the transformed graph, where v_1 is the redundancy.

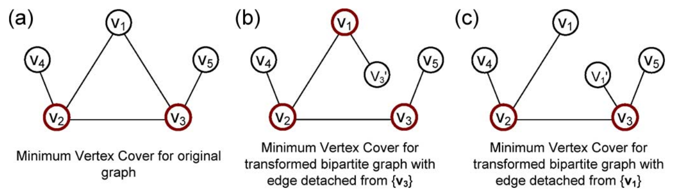


Fig. 6. Example of solution redundancy after topology transformation.

However, if we detach the edge from v_1, which has lower edge degree than v_3, as shown in Fig. 6(c), the solution will be $\{v_2, v_3\}$, which is the same as the exact solution to the original graph.

Generally, the transformation follows the following four steps. First, all faces of the graph are identified, and it is determined if they have an odd or even number of edges. Second, all the faces with an odd number of edges are transformed into faces with an even number of edges, by removing edges, adding shadow vertices, and creating fake edges accordingly. In this procedure, faces originally with an even number of edges may be affected, and they will be subsequently transformed. At the end of the procedure, all faces have an even number of edges, and a bipartite graph is obtained. Third intersections on the bipartite graph are selected by employing existing polynomial algorithms. Finally, the bipartite graph is transformed back to the original graph, and the solution is adjusted accordingly.

C. Graph Face Identification

Before we present how to identify all the faces, we present and prove a theorem to determine a face.

Theorem 3.1: Given a graph, place the graph in an arbitrary coordinate system. Start from an edge (v_i, v_j) and walk from v_i to v_j. At v_j, pick the leftmost edge of v_j in the coordinate system and walk to the other vertex connecting the edge. Keep walking this way until the original vertex v_i is reached. Then, the walking path consisting of all the visited vertices and edges form a face. ■

For example, as shown in Fig. 2, we start from edge (v_7, v_8) and walk from v_7 to v_8. We pick the leftmost edge (v_8, v_4) in the graph and keep walking the same way. Then, v_4 and v_6 are consequentially visited. In the end, the original vertex v_7 is reached, and face $F(v_7, v_8, v_4, v_6)$ is identified.

Proof of theorem 3.1: We prove the theorem by contradiction. According to the definition of a face in a planar graph, a face is a circle containing no subcircle inside. It is obvious that all the visited vertices and edges in Theorem 3.1 form a circle C_1. Thus, if these visited vertices do not form a face, it must contain subcircles inside. We pick the smallest one, i.e., subcircle C_2. There must be two edges sharing one vertex, one of which belongs to C_1, and the other belongs to C_2. Let the two edges be (v_i, v_k) and (v_i, v_m), which shares vertex v_i. Let (v_i, v_k) belong to C_1 and (v_i, v_m) belong to C_2. When vertex v_i is visited, edge (v_i, v_m) should be visited next, considering that C_2 is smaller; thus, (v_i, v_m) is the leftmost edge, instead of (v_i, v_k). However, since (v_i, v_k) belongs to C_1, this contracts to our operation that is always selecting a leftmost edge. Therefore, all the visited vertices and edges in Theorem 3.1 form a face. ■

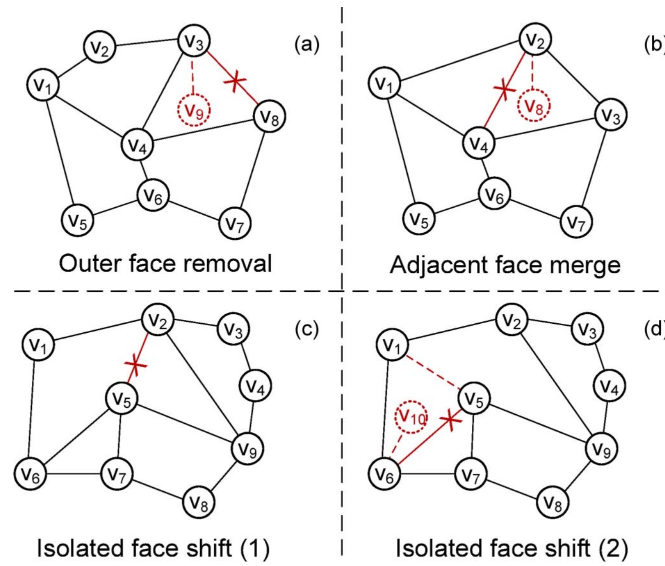


Fig. 7. Rules of topology transformation.

If a planar graph is extracted from a geographic area, the geographic position of every vertex can naturally serve as the coordinates of the vertices. Therefore, it is easy to identify the "leftmost" edge to visit. Starting from an edge, Theorem 3.1 identifies a face. Starting from every edge in the graph and applying the procedure in Theorem 3.1, all the faces of the graph will be identified.

D. Topology Transformation

After all the faces are identified, we start the transformation of faces with an odd number of edges. The topology of a face with an odd number of edges falls in one of the following three cases: 1) an outer face in the graph boundary; 2) an inner face that has at least one adjacent face with an odd number of edges; and 3) an inner face all of whose adjacent faces have an even number of edges. For the three cases, we apply *Outer face removal, Adjacent face merge, and isolated triangle shift*, respectively, to conduct the face transformation.

1) Outer Face Removal: It is easy to identify an outer face. If a face has an edge in the graph boundary, this face is an outer face. If an outer face has an odd number of edges, we simply remove the boundary edge, create a shadow vertex, and generate a new edge by connecting the shadow vertex and one vertex of the removed boundary edge. Fig. 7(a) shows an example. $F(v_3, v_4, v_8)$ is an outer face and has three edges. (v_3, v_8) is a boundary edge. We remove (v_3, v_8), create shadow vertex v_9, and connect v_9 to v_3. Then, a face with an odd number of edges is removed.

2) Adjacent Face Merge: After the outer face removal, none of the remaining faces with an odd number of edges has a boundary edge. The edges of these faces cannot be simply removed because adjacent faces may be affected. If one such face has at least one adjacent face with an odd number of edges, they can be merged by removing the shared edge. After that, a shadow vertex is created and attached to one vertex of the removed edge to generate a dead-end edge.

Theorem 3.2: In a planar graph, given two faces with an odd number of edges sharing an edge, the two faces can be merged into one face with an even number of edges if the shared edge is removed and a shadow vertex is added and attached to one vertex of the removed edge, generating a dead-end edge. ■

Proof of theorem 3.2: First, consider a face with k_1 edges and another face with k_2 edges sharing a common edge. Both k_1 and k_2 are odd numbers. If we merge these two faces by simply deleting this edge, then the merged face will have $(k_1 - 1 + k_2 - 1)$ edges, which is an even number. Then, we add a shadow vertex inside the face and attach to one of the two endpoints of the deleted edge. By doing so, we add a new edge to the graph. According to the definition of the face, this newly added edge is a dead end inside a face and needs to be counted twice when counting the number of edges for the face. Thus, now, this merged face has $(k_1 - 1 + k_2 - 1 + 2 = k_1 + k_2)$ edges. Obviously, $(k_1 + k_2)$ is an even number. ■

Fig. 7(b) shows an example. Two faces $F(v_1, v_2, v_4)$ and $F(v_2, v_3, v_4)$ have an odd number of edges, and they share common edge (v_2, v_4). We remove the edge, create shadow vertex v_8, and create dead-end edge (v_4, v_8). The new merged face $F(v_1, v_2, v_8, v_3, v_4)$ has an even number of edges.

