RC24903 (W0911-116) November 20, 2009
Mathematics

IBM Research Report

Real Time Traffic Estimation Using Data Expansion

Roger Lederman
Columbia Business School
Uris Hall
3022 Broadway
New York, NY 10027

Laura Wynter
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Real-Time Traffic Estimation Using Data Expansion

Roger Lederman* Laura Wynter†

October 17, 2009

Abstract

This paper presents a method for estimating traffic volumes on a road network using historical and real-time traffic data when a non-negligible subset of the network links do not have data. The approach involves both an offline phase and a real-time phase. In the offline phase, a bilevel program must be solved, thereby generating a new set of parameters for the real-time phase. The real-time phase is efficient enough to be scalable to full city-wide deployments. Simulations on several test networks show excellent results.

1 Introduction

Real-time traffic data is readily available in many cities around the world. Real-time data comes from numerous sources; some of these sources have been available for decades, such as inductive loops present at traffic signals, and others are more recently prevalent, such as GPS data from equipped vehicles, and digital video. One data type produced by these systems are traffic volumes, another common data type available in real time is average speed.

Increasingly, traffic authorities are interested in leveraging these types of data for real-time traffic analytics. Real-time traffic analytics include such capabilities as route guidance and real-time information provision on the road condition for drivers, as well as tools for improving traffic flow for network operators. All of these new and emerging capabilities require an accurate estimation of current and future predicted traffic on the road network.

In order to address these important challenges, a first step is to assess the availability in real time of traffic volumes across the road network. In many cases, while the data is available in principle, it includes many gaps, both spatially and temporally. In other words, traffic volumes are available some of the time on some of the links, but seldom all of the time on all of the links.

For certain applications, gaps in the real-time data availability present a serious impediment to their effective use. For instance, traffic-dependent route guidance requires estimates of the traffic across the links of the network. Missing data on parts of the network can lead to route suggestions that are highly sub-optimal for the current and future traffic levels. The same predicament arises for network managers who wish to optimize the flow of traffic in real time. If the incoming real-time data has significant gaps, or any gaps on critical links, operational decisions cannot necessarily be made with confidence.

Many techniques exist for describing traffic flow on a network, such as traffic simulation (see, for example, [18, 2, 12] and dynamic traffic assignment, or equilibrium, models (see, for example [16, 6]). Use of these approaches, however, to estimate traffic in real-time is generally less-than-satisfactory for two important reasons. On the one hand, the computation time of these approaches is often prohibitive. Although advances have been made in both computation capacity and algorithm design, and in spite of some efforts to use traffic simulation in real-time, results tend to be mixed. A significant reason for this is still today the heavy computational burden that simulation programs and dynamic traffic equilibrium models demand.

On the other hand, these approaches do not readily incorporate the real-time traffic characteristics, but rather are based on a typical set of parameters. Typical parameters include origin-destination demands as well as link cost functions. These parameters tend to reflect well average-case conditions, but may not reflect as well the real-time attributes of the traffic at any point in time.

Hence, although traffic simulation and dynamic traffic equilibrium models offer the desired type of information by providing traffic flows as an output, that output may not reflect closely the current traffic on the network and may

*Columbia Business School, Uris Hall, 3022 Broadway, New York, NY 10027, rlederman13@gsb.columbia.edu
†IBM Research, PO Box 704, Yorktown Heights, NY 10598, lwynter@us.ibm.com

not be achievable in a time frame permitting real-time decision making. We therefore propose a method here that both literally, and figuratively, fills these gaps. Literally, the goal of the work here is to fill in the gaps in real-time traffic flows. Figuratively, the method proposed fills the gaps described above by enabling a consistent and accurate set of link volumes to be estimated in real-time. Specifically, the problem we solve is to estimate traffic volume on all links of a road network in real-time, using a combination of current and historical data from road sensors. In practice, both the current and historical observations contain data for only a subset of the links in the network.

The approach presented consists of two phases: a real-time estimation phase, and an offline calibration phase. For the real-time, estimation problem, called (**J**), we propose a least-squares formulation with linear equality constraints. The real-time estimation problem can be solved efficiently even over large networks with only moderate computational complexity. The parameters of that estimation problem are determined through the offline calibration problem. The offline calibration problem, called (**Q**), will take the form of a bi-level program. The formulation which we develop and present here can be calibrated using only historical averages of link volumes. While the offline problem is computationally intensive, it need not be solved more often than, for example, once a week. The key considerations are that the offline problem can be solved periodically and that the real-time problem can be solved over a city network in a matter of seconds.

The following section presents our notation and assumptions. Section 3 describes the real-time estimation problem. In Section 4, we formulate the offline, calibration problem, and discuss our algorithmic approach. Section 5 presents numerical results obtained using our approach on test networks. Section 6 concludes with a discussion of the merits of our approach as well as some extensions of our model to incorporate additional information that may be available for some road networks.

2 Notation and Assumptions

The graph $G(N, A)$ represents our traffic network, with N the set of nodes, and A the set of links connecting the nodes. Each link $i \in A$ is directed from a tail node, $tail(i) \in N$, to a head node $head(i) \in N$. For convenience, we also define, for each link i, the sets $A^O(i) := \{j \in A |\, tail(j) = tail(i)\}$ and $A^I(i) := \{j \in A |\, head(j) = tail(i)\}$ to characterize the incidence relationships between links.

