Transportation Research Part C 36 (2013) 245–260

Contents lists available at ScienceDirect

Transportation Research Part C

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

A microstate spatial-inference model for network-traffic estimation

Adeyemi J. Fowe [a], Yupo Chan [b,*]

[a] Nokia (HERE Maps), 425 Randolph Street, Chicago, IL 60606, USA
[b] University of Arkansas at Little Rock, 2801 S. University Avenue, Little Rock, AR 72204, USA

ARTICLE INFO

Article history:
Received 21 July 2012
Received in revised form 24 August 2013
Accepted 24 August 2013

Keywords:
Spatial inference
Traffic estimation
Entropy maximization
Golden-section search

ABSTRACT

In an Advanced Traveler Information System (ATIS), sensors are often used to monitor and obtain traffic information on a real-time basis. Knowing that traffic sensors cover only a fraction of the road network, we investigate how to estimate traffic volumes on arcs that are not covered by sensors. By exploiting the spatial properties and the topology of a network, we derive a microstate model that can be used to estimate these traffic volumes. Based on entropy maximization, we present a microstate surrogate for competing techniques such as traffic assignment, and algebraic method or topological approach in estimating traffic flow. Being an entropy model, it also has advantage over these competing techniques in terms of the prerequisite information required to enable the model. Being a microstate rather than a steady-state model, it takes into account the fluctuation of traffic and it executes fast enough to allow real-time estimation of traffic flow. By covering the entire network flow this way with only a limited number of sensors, it will help in better driver routing decisions and traffic management tactics while being cognizant of today's budgetary constraints facing operating agencies. The algorithm has been tested successfully in Little Rock, Arkansas and in a controlled experiment with a randomly generated 100-node/522-arc grid network.

1. Introduction

Due to the extensive instrumentation requirement, complete coverage of traffic information in a road network requires high infrastructural and operational cost. This leaves a complete coverage of the network practically unachievable in agencies operating under a tight budget. Lack of sensor coverage for real-time traffic information, however, has impeded the deployment of many ATIS applications. This in turn has inspired research on optimal placement of sensors by Liu and Danczyk (2008), Ban et al. (2009), Kianfar and Edara, (2010), Gentili and Mirchandani, (2011), and Barcelo et al. (2013)just to name a few. Most recently, Ng (2012) has proposed a node-based model for the network sensor-location problem. He showed that a minimum subset of the network arcs and their locations existed for the estimation of the unobserved arc flows. Our paper goes beyond sensor placement. Starting with a limited number of sensors already in place, we present a way to estimate traffic volumes on arcs where traffic is unknown. Should the level of accuracy be acceptable, we are able to cover the entire network, possibly without increasing the number of sensors. This is achieved by exploiting the microstate relationship between vehicle movements, based on the topology of the network. In contrast to macrostate, a microstate description of traffic would identify each trip individually about where it is coming from and where it is heading, instead of just accounting for the trip ends (Golan, 1998). We will show that this is a viable approach for an ATIS, rather than the competing methodologies that estimate an aggregate, steady-state traffic volume for the network.

* Corresponding author.
E-mail addresses: james.fowe@nokia.com (A.J. Fowe), ychan@alum.MIT.edu (Y. Chan).

0968-090X/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trc.2013.08.011

A classic way of estimating traffic is through traffic assignment. In spite of significant advances in the field, Zhang (2011) pointed out that the behavioral foundations of the models, as well as the practical implications of various behavioral assumptions, were rarely discussed. Zhang found out that there were significant discrepancies between arc-flow estimates arising from different behavioral foundations. Aside from this, traffic assignment is often inappropriate for real-time applications, when accurate origin–destination (O–D) demands are not available. Traffic assignment is also computational intensive, limiting its use when traffic has to be estimated "on the fly." Since the network configuration is given and can be represented by its link-path incidence matrix, some argue that the problem of network observability can be well solved by either the *algebraic method* (Hu et al., 2009) or *topological approach* (Castillo et al., 2008). A study of network observability provides useful information on which selected O–D-pair flows or arc flows are informative on other O–D-pairs and arc flows. Based on this, a practical application is to guide the placement of sensors to collect the most relevant traffic counts to infer a network-wide traffic pattern. However, the topological approach requires prior knowledge of O–D or path flows, or route-choice probabilities given by some traffic-assignment rules, and to formulate the flow-conservation equations. This constrains the applicability of the approach in real-world situations, particularly over real time, when such information is an estimate at best. On the other hand, the algebraic method, in relating the unobserved arc flows to the arc-path incidence matrix, may give rise to multiple solutions in terms of the set of *basis links* for sampling. In addition, there is a need for complete path enumeration—a computationally intensive undertaking. This discounts greatly its potential for real-time applications. All these considerations mean that we need to go beyond traffic assignments, algebraic and topological methods to estimate network traffic volumes.

Going beyond traffic estimation for a particular time slice, we wish to endow traffic operators with a look-ahead capability regarding what the traffic pattern is like a few minutes from now. Kamarianakis and Prastacos (2005), Min et al. (2011), and Zhang et al. (2012) are among the first to apply space-time models for traffic forecasting, focusing exclusively on arcs covered by sensors. Their work distinguishes from others by embracing the fundamental principle that there is a closer traffic relationship between nearby arcs than further-away arcs (Tobler, 1965)—a principle that will be exploited in full in this paper as well. Starting from a microstate level, we extend their work by estimating traffic volumes on unknown arcs. Furthermore, the current approach is based on more robust information. Instead of just projecting forward the arc volumes recorded by sensors, we estimate unknown arc traffic using *information theory* (or *entropy maximization*). The Principle of Maximum Entropy states that, subject to known or testable information, the probability distribution which best represents the current state of knowledge is the one with largest entropy (Jaynes, 1982). Most recently, Giffin and Caticha (2007) stated that Bayes' Rule and the Principle of Maximum Entropy were completely compatible, lending further credence to the entropy maximization paradigm. To facilitate a look-ahead capability, we estimate traffic one time-period at a time, including a future time-period. When performed repeatedly, the proposed approach has the potential for real-time ATIS applications.

Synthesizing the above literature, we wish to take the best features of all existing techniques while avoiding their worst features. In this regard, we are happy to report here an approach that distills the flow-continuity feature of traffic assignment, the network geometry of the topological approach, and the node-arc incidence information of the algebraic method into a single model. In this model, we consider the upstream and downstream neighbors as the *region of influence* of an arc in traffic estimation. Our model attempts to determine the most appropriate traffic estimates based on neighboring observations in this region. In an eventual aggregate statement of a travel pattern it is useful to be robust enough to accommodate as many detailed, microstate patterns as possible, where each microstate represents a stochastic instance of traffic flow. The most likely aggregate flow-pattern is assumed to be one with the greatest number of possible microstates. This embraces the tenet of entropy maximization, which attempts to capture all possible microstate patterns. In this context, entropy is interpreted in the spatial context as a measure of the frequency with which a traffic event occurs between neighboring arcs. We will estimate the unknown arc traffic by Shannon's seminal work in information theory (Shannon, 1948), and show that it is not only accurate, but computationally feasible for real-time applications. While travel demand matrices have been routinely estimated by entropy maximization, this is the first attempt to apply entropy maximization to arc traffic estimation—to the best of our knowledge.

2. Model formulation

Let $\overline{Q}^l$ represent a one-dimensional column vector of observed traffic flows $\bar{q}_l$ in the lth order for every arc i, $i = 1, 2, ..., m$ in a network. Let $\widetilde{W}^l$ be an $m \times m$ matrix of the lth-order neighbors with weight entries w_{ij}^l denoting the relationship between neighboring arcs i and j. When $l = 1$, only first-order (or immediate) neighbors j of subject arc i are considered. This says that only the set of arcs directly connected to arc i exert influence on the subject arc, dismissing arcs more than one "hop" away. Arcs with non-zero weight values in each row of $\widetilde{W}^l$ represent upstream arcs, while the arcs with non-zero weight values in a column represent downstream arcs. The upstream-arc j for a subject-arc i suggests that there may exist some arc-j traffic-flow into the arc i, while the downstream-arc j for arc i implies possible arc-i traffic-flow into arc j.

2.1. Key parameters

Let the set $\Gamma^l(i)$ stand for the *region of influence* for arc i, with elements k representing both the upstream and downstream arcs among the set of lth-order neighbors for arc i, $\{l\}$. i.e.,

$$I^l(i) = \bigcup_{k\in\{l\}} [i-k \cup k-i] \quad \text{for all } i; \text{for all } l \tag{1}$$

Eq. (1) simply accounts for all the downstream neighboring arcs {i–k} and upstream neighboring arcs{k–i} within the lth order neighborhood. Splitting $\widetilde{W}^l$ into two, let $\widetilde{W}^l_U$ represent a two-dimensional matrix with row-wise normalized weights $(\leftarrow w^l_{ij} \rightarrow)$ of upstream spatial neighbors and $\widetilde{W}^l_D$ represent a two-dimensional matrix with column-wise normalized weights $(\leftarrow w^l_{ij} \rightarrow)^T$ of downstream spatial neighbors. This means that the row sum or column sum of the weight entries adds up to unity inasmuch as the normalized weights assume the forms $w^l_{ij}\big/\sum_j w^l_{ij}$ and $w^l_{ij}\big/\sum_i w^l_{ij}$ respectively. For a specific arc i, let $\overline{W}^l_U(i)$ represent the normalized weights of row vector of the lth-order upstream-neighbors of arc i, and let $\overline{W}^l_D$ represent the weights of a column vector of the lth-order downstream-neighbors of arc i. This simply refers to the row of $\widetilde{W}^l_U$ where arc i is located, and the column of $\widetilde{W}^l_D$ where arc i is located.