3) Isolated Face Shift: If all the adjacent faces of an inner face with an odd number of edges have an even number of edges, we cannot apply the methods described earlier. We call such a face an isolated face. Face $F(v_2, v_5, v_9)$ in Fig. 7(c) is an example.

To solve the problem, we apply an iterative method. We remove the shared edge that is closest to a boundary and shift it, transforming the original isolated face into one with an even number of edges, resulting in another face with an odd number of edges. If the new face has a boundary edge or has an adjacent face with an odd number of faces, either outer face removal or adjacent face merge can be applied. Otherwise, we follow the same procedure until the resulting face is not an isolated face. Since we move toward the closest boundary, the procedure terminates.

For example in Fig. 7(c), $F(v_2, v_5, v_9)$ is an inner face with an odd number of edges, and all of its adjacent faces have an even number of edges. We first delete edge (v_2, v_5) from v_2 and attach it to vertex v_1. After the operation, the topology of Fig. 7(c) is transformed to the topology shown in Fig. 7(d), in which both the original face $F(v_1, v_5, v_6)$ and the newly generated face $F(v_5, v_6, v_7)$ have an odd number of edges. Therefore, we merge them as shown in Fig. 7(d).

Note that, during the shifting procedure, if an edge is removed, it is unnecessary to create a shadow vertex with a new edge because the shifting procedure already creates a new edge and attach it to another vertex. Here, for example in Fig. 7(c), we can simply treat v_1 as the shadow vertex for vertex v_5.

E. Intersection Selection

After the topology transformation, a map with any arbitrary planar graph topology is transformed to a bipartite graph, where a polynomial solution to a vertex cover problem can be applied directly. Take the map of the dissemination area shown in Fig. 4(a) as an example. After our bipartite graph transformation, the transformed graph can be shown as in Fig. 8(a). Then, König's theorem [8] can be applied to transform the vertex cover problem to a maximum matching problem, whose exact solution can be obtained by various well-known existing algorithms, such as the Hopcroft–Karp algorithm [8]. According to Theorem 2.2, the TIS problem in Fig. 8(a) is solved, and the selected intersections in Fig. 8(a) are shown as the red circles in Fig. 8(b).

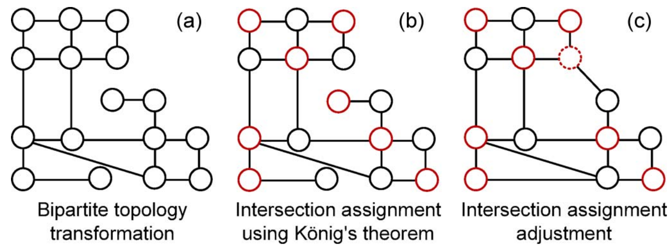


Fig. 8. Procedure of ABS.

F. Selection Adjustment

After obtaining the exact solution for the transformed bipartite graph, we transform it back to its original form and adjust our selections. Generally speaking, if a vertex is selected in the transformed graph, then we select the same vertex in the original graph. If the selected vertex is a shadow vertex, we select its corresponding original vertex. The detailed procedure works as follows.

1) We delete the shadow vertices and the corresponding edges created in the *outer face removal* and *adjacent face merge*, and add the original removed edges. If a shadow vertex is selected in the problem solution, we remove it and add the corresponding original vertex to the selection.
2) If an edge (v_i, v_j) in original graph shifts to (v_i, v_k) in the transformed bipartite graph, we disconnect v_i and v_k and reconnect v_i and v_j. If neither v_i or v_j is selected in the problem solution, we add either of them to the selection.

Taking the original graph in Fig. 4(a) as an example, the final selected vertices are shown as the red circles in Fig. 8(c). The dashed circle is the vertex added during selection adjustment because its shadow vertex is selected in the transformed graph, as shown in Fig. 8(b).

IV. Analysis of Adapted Bipartite-Based Heuristics

Here, we formally prove the correctness of our ABS and its approximation ratio being bounded by 4/3, analyze its computational complexity, and discuss its potential-wide application in many NP-C problems in planar graphs.

A. Correctness

Theorem 4.1: ABS obtains an approximate solution for the TIS problem. ■

Proof of theorem 4.1: We prove the theorem by contradiction. Let $G = (V, E)$ be the original graph and $G' = (V', E')$ be the transformed bipartite graph.

Suppose ABS obtains an exact vertex cover solution V'' for G', but V'' cannot cover all the edges in G. In this case, there exists an edge $(u, v) \in E$ such that $u \notin V''$ and $v \notin V''$. Since V'' covers all the edges in E' of graph G', it means $(u, v) \notin E'$ in graph G'. This is because of the edge removal. We analyze it in the following two cases. In the first case, the edge removal is done in *outer face removal* or *adjacent face merge*. Suppose the edge is detached from vertex u during the transformation. Since we always create a shadow vertex for every detached vertex, we let the shadow vertex be u' for u. Clearly, edge $(u', v) \in E'$. Then, we have $u' \in V''$ or $v' \in V''$ or both. Moreover, as presented in the *selection adjustment*, if a shadow vertex is selected in the transformed graph, we add its corresponding original vertex to the selection for the original graph. That is, in the solution for original graph G, we have $u \in V''$ or $v \in V''$ or both, which is a contradiction. In the second case, the edge removal is done in an *isolated face shift*. If an edge is not covered, it means none of its endpoints is selected. However, according to the *selection adjustment*, after an edge is shifted back, if none of its two endpoints is selected, we select either of them to the solution, which is also a contradiction.

Hence, ABS solves the TIS problem correctly. ■

B. Approximation Ratio

Let H be the exact vertex cover solution for the original graph $G = (V, E)$ and H' be the vertex cover solution obtained through ABS by transforming G to bipartite graph $G' = (V', E')$. $|H'|/|H|$ is the approximation ratio of ABS. Since H' obtains an approximate vertex cover solution for the original graph, we have $|H'|/|H| \geqslant 1$. According to Theorem 4.1, the case $|H'| > |H|$ means that the solution H' not only covers all the edges of the original graph but covers some edges with shadow vertices in the transformed graph as well.

To analyze the upper bound of the approximation ratio, we want to see at most how many redundant vertices that ABS uses to cover edges with shadow vertices. The shadow vertices can be only created during *outer face removal and adjacent face merge. Since outer face removal* only deals with faces in the graph boundary, to avoid special boundary cases, we only analyze *adjacent face merge*. We now construct a worst-case graph instance for ABS. The worst-case instance should be built based on the following philosophy.

