18th International Symposium on Transportation and Traffic Theory (ISTTT)
Honk Hong, July 2009

Optimal Sensor Placement for Freeway Travel Time Estimation

Xuegang (Jeff) Ban, Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute; Ryan Herring, Department of Industrial Engineering and Operations Research, University of California, Berkeley; JD Margulici, California Center for Innovative Transportation, University of California, Berkeley; Alexandre M. Bayen, Department of Civil and Environmental Engineering, University of California, Berkeley

Abstract This article presents a modeling framework and a polynomial solution algorithm for determining optimal locations of point detectors used to compute freeway travel times. First, an objective function is introduced to minimize the deviation of estimated and actual travel times of all individual sub-segments of a freeway route. By discretizing the problem in both time and space, we formulate it as a dynamic programming model, which can be solved via a shortest path search in an acyclic graph. Numerical examples are provided to illustrate the model and algorithm using microscopic traffic simulation and GPS data from the *Mobile Century* experiment recently conducted by the University of California, Berkeley, Nokia and California Department of Transportation (Caltrans).

1. Introduction

Intelligent Transportation Systems (ITS) applications rely on *data* to characterize traffic states such as flow or speed. The data are usually collected from traffic *sensors*. For example, freeway travel time estimation often requires speeds measured at specific locations. Traditionally, a large portion of traffic sensors were deployed on a case by case basis by practitioners without a systematic study of the quantity and locations of sensors.[1] Since traffic sensors are limited resources, determining optimal placement strategies maximizes the value of the resource.

In this article, we study the optimal sensor placement problem for providing freeway travel times. The travel time application is selected because travel time is one of the most useful roadway traffic metrics to both traffic management agencies and the driving public. First, travel time is a direct measure of traffic conditions and system performance. Travel time reliability has received particular attention from both researchers and practitioners (Li et al, 2007; AlDeek et al, 2006;

1 One exception is the optimal sensor location problem for origin-destination matrix estimation, which has been well studied before (Yang and Zhou, 1998; Bianco et al., 2001; Chen et al., 2007). Mirchandani et al. studied the real time network performance monitoring problem (2007).

Chen et al, 1999). Second, travel times represent information that is easy to understand and process. Numerous studies reveal that commuters value travel time, which reduces their uncertainty and stress (Lindveld et al., 2000; Khattak et al., 1994). For example, Khattak et al. (1994) found that most drivers would divert under unexpected congestion if provided with real time traffic information on their usual route plus *travel times* on alternative routes. Furthermore, relevant traffic information enables travelers to make educated choices about mode and departure time choices, which may result in a form of "system self-management."

In the past, numerous studies have contributed to algorithmic techniques to estimate travel times from available field data, which generally came from loop detectors (Rice and Zwet, 2001; Coifman, 2002). Most studies assumed given detector locations and proposed optimal ways of processing the data. The optimal sensor placement problem in this regard has not been widely studied. Existing research focused on empirical investigations of the impact of sensor locations on the quality of travel time estimation. By taking out existing loop detectors in a predefined way, Fujito et al. (2006) found that travel time estimation quality may not always decrease as detector spacing increases. Kwon et al. (2006) studied how the travel time estimates vary as the number of detectors changes by randomly taking out detectors. They concluded that 0.5 mile sensor spacing is appropriate for providing freeway travel times. Ban et al. (2007) showed that as sensor spacing increases, travel time estimation becomes more sensitive to actual sensor locations. This implies that the optimal sensor location problem is more critical if one aims to deploy a limited number of sensors on a relatively long freeway segment. The above studies are usually available for freeways. Thomas (1999) studied sensor location problems for arterial streets using microscopic traffic simulation.

Relatively little research has been devoted to develop computationally tractable methods for optimal sensor placement for travel time estimation. Eisenman et al. (2006) provide a conceptual framework of the sensor location problem for traffic detection systems. Sherali et al. (2006) propose a mixed-integer optimization model to determine optimal placement of vehicle identification readers for travel time estimation, although the model can only be solved approximately. Bartin et al. (2007) show that the optimal sensor placement for travel time estimation can be determined by minimizing the weighted summation of speed variations of all roadway segments, each associated with a sensor. A *Nearest Neighbor* (NN) algorithm was further developed. However, the NN algorithm cannot generate a globally optimal solution in polynomial time.

We address two major issues which are key to the problem of optimal sensor placement for travel time estimation.

- How to develop a model and algorithm to efficiently solve (in polynomial time) a sensor placement problem to minimize travel time estimation error,
- How to modify the algorithm to handle corridors where sensors are already deployed in such a way as to optimally supplement the existing sensors.

The present article focuses on the above two issues. We develop a modeling framework that uses vehicle trajectory data to perform the analysis. With the advent of GPS-enabled smartphone-based traffic monitoring, our algorithm can be

used with probe vehicle data, which we illustrate using the Mobile Century data set, presented in the last part of this article.

By discretizing both time and space, we first show that the optimal sensor location problem for travel time estimation can be formulated as a *Dynamic Programming* (DP) model with one sensor deployed at each stage. The model can be further represented as an acyclic graph and solving the problem is equivalent to find the shortest path in the graph. We then prove that such a search can be done in polynomial time, which can be used for solving large scale problems or for deploying sensors to many freeway segments. We also show that incorporating sensors that have already been deployed can be easily done via revising the graph representation of the DP model, and the solution complexity remains the same.

Distinct from most previous studies, we test the model and algorithm using both simulation data and GPS-equipped cellular phones. The results show that to have better travel time estimation, sensors should be deployed to cover major bottleneck areas and free-flow regimes. As more sensors are available, they should be placed at bottlenecks areas, while a single sensor is usually sufficient for free-flow areas. Compared with random sensor configurations and evenly spaced sensors, the DP solution has minimum estimation error and is more stable and predictable.

2. Preliminaries

The problem studied in this article can be stated as: given a freeway segment (called route r) and a number of fixed-location sensors (such as loop detectors), where should these sensors be placed so that the deployment is "optimal" in terms of providing travel time estimates? Here we assume the number of sensors is given (denoted as K), which may be determined by budget constraints. Or one can always solve the problem for different numbers of sensors and pick the one with the desired performance. The efficiency of our proposed algorithm in this article makes solving the problem multiple times tractable. Similar to other engineering problems, the answer to the above problem depends on several factors. Especially, there are numerous methods available to compute travel times and sensors can usually provide multiple types of data. Therefore, determining optimal sensor placement depends on the travel time estimation method and the sensor data type. This section discusses assumptions made in the article to address these concerns, most of which are consistent with what is currently used in practice.

2.1 Travel Time Estimation Methods

To be consistent with current practice, we assume that travel times are calculated using aggregated sensor speeds. Speeds can be obtained directly from double loop detectors or estimated from single loop detectors (Jia et al., 2001). We assume that every sensor has a spatial "influence area," called a *link*. Sensor speed

represents the (uniform) speed of the entire link associated with the sensor. There are a number of ways to define how a sensor is associated with its link (for example, PeMS defines a link as the segment between the middle points of two sensors (BTS04, 2004, pp. 3-1)). In this article, we assume a sensor is always in the middle of its corresponding link[2]. Different link definitions lead to slightly different ways to interpret sensor speeds, which in turn might result in small variations in travel time calculation. These variations however should not be significant.

As a result, the to-be-deployed *K* sensors divide the study route into *K* links, and the route travel time is the summation of link travel times. Note that such a definition will effectively eliminate certain travel time estimation methods based directly on routes (Rice and Zwet, 2001). However, it is widely used in practice (see for example BTS04, 2004, pp. 3-23). More importantly, the DP model presented in this article does not depend on how link travel times are calculated. This implies much flexibility regarding which travel time method to use in the model.

