Using Incomplete Information for Complete Weight Annotation of Road Networks

Bin Yang, Manohar Kaul, and Christian S. Jensen, *Fellow, IEEE*

Abstract—We are witnessing increasing interests in the effective use of road networks. For example, to enable effective vehicle routing, weighted-graph models of transportation networks are used, where the weight of an edge captures some cost associated with traversing the edge, e.g., greenhouse gas (GHG) emissions or travel time. It is a precondition to using a graph model for routing that all edges have weights. Weights that capture travel times and GHG emissions can be extracted from GPS trajectory data collected from the network. However, GPS trajectory data typically lack the coverage needed to assign weights to all edges. This paper formulates and addresses the problem of annotating all edges in a road network with travel cost based weights from a set of trips in the network that cover only a small fraction of the edges, each with an associated ground-truth travel cost. A general framework is proposed to solve the problem. Specifically, the problem is modeled as a regression problem and solved by minimizing a judiciously designed objective function that takes into account the topology of the road network. In particular, the use of weighted PageRank values of edges is explored for assigning appropriate weights to all edges, and the property of directional adjacency of edges is also taken into account to assign weights. Empirical studies with weights capturing travel time and GHG emissions on two road networks (Skagen, Denmark, and North Jutland, Denmark) offer insight into the design properties of the proposed techniques and offer evidence that the techniques are effective.

Index Terms—Spatial databases and GIS, correlation and regression analysis

1 Introduction

Reduction in greenhouse gas (GHG) emissions is crucial in combating global climate change. For example, the EU has committed to reduce GHG emissions to 20% below 1990 levels by 2020 [1]. To achieve these reductions, the transportation sector needs to achieve reductions. For example, in the EU, emissions from transportation account for nearly a quarter of the total GHG emissions [2], making transportation the second largest GHG emitting sector, trailing only the energy sector.

While improved vehicle and engine design are likely to yield GHG emission reductions, eco-routing is readily deployable and is a simple yet effective approach to reducing GHG emissions from road transportation [3]. Specifically, eco-routing can effectively reduce fuel usage and CO_2 emissions. Studies suggest that by providing eco-routes to drivers, approximately 8–20% in fuel savings and lower CO_2 emissions are possible in different settings, e.g., during peak versus off-peak hours, on highways versus areal roads, for light versus heavy duty vehicles [4], [5]. For example, an interesting municipal solid waste collection scenario, where a truck collects solid waste from several locations on Santiago Island, demonstrates a 12% fuel reduction due to eco-routes [6].

Vehicle routing relies on a weighted-graph representation of the underlying road network. To achieve effective eco-routing, it is essential that accurate edge weights that capture environmental costs, e.g., fuel consumption or GHG emissions, associated with traversing the edges are available. Given a graph with appropriate weights, eco-routes can be efficiently computed by existing routing algorithms, e.g., based on Dijkstra's algorithm or the A^* algorithm. However, accurate weights that capture environmental impact are not always readily available for a road network. This paper addresses the task of obtaining such weights for a road network from a collection of measured (*trip*, *cost*) pairs, where the *cost* can be any cost associated with a trip, e.g., GHG emissions, fuel consumption, or travel time.

Because the trips given in the input collection of pairs generally do not cover all edges of the road network and also do not cover all times of the day, data sparsity is a key problem. The cost of a trip, e.g., GHG emissions, differs during peak versus off-peak hours. Thus, it is inappropriate to use costs associated with peak-hour trips for obtaining edge weights to be used for eco-routing during off-peak hours.

Considering the road network and trips shown in Fig. 1, assume that the GHG emissions of trip 1 (traversed from 7:30 to 7:33) and trip 2 (traversed from 23:15 to 23:17) are also given, and assume that we are interested in assigning GHG emission weights to all edges in the network. The assignment of these weights to a large number of edges, e.g., *BC*, *BD*, *EG*, and *FG*, cannot be done directly since they are not covered by any trip. However, for example, *BD* can

- *B. Yang and M. Kaul are with the Department of Computer Science, Aarhus University, Aarhus DK-8200, Denmark. E-mail: {byang, mkaul}@cs.au.dk.*
- *C. S. Jensen is with the Department of Computer Science, Aalborg University, Aalborg Øst DK-9220, Denmark. E-mail: csj@cs.aau.dk.*

Manuscript received 31 July 2012; revised 24 Apr. 2013; accepted 28 May 2013. Date of publication 9 June 2013; date of current version 7 May 2014.
Recommended for acceptance by C. Shahabi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.89

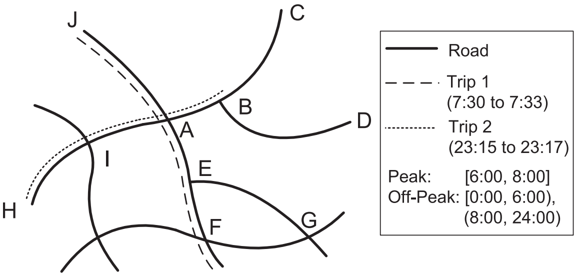


Fig. 1. Trips on a road network.

be annotated by considering its neighbor road segment *AB* which is covered by trip 2.

Assuming that the period from 6:00 to 8:00 is the sole peak-hour period (the remaining times being off-peak), trip 1 is not useful for assigning an off-peak weight to the edge *AE* because trip 1 traversed *AE* during peak hours. By taking into account the off-peak weights of *IA* and *AB* (covered by trip 2), it is, however, possible to obtain an off-peak weight for *AE*.

This paper proposes general techniques that take as input (i) a collection of (*trip*, *cost*) pairs, where *trip* captures the edges used and the times when the edges are traversed and the *cost* represents the cost of the entire trip; and (ii) an unweighted graph model of the road network in which the trips occurred. The techniques then assign travel cost based weights to all edges in the graph.

To the best of our knowledge, this paper is the first to study complete weight annotation of road networks using incomplete information. In particular, the paper makes four contributions. First, a novel problem, road network weight annotation, is proposed and formalized. Second, a general framework for assigning time-varying trip cost based weights to the edges of the road network is presented, along with supportive models, including a directed, weighted graph model capable of capturing time-varying edge weights and a trip cost model based on time varying edge weights. Third, two novel and judiciously designed objective functions are proposed to contend with the data sparsity. A weighted PageRank-based objective function aims to measure the variance of weights on road segments with similar traffic flows, and a second objective function aims to measure the weight difference on road segments that are directionally adjacent. Fourth, comprehensive empirical evaluations with real data sets are conducted to elicit pertinent design properties of the proposed framework.

The remainder of this paper is organized as follows. Following a survey of related work in Section 2, Section 3 covers problem definition and a general framework for solving the problem. Section 4 details the objective functions. Section 5 reports the empirical evaluation, and Section 6 concludes and discusses research directions.

2 Related Work

Little work has been done on weight annotation of road neworks. Trip cost estimation is a core component of our weight annotation solution. Given a set of (*trip*, *cost*) pairs as input, trip cost estimation aims to estimate the costs for trips that do not exist in the given input set. Weight annotation can be regarded as a generalized version of trip cost estimation, since if pertinent weights can be assigned to a road network, the cost of any trip on the road network can be estimated. For example, if a GHG emissions based weighted graph is available, the GHG emissions of a certain trip can be estimated as the sum of the weights of the road segments that the trip traverses.

Most existing work on trip cost estimation [7]–[10] focuses on travel-time estimation. In other words, their work focuses on travel time as the trip cost. In general, the methods for estimating the travel times of trips can be classified into two categories: (i) segment models and (ii) trip models.

Segment models [9]–[12] concern travel time estimation for individual road segments. For example, observers (e.g., Bluetooth sensors or loop detectors deployed along road segments) monitor the traffic on road segments, recording the flows of vehicles along the road segments. Thus, travel-time estimation tends to concern particular road segments. For example, some studies model travel time on a particular road segment as a time series and apply autoregressive models [9] to estimate the travel time on the road segment. T-Drive [10] models time-dependent travel time distributions on road segments using sets of histograms and enables the inference of future travel times using Markov chains [13]. One study incorporates Lagrangian measurements [12] into existing traffic flow models for freeways to estimate travel time distributions on specific freeways.

Segment models assume “hot” road segments where, preferably, substantial data is available. However, far from every road segment may have enough historical data in practical settings, e.g., due to the limited deployment of costly sensors. Segment models are not well suited for the weight annotation problem because the given (*trip*, *cost*) pairs typically fail to cover the whole network, meaning that many road segments lack the data needed to apply such models.

The trip models focus on estimating the costs of individual trips. Specifically, the costs of trips are considered more interesting than the costs of individual road segments. Given a collection of trips and their corresponding travel times, one study [8] proposes a Gaussian process regression based method to predict the travel times for unseen trips. However, the study has the limitation that all the trips are required to share the same source and target. This limitation renders the study of limited interest to us, since we aim at annotating every edge with a pertinent weight. Trajectory regression [7] was proposed recently to infer the travel times of arbitrary trips. The method is able to estimate the travel times of trips consisting of road segments with no or little traversal history by considering the travel time correlation of spatially adjacent road segments.

Trajectory regression is the most related method to our weight annotation problem. However, our study distinguishes itself with several unique characteristics. First, we propose a general framework for annotating edges in a road network with a range of trip cost based weights and are not constrained to travel time. Second, we identify the cost correlation of road segments sharing similar traffic flows, and we quantify this by using weighted PageRank values. Third, we consider the temporal cost correlation of adjacent road segments. For example, although two road segments *AB* and *BC* are adjacent, the cost of traversing *AB* during peak hours is not necessarily correlated to the

TABLE 1
Key Notation

Notation	Description
G, G'	The primal graph and the dual graph.
G'_k	The dual graph in traffic category tag tag_k.
$\mathbb{V}$, $\mathbb{E}$	The vertex set and the edge set.
$\mathbb{V}'$, $\mathbb{E}'$	The dual vertex set and the dual edge set.
$\mathbf{d}$	The cost variable vector for all edges.
$PR_k(v'_i)$	The weighted PageRank value of dual vertex v'_i in traffic category tag tag_k.

cost of traversing *BC* during off-peak hours. Fourth, we take into account the directionality of road segments and consider only *directional adjacency* when determining the cost correlation of spatially adjacent road segments. Last but not least, we conduct comprehensive experiments on real data sets (real trips and real road networks) to demonstrate the effectiveness of annotating road networks with both travel time based weights and GHG emissions based weights. The earlier study on trajectory regression [7] considers only synthetic data and estimates only travel times of trips.

