DEVELOPMENT OF A ROUTE GUIDANCE GENERATION SYSTEM FOR REAL-TIME APPLICATION

Moshe Ben-Akiva [*] **Michel Bierlaire** [**] **Jon Bottom** [***]
Haris Koutsopoulos [****] **Rabi Mishalani** [†]

[*] *Professor, Massachusetts Institute of Technology*
[**] *Research Associate, MIT*
[***] *Ph.D. candidate, MIT*
[****] *Associate Professor, Carnegie-Mellon University*
[†] *Research Associate, MIT*

Abstract. This paper describes the route guidance generation component of DynaMIT, a dynamic traffic assignment (DTA) system intended for deployment in a traffic center and capable of generating real-time prediction-based guidance information. After providing a general overview of the system, the paper discusses the principal theoretical and algorithmic issues which influenced the development of its guidance generation component.

Keywords. Kalman filter, networks, routing algorithms, simulators

1. INTRODUCTION

DynaMIT (Dynamic Network Assignment for the Management of Information to Travelers) is a real time dynamic traffic assignment system that provides traffic predictions and travel guidance. Travel guidance refers to information provided to tripmakers in an attempt to facilitate their decisions relative to departure time, travel mode and route. Clearly, departure time and (to some extent) travel mode information is only effective prior to trip departure, whereas route guidance recommendations may be useful both before and during a trip.

In order to guarantee the *credibility* of the information system, the guidance provided by DynaMIT is consistent, meaning that it corresponds to traffic conditions that will be experienced by drivers. DynaMIT provides user-optimal guidance, having the property that users cannot find a path that they would prefer to the one they chose based on the provided guidance.

DynaMIT is designed to operate in real time, using traffic volume and control system state data to estimate and predict time-dependent origin-destination (OD) flows and network conditions, and generating route guidance that is consistent with the predicted traffic conditions. It is designed to make departure time, mode and route recommendations for a variety of information system technologies and information dissemination strategies.

The system features:

- a mesoscopic-level traffic simulator which moves individual vehicles or vehicle packets according to macroscopic traffic relations;
- a disaggregate-level demand simulator which explicitly models the pre-trip departure time, mode and path choice behavior and the *en route* path switching behavior of a number of user classes;
- a dynamic OD matrix estimation capability which estimates and predicts time-dependent OD flows by applying Kalman filter methods to real-time data

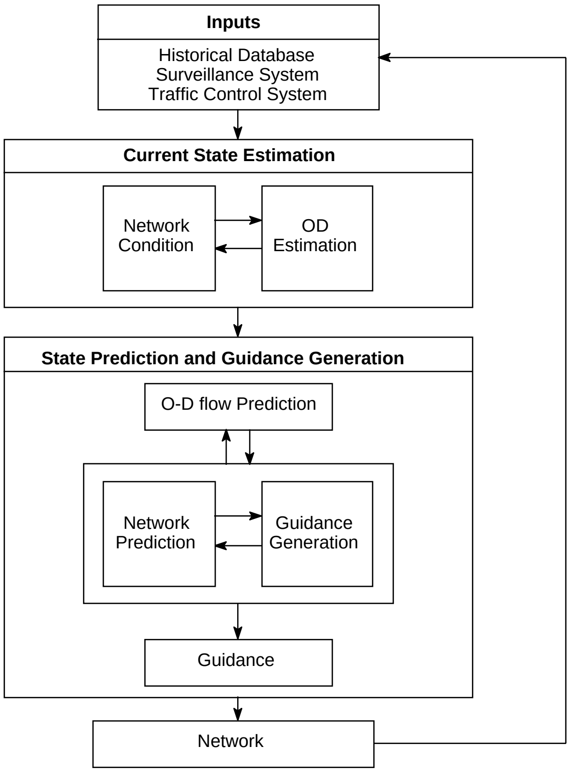

Fig. 1. Overall DynaMIT system structure

from traffic sensors and identifying deviations from historical traffic patterns;

- the ability to compute anticipatory (predictive) route guidance, including both descriptive (network conditions) and prescriptive (route recommendations) information.

Although the system can be used in a number of modes (including real-time simulation and forecasting of traffic networks lacking any form of ATIS), its capabilities related to the modeling and generation of guidance information are of particular interest here.

2. OVERVIEW OF THE SYSTEM

DynaMIT consists of two main components: network state estimation and prediction-based guidance generation. These two components utilize various models and algorithms to attain their objectives. Some of these models and algorithms are shared by the two components. Figure 1 indicates the overall system structure. The state estimation components determine the current state of the network given historical and surveillance data. The estimation is based on explicit and consistent simulation of the interaction between network supply and traveler demand. State prediction and guidance generation components take the state estimate as input and generate anticipatory guidance. The system enforces consistency between the travel times on which the guidance is based and the travel times which result from travelers' reactions to the guidance. The following two sections present this framework in more detail.