1) The graph instance needs to merge as many faces as possible to create as many shadow vertices as possible.
2) After face merge, the ratio of the number of newly created edges to the number of original edges should be as high as possible.
3) According to ABS, during face merge, an edge is always detached from the vertex with a lower degree to reduce the redundancy of the solution. To construct the worst-case scenario, we should make the most of the vertices having the same edge degree so that when creating new edges, ABS only randomly selects an endpoint to detach.

To follow the first rule, we need to create a graph with all the faces containing an odd number of edges so that every two faces needs to break an edge and merge. For the second rule, since merging two faces only creates one new edge, the number of original edges in each face should be as small as possible. Therefore, we let each face in the graph contains only three edges. According to the third rule, the shape of the graph should be symmetric.

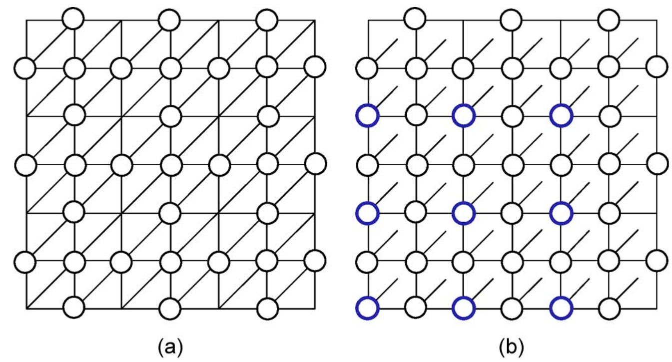


Fig. 9. Worst-case instance (all the grid intersections are vertices while circles are the selected vertices). (a) Instance of vertex cover for original graph. (b) Instance of vertex cover for transformed bipartite graph.

Thus, we construct a graph containing only triangles adjacent to each other, as shown in Fig. 9(a). Every two triangles need to break an edge and merge. All the vertices, except those in boundaries, have the same edge degree. It is obvious to see that the circles represent the exact minimum vertex cover for the original graph in Fig. 9(a). In ABS shown in Fig. 9(b), after topology transformation, we break edges for each two adjacent triangle faces and generate the bipartite planar graph shown. In addition to the black circles, extra blue circles are added to the minimum vertex cover solution for the transformed graph. The solution to the transformed graph consists of all the vertices of the graph (disregarding the special case in graph boundary). After we transform the graph back to the original, these blue circles cannot be eliminated and are the redundancy of our solution.

By comparing Fig. 9(a) and (b), it is easy to see that, in Fig. 9(a), for every two lines, the exact solution selects half of the vertices in one line and all the vertices in the other, whereas in Fig. 9(b), ABS selects all the vertices of the graph. Thus, in the worst case, we have

$$\frac{|H'|}{|H|} = \frac{4}{3}. \tag{3}$$

Therefore, ABS solves TIS with approximation ratio constantly bounded by 4/3.

In reality, the topology of a geographic area is not likely to look similar to the graph in Fig. 9(a). Instead, it contains faces mainly with four edges. That is, generally, most of the intersections in an area connect four road segments, and most blocks of the area are formed by four roads. In this case, ABS has a much lower approximation ratio. For example, for a normal graph from a geographic area with 10% of the intersections connecting three or five road segments, and the reset of intersections connecting four road segments, we have the average approximation ratio $|H'|/|H| < 1.1$.

According to Lund *et al.* [9], two other popular heuristics, i.e., random heuristics and greedy heuristics, have an approximation ratio of 1.7 and 2, respectively, in the worst case. This means that, in the worst case, greedy heuristics is about 1.3 times worse than our heuristics, and random heuristics is 1.5 times worse.

C. Computational Complexity

ABS contains four steps. In the first step, i.e., face identification, it takes $O(f_e)$ to identify a face, where f_e is the average number of edges in a face. Since face identification needs to start walking through all the edges, it takes $O(|E| \cdot f_e)$ to identify all the faces. In the second step, i.e., face transformation, it takes $O(|E|)$ to detach and shift edges. In the third step, i.e., intersection selection, the complexity of the well-known Hopcroft–Karp algorithm in solving the underlying TIS/vertex cover problem is $O(\sqrt{|V|} \cdot |E|)$. In the final step, it takes $O(|E|)$ to conduct the transformation back to the original and $O(|V|)$ for solution adjustment. In total, the computational complexity of ABS is

$$O\left(|E| \cdot f_e + |E| + \sqrt{|V|} \cdot |E| + |E| + |V|\right). \quad (4)$$

Considering a planar graph abstracted from a geographic area, the value of f_e is around 4, which is smaller than $\sqrt{|V|}$. Thus, the complexity of face identification is $O(\sqrt{|V|} \cdot |E|)$.

D. Discussion of Some Practical Issues

We deploy roadside stations at the selected intersections to capture all vehicle traffics of the area. In practice, it is very likely that not all the intersections are available to be selected for installing roadside stations or that the user may only want to rent the existing roadside stations deployed at certain intersections to disseminate messages. In this case, our goal is to select a minimum number of intersections from the available intersections where stations can be installed.

It is easy to see that solving this problem is equivalent to finding the solution of the vertex cover problem in the intersections where a roadside station can be installed, and it is also NP-hard. To address this problem, we first remove the vertices where we cannot install roadside stations from the graph. Here, if one of an edge's endpoints is removed, we add a fake vertex to it to complete the graph. If an edge's two endpoints are removed, then this edge is deleted because there is no way to cover the edge. We then transform the original graph to the one containing only applicable vertices and apply ABS to obtain the solution to the graph. The fake points will be eliminated during the *selection adjustment* phase.

An example is shown in Fig. 10. Roadside stations can be only installed at the intersections shown as bold circles in Fig. 10(a). Fig. 10(b) is the transformed graph where dashed circles are fake vertices. After applying ABS, the selected vertices are shown as red bold circles in Fig. 10(c).

E. The n-TIS Problem and Its Solution

As defined in Section II-B, in the TIS problem, the minimum number of intersections in each path is two. If the minimum

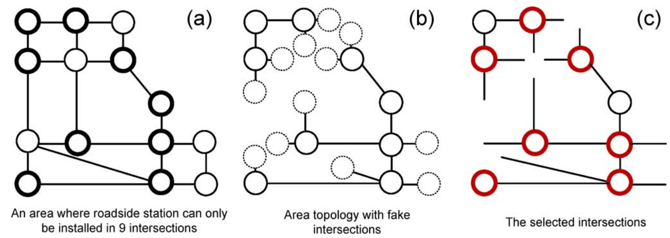


Fig. 10. Discussion of installation availability.

number of intersections in each path is $n (n \geq 3)$, we call this problem *n-TIS* problem $(n \geq 3)$. The *n-TIS* problem remains NP-complete, which is shown in Theorem 2.1. The solution to the vertex cover problem cannot be directly applied to address the *n-TIS* problem $(n \geq 3)$. The existing heuristic solutions proposed for the hitting set problem can be employed to solve it, which is proven in Theorem 4.2.