In the intelligent transportation system research field [3], [14], [15], other travel costs (besides travel time) of trips are studied. For example, fuel consumption and GHG emissions of a trip can be computed based on instantaneous vehicle velocities and accelerations, the slopes of the road segments traversed, and the engine type. However, these methods are designed to estimate the costs of individual trips and are not readily applicable to the problem of annotating graph edges with trip cost based weights, notably edges that do not have any traversed trips.

3 Preliminaries

We cover the modeling that underlies the proposed framework, and we provide an overview of the framework and its setting.

We use blackboard bold upper case letter for sets, e.g., $\mathbb{E}$, bold lower case letters for vectors, e.g., $\mathbf{d}$, and bold upper case letters for matrices, e.g., $\mathbf{M}$. Unless stated otherwise, the vectors used are column vectors. The i-th element of vector $\mathbf{d}$ is denoted as $\mathbf{d}[i]$, and the element in the i-th row and j-th column of matrix $\mathbf{M}$ is denoted as $\mathbf{M}[i, j]$. Matrix $\mathbf{M}^\mathbf{T}$ is $\mathbf{M}$ transposed. An overview of key notation used in the paper is provided in Table 1.

3.1 Modeling a Temporal Road Network

A road network is modeled as a directed, weighted graph $G = (\mathbb{V}, \mathbb{E}, L, F, H)$, where $\mathbb{V}$ and $\mathbb{E}$ are the vertex and edge sets, respectively; L is a function that records the lengths of edges; F is a function that maps times to traffic categories; and H is a function that assigns time-varying weights to edges. We proceed to cover each component in more detail.

A vertex $v_i \in \mathbb{V}$ represents a road intersection or an end of a road. An edge $e_k \in \mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is defined by a pair of vertices and represents a directed road segment that connects the (intersections represented by) two vertices. For example, edge (v_i, v_j) represents a road segments that enables travel from vertex v_i to vertex v_j. For convenience, we call this graph representation of a road network the *primal graph*.

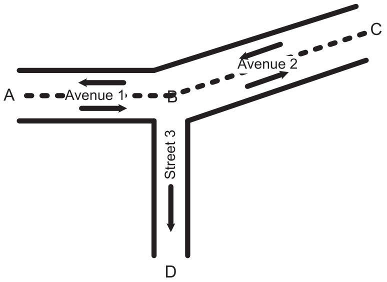


Fig. 2. Road network.

Fig. 2 captures the upper right part of the road network shown in Fig. 1 in more detail. Here, *Avenue 1* and *Avenue 2* are bidirectional roads, and *Street 3* is a one-way road that only allows travel from vertex *B* to vertex *D*.

The corresponding primal graph is shown in Fig. 3. In order to capture the bidirectional *Avenue 1*, two edges (A, B) and (B, A) are generated. Since *Street 3* is a one-way road, only one edge, (B, D), is created.

It is essential to model a road network as a directed graph because the cost associated with traveling in two different directions may differ very substantially. For example, traveling uphill is likely to have a higher fuel cost than traveling downhill. As another example, the congestion may also vary greatly for the two directions of a road.

Function $L : \mathbb{E} \rightarrow \mathbb{R}$ takes as input an edge and outputs the length of the road segment that the edge represents. If road segment *AB* is 135 meters long, we have $G.L((A, B)) = G.L((B, A)) = 135$.

Next, the cost of traversing the same edge may differ across time. This is typically due to varying degrees of congestions. Thus, GHG emissions or fuel consumption are likely to differ during peak versus off-peak times. To this end, function $F : TD \rightarrow TAGS$ models the varying traffic intensity during different periods. Specifically, F partitions time *TD* and assigns a *traffic category tag* in *TAGS* to each partition. The granularity of the tags are chosen so that the traffic intensity can be assumed to be constant during the time associated with the same tag. For example, F([0:00, 7:00)) = *OFFPEAK*, F([7:00, 9:00)) = *PEAK*, F([9:00, 17:00)) = *OFFPEAK*, etc.

Finally, function $H : \mathbb{E} \times TAGS \rightarrow \mathbb{R}$ assigns time dependent weights to all edges. In particular, H takes as input an edge and a traffic tag, and outputs the weight for the edge during the traffic tag.

Specifically, $G.H(e_i,\ tag_j) = d_{(e_i,\ tag_j)} \cdot\ G.L(e_i)$, where $d_{(e_i,\ tag_j)}$ indicates the cost per unit length of traversing edge e_i during tag tag_j and $G.L(e_i)$ is the length of edge e_i. To maintain the different costs on different edges during different traffic tags, function H maintains $|E| \cdot |TAGS|$ cost variables, denoted as $d_{(e_i,\ tag_j)}$ (where $1 \leqslant i \leqslant |E|$ and $1 \leqslant j \leqslant |TAGS|$).

We organize all the cost variables into a **cost vector** $\mathbf{d} \in \mathbb{R}^{(|E| \cdot |TAGS|)}$ and $\mathbf{d}$= $[d_{(e_1,\ tag_1)}, \ldots, d_{(e_{|E|},\ tag_1)}, d_{(e_1,\ tag_2)}, \ldots,$

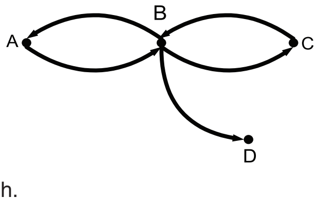


Fig. 3. Primal graph.

$d_{(e_{|E|},\ tag_2)}, \ldots, d_{(e_1,\ tag_{|TAGS|})}, \ldots, d_{(e_{|E|},\ tag_{|TAGS|})}]^{\mathbf{T}}$. The x-th element of the vector, i.e., $\mathbf{d}[x]$, equals $d_{(e_i,\ tag_j)}$ and $x = pos(i, j) = (j-1) \cdot |TAGS| + i$. Note that if the cost vector $\mathbf{d}$ becomes available, the function $G.H$ also becomes available.

The proposed model is attractive in our setting. It is simpler than existing models capable of capturing time-varying weights (e.g., time-expanded graphs [16] and time-aggregated graphs [17]), and yet it is sufficiently expressive for the problem we solve.

3.2 Trips and Trip Costs

Since vehicle tracking using GPS is widespread and growing, we take into account trips derived from GPS observations. A GPS trajectory $gpsTr = (gps_1, gps_2, \ldots, gps_n)$ is a sequence of GPS observations, where a GPS observation gps_i specifies the location of a vehicle at a particular time point. After map matching and some pre-processing, a GPS trajectory is transformed into a trip $t = (l_1, l_2, \ldots, l_m)$ that consists of a sequence of *link records* l_i of the form:

$$link\ record\ l_i\!:\ \ (e, t_s, t_e),$$

where $e \in \mathbb{E}$ indicates an edge in G and t_s and t_e indicate the time points of the first and last GPS observations on edge e_i.

If a graph G is available that contains relevant edge costs, the cost of a trip $t = (l_1, l_2, \ldots, l_m)$ can be estimated by Equation 1.

$$cost(t) = \sum_{l_i \in t} \sum_{tag_j \in TAGS} weight(l_i, tag_j) \cdot G.H(l_i.e, tag_j), \qquad (1)$$

where

$$weight(l_i, tag_j) = \frac{\sum_{I \in G.F^{-1}(tag_j)} |I \cap [l_i.t_s, l_i.t_e]|}{|[l_i.t_s, l_i.t_e]|}.$$

Here, $G.F^{-1}$ indicates the inverse function of F defined in G, which takes as input a traffic tag and outputs the set of its corresponding time intervals. Next, $|\cdot|$ denotes the length of an interval. For example, given a trip that contains link record $l_i = (e_j, 6{:}51, 7{:}05)$ and the traffic tags given in Section 3.1, the cost of the trip is $\frac{10}{15} \cdot G.H(e_j, OFFPEAK) + \frac{5}{15} \cdot G.H(e_j, PEAK) = \frac{10}{15} \cdot d_{(e_j, OFFPEAK)} \cdot G.L(e_j) + \frac{5}{15} \cdot d_{(e_j, PEAK)} \cdot G.L(e_j)$.

3.3 Framework Overview

Fig. 4 gives an overview of the framework for assigning trip cost based weights to a road network. Various types of raw data collected from a road network, such as GPS observations with corresponding CAN bus data and sensor data, are fed into a pre-processing module. While the GPS observations are obligatory, the CAN bus and sensor data are optional.

Pre-processing module: The GPS observations are map matched and transformed into trips as defined in Section 3.2. Next, a cost is associated with each trip. If only GPS observations are available, some costs, e.g., travel time, can be associated with trips directly. Other costs, e.g., GHG emissions, can be derived. For example, models are available in the literature that are able to provide an estimate of a trip's GHG emissions and fuel consumption based on the GPS observations of the trip [3]. If CAN bus data and sensor data are also available along with the GPS data, actual and more accurate fuel consumption and GHG emissions can be obtained directly, and thus can be associated with trips.

