

**Minimum Travel Time Paths in Dynamic  
Networks with Application to  
Intelligent Vehicle/Highway Systems**

David E. Kaufman  
Robert L. Smith  
Department of Industrial and Operations Engineering  
University of Michigan  
Ann Arbor, Michigan 48109

Technical Report 90-34



# Minimum Travel Time Paths in Dynamic Networks with Application to Intelligent Vehicle/Highway Systems\*

David E. Kaufman

Robert L. Smith

Department of Industrial and Operations Engineering

University of Michigan

Ann Arbor, Michigan 48109

November 12, 1990

## Abstract

Motivated by issues arising from route guidance for drivers using real-time information in an Intelligent Vehicle/Highway Systems (IVHS) environment, we consider a class of cyclic dynamic networks characterized by (possibly random) arc travel times which depend on the time when travel on the arc is begun. We provide a condition which is sufficient for a dynamic principle of optimality to hold, so that natural extensions of conventional shortest-path methods solve for the minimum travel time path with no more computational effort in the worst case than in a static cyclic network. We also discuss how static shortest-path algorithms may be used to improve previously proposed algorithms for cases in which the principle of optimality does not hold. The potential benefits of dynamic network modelling in the IVHS environment are demonstrated through modifications to the INTEGRATION traffic simulation program.

---

\*This research was supported in part by the IVHS Program at the University of Michigan

# 1 Introduction

An element of the emerging field of Intelligent Vehicle/Highway Systems (IVHS) is the enhancement of travel quality by improving drivers' route choice. The traffic network of freeway corridors and surface streets can be modeled as a directed graph, with travel times associated with each arc of the graph, corresponding to a link in the network. Ideally, IVHS would monitor the network through surveillance technologies, measuring these travel times in real time and disseminating them to on-board computers and displays installed in individual vehicles. Then all that would remain is for the driver to program his destination (and perhaps his current location, if Automatic Vehicle Location technology lags) and his personal guidance unit would employ a shortest-path algorithm to provide him with an optimal, i.e., minimum travel time route.

This approach overlooks the rapid real-time change which often characterizes road networks, particularly in rushhour conditions when benefits of IVHS may be greatest. Changing traffic volumes and incidents causing lane blockages will alter the time required to traverse links of the network. One way to approach this dynamic character of traffic networks is to update the system measurements very frequently, so that the inputs to the shortest-path algorithm are as current as possible. However, this does not address the inherent inaccuracy of such an approach. In particular, since link travel times will dynamically change, the costs should be modelled as varying with time. (Of course, this creates a related problem, not addressed in this paper, of forecasting these costs for each link of the network over an appropriate horizon.) We will provide a rigorous foundation for the modelling and solution of these networks, and we will demonstrate potential benefits to drivers who have the opportunity to determine fastest paths given the dynamic network model.

## 2 Literature Review and Synopsis

The fastest path problem with time-dependent travel times was first considered by Cooke and Halsey [2]. They cited Bellman's principle of optimality [1] to write a functional equation which implicitly defined a network having states incorporating not only the current location in the network (i.e., node), but also the current time. Travel times are assumed to be known for each arc at times  $t_0, t_0 + \Delta, t_0 + 2\Delta, \dots$  and are multiples of  $\Delta$ , for some  $\Delta > 0$ . Their solution algorithm proceeds recursively on the maximum number of nodes visited in a path, and if some upper bound  $T_{max}\Delta$  can be found for the optimal trip time, then the computational effort is  $O(N^3T_{max})$ .

Dreyfus [5] observed that Cooke and Halsey's implicit expansion of the state space and restriction to discrete time intervals can be avoided, and the problem solved by a generalization of Dijkstra's method [4], as efficiently as for static shortest path problems (constant

link travel times). However, we will provide a counterexample, and then rigorously establish conditions under which generalizations of conventional static shortest-path algorithms may be applied. We will also discuss how Cooke and Halsey's algorithm may be accelerated by the application of a generalized static shortest-path algorithm when the latter is not guaranteed to provide an optimal solution.

In [6], Hall considers networks with random time-dependent travel times and demonstrates that in general, static algorithms cannot be applied. He notes that adaptive routing is necessary, since policies in which the path chosen from an intermediate node depends on the time of arrival at that node may have expected travel times less than that of any fixed routing policy. We will furnish conditions which allow these networks to be solved by static algorithms at computational cost similar to that of static deterministic shortest path solutions. Because these conditions are somewhat restrictive, we will discuss how static methods can be used as heuristics to improve performance of the general-case algorithm proposed by Hall.

The minimum travel time problem with given time periods during which waiting at nodes is permitted is considered by Halpern [7]. He provides an algorithm which stores the possible arrival times at each node as they are discovered. In [9], Orda and Rom consider three classes of networks: first, where vehicles may wait an unlimited duration at any node; second, vehicles may wait only at the origin node of the trip; and third, where vehicles are not permitted to wait at any node. We address the last of these cases, where vehicles are not permitted to wait at nodes. The discussion of this case by Orda and Rom was limited to demonstrating that there may be no finite optimal path. For the IVHS environment we will assume conditions on link travel times which assure existence of a finite optimal path.