In this article, we focus on two specific travel time computation methods: the *instantaneous* method (Ban et al., 2007) and *Coifman* method (Coifman, 2002). The instantaneous method assumes that traffic conditions remain unchanged from the time a vehicle enters a route until it leaves the route. Therefore, travel time of the route can be computed by summing the travel times of the constituent links at the time a vehicle enters the route. This method is "naïve" in the sense that traffic condition changes are not considered; however, it is probably the most widely used method in practice due to its simplicity and the fact that it can be used in real time. The second method, originally developed in Coifman (2002), is a more sophisticated algorithm for calculating link travel times. The method constructs vehicle trajectories from sensor speeds using traffic flow theory, from which link travel times can be extrapolated. We use the instantaneous method in most parts of the article to illustrate the DP model and the solution algorithm. However, we discuss how Coifman method can also be considered in the model and solution method. In Section 5, we show results from the Coifman method, and provide comparisons with results obtained using the instantaneous method.

2.2 The Objective Function

We assume that trajectories of a certain number of vehicles (assumed to be *M*) are available. We denote $\hat{\tau}_k^m$ and τ_k^m the estimated and actual travel times of the

[2] One may argue that restricting sensors to be only in the middle of its link can potentially filter out better solutions. This issue was considered previously by some researchers. For example, a probability distribution was assumed in Bartin (2007), which describes the probability that a sensor will be deployed to each discretized section of a link. However, in practice, after sensors are deployed, practitioners need a straightforward way to define the link associated with each sensor. If sensors are allowed to be deployed arbitrarily within a link, the link boundary will have to be recorded to compute travel times. We argue that this is highly impractical in reality.

m-th vehicle ($1 \le m \le M$) traveling link *k* ($1 \le k \le K$), respectively. The travel time estimation error for the *m*-th vehicle on link *k* denoted as e_k^m, can be expressed as:

$$e_k^m = \hat{\tau}_k^m - \tau_k^m . \quad (1)$$

We use the same objective function as that in Bartin (2007) as follows:

$$\hat{E} = \sum_{i=1}^{M}\sum_{k=1}^{K}\left(e_k^m\right)^2 / M = \sum_{k=1}^{K}\hat{E}_k . \quad (2)$$

Here $\hat{E}$ represents the objective function, and $\hat{E}_k$ is the *Mean Square Error* (MSE) of the travel time estimation for all *M* vehicles on link *k*, defined as:

$$\hat{E}_k = \sum_{i=1}^{M}\left(e_k^m\right)^2 / M . \quad (3)$$

The objective defined in (2) focuses on estimation errors of all individual links, instead of only on the entire route. The reason for this is that we want to generate sensor locations that can provide accurate estimates of all link travel times, not only in terms of the entire route. If attention is only put on the entire route, it is possible that the resulting sensor locations may underestimate travel times for certain links and overestimate for other links, but as a whole, they cancel out each other and provide good estimation. This type of sensor placement is not desirable. It is easy to see that the objective function we use here can effectively eliminate such sensor deployment strategies since they will lead to large objective values using equation (2). In Section 5, we show that this definition is effective to generate sensor placement that is optimal for both the (entire) route *r* and its sub-routes.

3. A Dynamic Programming Formulation

3.1 Mean Square Error of A Link

We divide route *r* into small segments, called *sections*. We assume that if the length of a section is sufficiently small, speed will not change within the section and it does not matter where to place a sensor within the section. Thus we only need to determine where to deploy the given *K* sensors to these small sections. Assume the length of each section is Δx and that the given route *r* can be divided into *N* sections. We use n=1, ···, *N* to index a given section. A link then contains one or more sections, and the link boundaries are at the section boundaries. Since we assume a sensor is always in the middle of its link, the sensor deployment problem is now converted to determine the optimal *starting and ending indices* of the *K* links that comprise route *r*. In the time domain, we evenly divide time into *intervals* with length ΔT = 30 seconds as we assume sensors can only provide 30-sec average speeds. Assume the entire study period can be divided into *H* time intervals and h=1, ···, *H* is used to index a given interval. For simplicity, we assume route *r* starts with x=0 and time starts with t=0.

This space and time discretization is illustrated in Fig. 1. It is clear from the figure that the two-dimensional *x-t* space is divided into a grid of *sensor boxes*. Each sensor box represents a data collection unit (particularly for speeds in this article) at a specific location (section), which is only active for the designated time period (30-sec long). The average speed of each sensor box can be computed via available vehicle trajectories, and is defined as the average speed of all vehicles that pass the sensor at the specific time period. This mimics the way loop detectors collect average speeds in practice. Calculating the average speed for any sensor box (*n*, *h*) with *n*=1, ⋯, *N*, *h*=1, ⋯, *H* of the route will result in the *speed field* (also called *speed contour map*, see Ban et al., 2007) of the study route for the study period. Fig. 6(a) depicts the speed field of the micro-simulation data in this article. Notice that in case "blank" sensor boxes exist, for which no vehicle passes by, we estimate the speed as the average speed of all its surrounding sensor boxes whose speeds are already available. Fig. 11 (a) illustrates the estimated speed field using trajectories from 100 vehicles equipped with GPS cell phones.

Assume the *k*-th link starts at section s_k and ends at section $y_k \geq s_k$. Both s_k and y_k are integers to represent a section. Note that the starting and ending sections are inclusive, i.e., link *k* includes both sections s_k and y_k. This is illustrated in Fig. 1.To calculate the MSE of link *k* as expressed in equation (3), we focus on the given *M* vehicles. For any *m*-th vehicle, Fig. 1 depicts, in a solid thin line, the trajectory of the vehicle. In the figure, we denote $\tau^m_{s_k,y_k}$ the actual travel time of the vehicle traversing link *k*. The corresponding estimated travel times are denoted as $\hat{\tau}^{m,i}_{s_k,y_k}$ (instantaneous) and $\hat{\tau}^{m,c}_{s_k,y_k}$ (Coifman). It can be seen that $\tau^m_{s_k,y_k}$ can be expressed as $\tau^m_{s_k,y_k} = t^m_{y_k\Delta x} - t^m_{(s_k-1)\Delta x}$, where t^m_x denotes the time when the *m*-th vehicle passes location *x*. Suppose a sensor is deployed on the *k*-th link. Based on our assumptions, the sensor will be in the middle of the link. Denote n_k the section that the sensor on link *k* is located, we have:

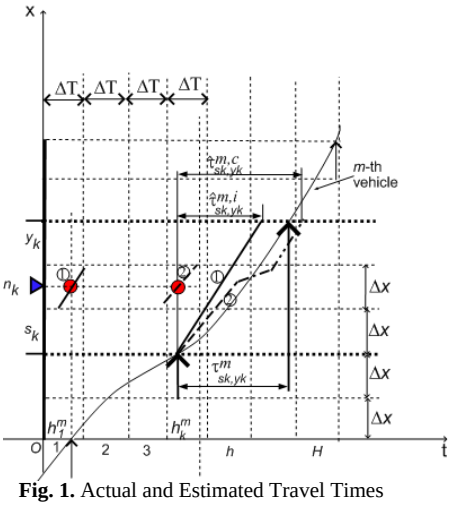

Fig. 1. Actual and Estimated Travel Times

$$n_k = \lfloor (s_k + y_k)/2 \rfloor \quad (5)$$

Here $\lfloor . \rfloor$ denotes the *rounding* operator. Assume the *m*-th vehicle enters route *r* at time interval h_1^m and it enters section s_k at time interval h_k^m. Then according to the definitions of the instantaneous travel time, the average speed of the sensor box (n_k, h_1^m), denoted as v_{n_k,h_1^m}, will be used for computing the instantaneous tra-

vel time of link *k*. This is shown as the solid bold line in Fig.1, which is marked as "1". Noticing that $(y_k - s_k + 1)\Delta x$ is the length of link *k*, we can compute the instantaneous travel time as:

$$\hat{\tau}^{m,i}_{s_k,y_k} = \frac{(y_k - s_k + 1)\Delta x}{v_{n_k,h_1^m}} \tag{6}$$