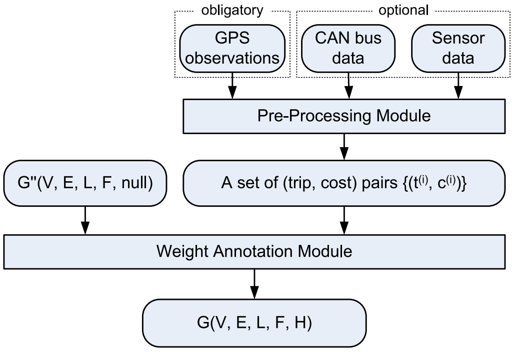


Fig. 4. Framework overview.

The pre-processing module outputs a set of (*trip*, *cost*) pairs $\{(t^{(i)}, c^{(i)})\}$, which then serve as input to the edge annotation module. For example, if the goal is to assign GHG emissions based weights, cost value $c^{(i)}$ indicates the GHG emissions of trip $t^{(i)}$. Note that the cost $c^{(i)}$ is the total cost associated with the i-th trip, meaning that the cost for each individual link record in the i-th trip is not required to be known. This makes it easier to collect (*trip*, *cost*) pairs. Because pairs may be obtained in wide variety of ways, the proposed framework has the potential for wide applicability.

Weight annotation module: The (*trip*, *cost*) pairs along with a corresponding un-weighted graph $G'' = (\mathbb{V}, \mathbb{E}, L, F, null)$ are fed into the weight annotation module. This module assigns pertinent weights to the edges of the graph, and it outputs an weighted graph $G = (\mathbb{V}, \mathbb{E}, L, F, H)$.

Recall that function $G.H$ from Section 3.1 is defined by the cost vector $\mathbf{d}$. Given a set of (*trip*, *cost*) pairs $\mathbb{TC} = \{(t^{(i)}, c^{(i)})\}$, the core task of this module is to estimate appropriate cost variables in vector $\mathbf{d}$. We formulate the weight annotation problem as a supervised learning problem, namely a regression problem [18] that employs $\mathbb{TC}$ as the training data set to estimate cost variables in vector $\mathbf{d}$.

The regression problem is solved by minimizing a judiciously designed objective function composed of three sub-objective items. The first item measures the misfit between the given actual cost and the estimated cost (i.e., the cost obtained from the cost model described in Equation 1) for every trip in $\mathbb{TC}$. The second item measures the differences between the cost variables of two edges whose expected traffic flows (based on topological structures) are similar. The third item measures the differences between the cost variables of two edges which are directionally adjacent. Further, other appropriate metrics that can quantify the difference between the cost variables of two edges can also be incorporated into the module. Finally, minimizing the objective function is handled by solving a system of linear equations.

4 Objective Functions

Since we regard the problem as a regression problem, we elaborate on the design of the proposed objective function and the solution to minimizing the objective function.

4.1 Residual Sum of Squares

In order to obtain an appropriate estimation of the cost vector $\mathbf{d}$, we need to make sure that for every (*trip*, *cost*) pair $(t^{(i)}, c^{(i)}) \in \mathbb{TC}$, the misfit between the actual cost (e.g., $c^{(i)}$) and the estimated cost (e.g., $cost(t^{(i)})$ evaluated by Equation 1, which employs $\mathbf{d}$), is as small as possible. To quantify the misfit, the residual sum of squares (*RSS*) function is applied, where

$$RSS(\mathbf{d}) = \sum_{(t^{(i)}, c^{(i)}) \in \mathbb{TC}} (c^{(i)} - cost(t^{(i)}))^2.$$

To facilitate the following discussion, we derive a matrix representation of the *RSS* function, as shown in Equation 2.

$$RSS(\mathbf{d}) = ||\mathbf{c} - \mathbf{Q^T d}||_2^2. \tag{2}$$

Let the cardinality of the set $\mathbb{TC}$ be N (i.e., $|\mathbb{TC}| = N$). We define a vector $\mathbf{c} \in \mathbb{R}^N = [c^{(1)}, c^{(2)}, \ldots, c^{(N)}]^\mathbf{T}$, where $c^{(i)}$ is the given actual cost of the trip $t^{(i)}$, and $(t^{(i)}, c^{(i)}) \in \mathbb{TC}$. A matrix $\mathbf{Q} \in \mathbb{R}^{|\mathbf{d}| \times N} = [\mathbf{q^{(1)}}, \mathbf{q^{(2)}}, \ldots, \mathbf{q^{(N)}}]$ is introduced to enable us to rephrase Equation 1 into a matrix representation. Specifically, $\mathbf{q^{(k)}}$ is the k-th column vector in $\mathbf{Q}$ which corresponds to trip $t^{(k)}$. If trip $t^{(k)}$ contains a link record l whose corresponding edge is e_i (i.e., $l.e = e_i$), then $\mathbf{q^{(k)}}[pos(i, j)] = G.L(e_i) \cdot weight(l, tag_j)$ where $1 \leqslant j \leqslant |TAGS|$; otherwise, it is set to 0.

Different from ordinary regression problems, minimizing Equation 2 is insufficient for determining every cost variable in $\mathbf{d}$ because the trips in $\mathbb{TC}$ may not cover all the edges in the road network, e.g., all the edges in $\mathbb{E}$. For the edges that are never traversed by any trip in $\mathbb{TC}$, their corresponding cost variables in $\mathbf{d}$ cannot be determined by only minimizing the *RSS* function.

In this case, annotating the edges that do not appear in $\mathbb{TC}$ with weights seems to be difficult and even unsolvable. In the following, we try to use the topology of the road network to further propagate and constrain the cost variables in order to assign an appropriate weight to every edge.

4.2 Topological Constraint

The topology of a road network is highly correlated with human movement flow [19], [20], including the movement of both pedestrians and vehicles. Edges with similar movement flows can be expected to have similar cost variables. Thus, if an edge is covered in $\mathbb{TC}$, its cost variable information can be propagated to the edges that have similar movement flows. To this end, we study how to quantify movement flow based similarity between edges using topological information of road networks.

4.2.1 Modeling Traffic Flows with PageRank

We transfer the idea of using PageRank for the modeling of web surfers to the modeling of vehicle movement in road networks. The original PageRank employs the hyperlink structure of the web to build a first-order Markov chain, where each web page corresponds to a state [21]. The Markov chain is governed by a transition probability matrix $\mathbf{M}$. If web page i has a hyperlink pointing to web page j then $\mathbf{M}[i, j]$ is set to $\frac{1}{outDegree(i)}$; otherwise, it is set to 0. $\mathbf{M}[i, j]$ indicates the probability of transition from state i to state j. PageRank models a user browsing the web as a Markov process based on matrix $\mathbf{M}$, and the final PageRank vector is the stationary distribution vector $\mathbf{x}$ of matrix $\mathbf{M}$. The PageRank of web page i, i.e., $\mathbf{x}[i]$, indicates the probability that the user visits page i or, equivalently, the fraction of time the user spends on page i in the long run [21].

The modeling movements of vehicles on a road network as stochastic processes is well studied in the transportation field [22]. In particular, the modeling of vehicle movements as Markov processes is an easy-to-use and effective approach [20]. Thus, we build a first-order Markov chain with a transition probability matrix derived from both the topology of the road network and the trips that occur in the road network. A state corresponds to an edge in the primal graph (i.e., a directed road segment), not a vertex (i.e., a road intersection).

The PageRank value of a state indicates the probability that a vehicle travels on the edge or, equivalently, the fraction of time a vehicle spends on the edge in the long run. Thus, the PageRank value is expected to reflect the traffic flow on the edge. Further, a series of topological metrics [19], including centrality-based metrics, small-world metrics, space-syntax metrics, and PageRank metrics, have been applied to capture human movement flows in urban environments. When using a graph representation of an urban environment, it is found that the classical and weighted PageRank metrics are highly correlated with human movements [19], [23]. Thus, if two edges have similar PageRank values, the traffic flow on the two segments should be similar.

When modeling web surfers, PageRank assumes that the Markov chain is time-homogeneous, meaning that the probability of transferring from page i to page j has the same fixed value at all times. In other words, matrix $\mathbf{M}$ is static across time. In contrast, the time-homogenous assumption does not hold for vehicles traveling in road networks. For example, during peak hours, the transition probability from edge i to edge j may be substantially different from the probability during off-peak hours. Thus, we maintain a distinct transition probability matrix $\mathbf{M}_k$ for each traffic category tag tag_k. During a particular traffic tag, we assume the Markov chain to be time-homogeneous.

4.2.2 PageRank on Dual Graphs

PageRank was originally proposed to assign prestige to web pages in a web graph, where web pages are modeled as vertices and the hyper-links between web pages are modeled as edges. Unlike the web graph, we are not interested in the prestige of vertices (i.e., road intersections) in the primal graph representation of a road network; rather, we are interested in the prestige of edges (i.e., directed road segments).

In order to assign PageRank values to edges, the primal graph $G = (\mathbb{V}, \mathbb{E}, L, F, H)$ is transformed into a dual graph $G' = (\mathbb{V}', \mathbb{E}')$, where each vertex in $\mathbb{V}'$ corresponds to an edge in the primal graph, and where each edge in $\mathbb{E}'$, denoted by a pair of vertices in $\mathbb{V}'$, corresponds to a vertex in the primal graph. Since functions L, F, and H are not of interest in this section, we do not keep them in the dual graph.

To avoid ambiguity, we use the terms edge and vertex when referring to primal graphs and use *dual edge* and *dual*

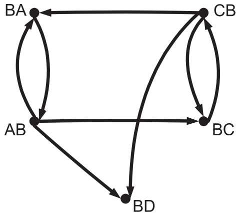


Fig. 5. Dual graph.

vertex when referring to dual graphs. Further, we use the term weight when referring to the weight of an edge in a primal graph, and we use *dual weight* in the context of dual edges in a dual graph.