For an arc i with unknown arc-flow q_i that is to be estimated, let $\mu_U(i)$ represent the weighted average of arc flows from the upstream neighbors of arc i, and let $\mu_D(i)$ represent the weighted average of arc flows to the downstream neighbors. Accounting for the first, second, and up to the lth-order neighbors, we can define these two measures as follows

$$\begin{aligned} \mu^l_U(i) &= \sum_{p=1}^{l} \overline{Q}^l \overline{W}^p_U(i) \Big/ p \\ \mu^l_D(i) &= \sum_{p=1}^{l} (\overline{Q}^l) \overline{W}^p_U(i) \Big/ p \end{aligned} \tag{2}$$

Recall that $\overline{Q}^l$ denotes a vector of observed arc traffic in the lth order neighbors. Notice that by summing over all l neighboring orders, these two equations consider not only arc traffic directly connected to arc i, but also more remote neighbors in the region of influence $I^l(i)$ that do not directly relate to arc i. Moreover, closer-by arcs are weighted more than further-away arcs.

In practice, only a subset of the upstream or downstream arc volumes is available. If $l > 1$, there are $l-1$ arcs in between k' and i for the lth-order neighbors, where i is the arc for which traffic is to be estimated, and k' stands for either a downstream or upstream neighboring arc where observed flows maybe available. Consider a region of influence $I^2(i)$ that includes up to the second-order ($l = 2$) neighboring arcs that are related to arc i. The observed arc-volumes in the sets $\overline{Q}^1$ and $\overline{Q}^2$ that contribute toward arc i are likely to be profuse, even though traffic is not necessarily available for every neighboring arc. The reason is that there is a multiplying effect among the *degrees* of the nodes along the branches of the two-order tree fanning out from the subject arc i either downstream or upstream. Take the network in Fig. 2, and assume that we include all second-order neighbors. The arc volumes in $\overline{Q}^1$ and $\overline{Q}^2$ that contribute toward the formal definitions of the "band gap" [μ_U, μ_D], namely Eq. (2), include all arcs in the network except arcs 2, 4 and 7.

The bounded values of $\mu^l_U(i)$ and $\mu^l_D(i)$ represent the major information in estimating the arc flow of i. Based on network properties, such as network geometry, the estimated traffic-flow q_i on arc i is supposed to be between the value of $\mu^l_U(i)$ and $\mu^l_D(i)$, after accounting for measurement errors. Should we consider higher-order neighbors, node-arc incidence information is exploited. Again take the network in Fig. 2. Should the traffic on upstream arc 10 be unavailable while traffic on arcs 1 is, μ^l_U is estimated based on additional node-arc incidence information. The traffic count of 50 on arc 1 will now serve the same function as arc 10, being the arc directly upstream from arc 10. In the numerical example to be shown below, the weighted average of arc flows from the upstream neighbors of arc i would be μ^2_U instead of μ^1_U, indicating that second-order neighbors are involved. The corresponding estimating equation, Eq. (3), becomes

$$\mu^2_U(7) = \overline{Q}^1 \overline{W}^1_U(7)/1 + \overline{Q}^2 \overline{W}^2_U(7)/2 = (34 \times 0.5 + 60 \times 0.5)/1 + (50 \times 1.0)/2 = 72$$

Remember here that the weights for each order sum up to unity, for example 0.5 + 0.5 = 1 for the first-order neighbors. As indicated in Eq. (2), the closer-by neighbors receive more emphasis than the further-away neighbors. Thus first-order observations on arcs 8 and 9 command twice the emphasis as the second-order observation on arc 1, the latter being divided by two.

Remember that in a microstate model, we are capturing the most robust traffic estimate based on the available information as expressed in the normalized Eq. (2). At the same time, we acknowledge that the traffic counts in $\overline{Q}^l$ are stochastic,

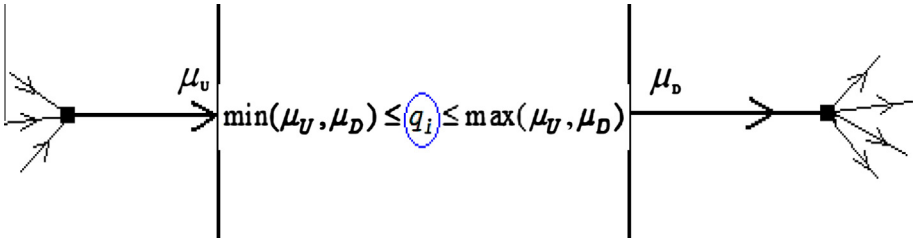

Fig. 1. Bounds on arc flow.

representing a mere observation instance of the ever-fluctuating traffic-flow. Accounting for the uncertainty, arc flow q_i most likely will fall within the range $[\mu_U^l, \mu_D^l]$, which we call the *band gap* separating upstream and downstream estimates. Fig. 1 illustrated this concept for first-order neighbors, or when $l = 1$. It is a graphic reminder that we have taken into account all the traffic flow information upstream and downstream to pin down the interval of uncertainty for determining the missing arc volume q_i.

2.2. Numerical example

This presents a numerical example that would illustrate a stepwise calculation for the proposed method. Fig. 2 shows a sample network which is a directed graph with 7 nodes {A,B,...,G} and 11 arcs {1,2,...,11}. The observed arc volumes $\bar{Q}$ are given below:

Arc:	1	2	3	4	5	6	7	8	9	10	11
Vol.:	50	55	39	-	70	5	?	34	60	26	-

Notice the observed arc traffic represents a diffuse travel pattern between multiple O-D pairs, and drivers follow more than just the shortest paths from origin to destination. For this example, it can be seen that while many arc volumes are known, several are not known, namely arcs 4, 7 and 11. This exercise focuses on arc 7 as the main unknown arc to be estimated.

The upstream and downstream spatial neighbors for arc 7 and their respective first-order weights $\bar{W}_U^1(7)$ and $\bar{W}_D^1(7)$ are provided in Fig. 3a and b. Examining upstream traffic, there are three arcs 8, 9, and 10 that are incident upon arc 7. Without any additional information, an equal weight of 1/3 (or 0.333) is assigned to each of these incident arcs, suggesting that each of these three arcs is equally likely to contribute toward traffic flow on arc 7. For first-order neighbors, or when $l = 1$, the region of influence for arc 7 is given by $\Gamma^1(7) = \{8, 9, 10, 6, 3\}$ according to Eq. (1), and the second order neighbors are $\Gamma^2(7) = \{1, 5, 11, 4\}$. The first-order neighbors are one hop away, while the second-order neighbors are two hops away. Notice that in Eq. (2), all the neighbors, not just the first-order neighbors, help to determine the band gap $[\mu_U^l(i), \mu_D^l(i)]$. For clarity of presentation, however, we will illustrate our methodology mainly for $l = 1$ in this paper, supplemented with higher orders in case studies to follow.

In this study, we are dealing with *unreliable* traffic counts gathered at a particular instance in time. We are no longer relying on "conservation of flow" to estimate arc-traffic volumes for such a stochastic network. To make our traffic-estimation procedure most robust, we wish to model on a *microstate* level, taking one vehicle at a time. For a car entering arc 7, it is equally likely to come from arc 8, 9 or 10—in the absence of site-specific information. For a car leaving arc 7, it is equally likely to go on arc 3 or 6. To handle all these possibilities, an entropy-maximization model is employed, where entropy is interpreted as a measure of the frequency with which an event occurs. Here, it means we wish to arrive at a traffic volume estimate that is robust enough to accommodate as many possible turning movement patterns as possible.