Finally, we will present computational results demonstrating the travel-time reduction which can be realized by a vehicle selecting its route in a vehicular traffic network by dynamic (or anticipatory) fastest-path calculation. These results are the product of modifications to the INTEGRATION traffic simulation program (Van Aerde and Yagar [14]), which is ideally suited to investigation of varying methods of route choice for individual drivers.

### 3 Modelling and Solution in Deterministic Dynamic Networks

Consider a network  $(\mathcal{N}, \mathcal{A})$  with node set  $\mathcal{N} = \{1, \dots, N\}$  and arc set  $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ . We assume that there exist paths from node 1 to all other nodes. Let  $c_{ij}(t)$  be the travel time on arc  $(i, j)$  departing node  $i$  at time  $t$  for  $t \in T$ , where the index set  $T$  and the real-valued function  $c_{ij}(\cdot)$  satisfy:

- 1)  $0 \in T$

2)  $t + c_{ij}(t) \in T$  for all  $t \in T$ .

We assume the absence of negative cycles, i.e., sequences of nodes  $i_1, i_2, \dots, i_M, i_{M+1}$  with  $i_{M+1} = i_1$  such that  $\sum_{k=1}^M c_{i_k i_{k+1}}(t_k) < 0$  where  $t_{k+1} = t_k + c_{i_k i_{k+1}}(t_k)$  for  $k = 1, \dots, M$  and  $t_1 \in T$ . To ensure conditions 1) and 2), we may have for example that  $T = \{t : t \geq 0\}$  and  $c_{ij}(\cdot) \geq 0$ , or we may select some arbitrary time unit and set  $T = \{0, 1, 2, \dots\}$  in these units if we require that  $c_{ij}(t)$  is given in integer numbers of these units. Such a network will be called *dynamic*. Our task is to determine a fastest path from node 1 to node  $N$ , assuming that we begin the trip at the current time  $t = 0$ .

Define the optimal value function  $f_i(t)$  to be the minimum trip time over all paths from node  $i$  to node  $N$  departing node  $i$  at time  $t$ . Then Cooke and Halsey's functional equation is:

$$f_i(t) = \begin{cases} \min_{j \neq i} \{c_{ij}(t) + f_j(t + c_{ij}(t))\} & \text{for } i = 1, \dots, N-1; t \in T \\ 0 & \text{for } i = N; t \in T \end{cases} \quad (1)$$

and the problem is to find  $f_1(0)$ . Implicitly, this defines a network with node set  $\mathcal{N}' = \{(i, t) : i \in \mathcal{N}, t \in T\}$  and arc set  $\mathcal{A}' = \{((i, t), (j, u)) : (i, j) \in \mathcal{A}; t, u \in T; c_{ij}(t) = u - t\}$ . Since this network has fixed travel times, we will refer to it as the *expanded static network*. The expansion of the state space increases the computational effort required to solve the problem; Cooke and Halsey's solution algorithm requires  $T$  to be finite, with a known upper bound  $T_{max}$  on  $f_1(0)$ , and requires  $O(N^3 T_{max})$  time. In comparison, a network with fixed travel times can be solved in  $O(N^2)$  time, or  $O(N \log N)$  time for sparse networks [3].

Dreyfus [5] observed that the problem could be solved by generalizing the method of Dijkstra [4]. He proposes that labels  $v_i$  be maintained, and at each iteration the node  $i$  with minimum  $v_i$  be *expanded*; that is,  $v_i$  is made permanent and the labels  $v_j$  are updated according to the formula

$$v_j = \min\{v_j, v_i + c_{ij}(v_i)\}$$

for each  $j$  such that  $(i, j) \in \mathcal{A}$ . The claim is that at termination,  $v_i$  represents the optimal trip time from node 1 to node  $i$  for all  $i \in \mathcal{N}$ . However, Figure 1 demonstrates that this procedure may fail in dynamic networks. The procedure would identify (1,3,4) as the optimal path from node 1 to node 4, with trip time 30, but the path (1,2,3,4) has a trip time of 25.

-Dreyfus' suggestion implicitly assumes the following functional equation:

$$f_j = \begin{cases} \min_{i \neq j} \{f_i + c_{ij}(f_i)\} & \text{for } j = 2, \dots, N \\ 0 & \text{for } j = 1 \end{cases} \quad (2)$$

where  $f_i$  is the minimum trip time over all paths from node 1 to node  $i$  departing node 1 at time 0. This assumes Bellman's principle of optimality [1] in its forward-recursive form. For (2) to hold, it must be true that on an optimal path, each intermediate node is reached as soon as possible. Clearly, this is not the case in Figure 1.

However, as we will show, the following assumption [12] will suffice to imply (2).

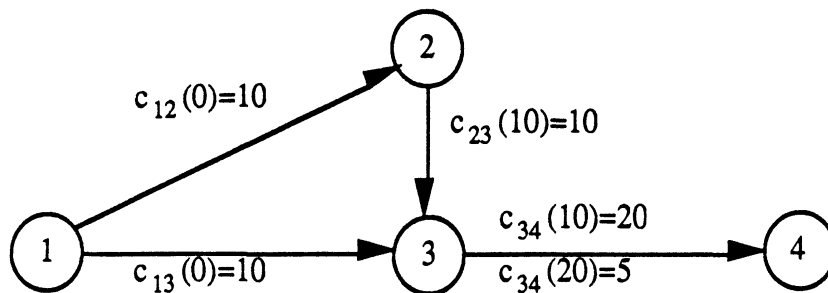


Figure 1: Violation of the Dynamic Principle of Optimality

**Assumption 1** For any  $(i, j) \in \mathcal{A}$ ,  $s + c_{ij}(s) \leq t + c_{ij}(t)$  for all  $s, t \in T$  such that  $s \leq t$ .