We define a mapping $D2P : \mathbb{V}' \cup \mathbb{E}' \rightarrow \mathbb{V} \cup \mathbb{E}$ to record the correspondence between the elements in the dual and primal graphs. Fig. 5 show the dual graph that corresponds to the primal graph shown in Fig. 3. Since the dual vertex AB corresponds to the edge (A, B) in Fig. 3, $D2P(AB) = (A, B)$. Similarly, since the dual edge (CB, BA) corresponds to the vertex B in Fig. 3, $D2P((CB, BA)) = B$.

The dual graph is able to model an important characteristic of a road network: at a particular intersection, the probability of which segment a vehicle follows depends on the segment via which the vehicle entered the intersection. Considering the road network shown in Fig. 2, at intersection (i.e., vertex) B, a vehicle can proceed to follow segments (i.e., edges) (B, A), (B, C), or (B, D). If a vehicle entered the intersection using segment (C, B), it may be unlikely that the vehicle takes a u-turn to follow segment (B, C), while is more likely that it will use the other segments. Similar cases exist if a vehicle arrived at the intersection using segment (A, B).

Modeling this characteristic in a primal graph is not easy. For example, we need to maintain two sets of probabilities on edge (B, C), for the vehicles came from edge (C, B) versus edge (A, B). In contrast, modeling this in a dual graph is straightforward, as how a vehicle entered a particular intersection is clearly represented as a dual vertex. For example, the probabilities on dual edges (CB, BC) and (AB, BC) record the probabilities that a vehicle entered intersection B from edge (C, B) and edge (A, B), respectively, and continues along edge (B, C).

Given the dual graph $G' = (\mathbb{V}', \mathbb{E}')$, original PageRank values are defined formally as follows.

$$PR(v'_i) = \frac{1 - df}{|\mathbb{V}'|} + df \cdot \sum_{v'_j \in IN(v'_i)} \frac{PR(v'_j)}{|OUT(v'_j)|}, \quad v'_i \in \mathbb{V}', \tag{3}$$

where $PR(v'_i)$ indicates the PageRank value of dual vertex v'_i; $IN(v'_i)$ indicates the set of in-link neighbors of v'_i, i.e., $IN(v'_i) = \{v'_x | (v'_x, v'_i) \in \mathbb{E}'\}$; and $OUT(v'_j)$ indicates the set of out-link neighbors of v'_j, i.e., $OUT(v'_j) = \{v'_x | (v'_j, v'_x) \in \mathbb{E}'\}$. Further, $df \in [0, 1]$ is a damping factor, which is normally set to 0.85 for ranking a web graph.

The intuition behind Equation 3 is that the PageRank values are composed of two parts: jumping to another random vertex and continuing the random walk. This assumption works fine on the web graph, but we need to adapt this to the different characteristics of the graph representing a road network. In a road network, it is impossible for a vehicle to choose a random edge to traverse when at an intersection. Rather, it can only choose to continue along one of the out-link (dual) edges. Based on this observation, we set the damping factor df to 1. Some existing empirical studies [19] also suggest that with the damping factor set to 1, the resulting PageRank values have the best correlation with the human movement flows.

TABLE 2
Numbers of Trips Occurred on Dual Edges

Tags	(AB, BC)	(AB, BD)	(AB, BA)
PEAK	30	10	0
OFFPEAK	5	5	0

4.2.3 Weighted PageRank Computation

Definition of Dual Weights: In the original PageRank algorithm, a vertex propagates its PageRank value evenly to all its out-link neighbors. In other words, the dual weight for each dual edge from dual vertex v'_j is set uniformly to $\frac{1}{|OUT(v'_j)|}$. The uniform weights on the web graph indicate that a web surfer chooses its next target web page without any preferences to continue its random surfing. However, in a road network, such non-preference surfing usually does not occur. For example, the next step where a vehicle continues often depends on where the vehicle came from, as discussed in Section 4.2.2. Also, if *Avenue 1* and *Avenue 2* are the main roads in the road network shown in Fig. 2, more vehicles travel from AB to BC than from AB to BD. Further, during different traffic category tags, the transitions between dual vertices may also be quite different.

With the availability of very large collections of GPS data, we are able to capture the probability that a vehicle transits from one road segment to another at an intersection during different traffic category tags. Assume we only distinguish between peak and off-peak hours, i.e., there are only two corresponding tags in $TAGS$. Suppose we obtain the number of trips occurred on dual edges, as shown in Table 2.

For example, among all the trips that occurred on dual vertex AB during the peak hours, 30 trips proceeded to follow BC, and 10 trips followed BD; during off-peak hours, 5 trips followed BC, and 5 trips followed BD. These observations suggest that the dual weight on dual edge (AB, BC) should be greater than the dual weight on dual edge (AB, BD) during peak hours; while they should be the same during off-peak hours.

As the dual graph has different dual weights for different traffic tags, we need to maintain a dual graph for each traffic tag. Specifically, the training data set $\mathbb{TC}$ is partitioned into $\mathbb{TC}_1, \mathbb{TC}_2, \ldots, \mathbb{TC}_{|TAGS|}$ according to the traversal times. Partition $\mathbb{TC}_k$ consists only of the trips that are occurred during the time period indicated by the traffic tag tag_k, i.e., $G.F^{-1}(tag_k)$.

The dual weight of a dual edge (v'_i, v'_j) during tag tag_k is related to the ratio of the number of trips that traversed the dual vertices v'_i and v'_j to the number of trips that traversed the dual vertex v'_i, during tag tag_k. Further, to contend with data sparsity, Laplace smoothing is applied to smooth the dual weight values for the dual edges that are not covered by any trip in $\mathbb{TC}$. The dual weight of dual edge (v'_i, v'_j) for the dual graph within tag_k (denoted as G'_k) is computed based on Equation 4.

$$W_k(v'_i, v'_j) = \frac{|Trip_k(v'_i, v'_j)| + 1}{\sum_{v'_x \in OUT(v'_i)} |Trip_k(v'_i, v'_x)| + |OUT(v'_i)|}, \quad (4)$$

where $Trip_k(v'_i, v'_j)$ returns the set of trips in partition $\mathbb{TC}_k$ that traversed the dual vertices v'_i and v'_j.

Continuing the example shown in Table 2, although no trip goes from the dual vertex AB directly back to BA in $\mathbb{TC}$, this does not mean that such a trip will not occur in the future. Thus, we need to give a small, non-zero value to the dual weight of dual edge (AB, BA). Using the dual weights provided by Equation 4, the dual weights of the out-linking dual edges of dual vertex AB are: $W_{PEAK}(AB, BC) = \frac{31}{43}$, $W_{PEAK}(AB, BD) = \frac{11}{43}$, and $W_{PEAK}(AB, BA) = \frac{1}{43}$; and $W_{OFFPEAK}(AB, BC) = \frac{6}{13}$, $W_{OFFPEAK}(AB, BD) = \frac{6}{13}$, and $W_{OFFPEAK}(AB, BA) = \frac{1}{13}$.

Note that for a given dual vertex v'_i, if no trips in $\mathbb{TC}$ are available to assign the dual weights during a traffic tag tag_k, i.e., $|Trip_k(v'_i, v'_x)| = 0$ for every $v'_x \in OUT(v'_i)$, Equation 4 assigns weights with $\frac{1}{|OUT(v'_i)|}$ to each dual edge, which is exactly what the original PageRank algorithm does. For instance, if no trips are available for dual vertex AB (i.e., if the numbers in Table 2 are all zeros), the dual weights for $W_k(AB, BC)$, $W_k(AB, BD)$, and $W_k(AB, BA)$ are all $\frac{1}{3}$.

Computing Weighted PageRank Values: Based on the dual weights obtained from Equation 4, we construct the transition probability matrices $\mathbf{M_k} \in \mathbb{R}^{|\mathbb{V}'| \times |\mathbb{V}'|}$. Specifically, the ith row and jth column element in $\mathbf{M_k}$, i.e., $\mathbf{M_k}[i, j]$, equals $W_k(v'_i, v'_j)$ if the dual edge (v'_i, v'_j) exists in the dual graph; otherwise, it equals 0. Note that the sum of all elements in a row equals 1, i.e., $\sum_{j=1}^{|\mathbb{V}'|} \mathbf{M_k}[i, j] = 1$ for every $1 \leqslant i \leqslant |\mathbb{V}'|$.

Let vector $\mathbf{v_k} \in \mathbb{R}^{|\mathbb{V}'|}$ record the PageRank values for every dual vertex in G'_k. Specifically, $\mathbf{v_k}[i] = PR_k(v'_i)$, which is the PageRank value of v'_i during traffic category tag tag_k. This way, the PageRank values can be computed iteratively as follows until converged.

$$\mathbf{v_k}^{(n+1)} = \mathbf{M_k}^{\mathbf{T}} \cdot \mathbf{v_k}^{(n)},$$

where $\mathbf{v_k}^{(n)}$ is the PageRank vector in the n-th iteration.

4.2.4 PageRank-Based Topological Constraint Objective Function

After obtaining the weighted PageRank values for every dual edge, the topological similarity between two edges in the primal graph is quantified in Equation 5.

$$S_k^{PR}(e_i, e_j) = \frac{\min(PR_k(v'_{e_i}), PR_k(v'_{e_j}))}{\max(PR_k(v'_{e_i}), PR_k(v'_{e_j}))}. \quad (5)$$

The topological similarity between edges e_i and e_j, denoted as $S_k^{PR}(e_i, e_j)$, is defined based on the weighted PageRank values of the two dual vertices representing the edges. To be specific, v'_{e_i} and v'_{e_j} indicate the corresponding dual vertices of edges e_i and e_j, i.e., $D2P(v'_{e_i}) = e_i$ and $D2P(v'_{e_j}) = e_j$. Note that Equation 5 returns a high similarity if two edges have similar weighted PageRank scores and that it returns a low similarity, otherwise.

