



*The University of Michigan*

**IVHS**

*Intelligent Vehicle-Highway Systems*

# A Dynamical System Model for Traffic Assignment in Networks

Stephane Lafortune and Raja Sengupta

Department of Electrical Engineering and Computer Science

and

David E. Kaufman and Robert L. Smith

Department of Industrial and Operations Engineering  
University of Michigan  
Ann Arbor, Michigan 48109

May 1991

IVHS Technical Report 91 - 01  
For Release on May 1, 1992

# A Dynamical System Model for Traffic Assignment in Networks

Stephane Lafortune and Raja Sengupta

Department of Electrical Engineering and Computer Science

and

David E. Kaufman and Robert L. Smith

Department of Industrial and Operations Engineering  
University of Michigan  
Ann Arbor, Michigan 48109

May 1991

IVHS Technical Report 91 - 01  
For Release on May 1, 1992

# A Dynamical System Model for Traffic Assignment in Networks\*

*Stéphane Lafortune and Raja Sengupta*

Department of Electrical Engineering and Computer Science

*and*

*David E. Kaufman and Robert L. Smith*

Department of Industrial and Operations Engineering

University of Michigan

Ann Arbor, MI 48109

## Abstract

We propose a formulation for the problem of dynamic traffic assignment in networks with multiple origins and destinations. This problem is motivated by route guidance issues that arise in an Intelligent Vehicle-Highway Systems (IVHS) environment. We assume that the network is subject to known time-varying demands for travel between its origins and destinations during a given time horizon. The objective is to assign the vehicles to links over time so as to minimize the total travel time experienced by all the vehicles using the network. We model the traffic network over the time horizon as a discrete-time dynamical system. The system state at each time epoch is defined in a way that avoids complete microscopic detail by grouping vehicles into platoons irrespective of origin node and time of entry to network. Moreover, the formulation contains no path enumeration. The state transition function can model link travel times by either impedance functions or link outflow rate constraints, or by a combination of both. Overall, the optimization problem becomes one of optimal control for a nonlinear discrete-time dynamical system, and it is amenable to an algorithmic solution based on dynamic programming.

## Keywords

intelligent vehicle-highway systems, dynamic traffic assignment,  
impedance functions, link outflow rate functions,  
state space model, dynamic programming

---

\*Research supported in part by the IVHS Program at the University of Michigan and by the National Science Foundation under grant ECS-9057967.

May 14, 1991

## 1. Introduction

We consider the problem of dynamic traffic assignment in networks with multiple trip origins and destinations. Our approach is as follows. The traffic network, which may include both freeway corridors and surface streets, is modeled as a directed graph. The sets of origins and destinations are subsets of the set of nodes of the graph. The arcs of the graph are links in the network. Some information is available about these links, in the form of impedance functions, which express link travel times in terms of the number of vehicles on the links, or link outflow rate functions, which constrain the departure or exit rate of vehicles from a link in terms of the number of vehicles on this link. We assume that the network is subject to known time-varying demands from vehicles for travel between its origins and destinations during a given (finite) time horizon (e.g., a period of a few hours). The objective is to assign the vehicles to links over time in order to minimize the total travel time experienced by all the vehicles using the network. Thus the resulting assignment will be system-optimal. We are dealing with a dynamic as opposed to static problem because the demand is dynamic and the routes assigned to vehicles from their origins to their destinations depend on the entire set of demands over the whole time horizon considered.

Our motivation for studying this problem comes from route guidance issues that arise in an Intelligent Vehicle-Highway Systems (IVHS) environment [8]. In IVHS, it is desired to perform “anticipatory” route guidance, i.e., to route the vehicles on the network on the basis of the future travel times that they will experience on the links that they will be traveling. However, these future travel times depend on the routing decisions made for all the vehicles traveling on the network, and thus they have to be forecasted using a combination of historical and real-time information (see, e.g., [5, 4, 10]). Our objective is not to address this real-time forecasting/assignment problem where the demand is not known a priori and the decisions may be affected by the occurrence of incidents and other unpredictable events. By making the simplifying assumption that the demand, although dynamic, is known beforehand, and by considering a fixed time horizon (with no incidents occurring), we obtain an optimization problem that may be solved completely, at least in principle. It is our thesis that the solution of the dynamic traffic assignment problem considered in this paper, as well as the properties of this solution, will provide considerable insight into the problem of real-time “anticipatory” route guidance in IVHS. This thesis is the primary motivation of the work that follows.

We model the traffic network over the time horizon as a discrete-time dynamical system. At each time epoch, the system state consists of the sizes of all platoons, each of which represents all vehicles on a certain link with the same destination and the same earliest possible time of departing the link. This state definition avoids complete microscopic detail

by grouping vehicles into platoons irrespective of origin node and time of entry to network, yet it conforms to the requirement that the state should summarize all relevant past behavior so as to contain sufficient information for the determination of the future behavior of the system. Also, since this formulation is based on links, it contains no path enumeration. We provide a general form for the state transition function giving the possible states at time  $n + 1$  as a function of the state at time  $n$  and of the feasible assignment or routing decisions for platoons that exit a link or join the network in the time interval  $(n, n + 1]$ . Specific forms of the state transition function can model link travel times by either impedance functions or link outflow rate constraints. Further, the two can be combined in a way which represents interaction between links due to recurring congestion and capacity reductions, in addition to single link impedances.

Overall, the optimization problem becomes one of optimal control for a nonlinear integer-valued discrete-time dynamical system. The number of control variables (i.e., the number of routing decisions that have to be made) is equal to the product of: (1) the number of links, (2) the number of destinations, and (3) the time horizon considered. As formulated, the optimization problem is amenable to an algorithmic solution based on dynamic programming. Hence, Dijkstra's algorithm can be employed to determine the system optimal routing decisions in the context of a forward dynamic programming search over the state space. Other search techniques (optimal or heuristic) could also be employed.