In the absence of any additional information, the above assumption of equally-likely detailed-travel is the most robust (Jaynes, 1982). Any additional information would obviously fine tune the homogeneous travel pattern, to be consistent with the newly acquired information. A fundamental rule of spatial inference states that traffic at arc i is similar to closer-by neighbors more than farther-away neighbors (Tobler, 1965). In this illustrative example, we are deliberately showing only the closest first-order neighbors and ignoring second-order and higher-up inferences. Following Eq. (2), $\mu_U^1(i) = \bar{Q}^1 \bar{W}_U^1(i)$ and $\mu_D^1(i) = (\bar{Q}^1)^T \bar{W}_D^1(i)$, or numerically:

$$\begin{aligned} \mu_U^1(7) &= \bar{Q}^1 \bar{W}_U^1(7) = 0.33 \times 34 + 0.33 \times 60 + 0.33 \times 26 = 40 \\ \mu_D^1(7) &= (\bar{Q}^1)^T \bar{W}_D^1(7) = 0.5 \times 39 + 0.5 \times 5 = 22 \end{aligned} \tag{3}$$

Here a vehicle traveling on arc 7 is equally likely to come from arcs 8, 9, or 10—hence the weights 0.33, 0.33 and 0.33 are assigned to these three arcs. Similarly, the equal likelihood applies toward the downstream traffic as well. In the microstate

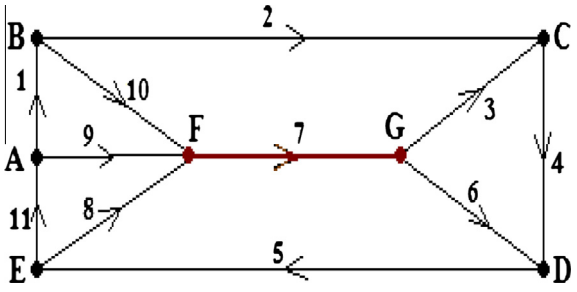

Fig. 2. Sample network.

(a)

	1	2	3	4	5	6	7	8	9	10	11
1											1
2	1										
3							1				
4		0.5	0.5								
5				0.5		0.5					
6							1				
7								0.33	0.33	0.33	
8					1						
9											1
10	1										
11					1						

(b)

	1	2	3	4	5	6	7	8	9	10	11
1		0.5								0.5	
2				1							
3				1							
4					1						
5								0.5			0.5
6					1						
7			0.5			0.5					
8							1				
9							1				
10							1				
11	0.5								0.5		

Fig. 3. (a) Upstream weight matrix $\overline{W}_U^1(7)$. (b) Downstream weight matrix $\overline{W}_D^1(7)^T$.

level, the equally weighted factors are imposed to be consistent with the entropy maximization model in the absence of additional information [Golan, 1998]. Notice that collectively these microstates result in the final estimation of the missing arc volume, as we will show sequentially.

3. Algorithms

Now that the band gap (or the range) of possible values of q_i has been defined, we present three models to estimate the most appropriate arc-flow value within the band.

3.1. Naïve method

This method simply picks a value at the center of the band gap as the estimated value. It finds the mean of the upstream and downstream arc flows. For the example shown, when $l = 1$, it is given by:

$$q_i = \frac{\mu_U^1(i) + \mu_D^1(i)}{2} \tag{4}$$

For up to l orders of neighbors, the band gaps μ_U^l and μ_D^l are now defined. In this case Eq. (4) simply generalizes to

$$q_i = \frac{\mu_U^l(i) + \mu_D^l(i)}{2} \tag{5}$$

3.2. Entropy maximization

Let X be a random variable representing the possible values of the flow on arc i. We can say that X has a sample space of real numbers within the range $[\mu_U^l, \mu_D^l]$. To discretize, we divide the sample space of X into n equal segments, or the set X

$=\{x_1, x_2, \ldots, x_n\}$, where x_v denotes the mean within a segment v. Our goal is to find the best value x_v that produces the most appropriate value of q_i. To obtain this value we employ the information-theory concept of entropy maximization, utilizing the relevant information on the possible arc-flows within the band-gap. The relevant information is contained in the region of influence, $\Gamma^l(i)$, wherein we consider the neighbors with observed arc-flows falling within the band gap. Naturally, such a "zone of influence" changes from site to site depending on network topology. The estimated arc-flow value $x_v$is given by the familiar Stirling's approximation of an entropy-maximization model:

$$\max_{1\leqslant v\leqslant n} H_v(X) = \sum_{j=1}^{\Gamma^l(i)} -\beta(x_v, x_j)\log\beta(x_v, x_j) \tag{6}$$

where $\beta(x_v, x_j)$ is a function that compares an estimated value x_v with an observed arc-flow x_j that falls within the band gap, i.e., $j \in \Gamma^l(i)$. Since $\Gamma^l(i)$ defines the "traffic zone of influence" on arc i, we are only concerned with the observations within the band gap, a gap defined by $\min\,[\mu_U^l, \mu_D^l] \leqslant x_j \leqslant \max[\mu_U^l, \mu_D^l]$. The "comparison function" now takes on the form

$$\beta(x_v, x_j) = \frac{c|x_v - x_j|}{|\mu_D^l - \mu_U^l|} \tag{7}$$

where c is a calibration constant to ensure that this function is greater than one—as consistent with the logarithm function in the Stirling approximation for entropy as shown in Eq. (6), where the entropy function $H_v(X)$ is negative in value. Assuming the "zone of influence" is determined carefully, $\Gamma^{(l)}(i)$ has already included all those observations within the lth order of neighbors that have a bearing upon the traffic on arc i. Effectively, the estimated arc volume is defined by these cogent observations.

3.3. Entropy Maximization with Golden Section Search (EMGSS)

It is clear from the above that the computational complexity of the entropy maximization algorithm is $O(|Q|n)$, where $|Q|$ is the number of arcs for which traffic is to be estimated and n is the number of intervals within the band-gap. Accordingly, the above method leaves open an important question: what are the best discretization intervals of the band gap? Obviously, this is a tradeoff between accuracy and computational complexity. More discretization intervals mean better accuracy. However, this is achieved at the expense of computation time. To resolve this issue, we propose a "divide and conquer" algorithm like the Golden Section Search (GSS), which enables us to recursively search through the band-gap by continuously discarding the intervals that no longer need to be considered. The relevant portion will be discarded by comparing the relevant entropy values.

According to GSS, the band gap Δ_1 as divided into two segments in the golden section ratio: |-- Δ_2 --|——Δ_3——|, where there is a sampling point δ_1 to the right of Δ_2 and another sampling point δ_2 to the left of Δ_2. Let us define these two entropy functions, corresponding to these two sampling points δ_1 and δ_2 within the band gap Δ_1, as determined by the golden ratio $\Delta_1/\Delta_3 = \Delta_3/\Delta_2 = \gamma = 1.618$.

$$H_1(X) = -\sum_{j=1}^{\Gamma^l(i)}\beta(\delta_1, x_j)\log_e\beta(\delta_1, x_j)\ \forall\ \delta_1, x_j \in X \tag{8}$$

$$H_2(X) = -\sum_{j=1}^{\Gamma^l(i)}\beta(\delta_2, x_j)\log_e\beta(\delta_2, x_j)\ \forall\ \delta_2, x_j \in X \tag{9}$$

Eq. (8) computes the entropy for a possible value of x_v on the left portion, represented by the sampling point δ_1. Eq. (9), on the other hand, computes another one on the right, represented by δ_2. Eq. (10) picks the bigger of the two, and a new (smaller) band gap is formed after discarding one of the two portions: Δ_2 or Δ_3.

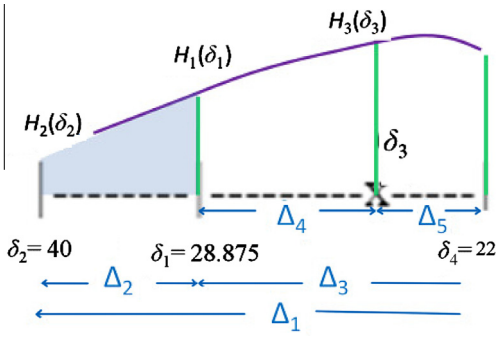

Fig. 4. First Iteration of the EMGSS algorithm.

$$H_v(X) = \max(H_1(X), H_2(X)) \tag{10}$$

As illustrated in Fig. 4, Δ_2 is discarded since $H_2(\delta_2) < H_1(\delta_1)$. This procedure is repeated following a typical GSS algorithm.

Please note that this algorithm is based on the fact that the $H_v(X)$ function is unimodal between the interval $[\mu_U, \mu_D]$, which guarantees a global maximum. We will prove this functional property in Proposition 1 in the Appendix. For a sufficiently large database, the EMGSS algorithm converges to an estimate of arc flow. Combined with the large-scale numerical experiments reported later on in this paper, we suggest that the EMGSS algorithm is competitive with traffic assignment and algebraic/topological approaches in estimating traffic flow. To show the computational efficiency of the algorithm, we will prove algorithmic convergence in Proposition 2 in the Appendix.

4. Illustrating the algorithms

Here we will illustrate the above three algorithms with numerical examples. Continuing with the numerical example as shown in Fig. 2, we will illustrate the three algorithms as follows.

4.1. Naïve algorithm

For the numerical example, flow on arc i, q_i, is to be estimated, with μ_U^l and μ_D^l representing the weighted average of arc flow from the upstream and downstream neighbors. Considering only first-order neighbors, the solution according to Eq. (5) of the *Naïve Algorithm* yields:

$$q_i \leftarrow \frac{\mu_U^1(i) + \mu_D^1(i)}{2}, \text{or}$$
$$q_7 \leftarrow \frac{\mu_U^1(7) + \mu_D^1(7)}{2} = \frac{40 + 22}{2} = 31$$

Obviously, a better estimate will result if higher-order neighbors are included in the estimation.