Let $W \subseteq N \times N$ be a set of origin-destination (OD) pairs. For each pairing $w = (orig(w), dest(w)) \in W$, there is a demand for travel from $orig(w)$ to $dest(w)$. Traffic enters the network at $orig(w)$, bound for $dest(w)$, at a rate r_w. The full set of travel demands are contained in the $|W|$-vector, r, which may be restricted to belong to some set $R \subset \Re^{|W|}$. For each link $i \in A$, we also define the incidence sets: $W^O(i) := \{w \in W |\, orig(w) = tail(i)\}$ and $W^I(i) := \{w \in W |\, dest(w) = tail(i)\}$.

Drivers choose a path from their origin to their destination. Let $\mathcal{P}$ be the set of possible paths, P_k, through the network. For each $w \in W$ we define the set $\mathcal{P}_w \subset \mathcal{P} := \{P_k \in \mathcal{P}, P_k \text{ from } orig(w) \text{ to } dest(w)\}$. The parameter z_k is the volume of flow on path k, with the property $\sum_{P_k \in \mathcal{P}_w} z_k = r_w$. In the subsequent sections we will discuss assumptions regarding driver behavior, leading to additional properties of z.

We relate paths and links through a set of indicator functions. The notation $\mathbf{1}_i^k$ takes a value equal to 1 if link i is contained in path P_k, and 0 otherwise. We denote l_i as the volume of flow on link i, with the property that $l_i = \sum_{P_k \in \mathcal{P}} \mathbf{1}_i^k z_k$. Travel time on a link is dependent on link volume, with actual times determined by a link impedance function, $V_i(l_i)$. Path travel time, c_k, is defined by summing link travel times, so that $c_k = \sum_{i \in A} \mathbf{1}_i^k V_i(l_i)$.

We use the notation l to represent the collection of data in the current observation. We are only able to observe a subset $\mathcal{O}$ of the link volumes, so that l consists only of l_i for $i \in \mathcal{O} \subseteq A$. The real-time observation problem is to determine volume estimates, $\hat{l}$, for all links $i \in A$. The desired output will then be the volume estimates $\hat{l}_i$, on those links $i \in A \backslash \mathcal{O}$ without real-time data.

In dealing with historical data, we divide observations into segments, each corresponding to a set of time intervals (e.g. 7-8 AM, Monday - Friday). We create S segments, so that each observation falls into a segment $s \in \{1 \dots S\}$. Historical data is thus represented by a set $H^s = \{l^{s1}, \dots, l^{s|H^s|}\}$ for each segment $s \in \{1 \dots S\}$, where l^{sn} contains the historical link volume observations, l_i^{sn}, for each link $i \in \mathcal{O}^{sn} \subseteq A$. We define the set $\mathcal{O}^s := \bigcup_{n \in \{1 \dots |H^s|\}} \mathcal{O}^{sn}$ containing links for which there is some amount of historical data for time segment s. We call the segment associated with the current observation s^*.

The mean rates r^s apply to all instances within segment s, but the actual demand at any time point, current or historical, is assumed to vary around this mean. Because of this variation, our estimate, $\hat{r}$, of the current demand, need not match r^{s^*}. Rather, we view the current rate as a random variable, with mean r^{s^*}. We note that the rates r^s are not observed. They can only be inferred from the historical link flows in the set H^s.

Link ID	03-04	03-05	03-06	03-09	03-10	03-11	03-12
0111	37	-	45	-	-	71	47
0112	98	106	103	95	110	102	111
0113	12	-	-	9	-	-	7
0114	0	-	4	0	0	2	0
0115	-	84	-	56	-	-	-
0116	22	30	29	15	30	31	35
0117	5	20	-	35	7	-	-
0118	-	-	-	-	-	-	-
0119	-	178	200	154	-	205	220
0120	70	-	120	150	140	65	72
0121	-	-	-	-	-	-	-

Link ID	Current Sensor Output
0111	-
0112	102
0113	-
0114	0
0115	40
0116	30
0117	-
0118	-
0119	180
0120	-
0121	-

Figure 1: Sample of historical and current link volume data with spatial and temporal gaps for a given time of day.

Despite this variability, a critical assumption underlying our approach concerns the stability of the traffic network and the traffic demands. While the network and the demands may change over time, they should be relatively stable. In particular, we should be able to segment such that mean demands in each segment remain constant. With regard to changes in network structure itself, this framework is amenable to cyclical patterns such as seasonal road closings or peak-period lane adjustments. Significant changes in the network structure on a daily basis would, on the other hand, pose a problem for the use of our method.

2.1 Data Expansion Overview

A stylized example will provide some intuition for our approach. Suppose a traffic authority is managing the network in Figure 2 consisting of five links and a single OD pair. The travel times are known to satisfy the impedance function $V_i(l_i) = l_i^2$. In this case the drivers have the choice of three paths through the network. The possibilities are $P_1 = \{1, 2, 3\}$, $P_2 = \{1, 2, 4\}$ or $P_3 = \{5\}$. At present, information on link flows is available only for link 5. The process of estimating the four remaining link flows is what we call *data expansion*.

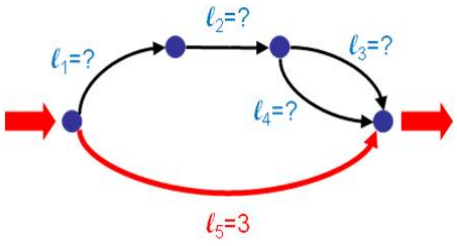

Figure 2: Example of a traffic network with five links and only one real-time observation.

We would like to formulate estimates for the remaining link flows that are based on reasonable assumptions of driver behavior. A common assumption in traffic planning is that drivers choose the path with the shortest travel time, from which it is inferred that all three paths will yield the same travel time. (This line of reasoning is the basis for the Wardrop Equilibrium [17] concept, presented formally in Section 4.) Using this logic, we can estimate that l_3 and l_4 are equal. Furthermore, the travel time on any path is $l_5^2 = 9$. Seeing that $l_1 = l_2 = 2l_3$ for any choice of path flows, the travel time on P_1 is $\frac{9}{4}l_1^2$, from which we determine our estimate $\hat{l}_1 = 2$.