Our presentation is organized as follows. We compare our approach with related work on dynamic traffic assignment in Section 2. The dynamical system model is presented in Section 3, and the resulting optimal control problem is formulated in Section 4. This section also presents forward and backward dynamic programming recursions for solving the problem. The forward dynamic programming recursion is applied in Section 5 to a simple triangle network. Finally, Section 6 discusses extensions of the model for the inclusion of background traffic and blocking controls, and Section 7 concludes the paper.

## 2. Comparison with Other Work

Janson [3] has recently addressed the dynamic user-equilibrium traffic assignment problem for networks with multiple origins and destinations and known time-varying travel demands. He has developed a mathematical program for dynamic user-equilibrium assignment and proposed an algorithm for the solution of this problem. The algorithm is based on a two-step decomposition of the problem that is then solved iteratively. His simulations have shown that the iterative procedure exhibits good near convergence properties. Our approach differs from Janson's work in at least two respects. First, we are considering a

system-optimal objective rather than a user-equilibrium one. Second, our discretization of time is fundamentally different. In [3], the discrete time interval  $\Delta t$  must be chosen large enough so that vehicles completely traverse any link in one time interval. Otherwise, the links need to be broken into smaller ones resulting in an increase of the dimensionality of the mathematical program. In our formulation,  $\Delta t$  must be chosen small enough so that vehicles will not traverse more than one link in one time interval. The number of decision variables in our optimization problem increases linearly in the number of time intervals.

Merchant and Nemhauser [6] formulated a mathematical program for system-optimal dynamic traffic assignment in a network with multiple origins and a single destination. They assume that all links are uncapacitated and that in each time interval the number of vehicles departing a link is a function (nonlinear in general) only of the volume on that link. Therefore, congestion caused by blocking of one link by another congested link is not modelled. Carey [1] reformulated this model as a convex program by assuming the link departure function to be a maximum outflow instead of an actual outflow, hence constraining by nonlinear inequality instead of equality. Thus vehicles may be held back to benefit the system-optimality criterion, but not due to congestion on other links since the model remains uncapacitated. Departure functions can be included in the dynamical system model presented in this paper, and combination with impedance functions for link travel times will prevent unreasonably short link travel times that might otherwise occur with short time intervals  $\Delta t$ .

Papageorgiou *et al.* [7] have studied the general requirements of dynamic models based on standard state space methods. While their more general remarks are applicable to our work, our commitment to the use of specific traffic engineering tools such as impedance functions or link outflow rates, makes our model more specialized. In their more specific comments Papageorgiou *et al.* seem to represent two kinds of information in the state. They are the number of vehicles per unit length on a link and the composition rates that represent the proportion of vehicles on the link flowing to a particular destination. In contrast to this our state does not explicitly represent the spatial position of vehicles beyond recognizing their current link location. Instead the exit time of a vehicle from its current link is recorded. Thus we substitute some measure of temporal position for spatial position. The vehicles on a link are grouped by destination. Thus something similar to the composition rates also exists in our model.

The work of Friesz *et al.* [2] on dynamic traffic assignment is also in the system theoretic paradigm. However, they have modeled traffic flow as a real valued continuous time process. The model that we present is discrete time and integer valued. It is developed in the hope



of providing a compact and elegant mathematical representation to which methods in areas such as combinatorial optimization or mixed integer programming may be applied. We do not aim at analysis on the basis of the Pontryagin principle.

### 3. System Description

We present in this section a dynamical system model of a traffic system having multiple origins and destinations. Each link in the network may be described by a first order difference equation and the network as a whole is represented by another first order difference equation that is an aggregation of the link equations. Thus our dynamical equation possesses an attractive and simple modular structure.

The geographical network is viewed as a finite-vertex directed graph in which every edge is associated with one and only one ordered pair of vertices. Thus any origin or destination must necessarily be a vertex and any highway or arterial connecting two points with no point of interest in between is an edge on the directed graph. The structure is formalized as follows.

- (i)  $|X|$  = Number of elements of a finite set  $X$ .
- (ii)  $V$  = Set of vertices of a network assumed to be arranged into a sequence  $\langle v_i \rangle_{i=1}^{i=|V|}$ .
- (iii)  $E$  = Set of directed edges or links assumed to be arranged into a sequence  $\langle \ell_i \rangle_{i=1}^{i=|E|}$ .
- (iv)  $D$  = Set of destination nodes or vertices, assumed to be arranged into a sequence  $\langle d_i \rangle_{i=1}^{i=|D|}$ . This is a subset of  $V$ .

The set of origin vertices does not have to be explicitly referred to in the dynamical equations that follow. Therefore, without loss of generality, we assume that this set is equal to  $V$ .

Since the formulation is discrete time, we define

- (i)  $n$  = Discrete time index. Thus  $n \in \mathbb{N}$ .
- (ii)  $N_j$  = Maximum possible number of sampling instants spent by a vehicle on the link  $\ell_j \in E$ .

Thus it is assumed that there exists an upper bound on the time that a vehicle may spend on a given link. We consider such an assumption tenable if a link is never loaded in excess of a defined capacity. This restriction will be introduced when the region of admissible behaviour

of the model is defined. Moreover, it may be noted that  $N_j$  is independent of the time at which a vehicle joins the link  $\ell_j$ , i.e., of prevailing traffic conditions at the time of entry.  $N_j$  is a function of highway or arterial capacity that is also time invariant.

The state of the network at any given sampling time must reflect the location of each vehicle on the network at that time. For this purpose, we define the following state variable that groups vehicles on a link according to their destinations and exit times from the link:

$$x_k^{ij}(n) = \text{Number of vehicles on link } \ell_j \text{ and traveling to destination } d_i, \text{ present on the link at time } n \text{ and leaving the link at time } (n+k)^-.$$

Thus the above is defined for all  $\ell_j \in E$ ,  $d_i \in D$  and  $1 \leq k \leq N_j$ . Our convention is that traffic leaves a link at time  $(n+k)^-$  to enter the next link at time  $(n+k)$ .