Theorem 4.2: For graph G, the solution to the hitting set problem constructed in the way in Theorem 2.1 with $|B_i| \geq 3$ is the solution to the *n-TIS* problem $(n \geq 3)$. ■

Proof of theorem 4.2: If solution H solves the hitting set problem constructed in Theorem 2.1 with $|H| < k$, $H \cap B_i \neq \emptyset$ for $1 \leq i \leq m$ and $|B_i| \geq 3$, replacing H by V' and B_i by P_i, we obtain a solution for the *n-TIS* problem with $V' \cap P' \neq \emptyset$, $|V'| \leq k$, and $|P_i| \geq 3$. ■

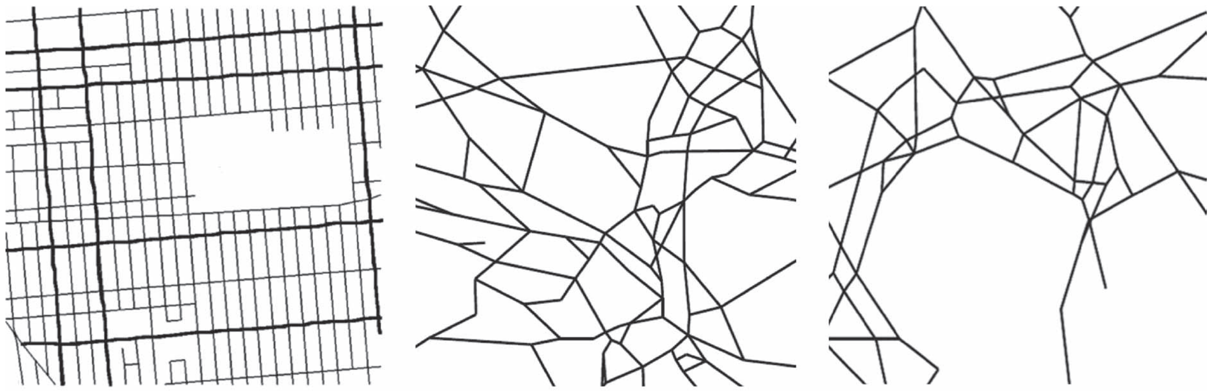
F. Remarks

By using ABS, we are able to convert an arbitrary planar graph to a bipartite planar graph and address it with very low worst-case approximation ratio of 4/3. Compared with the greedy heuristics with an approximation ratio higher than 1.7 [9] and the random heuristics with an approximation ratio of 2, ABS addresses TIS effectively. ABS can be adapted to be efficient heuristics for addressing the problems that are NP-complete in general planar graphs and polynomial-time solvable in bipartite graphs, such as the maximum independent set problem [5].

V. Evaluation of Adapted Bipartite-Based Heuristics

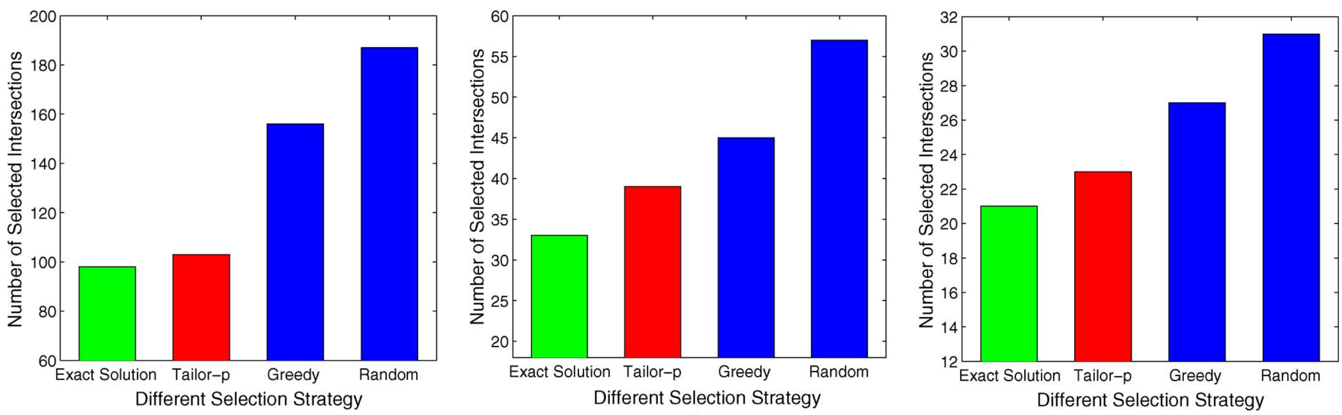
In Section IV-B, we formally prove that the approximation ratio of ABS in the worst case is 4/3. Here, we evaluate the performance of ABS in three real maps, instead of an artificially generated graph, and compare it with the greedy heuristics, the random heuristics, and the exact solution.

The first map is a 2 km × 2 km area in Brooklyn, New York City, NY, USA, shown in Fig. 11(a), which is extracted from the Topologically Integrated Geographic Encoding and Referencing (TIGER/Line) database of the U.S. Census Bureau [10]. It is a typical urban area with 332 intersections, in which most faces contain four vertices. To further evaluate the performance of the ABS in extreme cases, two other maps, i.e., a 10 km × 10 km area in Zurich, Switzerland, shown in Fig. 11(b) and a 10 km × 10 km area in Baar, Switzerland, shown in Fig. 11(c), are also employed. These two maps are extracted from the Open Street Map [11]. Fig. 11(b) has 75 intersections, and Fig. 11(c)



(a) (b) (c)

Fig. 11. Simulation maps. (a) Map of New York City, NY, USA, with 332 intersections. (b) Map of Zurich, Switzerland, with 75 intersections. (c). Map of Baar, Switzerland, with 46 intersections.



(a) (b) (c)

Fig. 12. Performance comparison of different heuristics. (a) Map of NYC, USA, with 332 intersections. (b) Map of Zurich, Switzerland, with 75 intersections. (c) Map of Baar, Switzerland, with 46 intersections.

has 46 intersections. They have blocks of irregular shapes, in which most faces contain three or five vertices. The three maps are of completely different topologies, which can evaluate the performance of the ABS comprehensively.

We run ABS, the greedy heuristics, and the random heuristics on the three maps and compare them with the exact solution obtained by an exponential algorithm. Fig. 12 shows the number of intersections selected by the three heuristics and the exact solution. In Fig. 11(a), ABS selects 103 intersections, whereas the exact solution selects 98 intersections, as shown in Fig. 12(a). In this instance, the approximation ratio achieved by ABS is 1.051. When most faces in a map contain an even number of edges, the topology of the map is close to a bipartite graph, and our ABS can achieve nearly optimal performance. The greedy heuristics selects 155 intersections and the random heuristics selects 190 intersections, which are 1.5 times worse and 1.84 times worse than our heuristics, respectively. In Fig. 12(b) and (c), we can see that ABS has the approximation ratios of 1.182 and 1.095, respectively, which are much better than that of the greedy heuristics and the random heuristics. When the number of faces with an odd number of edges increases, ABS tends to select more redundant intersections. From the evaluation, we can see that, in a real map, the performance of ABS is very close to the exact solution, and ABS outperforms other heuristics even more, compared with that in the worst-case graph.