It is appealing to describe behavior in terms of shortest paths for long-term, as opposed to real-time, traffic planning. The Wardrop Equilibrium paradigm has been adopted to solve origin-destination matrix estimation problems (see [4, 15, 11]) related to the former setting. Unfortunately, to carry out these calculations on a larger scale requires the solution of bilevel program which, from a computational standpoint, renders this type of estimation unsuitable for a real-time deployment in a city-sized network.

Suppose instead we have a simplified model of drivers, wherein drivers reaching a particular node choose among the outgoing links in some fixed proportion (the "splitting probabilities" are known to the planner). This model enables an efficient approach to traffic assignment based on flow propagation. For data expansion, in the context of the network in Figure 2, we can use the requirement that $\hat{l}_1/(\hat{l}_1 + l_5) = \frac{2}{5}$, to determine that $\hat{l}_1 = \hat{l}_2 = 2$. Requiring

as well that $\hat{l}_3/(\hat{l}_3 + \hat{l}_4) = \frac{1}{2}$, we conclude that $\hat{l}_3 = \hat{l}_4 = 1$. Moreover, we will see that this propagation technique scales up effectively to large-scale networks in a way that the path-based approach does not.

The propagation approach is in effect a linearization of route-choice behavior. With fixed splitting probabilities, an increase in the travel demand leads to a proportional increase of the flow on each link. When impedance functions are nonlinear, Wardrop Equilibria do not scale linearly, so that no fixed set of splitting probabilities can consistently match the path-based estimates when travel demands vary randomly. However, we have found that a careful parametrization of the fixed-proportion model can lead to very good estimates in realistic traffic networks.

Our proposed two-phase method seeks to exploit the efficiency of flow propagation in the real-time phase, while relying on path-based estimates to calibrate the model offline. As a result, we are able to approximate the OD matrix estimation procedure effectively in a real-time setting.

3 Real-Time Estimation

The real-time estimation problem assumes the existence of a calibrated set of parameters α^s for each segment. For each arc $i \in A$, α^s contains the weights $\{\alpha^s_{ij}; j \in A^I(i)\}$ and an additional parameter α^s_{io} that relates link i directly to the travel demands. These calibrated parameters define a model such that a flow l in segment s is expected to satisfy the constraints:

$$\begin{aligned} l_i &= \sum_{j \in A^I(i)} \alpha^s_{ij} l_j + \sum_{w \in W^O(i)} \alpha^s_{io} r_w - \sum_{v \in W^I(i)} \alpha^s_{io} r_v \quad (\forall\, i \in A) \\ l_i &\geq 0 \quad (\forall\, i \in A) \\ r &\in R^s \end{aligned} \tag{1}$$

Unless otherwise stated, R^s contains only nonnegativity constraints for each term r_w. We denote the set of pairs (l, r) satisfying (1) as $\Lambda^s(\alpha^s)$.

The weights are interpreted in terms of propagating of traffic through the network. The parameter α^s_{ij} is the proportion of the flow on link j that continues onto link i (as such, weights for link j will satisfy $\sum_{i|j \in A_I(i)} \alpha^s_{ij} \leq 1$, with any slack signifying a portion of flow that remains at the head node). In other words, the real-time estimation problem seeks to determine a set of volumes that propagate the flow in a manner consistent with network flow principles.

The approach presented here involves satisfying flow propagation on the network while ensuring that the estimated volumes do not stray too far from those that are observed on the links that have real-time volume observations. That is, we select a flow $\hat{l}$ from $\Lambda^s(\alpha^{s^*})$ that minimizes the error, $\sum_{i \in \mathcal{O}} (\hat{l}_i - l_i)^2$. We call this approach *data expansion* as it expands known link volumes to the rest of the network via flow propagation.

In general, we expect that a set of segment-level historical link flow estimates, $\hat{l}^s$, are also made available after the calibration phase. These estimates, which ignore any real-time data, can be used to induce more conservative real-time estimates. We allow for weight to be placed on both real-time observations and historical estimates in our formulation of the real-time problem. In its most general form, the estimation problem, $\mathbf{J}^s(l, \hat{l}^{s^*}, \alpha^{s^*}, M)$, for given user-defined constants, M, is then:

$$\min_{(\hat{l}, \hat{r})} \left[M_1 \cdot \left(\sum_{i \in \mathcal{O}} (\hat{l}_i - l_i)^2 \right) + M_2 \cdot \left(\sum_{i \in A} (\hat{l}_i - \hat{l}^{s^*}_i)^2 \right) \right] \tag{2}$$

$$\textit{s.t.} \qquad (\hat{l}, \hat{r}) \in \Lambda^s(\alpha^{s^*}) \tag{3}$$

In the numerical portion of the paper, we have focused on implementing a particular formulation of this framework. We restrict the set α^s to weights where all flow into a node is propagated in the same proportions. Specifically, for any link i, the weights α^s_{ij} take the same value as α^s_{io} for all links $j \in A^I(i)$. For this formulation we streamline notation for splitting probabilities to an $|A|$-vector. To avoid confusion with differing numbers of indices, we avoid using the notation α for this $|A|$-vector of splitting probabilities and instead use the notation, p^s. Hence, the feasible set, $\tilde{\Lambda}(p^s)$ is defined by those pairs (l, r) satisfying:

$$\begin{aligned} l_i &= \sum_{j \in A^I(i)} p^s_i l_j + \sum_{w \in W^O(i)} p^s_i r_w - \sum_{v \in W^I(i)} p^s_i r_v \quad (\forall\, i \in A) \\ l_i &\geq 0 \quad (\forall\, i \in A) \\ r &\geq 0 \end{aligned} \tag{4}$$