In Figure 1,  $c_{34}(\cdot)$  violates Assumption 1. If we arrive at node 3 at time 10, we travel link (3,4) in 20 minutes, arriving at time 30. By arriving 10 minutes later, we can travel the link in only 5 minutes, arriving at time 25. In the context of vehicle routing, this requires us to believe that if a driver leaves node 3 at time 10 and another driver leaves node 3 ten minutes later, the second driver necessarily *passes* the first. However, both of these drivers represent the single driver whose optimal path is being sought, and so violation of the assumption seems to require the driver's behavioral characteristics to be a function of time. In the absence of such an unlikely effect, Assumption 1 does not appear to be overly restrictive.

There can be situations within the traffic environment, however, in which the assumption is violated due to non-behavioral causes. In a surface street area, stoplights may be coordinated so that traffic moving in a given direction without turning will hit a string of green lights, avoiding stop-and-go conditions. Suppose one car is stopped at the first of such a string of lights waiting for the light to change from red to green, and a second car reaches the intersection just after the change occurs. The second car will not have to stop, and will pass the first car easily, reaching the next light (*i.e.*, the end of the next link) earlier even though he reached the beginning of the link later [13]. This effect may be small because it will be difficult to predict at what part of the stoplight cycle a driver will reach an intersection.

It should also be noted that modelling travel times  $c_{ij}(t)$  by a standard impedance function (determining travel time strictly as a function  $f$  of the number of vehicles  $v_{ij}(t)$  occupying link  $(i, j)$  at time  $t$ ) may contradict Assumption 1. Suppose that at time  $t$ ,  $N$  vehicles on link  $(i, j)$  are joined by one more; the link travel time for that vehicle is  $f(N + 1)$ . Then if the  $N$  vehicles depart during period  $t$ , the link travel time for a vehicle joining the link at time  $t + 1$  is  $f(2)$ . There is no guarantee that  $1 + f(2) \geq f(N + 1)$ , and hence Assumption 1 may be violated [10].

When Assumption 1 does hold, we may prove the desired principle of optimality by working with the expanded static network  $(\mathcal{N}', \mathcal{A}')$  defined by (1).

**Lemma 1** *Under Assumption 1, for all  $j \in \mathcal{N}$  there exists a finite optimal path  $(1, 0), \dots, (i, s^*), (j, t^*)$  in the expanded static network from node 1 at time 0 to node  $j$  at a minimal time  $t^*$ , such that the truncation  $(1, 0), \dots, (i, s^*)$  is an optimal path in the static network from node 1 at time 0 to node  $i$  at a minimal time  $s^*$ .*

**Proof:** Consider any finite optimal path  $P' = (1, 0), \dots, (i, s'), (j, t')$  in the static network. (Under Assumption 1, such a path must exist in the absence of negative cycles, since any infinite path will have its trip time reduced by removing cycles and this will reduce the arrival time at each succeeding node.) Consider an optimal path  $P_i^* = (1, 0), \dots, (i, s^*)$  from node 1 to node  $i$ . (Unlike the notation in the statement of the theorem,  $P'$  and  $P_i^*$  are not assumed to have the same intermediate nodes.) Let

$$t^* = s^* + c_{ij}(s^*)$$

The claim is that the path  $P^* = (1, 0), \dots, (i, s^*), (j, t^*)$  travelled by following  $P_i^*$  and then proceeding directly from node  $i$  to node  $j$  is an optimal path from node 1 at time 0 to node  $j$ .

Because  $P_i^*$  is optimal,  $s^* \leq s'$ . Then

$$\begin{aligned} t^* &= s^* + c_{ij}(s^*) \\ &\leq s' + c_{ij}(s') && \text{by Assumption 1} \\ &= t' \end{aligned}$$

By assumption,  $t'$  is a minimal time to reach node  $j$ . Therefore,  $t^*$  is also minimal, and hence  $P^*$  is optimal. ■

**Theorem 2** *(Principle of Optimality for Dynamic Networks) Under Assumption 1, for all  $j \in \mathcal{N}$  there exists a finite optimal path  $(1, 0), \dots, (j, t^*)$  in the expanded static network from node 1 at time 0 to node  $j$  at a minimal time  $t^*$ , such that for each node  $i$  lying on this path, the truncation  $(1, 0), \dots, (i, s^*)$  is an optimal path in the static network from node 1 at time 0 to node  $i$  at a minimal time  $s^*$ .*

**Proof:** By Lemma 1, there exists a finite path  $P^*$  reaching node  $j$  at a minimal time  $t^*$  such that the truncation of this path by one link reaches  $k_1$ , the node preceding  $j$  on  $P^*$ , at an optimal time  $u_1^*$ . Again applying Lemma 1, we find that the further truncation by one link reaches  $k_2$ , the node preceding  $k_1$  on  $P^*$ , at an optimal time  $u_2^*$ . In the absence of negative cycles,  $P^*$  is finite, and hence we may apply Lemma 1 recursively until we find that the truncation  $(1, 0), \dots, (i, s^*)$  is an optimal path to node  $i$ . ■



Therefore, under Assumption 1, it is optimal to reach each intermediate node of a path as early as possible, and (2) is valid. Note how the optimal value function in (2) serves the function of the state variable  $t$  in (1), thus allowing for a more computationally efficient solution. In fact, we shall see that when  $c_{ij}(\cdot) \geq 0$ , as is typical for transportation networks, *any* algorithm which suffices to solve a deterministic shortest path problem will solve (2).