4.2. Entropy maximization algorithm

In this algorithm the band-gap is discretized by dividing it into four segments, with an arc-volume increment of 5 each. This means a discretized band-gap scale of 22–27–32–37–40. The observed neighboring arc volumes within the range between $\mu_U^1(7) = 22$ and $\mu_D^1(7) = 40$ are $\{x_j\} = \{26, 34, 39\}$. For Eq. (7), the choice of constant c is left to the modeler; it should however be a number that ensures $\beta(x_v, x_j)$ is greater than unity. For this example we let $c = 100$. As described above in Eqs. (6) and (7), we compare an estimated value x_v with the three observed values {26, 34, 39} one at a time. Following the sequential algorithm, these iterations are obtained, where the logarithms are all to the base e. Application of Eq. (7) yields $\beta(27,26) = 5.56$, $\beta(27,34) = 38.89$, and $\beta(27,39) = 66.67$. When these terms are substituted into Eq. (6), we have

$H_1(X) = -5.56\log 5.56 - 38.89\log 38.89 - 66.67\log 66.67 = \mathbf{-431.9.}$

Subsequent calculations are as follows:

$$\beta(32,26) = 33.33$$
$$\beta(32,34) = 11.11$$
$$\beta(32,39) = 38.89$$
$$H_2(X) = -33.33\log 33.33 - 11.11\log 11.11 - 38.89\log 38.89 = \mathbf{-285.98}$$
$$\beta(37,26) = 61.11$$
$$\beta(37,34) = 16.66$$
$$\beta(37,39) = 11.11$$
$$H_3(X) = -61.11\log 61.11 - 11.11\log 11.11 - 16.66\log 16.66 = \mathbf{-324.94}$$

$H_2(X)$ is the maximum, and it remains maximal even for other $H_v(X)$, where $v > 3$, should a finer granularity be employed. Therefore the estimated arc flow $q_7 = 32$.

4.3. EMGSS algorithm

Instead of a predetermined set of discretized trial estimates, we employ the random variable X within the sample space: $\mu_U^1 \leqslant X \leqslant \mu_D^1$, where $\mu_U^1 = 22$ and $\mu_D^1 = 40$. The EMGSS generalizes the basic GSS procedure, with additional iterations when the remaining segments are further subdivided according to the golden ratio 1.618, as illustrated in Fig. 4. Once again, $\{x_j\} = \{26, 34, 39\}$ is the set of all arc flows that falls within the band between 22 and 40. Application of Eq. (7) yields

$$\beta(\delta_1, x_j) = \frac{c|\delta_1 - x_j|}{|\mu_D^1 - \mu_U^1|} \text{ and } \beta(\delta_2, x_j) = \frac{c|\delta_2 - x_j|}{|\mu_D^1 - \mu_U^1|}.$$

The choice of constant c is again at the discretion of the modeler. It should, however, be a number that ensures $\beta(x_v,x_j)$ is greater than unity. Again, we let $c = 100$. The following calculations ensued:

$$\beta(40,26) = (100 \times |40-26|)/(40-22) = 77.778$$
$$\beta(40,34) = (100 \times |40-34|)/(40-22) = 33.333$$
$$\beta(40,39) = (100 \times |40-39|)/(40-22) = 5.5556$$
$$\beta(22,26) = (100 \times |22-26|)/(40-22) = 22.222$$
$$\beta(22,34) = (100 \times |22-34|)/(40-22) = 66.66667$$
$$\beta(22,39) = (100 \times |22-39|)/(40-22) = 94.4444$$
$$\beta(28.875,26) = (100 \times |28.875-26|)/(40-22) = 15.972$$
$$\beta(28.875,34) = (100 \times |28.875-34|)/(40-22) = 28.472$$
$$\beta(28.875,39) = (100 \times |28.875-39|)/(40-22) = 56.25$$

From Eqs. (8) and (9) we have

$$H_2(X=40) = -77.778 \times \log\ 77.778 - 33.333 \times \log\ 33.333 - 5.5556 \log\ 5.5556 = -465.04$$
$$H_4(X=22) = -22.222 \times \log\ 22.22 - 66.667 \times \log\ 66.667 - 94.44 \log\ 94.44 = -778.425$$
$$H_1(X=28.875) = -15.972 \times \log\ 15.972 - 28.472 \times \log\ 28.472 - 56.25 \log\ 56.25 = -366.28$$

Since $H_2(\delta_2) < H_1(\delta_1)$, the segment (δ_1, δ_2) is discarded, with H_4 (δ_4) as shown. The new search range is now Δ_3, where the potential arc volume x_v lies. Correspondingly, another set of calculations is performed for δ_3, a point within Δ_3 as determined by the golden ratio. Since it is a recursive process, the search continues in the same manner. Let us assume that the stop condition for the recursion is when $\Delta_{n'} < 3.0$. Then the recursion stops at recursion n', and the estimated arc volume in X is given as the latest value of x_v. In this case $x_v = 33.125$, therefore the estimated arc volume for arc 7, q_7, is 33.125.

4.4. Summary

Using the Naïve, Entropy Maximization, and EMGSS methods yield consistent, yet slightly different arc-volumes for arc 7, q_7. They turned out to be 32, 31, and 33.125 respectively for the above example. Obviously, there is no reason to expect the estimated flows to be identical, inasmuch as they are derived from different algorithms. By checking the network configuration shown in Fig. 2 and the hypothetical observed link flows of the in-degree arcs {8, 9, 10} and out-degree arcs {3, 6} of link 7, the flow conservation rule is not followed—as expected and for a good reason. In our microstate model, we expect the observed arc flows to be noisy; enforcing the conservation rule would only introduce complications.

A major effort on our arc-volume estimation model is to improve the computational complexity of the entropy maximization algorithm and determine the proper granularity of discrete segments, n, within the band gap. In Proposition 1 of the Appendix, we show that the defined entropy function, $H_v(X)$, is unimodal.

Inasmuch as the unconstrained maximum of $H_v(X)$ may be truncated by the search interval $[\mu_U^l, \mu_D^l]$, a prudent solution is to employ a direct search procedure such as the GSS. This is achieved by recursively discarding the *dominated* portions of the band-gap. The procedure continues until the search closes up on the maximum entropy value in the interval that remains. For this reason, we call the algorithm Entropy Maximization by Golden Section Search (EMGSS). This algorithm has these advantages:

- It minimizes the number of recursions n' to achieve the desired granularity of the solution (Avriel and Wilde, 1966).
- The algorithm improves on the $O(n)$ computational complexity term of the regular entropy maximization algorithm to become $O(\log(n'))$, where n' is the number of iterations.

Notice the set $\Gamma^l(i)$ could be empty for an arc i. The EMGSS algorithm checks to see if the set of $\bar{Q}^l$ is empty at the end of each pass (and then repeats if it is not). In case of an empty $\Gamma^l(i)$, the order of neighbors l can be raised ahead of time until a meaningful, nonempty $\Gamma^l(i)$ can be found. Alternatively, for the instances that $\Gamma^l(i)$ becomes empty as the EMGSS progresses, one will be able to compute $H_v(X)$ based on arc flows that have been estimated in earlier passes of the algorithm, if there were enough arc sample data to start with. In short, a tradeoff is made between computational efficiency and estimation accuracy in determining the order of neighbors to include. The convergence property of the EMGSS algorithm is formally discussed in Proposition 2 in the Appendix. To conclude the Appendix, a pseudo code for the EMGSS algorithms is also documented.

5. Case study

Having shown the serviceability of the EMGSS algorithm, a case study was performed on Colonel Glen Road and Interstate 430 in Little Rock, Arkansas—a key intersection where several adjacent developments were taking place. By choice, this first case study represents a complex highway diamond interchange that is tied to two "T with separate turning lanes" at-grade interchanges to the east and to the west. It was at this locale that a new shopping mall was being constructed. To the east is the T-intersection between Colonel Glen and Shackelford and to the west is between Colonel Glen and Bowman. As illustrated in Fig. 5, entitled "Intersection of Colonel Glen Road and Interstate 430," the area is about one mile (1.61 km) across,