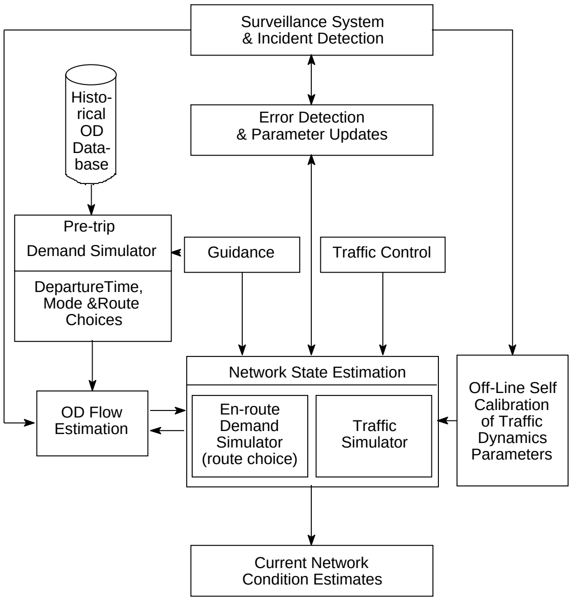

Fig. 2. Network state estimation function

2.1 *Network state estimation*

The principal function of this module is to provide estimates of the current state of the network in terms of OD flows and of link volumes, densities and queue characteristics. To this end, it processes real-time data obtained from traffic sensors, integrating this data with other information accessible in a database of historical data about time-dependent OD demands, and traffic patterns and conditions.

This module uses the following models:

- real-time OD matrix estimation;
- departure time and mode choice models; and
- traffic simulation.

The above models interact with each other in order to provide OD matrix and network state estimates that are consistent and that utilize the most recent information available from the surveillance system. They are organized around three functions (see Figure 2):

- pre-trip demand simulator;
- OD flow estimation; and
- network state estimation.

Knowledge of time-dependent OD flows is a key requirement for the DTA. The estimation of the OD flows is sensitive to the guidance generated and information provided to users. This is accomplished through an explicit simulation of departure time and mode choice decisions, which ultimately produce the OD flows used by the OD estimation model (Antoniou *et al.*, 1997). The pre-trip demand simulator updates historical OD matrices by modeling the reaction of individual tripmakers to guidance information. The consequent changes are then aggregated to obtain updated historical OD matrices.

Of course, these estimated OD flows may not always correspond accurately to the actual travel demand in the network. Consequently, one of the requirements for dynamic traffic modeling is the capability to estimate (and predict) OD flows in real time. Using a Kalman filter approach developed by Ashok and Ben-Akiva (1993), the OD model integrates historical OD flows, real-time measurements of actual link flows on the network, and estimates of assignment fractions (the mapping from OD flows to link flows based on route choice fractions and travel times provided by the traffic simulator) to estimate the OD flows for the current interval.

The network state estimator uses a mesoscopic traffic simulation model which simulates traffic conditions in the network from the beginning of the previous time period to the current simulation period. Inputs include OD flows estimated by the Kalman filter, updated network capacities and traffic dynamics parameters, and the traffic control strategies implemented and guidance information disseminated during the previous time period. The *en route* demand simulator captures traveler responses to ATIS in the form of driver path switching choices. The simulator has a flexible design that allows for full investigation of trade-offs between accuracy and run-time efficiency and facilitates the selection of the most suitable combination for each case; design details are presented in Ben-Akiva *et al.* (1992).

It may be necessary for the OD estimation model and network state estimator to iterate several times to obtain a consistent state estimate. This is so since one of the inputs to the OD estimation model is the assignment matrix obtained from the traffic simulator.

Operation of the state estimation module is supported by an off-line parameter calibration model. This model uses information obtained from the sensors and updates the parameters of the traffic dynamics models used in the traffic simulation. In addition, an error detection module filters the surveillance data and updates capacity levels based on any detected errors prior to passing this data to the network state estimation module.