The state of the network is affected by vehicles entering or leaving links. The exit of vehicles is modeled as an internal dynamic of the link and is reflected in the time evolution of the homogeneous (zero-input) dynamical equation. However, the entry of vehicles into a link is modeled as an input. Accordingly we define

$$u^{ij}(n) = \text{Number of vehicles traveling to destination } d_i \text{ and entering link } \ell_j \text{ at time } (n+1).$$

Our convention is that the input at time  $n$  will affect the state at time  $n+1$ . A system output is also defined. This is required for the conservation of flow.

$$y^{ij}(n) = \text{number of vehicles on link } \ell_j \text{ which are traveling to destination } d_i \text{ and are on the link at time } n \text{ and leaving at time } (n+1)^-.$$

The last prerequisite for formulating a state equation is a functional relationship between input and state. This is where we resort to the impedance function. For all  $\ell_j \in E$  we define

$$\begin{aligned} f_j(v) &= \text{Impedance of link } \ell_j \text{ when loaded with } v \text{ vehicles} \\ &= \text{travel time of a vehicle joining } \ell_j \text{ when } v \text{ vehicles are traveling on this link.} \end{aligned}$$

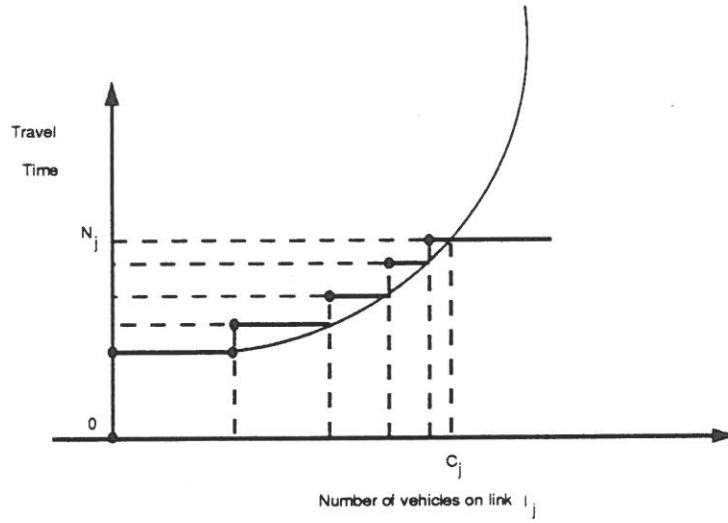
The impedance function is assumed to be an integer valued staircase function that is right continuous. A typical function is sketched in Figure 1. Formally the impedance function may be defined as

$$f_j(v) = \sum_{k=N_0}^{N_j} k \chi_{[a_k, b_k)}(v)$$

where  $a_{N_0} = 0$ ,  $b_{N_j} = \infty$  and  $b_k = a_{k+1}$  for  $N_0 \leq k \leq N_j - 1$ , and  $\chi$  is the usual characteristic function.

We adopt for all  $x = (x_1, \dots, x_n)^T \in R^n$  the notation  $\|x\| = |x_1| + |x_2| + \dots + |x_n|$ .





$c_j = \text{capacity}$   
 $N_j = \text{Max travel time}$

Figure 1: Impedance function

### 3.1. The link dynamical equation

We define a dynamical equation for the vehicles on link  $\ell_j$  and traveling to various destinations  $d_i$ . Let the link have no input (i.e., no new vehicles entering the link). We assume that the exit time of a vehicle from link  $\ell_j$  is fixed at the time of entry by the impedance function as follows:

$$x_k^{ij}(n+1) = x_{k+1}^{ij}(n) \quad 1 \leq k \leq N_j - 1$$

and

$$x_{N_j}^{ij}(n+1) = 0 .$$

This will be generalized in Section 6.2 where blocking of vehicles at the end of a link will be incorporated into the model.

Thus we obtain a linear homogenous first order difference equation

$$x^{ij}(n+1) = A^{ij}x^{ij}(n)$$

where

$$x^{ij}(n) = [x_1^{ij}(n) \cdots x_{N_j}^{ij}(n)]^T$$

and

$$A^{ij} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}_{N_j \times N_j}$$

To incorporate the input  $u^{ij}(n)$ , a simple extension of the state and input is necessary.

Let

$$x^j(n) = [x^{1j}(n)^T \cdots x^{D|j}(n)^T]_{N_j|D| \times 1}^T$$

and

$$u^j(n) = [u^{1j}(n) \cdots u^{D|j}(n)]_{|D| \times 1}^T.$$

Observe that

$$x^j(n+1) = A^j x^j(n)$$

where

$$A^j = \text{diag}\{A^{1j} \cdots A^{D|j}\}_{(N_j|D|) \times (N_j|D|)}$$

**Remark:** For given matrices  $M^j$   $j = 1, \dots, J$ , we shall use the notation  $\text{diag}\{M^1 \cdots M^J\}$  to represent the block-diagonal matrix

$$\begin{bmatrix} M^1 & 0 & \cdots & 0 \\ 0 & M^2 & & \vdots \\ \vdots & & \cdots & 0 \\ 0 & \cdots & 0 & M^J \end{bmatrix}.$$

For each link and destination we define for all  $k$  such that  $1 \leq k \leq N_j$

$$g_k^{ij}(x^j(\cdot), u^j(\cdot)) = \begin{cases} 1 & \text{if } f_j(\|A^j x^j(\cdot)\| + \|u^j(\cdot)\|) = k \\ 0 & \text{otherwise.} \end{cases}$$

The function  $f_j$  computes the time spent on link  $\ell_j$  by vehicles joining the link at a given time. Therefore the appropriate state variable may be incremented. Accordingly we obtain the first order difference equation

$$x^{ij}(n+1) = A^{ij} x^{ij}(n) + g^{ij}(x^j(n), u^j(n)) \cdot u^{ij}(n)$$

where

$$g^{ij}(x^j, u^j) = [g_1^{ij}(x^j, u^j) \cdots g_{N_j}^{ij}(x^j, u^j)]^T .$$

Next let

$$G^j(x^j, u^j) = \text{diag}\{g^{1j}(x^j, u^j) \cdots g^{|D|j}(x^j, u^j)\}$$

which is of dimension  $(N_j|D| \times |D|)$ . Then it is evident that

$$x^j(n+1) = A^j x^j(n) + G^j(x^j(n), u^j(n))u^j(n) .$$

The following output equation is also defined

$$y^{ij}(n+1) = C^{ij} x^{ij}(n)$$

where

$$C^{ij} = [1 \ 0 \ \cdots \ 0]_{1 \times N_j} .$$

The extension to link output is made as usual. Let

$$y^j(n) = [y^{1j}(n) \cdots y^{|D|j}(n)]_{|D| \times 1}^T$$

and

$$C^j = \text{diag}\{C^{1j} \cdots C^{|D|j}\}$$

which is of dimension  $|D| \times (N_j|D|)$ . Then

$$y^j(n) = C^j x^j(n) .$$

### 3.2. The network dynamical equation

The network dynamical equation is very easily written by exploiting the modular structure of the system. We define the following notation.