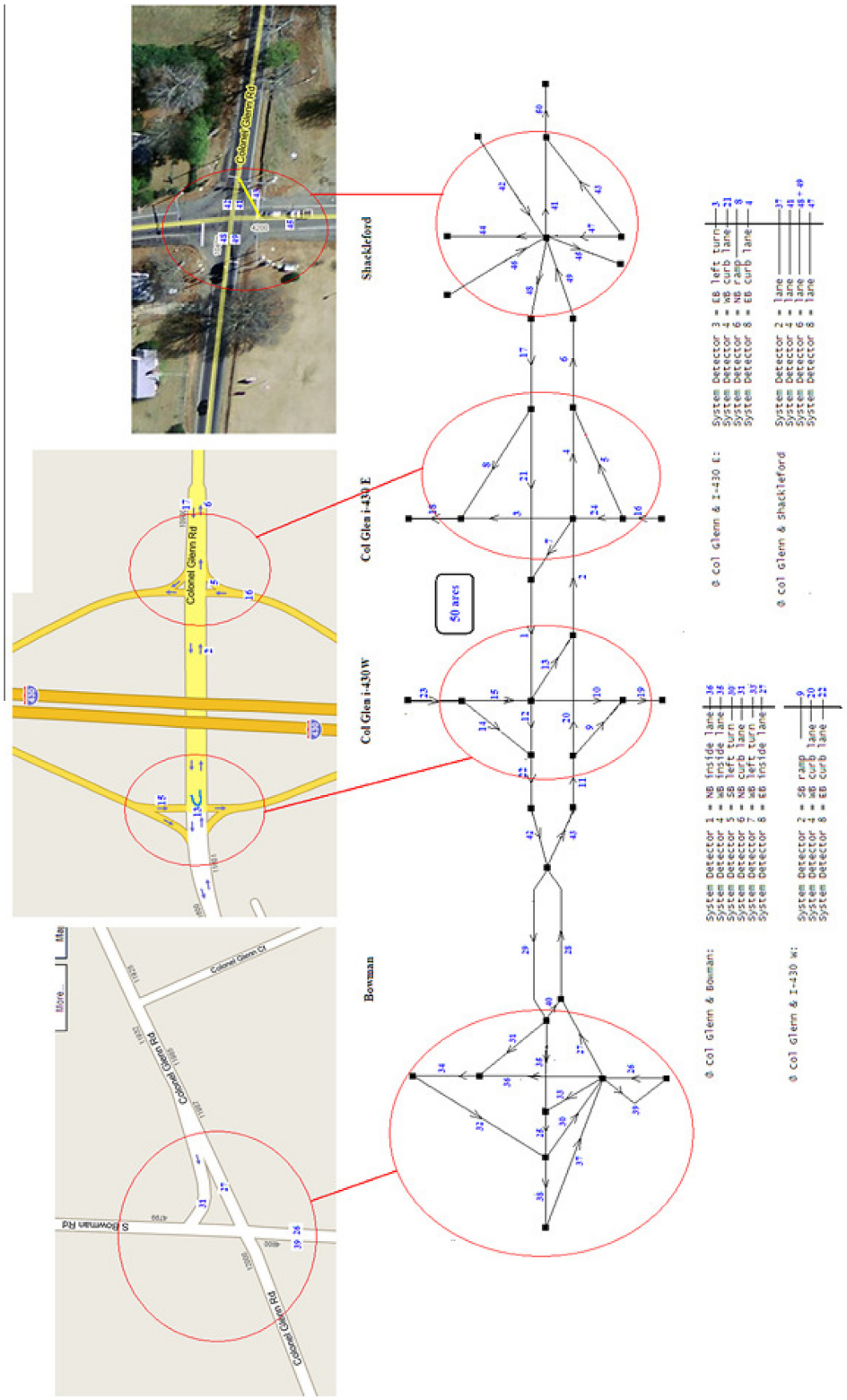

Fig. 5. Intersection of Colonel Glen Road and Interstate 430.

with traffic-detection cameras mounted at all three intersections, with the diamond interchange equipped with six of the eight cameras. Our case study was motivated by the need for additional automated signalization to handle the projected traffic increase and the large number of turning movements.

A map has been obtained from GoogleMaps (2009) as shown in Fig. 5, where each traffic movement is coded as a numbered arc, including turning movements. The purpose of the study is two-fold. First, we wish to show a real-world application of the EMGSS algorithm. More important, the case study serves as a validation of our model and algorithms. The City of Little Rock Traffic Engineering Department provided us with a 24-h traffic-volume data, collected on 26 June 2006 on 15-min time-intervals. We focus on seven arcs at the diamond highway interchange as observed from four camera detectors: d2, d3, d6 and d8. Here "d" stands for "detector" as shown in Fig. 5, where each "system detector" is identified:

Arc:	4	21	22	20	8	3	9
Camera:	d8	d4	d8	d4	d6	d3	d2

By inspecting Fig. 5 and the respective monitored traffic volumes, the traffic cameras can be ranked in importance. Thus a camera that monitors heavy traffic is more important than one that monitors light traffic. On the other hand, redundant readings are not important. Thus, the camera d2 on arc 9 is both important and not so important. It is important since it provides a unique reading that no other cameras cover; it is not so important since the traffic volume is small.

Based on the seven arc-traffic records in a 24-h day, we simulate information loss by removing detector cameras from the seven locations. We use EMGSS to estimate the "unknown" (missing) arc volumes after selected cameras have been removed. The accuracy of the estimated volume in replicating actual traffic counts is assessed using a paired t-test for the 24 arcs shown in the two circles in the center of Fig. 5. The comparison is made between two situations: (1) when all seven camera applications are used and (2) after selected camera applications are removed. The test is performed at time $t = 32$ (480 min into the day) and at time $t = 52$ (780 min into the day), representing the morning and afternoon peak period respectively. We judge that traffic at the two peak periods is much more interesting than the rest of the day.

For a comparative statistic, the two-tailed paired t-test is employed for the following null-hypothesis. The null hypothesis states that the difference in the mean is zero, when comparing the arc traffic after camera removal (the estimated) and

Table 1
Paired t-test result for the afternoon and morning peaks.

Removed arcs	t-Value	95% confidence interval ($t = 2.02$)	Comments on the afternoon peak
9	1.780	Cannot reject null-Hypothesis	Losing only arc 9 is not enough to cause a significant difference on the estimation
4, 9, 21	3.193	Reject null-hypothesis	Losing too much information would lead to a large difference. Hence the null-hypothesis is rejected
4	3.058	Reject null-hypothesis	Losing arc 4, which is a highly influential arc with very high peak values, has a significant effect
9, 20, 21	2.176	Reject null-hypothesis	Lost too much information
4, 9, 20, 21	4.540	Reject null-hypothesis	Lost a larger percentage of information and so the difference is statistically significant
9, 21	*0.085*	*Cannot Reject null-hypothesis*	*Even the loss of arc 9 and 21 can still produce a good estimate*
20, 21	*0.990*	*Cannot Reject null-hypothesis*	*Even the loss of arc 20 and 21 can still produce a good estimate*
3	2.111	Reject null-hypothesis	Being an influential arc, arc 3 is similar to arc 4. But arc 3 has one of the lowest peak volumes. Being strategically located, however, losing it still has a major effect
Removed arcs	t-Value	95% confidence interval ($t = 2.02$)	Comments on the morning peak
9	2.380	Reject null-hypothesis	Removing arc 9 results in rejecting the hypothesis because the traffic volume of arc 9 is a unique reading, arc 9 being isolated from others as shown in Fig. 5
3	0.986	Cannot Reject null-hypothesis	Dropping arc 3 does not affect the system tangibly
4	3.001	Reject null-hypothesis	Arc 4 is the arc with highest peak value and also a highly influential arc. Losing it results in rejecting the null-hypothesis
9, 21	2.726	Reject null-hypothesis	Losing arc 9 and 21 would result in rejecting the null-hypothesis
4, 9, 21	*0.804*	*Cannot reject null-hypothesis*	*The loss of the highest peak-volume arc 4 and the lowest-volume arc 9 somehow cancels each other, retaining the validity of the estimate*
9, 20, 21	3.266	Reject null-hypothesis	Removing arc 9 proves to be a significant loss once again
4, 9, 20, 21	*1.978*	*Cannot reject null-hypothesis*	*The loss of the highest peak-volume arc 4 and the lowest-volume arc 9 somehow cancels each other, retaining the validity of the estimate*
20, 21	*0.025*	*Cannot reject null-hypothesis*	*The loss of arc 20 and 21 is not critical because their arc volumes happen to be redundant information*

before camera removal (the observed). Using a 95% confidence interval that is equivalent to a t-value of 2.02, we determine whether the null-hypothesis can or cannot be rejected. If the t-value is less than 2.02, then the null-hypothesis that the difference in mean is due mainly to a random occurrence cannot be rejected. A t-value greater than 2.02 suggests that the result is statistically significant, whereby the null-hypothesis can be rejected. In the former case, it means we do not miss the dropped camera(s), while the latter case suggests that the dropped camera(s) do matter. The results of this test are provided in Table 1, where the former case is highlighted with italicized font. The results and corresponding comments are better understood when referenced against Fig. 5.