Furthermore, we place our focus on closely matching real-time observations, with no significant weight placed on the historical estimates. The vector $\hat{l}^s$ is used only as a secondary criterion. This is accomplished by applying a large positive multiplier, M to the real-time error term. The formulation, $\tilde{\mathbf{J}}(l, \hat{l}^{s^*}, p^{s^*})$, is then:

$$\min_{(\hat{l},\hat{r})} \left[M \cdot \left(\sum_{i \in \mathcal{O}} (\hat{l}_i - l_i)^2 \right) + \left(\sum_{i \in A} (\hat{l}_i - \hat{l}_i^{s^*})^2 \right) \right] \tag{5}$$

$$\text{s.t.} \qquad (\hat{l}, \hat{r}) \quad \in \quad \tilde{\Lambda}(p^{s^*}) \tag{6}$$

Since the real-time estimation problem is a linearly-constrained least squares problem, it can be solved efficiently with custom software or commercial packages such as CPLEX [5].

4 Offline Calibration

The purpose of the offline calibration problem is to determine the parameters of the real-time estimation. In particular, the parameters p^s (or α_{ij}^s in the more general case), must be computed for each time segment, s. Recall that the real-time estimation problem makes use of these pre-calibrated parameters to enforce a method of propagation of traffic. Clearly, the way in which traffic is propagated through the road network is a reflection of the paths that are chosen by drivers. As such, we are motivated to carry out flow propagation in a way that most closely mimics path-based estimation.

To help estimate link volumes, we will employ the assumption that drivers choose shortest paths as they perceive them. As a result, path flows should satisfy conditions for Wardrop Equilibrium. In our setting here, we make use of a deterministic definition of Wardrop Equilibrium. Our approach however is quite general and would equally well accommodate stochastic user equilibrium in the formulation. Using our notation, therefore, in the deterministic setting, we require that path flows satisfy

$$P_k \in \mathcal{P}_w, z_k > 0 \Rightarrow c_k \leq c_{k'} \text{ for all } P_{k'} \in \mathcal{P}_w. \tag{7}$$

In other words, for any path k in the set of paths serving origin-destination pair, w, if there is any flow on the path, it must be a shortest path between that pair. Since link impedance functions depend upon flow, determining which paths are minimum cost paths requires iteration. This is the typical approach common in network equilibrium models. See, for example, [13] for an overview of models and algorithms for solving traffic network equilibrium.

The traffic equilibrium problem, however, like dynamic traffic assignment or simulation models relies upon average-case data and does not readily incorporate real-time data nor do they typically accurately reflect current traffic conditions. Hence, traffic equilibrium models on their own, be it deterministic or stochastic, are not satisfactory for our purpose of accurately calibrating parameters for estimating real-time traffic. Therefore, our calibration approach will involve *expanding* historical observations to the entire network by computing the most likely Wardrop Equilibria, as determined by the recent link flows that we have observed under similar circumstances. The relevance of the historical observations used in calibration is assured through segmentation.

For each segment, we expand our historical observations to a complete Wardrop Equilibrium, giving us a full characterization of flow on the network. We will then use the estimated historical data to calibrate our model of flow propagation. This is the crux of our approach. We must therefore update the data of the equilibrium model on a regular or punctual basis so that it closely reflects the situation as observed on the traffic network.

We denote the historical link flow estimates for each segment by $\hat{l}^s$. The historical estimates themselves are used, as we have seen, as a baseline for real-time estimation, and crucially, to calibrate splitting parameters. We would like for historical estimates to closely match the historical average volumes of flow on each link, while also adhering to the Wardrop Equilibrium principle. Historical average flows are defined for links in the set $\mathcal{O}^s$, and are computed as $\bar{l}_i^s = \left(\sum_{\{n: i \in \mathcal{O}^{sn}\}} l_i^{sn} \right) / \left(\sum_{\{1 \ldots |H^s|\}} \mathbf{1}\{i \in \mathcal{O}^{sn}\} \right)$.

For a given vector r of demands, the set $Z(r)$ of feasible link flows is given by all flows, l, that satisfy the following:

$$\begin{aligned} l_i &= \sum_{P_k \in \mathcal{P}} \mathbf{1}_i^k z_k \qquad (\forall\, i \in A) \\ \sum_{P_k \in \mathcal{P}_w} z_k &= r_w \qquad (\forall\, w \in W) \\ l_i &\geq 0 \qquad (\forall\, i \in A) \end{aligned} \tag{8}$$

Then, $L(r)$, the set of Wardrop equilibria corresponding to demand r is defined formally by:

$$\{l \in Z(r) : \sum_{i \in A} (V_i(l_i)(l'_i - l_i)) \geq 0, \ \forall \, l' \in Z(r)\} \tag{9}$$

Equivalently, $L(r)$ consists of those elements of $Z(r)$, for which (7) is satisfied. Computationally, we can find an equilibrium corresponding to the demands r, by solving a convex optimization problem (see [1]) over the set $Z(r)$. That is, the set of Wardrop equilibria is equivalent to:

$$\arg\min_{l \in Z(r)} \left[\sum_{i \in A} \int_0^{l_i} V_i(u) du \right] \tag{10}$$

In the offline phase, we seek for each segment a pair $(\hat{l}^s, \hat{r}^s)$ such that $\hat{l}^s \in L(\hat{r}^s)$ and $\hat{l}^s$ is a close match to $\bar{l}^s$. The most likely flows can be computed via techniques of origin-destination (OD) matrix estimation. The purpose of typical instances of OD matrix estimation is to determine from a sample of link flow data a likely origin-destination matrix that may have produced those observed link flows. This has been an active area of research for several decades. A seminal paper was that of Cascetta and Ennio [3], which presented a least-squares formulation of the problem. Since then, the model has been generalized to take the form of a bilevel program [4, 15, 11].