To see this, consider such algorithms as being divided into two classes, label-setting and label-correcting (also called best-first and list-search, respectively). Label setting algorithms such as Dijkstra's method [4] under nonnegative link travel times, or more generally, the  $A^*$  algorithm under a consistency condition [11], select a label  $v_i$  to make permanent, and guarantee that when this is done, the label is indeed at the optimal value for node  $i$ . In contrast, label-correcting algorithms may require  $c_{ij}(t)$  for multiple values of  $t$ . Each time the expansion of a node  $i$  causes a decrease of a label  $v_j$  (indicating the discovery of a superior path to node  $j$ ), node  $j$  is added to a list of nodes to be expanded, and the algorithm terminates only when the list is empty. We shall refer to the application of any label-setting or label-correcting algorithm  $L$  to dynamic networks, where whenever a node  $i$  is expanded the travel times  $c_{ij}(v_i)$  are used, as the *natural generalization*  $\overline{L}$ .

**Lemma 3** *Under Assumption 1, the natural generalization  $\overline{LS}$  of any algorithm  $LS$  which is label-setting in static networks is label-setting in dynamic networks.*

**Proof:** If we apply algorithm  $LS$  to the expanded static network, then at any stage of the algorithm all of the states  $(i, t_i)$  are under consideration, where  $t_i$  is the trip time from node 1 to node  $i$  on any path whose intermediate nodes have all been expanded previously. Due to the label-setting character of  $LS$  in static networks, we are assured that the value associated with the next state  $(i, t)$  to be expanded is its optimal value. Thus,  $t = f_i$ . However, by Assumption 1, the Dynamic Principle of Optimality applies. Hence  $t$  is minimal over all trip times to node  $i$ , and in particular  $t$  is minimal over all paths to  $i$  whose intermediate nodes have been expanded previously. But by definition,  $v_i$  is simply the minimal  $t_i$  yet discovered. Thus  $t = v_i$ . Therefore, when a node is expanded, its label  $v_i$  is equal to the optimal value  $f_i$ , and hence  $\overline{LS}$  is label-setting. ■

**Theorem 4** *Under Assumption 1, the natural generalization  $\overline{LS}$  of a label-setting shortest-path algorithm  $LS$  solves for the shortest path between any two nodes of the dynamic network optimally, with worst-case performance bounds identical to those for  $LS$  in a static network.*

**Proof:** Since  $\overline{LS}$  remains label-setting, a node  $i$  is expanded only when its current value equals its optimal value, i.e., when  $v_i = f_i$ .  $\overline{LS}$  requires only the link travel time  $c_{ij}(v_i)$  whenever node  $i$  is expanded, so in the entire course of the algorithm, the only link travel time required for link  $(i, j)$  is  $c_{ij}(f_i)$ . Hence there exists an alternate static network, with

topology identical to that of the dynamic network but with fixed link costs  $c_{ij}(f_i)$ , on which  $LS$  proceeds identically as would  $\overline{LS}$  on the dynamic network. Since performance bounds for  $LS$  depend only on network topology, they apply to  $\overline{LS}$  on the dynamic network. ■

**Theorem 5** *Under Assumption 1, the natural generalization  $\overline{LC}$  of a label-correcting shortest-path algorithm  $LC$  solves for the shortest path between any two nodes of the dynamic network optimally, with worst-case performance bounds identical to those for  $LC$  in a static network.*

**Proof:** We will prove a slightly stronger statement: that on an optimal path to any node  $j$ ,  $\overline{LC}$  terminates with  $v_i = f_i$  for each node  $i$  on the optimal path. Suppose the reverse, that  $\overline{LC}$  terminates with  $v_i > f_i$  for some node  $i$  which lies on an optimal path to node  $j$ . Let node  $k$  be the node preceding  $i$  on an optimal path from node 1 to node  $i$ . If  $v_k = f_k$ , then by Lemma 1, expansion of node  $k$  would discover the optimal trip time from node 1 to node  $i$ , setting  $v_i = f_i$ . Thus, node  $k$  was not expanded after  $v_k$  was set to  $f_k$ , contradicting the stopping rule which  $\overline{LC}$  inherits from  $LC$ . Hence  $v_k > f_k$ . We may continue this argument recursively on the number of nodes visited on a path until we find that  $v_1 > f_1$ . But since both of these values must be 0, a contradiction ensues. To see that the performance bounds of  $LC$  apply to  $\overline{LC}$ , note that the algorithms differ only in which link costs they apply at various stages of the algorithm, and performance bounds are independent of these costs. ■

Even when Assumption 1 does not hold, generalized shortest-path algorithms may be used to obtain a feasible path whose trip time provides an upper bound  $T_{max}$  on the optimal trip time from the origin node to the destination node. Recall that such an upper bound is required by the Cooke and Halsey algorithm, and that the performance bound is  $O(N^3 T_{max})$ . The generalized shortest-path algorithms may provide the upper bound more rapidly than Cooke and Halsey's enumerative scheme, and may be as tight or tighter when Assumption 1 is not very strongly violated.