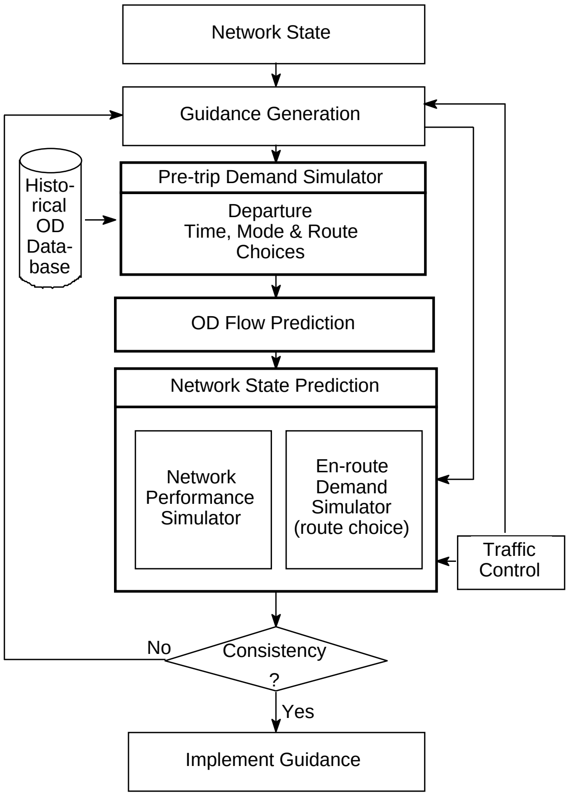

Fig. 3. Prediction-based guidance generation

2.2 *Prediction-based guidance*

The prediction-based guidance module consists of several interacting components (see Figure 3):

- pre-trip demand simulation;
- OD flow prediction;
- network state prediction; and
- guidance generation.

The OD prediction model uses as inputs the aggregated historical demand adjusted by the pre-trip demand simulator to account for departure time and mode choices in response to guidance, and provides the required estimates of future OD flows.

The network state prediction function provides traffic predictions for a given control and guidance strategy, and for a predicted set of OD flows using as a starting point the current network condition as estimated by the state estimation module. It uses a traffic simulation model and driver *en route* behavior models to predict the performance of the network in the future.

The guidance generation function uses predicted traffic conditions to generate guidance taking account of the ATIS technology in place. For the current version of DynaMIT, traffic control strategies are generated outside DynaMIT and provided as input to it.

An iterative process is used to generate guidance. An iteration consists of a trial strategy, state prediction (network and OD) under the trial strategy, and evaluation of the predicted state. Since, in general, the updated historical OD flows depend on future guidance and information, the update of the historical OD flows (using the departure time and mode choice models) and the OD prediction models are included in the iteration. This general case represents the situation where pretrip guidance is available to the drivers. In the special case where only *en route* guidance is available, the pretrip demand simulator is bypassed during the iteration. The initial strategy, used to start the iterations, is obtained from the prediction and guidance generation of the previous period. Results of an iteration's simulation are evaluated for consistency. If a trial guidance strategy is rejected, a new one is generated and the iteration is continued until a consistent strategy is found, or until a prespecified number of iterations have been performed. In the latter case, the iteration with the most consistent results is chosen as the basis for guidance generation.

The following section discusses DynaMIT's guidance generation capabilities in greater detail.

3. GUIDANCE GENERATION METHODOLOGY

3.1 *Assumptions*

DynaMIT can generate both anticipatory descriptive and prescriptive guidance. Anticipatory descriptive guidance informs travelers about the traffic conditions they are likely to encounter on different feasible paths from their current position to their destination. Anticipatory prescriptive guidance recommends a path to travelers based on expected traffic conditions along alternative feasible paths.

The methodology used for guidance generation assumes an ATIS with the following generic characteristics:

- State estimation and subsequently guidance generation take place every ΔT time units. (As a rough indication, a typical value of ΔT might be around 15 minutes.) The generated guidance will be applicable to the interval $[T, T + \Delta T]$, where T is the time at which the state estimation takes place (a rolling horizon implementation).
- Guidance information is provided at the same time resolution as major simulation updates, at subintervals u within the interval $[T, T + \Delta T]$. (A typical value of u might be around 1 minute.)
- The guidance computations for the interval $[T, T + \Delta T]$ will take account of network conditions at least until the time when all vehicles generated during $[T, T + \Delta T]$ will have exited the network. Call this analysis period the guidance horizon ΔH. The value of ΔH will be approximately equal to the maximum trip duration of any trip generated during $[T, T + \Delta T]$ (say 2 or 3 hours for a major metropolitan area.)
- Although a number of factors are known to enter drivers' path choice processes, the initial implementation of guidance generation in DynaMIT considers only travel time as a determining impedance variable. (The guidance information representation and the guidance generation algorithm described below are sufficiently general to handle additional impedance variables.)
- It is also assumed that the computation and communications technologies supporting the traffic information system are sufficiently powerful to allow the computation and dissemination of guidance information relatively frequently. More specifically, it is assumed that (i) guidance computations which generate distinct guidance for each simulation update interval will not be a system bottleneck and (ii) the guidance generated for each simulation update interval can be disseminated sufficiently quickly so as to be received and (possibly) acted upon during the interval.