Based on the topological similarity, a PageRank-based Topological Constraint (*PRTC*) function is incorporated into the overall objective function. The intuition behind the *PRTC* function is that for the same traffic category tag, if two edges have similar traffic flows (as measured by Equation 5), their cost variables tend to be similar as well. The *PRTC* function is defined in Equation 6.

$$PRTC(\mathbf{d}) = \sum_{k=1}^{|TAGS|} PRTC(\mathbf{d}, k), \quad (6)$$

where

$$PRTC(\mathbf{d}, k) = \sum_{i,j=1}^{|G.\mathbb{E}|} S_k^{PR}(e_i, e_j) \cdot (d_{(e_i, tag_k)} - d_{(e_j, tag_k)})^2.$$

The value of the *PRTC* function over the cost vector $\mathbf{d}$ is the sum of $PRTC(\mathbf{d}, k)$ for every $1 \leqslant k \leqslant |TAGS|$. The function $PRTC(\mathbf{d}, k)$ computes the weighted (decided by S_k^{PR}) sum of the squared differences of between each pair of road segments' cost variables during traffic tag tag_k.

The *PRTC* function has two important features: (i) if the PageRank values of two edges are similar, the similarity value S_k^{PR} is large, thus making the difference between their cost variables obvious; (ii) if two edges' PageRank values are dissimilar, the similarity value S_k^{PR} with a small value smoothes down the difference between their cost variables. This way, minimizing the *PRTC* function corresponds to minimizing the overall difference between two cost variables whose corresponding road segments have similar traffic flows.

To obtain the matrix representation of the *PRTC* function, we introduce a matrix $\mathbf{A} \in \mathbb{R}^{|\mathbf{d}| \times |\mathbf{d}|}$, which is a block diagonal matrix.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} & & & \\ & \mathbf{A_2} & & \\ & & \cdots & \\ & & & \mathbf{A}_{|TAGS|} \end{bmatrix}, \quad (7)$$

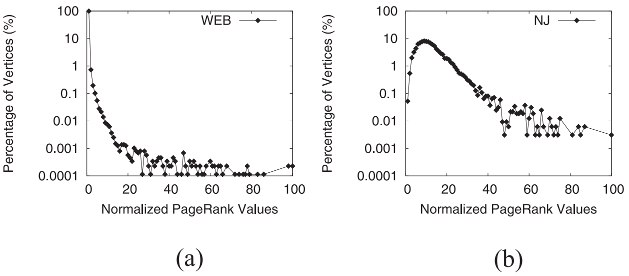
where $\mathbf{A_k} \in \mathbb{R}^{|E| \times |E|}$ and $\mathbf{A_k}[i, j] = S_k^{PR}(e_i, e_j)$, which obviously is a symmetric matrix. Let matrix $\mathbf{L_A}$ be the graph Laplacian induced by the similarity matrix $\mathbf{A}$. Specifically, $\mathbf{L_A}[i, j] = \delta_{i,j} \cdot \sum_x \mathbf{A}[i, x] - \mathbf{A}[i, j]$, where $\delta_{i,j}$ returns 1 if i equals j, and 0 otherwise. The matrix representation of *PRTC* function is shown in Equation 8.

$$PRTC(\mathbf{d}) = \mathbf{d}^{\mathbf{T}} \mathbf{L_A} \mathbf{d}. \quad (8)$$

4.2.5 Properties of PageRank on Road Networks

Web graphs and road network graphs are quite different, rendering it of interest to study the distributions of PageRank values on the two kinds of graphs. Fig. 6 shows the normalized (to (1, 100]) PageRank values with respect to the percentage of vertices having the PageRank values, on a graph (*WEB*) representing a part of the Web[1] and a dual graph (*NJ*) representing the road network of North Jutland, Denmark.

[1] http://snap.stanford.edu/data/web-Google.html



(a) (b)

Fig. 6. PageRank on the web and a road network. (a) Web. (b) Road networks.

Fig. 6 suggests that PageRank values on *NJ* are distributed more uniformly than for *WEB*. With this type of distribution, many vertices have the same or very similar high PageRank values, which renders the distribution ineffective for ranking when compared to *WEB*. However, the distribution is effective for our objective of identifying road segments with similar traffic flows based on PageRank values.

4.3 Adjacency Constraint

The *PRTC* function is derived from the overall structure of the road network. In this section, we consider a finer-grained topological aspect of the road network, namely, **directional adjacency**.

An important feature of a road network is that an event at one road segment may propagate to influence adjacent road segments. Consider a typical event in a road network, e.g., traffic congestion. If congestion occurs on road segment (A, B) in Fig. 2, road segment (B, C) may also experience congestion, or at least the traffic on (B, C) is affected by the congestion that occurs on (A, B). Thus, the cost variables of two directionally adjacent road segments should be similar.

The directional adjacency we discus here is represented clearly in the dual graph. If and only if two dual vertices are connected by an dual edge in the dual graph, the two corresponding road segments are directionally adjacent. For example, although edges (B, D) and (B, C) (in Fig. 3) intersect, their cost variables may not necessarily tend to be similar because no vehicle can travel between these two edges. Directional adjacency is distinct from the "non-directional" adjacency considered in previous work [7].

Another point worth noting is that if two road segments represent opposite directions of the same physical road segment, they are not directionally adjacent. It is natural that an event on a physical road only yields congestion in one direction, but not both directions. Considering the edges (A, B) and (B, A) (in Fig. 3), their corresponding vertices in the dual graph (AB and BA in Fig. 5) are connected by two edges, however, their cost variables are not necessarily similar.

Directional adjacency is also temporally sensitive. For example, although edges (A, B) and (B, C) are directionally adjacent, the general traffic situation (indicated by the cost variable) on edge (A, B) during peak hours is not necessarily correlated with the traffic on edge (B, C) during non-peak hours.

To incorporate directional adjacency, we incorporate a Directionally Adjacent Temporal Constraint (*DATC*) function into the overall objective function.

$$DATC(\mathbf{d}) = \sum_{k=1}^{k=|TAGS|} DATC(\mathbf{d}, k), \tag{9}$$

where

$$DATC(\mathbf{d}, k) = \sum_{i,j=1}^{|G.\mathbb{E}|} W'_k(v'_{e_i}, v'_{e_j}) \cdot (d_{(e_i, tag_k)} - d_{(e_j, tag_k)})^2,$$

and where v'_{e_i} and v'_{e_j} have the same meaning as in Equation 5. $W'_k(v'_{e_i}, v'_{e_j})$ is as defined in Equation 4 if v'_{e_i} and v'_{e_j} do not indicate the same physical road segment; and $W'_k(v'_{e_i}, v'_{e_j})$ equals 0 otherwise. For instance, although $W_{PEAK}(AB, BA) = \frac{1}{43}$ as discussed in Section 4.2.3, $W'_{PEAK}(AB, BA) = 0$ since AB and BA indicate the same physical road segment, *Avenue* 1.

The *DATC* function aims to make the cost variables satisfy the following property: given road segments e_i and e_j, if a many of the trips that follow e_i also follow e_j, as indicated by $W'_k(v'_{e_i}, v'_{e_j})$, the cost variables on the two edges tend to be more correlated.

Similar to the discussion in Section 4.2.4, we introduce a block diagonal matrix $\mathbf{B} \in \mathbb{R}^{|\mathbf{d}| \times |\mathbf{d}|}$ with the same format as matrix $\mathbf{A}$ (defined in Equation 7). In particular, in each block matrix, $\mathbf{B_k}[i, j] = \max(W'_k(v'_{e_i}, v'_{e_j}), W'_k(v'_{e_i}, v'_{e_j}))$, which guarantees that matrix $\mathbf{B_k}$, and hence matrix $\mathbf{B}$, are symmetric. Note that it is not possible that both $W'_k(v'_{e_i}, v'_{e_j})$ and $W'_k(v'_{e_j}, v'_{e_i})$ are non-zero because if edge $D2P(v'_{e_i})$ is directionally adjacent to edge $D2P(v'_{e_j})$ then edge $D2P(v'_{e_j})$ cannot be directionally adjacent to edge $D2P(v'_{e_i})$. Let $\mathbf{L_B}$ to be the graph Laplacian derived by matrix $\mathbf{B}$. The *DATC* function is represented by Equation 10.

$$DATC(\mathbf{d}) = \mathbf{d^T L_B d} \tag{10}$$

4.4 Solving the Problem

Combining the three individual objective functions and a classical *L2* regularizer, we obtain the overall objective function $O(\mathbf{d})$:

$$O(\mathbf{d}) = RSS(\mathbf{d}) + \alpha \cdot PRTC(\mathbf{d}) + \beta \cdot DATC(\mathbf{d}) + \gamma \cdot ||\mathbf{d}||_2^2,$$

where α, β, and γ are hyper-parameters that control the tradeoff among the losses on *RSS*, *PRTC*, *DATC*, and the *L2* regularizer. The matrix representation of the objective function is shown in Equation 11.

$$O(\mathbf{d}) = ||\mathbf{c} - \mathbf{Q^T d}||_2^2 + \alpha \cdot \mathbf{d^T L_A d} + \beta \cdot \mathbf{d^T L_B d} + \gamma \cdot ||\mathbf{d}||_2^2. \tag{11}$$

By differentiating Equation 11 w.r.t. vector $\mathbf{d}$ and setting it to 0, we get

$$[\mathbf{QQ^T} + \alpha \cdot \mathbf{L_A} + \beta \cdot \mathbf{L_B} + \gamma \cdot \mathbf{I}]\mathbf{d} = \mathbf{Qc}. \tag{12}$$

The solution to Equation 12 is the optimal solution to the cost vector, denoted as $\widehat{\mathbf{d}}$, that minimizes the overall objective function in Equation 11. The linear system in

Equation 12 can be solved efficiently by several iterative algorithms such as the conjugate gradient algorithm [24].