Hence, the offline calibration problem involves solving an OD matrix estimation problem, as a function of observed link flows, $\mathbf{Q}^s(\bar{l}^s)$, for every time segment, s:

$$\min_{(\hat{l}^s, \hat{r}^s)} \left[\sum_{i \in \mathcal{O}^s} (\hat{l}_i^s - \bar{l}_i^s)^2 \right] \tag{11}$$

$$\begin{aligned} \textit{s.t.} \quad \hat{l}^s &\in \arg\min_{x \in Z(\hat{r}^s)} \left[\sum_{i \in A} \int_0^{x_i} V_i(u) du \right] \\ \hat{r}^s &\in R^s \end{aligned} \tag{12}$$

There are several heuristic approaches that can be used to solve the OD matrix estimation problem, including a gradient-based approach presented in [14] and developed further in [9]. Based on a sensitivity analysis of the bilevel program having a lower level defined by the separable, discrete traffic assignment problem, the authors refine older approaches and provide a rigorous model that can be used to derive gradients (or subgradients) of the bilevel program. While the authors provide an instance of their sensitivity analysis to use in a descent method for solving network design, it can be readily adapted to OD matrix estimation when the problem takes the form of a bilevel program. A recent work [10] presents a different way of obtaining gradients that can also be used in a descent algorithm for the OD matrix estimation problem (or network design problem) presented here.

In either case, the OD matrix estimation problem must be solved periodically so as to obtain updated link flows that correspond to the recent observed traffic counts. Then, in the next step, the offline calibration procedure makes use of the updated link flows to compute the parameters used in the real-time estimation problem. While there are several ways to obtain those parameters, we present here one such approach which has the benefit of simplicity and still provides very good results in our tests.

The formulation that we implement is consistent with the real-time implementation ($\tilde{\mathbf{J}}$) in that within each segment, the proportion of flow choosing link i is described by a single parameter p_i^s. Given a set of historical flow estimates, it is straight-forward to calibrate parameters of this type. The traffic at any node for a particular time period has a typical pattern of splitting across outgoing links, and this pattern can be deduced uniquely from the relative size of the outgoing link flow estimates. We wish to capture this behavior, and do so by computing the following:

$$p_i^s \;=\; \hat{l}_i^s \;/ \left(\sum_{j \in A^I(i)} \hat{l}_j^s \;+\; \sum_{w \in W^O(i)} \hat{r}_w^s - \sum_{v \in W^I(i)} \hat{r}_v^s \right); \quad (\forall \, i \in A) \tag{13}$$

Using these splitting percentages as parameters, the real-time traffic estimation can be solved very efficiently; indeed, they are expressed as linear constraints to the least squares estimation problem.

5 Numerical Results

In this section, we test and evaluate our approach for real-time traffic estimation on two test networks taken from the Berlin, Germany regional road network. These networks are among those presented and used by Jahn et al. in [8]. We use the relatively small Friedrichshain network, which has 224 nodes, 523 links, and 23 demand zones, resulting in 506 origin-destination pairs. In addition, we use the Berlin network which is considerably larger with 12,981 nodes, 28,376 links, and 865 zones, resulting in 46,689 origin-destination pairs. The network descriptions and the OD matrices can be obtained from the website of Hillel Bar-Gera at [7].

In these tests we seek to evaluate two aspects of the traffic estimation method: the ability of the method to predict accurately the traffic volumes on the network, and the level of coverage of missing values that can be achieved. To do so, we use the German networks and their OD matrices to generate a set of link flows. Those link flows are considered the true flows on the network at the current point in time. We then simulate historical data and current sample observations using that base set of link flows by randomizing them. Randomization takes two forms: modification and suppression. By modification, we mean that the flow values are modified by a random value following a normal distribution and a standard deviation of given percent of the mean. By suppression we mean that there is a random chance that a link's value is suppressed in the sample. We then allow these two forms of random error to increase and generate multiple such samples. The data sample sets are summarized in the two tables found in Figure 3.

Friedrichshain Data Sets	Historical Data Modification %	Historical Data Suppression %	Real-time Data Modification %	Real-time Data Suppression %
Data Set 1	20	20	20	20
Data Set 2	20	40	20	40
Data Set 3	20	40	30	50
Data Set 4	20	40	40	60
Data Set 5	20	40	50	70
Data Set 6	20	40	50	80
Data Set 7	30	40	30	40
Data Set 8	30	50	30	50
Data Set 9	30	50	20	60
Data Set 10	30	50	40	60
Data Set 11	30	50	50	80
Data Set 12	40	50	40	50
Data Set 13	40	60	40	60
Data Set 14	50	80	40	60
Data Set 15	50	90	50	90

Berlin Data Sets	Historical Data Modification %	Historical Data Suppression %	Real-time Data Modification %	Real-time Data Suppression %
Data Set 1	20	20	20	40
Data Set 1	20	40	20	50
Data Set 1	20	40	30	60
Data Set 1	20	40	40	70
Data Set 1	30	50	30	50
Data Set 1	30	50	50	80

Figure 3: Degrees of random data modification and suppression in the historical and real-time data sets used in numerical study on Friedrichshain and Berlin networks

What we observe in general is that, even for high degrees of random input data error, both via modification and suppression, our traffic estimation method works exceptionally well. The figures below give an indication of this.