The last assumption is necessary to avoid perverse situations in which the intrinsic delays of the guidance generation and dissemination systems result in a deterioration rather than an improvement of traffic conditions. This might happen when the sluggishness of the system prevents a distributive (multi-path) guidance strategy from being implemented, even when multiple paths are required to achieve consistent guidance.

The assumptions are not unreasonable. On the one hand, DynaMIT is designed to be executed in a multi-processor system; the incorporation of additional or more powerful processors should ensure that the guidance computation does not become a system bottleneck. On the other, a number of studies, such as (Cheslow and Hatcher, 1993; von Tomkewitsch, 1991), have examined the communications bandwidth requirement of alternative ATIS designs and concluded that guidance dissemination intervals on the order of five minutes or less are feasible with currently-available communications technologies.

3.2 *Objective*

The objective of the guidance generation function implemented in DynaMIT is to provide guidance which is consistent and unbiased. Consistent and unbiased guidance implies that the travel times which are used for guidance generation are unbiased estimates of the conditions which users actually experience on the paths they are predicted to take (based on the travel behavior models.) This objective eventually leads to user-optimality.

Hence, user-optimal guidance facilitates travelers' decision-making processes and reduces disequilibrium behavior resulting from imperfect information. In the case of user-optimal descriptive guidance, this is done by improving the quality of the information about traffic conditions which is available to travelers. In the case of user-optimal prescriptive guidance, it is done by recommending the path which would appear best according to travelers' presumed path choice logic. Drivers receiving prescriptive guidance would in principle not be able to find a different path which they would prefer to the recommended one. In both cases, guided drivers will be induced to follow paths having the minimum individual impedance, and they will not be able to find better paths through unilateral action; this is the defining criterion for user optimality.

3.3 *Solution approach*

Generation of consistent unbiased guidance is not a trivial task. The complication arises from the fact that drivers who receive guidance information may change their paths as a result of that information. This may lead in turn to a change in the time-dependent link travel times on which the guidance information was based, and so the original guidance may be invalidated. Guidance generation is therefore a fixed point problem which may be stated as follows: find x such that

$$x = f(x) \tag{1}$$

where x is the vector of the values of a variable that is used as the basis for guidance generation, and $f(x)$ is the vector of values which results from travelers' reactions to the guidance they receive.

Solving the fixed point problem is difficult, particularly in light of the fact that $f(x)$ does not have a closed form, but rather is simulated by the interaction between the supply and demand components of the system. Furthermore, given the structure of the problem, it is possible that an exact solution may not even exist. Hence, a relaxed version of the problem is used:

$$\min I = \|x - f(x)\| \tag{2}$$

I is a measure of guidance inconsistency; it is a function (for example, the quadratic norm) of the difference between assumed and actual guidance variable values.

In general, inconsistency can be quantified in various ways, using for example time-dependent link travel times, time-dependent path travel times, or actual behavior in the form of traveler decisions. The time-dependent path travel time is used in DynaMIT because it is the most relevant in terms of ATIS applications and because it is straightforward to compute as well.

An iterative algorithm, referred to as the *time smoothing* algorithm, has been developed for the solution of the above problem. The algorithm is inspired to some extent by the method of successive averages (MSA), which is commonly used in stochastic user equilibrium assignment algorithms.

The following notation is used in the description of the algorithm:

- k: one of K vehicle guidance classes, representing different levels and types of information available to a vehicle. Class 0 represents background (fixed route) vehicles and class 1 represents unguided vehicles which can only access public information such as VMS or radio broadcasts;
- $X_{OD}^{u,k}$: estimated number of trips from O to D, departing from O in simulation interval u, by vehicles of guidance class k;
- $X_{OD,p}^{u,k}$: estimated number of trips from O to D on path p, departing from O in simulation interval u, by vehicles of guidance class k;
- ${}^{n}T_{L}^{u}$: travel time on link L during simulation interval u, as simulated during iteration n of the guidance generation algorithm;
- ${}^{n}S_{L}^{u}$: "auxiliary" travel time on link L during simulation interval u, as simulated during iteration n of the guidance generation algorithm;
- α_n: a factor which determines how heavily auxiliary travel times are weighted in the smoothing of travel times during iteration n.