VI. TAILOR-F: TRAFFIC-AWARE SELECTION

Among all the theoretically possible paths, only a subset will be taken by drivers in most cases. For example, drivers are not likely to drive back and forth, although they can do so. Therefore, given traffic information on an area, we can further improve the solution calculated by Tailor-p to select an even smaller number of intersections while still covering all the paths taken by the drivers. In addition, the traffic information can also tell which paths are most frequently visited by most drivers. When there is a budget limit, it is desirable that an even smaller number of intersections are selected, and most of the drivers can still be reached.

Here, we assume that the traffic information is given, based on which, we first formally prove that the problem to select the minimum number of intersections to intersect all the paths taken by the drivers is still NP-hard. Then, we discuss how to obtain a tradeoff between installation cost and the percentage of drivers to reach by employing the greedy heuristics [7], which is the best known heuristics to solve similar problems. We name our algorithm Tailor-f. Previous work MCP-sz [12] addresses a similar problem compared with Tailor-f. However, Tailor-f only selects intersections from those calculated by Tailor-p and thus reduces the traffic analysis complexity, as shown in Section VI-C. Both Tailor-f and MCP-sz assume that the traffic information is known and employ the greedy heuristics in [7].

Our contribution resides in our formally proving the NP-C nature of Tailor-f. Note that, in reality, we cannot obtain the traffic information beforehand. Only historical information can be used as a prediction of future traffic, and the predication accuracy cannot be perfect and will affect the correctness of the intersection calculation to a certain degree.

A. Theoretical Proof

The problem we aim to address is to select a minimum number of intersections so that all the vehicle paths given in the traffic information intersect with at least one of them. The given traffic information on an area can be in such a format: a list of intersections associated with vehicle IDs passing them. The information can be abstracted into a graph $G = (I + M, E)$. Here, M is the set of vehicles, and I is the set of intersections visited by them. $\forall u \in I$ and $\forall v \in M$, $(u, v) \in E$ if and only if vehicle v has visited intersection u. Formally speaking, the problem is to identify a set with a minimum number of vertices $I_{\mathrm{tc}} \subseteq I$, such that for all $u \in M$, there is a $v \in I_{\mathrm{tc}}$ with $(u, v) \in E$. We call it the *TC problem*.

Theorem 6.1: The TC problem is NP-complete. ■

Proof of theorem 6.1: To facilitate the proof, we formulate the decision version of the TC problem as follows: Given an integer $k \leq |I|$, is there a set of vertices $I_{\mathrm{tc}} \subseteq I$ that solves the problem with $|I_{\mathrm{tc}}| \leq k$?

It is easy to see that TC $\in$ NP since a nondeterministic algorithm only needs to guess a subset of vertices $I_{\mathrm{tc}} \subseteq I$ with an appropriate size and to check in polynomial time whether every vertices in M connect to at least one vertex in I_{tc}.

We reduce the dominating set problem [5] to the TC problem. We construct a graph $G' = (V', E')$ and claim that G' has dominating set $V_d \subseteq V'$ with $|V_d| \leq k$, in which for each $u \in V' - V_d$, there exists $v \in V_d$ such that $(u, v) \in E'$, if there is a solution I_{tc} to TC in graph G with $|I_{\mathrm{tc}}| \leq k$.

To prove the claim, we let $G' = G$. That is, $G' = (I' + M', E')$, $I' = I$, $M' = M$, and $E' = E$. Then, we create a set of edges E'' for graph G', such that every two vertices in I' are connected by some edge in E''. Therefore, I' forms a clique with size $|I'|$. Fig. 14 shows an instance of graph $G = (I + M, E)$ and $G' = (I' + M', E' + E'')$.

If there is a polynomial solution I_{tc} to TC problem in graph G with $|I_{\mathrm{tc}}| \leq k$, we let $V_d = I_{\mathrm{tc}}$. Then, in graph G', every vertex in M' must connect to V_d via some edge in $E' - E''$. Moreover, since I' is a clique, then every vertices in $I' - V_d$ are connecting to V_d. Thus, V_d is a dominating set of G' with $|V_d| \leq k$.

Conversely, if there is a polynomial solution $V_d \subseteq I'$ to the dominating set for graph G' with $|V_d| \leq k$, then every vertex in M' must connect to V_d via some edge in $E' - E''$, and every vertex in $I' - V_d$ connect to V_d via some edge in E''. We simply let $I_{\mathrm{tc}} = V_d$. According to our construction, in graph $G = (V, E)$, there is a TC solution $I_{\mathrm{tc}} \subseteq I$ with $|I_{\mathrm{tc}}| \leq k$, such that, for all $u \in M$, there is a $v \in I_{\mathrm{tc}}$ for which $\{u, v\} \in E$.

By our reduction, the TC problem in graph G can be solved in polynomial time if and only if the dominating set problem in graph G' can be solved in polynomial time.

Therefore, TC problem is NP-complete. ■

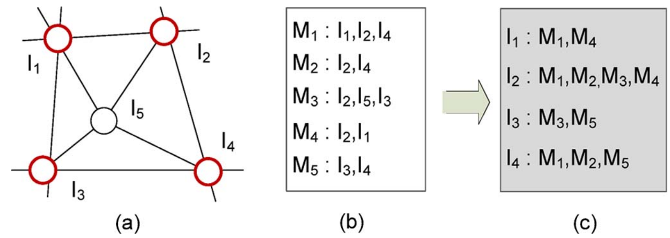


Fig. 13. Example of traffic collection. (a) A geographic area. Red circles are intersections selected by Tailor-p. (b) Real trajectories of all the vehicles. (c) IDs of vehicles associated with each visited intersection.

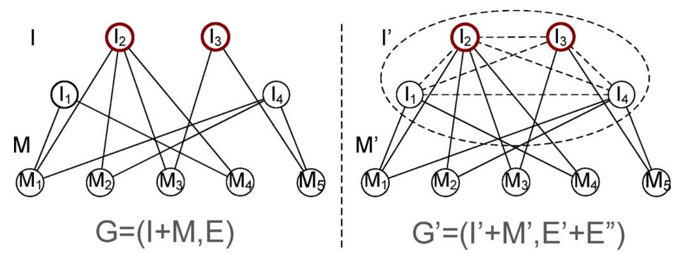


Fig. 14. "Yes" instance for TC problem (k = 2).

Fig. 13(a) is an example of an interested area. The four red circles are the selected intersections after applying Tailor-p. In Fig. 13(b), I_i and M_i are the ID of an intersection and a vehicle, respectively. The information can be transformed to a list of intersections associated with vehicle IDs passing them, as shown in Fig. 13(c). Note that, here, I_1, I_2, I_3, and I_4 are the selected intersections by Tailor-p. Therefore, the traffic information on I_5 is not needed and is, thus, discarded because Tailor-f is a further refinement of Tailor-p.

In Fig. 14, a set of two vertices $\{I_2, I_3\}$ is a "yes" instance for graph G and G' with $k = 2$. In Figs. 13 and 14, we can see that, compared with Tailor-p, by taking the advantage of traffic information, Tailor-f further reduces the number of selected intersection from 4 to 2.