Finally, feeding the optimized cost variable vector $\widehat{\mathbf{d}}$ to function *G.H*, the time varying weights of the graph become available.

4.5 Discussion

In addition to the topology of a road network, other aspects of edges may be useful for identifying similarities among edges, e.g., the shapes and capacities of edges and the points of interest along edges [25]. Such information is not always available in digital maps and can be difficult to obtain. However, it is of interest to extend the proposed methods to take additional information, when available, into account. To achieve general applicability of the paper's methods, we minimize the requirements of the input graph G'': both *PRTC* and *DATC* rely solely on the topology of a road network, which can be obtained easily from any digital map.

The weight annotation problem is finally handled by solving a system of linear equations, i.e., Equation 12. Alternative edge similarity metrics (e.g., considering the shapes and capacities of edges) can be easily incorporated into the linear system by adding new terms of the form $\varphi \cdot \mathbf{L_M}$, where φ is the hyper-parameter and $\mathbf{L_M}$ is the Laplacian matrix derived by an alternative similarity metric. An alternative similarity metric *sim* should satisfy symmetry: $sim(e_i, e_j) = sim(e_j, e_i)$. Both *PRTC* and *DATC* satisfy symmetry.

The core operations in solving a system of linear equations using a conjugate gradient algorithm are matrix multiplication and transposition. This means that existing scalable matrix computation algorithms [26], [27] can be applied directly to make the proposed framework scalable and applicable to large road networks.

5 Experimental Study

We study the effectiveness of the proposed method for weight annotation of road networks with both *travel time* (*TTWA*) and *GHG emissions* (*GEWA*).

5.1 Experimental Setup

Road Networks: We use two road networks. The SK network is from Skagen, Denmark and has a primal graph with 543 vertices and $1,244$ edges. The NJ network contains almost all of North Jutland, Denmark and has a primal graph with $17,956$ vertices and $39,372$ edges.

Trips: We use GPS observations collected from 28 vehicles in the period 2007-10-01 to 2007-10-15. When the vehicles were moving, positions were sampled at 1 Hz. The data is collected as part of an experiment where young drivers start out with a substantial rebate on their car insurance and then are warned if they exceed the speed limit and are penalized financially if they continue to speed.

We apply an existing tool for map matching GPS observations onto road segments, thus obtaining 431 trips in the SK network and $11,516$ trips in the NJ network.

For *TTWA*, we use the total travel time for each trip, which can be obtained directly from the GPS observations of the trip, as the cost.

TABLE 3
Traffic Category Tag Function *G.F*

Periods		Tags
Weekdays	[0:00, 7:00)	*OFFPEAK*
Weekdays	[7:00, 8:00)	*PEAK*
Weekdays	[8:00, 15:00)	*OFFPEAK*
Weekdays	[15:00, 17:00)	*PEAK*
Weekdays	[17:00, 24:00)	*OFFPEAK*
Weekends	[0:00, 24:00)	*WEEKENDS*

For *GEWA*, we use the GHG emissions of each trip as trip cost. Ideally, the exact fuel consumption should be obtained from CAN bus sensor data. Since such data is hard to obtain in a scalable fashion, we use instead the VT-micro model [15] that is able to compute the GHG emissions of trips based on the instantaneous velocities and accelerations derived from the GPS records of the trips in a robust fashion [3]. The 1 Hz GPS sampling frequency makes the VT-Micro model easy to use.

Traffic Category Tags: In transporation research, *PEAK* and *OFFPEAK* periods are used widely to distinguish different traffic flows over the course of a day [28]. Thus, we use *PEAK* and *OFFPEAK* as traffic category tags. Further, we distinguish between weekdays from weekend days, as traffic differs between weekdays and weekend days. To appropriately assign *PEAK* and *OFFPEAK* tags to the data set, we plot the numbers of GPS records according to their corresponding observed time at an one-hour granularity for weekdays and weekend days, respectively. Based on the generated histograms, we identify *PEAK* and *OFFPEAK* periods for weekdays. We find no clear peak periods during weekends and thus use *WEEKENDS* as the single tag for weekends. Table 3 provides the mapping (i.e., the function *G.F*) from time periods to tags.

T-Drive [10] is able to assign distinct and fine-grained traffic tags to individual edges. The precondition of the method is that sufficient GPS data is associated with edges. However, a substantial fraction of all edges have no GPS data in our setting. Thus, we use traffic tags at the coarse granularity shown in Table 3.

Implementation Details: The PageRank computation is implemented in C using the iGraph library version 0.5.4 [29]. All remaining experiments are implemented in Java, where the conjugate gradient algorithm for solving a linear system is implemented using the MTJ (matrix-toolkits-java) package [30].

We use the threshold 0.95 to filter the entries in the PageRank-based similarity matrix $\mathbf{A}$ (Equation 7): if the value of an entry in $\mathbf{A}$ is smaller than 0.95, the entry is set to 0. We use the speed limits associated with roads to classify the edges into two categories, *highways* (with speed limits above 90 km/h) and *urban roads* (with speed limits below 90 km/h). We only apply adjacency constraint on pairs of edges in the same category.

Due to the space limitation, the experiments only report the results using the best set of hyper-parameters, which are is obtained by manual tuning on a separate data set using cross validation. This is a well known method [18] for choosing hyper-parameters.

TABLE 4
Effectiveness on *TTWA*

	SSL_{F_1}	$Ratio_{F_2}$	$Ratio_{F_3}$	$Ratio_{F_4}$
SK	88,656	99.2%	44.0%	43.8%
NJ	14,823,752	92.2%	49.2%	43.1%

TABLE 5
Coverage of Weight Annotation

	$Cove_{F_1}$	$Cove_{F_2}$	$Cove_{F_3}$	$Cove_{F_4}$
SK	22.8%	28.8%	100%	100%
NJ	34.8%	86.7%	99.6%	100%

5.2 Experimental Results

5.2.1 Effectiveness Measurements

To gain insight into the accuracy of the obtained trip cost based weights, we split the set of (*trip*, *cost*) pairs into a training set $\mathbb{TC}_{train}$ and a testing set $\mathbb{TC}_{test}$. We use the training set to annotate the spatial network with weights, and we use the testing set to evaluate the accuracy of the weights. In the following experiments, we randomly choose 50% of the pairs for training and the remaining 50% for testing, unless explicitly stated otherwise.

Since no ground-truth time-dependent weights exist for the two road networks, the accuracy of the obtained weights can only be evaluated using the trips in testing set $\mathbb{TC}_{test}$. If the obtained weights (using $\mathbb{TC}_{train}$) actually reflect the travel costs, the difference between the actual cost and the estimated cost using the obtained weights (i.e., by using Equation 1 defined in Section 3.2) for each trip in the testing set $\mathbb{TC}_{test}$ should be small.

We use the sum of squared loss (*SSL*) value (defined in Equation 13) between the actual cost $c^{(i)}$ and the estimated cost $cost(t^{(i)})$ over every trip in the testing set $\mathbb{TC}_{test}$ to measure the accuracy of the obtained weights.

$$SSL(\mathbb{TC}_{test}) = \sum_{(t^{(i)},c^{(i)})\in\mathbb{TC}_{test}} (c^{(i)} - cost(t^{(i)}))^2 \tag{13}$$

For example, if the GHG emissions based weights really reflect the actual GHG emissions, the sum of squared loss between the actual GHG emissions and the estimated GHG emissions over every testing trip should tend to be small. The smaller the sum of squared loss, the more accurate the weights.

To gain insight into the effectiveness of the proposed objective functions, we compare four combinations of the functions:

1) $F_1 = RSS(\mathbf{d}) + \gamma \cdot ||\mathbf{d}||_2^2$.
2) $F_2 = RSS(\mathbf{d}) + \alpha \cdot PRTC(\mathbf{d}) + \gamma \cdot ||\mathbf{d}||_2^2$.
3) $F_3 = RSS(\mathbf{d}) + \beta \cdot DATC(\mathbf{d}) + \gamma \cdot ||\mathbf{d}||_2^2$.
4) $F_4 = RSS(\mathbf{d}) + \alpha \cdot PRTC(\mathbf{d}) + \beta \cdot DATC(\mathbf{d}) + \gamma \cdot ||\mathbf{d}||_2^2$.

Function F_1 only considers the residual sum of squares. Functions F_2 and F_3 take into account the PageRank-based topological constraint and the directional adjacency constraint, respectively. Function F_4 takes into account both constraints.

As the objective function used in trajectory regression [7] also considers adjacency, we can view the method using function F_3 as an improved version of trajectory regression because (i) function F_3 works not only for travel times, but also other travel costs, e.g., GHG emissions; (ii) function F_3 considers the temporal variations of travel costs, while trajectory regression does not; and (iii) function F_3 considers directional adjacency, while trajectory regression models a road network as a undirected graph and only considers undirected adjacency.

The sum of squared loss value for using objective function F_i is denoted as $SSL_{F_i}(\mathbb{TC}_{test})$. In order to show the relative effectiveness of the proposed objective functions, we report the ratios $Ratio_{F_2} = \frac{SSL_{F_2}(\mathbb{TC}_{test})}{SSL_{F_1}(\mathbb{TC}_{test})}$, $Ratio_{F_3} = \frac{SSL_{F_3}(\mathbb{TC}_{test})}{SSL_{F_1}(\mathbb{TC}_{test})}$, and $Ratio_{F_4} = \frac{SSL_{F_4}(\mathbb{TC}_{test})}{SSL_{F_1}(\mathbb{TC}_{test})}$.

Coverage, defined in Equation 14, is introduced as another measurement.

$$Cove_{F_i}(\mathbb{TC}_{train}) = \frac{|\{e|e \in G.\mathbb{E} \wedge annotated(e)\}|}{|G.\mathbb{E}|}, \tag{14}$$

where $annotated(e)$ holds if edge e is annotated with weights using $\mathbb{TC}_{train}$. Function $Cove_{F_i}$ indicates the ratio of the number of edges whose weights have been annotated by using objective function F_i to the total number of edges in the road network. The higher the coverage is, the more edges in the road network are annotated with weights, and thus the better performance.