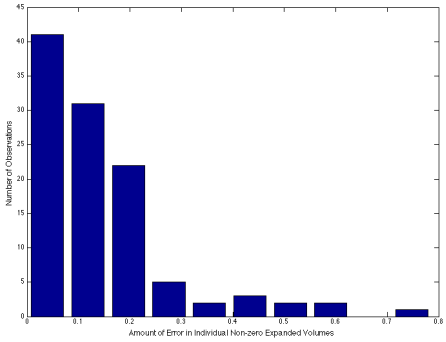

Figure 4: Friedrichshain (Germany) network; Histogram of numbers of links by DEA error (ARE) level when input data error is low to moderate: 20% input data error in the historical and current data samples, and 40% of the values in both suppressed.

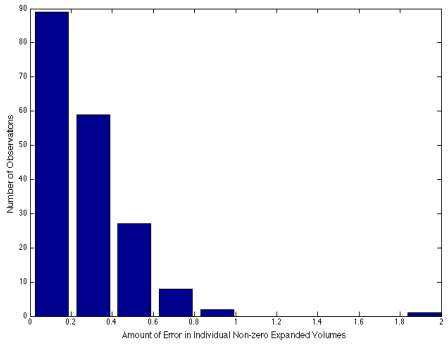

Figure 5: Friedrichshain (Germany) network; Histogram of numbers of links by DEA error (ARE) level when data error is low to moderate in the historical data and high in the current data sample: 20% error in the historical and 50% error in the current data samples; 40% of the values suppressed in the historical and 70% of the values suppressed in the current data sample.

We compute error in the output data expansion volumes as follows. The absolute relative error (ARE) of a computed value, v, is x if the true value is w and $x = |v - w|/w$.

Figure 4 has what we consider to be a low to moderate amount of error in the input data, both historical and current data samples. Error in the DEA-expanded (computed) volumes is very low: for over 40 links, the DEA error is only a few percentage points, and for over 30 links, the DEA error in the expanded volumes is in the low single digits such as 10-15%, and over 20 links have error around 10-22%. Only a few links have errors higher than that.

Figure 5 has the same random level of noise in the historical data samples, that is 20% error in the values themselves with 40% of the values randomly suppressed. However, the level of error is much higher in the sample representing a current data set. This situation is reflective of the case where multiple historical data sets are available and some form of averaging can be undertaken to result in a richer set of historical data than any one single sample can provide. In that case, the current data set is more likely to be poor in data than the historical set. In this figure, the current data set has a 50% error in the input data values themselves and 70% of them are missing. As can be viewed from the figure, error in the expanded volumes increases, but is still excellent. Indeed, the largest group of links (around 90 of them) have error levels under 20%, and the next largest group, around 60 links, has between 20 and 40% error.

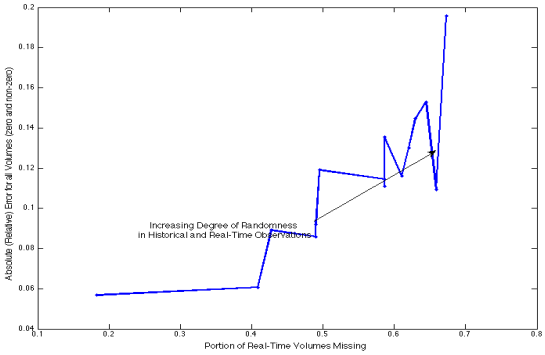

Figure 6: Friedrichshain (Germany) network; Graph of the absolute relative error (ARE) in the expanded volumes as the error level in the data increases.

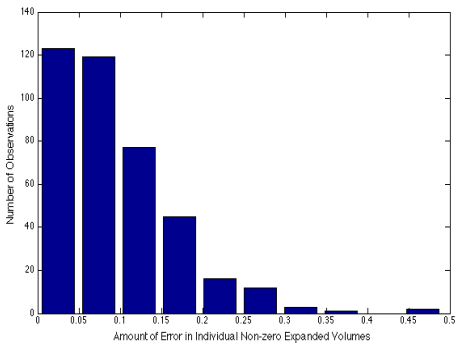

Figure 7: Berlin (Germany) network; Histogram of numbers of links by DEA error (ARE) level when data error is low: 20% error in the historical and current data samples, as well as in the percent of historical data suppressed, and 40% of current values suppressed.

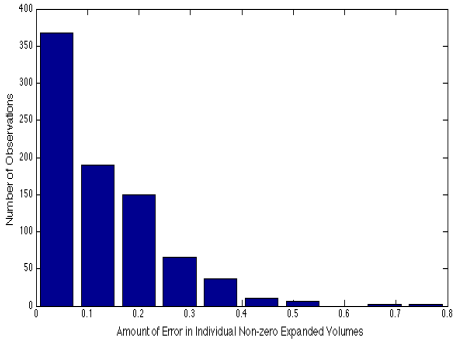

Figure 8: Berlin (Germany) network; Histogram of numbers of links by DEA error (ARE) level when data error is low to moderate in the historical data and high in the current data sample: 20% error in the historical and 30% error in the current data samples; 40% of the values suppressed in the historical and 60% of the values suppressed in the current data sample.

In Figure 6, the graph shows the absolute relative error in the expanded, or computed, volumes, as the input data

error level increases. Since we have four different dimensions along which the input data error/suppression increases, this graphic is only roughly schematic of increasing input data error. However, it does give an idea of the level to which the error reaches as data quality worsens. Since there are a finite set of data samples, only the points on the graph have significance and are associated with the amount of missing data. The data samples are ordered so that error in the observed values may increase towards the left as well. The reason that the absolute relative error looks like it is capped at 20% for these data sets is that some links have zero flows in the true solution and also in our estimation.