For the Coifman method, the vehicle trajectory can be estimated by a piecewise linear curve using traffic flow theory. This is shown as the bold dash line in Fig. 1 (marked as "2"). The Coifman link travel time for link *k* does not have a closed form expression. However, it only depends on the starting and ending sections of link *k* provided speeds of all sensor boxes are given, and the entrance time is assumed to be $t^m_{(s_k-1)\Delta x}$. Denote $\hat{E}^i_k$ and $\hat{E}^c_k$ the MSE of travel time estimation for link *k* for instantaneous and Coifman travel times, respectively. Following equation (3), they are functions of s_k, y_k and can be expressed as:

$$\hat{E}^i_k(s_k,y_k) = \frac{\sum_{m=1}^{M}(\hat{\tau}^{m,i}_{s_k,y_k} - \tau^m_{s_k,y_k})^2}{M} = \frac{\sum_{m=1}^{M}[\frac{(y_k - s_k + 1)\Delta x}{v_{n_k,h_1^m}} - t^m_{y_k\Delta x} + t^m_{(s_k-1)\Delta x}]^2}{M} \tag{7}$$

$$\hat{E}^c_k(s_k,y_k) = \frac{\sum_{m=1}^{M}(\hat{\tau}^{m,c}_{s_k,y_k} - \tau^m_{s_k,y_k})^2}{M} \tag{8}$$

The above procedures for calculating link MSE show that link MSE only depends on the starting and ending sections of the link, i.e., s_k and y_k. The calculation is independent of how the (*k*-1) sensors for the previous (*k*-1) links are deployed once s_k and y_k are known. This motivates us to formulate the optimal sensor placement problem using dynamic programming.

3.2 Dynamic Programming Model

Denote $\hat{E}^i$ and $\hat{E}^c$ the objective functions for instantaneous and Coifman travel times respectively. We will have, according to (2):

$$\hat{E}^i = \sum_{k=1}^{K}\hat{E}^i_k(s_k,y_k) \tag{9}$$

$$\hat{E}^c = \sum_{k=1}^{K}\hat{E}^c_k(s_k,y_k) \tag{10}$$

We see that the objective functions for instantaneous and Coifman travel times are similar, and the only difference is which link MSE to use. In the reminder of this section, we use the instantaneous travel time to illustrate the DP model.

Given the objective function, the optimal sensor location problem can be stated as: find the optimal values of $s_k, y_k, k = 1,\cdots,K$ such that (9) can be minimized. That is, one needs to solve the following optimization problem:

$$\min_{1\le s_k\le y_k\le N, k=1,\cdots,K} \sum_{k=1}^{K} \hat{E}_k^i(s_k, y_k) \tag{11}$$

Subject to constraints (12) – (15) below.

The above optimization model is a linear integer program since $\hat{E}_k^i(s_k, y_k)$ is computable for any (k, s_k, y_k), and (s_k, y_k) are integer-valued. Directly solving the model may not be easy if the problem dimension is large. We thus divide the problem into stages: at each stage, the optimal location of one sensor is obtained, which can be achieved by finding the optimal starting and ending locations of its associated link. We assign the starting location (section) of link *k* (i.e. s_k) as the state variable, and the ending location of link *k* (i.e. y_k) as the decision variable.

We first look at the constraints for s_k and y_k. Clearly, we have

$$s_1 = 1 \tag{12}$$

$$y_K = N \tag{13}$$

This means that the first link must start at section 1 and the last link (link *K*) must end at section *N*. Also, we have the state transfer function as

$$s_{k+1} = y_k + 1 \tag{14}$$

That is, knowing the ending section of link *k* (y_k), the starting section of link (*k*+1) must be the next section. Since a link contains at least one section, we have

$$k \le s_k \le y_k \le N - K + k \tag{15}$$

The first inequality holds since there are *k*-1 links before link *k*, which contain at least *k*-1 sections. Similarly, the last inequality holds since there are *K*-*k* links after link *k*, which contain at least *K*-*k* sections. Equations (12)-(15) show that there is only one possible state for stage 1 ($s_1 = 1$), but multiple states for stage $k \ge 2$. In particular, (15) means that the possible states for any stage $k \ge 2$ are from *k* to *N*-*K*+*k*, i.e. the total number of states is *N*-*K*+1.

At any stage *k*, the cost of deploying a sensor is the link MSE $\hat{E}_k^i(s_k, y_k)$, which is consistent with the objective function (9) and (2). Since $\hat{E}_k^i(s_k, y_k)$ is only a function of (s_k, y_k), the optimal value of y_k can be obtained by minimizing $\hat{E}_k^i(s_k, y_k)$ if s_k is known. In particular, if we denote $F_k(s_k)$ as the total cost from stage *k* (including stage *k*) to the last stage (i.e. stage *K*), a recursive formulation for $F_k(s_k)$ can be given as:

$$F_1(s_1) = F_1(1) = \min_{1\le y_1\le N-K+1} \{\hat{E}_1^i(1, y_1) + F_2(y_1 + 1)\}, \tag{16}$$

$$F_k(s_k) = \min_{s_k\le y_k\le N-K+k} \{\hat{E}_k^i(s_k, y_k) + F_{k+1}(y_k + 1)\}, 2 \le k \le K-1, \tag{17}$$

$$F_K(s_K) = \hat{E}_K^i(s_K, N). \tag{18}$$

The above equations are for stage 1, stage $k \in \{2, \cdots, K-1\}$, and stage *K* respectively. First, due to (12), we have $F_1(s_1) = F_1(1)$ for stage 1, which is a summation of the cost of stage 1 (i.e. $\hat{E}_1^i(1, y_1)$) and that from stage 2 to stage *K* (i.e. F_2

). For stage $2 \le k \le K-1$, the cost F_k is a function of the state variable s_k, which is the summation of the cost of the current stage *k* and that from stage *k*+1 to the last stage. Note that in both equations, the starting location of the next stage (i.e. stage 2 or *k*+1) is the *immediate next* section of the ending location of current stage due to (14). For the last stage, since the ending location must be *N*, $F_K(s_K)$ is automatically computable given s_K.

We can observe that (i) all constraints (12) - (15) are satisfied in the above equations and there are no extra constraints introduced, (ii) $F_1(1) = \sum_{k=1}^{K} \hat{E}_k^i(s_k, y_k)$. Hence solving (11) is equivalent to solve (16) – (18). Furthermore, from these recursive equations, we can see that if (s_k^*, y_k^*), $1 \le k \le K$ is an optimal solution, (s_k^*, y_k^*), $k_1 \le k \le k_2$ must be an optimal solution from stage $k_1 \ge 1$ to stage $k_2 \le K$. This illustrates that the *optimality principle* (Bellman, 1962) holds for the model (16) – (18). Therefore, the model is a DP problem.