5.2.2 Travel Time Based Weight Annotation

Effectiveness of objective functions: Table 4 reports the results on travel time based weight annotation. Column SSL_{F_1} reports the absolute SSL values over all test trips when using objective function F_1 for both data sets. NJ has much larger SSL values than SK because it has much more testing trips. For both road networks, the weights annotated using objective function F_4 have the least SSL values.

We also observe that the PageRank based topological constraint works more effectively on NJ than on SK. The reason is that Skagen is a small town in which few road segments have similar topology (e.g., similar weighted PageRank values). In the NJ network, the PageRank based topological constraint gives a better accuracy improvement since more road segments have similarly weighted PageRank values.

The coverage reported in Table 5 also justifies the observation. When using objective function F_1, only the edges in the set of training trips can be annotated, which can be expected to be a small portion of the road network. When using objective function F_2, the coverage of the SK network increases much less than for the NJ network. This suggests that in a large road network, the PageRank based topological constraint substantially increases the coverage of the annotation, thus improving the overall annotation accuracy.

The directed adjacency topological constraint yields similar accuracy improvements on both road networks, and the accuracy improvement is more substantial than the improvement given by the PageRank based topological constraint. This is as expected because a road network is fully connected, and *DATC* is able to finally affect almost every edge, which gives more information for the edges that

TABLE 6
Comparison with Baselines on *TTWA*

	$Ratio_{\lambda=2}$	$Ratio_{\lambda=1}$
SK	36.0%	78.8%
NJ	24.2%	90.8%

are not traversed by trips in the training set. This can be observed from the third column of Table 5.

For both road networks, *PRTC* and *DATC* together give the best accuracy, as shown in column $Ratio_{F_4}$ in Table 4. This finding offers evidence of the overall effectiveness of the proposed objective functions.

Accuracy comparison with a baseline: The test tips contain edges that are not covered by any training trips. Therefore, existing methods [10] that can estimate travel time based on historical data are inapplicable as baseline.

If the speed limit of every edge in a road network is available, we can use speed limit derived weights as a baseline for travel time based weight annotation. While it is difficult to obtain a speed limit for every road segment in a road network, we can use default values were values are missing. In the NJ network, 62 edges lack a speed limit and are assigned a default value (50 km/h).

Given an edge e and its speed limit $sl(e)$ and length $G.L(e)$, the corresponding travel time based weight for e is $\lambda \cdot \frac{G.L(e)}{sl(e)}$ if e is an urban road (where $\lambda \geq 1$) and $\frac{G.L(e)}{sl(e)}$ if e is a highway.

The factor λ is used because vehicles tend to travel at speeds below the speed limit on urban roads and at the speed limit on highways. Previous work [7] uses $\lambda = 2$, meaning that vehicles normally travel at half the speed limit in urban regions. However, we find that $\lambda = 1$ works the best for our data. The reason may be two-fold: (i) the data we use is collected from young drivers who tend to drive more aggressively than average drivers. (ii) the SK and NJ networks are relatively congestion-free when compared to Kyoto, Japan, which is simulated in previous work [7].

The above allows us to treat the speed limit derived weights as a baseline method for travel time based weight annotation. To observe the accuracy of the baseline method, its accuracy is also evaluated using *SSL* over every testing trip. Specifically, the baseline with $\lambda = 2$ is denoted as $SSL_{BL,\lambda=2}(\mathbb{TC}_{test})$, and the baseline with $\lambda = 1$ is denoted as $SSL_{BL,\lambda=1}(\mathbb{TC}_{test})$. The two resulting baselines are compared with the proposed method, and the results are reported in Table 6, where $Ratio_{\lambda=2} = \frac{SSL_{F_4}(\mathbb{TC}_{test})}{SSL_{BL,\lambda=2}(\mathbb{TC}_{test})}$ and $Ratio_{\lambda=1} = \frac{SSL_{F_4}(\mathbb{TC}_{test})}{SSL_{BL,\lambda=1}(\mathbb{TC}_{test})}$. The ratios $Ratio_{\lambda=1}$ on the two road networks show that the weights obtained by our method are substantially better than the best cases of the weight obtained from the speed limits.

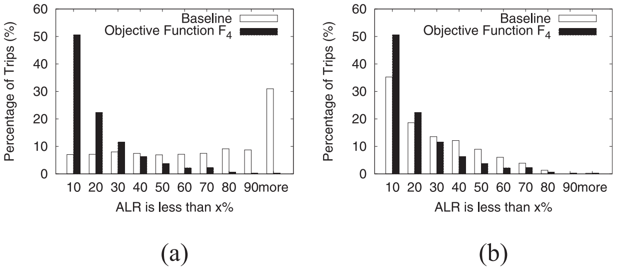


Fig. 7. *ALR* comparison on *TTWA* of NJ. (a) Baseline with $\lambda = 2$. (b) Baseline with $\lambda = 1$.

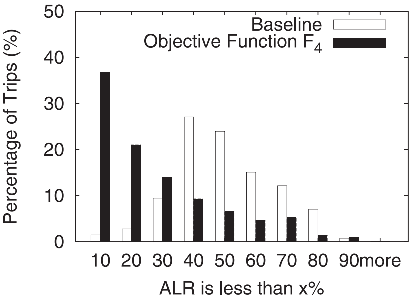


Fig. 8. *ALR* Comparison on *GEWA* of NJ.

The same deviation has quite a different meaning for long versus short trips. For example, a 50-second deviation can be considered as a very good estimation error for a 30-minute trip, while it is a poor estimation error for a 2-minute trip. Thus, to better understand how the overall *SSL* values are distributed, we plot the number of test trips whose *absolute loss ratio* (*ALR*) values are within x percentage in Fig. 7. Given a test pair $(t^{(i)}, c^{(i)}) \in \mathbb{TC}_{test}$, its *ALR* value equals the absolute difference between the estimated and actual costs divided by the actual cost, as defined in Equation 15.

$$ALR((t^{(i)}, c^{(i)})) = \frac{absolute(cost(t^{(i)}) - c^{(i)})}{c^{(i)}}. \qquad (15)$$

Our method shows the best result as the majority of the test trips have smaller *ALR* values. Assume that we consider and *ALR* below 30% as a good estimation. Fig. 7 shows that 84.3% of test trips have good estimations using the proposed method. In contrast, only 67.4% and 22.1% of test trips have good estimations using baseline methods with $\lambda = 1$ and $\lambda = 2$, respectively.

We do not integrate speed limits into our method because (i) for edges without available speed limits, the obtained weights are quite sensitive to the assigned default speed limits: inaccurate defaults deteriorate the performance severely; and (ii) speed limits do not give obvious benefits when annotating edges with GHG emissions based weights, as we will see shortly in Section 5.2.3 (in particular, in Fig. 8).

5.2.3 GHG Emissions Based Weight Annotation

Effectiveness of objective functions: Table 7 reports the results on GHG emissions based weight annotation. In general, the results are consistent with the results from the travel time based weight annotation (as shown in Table 4): (i) The PageRank-based topological constraint works more effectively on the NJ network than on the SK network; (ii) the directed adjacency constraint works more effectively

TABLE 7
Effectiveness on *GEWA*

	SSL_{F_1}	$Ratio_{F_2}$	$Ratio_{F_3}$	$Ratio_{F_4}$
SK	175.931	99.9%	40.3%	30.0%
NJ	87,362,465	94.5%	66.2%	44.3%

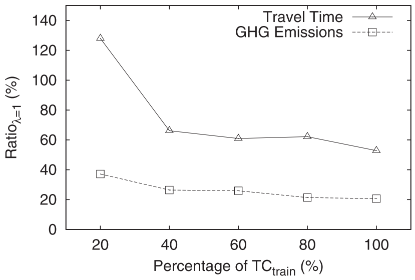


Fig. 9. Results on different size of $\mathbb{TC}_{train}$.

than the PageRank-based topological constraint; (iii) the weights obtained by using both *PRTC* and *DATC* give the best accuracy. The coverage when using the different objective functions is exactly the same as what was reported in Table 5.

Comparison with a baseline: As we did for travel times, we use speed limits to devise a baseline for GHG emissions based weight annotation. Assuming a vehicle travels on an edge at constant speed (e.g., the speed limit of the edge), we can simulate a sequence of instantaneous velocities. For example, let an edge be 100 meters long and the speed limit be 60 km/h. The simulated trip on the road segment is represented by a sequence of 6 records, each with 60 km/h as the instantaneous velocity. This allows us to apply the VT-micro model to estimate GHG emissions based edge weights. Since in the previous set of experiments, we have already found that the speed limit (i.e., $\lambda = 1$) is the best fit for our data we simply use the speed limit here.

We obtain $Ratio_{\lambda=1} = 24.7\%$ for SK and $Ratio_{\lambda=1} = 29.8\%$ on NJ. Fig. 8 shows the percentage of test trips whose *ALR* values are less than x% using the baseline with $\lambda = 1$ and the proposed method, respectively. These results clearly show the better performance of the proposed method, as the majority of test trips have smaller *ALR* values.

5.2.4 Effectiveness of the Size of Training Trips

In this section, we study the accuracy when varying the training set size. Specifically, on the NJ network, we reserve 20% of the (*trip*, *cost*) pairs as the testing set, denoted as $\mathbb{TC}_{test}$, and the remaining 80% as the training set, denoted as $\mathbb{TC}_{train}$. In order to observe the accuracy of weight annotation on different sizes of $\mathbb{TC}_{train}$, we use 100%, 80%, 60%, 40%, and 20% of $\mathbb{TC}_{train}$ to annotate the weights, respectively. The results are shown in Fig. 9.