## 4 Stochastic Dynamic Networks

Travel time on a link of the road network is subject to uncertainty for many reasons. These include the chance of an incident occurring to reduce link flow capacity, uncertainty in the volume of traffic which will be on the link, interaction of the drivers on the link, and delays caused by drivers making turns onto links which may themselves be congested, causing

spillback. Modelling travel times as stochastic quantities may therefore significantly improve the quality of the routes obtained.

In the network  $(\mathcal{N}, \mathcal{A})$ , let  $\{C_{ij}(t) : t \in T, (i, j) \in \mathcal{A}\}$  be a collection of random variables giving the travel time on each link  $(i, j)$  at each time  $t \in T$ , with the property that the distribution of  $C_{ij}(t)$  given the condition of being at node  $i$  at time  $t$  is independent of all other events which may have occurred from time 0 to time  $t$ . Then we may model the network with stochastic dynamic programming states  $(i, t)$ , giving location node  $i$  and time  $t$ . We also require

- 1)  $0 \in T$
- 2) With probability 1,  $t + C_{ij}(t) \in T$  for all  $t \in T, (i, j) \in \mathcal{A}$ .

Constructing the corresponding stochastic expanded static network, we define the optimal value function  $f_i(t)$  to be the minimum expected trip time over all paths from node  $i$  to node  $N$  departing node  $i$  at time  $t$ . Then we have the functional equation

$$f_i(t) = \begin{cases} \min_{j \neq i} \{E[C_{ij}(t)] + E[f_j(t + C_{ij}(t))]\} & \text{if } i = 1, \dots, N - 1; t \in T \\ 0 & \text{if } i = N; t \in T. \end{cases} \quad (3)$$

Then the problem is again to find  $f_1(0)$ . As before, we can make an assumption which will allow us to compress the state space of the static network. For any path  $(1 = i_0, i_1, i_2, \dots, i_m = i)$  to node  $i$ , let the *random arrival times* for each node be given recursively by

$$A_{i_{k+1}} = A_{i_k} + C_{i_k i_{k+1}}(A_{i_k}) \quad \text{for } k = 0, \dots, m - 1$$

with  $A_1 = 0$ .

**Assumption 2**  $E[U + C_{ij}(U)] \leq E[V + C_{ij}(V)]$  whenever  $E[U] \leq E[V]$  for all random variables  $U$  and  $V$  which are arrival time variables for node  $i$  on any paths to node  $i$ .

**Theorem 6** (*Principle of Optimality for Stochastic Dynamic Networks*). Under Assumption 2, for all  $j \in \mathcal{N}$  there exists a finite optimal path  $(1, 0), \dots, (j, T^*)$  in the stochastic expanded static network from node 1 at time 0 to node  $j$  at a random time  $T^*$  with minimal expectation, such that for each node  $i$  lying on this path, the truncation  $(1, 0), \dots, (i, S^*)$  is an optimal path in the static network from node 1 at time 0 to node  $i$  at a random time  $S^*$  with minimal expectation.

**Proof:** Follows the proofs of Lemma 1 and Theorem 2. ■

As before, we can replace the backward recursion of (3) by a forward recursion with a reduced state space. Define  $f_i$  to be the minimum expected trip time over all paths from node 1 to node  $i$  departing node 1 at time 0. The functional equation is

$$f_j = \begin{cases} \min_{i \neq j} \{f_i + E[C_{ij}(f_i)]\} & \text{for } j = 2, \dots, N \\ 0 & \text{for } j = 1 \end{cases} \quad (4)$$

Because  $C_{ij}(t)$  given location node  $i$  at time  $t$  is independent of all other prior events,  $E[C_{ij}(f_i)]$  is a deterministic function of  $f_i$ , and hence this functional equation has structure identical to that of the deterministic functional equation (2), so *we may apply identical solution procedures.*

Therefore, when Assumption 2 holds, we may find the shortest path with a single generalized application of a standard shortest-path algorithm, saving greatly over Hall's procedure [6], which may require many  $k$ th shortest path solutions. However, Assumption 2 is more restrictive than its deterministic counterpart, and when the condition holds, it will be difficult to verify. Hall's counterexample shows that a simple network operating in a reasonable fashion can violate Assumption 2 and hence prevent optimal solution by static shortest-path algorithms.

Even when Assumption 2 does not hold, applying static methods based on functional equations (2) and (4) will improve the algorithm by tightening both the upper and lower bounds, and hence terminating as early or earlier. Hall's algorithm proceeds by identifying a candidate path in each iteration, and maintaining upper and lower bounds on the optimal expected trip time from these candidate paths. The  $k$ th candidate path is the  $k$ th shortest path in an alternate network with identical topology, where the static link travel time  $d_{ij}$  is the minimum possible travel time on link  $(i, j)$  at any time  $t$ . Hence the trip time on the  $k$ th candidate path provides a lower bound on the expected trip time of all paths not yet considered, and the actual expected trip time on this path provides an upper bound. The algorithm terminates when the bounds meet.