(i)  $X(n) = [x^1(n)^T \cdots x^{|E|}(n)^T]^T$ .

Thus  $X(n)$  is a  $(\sum_{j=1}^{|E|} N_j|D|) \times 1$  dimensional vector.

(ii)  $U(n) = [u^1(n)^T \cdots u^{|E|}(n)^T]^T$ .

Thus  $U(n)$  is a  $(|D||E|) \times 1$  dimensional vector.

(iii)  $A = \text{diag}\{A^1 \cdots A^{|E|}\}$ .

Thus  $A$  is of dimension  $(\sum_{j=1}^{|E|} N_j|D|) \times (\sum_{j=1}^{|E|} N_j|D|)$ .

(iv)  $G(X, U) = \text{diag}\{G^1(x^1, u^1) \dots G^{|E|}(x^{|E|}, u^{|E|})\}$ .

Thus  $G(X, U)$  is of dimension  $(\sum_{j=1}^{|E|} N_j |D|) \times (|D| |E|)$ .

(v)  $C = \text{diag}\{C^1 \dots C^{|E|}\}$ .

Thus  $C$  is of dimension  $(|E| |D|) \times (\sum_{j=1}^{|E|} N_j |D|)$ .

The required dynamical equation is for the whole network is then

$$X(n+1) = AX(n) + G(X(n), U(n))U(n) .$$

The homogeneous system remains linear first order and the overall system is also first order. The network output equation is

$$Y(n) = CX(n) .$$

### 3.3. Definition of the feasible region for routing decisions

The following are the input and state constraints of the dynamical system.

(i) Flow conservation equations.

For all vertices  $v_p \in V$  we have the following equations for each destination  $d_i \in D$ :

$$d_i \neq v_p \implies \sum_{\ell_j \in S_{v_p}} u^{ij}(n) = \sum_{\ell_j \in P_{v_p}} y^{ij}(n) + r^{ip}(n) , \quad n = 0, 1 \dots$$

$$d_i = v_p \implies \forall \ell_j \in S_{v_p} , \quad u^{ij}(n) = 0 , \quad n = 0, 1 \dots$$

where

$S_{v_p}$  = Set of successor links of node  $v_p$ ;

$P_{v_p}$  = Set of predecessor links of node  $v_p$ ;

$r^{ip}(n)$  = Number of new vehicles entering the network at node  $v_p$  at time  $(n+1)$  and traveling to destination  $d_i \in D$  .

$R(n) = [r^{11}(n)r^{21}(n) \dots r^{|D|1}(n)r^{12}(n) \dots r^{|D| |V|}(n)]^T$  is the  $(|V| |D| \times 1)$  vector of travel demands for all vertices  $v_p \in V$  and all destinations  $d_i \in D$ .

(ii) Headway constraints.

We define for each  $\ell_j \in E$  the quantity  $K_j$  as

$$K_j = \frac{\Delta t}{t_{\text{headway}}} \times \text{number of lanes on } \ell_j$$

where  $\Delta t$  is the real-time value of the sampling interval and  $t_{\text{headway}}$  is the specified separation time between two vehicles. The constraint equation is

$$\sum_{i=1}^{|D|} x_k^{ij}(n) \leq K_j, \quad 1 \leq k \leq N_j, \quad n = 0, 1, \dots$$

(iii) Capacity constraints.

For all  $\ell_j \in E$  we require,

$$\sum_{i=1}^{|D|} \|x^{ij}(n)\| \leq c_j, \quad n = 0, 1, \dots$$

where  $c_j$  is the capacity of link  $\ell_j$ .

The three equations in (i)–(iii) together with a given state  $X(n)$  and demand  $R(n)$  define a feasible region for the input  $U(n)$ . We refer to this set of admissible controls by the function  $\Omega(X(n), R(n))$ . Formally,

$$\begin{aligned} \Omega(X, R) = \left\{ U \in \mathbb{N}^{|D| \times |E|} : \right. & \left[ (\forall v_p \in V)(\forall d_i \in D) \left( d_i \neq v_p \implies \left[ \sum_{\ell_j \in S_{v_p}} u^{ij} = \sum_{\ell_j \in P_{v_p}} C^{ij} x^{ij} + r^{ip} \right] \right. \right. \\ & \left. \left. \wedge d_i = v_p \implies \left[ (\forall \ell_j \in S_{v_p}) u^{ij} = 0 \right] \right) \right] \\ & \wedge \left[ (\forall \ell_j \in E) \left( \left[ i \leq k \leq N_j \implies \|(A^j x^j + G^j(x^j, u^j)u^j)_k\| \leq K_j \right] \right. \right. \\ & \left. \left. \wedge \left[ \|A^j x^j + G^j(x^j, u^j)u^j\| \leq c_j \right] \right) \right] \left. \right\}. \end{aligned}$$

(The notation  $(x^j)_k$  extracts all  $x_k^j$  components from  $x^j$ .)

**Remark:** For simplicity of notation, we have omitted possible time dependencies of certain variables in the above presentation. In general,  $K_j$ ,  $c_j$ , and even the impedance function  $f_j(\cdot)$  could be time-varying.