For travel time, when only 20% of $\mathbb{TC}_{train}$ is used, the accuracy of our method is worse than the baseline method with $\lambda = 1$ because the baseline has a rough estimation for the costs of all edges, while the 20% of $\mathbb{TC}_{train}$ covers only 16.3% of the edges in the road network. Although our method propagates weights to edges that are not covered by the training trips, the accuracy suffers when the initial coverage of the training trips is low. When 40% of $\mathbb{TC}_{train}$ is used, the accuracy of our method is much better than that of the baseline. In this case, the training trips cover 23.3% of all edges. As the training set size increases, the accuracy of the travel time weights also increases. When we use all trips in $\mathbb{TC}_{train}$, the accuracy of our method is almost twice that of the baseline.

For GHG emissions, we observe a similar trend: with more training trips, the accuracy of the corresponding weights improves, and our method always outperforms the baseline when annotating edges with GHG emissions based weights.

This experiment justifies that (i) our method works effectively even when the coverage of the trips in the training set is low; (ii) if the coverage of the trips in the training set increases, e.g., by providing more (*trip*, *cost*) pairs as training set, the accuracy of the obtained weights also increases.

6 Conclusion and Outlook

Reduction in GHG emissions from transportation calls for effective eco-routing, and road network graphs where all edges are annotated with accurate weights that capture environmental costs, e.g., fuel usage or GHG emissions, are needed for eco-routing. However, such weights are not always readily available for a road network. This paper proposes a general framework that takes as input a collection of (*trip*, *cost*) pairs and assigns trip cost based weights to a graph representing a road network, where trip cost based weights may reflect GHG emissions, fuel consumption, or travel time. By using the framework, edge weights capturing environmental impact can be computed for the whole road network, thus enabling eco-routing. To the best of our knowledge, this is the first work that provides a general framework for assigning trip cost based edge weights based on a set of (*trip*, *cost*) pairs.

Two directions for future work are of particular interest. It is of interest to explore whether accuracy improvement is possible by using distinct *PEAK* and *OFFPEAK* tags for different road segments. Likewise, it is of interest to explore means of updating weights in real time. A module that takes as input real time streaming data, e.g., real time GPS observations along with costs, can be incorporated into the framework.

Acknowledgments

This work was supported by the Reduction project that is funded by the European Commission as FP7-ICT-2011-7 STREP project 288254.

References

[1] *What is the EU Doing on Climate Change?* [Online]. Available: http://ec.europa.eu/clima/policies/brief/eu/index_en.htm

[2] *Reducing Emissions from Transport* [Online]. Available: http://ec.europa.eu/clima/policies/transport/index_en.htm

[3] C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul, "EcoMark: Evaluating models of vehicular environmental impact," in *Proc. 20th Int. Conf. GIS*, Redondo Beach, CA, USA, 2012, pp. 269–278.

[4] T. Kono, T. Fushiki, K. Asada, and K. Nakano, "Fuel consumption analysis and prediction model for ŞecoŤ route search," in *Proc. 15th World Congr. Intelligent Transport Systems ITS America's Ann. Meeting*, New York, NY, USA, 2008.

[5] E. Ericsson, H. Larsson, and K. Brundell-Freij, "Optimizing route choice for lowest fuel consumption-potential effects of a new driver support tool," *Transp. Res. C Emerg. Technol.*, vol. 14, no. 6, pp. 369–383, 2006.

[6] G. Tavares, Z. Zsigraiova, V. Semiao, and M. G. Carvalho, "Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling," *Waste Manage.*, vol. 29, no. 3, pp. 1176–1185, 2009.

[7] T. Idé and M. Sugiyama, "Trajectory regression on road networks," in *Proc. Nat. Conf. AAAI*, 2011, pp. 203–208.

[8] T. Idé and S. Kato, "Travel-time prediction using Gaussian process regression: A trajectory-based approach," in *Proc. SDM*, 2009, pp. 1183–1194.

[9] S. Clark, "Traffic prediction using multivariate nonparametric regression," *J. Transp. Eng.*, vol. 129, no. 2, pp. 161–168, 2003.

[10] J. Yuan *et al.*, "T-drive: Driving directions based on taxi trajectories," in *Proc. 18th SIGSPATIAL Int. Conf. GIS*, New York, NY, USA, 2010, pp. 99–108.

[11] J. Ygnace, C. Drane, Y. B. Yim, and L. Renaud, "Travel time estimation on the san francisco bay area network using cellular phones as probes," Inst. Transp. Stud., UC, Berkeley, CA, USA, Tech. Rep. UCB-ITS-PWP-2000-18, 2000.

[12] J. C. Herrera and A. M. Bayen, "Incorporation of Lagrangian measurements in freeway traffic state estimation," *Transp. Res. B Methodol.*, vol. 44, no. 4, pp. 460–481, 2010.

[13] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proc. 17th ACM SIGKDD Int. Conf. KDD*, New York, NY, USA, 2011, pp. 316–324.

[14] G. Song, L. Yu, and Z. Wang, "Aggregate fuel consumption model of light-duty vehicles for evaluating effectiveness of traffic management strategies on fuels," *J. Transp. Eng.*, vol. 135, no. 9, p. 611, 2009.

[15] K. Ahn, H. Rakha, A. Trani, and M. Van Aerde, "Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels," *J. Transp. Eng.*, vol. 128, no. 2, pp. 182–190, 2002.

[16] E. Köhler, K. Langkau, and M. Skutella, "Time-expanded graphs for flow-dependent transit times," in *Proc. 10th Annu. ESA*, Rome, Italy, 2002, pp. 599–611.

[17] B. George and S. Shekhar, "Time-aggregated graphs for modeling spatio-temporal networks." in *Proc. ER (Workshops)*, Tucson, AZ, USA, 2006, pp. 85–99.

[18] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[19] B. Jiang, "Ranking spaces for predicting human movement in an urban environment," *Int. J. Geographical Inform. Sci.*, vol. 23, no. 7, pp. 823–837, 2009.

[20] E. Crisostomi, S. Kirkland, and R. Shorten, "A google-like model of road network dynamics and its application to regulation and control," *Int. J. Control*, vol. 84, no. 3, pp. 633–651, 2011.

[21] A. N. Langville and C. D. Meyer, "Survey: Deeper inside pagerank," *Internet Math.*, vol. 1, no. 3, pp. 335–380, 2003.

[22] C. F. Daganzo and Y. Sheffi, "On stochastic models of traffic assignment," *Transp. Sci.*, vol. 11, no. 3, pp. 253–274, 1977.

[23] B. Jiang, S. Zhao, and J. Yin, "Self-organized natural roads for predicting traffic flow: A sensitivity study," *J. Statist. Mech. Theory Exp.*, vol. 2008, pp. 7008–7035, Jul. 2008.

[24] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[25] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie, "T-Finder: A recommender system for finding passengers and vacant taxis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2390–2403, Oct. 2013.

[26] S. Seo *et al.*, "HAMA: An efficient matrix computation with the mapreduce framework," in *Proc. IEEE 2nd Int. Conf. CloudCom*, Indianapolis, IN, USA, 2010, pp. 721–726.

[27] J. Lin and C. Dyer, "Data-intensive text processing with mapreduce," *Synthesis Lect. Hum. Lang. Technol.*, vol. 3, no. 1, pp. 1–177, 2010.

[28] P. Cantos-Sanchez, R. Moner-Colonques, J. J. Sempere-Monerris, and A. Alvarez-SanJaime, "Viability of new road infrastructure with heterogeneous users," *Transp. Res. A*, vol. 45, no. 5, pp. 435–450, 2011.

[29] *Igraph Library* [Online]. Available: http://igraph.sourceforge.net/

[30] *Matrix-Toolkits-Java Package* [Online]. Available: http://code.google.com/p/matrix-toolkits-java

Bin Yang received the B.E. and M.E. degrees from Northwestern Polytechnical University, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from Fudan University, China, in 2010. He was a Research Assistant at the Aalborg University, Denmark, during 2008–2009. He was a Post-Doctoral Researcher at the Max-Planck-Institut für Informatik, Germany, from 2010-2011. In September 2011, he joined Aarhus University, Denmark, as a postdoc at the level of Research Assistant Professor. His current research interests include data management and data analytics. He has served on program committees of several database conferences and has been invited as a Reviewer for several database journals, including *ICDE*, *TKDE*, and *The VLDB Journal*.

Manohar Kaul received the B.E. (Honors) degree from the Department of Computer Science and Electronic Engineering, Latrobe University, VIC, Australia, in 2000. From 2000 to 2009, he was with the industry, primarily at ORACLE for 5 years as a Senior Systems/Database Architect, specializing in handling very large datasets, especially in the utilities, banking, and telecommunication sectors. In late 2009, he joined the M.Sc. Computer Science Programme at the Computer Science Department, Uppsala University, Sweden, and graduated in 2011. Currently, he is a Ph.D. student with the Data Intensive Systems Group at Aarhus University, Denmark, under the supervision of Prof. C. S. Jensen. His current research interests include spatio-temporal databases, indexing, and graph theory.

Christian S. Jensen is an Obel Professor of Computer Science at Aalborg University, Denmark. He was a Professor at Aarhus University 2010 to 2013, and he was at Aalborg University for two decades prior to that. He recently spent a 1-year sabbatical at Google Inc., Mountain View. His current research interests include data management and data-intensive systems, and its focus is on temporal and spatio-temporal data management. He is an ACM and an IEEE fellow, and he is a member of the Academia Europaea, the Royal Danish Academy of Sciences and Letters, and the Danish Academy of Technical Sciences. He has received several national and international awards for his research. He is an Editor-in-Chief of *The VLDB Journal* and will become Editor-in-Chief of *ACM TODS* in June 2014.