The proposed DP model is for both the instantaneous and Coifman travel times due to the calculations of their link MSEs in Section 3.1. In fact, it is easy to see that the DP model can be used for any other link travel time methods as long as the methods only depends on the starting and ending locations of the link.

4. Solution Algorithm and Complexity

We present a graph representation of the DP model. We start from the case in which there is originally no sensor on the freeway and one needs to deploy *K* sensors. We then show in Section 4.3 how the graph can be revised to incorporate the case in which there are *K*' < *K* existing sensors.

4.1 A Graph Representation

A graph representation of the DP model is depicted in Fig. 2, where stages are listed horizontally and sections are listed vertically. Since we deploy one sensor per stage, we also associate each link with a stage. Based on the DP model in Section 3.2, the state of a stage represents the starting section of the link associated with the stage. In this figure, all possible states of a stage are represented as *nodes*. That is, a node represents a section of the roadway, and the node number is the section number. For example, the node at stage 2 and Section 2 represents that the starting location of link 2 could be section 2. As shown in equations (12)-(15), there is one state in stage 1 (s_1 = 1) and (*N-K+1*) states (from *k* to *N-K+k*) for stage $k = 2, \cdots, K$. We create a fake stage (*K*+1) that has one fake state *N*+1.

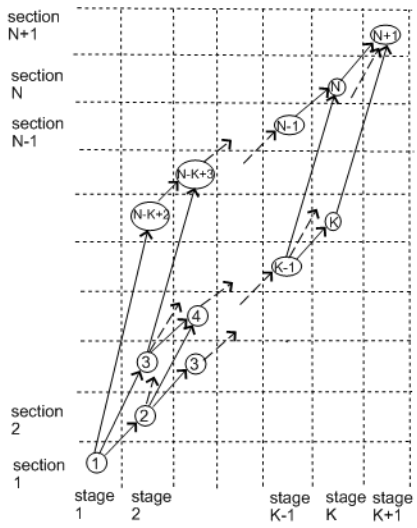

Fig. 2 Graph Representation of DP Model.

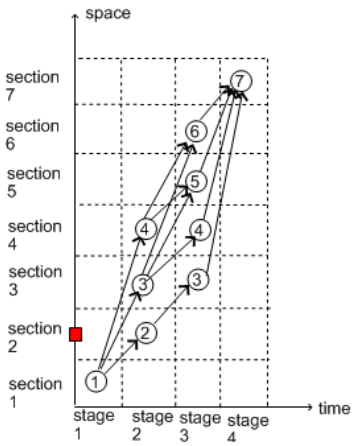

Fig. 3 Consideration of Existing Sensors

A connection, denoted as an *arc*, may be created from a node in stage k to another node in the immediate next state k+1 if the latter node has a higher node number. Each arc actually represents a possible roadway link by defining the link's starting and ending sections. An arc from node s_k in stage k to node s_{k+1} in stage k+1 represents one possible configuration for link k: it starts at section s_k and ends at section s_{k+1}-1 (because the next link starts at s_{k+1}). Therefore, we must have $s_{k+1} > s_k$ in order to construct the arc. For example, the arc from node 2 in stage 2 to node 4 in stage 3 (marked in bold line) in Fig. 2 represents one possible configuration for link 2: it starts at node 2 and ends at node 3 (both are inclusive). There are no arcs between any two stages that are not adjacent to each other. We associate a cost with each arc in Fig. 2. For the arc from node s_k in stage k to node s_{k+1} in stage k+1, the arc cost is $\hat{E}_k^i(s_k, s_{k+1}-1)$ as shown in (7). That is, the cost of an arc is the MSE of travel time estimation for its corresponding roadway link.

It is easy to check that the graph constructed in the above manner enumerates all possible states in each stage and all possible configurations of each link. It also incorporates all the constraints of the model shown in equations (12)-(15). Furthermore, each path from node 1 in stage 1 to node N+1 in stage K+1 contains exactly K arcs, each of which represents a possible configuration of a particular roadway link. In other words, each path represents a potential sensor deployment scenario. Thus the optimal sensor locations can be achieved by finding the minimum-cost path from node 1 in stage 1 to node N+1 in stage K+1. As arc costs are positive, the DP model can be solved by a shortest-path algorithm.

4.2 Complexity of the Algorithm

The complexity of the shortest path search algorithm depends on the structure and size of the graph in Fig. 2. The following theorem provides its complexity.

Theorem 1. The DP model can be solved in polynomial time. In particular, the complexity of solving the DP model is $O((K-2)(N-K+1)^2)$.

Proof. From the way the graph is constructed in Section 4.1, all arcs are from lower-numbered stages to higher-numbered stages and from lower-numbered nodes to higher-numbered nodes. Therefore, the graph is acyclic. Since the DP model can be solved via a shortest path search from node 1 in stage 1 to node N+1 in stage K+1 in an acyclic graph, the complexity of the shortest-path search is linear in terms of the number of arcs in the graph (Bertsekas, 1998). The number of arcs in the graph in Fig. 2 can be easily calculated: from stage 1 to stage 2 or from stage K to stage K+1, there are N-K+1 arcs. Between any other two stages, there are $(N-K+1)^2/2$ arcs. Therefore, the total number of arcs in the graph is: $2(N-K+1)+(K-2)(N-K+1)^2/2$. As a result, the complexity of the solution algorithm is $O((K-2)(N-K+1)^2)$, which is polynomial.

It is easy to see that Corollary 1 below immediately follows Theorem 1.

Corollary 1. If $N>>K>>2$, the complexity of the DP algorithm is $O(KN^2)$.

Theorem 1 and Corollary 1 states that the complexity of solving the DP model depends linearly on the number of sensors and quadratically on the number of sections. Furthermore, it does not depend on the number of time intervals. This implies that the proposed model can be efficiently solved, even for large-scale problems. In addition, the DP model produces the exact solution for the optimal sensor location problem. Hence, at least in theory, the DP model and solution algorithm are more efficient than previous methods.

4.3 Consideration of Existing Sensors

It may be operationally useful to find an optimal way to add additional sensors to a highway segment that already contains existing sensors. In this case, we make a simple adjustment to the DP graph representation. First, we match all existing sensors to the appropriate section they reside in. Then, every possible link (represented as an arc in the graph) that covers a section with an existing sensor in it but does not have the existing sensor at the center of the link is removed from consideration. This is because we assume a sensor is in the middle of its link.

As an example, Fig. 3 shows a highway section that is broken down into 6 sections. Suppose that we already have a sensor in section 2. Then we cannot consider links that cover section 2 but do not have section 2 as the middle of the link. This means that a link covering sections 1 through 4 would not be permissible in the solution (because that would imply a sensor in 3 and not on section 2 according to equation (5)). On the other hand, a link covering sections 1 through 3 would

be permissible, implying that the arc from node 1 in stage 1 to node 4 in stage 2 is included. This is also the case for the arc from node 1 in stage 1 to node 2 in stage 2 (represents a link that only contains section 1), and the arc from node 1 in stage 1 to node 3 in stage 2. Similarly, the arcs from node 2 in stage 2 to nodes 4, 5, and 6 in stage 3 should all be eliminated. The graph in Fig. 3 shows the adjusted DP graph after removing all impermissible links.

Therefore, to account for existing sensors, one can use a simple linear search on all of the links to identify which ones to remove. The shortest path algorithm can then be used to compute the solution on the adjusted graph. As a result, the complexity of the algorithm remains the same as the original DP algorithm.