Therefore, even in the general case we can improve Hall's algorithm in two ways. First, we may begin by applying a generalized stochastic shortest path algorithm, which will rapidly provide an initial solution which may serve as a good upper bound. Second, we may improve the lower bounds in each iteration by choosing the  $k$ th candidate path as the  $k$ th shortest *time-dependent* path. That is, if we construct the alternate network with varying link travel times  $d_{ij}(t)$ , the minimum possible travel time on link  $(i, j)$  departing node  $i$  at time  $t$ , then we have a deterministic dynamic network in which Assumption 1 is likely to hold. Hence the Principle of Optimality allows us to apply standard  $k$ th-shortest path methods, with no additional computational effort. Furthermore, the trip time on any path in the revised dynamic network is no less than the trip time on the same path in the revised static network, hence the lower bounds of Hall's algorithm are tightened. Note that Assumption 1 is satisfied

in the revised network of Hall’s bus network example.

## 5 Computational Results: Anticipatory Route Choice in INTEGRATION

The benefits available to individual drivers using a dynamic network model have been investigated through the INTEGRATION traffic simulator [14], developed by M. Van Aerde at Queens University, Kingston, Ontario, Canada. INTEGRATION models traffic microscopically, maintaining location and destination information for each vehicle in the simulated network. The main issues addressed by INTEGRATION are first, interaction of freeway corridors and signalized networks, and second, opportunities for optimal route choice by individual drivers based upon real-time information. The latter makes INTEGRATION well-suited to examine the potential benefits of a dynamic network model.

INTEGRATION maintains one category of vehicles which may be considered background traffic; these drivers follow a fixed route from their origin to their destination regardless of real-time network conditions. Vehicles in a second category are provided with an updated “optimal” routing each time they reach nodes, hence having decision opportunities. This optimal routing is calculated by the application of a static shortest-path algorithm to a network model with fixed travel times determined from the latest real-time measured travel times within the simulation. These travel times are determined by the difference between link entry and departure times for the vehicles which have most recently traveled the link, and hence include not only the time required to travel the link distance at some projected speed but also the delay incurred due to stoplights, incidents and lane blockages, and queueing caused by congestion. However, this “dynamic” routing optimization depends on a static network model with fixed link travel times and hence can be considered *quasi-dynamic*.

We have modified INTEGRATION to add a third category, which performs vehicle-dependent routing optimization on a dynamic network model. The second class of drivers may be routed to links which are currently uncongested, only to discover that the links have become congested by the time the drivers reach that link. In contrast, drivers in the third category look ahead in time to avoid links which will become congested. We describe this dynamic-network route choice as *anticipatory* or *fully dynamic*.

We tested the value of anticipatory routing on the simple network pictured in Figure 2. The construction of the network is described in detail in Appendix A. All traffic enters the network at node 1 with destination node 2. The upper path is a series of freeway links and the lower path consists of arterial links with lower speed limit and less capacity, hence greater sensitivity of link travel time to traffic volume. The simulation begins with the network empty, and vehicles are added to the network as time passes. Most of these vehicles have no real-time guidance capability, and they all choose the freeway path, causing it to become

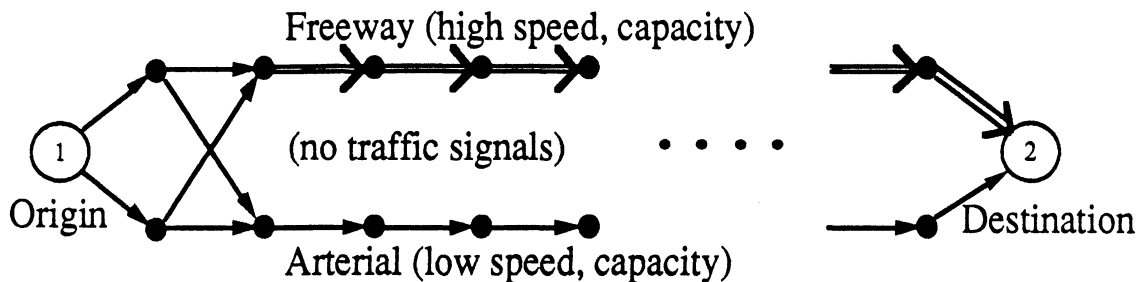


Figure 2: Test Network for Anticipatory Routing in INTEGRATION.

Time in minutes	Route
0-20	Freeway
21-70	Arterial
71-120	Freeway

Table 1: Routing policies for quasi-dynamic vehicles as a function of time of entry to network.

congested. When this congestion has accumulated to the degree that static shortest-path calculations for latest measured travel times give a faster time for the arterial route, the quasi-dynamic (second category) vehicles entering the network choose the arterial route. Later, when no more background traffic is entering the network, the freeway congestion clears, the measured travel times on the freeway decrease, and quasi-dynamic vehicles entering in this later period will now choose the freeway route. The route choice of quasi-dynamic vehicles as a function of time is given in Table 1.

However, the quasi-dynamic vehicles entering the network do not begin to route themselves away from the freeway until after the freeway has become congested. In contrast, anticipatory drivers will route themselves to the arterial when they foresee freeway congestion not yet visible which would impede them. Similarly, later in the simulation they anticipate the clearing of the congestion instead of waiting for it to happen. The route choice resulting from application of fully dynamic shortest path calculations to the travel times resulting from the simulation are shown in Table 2. Note that as expected, the fully dynamic traffic reacts to changing conditions earlier than does the quasi-dynamic traffic.