The Berlin network is much larger than the Friedrichshain network and resembles that of many major metropolitan areas in terms of its size and number of origin-destination pairs. Because of the very large number of links, the percentages appear to be much smaller. To make the comparison realistic, we do not include zero-flow links in the error measurements. Hence, even if flow is correctly estimated to be zero, we do not include it as there are very many zero-flow links. The errors reported here are thus more stringent as they concern errors in the non-zero flow values.

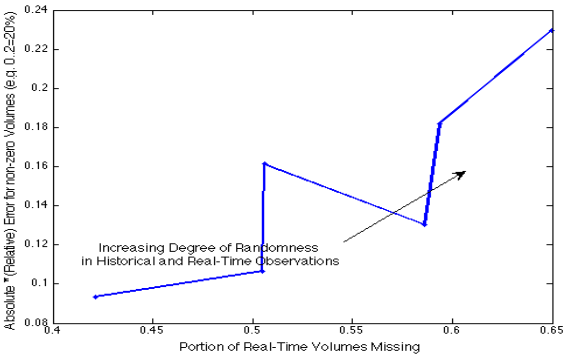

Figure 9: Berlin (Germany) network; Graph of the absolute relative error (ARE) in the expanded volumes as the error level in the data increases.

In Figure 7, there is 20% error in the historical data used as input for the model, both in terms of the values and the number of links with data. In the current data sample, there is 20% error in the values as well, but 40% of the links' values are suppressed. The DEA volumes output error level is very low, we see that the first bar in the histogram represents those links with error of under 5%, and it is the bar with the highest number of links. Just after it, those links with 5 to 10% error are almost as high in number. Clearly, there are numerous zero-flow links in the Berlin network, but of those several hundred links with flow but whose value is "missing" in the current data sample, most have very low error in their estimated values.

Similarly, Figure 8 illustrates analogous results. In this case, there is 20% error in the historical data, 30% missing values, 40% error in the current sample and 60% missing data in the current sample. As with the Friedrichshain network, this choice of input data error levels to our model represents the case where historical data can be averaged over many samples and hence is of better quality than any single current data set. As before, we see that the error in the estimated values is very low: most all links with estimated values have computed error lower than 22%.

Finally, as we illustrated with the Friedrichshain network, the absolute relative error can be averaged and plotted as the input data error level, on the whole, increases. Again, this is purely schematic since the input data error level is not increasing linearly on the x-axis, as there are four different dimensions along which error increases (modification and suppression in both historical and current data sets). However, it does give an impression as to how high the DEA output error in the estimated values is increasing.

In addition to data accuracy, data coverage is another metric of importance for this type of real-time traffic estimation, in which our goal is to fill in missing data. The following two figures illustrate data coverage on the Friedrichshain and the Berlin networks. Our definition of coverage is the ability of the method to assign a non-zero flow to any link that should have a non-zero flow in the true data set. Recall that the true data values are not known by the method since we apply both data modification and data suppression to the historical as well as the current data samples.

Because the Friedrichshain network is quite small, the coverage achieved via our method is nearly perfect. The Figure 10 shows that coverage is at 100% until the amount of missing data increases beyond 40%. Again, the data

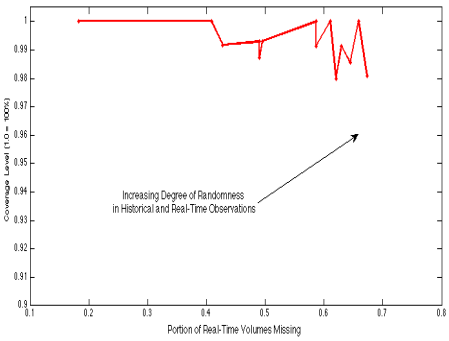

Figure 10: Friedrichshain (Germany) network; Histogram of numbers of links by DEA error (ARE) level when data error is low to moderate: 20% error in the historical and current data samples, and 40% of the values in both suppressed.

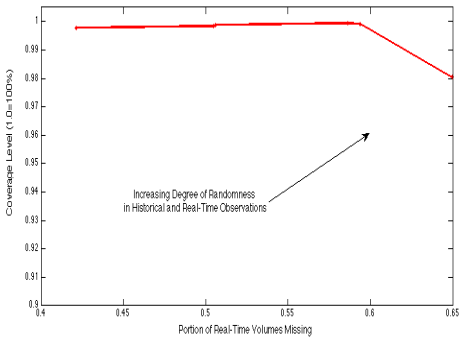

Figure 11: Berlin (Germany) network; Histogram of numbers of links by DEA error (ARE) level when data error is low to moderate: 20% error in the historical and current data samples, and 40% of the values in both suppressed.

sets do not have a linearly-increasing error level along all four dimensions so that the figure is somewhat schematic, but it conveys the high level of coverage that is achieved by our method.

The last figure, 11, shows the same on the Berlin network. The coverage level is close to 100% until the amount of missing data exceeds 60%.

6 Conclusions

We presented a method for traffic estimation via an approach that we call *data expansion*. The goal of the method is to fill in missing values in real-time traffic volumes. This is important for enabling real-time traffic data to be used in many new and emerging traffic applications. Indeed, in practice, real-time data on the network flows is often missing, both spatially, with gaps on some links, as well as temporally, with gaps at some points in time. Such gaps in the real-time traffic data render difficult the use of analytic tools such as those for real-time route guidance and network control. We sought a method that would meet those objectives while being computationally lightweight enough to run in real-time. Realizing that much of the computational overhead comes from reading and writing to database, the method itself needs to run in a matter of seconds on a city-wide traffic network.