B. Greedy Heuristics and Reachability

A greedy heuristics [7] is employed to solve the given NP-complete problem. According to [7], it can achieve an approximation ratio of $H(s)$, where s is the size of the largest set, and $H(n)$ is the nth harmonic number, i.e.,

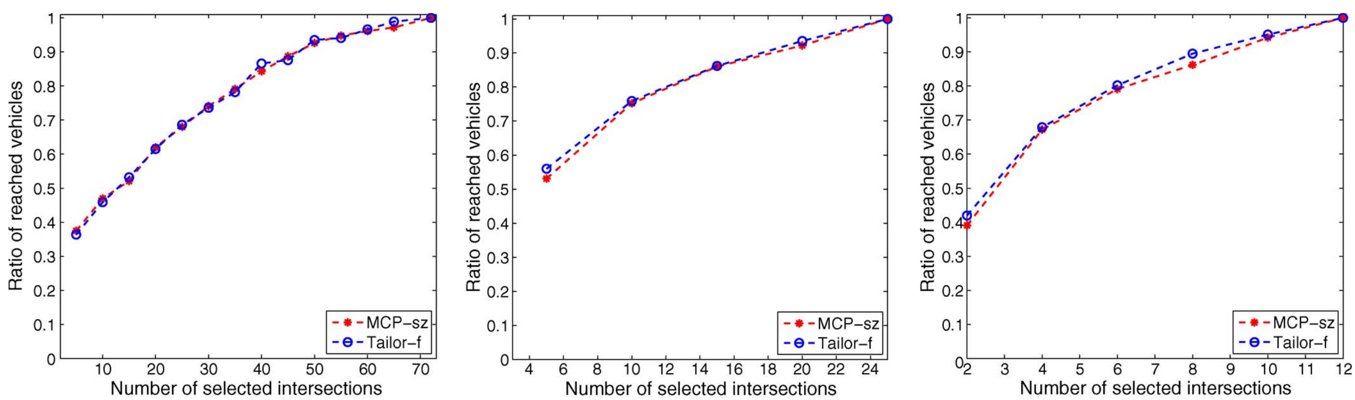
$$H(n) = \sum_{k=1}^{n} \frac{1}{k} \leq \ln n + 1. \tag{5}$$

Greedy heuristics is essentially the best possible polynomial-time approximation algorithm for a general set-cover-like problem, under plausible complexity assumptions according to [9].

Given traffic information, we have the TC graph $G = (I + M, E)$, where the degree of each vertex in I represents the amount of visited vehicles at its corresponding intersection. In Tailor-f, we always select a vertex in I with the highest degree, which is the vertex in I connecting most vertices in M. Then, we delete the vertex along with the edges and vertices connected to it. We repeat the procedure until all the vertex in M are deleted.

TABLE I
SIMULATION SETTINGS

Parameter	Value
Number of vehicles	600
Number of vehicles enter/exit per sec	15
Vehicle speed	10∼20 m/s
Traffic collection time	24 hours
Average NYC trace file size	2.6 G Byte, 5.2×10^8 lines
Average Zurich trace file size	1.7 G Byte, 3.5×10^8 lines
Average Baar trace file size	1.5 G Byte, 3.0×10^8 lines
Hardware	Intel Core 2 Due 2.2 GHz CPU, 4 GB memory



(a) (b) (c)

Fig. 15. Percentage of reached vehicles under different number of selected intersections. (a) Map of New York City, NY, USA. (b) Map of Zurich, Switzerland. (c) Map of Baar, Switzerland.

The greedy heuristics can naturally provide users a guideline to achieve a balance between cost and performance (defined as the percentage of vehicles reached). More specifically, given a budget, which can afford to install or rent a certain number of access points, greedy heuristics can provide the intersections that can reach most vehicles. In addition, a chart can be provided to illustrate the percentage of vehicles that can be reached under different budgets.

C. Performance Evaluation

We compare Tailor-f and MCP-sz [12]. Both heuristics are based on greedy heuristics in [7].

The difference between Tailor-f and MCP-sz or other general greedy heuristics is that, in MCP-sz and other general greedy heuristics, all the intersections of the map are candidate intersections, whereas in Tailor-f, only the intersections calculated by Tailor-p can be selected. This drastically saves the computational effort in traffic analysis. Tailor-f is a further refinement of Tailor-p, given traffic information.

We run the evaluation also on the maps shown in Fig. 11. The moving trace of vehicles is generated by the open-source microscopic space-continuous and time-discrete vehicular traffic generator package Simulation of Urban Mobility (SUMO) [13]. SUMO uses a collision-free car-following model to determine the speeds and the positions of the vehicles. The output from SUMO is converted into traffic information required for both Tailor-f and MCP-sz. We generate ten traffic files. The employed system parameters are listed in Table I. Each figure shown in the following is the average of the results on ten traffic files.

Fig. 15 shows the percentage of vehicles reached under a different number of selected intersections by Tailor-f and MCP-sz. In Fig. 15, we can see that the performance of both methods is very close to each other. We can also see that the tradeoff between cost and the percentage of vehicles can be reached. Users with a limited budget can make a decision based on the figure.

The advantage of Tailor-f is that it requires less traffic information than MCP-sz because Tailor-f selects intersections from those calculated by Tailor-p. Thus, only the traffic information associated with the candidate intersections is needed by our algorithm, which reduces the cost of traffic collection and trace analysis.

To evaluate how Tailor-f saves computing effort comparing with MCP-sz, we plot the computation time of Tailor-p, Tailor-f, and MCP-sz in Fig. 16. For the New York City map with a 2.6-GB trace file, Tailor-p takes about 0.3 h to compute a solution because it only needs to analyze the area topology and has nothing to do with the vehicle traces. After the calculation of Tailor-p, Tailor-f takes 2.9 h to select intersections from the intersections selected by Tailor-p to intersect all the vehicle traces. Therefore, the total is 3.2 h. MCP-sz takes 8.3 h to compute a solution, as shown in Fig. 16(a). Compared with MCP-sz, Tailor-f requires only one third of the computing effort in calculating a solution because it only needs to analyze the vehicle traces at the intersections that are selected by Tailor-p (103 intersections in this instance), whereas MCP-sz needs to analyze the vehicle traces at all the intersections of the area (332 intersections in this instance). Similarly, for Zurich map and Baar map, Tailor-p takes less than half of the amount of time of MCP-sz to compute a solution, as shown in Fig. 16(b) and (c), respectively.