We tried different combinations of removed cameras, as shown in Table 1. The removed cameras range from a singleton, such as removing the lone camera at arcs 9, 4, and 3 respectively. At the same time, we investigated how EMGSS works when there are multiple sensors missing in a given neighborhood. In this regard, we removed the following combination of cameras: {4, 9, 20, 21}. As shown in Fig. 5, these are cameras located in the same neighborhood. From the results obtained, it can be seen that based on the information content of the traffic reading, the camera on arcs 4, 9, 20, and 21 are not as critical as those mounted on other locations. The events representing removal of these cameras are highlighted in italics in Table 1. Notice the statement should be qualified on the basis of the *combination* of cameras, rather than suggesting the importance of a single camera. As seen in a couple of instances, the effect of cameras can be reinforcing or compensating. Whether cameras are reinforcing is contingent upon how the cameras are placed amongst the network geometry as much as the distance between two cameras. Consider a major arterial that runs through town for a mile (1.61 km) without another intersection. Placing a camera at the beginning intersection of the one mile stretch and at the end of the stretch would be sufficient, in spite of the large distance between the two cameras.

The result we obtained for removal of {4, 9, 20, 21} is different for the morning vs. afternoon peak, as the traffic patterns are different. In the morning peak, we can miss the combination, but not in the afternoon peak. As another example, removing cameras at arcs 9 and 21 yields different t-values, hence different implications, suggesting the case-specific nature of applying the EMGSS estimation procedure. However, it is clear that cameras with redundant readings are not as important as cameras that provide independent traffic information (or fresh information). By trying different combinations, the findings would help the most productive location of traffic cameras. At the same time, it also identifies cameras that can be dispensed with under budgetary scrutiny. Notice this case study is not only time specific, but also site-specific. Until we perform the same study in a sufficiently large number of networks at different time periods, guidelines for sensor placement cannot be formulated.

6. Additional experiments

When we tried to estimate arc flows beyond the immediate I-430 and Colonel Glen diamond interchange, as marked by two center circles in Fig. 5, we found that the algorithm no longer provided credible traffic volumes at the two adjacent "T with separate turning lanes" at-grade interchanges. With only eight sensors to estimate a total of 50 arc volumes scattered across a mile (1.61 km) in Fig. 5, this is understandable. The situation is even worse when only four camera detectors – d2, d3, d6 and d8 – were actually effective at the diamond interchange at I-430 during the morning peak. Bowman to the west is about 0.4 mile (0.6 km) from I-430, and Shackelford to the east is about half a mile (0.8 km) away. There were three intersections in between Bowman and I-430, and there was one intersection between Shackelford and I-430. These two T-interchanges are simply too far away and there were simply too many intermediate intersections for their traffic to be estimated from cameras located at the I-430 vicinity.

6.1. A synthesized network

Since this is the only relevant dataset we obtained from the City of Little Rock, we decided to perform additional computational experiments on a synthesized network. Recognizing that the real test of the worth of EMGSS is the accuracy of its output, the synthesized network will allow controlled experiments to be carefully designed and conducted. For our purpose, we created a randomly generated 100 nodes 522 arcs grid-network, where random weights are assigned to the arcs, simulating the stochastic nature of travel speeds throughout the day. To simulate the observed road traffic, we performed "traffic assignments" using the k-shortest path algorithm (Qin et al., 2005). The O–D demands are randomized to represent peak vs. off-peak travel. Once we have the "observed" traffic for the instance, we used the EMGSS algorithm to see how well we could replicate the "observed" based on a limited number of "camera readings" at a subset of arcs. Again, we randomly placed and removed sensors. In so doing, we investigated how EMGSS works when there are multiple sensors missing in a given neighborhood. The closeness of the EMGSS estimates to the "ground truth" was assessed once again by paired t-tests.

As shown in Fig. 6, the t-statistics go up with an increasing number of arc-based sensors, suggesting that the increase in traffic-estimation accuracy as one increases the number of monitoring sensors. This is the case whether only the first-order ($p = 1$) or both the first- and second-order ($p = 2$) neighbors are used for traffic estimation. However, there is a "dip" of the curves between 0 to 90 deployed sensors for the case when only observed traffic from first-order neighbors were used. Likewise, there is a dip between 0 and 120 sensors when first- and second-order neighbors were used. When sensors are placed sparsely, accuracy can be increased with the removal of a minor number of redundant sensors, up to a certain point. Their redundancy might have introduced "noise" that downgrade the estimation accuracy. A similar situation can be found when

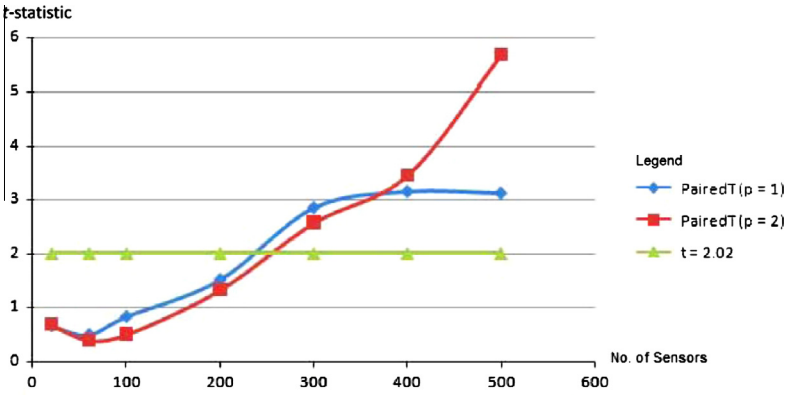

Fig. 6. Paired-*t* test result using EMGSS algorithm on a synthesized network.

sensors are place profusely, where redundancy can be abound. This is shown for the case when only first-order neighboring arcs were used, where additional placement of sensors beyond 400 were useless—although "noise" could be dampened by the large number of sensor placements. In general, the EMGSS algorithm obtained a close estimate to the ground-truth data when the number of removed arcs is less than 250 (approximately). In other words, we can remove about 250 out of the 522 arc-based sensors (or 48% of the sensors), before we notice a substantial traffic-estimation error. In the case of $p = 2$, further accuracy can be obtained by increasing the number of sensors beyond 400. The experimental results suggest that redundancy is more critical in a sparse-placement strategy for p =2, while it is more critical in a profuse-placement strategy for p =1. These results are not at all clear without the conduct of this experiment. For the field experiment, the redundancy threshold is indicated by the italics markings in the appropriate rows in Table 1, where the paired-*t* test values are less than 2.02. These are the cameras that can be removed without sacrificing traffic-estimation accuracy.

Another interesting observation is that factoring the second-order neighbors into the algorithm does not necessarily produce better performance, as one compares the $p = 1$ vs. the $p = 2$ curves. In fact, better accuracy can be obtained with only traffic counts from first-order neighbor up to 380 arc-based sensors in a 522 arc network. This delineates the importance of the neighboring order l in Eq. (2). Naturally, these results were obtained for the synthesized network as described above; it does not necessarily apply to other networks. However, since the synthesized network represents a generic urban grid-network, the result could provide a "rule of thumb" for sensor placement in urban areas. Based on our computational results, therefore, at least a 50% coverage of the arcs will allow the EMGSS algorithm to work properly. Sensor redundancy should be taken into account differently for p =1 vis-a-vis p =2.

6.2. Some practical guidelines

As shown in the pseudo code of the Appendix, the EMGSS algorithm executes at a computational complexity of only $O(|Q|\log(n'))$, where Q is the set of arc-traffic volumes to be estimated, and n' is the number of search iterations. By the nature of a GSS, the algorithm minimizes the number of recursions to achieve the desired solution granularity. If a coarser granularity is all that is required, the EMGSS algorithm becomes particularly efficient. The algorithm is therefore very competitive in obtaining useful traffic volumes real-time. Best of all, this can be accomplished with a minimal number of sensors. While the absolute minimum is not ascertained in this study, our experiment with the synthesized grid network provides some guidelines for ameliorating the "noise" that can be found in sparse sensor placement. Notice the computational complexity is not a function of the weight matrices that we apply in the algorithm. However, traffic estimation accuracy is. As shown in Eq. (2), the weight matrices determine the band-gap $[\mu_U^l, \mu_D^l]$, which in turn determines the accuracy of our traffic estimation. To obtain a realistic weight matrix (as shown in Fig. 3a and b) is a challenging work in practice. The reason is that traffic patterns change from peak to off peak, and they change over time in the long run. Remember that the assigned weights reflect the influence of each incident-upstream or downstream emanating-downstream arc traffic on the subject arc traffic-volume. As a result, the weight matrices are ever changing, inasmuch as the upstream and downstream traffic is changing in real time.

Having said that, there are some practical guidelines for composing a weight matrix. Again let us refer to Fig. 2. Suppose arc 9 is a segment of an arterial corridor that runs through arc 7. Being part of the corridor, arc 9 exerts more influence on the traffic on arc 7 than arcs 8 and 10. Accordingly, it should have a larger weight in the weight matrix than arcs 8 and 10. Likewise, if the corridor continues to arc 6 from arc 7, it can also be inferred that arc 6 should command a larger weight than arc 3. A natural question that follows is "how much more important?" Naively, one may suggest that if the traffic on arc 6 is twice that on arc 3, for example, arc 6 should receive twice the weight than arc 3. However, this argument should serve merely as the starting point. The reason is that arc 3 can very well be a major exit from the arterial, drawing a large portion of the traffic from arc 7. This will elevate the importance of arc 3 in comparison with arc 6. In general, the above observations

pertain more to peak-period traffic than off peak, since Poisson or random traffic is likely to be dominant during the latter time of the day. For a random traffic pattern, equal weights should remain to be the most robust.