## 4. The Mathematical Statement of the Optimal Control Problem

From Sections 3.2 and 3.3 we obtain the equation of motion of the system and the control constraint set function, i.e., the set of feasible in flows. These are

$$X(n+1) = AX(n) + G(X(n), U(n))U(n)$$

$$U(n) \in \Omega(X(n), R(n)) .$$

(For simplicity of notation, we will abbreviate  $G(X(n), U(n))U(n)$  as  $GU(n)$  in this section.) With respect to the boundary conditions, we define two problems. One is a fixed-endtime free-endpoint problem and the other is a free-endtime fixed-endpoint problem.

#### 4.1. Fixed-endtime free-endpoint problem

For obvious reasons we define a fixed integer terminal time denoted by  $N_f$ . The boundary conditions are then

$$\begin{aligned} X(n_0) &= X_0 \\ X(N_f) &\in \mathbb{N}_{\sum_{j=1}^{|E|N_f|D|}} . \end{aligned}$$

Having obtained our two point boundary value problem we proceed to define the cost. For each state trajectory  $X(n; n_0, N_f)$  the associated cost  $J$  is defined as

$$J(X(n; n_0, N_f)) = \sum_{j=n_0}^{N_f} \|X(j)\| + F(X(N_f)) .$$

The rationale for our definition is as follows. The norm of the state represents the number of vehicles on the network during one sampling interval. Thus if the sampling time is one hour then the first term of the cost function gives us the total number of vehicle-hours incurred within the time horizon. The second term represents a penalty for not clearing the network within the allotted time  $N_f$ , since all non-zero terms in  $X(N_f)$  represent vehicles which have not yet reached their destinations.

The optimal cost as a function of the initial state may be defined as

$$\begin{aligned} J^*(X_0) = \min \left\{ J(X(n; n_0, N_f)) \quad : \quad X(n_0) = X_0, X(j+1) = AX(j) + GU(j) , \right. \\ \left. \text{and } U(j) \in \Omega(X(j), R(j)) , n_0 \leq j < N_f \right\} . \end{aligned}$$

Pick points  $n_1, \dots, n_p$  such that  $n_0 < n_1 < \dots < n_p < N_f$ . From the definition of  $J$  it is evident that

$$J(X(n; n_0, N_f)) = J(X(n; n_0, n_1)) + \dots + J(X(n; n_{p-1}, n_p)) + J(X(n; n_p, N_f))$$

where if  $n_k < N_f$  then  $J(X(n; n_0, n_k)) = \sum_{j=n_0}^{n_k} \|X(j)\|$ . Thus  $J$  obeys the principle of optimality by additivity. Consequently we develop a recursive dynamic programming equation which may be used to solve the optimization problem.



## 4.2. Dynamic programming recursion equation for the fixed-endtime free-endpoint problem

### (A) Forward recursion equation

For a given initial state  $X_0$ , we define the cost  $J_F$  to reach state  $X$  at time  $n$  along trajectory  $X(j; n_0, n)$  to be

$$J_F(X, X(j; n_0, n), n) = \begin{cases} \sum_{j=n_0}^n \|X(j)\| & \text{if } n < N_f \\ \sum_{j=n_0}^n \|X(j)\| + F(X) & \text{if } n = N_f \end{cases}$$

where  $X(j; n_0, n)$  must satisfy  $X(n_0) = X_0$  and  $X(n) = X$ . Then

$$J(X(n; n_0, N_f)) = J_F(X(N_f), X(j; n_0, N_f), N_f) .$$

For each feasible  $X_n$  at a time instant  $n$  we define the following forward value function  $V_F$ .

$$V_F(X_n, n) = \min \left\{ J_F(X_n, X(j; n_0, n), n) : X(n) = X_n, X(n_0) = X_0, X(j+1) = AX(j) + GU(j), \right. \\ \left. \text{and } U(j) \in \Omega(X(j), R(j)), n_0 \leq j < n \right\} .$$

Thus  $V_F(X_n, n)$  represents the lowest cost to reach state  $X_n$  at time  $n$  among all admissible state trajectories. It is evident that

$$J^*(X_0) = \min \{ V_F(X_{N_f}, N_f) : X_{N_f} \in \mathcal{X}_{N_f} \}$$

where

$$\mathcal{X}_{N_f} = \left\{ X : X(N_f) = X, X(n_0) = X_0, X(j+1) = AX(j) + GU(j), \right. \\ \left. \text{and } U(j) \in \Omega(X(j), R(j)), n_0 \leq j < N_f \right\}$$

is the set of all feasible terminal states for this fixed endtime problem.

The initial condition on  $V_F(X, n)$  is

$$V_F(X_0, n_0) = \|X_0\| .$$

By the principle of optimality the recursion equation is as follows.

Case  $n_0 < n < N_f$ :

$$V_F(X_n, n) = \|X_n\| + \min_{X_{n-1} \in \mathcal{X}_{n-1}} V_F(X_{n-1}, n-1)$$

where

$$\mathcal{X}_{n-1} = \{X : X_n = AX + GU, U \in \Omega(X, R(n-1))\} .$$

Case  $n = N_f$ :

$$V_F(X_{N_f}, N_f) = F(X_{N_f}) + \|X_{N_f}\| + \min_{X_{N_f-1} \in \mathcal{X}_{N_f-1}} V_F(X_{N_f-1}, N_f - 1)$$

where  $\mathcal{X}_{N_f-1}$  is defined as above.