5. Case Studies

We illustrate the proposed DP model and solution algorithm using two case studies. The first case study is based on micro-simulation, which provides an ideal analysis framework since all individual trajectories are known. This allows us to investigate for example how the penetration rate will impact the sensor location quality. The second case study is based on vehicle trajectory data from GPS-enabled cellular phones, obtained from the Mobile Century field experiment (Amin et al., 2008; Work et al, 2008). The goal of the experiment was to test the ability of using GPS cell phones to collect and disseminate traveler information.

5.1 Case Study I: Micro-Simulation Data Set

The micro-simulation is for an 8.7-mile freeway segment along I-880 in the San Francisco Bay Area. Fig. 4 provides an overview of the network in Paramics, in which “1” and “3” indicates respectively the origin and destination of the route. The simulation model was developed as part of the Corridor Management Planning Demonstration project for Caltrans (CCIT07, 2007). Detailed descriptions on how the simulation network was constructed, calibrated, and validated can also be found in Ban et al. (2007). We run the simulation for 2 hours 30 minutes for morning peak hours from 5:30 am to 8:00 am. We then choose the last 2 hours as the study period, i.e. H=240. We divide the freeway segment into 100-foot sections, resulting in N=459. The “representative” vehicles are selected as those who traveled the entire segment and started their trips within the 2-hour study period. There are 3,586 such vehicles, i.e., M=3586. The average travel time of the vehicles is 796 seconds and standard deviation is 227 seconds.

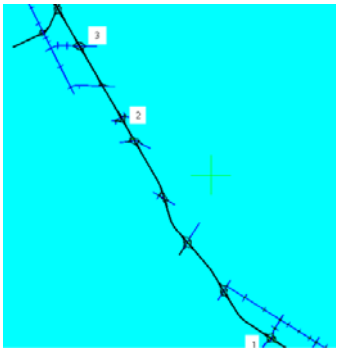

Fig. 4 Paramics Simulation Network

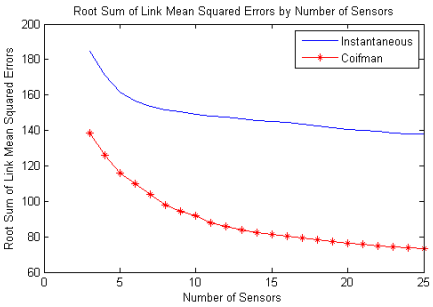

Fig. 5 DP Objective Values vs. Number of Sensors

We solve the shortest path problems on directed acyclic graphs in order to solve our DP model. We vary the number of sensors from K=3 to K=25, or equivalently an average spacing from about 3 miles to 0.3 mile. Fig. 5 depicts how the objective value computed by equation (2) decreases as the number of sensors increases from 3 to 25. The decrease is monotonic, but the marginal benefit decreases as well. We can also observe that the Coifman method usually has a smaller objective value than that of the instantaneous method. As one example, Fig. 6(a) depicts the obtained optimal sensor locations (marked using triangles on the y-axis in the figure) when K=6 using the instantaneous travel time method. In the figure, the speed contours of the segment are also displayed. We can observe that this freeway segment has two major bottlenecks, which are due to merging at about PM26.0 and PM 23.5 respectively. In the latter half of the simulation, the first bottleneck propagates backward and combines with the second one. In this sense, we can treat them as a single congested area, spanning from PM 26 to PM20. At PM 18.5, there is also a minor bottleneck for a short period of time (roughly from 7:20 am to 8:00 am).

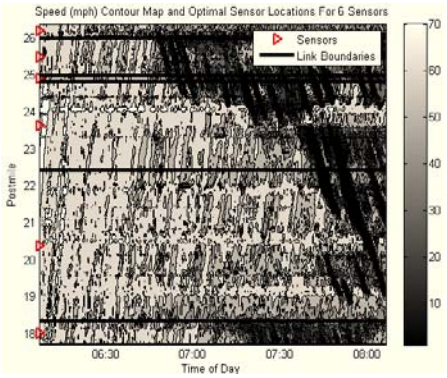

Fig. 6(a) Optimal Sensor Locations for Instantaneous Method for Simulation Data

The DP model, using the instantaneous method, puts four sensors at the major bottleneck area (PM 26.2, 25.5, 24.8, 23.5), one at the free flow regime (PM 20.4), and another one at the minor bottleneck (PM 18.0), which intuitively makes sense. We also ran the model using the Coifman method and the solution is shown in Fig. 6 (b). The Cofiman method generates similar results to the instantaneous method, i.e. four sensors in the congested area, one in the free flow area, and the last one in the minor bottleneck area. There are however some differences. Especially, the

Coifman method tends to be able to distinguish the two major bottlenecks by putting three on the first bottleneck and one on the second bottleneck. This is intuitive since the Cofiman method constructs travel times by "walking through" both the spatial and temporal domains and thus is able to capture the dynamic evolution of bottlenecks. The instantaneous method on the other hand only focuses on a snapshot of traffic conditions and is thus less sensitive to the actual shapes of bottlenecks.

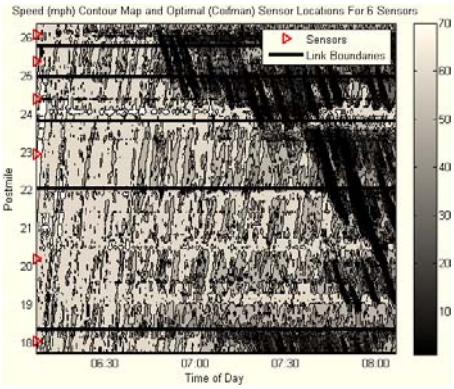

Fig. 6(b) Optimal Sensor Locations for Coifman Method for Simulation Data

To illustrate how bottleneck areas impact the optimal sensor locations generated by the DP algorithm, we show in Fig. 7 how the sensor locations change as we increase the number of sensors from 3 to 12, using the instantaneous travel time method. We can observe from this figure that when the number of sensors is small (e.g. K=3), they will be first deployed to major bottlenecks (i.e. PM 25.5 and PM 23.7). For the free-flow area, only one sensor is needed (i.e. at PM 20.1). As more sensors are available, they will be deployed to bottleneck areas to capture the complicated traffic conditions in bottlenecks. Also, as the number of sensors increases, minor bottlenecks may also be captured and enhanced by additional sensors, while usually one sensor is sufficient for free flow areas. More importantly, as additional sensors are added in, the locations of previously deployed sensors in bottleneck areas remain almost unchanged. This is illustrated using the thin lines in the figure, which show that locations of newly deployed sensors just "branch out" from existing sensors in bottleneck areas. This implies that the DP algorithm has the ability to capture the most significant bottlenecks and if more sensors are available, the second most significant bottlenecks can be captured and so on. The locations of sensors in free flow areas however may change since the speeds detected in free flow areas are not sensitive to the actual sensor locations. The evolution of optimal sensor locations for the Coifman method is similar to the one shown in Fig. 7. The above discussions illustrate the close relation between the optimal sensor locations generated by the DP algorithm and the bottleneck areas of the freeway. They also show that the results from DP are stable and predictable, which is desirable in practice.