While the accuracy of the estimated traffic can be improved with more realistic weights (where available), the efficiency of the algorithm is independent of the weights, as mentioned. The order of neighbors (l) has a minimal and indirect effect upon algorithm efficiency. Remember that if the "zone of influence" $\Gamma(i)$ is determined carefully, it has already included all those observations within the lth order of neighbors that have a bearing upon the traffic on arc i, and there is no need to go back on the algorithm to increase the order of neighbors. Finally, computational efficiency is only linearly related to the number of arc volumes to be estimated $|Q|$ and the log of the number of iterations in the GSS procedure n'. It does not depend on the magnitude of number of arcs/nodes in the network. Once again, remember that the EMGSS traffic-estimation procedure is performed in the absence of a regular traffic assignment, which is network-size dependent—a blessing for the reasons we have already enumerated.

For the topological approach, the observability updating procedure —an absolute requirement in real-time traffic monitoring—has a large computational complexity (Castillo et al., 2008). The complexity of the algorithm is similar to the complexity of inverting a matrix. If one has N essential variables, the number of operations required for obtaining a minimum set of observed arcs for observability is $(3N-1)N^2$. At the same time, the algebraic method identifies the "basis arcs," and then derives the non-basis arc-flows through information contained in the basis arc-flows measured by sensors. Hu et al. (2009) found out that the number of basis arcs is related to the network scale in terms of the number of arcs/nodes or used paths under different network topologies. But there is no direct correlation between the number of arcs/nodes or used paths in the network and the percentage of arcs to equip with sensors. As a result, no general computational complexity statement can be made on their algorithm. Extending Hu et al.'s work, Ng (2012) requires only node enumeration. Ng went on to derive an explicit relationship between the number of nodes and arcs, and the minimum number of sensors to install in order to be able to infer all arc flows. While this is a significant improvement, there is still no guidance on where these minimum number of sensors should be placed. This is to be contrasted with our approach that have very realistic sensor-placement locations. Moreover, both of the topological and the algebraic methods are not at all competitive with the EMGSS algorithm, which has a complexity of $O(|Q|\log(n'))$, which is divorced from the network size and geometry.

7. Conclusion

In many real-time ATIS applications, arc-traffic estimation by traffic assignment is either infeasible for lack of accurate origin–destination data or that it is infeasible computationally. Instead of solving the network observability problem by alternate means such as the algebraic method or topological approach, this paper presents a microstate procedure to estimate arc traffic volumes in regions not covered by sensors. Based on spatial inference, the traffic volume of an arc to be estimated is related to its neighboring arcs. In this regard, closer-by neighboring arcs provide more information than distant arcs. Tracing one vehicle at a time, we use *entropy* to represent the frequency with which a microstate occurs. By maximizing the entropy measure, we estimate an aggregate travel pattern that is robust enough to accommodate as many possible detailed vehicular movement patterns (or microstates) as possible.

We show that the entropy function so formulated can facilitate the construction of practical algorithms for traffic estimation on a real-time basis. The convergence properties limit the computational complexity to $O(|Q|\log(n'))$, which is independent of network size and geometry. We show the procedure for microstate spatial inference by three algorithms, with Entropy Maximization by Golden Section Search (EMGSS) being the most promising. We validated the EMGSS algorithm by field data in Little Rock, Arkansas, supplemented by a controlled experiment with a randomly generated 100-Nodes/522-arcs grid network.

We proceeded to use the EMGSS algorithm to show that the estimated traffic volumes, when some traffic-observing sensors are removed, are still competitive with the arc-volume estimates when all the sensors are present. This helps to distinguish between redundant sensors and critical sensors within the network, providing a way to economize on the number of sensors. Highway agencies are always interested in reducing their budget to provide full observability on network traffic. While it is not formulated explicitly as a sensor placement model, the EMGSS algorithm has been shown to be capable of answering such a key question.

With thousands of arcs in a typical network, it is impossible to cover real-time traffic comprehensively by sensors alone. Conventional practice often places the limited number of sensors on major roadways. However, it is absolutely necessary to examine alternate routings when incidents occur on the major roadways. These alternate routings—often on minor roadways—can only be found when traffic conditions on these roadways are known. The model and algorithm presented here respond to such a need in real time. Best of all, this is accomplished with a minimal requirement on computation and the number of sensors.

The effort reported here is a first step into an eventual goal of real-time traffic forecasting. Currently, the algorithm looks at one time interval at a time (or just a "snapshot"). A way to further this work would be to look at additional time intervals into the future. Since the unexpected changes that occur in a road network have a ripple effect on neighboring arcs, a microstate model that can project the propagation of incidents spatially in a network would be of great interest. Running the EMGSS algorithm for every time interval produces a time-series dataset for the arcs of interest. Such data, when interpreted with the spatial weight matrices inherent in EMGSS, provide a complete set of spatial time-series data covering the entire network. In short, these spatial time-series will facilitate real-time traffic forecasting. This is a promising way to forecast network-wide traffic with minimal resources.

Under this extension, an alternate measure of performance should be investigated, other than the paired *t*-test. For example, the mean absolute percentage error (MAPE) is a measure of accuracy of fitted time-series values in statistics, specifically in trend estimation. Essentially, it is the percentage error as measured by the ratio of difference between the actual observation and the forecast value, normalized by the observed value. Simple as it may be, it has a major drawback in our current application. First, if there are zero values (which is the case with many observed time series) there will be a division by zero in MAPE.

Compared to existing procedures, this paper proposes a very different way for traffic estimation or forecasting based on a limited number of monitoring sensors. The concept has been amply illustrated in two case studies, one real-world and another simulated. The studies include comprehensive parametric variation on the number of monitoring sensors. The most gratifying result is that our method appears to be well suited for real-time ATIS applications. As we pursue further studies on ATIS implementation, multiple datasets should be generated and each can be tested across the range of sensor reduction in a more comprehensive simulation. Noise can be introduced into the simulated traffic to deviate from flow conservation to test the robustness of our microstate approach. Beyond verification through simulated data, other real-world field-tests would be necessary to provide eventually the "ground truth" validation of the method.

Acknowledgments

The authors thank the referees for their detailed and thoughtful suggestions and ideas that greatly improved the quality of the paper. During the conduct of this research, we were grateful for the support from Gary DelPorto and Joseph Heflin of the Federal Highway Administration, plus Mark Bradley and David Pearce of the Arkansas Highway and Transportation Department. We are also appreciative of Casey Covington, Minh Le, Jim McKenzie, Richard McGee, and Weihua Xiao who participated as members of the research team. The statements and opinions expressed here in the paper belong to the authors and do not necessarily reflect those of the sponsors or the organizations mentioned in this manuscript. The research is supported by Federal-aid Project Number: ITSR (001) to the University of Arkansans at Little Rock.

Appendix A

Proposition 1. *The $H_v(X)$ entropy function is unimodal between the interval $[\mu_U^l, \mu_D^l]$.*

Proof. The $H_v(X)$ entropy function as shown in Eq. (6) boils down to the following form for a given bandwidth $[\mu_U^l, \mu_D^l]$and a calibration constant c, where c is fixed such that

$$\beta(x_v, x_j) = \frac{c|x_v - x_j|}{|\mu_D^l - \mu_U^l|} > 1 :$$

Collapsing the constants into a parameter "kappa," we obtain $(\sum_j \kappa|x_v - x_j| \log \kappa|x_v - x_j|)$. The entropy function now looks like

$$\sum_j -\kappa|x_v - x_j| \log \kappa|x_v - x_j|$$

which is to be maximized with respect to x_v. Without loss of generality, let us set $\kappa = 1$. Taking the partial derivative with respect to x_v yields

$$\partial\left(-\sum_j |x_v - x_j| \log |x_v - x_j|\right) \Big/ \partial x_v = \left(-\sum_j |x_v - x_j| \log |x_v - x_j|\right) x_v$$

Since all the terms in this expression are nonnegative except for the negative sign in front, the derivative is negative in value. This means the entropy function [Eq. (6)] is a unimodal concave function, where a single maximum exists. □

Proposition 2. *For a sufficiently large database, the EMGSS algorithm converges to an estimate of arc flow.*

Proof. It is well known that one important property of unimodal functions is that the extremum can be found using search algorithms such as the Golden Section Search (GSS), a special form of Fibonacci search (Avriel and Wilde, 1966). For a given precision threshold, the GSS algorithm reaches the threshold in the least number of steps. As part of the EMGSS algorithm, the "zone of influence" $\Gamma^l(i)$ is determined carefully to include a large enough sample of observed arc-flows $\{x_j\}$, wherein it will not be empty at least at the beginning steps of the algorithm. For a $\Gamma^l(i)$ that becomes empty as the EMGSS progresses, it will eventually be able to compute the entropy function $H_v(X)$ based on arc flows that have been estimated in earlier passes of the algorithm. This additional feature ensures the convergence of the EMGSS algorithm. □

A.1. Pseudo code for Entropy Maximization by Golden Section Search

Here, we will formalize the EMGSS algorithms in terms of a pseudo code and their computational complexity. Without loss of generality, we will present the pseudo code for a given "zone of influence" $\Gamma^l(i)$.