### (B) Backward recursion equation

Analogous to the prior case we define the cost to complete from a given state  $X$  at a given time  $n$  along a trajectory  $X(j; n, N_f)$  as

$$J_B(X, X(j; n, N_f), n) = \sum_{j=n}^{N_f} \|X(j)\| + F(X(N_f))$$

where  $X(j; n, N_f)$  must satisfy  $X(n) = X$ . Then

$$J(X(n; n_0, N_f)) = J_B(X(n_0), X(j; n_0, N_f), n_0) .$$

For each feasible  $X_n$  at time instant  $n$  we define

$$V_B(X_n, n) = \min \left\{ J_B(X_n, X(j; n, N_f), n) : \begin{array}{l} X(n) = X_n, X(j+1) = AX(j) + GU(j) \\ \text{and } U(j) \in \Omega(X(j), R(j)), n \leq j < N_f \end{array} \right\}$$

as the backward value function at state  $X_n$  at time  $n$ . Accordingly

$$J^*(X_0) = V_B(X_0, X(j; n_0, N_f), n_0) .$$

For each  $X \in \mathcal{X}_{N_f}$  ( as defined in part A) the boundary condition is

$$V_B(X, N_f) = \|X\| + F(X) .$$

By the principle of optimality, for all  $n$  such that  $n_0 \leq n < N_f$ , the backward recursion equation is

$$V_B(X, n) = \|X\| + \min_{U \in \Omega(X, R(n))} V_B(AX + GU, n+1) .$$

While the backward recursion equation appears more elegant in its formulation, we consider the forward recursion equation to be more useful. In the absence of a definite endpoint the set of feasible terminal states is too large to allow computation. The well defined initial condition on the other hand allows forward chaining through the solution space in a well defined recursive manner. An example illustrating the pruning of forward search using Dijkstra's implementation of dynamic programming is presented in Section 5.

### 4.3. Free-endtime fixed-endpoint problem

Here the control objective is to clear the network in as short a time as possible. Thus the target state is the zero vector. The dynamical equation and control constraint set  $\Omega(X, R(n))$  are as usual.

We define the cost associated with a state trajectory to be

$$J(X(n; n_0, N)) = \sum_{j=n_0}^N \|X(j)\| \quad \text{where } N \in \mathbb{N}.$$

The optimal cost  $J^*$  as a function of the initial state  $X_0$  at time  $n_0$  is

$$J^*(X_0, n_0) = \min \left\{ J(X(n; n_0, N)) : X(n_0) = X_0, X(N) = 0, X(j+1) = AX(j) + GU(j), \right. \\ \left. \text{and } U(j) \in \Omega(X(j), R(j)), n_0 \leq j < N, N \in \mathbb{N} \right\}.$$

However, if this problem is viewed as an infinite horizon, i.e., all state trajectories are assumed defined on  $[n_0, \infty)$ , then the problem may be viewed as free-endpoint free-endtime. We define

$$J(X(n; n_0)) = \sum_{j=n_0}^{\infty} \|X(j)\|$$

and

$$J^*(X_0, n_0) = \min \left\{ J(X(n; n_0)) : X(n_0) = X_0, X(j+1) = AX(j) + GU(j), \right. \\ \left. \text{and } U(j) \in \Omega(X(j), R(j)), n_0 \leq j \right\}.$$

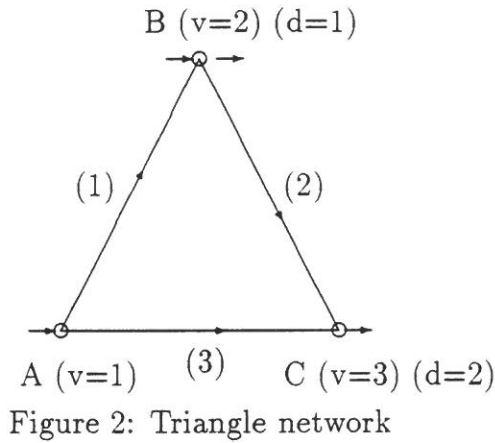
Thus if  $J^*(X_0, n_0)$  exists then  $\lim_{j \rightarrow \infty} \|X(j)\| = 0$  which means that the target is implicitly achieved and the exogenous demand  $R$  terminates in finite time. Moreover the requirements of integrality force  $X(j) = 0$  in finite time.

## 5. An Example Illustrating the Forward DP Recursion

The triangle network of Figure 2 is used. A and B are valid origin vertices and B and C are valid destination vertices. Accordingly,

$$\begin{aligned} V &= \{v_1 = A, v_2 = B, v_3 = C\} \\ E &= \{\ell_1 = (A, B), \ell_2 = (B, C), \ell_3 = (A, C)\} \\ D &= \{d_1 = B, d_2 = C\}. \end{aligned}$$

The impedance functions are presented in tabular form.



Number of Vehicles	Travel Time-AC	Travel time-AB/BC
1	20	15
2	20	17
3	30	22
4	40	33
5	60	45
6	100	70

Capacity = 6  
Max travel time = 100/70

Table 1: Impedance Functions

We wish to find a system optimal routing for the following demand pattern:

$$\begin{aligned} r^{11}(0) &= 1 \\ r^{21}(0) &= 1 \\ r^{21}(10) &= 3. \end{aligned}$$

Our objective is to clear the network, as in Section 4.3. To better illustrate the working of the DP algorithm an equivalent but different formulation of the cost function is used. We define

$$J(X(n; 0, N)) = \sum_{n=0}^{N-1} \sum_{d_i \in D} \sum_{\ell_j \in E} \left[ \sum_{k=1}^{N_j} k g_k^{ij}(x^j(n), u^j(n)) \right] u^{ij}(n).$$

Thus the total cost of traveling a link is incurred as soon as the vehicle is routed to a link.

The flow of the algorithm is presented as a tree in Fig. 3 and the important computations are shown in Table 2. (By “Terminal” we mean a node that will reach the terminal state without any further inputs. The “not expanded” nodes are ones that are not expanded further since they cannot yield optimal solutions.)

0: J=0; (0,0,0)	01; J=34; (2,0,0)	011: J=169; (5,0,0)	Not expanded
		012: J=124; (2,0,3)	0121: J=139; Terminal
		013: J=120; (0,3,1)	0131: J=135; Not Expanded
		014: J=116; (0,2,2)	0141: J=131; Not Expanded
	02; J=35; (1,0,1)	021: J=134; (4,0,1)	Not expanded
		022: J=170; (1,0,4)	Terminal
		023: J=104; (3,0,2)	0231: J=138; Terminal
		024: J=112; (2,0,3)	0241: J=127; Terminal
		and OPTIMAL	

$(x, y, z) = (\text{number of vehicles on AB, number of vehicles on BC, number of vehicles on AC})$

Figure 3: DP Tree

## 6. Some Extensions

### 6.1. Inclusion of background traffic

The model of Section 3 assumes that all vehicles are to be routed during their travel in the network. We now discuss how to include background traffic into the model. By background traffic we mean vehicles whose routing is not part of the optimization but rather is a known deterministic function of time. This function could for instance be based on historical data, on shortest path calculations, or on some traffic equilibrium solution.