Time in minutes	Route
0-6	Freeway
7-60	Arterial
61-120	Freeway

Table 2: Routing policies for fully-dynamic vehicles as a function of time of entry to network.

Time in minutes	Routes	Agree/Disagree
0-6	Freeway	Agree
7-20	QD→Fwy, FD→Art	Disagree
21-60	Arterial	Agree
61-70	QD→Art, FD→Fwy	Disagree
71-120	Freeway	Agree

Table 3: Comparison of quasi- and fully dynamic routing policies as a function of time of entry to network.

Vehicle Type	Trip Time (min)
Background	36.2
Quasi-dynamic	30.4
Fully dynamic	27.6

Table 4: Comparison of average trip times for vehicles entering the network during periods of disagreement.

Hence, during two distinct periods of the simulation, the quasi- and fully dynamic routing policies disagree; the comparison is made more clearly in Table 3.

To determine which of these policies is superior, we execute the simulation a second time, where fully dynamic vehicles are present and make routing decisions based on the application of dynamic shortest path calculations to the travel time data measured in the first simulation run. We only allow quasi- and fully dynamic vehicles to enter the network during the periods of disagreement given in Table 3, since we know that during periods of agreement, the two categories of vehicles will have the same performance. In Table 4, we compare the average trip times in periods of disagreement for each class of vehicle. As expected, the quasi-dynamic vehicles have average trip time superior to that for the background traffic. But the anticipatory vehicles have an average trip time 9% less than the that for the quasi-dynamic traffic, because anticipatory drivers divert from the freeway only when it would be slower than the secondary route *during the trip*, not at the time the decision is made. Although the data was not real world driven, the results suggest the potential in principle for significant trip time savings by implementing anticipatory routing.

This testing procedure requires that the proportion of anticipatory drivers be kept very small in the second simulation, so that the link travel times observed in the first simulation may be considered accurate forecasts of future link travel times during the second simulation. If there are many anticipatory drivers, their improved route choice will cause the network to be loaded differently over time, resulting in different observed link travel times, and invalidating the route choice of the anticipatory drivers. Operating an IVHS system which

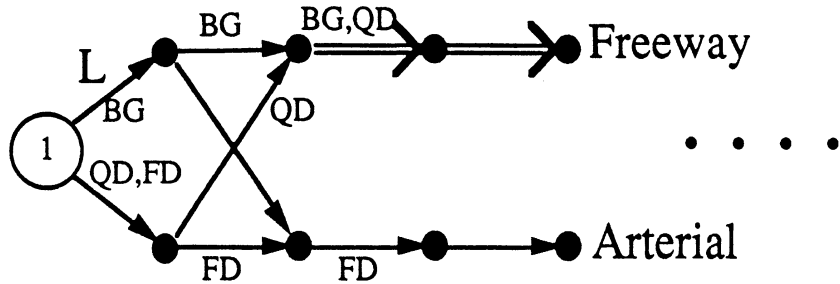


Figure 3: Route selection by vehicle class during first period of disagreement.

disseminates short-term travel time forecasts to large numbers of anticipatory drivers in a physical road network will require substantial research on the subject of determining the future of a traffic network based on real-time traffic conditions, incident modelling, stochastic driver behavior and route choice, and time-varying travel demands [8].

### The importance of anticipatory routing

To gain understanding beyond mere numerical results, let us examine the question of why the quasi-dynamic traffic entering the network had an average trip time less than that for the background traffic. After all, the quasi-dynamic traffic makes the wrong choice during both periods of disagreement, while the background traffic always follows the freeway path, coinciding with the fully dynamic traffic during the second period of disagreement. Since the background traffic is right once and the quasi-dynamic traffic is wrong both times, we should expect the background traffic to have a better average trip time than the quasi-dynamic traffic.

The discrepancy is explained by a feature of the network we have not previously discussed, namely that the decision between freeway and arterial is not really made until each vehicle chooses the *second* link on its path, due to the links crossing between the arterial and freeway paths. In Figure 3 we show the routings of the various classes of traffic during the first period of disagreement. The manner in which the network is loaded over time causes considerable congestion on the link labelled *L* in the figure. The quasi-dynamic vehicles are able to reduce their trip times by avoiding link *L*, even while following the incorrect long-range policy of selecting the freeway.

The hypothesis suggested by this analysis is that quasi-dynamic routing performs well when decision opportunities are frequent and decisions which may be rendered suboptimal in hindsight by updated link travel time data are easily reversed. However, in an environment of infrequent decision opportunities, with substantial costs incurred by any single poor decision, anticipatory routing offers significant benefits. This is affected not only by network topology but also by how frequently link travel times are updated, since without



new information, route optimization will not cause a change in individual route choice. In future research, we will investigate this hypothesis in more detailed and realistic networks, with higher proportions of fully dynamic traffic and the presence of incidents, i.e., capacity reductions causing non-recurring congestion.

## **Acknowledgement**

We are grateful to Michel Van Aerde for providing us with a copy of the INTEGRATION software for the purpose of testing anticipatory routing.