The DP objective function defined in (2) focuses on the summation of all link MSEs. We next show that this produces reasonable sensor configurations to both the entire route and its sub-routes. First, to evaluate the performance of the entire route, we define the following objective function:

$$\overline{E} = \frac{\sum_{m=1}^{M}(\sum_{k=1}^{K} e_k^m / \sum_{k=1}^{K} \tau_k^m)^2}{M} \qquad (19)$$

Equation (19) defines the MSE on the entire route in a relative sense since $\sum_{k=1}^{K} t_k^m$ is the actual travel time of the *m*-th vehicle and $\sum_{k=1}^{K} e_k^m$ is the estimation error for that vehicle. To show that the DP results are also (nearly) optimal for the objective in (19), we compare such objective values of the DP solutions with those from 1,000 randomly generated sensor configurations for 2 to 25 sensors. The results are shown in Fig. 8 for the instantaneous method. In this figure, the solid line represents the average objective values across all random configurations with the best and worst random configurations represented by the ends of the error bars. The line with rectangle signs represents objective values via evenly spaced configurations, while the line with asterisks represents the objective values from the DP solution. Clearly, the DP solution curve is very close to or lower than the smallest objective values by all random configurations. This indicates that the DP solution significantly outperforms the random configuration even evaluated under (19) (note that the DP solution was generated using equation (9)). We can also observe that evenly spaced sensors cannot produce satisfactory travel time estimation especially when the number of sensors is small. For example, when the number of sensors is K=3, the objective value of the DP solution is 32%, while evenly spaced configuration produces 68% error. In addition, the performance of the evenly spaced configurations tends to vary significantly when the number of sensors varies. The performance of the DP solution however is very stable. These differences tend to reduce as the number of sensors increases. For example when K=25, the objectives values for the DP solution and evenly spaced configuration become 28% and 37% respectively. This indicates that optimal sensor placement is more critical for limited number of sensors than that for sufficient number of sensors (in this case, evenly spacing the sensors may work properly).

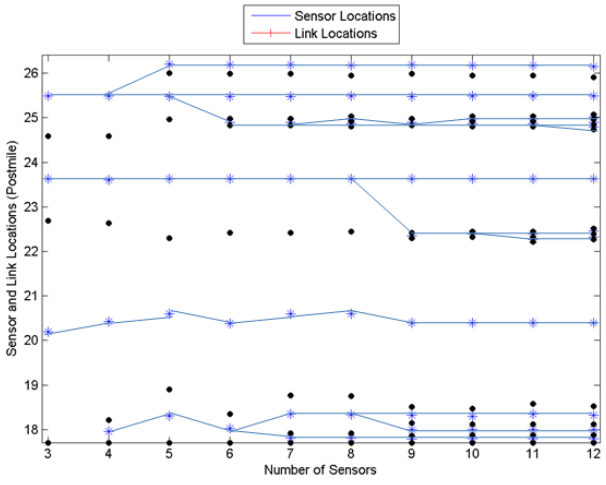

Fig. 7 Evolution of Optimal Sensor Locations (Instantaneous Method) for Simulation Data

We now focus on the sub-route as indicated in Fig. 4 using "2" and "3", which is about 2.5 miles. We particularly evaluate how the obtained DP solution and the best random configuration perform on this sub-route for each given number of sensors (2 to 25). For this purpose, we first select sensors that are deployed on this sub-route by the DP solution; we then calculate the objective value similar to that in (19) using these selected sensors on the sub-route. Also, we select the best random configuration for the entire route for each given number of sensors, and evaluate its objective value as in (19) for the sub-route. The results are depicted in Fig. 9. In the figure, the solid line with asterisks and the dash line represent, respectively, the objective values on this sub-route by the DP solutions and the best random configurations. The two curves show that the DP solution is consistently superior to the best random configuration on the sub-route. More importantly, the performance of the DP solutions is very stable across different numbers of sensors, while the random configurations tend to have varied performances depending on the actual number of sensors. Hence we can conclude that random configurations may perform well on the entire route, but may be poorly performed on sub-routes. The DP solution, however, performs well on both the entire route and the sub-routes. This is mainly due to the objective function used by the DP model, which focuses on individual links rather than just the entire route.

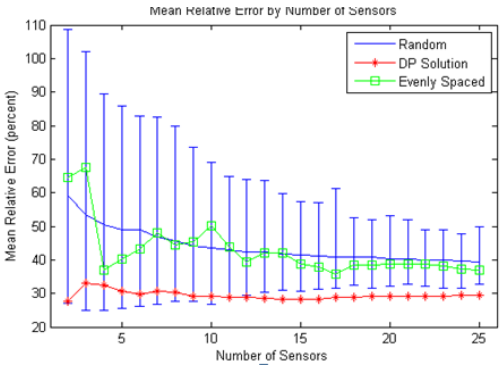

Fig. 8 Comparison of DP and Random Config. on the Entire Route for Simulation Data

In reality, it is almost impossible to measure trajectories of all vehicles traversing a given segment of freeway. Therefore, one crucial issue is to study how sampling rate impacts the results of the DP algorithm. Since DP solutions are closely related to bottleneck areas of the route as discussed above, we focus on the speed contour map for this purpose. In particular, we vary the sampling rate from 0.5% to 100%. For a given sampling rate α, we select each of the M total vehicles with probability α. The objective function is the root mean squared difference between the speed contour map by all the vehicles and that by the sampled vehicles, defined as follows:

$$E_s = \sqrt{\frac{\sum_{n=1}^{N}\sum_{t=1}^{T}(v_{n,t} - \hat{v}_{n,t})^2}{NT}} \qquad (20)$$

Here E_s denotes the average absolute error of two speed contour maps, $v_{n,t}$ and $\hat{v}_{n,t}$ are the average speeds for section n at time interval t computed using all vehicles and sampled vehicles respectively.

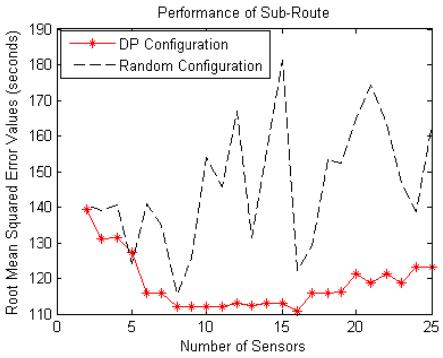

Fig. 9 Comparison of DP and Random Config. on the Sub-Route for Simulation Data

Fig. 10 shows that the error decreases quickly from 0.5% to 25%, after which point the error is less than 1 mph. At 5%, the error is less than 2 mph. Therefore, 5% to 10% is a reasonable range in which one would expect the speed maps by the sampled vehicles to be very close to that from all vehicles. This is consistent with previous studies (e.g. 4% is concluded as sufficient for travel time estimation in Sanwal and Walrand (1995)).

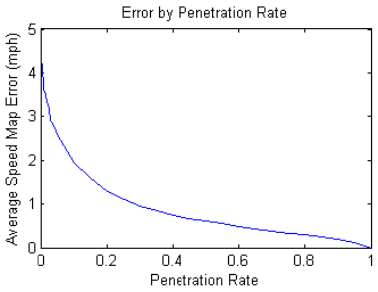

Fig. 10 Impact of Sampling Rate

5.2 Case Study II: Mobile Century GPS-Equipped Cellular Phones

We further test the DP model and algorithm on the Mobile Century data. Mobile Century is an experiment performed on February 8th, 2008, in which 165 drivers drove 100 vehicles on Interstate 880 (see Fig. 12) for 10 hours in loops of length 5.5 to 10 miles (Amin et al., 2008). The experiment involved each vehicle carrying a Nokia N95 GPS-enabled smartphone, transmitting in real time loop detector-like data (called VTL data for Virtual Trip Line), which consists of speed readings at GPS-defined locations upon crossing of the location. These VTLs represent "virtual" loop detectors, which are smartphone-based and may be used by phone manufacturers and access providers to monitor traffic in the near future. The experiment achieved a 2% to 5% penetration rate on the highway throughout the day, thus mimicking smartphone penetration in the driving population in about 18 months. In addition to this online transmitted data to a central server and processed in real time to produce speed estimates, each of the GPS logs collected by the phones at a 1/3 Hz rate was saved in the memory of the phone. While using trajectory data is not part of the Mobile Century technology development plan, the

archival data collected from the experiment can be of great use for traffic analysis. The present study uses the data to showcase the algorithm.