Iteration
$H_v(X) \leftarrow 0; q_i \leftarrow 0$ *for all* $i \in \overline{Q}^l$
$\beta(a,b) = \frac{c \times |b-a|}{|\mu_D - \mu_U|}$
$\varphi = \frac{1+\sqrt{5}}{2} - 1$ // *Golden Ratio*
For all $i \in \overline{Q}^l$
Function EMGSS$(\mu_U^l(i), \mu_D^l(i), \Gamma^l(i))$ *is*:
$mi = \min(\mu_U^l(i), \mu_D^l(i)); ma = \max(\mu_U^l(i), \mu_D^l(i))$
$a2 = \varphi(ma - mi)$; $a1 = (ma - mi) - a2$; $a3 = a2 - a1$;
$x_1 = mi$; $x_3 = ma$; $x_2 = x_1 + a1$; $x_4 = x_2 + a3$;
$H_1(Q') \leftarrow 0$; $H_2(Q') \leftarrow 0$; $H_3(Q') \leftarrow 0$; $H_4(Q') \leftarrow 0$;
$\mu_U(i) = Q^l \overline{W}_U^l(i)$; $\mu_D(i) = (Q^l)^{\top} \overline{W}_D^l(i)$
For $k \leftarrow 1$ to $|\Gamma^l(i)|$
$H_1(Q') = -\beta(x_1, q_k) \log_e \beta(x_1, q_k) + H_1(Q')$
$H_2(Q') = -\beta(x_2, q_k) \log_e \beta(x_2, q_k) + H_2(Q')$
$H_3(Q') = -\beta(x_3, q_k) \log_e \beta(x_3, q_k) + H_3(Q')$
$H_4(Q') = -\beta(x_4, q_k) \log_e \beta(x_4, q_k) + H_4(Q')$
End For
If $(H_1(Q') > H_3(Q')$
if $(H_2(Q') > H_4(Q'))$ *EMGSS* $(x_1, x_2, \Gamma^l(i))$
if $(H_4(Q') > H_2(Q'))$ *EMGSS* $(x_1, x_4, \Gamma^l(i))$
elseif $(H_3(Q') > H_1(Q')$
if $(H_2(Q') > H_4(Q'))$ *EMGSS* $(x_2, x_3, \Gamma^l(i))$
if $(H_4(Q') > H_2(Q'))$ *EMGSS* $(x_4, x_3, \Gamma^l(i))$
End EMGSS
End For
$Q' \leftarrow Q$ *for all* $i \in Q \ \forall q_i \neq 0$
Update Q
If $Q' = \Phi$ *then END*
Else repeat Iteration

The computation of entropy $H_1(X)$ and $H_2(X)$ takes $O(|\Gamma^l(i)|)$ times, while the whole recursive function EMGSS(•) is repeated $\log(n')$ times. For $|Q|$ missing arc volumes to be estimated, the complexity of the algorithm is therefore $O(|Q||\Gamma^l(i)|\log(n'))$. Since $n' \gg |\Gamma(i)|$ and $|Q| \gg |\Gamma^l(i)|$, we can simplify the expression to $O(|Q|\log(n'))$.

A.2. List of symbols

$\beta(x_v, x_j)$	a function that compares an estimated value x_v with an observed arc-flow x_j that falls within the band gap, i.e., $j \in \Gamma(i)$
$\Gamma^l(i)$	the region of influence for arc i, with elements k representing both the upstream and downstream arcs among the set of lth-order neighbors for arc i, $\{l\}$
$H_v(X)$	Stirling's approximation of an entropy for the band-gap range of random variables X, with x_v being the estimated value
$\mu_U^l(i)$	weighted average of arc flows from the upstream 1st, 2nd, … and lth neighbors of arc i
$\mu_D^l(i)$	weighted average of arc flows from the downstream 1st, … and lth neighbors of arc i
n'	the number of search iterations in the EMGSS algorithm
q_i	unknown flow on arc i to be estimated
$Q = \{q_i\}$	set of arc volumes to be estimated
$\overline{Q}^l = (q_1, \bar{q}_2 \cdots \bar{q}_m)^T$	a one-dimensional column vector of observed traffic flows $\bar{q}_l$ in the lth order for every arc i, where q_l is non-zero for observed flows and zero for unobserved flows $(i = 1, 2, ..., m)$
$\widetilde{W}^l = [w_{ij}^l]$	an $m \times m$ matrix of the lth-order neighbors with weight entries w_{ij}^l denoting the relationship between neighboring arcs i and j, where $1 \leqslant w_{ij}^l \leqslant 1$
$\overline{W}_U^l(i) = (\leftarrow w_{ij}^l \rightarrow)$	normalized weights of row vector of the lth-order upstream-neighbors of arc i
$\overline{W}_D^l(i) = (\leftarrow w_{ij}^l \rightarrow)^T$	normalized weights of a column vector of the lth-order downstream-neighbors of arc i
X	a range of random variables within which the unknown arc traffic is to be estimated from the set of observed arc traffic x_j that falls within the band-gap.

References

Avriel, M., Wilde, D.J., 1966. Optimality proof for the symmetric Fibonacci search technique. Fibonacci Quarterly 4, 265–269.

Ban, X., Herring, R., Margulici, J.D., Bayen, A., 2009. Chapter 34 – optimal sensor placement for freeway travel time estimation. In: Lam, W.H.K., Wong, S.C., Lo, H.K. (Eds.), Transportation and Traffic Theory 2009: Golden Jubilee. Springer, Heidelberg.

Barcelo, J., Gillieron, F., Linares, M.P., Serch, O., Montero, L., 2013. Exploring link covering and node covering formulations of detection layout problem. Transportation Research Record: Journal of the Transportation Research Board 2308, 17–26.

Castillo, E., Conejo, A.J., Menéndez, J.M., Jiménez, P., 2008. The observability problem in traffic network models. Computer-Aided Civil and Infrastructure Engineering 23, 208–222.

Gentili, M., Mirchandani, P.B., 2011. Survey of models to locate sensors to estimate traffic flows. Transportation Research Record: Journal of the Transportation Research Board 2243, 108–116.

Giffin, A., Caticha, A., 2007. Updating Probabilities with Data and Moments. Presented at the 27th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, July 8–13. Saratoga Springs, NY.

Golan, A., 1998. Maximum entropy likelihood and uncertainty: a comparison. In: Erikson, G.J. (Ed.), Maximum Entropy and Bayesian Methods. Kluwer Academic Publishers, Boston.

GoogleMaps, 2009. <http://maps.google.com/> (accessed 26.02.09).

Hu, S.-R., Peeta, S., Chu, C.-H., 2009. Identification of vehicle sensor locations for link-based network traffic applications. Transportation Research Part B 43, 873–894.

Jaynes, E.T., 1982. On the rationale of maximum entropy methods. Proceedings of the IEEE 70, 939–952.

Kamarianakis, Y., Prastacos, P., 2005. Space-time modeling of traffic flow. Computers & Geosciences 31, 119–133.

Kianfar, J., Edara, P., 2010. Optimizing freeway traffic sensor locations by clustering global-positioning-system-derived speed patterns. IEEE Transactions on Intelligent Transportation Systems 11, 738–747.

Liu, H., Danczyk, A., 2008. Optimal detector placement for freeway bottleneck identification. In: Compendium of Papers CD-ROM, 87th Annual Meeting of the Transportation Research Board, Washington, DC.

Min, W., Wynter, L., Amemiya, Y., 2011. Road traffic prediction with spatio-temporal correlations. Transportation Research Part C 19, 606–616.

Ng, M.W., 2012. Synergistic sensor location for link flow inference without path enumeration: a node-based approach. Transportation Research Part B 46 (6), 781–788.

Qin, X., Wang, Y., Lu, H., 2005. A K-shortest-paths-based algorithm for stochastic traffic assignment model and comparison of computation precision with existing methods. Proceedings of the Eastern Asia Society for Transportation Studies 5, 1218–1232.

Shannon, C.E., 1948. A mathematical theory of communication. The Bell System Technical Journal 27, 379–423.

Tobler, W.R., 1965. Computation of the correspondence of geographical patterns. Papers and Proceedings of the Regional Science Association 15, 131–139.

Zhang, L., 2011. Behavioral foundation of route choice and traffic assignment: comparison of principles of user equilibrium traffic assignment under different behavioral assumptions. Transportation Research Record, Journal of the Transportation Research Board 2254, 1–10.

Zhang, L.-Y., Peng, Z.-R., Sun, D.J., Liu, X., 2012. Rule-based forecasting of traffic flow for large-scale road networks. Transportation Research Record, Journal of the Transportation Research Board 2279, 3–11.