The method we developed works in two phases. The offline phase involves the resolution of a set of bilevel programs for a number of pre-determined time segments. The online phase of our method is designed to be fast and scalable so that it can be run in real-time and makes use of the parameters computed in the offline phase along with real-time data on traffic flows. Our method was tested here on two networks from Germany and shows excellent results, both in terms of accuracy as well as in terms of coverage of the missing values on the network.

A related application of bilevel programming is OD matrix estimation, which is itself a computationally challenging problem to solve accurately. Among our contributions is the implementation of recently developed sensitivity analysis techniques to solve this problem on a large scale. While unsuitable for real-time use, our offline implementation is scalable to realistic networks for applications where computation time is not constrained as severely.

Within the two-phase framework we present, there are a number of variations to the specific approach we have implemented. Depending on the details of the problem setting, it may be possible to achieve a closer approximation to path-based estimation. For instance, let $f(p, l)$ be a measure of the distance (squared norm), between a link flow l and the closest approximation to that flow from the set $\Lambda(p)$. The current implementation fits the splitting probabilities p^s to the average historical flows, $\bar{l}^s$, in effect minimizing $f(p^s, \bar{l}^s)$. Alternately, we can expand each historical observation l^{sn} to a full estimate $\hat{l}^{sn}$, and choose p^s to minimize the sum of $f(p^s, \hat{l}^{sn})$ over our history. While, given a long enough history with enough coverage, $\bar{l}_i^{\,s}$ is a close approximation to the expected flow, $E[l_i]$, this observation-based approach aims directly at minimizing the expectation of $f(p, l)$. Another variation of this work involves the use of different assignment algorithms, some of which may offer more spreading of flow across paths, a characteristic that would be of use for this particular application.

Among the various choices, the specific procedure we implemented in this article was chosen on the strength of its robustness. When the percentage of links included in each observation is low, an observation-based approach will

introduce a higher degree of error in the offline expansion phase. By aggregating observations into the term $\bar{l_i}^s$, we ease these difficulties. On the other hand, when travel demands do not vary, nothing is gained through the real-time phase, making an entirely offline approach desirable. By including a component that works with real-time data, we have geared our method to work in the presence of demand variability. On balance, our approach was chosen to be robust with respect to both missing data and demand variability. In this paper, we have tested for both of these scenarios, with very encouraging results.

7 Acknowledgements

The authors would like to acknowledge Lalit Agarwala, of IBM, who coded the final version of the algorithm, produced the randomized data sets for the test networks and provided the numerical results that have been illustrated in this paper.

References

[1] M.J. Beckmann, C.B. McGuire, and C.B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, New Haven, CT, 1956.

[2] Moshe Ben-Akiva, Michel Bierlaire, Haris Koutsopoulos, and Rabi Mishalani. Dynamit: a simulation-based system for traffic prediction, 1998.

[3] Ennio Cascetta. Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator. *Transportation Research Part B: Methodological*, 18(4-5):289–299, 1984.

[4] Ennio Cascetta and Maria Nadia Postorino. Fixed point approaches to the estimation of o/d matrices using traffic counts on contested networks. *Transportation Science*, 35(2):134–147, May 2001.

[5] CPLEX. http://www.ilog.com/products/cplex/.

[6] Terry L. Friesz, David Bernstein, Tony E. Smith, Roger L. Tobin, and B. W. Wie. A variational inequality formulation of the dynamic network user equilibrium problem. 41(1):179–191, 1993.

[7] Hillel Bar Gera. http://www.bgu.ac.il/ bargera/tntp/.

[8] Olaf Jahn, Rolf H. Mohring, Andreas S. Schulz, and Nicolas E. Stier Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Operations Research*, 53(4):600–616, 2005.

[9] Magnus Josefsson and Michael Patriksson. Sensitivity analysis of separable traffic equilibrium equilibria with application to bilevel optimization in network design. *Transportation Research Part B: Methodological*, 41(1):4–31, January 2007.

[10] Shu Lu. Sensitivity of static traffic user equilibria with perturbations in arc cost function and travel demand. *Transportation Science*, 42(1):105–123, 2008.

[11] Jan T. Lundgren and Anders Peterson. A heuristic for the bilevel origin-destination-matrix estimation problem. *Transportation Research Part B: Methodological*, 42(4):339–354, 2008.

[12] Michael Mahut, Michael Florian, and Nicolas Tremblay. Comparison of assignment methods for simulation based dynamic-equilibrium traffic assignment. TRISTAN, 2004.

[13] Michael Patriksson. *The Traffic Assignment Problem—Models and Methods*. 1994.

[14] Michael Patriksson. Sensitivity analysis of traffic equilibria. *Transportation Science*, 38(3):258–281, 2004.

[15] Michael Patriksson. On the applicability and solution of bilevel optimization models in transportation science: A study on the existence, stability and computation of optimal solutions to stochastic mathematical programs with equilibrium constraints. *Transportation Research Part B: Methodological*, 42(10):843–860, 2008.

[16] Bin Ran, David E. Boyce, and Larry J. LeBlanc. A new class of instantaneous dynamic user-optimal traffic assignment models. *Operations Research*, 41(1):192–202, 1993.

[17] J.G. Wardrop. Some theoretical aspects of road traffic research. *Proc. Ins. Civil Engineers, Part II*, 1(36):325–378, 1952.

[18] Brian J.N. Wylie, Gordon Cameron, Matthew White, Mark Smith, and David Mcarthur. Paramics: Parallel microscopic traffic simulator, 1993.