The experiment was conducted from 9:00 am to 7:00 pm. Fig. 11 (a) and (b) show the speed contour map for the entire freeway segment generated by GPS data and loop detector data respectively. We can observe that the GPS data can reproduce almost exactly the same speed contour as the detectors do. Note also that this section of freeway is very unrepresentable of US freeways because of its unusually high concentration of loop detectors (19 over 8.7 miles). It clearly shows that the GPS data is sufficient to capture speed contours or bottlenecks of the freeway. The high accuracy of the data is of interesting, especially in light of the penetration rate for the study (up to 5%). Notice that the bottleneck at 10:45 am was due to a five car pile-up accident on the freeway.

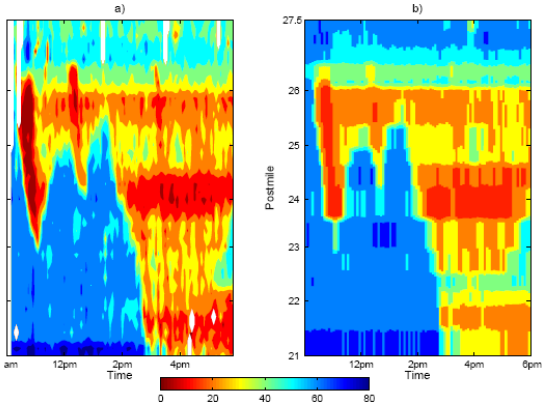

Fig. 11 Speed Contour Maps

We select the shortest loop which has the largest penetration rate. The selected route is depicted in Fig. 12 as labeled "1" and "3" for the origin and destination respectively. The average travel time of the loop is about 20 minutes, implying that the 100 experiment cars represent about 300 vehicles/hours extra freeway traffic volume. Since the freeway has three through lanes in this area, the capacity of the freeway is roughly 6000 vehicles/hour. In other words, the resulting sampling rate of the obtained trajectories is about 5%. In this article, we focus on the northbound of the loop from 10:15 am to 1:45 pm.

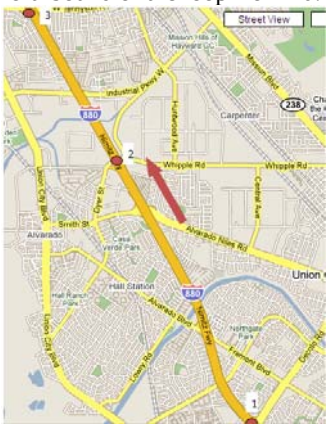

Fig. 12 Experiment Site of Mobile Century

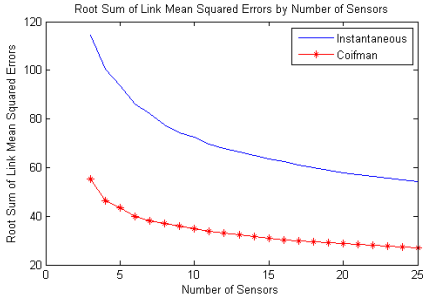

Fig. 13 DP Objective Value vs. Number of Sen-

We run the DP algorithm by varying the number of sensors from 2 to 25, or equivalently for an average spacing from about 3 miles to 0.2 mile. Fig. 13 depicts how the objective value used in the DP model changes as the number of sensors increases. Similarly to the results for simulation data, the objective value decreases as the number of sensors increases, and the Coifman method always has smaller objective values. Fig. 14 depicts the obtained optimal sensor locations when K=6 using the instantaneous travel time method. Similarly to the results in Section 4.1 for simulation data, the DP method puts most sensors (5) to the only bottleneck at the far north of the segment, while only one sensor is deployed to the free flow area. Furthermore, if we look at the evolution of optimal sensor locations as the number of sensors increases from 2 to 12, as shown in Fig. 15, similar observations can also be obtained: most sensors are deployed to the major bottleneck area and only one sensor to the free flow region (2 sensors when the number of sensors is 10 or 11); as the number of sensors increase, previously deployed sensors in the bottleneck area remain almost unchanged and new sensors branch out from existing sensors. Same results can also be obtained for the Coifman method.

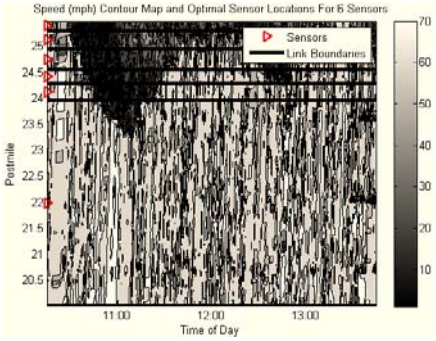

Fig. 14 Optimal Locations for 6 Sensors for the Mobile Century

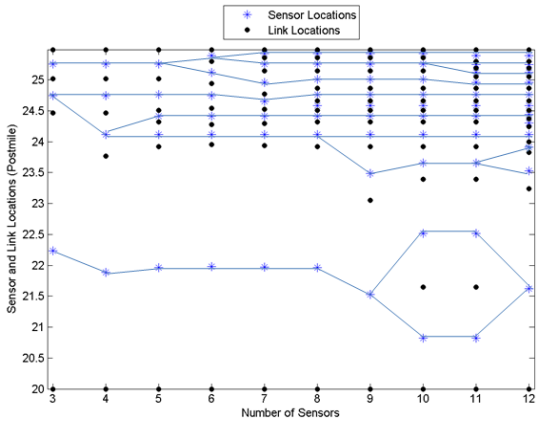

Fig. 15 Evolution of Optimal Sensor Locations for Mobile Century

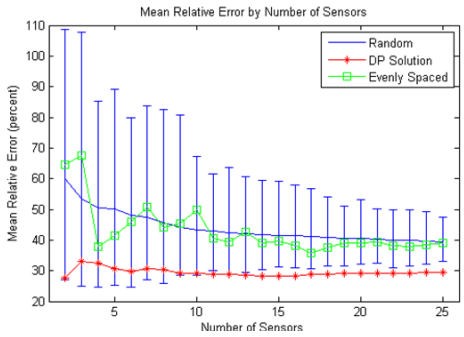

Fig. 16 Comparison of DP and Random Config. on the Entire Route for Mobile Century

As previously done, we compute the objective values as defined in equation (19) for the DP solution and 1,000 randomly generated sensor configurations for the entire route. Fig. 16 depicts that the DP solution is near-optimal compared with the best random configuration for any given number of sensors. Again, the performance of evenly spaced configurations varies significantly and cannot compare with the DP solutions, whose performance is very stable. We then select the best random configuration for the entire route and evaluate its objective value (19) on the sub-route (from "2" to "3" as shown in Fig. 12). The results are shown in Fig. 17 using the dashed line. The figure also depicts the DP solution on this sub-route using the solid line with asterisks. We can see that objective values of the best random configurations vary significantly and are inferior to those of the DP solutions. This again verifies that the DP solutions work well for both the entire route and the sub-routes of the studied freeway segment.