We need to distinguish between “guided” and “background” vehicles. Let  $X_g(\cdot)$  denote the state vector (as defined in Section 3.2) for guided vehicles and  $X_b(\cdot)$  that for background vehicles. Similarly for  $U_g(\cdot)$  &  $U_b(\cdot)$ ,  $Y_g(\cdot)$  &  $Y_b(\cdot)$  and  $R_g(\cdot)$  &  $R_b(\cdot)$ . Finally, define

$$\begin{aligned} X_t(n) &= X_g(n) + X_b(n) \\ U_t(n) &= U_g(n) + U_b(n) \\ R_t(n) &= R_g(n) + R_b(n) \end{aligned}$$

for all  $n$ , where  $t$  stands for “total.”

For the background traffic, by assumption,

$$u_b^{ij}(n) = f_b(n, Y_b(n), r_b^{ip}(n))$$

where  $f_b$  is a known function and where  $v_p$  is the origin vertex of  $\ell_j$ . In order to obtain a

meaningful problem formulation, we assume that

$$U_b(n) \in \Omega(X_b(n), R_b(n))$$

i.e., the routing of the background traffic should satisfy the flow conservation equations and the headway and capacity constraints in the absence of guided vehicles ( $X_g(\cdot) \equiv 0$  and  $R_g(\cdot) \equiv 0$ ).

The network dynamical equation now consists of two parts:

$$\begin{aligned} X_g(n+1) &= AX_g(n) + G(X_t(n), U_t(n))U_g(n) \\ X_b(n+1) &= AX_b(n) + G(X_t(n), U_t(n))U_b(n) \\ Y_g(n) &= CX_g(n) \\ Y_b(n) &= CX_b(n) \end{aligned}$$

with appropriate initial conditions  $X_b(n_0)$  and  $X_g(n_0)$ . The feasible region for routing decisions  $U_g(\cdot)$  is defined by the flow conservation equations

$$\begin{aligned} \sum_{\ell_j \in S_{v_p}} u_g^{ij}(n) &= \sum_{\ell_j \in P_{v_p}} y_g^{ij}(n) + r_g^{ip}(n) && [d_i \neq v_p] \\ \forall \ell_j \in S_{v_p}, u_g^{ij}(n) &= 0 && [d_i = v_p] \end{aligned}$$

and by the headway and capacity constraints

$$\begin{aligned} \sum_{i=1}^{|D|} (x_t^{ij}(n))_k &\leq K_j, \quad 1 \leq k \leq N_j, \quad n = 0, 1, \dots \\ \sum_{i=1}^{|D|} \|x_t^{ij}(n)\| &\leq c_j, \quad n = 0, 1, \dots \end{aligned}$$

In other words, the headway and capacity constraints are in terms of the total traffic, while the flow conservation equations are in terms of the guided traffic, since by assumption the routing of the background traffic satisfies its own set of flow conservation equations.

Finally, the cost of a state trajectory

$$X(n; n_0, N_f) = \begin{bmatrix} X_g(n; n_0, N_f) \\ X_b(n; n_0, N_f) \end{bmatrix}$$

could in general depend on both types of traffic:

$$J(X(n; n_0, N_f)) = \sum_{j=n_0}^{N_f} \alpha_1 \|X_g(j)\| + \beta_1 \|X_b(j)\| + \alpha_2 F(X_g(N_f)) + \beta_2 F(X_b(N_f))$$

where  $\alpha_i$  and  $\beta_i$   $i = 1, 2$  are weighting factors.





$$\begin{aligned} \forall \ell_j \in S_{v_p}, u^{ij}(n) &= 0 & [d_i = v_p] \\ \forall \ell_j \in P_{v_p}, u_{\text{block}}^{ij}(n) &= 0 & [d_i = v_p]. \end{aligned}$$

Observe that no modification of the cost  $J$  is necessary since the extra travel time incurred by the blocked vehicles is automatically summed up in  $\|X(j)\|$ .

Without any further constraints, the increase in the dimensionality of the decision space due to the consideration of all feasible blocking controls is likely to render the problem intractable even for small networks. A reasonable heuristic would be to allow blocking controls *only* when the feasible region  $\Omega(X(n), R(n))$  (as defined in Section 3) is empty, i.e., when there is no routing decision  $U(n)$  that satisfies (1) the flow conservation equations, (2) the headway constraints, and (3) the capacity constraints.

## 7. Conclusion

We have approached the problem of dynamic traffic assignment in networks from the viewpoint of dynamical systems and have proposed a new model for this problem. This model is more detailed than typical macroscopic models, yet it avoids complete microscopic detail by grouping vehicles into platoons irrespective of origin node and time of entry to the network. It has been observed in the literature that traffic models based on impedance functions alone or on link outflow rate functions alone suffer from certain deficiencies. By using impedance functions to first determine a minimum travel time for a vehicle on a link and then by using headway constraints and blocking controls (Section 6.2) to limit the outflow from a link, our model effectively combines both of these approaches. We believe that this feature of our model is interesting because it closely resembles what happens in a microscopic simulation of traffic (e.g., the INTEGRATION traffic simulator of Van Aerde [9]).

The work that we have presented opens several avenues for future investigations. Among these, we wish to mention the following.

- Study of the computational limitations of this approach. How large a network can be solved optimally using the dynamic programming algorithm of Section 4?
- Determination of structural properties of optimal routing policies. Such properties could then be used to develop heuristics to accelerate the forward search.
- Investigation of the real-time feasibility of this approach, where routing decisions at each time instant would be made on the basis of a limited lookahead forward search

and where the state of the model would be updated continuously on the basis of the data collected from the various sensors in the network.

## Acknowledgment

The authors wish to acknowledge stimulating discussions with Steve Underwood and Karl Wunderlich.