The time smoothing algorithm can then be described as follows:

Given:

- The initial network state at time T, as obtained from the state estimation module;
- Initial link travel times: ${}^{0}T_{L}^{u}, u \in [T, T + \Delta H]$
- Forecast OD trips by vehicle guidance class over the guidance horizon: $X_{OD}^{u,k}, k = 0, \ldots, K, u \in [T, T + \Delta H]$;
- Estimated OD trips by path for background ($k = 0$) and unguided ($k = 1$) vehicles over the guidance horizon: $X_{OD,p}^{u,0}, X_{OD,p}^{u,1}, u \in [T, T + \Delta H]$
- A prespecified maximum allowable number of iterations N

Algorithm (guidance generation by time smoothing):

```
$n := 0$
loop
  1: $n := n + 1$
  2: Get latest link time table $^{n-1}T^u_L$;
  3: Simulate the network for $[T, T+\Delta H]$, storing
     link times in $^{n}S^u_L$;
  4: $^{n}T^u_L := {}^{n-1}T^u_L + \alpha_n({}^{n}S^u_L - {}^{n-1}T^u_L)$.
  5: Check consistency
until $\{$consistency achieved or $n = N\}$
```

The algorithm proceeds by iteratively revising forecasts of travel times on all links and for all simulation update intervals during the guidance horizon (this data structure will be called the link time table). In a particular iteration (Step 3) it uses the simulator to (i) provide guidance based on the latest link time table; (ii) run the appropriate driver behavior models to predict drivers' response to guidance information and to traffic conditions; and (iii) move vehicles accordingly. The link travel times which result from the iteration's simulation run are stored as an "auxiliary" link time table. Travel times from the auxiliary table are combined (Step 4) with those from the current link time table to derive the link time table that will be used for the next iteration. In the current version of DynaMIT, consistency evaluation is based on the path travel time resulting from the computed link times. The process stops either after consistency is achieved (to within a given tolerance) or after the prespecified maximum allowable number of iterations, N, has been performed. Thus, the presented algorithm is guaranteed to terminate but not necessarily to converge within N iterations. If the algorithm terminates without convergence, the most consistent trial solution is used as the basis for guidance generation.

3.4 *Example Application*

The time-smoothing algorithm was tested on Boston's Central Artery and Third Harbor Tunnel network currently under construction. The network abstracted for our purposes consists of two origins, two destinations, four route choice nodes and eight links (approximately 12 km). The two main paths from the origins to the destinations traverse one of two tunnels.

Under an incident scenario where the capacity of one tunnel is reduced by half, two guidance scenarios were tested: (i) no guidance, and (ii) anticipatory consistent guidance. Reasonable consistency was attained within 20 iterations by selecting the trial solution which had the highest consistency. Moreover, the system-wide improvement in travel times when this solution was implemented was significant compared to the no guidance scenario.

Among the 20 generated trial solutions, other less consistent solutions that resulted in higher system-wide travel time improvements (with respect to the no guidance scenario) were identified. Some of these solutions had significantly lower consistency than the selected one. Therefore, travel time improvement might be useful as a basis for guidance selection only in choosing among multiple consistent solutions.

It is important to emphasize that the results reported here are preliminary in nature. More extensive testing is necessary before reaching any final conclusions.

4. CONCLUSION

In this paper DynaMIT is presented as a system of models for guidance generation based on predictive traffic conditions. DynaMIT interacts with the traffic surveillance system and generates user-optimal guidance which takes into account estimated network conditions (including OD flows) and traveler response to information (both pre-trip and *en route*). Preliminary results based on a simple application network provide promising indications and direction for more extensive testing.

5. REFERENCES

Antoniou, Constantinos, Moshe Ben-Akiva, Michel Bierlaire and Rabi Mishalani (1997). Demand simulation for dynamic traffic assignment. Presented at the 8th IFAC Symposium on Transportation Systems, Chania, Greece.

Ashok, Kalidas and Moshe Ben-Akiva (1993). Dynamic O-D matrix estimation and prediction for real-time traffic management systems. In: *Transportation and Traffic Theory* (Carlos F. Daganzo, Ed.). Elsevier Science Publishing Company, Inc.

Ben-Akiva, Moshe, Haris N. Koutsopoulos and Anil Mukundan (1992). A dynamic traffic model system for ATIS/ATMS operations. *IVHS Journal* **2**(1).

Cheslow, M.D. and S.G Hatcher (1993). Estimation of communication load requirements for five ATIS/ATMS architectures. In: *Proceedings of the IVHS America Annual Meeting.* pp. 473–479.

von Tomkewitsch, R. (1991). Dynamic route guidance and interactive transport management with ALI-SCOUT. In: *IEEE Transactions on Vehicular Technology.* pp. 45–54.