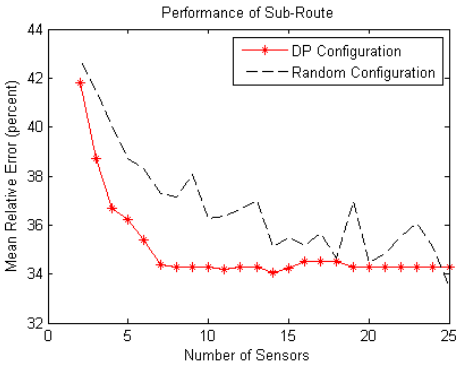

Fig. 17 Comparison of DP and Random Config. on the sub-Route for Mobile Century

6. Conclusion

We studied the optimal sensor placement problem for freeway travel time estimation. The study is based on the assumption that (some) vehicle trajectories are available and sensors only provide aggregated speeds. This is a reasonable assumption for analysis, and becomes a reality with the advent of GPS-enabled cellular phones as a new way to collect traffic information. Based on an objective defined on link MSEs, the problem of determining optimal sensor locations can be modeled as a dynamic programming (DP) formulation and solved using shortest-

path search in an acyclic graph. Therefore, the proposed model can be solved in polynomial time and can be applied to large-scale problems. We also showed how to incorporate existing sensors in the proposed DP framework. We provided two case studies based on trajectory data from micro-simulation and the Mobile Century experiment. The results showed that 1) it is optimal to place many sensors in bottleneck areas and place just a few sensors in free flow areas (one is usually sufficient for one free flow area); 2) the DP solution is more stable and predictable than random configurations, and thus is more desirable in practice; and 3) there seems to be an optimal number of sensors that should be deployed, and beyond which deploying more sensors is not very beneficial.

The DP model and solution algorithm are the first step in determining optimal sensor placement to provide freeway travel times. There are several issues that remain unanswered. Below are some of them:

- How sensitive is the model to different travel time estimation methods? We only tested the instantaneous and Coifman methods. How sensitive the resulting sensor locations to other travel time methods merits further study.
- How sensitive is the model to different sets of vehicle trajectories? In this article, we only utilized trajectories from one simulation run and one experiment. Therefore, day-to-day traffic variations are not considered. How different sets of trajectories will impact the "optimal" sensor locations is an interesting research topic. This issue is under investigation now and results will be reported in subsequent papers.
- How to account for sensor errors? In reality, most sensor data are subject to detection errors. How to consider sensor errors when determining sensor locations is a practical yet challenging problem. In this regard, quantifying the errors of different types of sensors seems necessary. The proposed DP model may be extended for this purpose, which may result in the so-called stochastic dynamic programming problems and merit further investigations.

Acknowledgements The authors would like to thank the four anonymous referees for their insightful comments and helpful suggestions on an earlier version of the paper. This research is partially supported by the California Department of Transportation (Caltrans).

References

Al-Deek, H., & Emam, E.B.(2006) New methodology for estimating reliability in transporta-tion networks with degraded link capacities. Journal of Intelligent Transportation Systems, 10(3), 117-129.

Amin, S., et al (2008). Mobile century-using GPS mobile phones as traffic sensors: a field experiment. In Proceedings of the 15th World congress on ITS, New York, NY, USA.

Bianco, L., Confessore, G., and Reverberi, P. (2001). A network based model for traffic sensor location with implications on O-D matrix estimates. Transportation Science, 35(1), 50-60, 2001.

Ban, X., Li, Y., Skabardonis, A., Margulici, J.D. (2007). Performance evaluation of travel time methods for real time traffic applications. In Proceedings of the 11th World Congress on Transport Research (CD-ROM).

Ban, X., Chu, L., Benouar, H. (2007) Bottleneck identification and calibration for corridor management planning. Transportation Research Record, 1999, 40-53.

Bartin, B., Ozbay, K., Iyigun, C. (2007). A clustering based methodology for determining the optimal roadway configuration of detectors for travel time estimation. Transportation Research Record, 2000, 98-105.

Bellman, R., & Dreyfus, S. (1962). Applied Dynamic Programming. Princeton University Press.

Berkeley Transportation Systems (BTS04). Pems user guide, version 5.2, 2004.

Bertsekas, D.P. (1998). Network Optimization: Continuous and Discrete Models. Athena Scientific.

California Center for Innovative Transportation (CCIT07). 2007. Corridor Management Plan Demonstration. Final report for CCIT Task Order 3. Internet Link: http://www.calccit.org/resources/2007_PDF/CCIT_TO3_FinalReport-Jan5-07.pdf

Chen, A., Yang, H., Lo, H.K., Tang, W. (1999) A capacity related reliability for transportation networks. Journal of Advanced Transportation, 33, 183-200.

Chen, A., Pravinvongvuth, P., Chootinan, P., Lee, M., and Recker, W. (2007). Strategies for selecting additional traffic counts for improving O-D trip table estimation. Transportmetrica, 3(3), 191-211.

Coifman, B (2002). Estimating travel times and vehicles trajectories on freeways using dual loop detectors. Transportation Research Part A, 36(4), 351-364.

Eisenman, S.M., Fei, X., Zhou, X., Mahmassani, H.S. (2006). Number and location of sensors for real-time network traffic estimation and prediction: A sensitivity analysis. Transportation Research Record, 1981, 253-259.

Fujito, I., Margiotta, R., Huang, W., Perez, W.A. (2006) The effect of sensor spacing on performance measure calculations. In Proceedings of the 85th Annual Meeting of Transportation Research Board (CD-ROM).

Jia, Z., Chen, C., Coifman, B., Varaiya, P. (2001). Pems algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In Proceeding of IEEE ITS Annual Meeting, 536-541.

Khattak, A., Kanafani, A., Colletter, E.L. (1994). Stated and reported route diversion behavior: implications of benefits of advanced traveler information system. Transportation Research Record, 1464, 28-35.

Kwon, J., McCullough, B., Petty, K., Varaiya, P. (2006). Evaluation of PeMS to improve the congestion monitoring program. Technical report, Final Report for PATH TO 5319.

Li, Z.C., Lam, W.H.K., Wong, S.C., Huang, H.J., Zhu, D.L. (2007) Reliability evaluation for stochastic and time-dependent networks with multiple parking facilities. Networks and Spatial Economics, in press.

Lindveld, C.D.R., Thijs, R., Bovy, P.H.L., Zijpp, N.J.V (2000). Evaluation of online travel time estimators and predictors. Transportation Research Record, 1719, 45-53.

Mirchandani, P., Gentili, M., and Ye, Y. (2007) Sensor locations on a network to monitor travel-time performance. In Proceedings of the 3rd International Symposium on Transportation Network Reliability (INSTR07).

Rice, J., & Zwet, E.V. (2001). A simple and effective method for predicting travel times on freeways. In Proceedings of IEEE Intelligent Transportation Systems, 227-232.

Sherali, H.D., Desai, J., Rakha, H. (2006). A discrete optimization approach for locating automatic vehicle identification readers for the provision of roadway travel times. Transportation Research B, 40, 857-871.

Sanwal, K.K. & Walrand, J. (1995) Vehicle as probes. Technical Report UCB-ITS-PWP-95-11, California Path.

Thomas, G.(1999) The relationship between detector location and travel characteristics on arterial streets. Institute of Transportation Engineers Journal, 69(10), 36-42.

Work, D., Tossavainen, O.P., Blandin, S., Bayen, A., Iwuchukwu, T., Tracton, K. (2008) An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In Proceedings of the 47th IEEE Conference on Decision and Control.

Yang, H., & J. Zhou (1998). Optimal traffic ccounting locations for origin-destination matrix estimation. Transportation Research Part B, 32(1), 109-126.