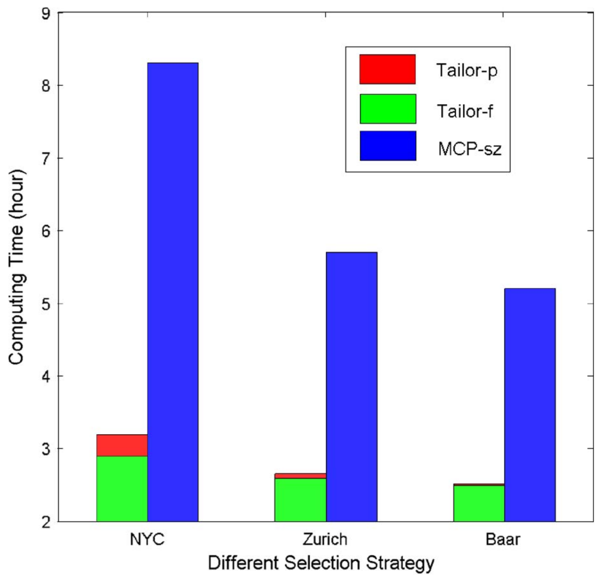


Fig. 16. Computation time of different strategies.

In Fig. 16, we can see that Tailor-f greatly reduces computational effort by only analyzing the vehicle traces at the intersections selected by Tailor-p. Moreover, practically, there may be more vehicles in an area (e.g., 6000 vehicles), and we may need to collect vehicle traces over a longer time (e.g., a week). The size of vehicle traces may be over 100 GB and Tailor-f can save more time in computing a solution. Compared with the analysis of vehicle traces, the operation of Tailor-p only adds very small overhead to the entire computation time. Thus, it is worth it to first apply Tailor-p and then apply Tailor-f to calculate a solution.

VII. Related Work

This paper is on infrastructure planning for data dissemination in VANETs with or without traffic information. Current research literature mainly focuses on balancing the access point coverage and installation cost, assuming the traffic information is known. MCP-sz [12] addresses a similar problem as our Tailor-f. The difference is that Tailor-f selects intersections from those calculated by Tailor-p, which reduces the traffic analysis time. In addition, we formally prove that Tailor-f addresses an NP-complete problem. Cataldi *et al.* maximize the joint user/operator benefit based on the user's utility [14]. They also consider many practical issues, such as polygon-based coverage. Minimum required transmission time is considered in [15] when planning roadside infrastructures, whereas Liang *et al.* make a tradeoff between the number of hops in intervehicle communication and the number of installed access points [16]. For improving driving convenience, Lee *et al.* aim to maximize the connectivity of the network [17]. Given the number of access points, the communication range, and the collected vehicle traces, the proposed algorithm calculates the locations to install access points to cover most of the vehicle traces. The scheme of planning virtual access points is proposed in [18], in which vehicles cache the content obtained from access points and disseminate in places where no access point is present. A statistical model is built to calculate the probability of each vehicle visiting each intersection in [19]. Based on the model, a planning heuristics is proposed to select intersections to install access points to maximize the connectivity. An adapted generic algorithm is proposed in [20] to select a minimum number of positions along the road to install access points, such that the whole road is covered and any wireless device on the road can communicate with one of the access points. Poff *et al.* develops a framework that provides methods on how to collect vehicle traffic data and how to train the system with the collected information [21]. Dubey *et al.* analyze the positions to place the access points in an intersection to maximize radio coverage [4].

Extensive research has been conducted in data dissemination through vehicle-to-vehicle communication with or without the help of roadside stations. Abiding Geocast [22] delivers time-stable message to a target area, and it has been well studied in [1], [2], and [23]. Protocols for dissemination relay are proposed in [1]. Yu and Heijenk propose to let the vehicles traveling in the opposite direction relay the message broadcasts [2]. The value of timer length is analyzed in [23]. Persistent data dissemination in VANETs has been addressed in [24]–[29]. Mylonase *et al.* propose a flooding-based dissemination protocol [24]. For dissemination collaboration and persistence, the relay strategy are studied in [25] and [27]. The scenario of dynamic region is studied in [26]. Leontiadis *et al.* propose a persistent dissemination scheme for a publish/subscription system, which allows a message to be disseminated persistently in an area to reach subscribers [28]. Wagner *et al.* evaluates the performance of data dissemination [29]. The area map information is used in [30]–[32] to improve the efficiency in dissemination and location-based service. A large-scale geocast scheme is proposed in [33], which handles obstacles and topology complexity in different terrains. Schwartz *et al.* designs a directional data dissemination protocol by using carry-and-forward method based on the road layout [3].

VIII. Conclusion

In this paper, we have designed Tailor-p and Tailor-f to select a minimum number of intersections to install access points with or without the traffic information. We formally prove the NP-C nature of both problems and employ effective heuristics to solve them. For Tailor-p, we prove that the approximation ratio of our ABS heuristics outperforms existing heuristics.

References

[1] S. D. Hermann, C. Michl, and A. Wolisz, "Time-stable geocast in intermittently connected ieee 802.11 manets," in *Proc. VTC*, 2007, pp. 1922–1926. [Online]. Available: http://dx.doi.org/10.1109/VETECF.2007.404

[2] Q. Yu and G. Heijenk, "Abiding geocast for warning message dissemination in vehicular ad hoc networks," in *Proc. ICC Workshops*, 2008, pp. 400–404. [Online]. Available: http://dx.doi.org/10.1109/ICCW.2008.81

[3] R. Schwartz, R. Barbosa, N. Meratnia, G. Heijenk, and H. Scholten, "A directional data dissemination protocol for vehicular environments," *Comput. Commun.*, vol. 34, no. 17, pp. 2057–2071, Nov. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2011.03.007

[4] B. Dubey, N. Chauhan, and S. Pant, "Effect of position of fixed infrastructure on data dissemination in VANETs," *Int. J. Res. Rev. Comput. Sci. (IJRRCS)*, vol. 2, no. 2, p. 482, Mar. 2011. [Online]. Available: http://www.researchgate.net/publication/228653668