## References

- [1] Bellman, R., *Dynamic Programming*, Princeton University Press, Princeton, N.J. (1957).
- [2] Cooke, K. L. and E. Halsey, "The Shortest Route Through a Network with Time-Dependent Internodal Transit Times", *Journal of Mathematical Analysis and Applications* **14**, 493-498 (1966).
- [3] Denardo, E. V., *Dynamic Programming: Models and Applications*, Prentice-Hall, Inc., Englewood Cliffs, N.J. (1982).
- [4] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* **1**, 269-271 (1959).
- [5] Dreyfus, S. E., "An Appraisal of Some Shortest-Path Algorithms", *Operations Research* **17**, 395-412 (1969).
- [6] Hall, R. W., "The Fastest Path through a Network with Random Time-Dependent Travel Times," *Transportation Science* **20**, 182-188 (1987).
- [7] Halpern, J., "Shortest Route with Time Dependent Length of Edges and Limited Delay Possibilities in Nodes", *Zeitschrift für Operations Research* **21** 117-124 (1977).
- [8] Kaufman, D. E., Lee, J., and Smith, R. L., "Anticipatory Traffic Modelling and Route Guidance in Intelligent Vehicle/Highway Systems", Technical Report 90-2, Intelligent Vehicle/Highway Systems Program, The University of Michigan, Ann Arbor, MI (1990).
- [9] Orda, A. and Rom, R., "Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length", *Journal of the Association for Computing Machinery* **37** (3), 607-625 (1990).
- [10] Lafortune, S., Private communication, *IVHS Program, University of Michigan* (1990).
- [11] Pearl, J., *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Publishing Company, Reading, Mass. (1984).
- [12] Prakash, A., Private communication, *IVHS Program, University of Michigan* (1989).
- [13] Underwood, S., Private communication, *IVHS Program, University of Michigan* (1990).
- [14] Van Aerde, M., and Yagar, S., "Dynamic Integrated Freeway/Traffic Networks: A Routing-Based Modelling Approach", *Transportation Research A* **22A** (6), 445-453 (1988).

## Appendix A: Data for Test Network in INTEGRATION

In this appendix we describe the detailed construction of the network, shown in Figure 2 previously, which was used to evaluate the anticipatory routing strategy in the INTEGRATION traffic simulator.

The simulation duration was 7200 seconds, and quasi-dynamic and fully-dynamic minimum travel time path trees were recalculated after each 60 seconds of simulation time. The network consisted of 57 nodes, of which nodes 1, 2, and 3 were zones, i.e., origin/destination points. (Node 3 is not shown in Figure 2; it will be discussed shortly.) No traffic signals are present in the network.

The links present in the network are of three types: freeway, arterial, and intermediate. Table 5 gives the characteristics of these link types. Each link in the network is unidirectional, as shown in the figure. The intermediate links are the four links which may be chosen as the second link on a path from node 1. All other links which are on the upper path are of the freeway type, while the remaining links on the lower path are arterial. In addition, node 3 has as its only connection to the network a link from node 2 with zero travel time. This was done because INTEGRATION provides statistics such as average trip time broken down only by origin/destination pair and not by time period. In order to measure average trip time for vehicles departing during the periods of policy disagreement given in Table 3, those vehicles have destination node 3 instead of 2. This does not affect their trip time, since node 3 can only be reached by going to node 2 and then traversing the zero travel-time link. (To be precise, this link is 10 meters long and has 7 lanes and a maximum speed of 1000 kilometers/hour.) The number of links on each path from node 1 to node 2 is 28.

The travel demands placed on the network are given in Table 6. None of the vehicles entering the network have random headways; that is, the times of vehicles entering the network are spaced uniformly over each interval given in the table. All traffic with node 2 as destination is background traffic; traffic to node 3 is 2% quasi-dynamic and 2% fully

Link type	Length (km)	Maximum speed (km/hour)	# lanes	Saturation flow per lane (veh/hour)
Freeway	1 <sup>a</sup>	90	2	2000
Intermediate	2	80	2	1700
Arterial	1 <sup>b</sup>	65	1	1400

<sup>a</sup>except links out of node 1 and into node 2, which have length 2 km.

<sup>b</sup>except links out of node 1 and into node 2, which have length 2 km.

Table 5: Link types

Start (sec)	End(sec)	Orig. node	Dest. node	Veh. per hour
0	300	1	2	4000
300	900	1	3	4000
900	1150	1	3	6000
1150	1800	1	2	6000
1800	2700	1	2	1000
3600	4200	1	3	225

Table 6: Travel demands over simulation time.

dynamic except for the flow beginning at 3600 seconds, which is one-third background, one-third quasi-dynamic, and one-third fully dynamic.

We made one further change to INTEGRATION. The link travel times provided to the static and dynamic route optimization modules are computed as rolling averages. That is, the optimizations use travel times  $c_{ij}$ , which are updated with the most recently experienced link travel time  $c$  on link  $(i, j)$  by the formula  $c_{ij}^{new} = \alpha c + (1 - \alpha)c_{ij}^{old}$ , for some  $0 < \alpha < 1$ . In order to have the forecasts  $c_{ij}(t)$  for anticipatory routing be more up-to-date, we used  $\alpha = 0.4$  instead of the standard setting  $\alpha = 0.05$ .