## References

- [1] M. Carey. Optimal time-varying flows on congested networks. *Operations Research*, 35(1):58-69, 1987.
- [2] T. L. Friesz, J. Luque, R. L. Tobin, and B. Wie. Dynamic network traffic assignment considered as a continuous-time optimal control problem. *Operations Research*, 37(6):893-901, 1989.
- [3] B. N. Janson. A convergent algorithm for dynamic traffic assignment. In *70th Transportation Research Board Meeting*, January 1991.
- [4] D. E. Kaufman, J. Lee, and R. L. Smith. Anticipatory traffic modeling and route guidance in Intelligent Vehicle-Highway Systems. IVHS Technical Report 90-2, University of Michigan, February 1990.
- [5] D. E. Kaufman and R. L. Smith. Minimum travel time paths in dynamic networks with application to Intelligent Vehicle-Highway Systems. IVHS Technical Report 90-11, University of Michigan, November 1990.
- [6] D. K. Merchant and G. L. Nemhauser. A model and an algorithm for the dynamic traffic assignment problem. *Transportation Science*, 12(3):183-199, 1978.
- [7] M. Papageorgiou, J. C. M. Banos, and A. Messmer. Optimal control of multideestination traffic networks. In *Proc. 29th IEEE Conf. on Decision and Control*, pages 1355-1361, Honolulu, HI, December 1990.
- [8] L. Saxton, Guest Editor. Special issue on Intelligent-Vehicle Highway Systems. *IEEE Trans. Vehicular Technology*, 40(1):1-158, February 1991.
- [9] M. Van Aerde and S. Yagar. Dynamic integrated freeway/traffic signals networks: A routing-based modelling approach. *Transportation Research A*, 22A(6):445-453, 1988.
- [10] K. Wunderlich. Time-variant travel cost calculation under anticipatory routing. IVHS Technical Report 90-5, University of Michigan, August 1990.

Node #	Time	State	Demand	Inputs	$g$	Cost
0	0	Start State				
01	0 1	$x_{17}^{11} = 1, x_{17}^{21} = 1$	$r^{11} = 1, r^{21} = 1$	$u^{11} = 1, u^{21} = 1$	$g_{17}^{11} = 1, g_{17}^{21} = 1$	34
02	0 1	$x_{15}^{11} = 1, x_{20}^{23} = 1$	$r^{11} = 1, r^{21} = 1$	$u^{11} = 1, u^{23} = 1$	$g_{15}^{11} = 1, g_{20}^{23} = 1$	35
011	10 11	$x_8^{11} = 1, x_8^{21} = 1$ $x_7^{11} = 1, x_7^{21} = 1$ $x_{45}^{21} = 3$	$r^{23} = 3$	$u^{21} = 3$	$g_{45}^{21} = 1$	169
012	10 11	$x_8^{11} = 1, x_8^{21} = 1$ $x_7^{11} = 1, x_7^{21} = 1$ $x_{30}^{23} = 3$	$r^{23} = 3$	$u^{23} = 3$	$g_{30}^{23} = 1$	124
013	10 11	$x_8^{11} = 1, x_8^{21} = 1$ $x_7^{11} = 1, x_7^{21} = 1$ $x_{33}^{21} = 2, x_{20}^{23} = 1$	$r^{23} = 3$	$u^{21} = 2, u^{23} = 1$	$g_{33}^{21} = 1, g_{20}^{23} = 1$	120
014	10 11	$x_8^{11} = 1, x_8^{21} = 1$ $x_7^{11} = 1, x_7^{21} = 1$ $x_{22}^{21} = 1, x_{25}^{23} = 2$	$r^{23} = 3$	$u^{21} = 1, u^{23} = 2$	$g_{22}^{21} = 1, g_{25}^{23} = 1$	116
021	10 11	$x_6^{11} = 1, x_{23}^{23} = 1$ $x_5^{11} = 1, x_{10}^{23} = 1$ $x_{33}^{21} = 3$	$r^{23} = 3$	$u^{21} = 3$	$g_{33}^{21} = 1$	134
022	10 11	$x_6^{11} = 1, x_{11}^{21} = 1$ $x_5^{11} = 1, x_{10}^{23} = 1$ $x_{45}^{23} = 3$	$r^{23} = 3$	$u^{23} = 3$	$g_{45}^{23} = 1$	170
023	10 11	$x_6^{11} = 1, x_{11}^{23} = 1$ $x_5^{11} = 1, x_{10}^{23} = 1$ $x_{21}^{21} = 2, x_{25}^{23} = 1$	$r^{23} = 3$	$u^{21} = 2, u^{23} = 1$	$g_{22}^{21} = 1, g_{25}^{23} = 1$	104
024	10 11	$x_6^{11} = 1, x_{11}^{23} = 1$ $x_5^{11} = 1, x_{10}^{23} = 1$ $x_{17}^{21} = 1, x_{30}^{23} = 2$	$r^{23} = 3$	$u^{21} = 1, u^{23} = 2$	$g_{17}^{21} = 1, g_{30}^{23} = 1$	112
0111	Not expanded by algorithm					
0121	17	$x_1^{11} = 1, x_1^{21} = 1$ $x_{39}^{21} = 3$		$u^{22} = 1$	$g_{15}^{22} = 1$	139
0131	17	$x_1^{11} = 1, x_1^{21} = 1$ $x_{27}^{21} = 2, x_{14}^{23} = 1$		$u^{22} = 1$	$g_{15}^{22} = 1$	135
0141	17	$x_1^{11} = 1, x_1^{21} = 1$ $x_{16}^{21} = 2, x_{18}^{23} = 1$		$u^{22} = 1$	$g_{15}^{22} = 1$	131
0211	Not expanded by algorithm					
0221	Terminal state					
0231	32	$x_1^{21} = 2, x_4^{23} = 1$		$u^{22} = 2$	$g_{17}^{22} = 1$	138
0241	27	$x_1^{21} = 1, x_{14}^{23} = 2$		$u^{22} = 1$	$g_{15}^{22} = 1$	127

Table 2: Computational data for nodes of DP tree

All data not stated in the table is zero