[5] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979. [Online]. Available: http://dl.acm.org/citation.cfm?id=574848
[6] J. Hopcroft and R. Karp, "An n5/2 algorithm for maximum matchings in bipartite graphs," *SIAM J. Comput.*, vol. 2, no. 4, pp. 225–231, Jan. 1973. [Online]. Available: http://dx.doi.org/10.1109/SWAT.1971.1
[7] V. Chvatal, "A greedy heuristic for the set-covering problem," *Math. Oper. Res.*, vol. 4, no. 3, pp. 233–235, Aug. 1979. [Online]. Available: http://mor.journal.informs.org/content/4/3/233.full.pdf
[8] D. West, *Introduction to Graph Theory*., 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.
[9] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems," *J. ACM*, vol. 41, no. 5, pp. 960–981, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1145/185675.306789
[10] U.S. Census Bureau TIGER/Line 2009. [Online]. Available: http://www.census.gov/geo/www/tiger/
[11] Open Street Map. [Online]. Available: http://www.openstreetmap.org/
[12] O. Trullols, M. Fiore, C. Casetti, C. Chiasserini, and J. Ordinas, "Planning roadside infrastructure for information dissemination in intelligent transportation systems," *Comput. Commun.*, vol. 33, no. 4, pp. 432–442, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2009.11.021
[13] Centre for Applied Informatics (ZAIK) and the Institute of Transport Research German Aerospace Centre, Sumo—Simulation of urban mobility. [Online]. Available: http://sumo.sourceforge.net/
[14] P. Cataldi and J. Harri, "User/operator utility-based infrastructure deployment strategies for vehicular networks," in *Proc. IEEE Veh. Technol. Conf., Fall*, 2011, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/VETECF.2011.6093125
[15] E. Cavalcante, A. Aquino, G. Pappa, and A. Loureiro, "Roadside unit deployment for information dissemination in a VANET: An evolutionary approach," in *Proc. 14th Int. Conf. Genetic Evol. Comput. Conf. Companion*, 2012, pp. 27–34. [Online]. Available: http://dx.doi.org/10.1145/2330784.2330789
[16] Y. Liang, H. Liu, and D. Rajan, "Optimal placement and configuration of roadside units in vehicular networks," in *Proc. IEEE Veh. Technol. Conf., Spring*, 2012, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/VETECS.2012.6240345
[17] J. Lee, "Design of a network coverage analyzer for roadside-to-vehicle telematics networks," in *Proc. Int. Conf. Softw. Eng., Artif. Intell., Netw., Parallel/Distrib. Comput.*, 2008, pp. 201–205. [Online]. Available: http://dx.doi.org/10.1109/SNPD.2008.27
[18] N. Frangiadakis, D. Camara, F. Filali, A. Loureiro, and N. Roussopoulos, "Virtual access points for vehicular networks," in *Proc. Mobilware*, 2008, p. 14. [Online]. Available: http://dl.acm.org/citation.cfm?id=1361492.1361509
[19] B. Aslam, F. Amjad, and C. Zou, "Optimal roadside units placement in urban areas for vehicular networks," in *Proc. IEEE ISCC*, 2012, pp. 000423–000429. [Online]. Available: http://dx.doi.org/10.1109/ISCC.2012.6249333
[20] H. Ghaffarian, M. Soryani, and M. Fathy, "Planning VANET infrastructures to improve safety awareness in curved roads," *J. Zhejiang Univ.*, vol. 13, no. 12, pp. 918–928, 2012. [Online]. Available: http://www.zju.edu.cn/jzus/article.php?doi=10.1631/jzus.C1200082
[21] L. Poff, A. Gokhale, and M. McDonald, "A framework for broker placement in vehicular ad hoc networks," in *Proc. Int. Conf. CTS*, 2012, pp. 182–189. [Online]. Available: http://dx.doi.org/10.1109/CTS.2012.6261048
[22] C. Maihofer, T. Leinmuller, and E. Schoch, "Abiding geocast: Timecstable geocast for ad hoc networks," in *Proc. VANET*, 2005, pp. 20–29. [Online]. Available: http://dx.doi.org/10.1145/1080754.1080758
[23] H. Joshi, M. Sichitiu, and M. Kihl, "Distributed robust geocast: A multicast routing for inter-vehicle communication," in *Proc. Workshop WiMAX, Wireless Mobil.*, 2007. [Online]. Available: http://www4.ncsu.edu/~mlsichit/Research/Publications/geocastHarsh.pdf
[24] Y. Mylonase, M. Lestas, and A. Pitsillides, "Speed adaptive probabilistic flooding in cooperative emergency warning," in *Proc. Int. Conf. Wireless Internet*, 2008, p. 81. [Online]. Available: http://dl.acm.org/citation.cfm?id=1554228
[25] S.-B. Lee, G. Pan, J.-S. Park, M. Gerla, and S. Lu, "Secure incentives for commercial ad dissemination in vehicular networks," in *Proc. Mobihoc*, 2007, pp. 150–159. [Online]. Available: http://dx.doi.org/10.1145/1288107.1288128
[26] Y.-S. Chen, Y.-W. Lin, and S.-L. Lee, "A mobicast routing protocol in vehicular ad-hoc networks," *Mobile Netw. Appl.*, vol. 15, no. 1, pp. 20–35, Feb. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11036-009-0176-3
[27] I. Leontiadis and C. Mascolo, "Opportunistic spatio-temporal dissemination system for vehicular networks," in *Proc. Workshop Mobile Opportunistic Netw.*, 2007, pp. 39–46. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.113.7998
[28] I. Leontiadis, P. Costa, and C. Mascolo, "A hybrid approach for content-based publish/subscribe in vehicular networks," *Pervas. Mobile Comput.*, vol. 5, no. 6, pp. 697–713, Dec. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.pmcj.2009.07.016
[29] C. Wagner and W. Franz, "Stochastic evaluation of geographical forwarding in vehicular ad hoc networks," in *Proc. GI Jahrestagung*, 2004, pp. 95–99. [Online]. Available: http://subs.emis.de/LNI/Proceedings/Proceedings50/article3114.html
[30] S. Kim, Y. Jung, S. Lee, E. Lee, and S. An, "Efficient geocast utilizing topology information database," in *Proc. Int. Conf. Comput. Inform. Technol. Workshops*, 2008, pp. 210–215. [Online]. Available: http://dx.doi.org/10.1109/CIT.2008.Workshops.23
[31] B. Dubey, N. Chauhan, L. Awasthi, N. Chand, and S. Pant, "Efficient VANET-based traffic information dissemination using centralized fixed infrastructure," in *Proc. Commun. Comput. Inf. Sci.*, 2011, vol. 168
[32] D. Ashok, M. Pai, and J. Mouzna, "Efficient map based location service for VANETs," in *Proc. Int. Conf. ITS Telecommun.*, 2011, pp. 387–392. [Online]. Available: http://dx.doi.org/10.1109/ITST.2011.6060087
[33] R. Hall, "An improved geocast for mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 2, pp. 254–266, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1109/TMC.2010.56

Tan Yan received the B.E. degree from Southeast University, Nanjing, China, and the M.E. degree from the New Jersey Institute of Technology, Newark, NJ, USA. He is currently working toward the Ph.D. degree with the Department of Computer Science, New Jersey Institute of Technology.

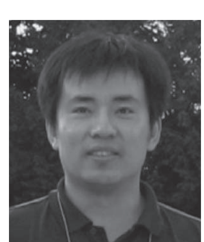
Wensheng Zhang received the B.S. degree from Tongji University, Shanghai, China; the M.S. degree from the Chinese Academy of Sciences, Beijing, China; and the Ph.D. degree from the Pennsylvania State University, University Park, PA, USA, all in computer science and engineering.

Since 2005, he has been a Faculty Member with the Department of Computer Science, Iowa State University, Ames, IA, USA, where he is currently an Associate Professor.

Dr. Zhang is a member of the Association for Computing Machinery.

Guiling Wang received the B.S. degree in software from Nankai University, Tianjin, China, and the Ph.D. degree in computer science and engineering and a minor in statistics from The Pennsylvania State University, University Park, PA, USA, in May 2006.

Since 2006, she has been with the New Jersey Institute of Technology, Newark, NJ, USA, where she is currently an Associate Professor.



Yujun Zhang received the B.S. degree in software from Nankai University, Tianjin, China, in 1999 and the Ph.D. degree in computer science from the Chinese Academy of Sciences, Beijing, China, in 2004.

He is currently an Associate Professor with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include future Internet architecture, network security, and trust.