Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

The Fastest Path through a Network with Random Time-Dependent Travel Times

Randolph W. Hall,

The Fastest Path through a Network with Random Time-Dependent Travel Times

RANDOLPH W. HALL

University of California, Berkeley, California

This paper introduces the problem of finding the least expected travel time path between two nodes in a network with travel times that are both random and time-dependent (e.g., a truck, rail, air or bus network). It first shows that standard shortest path algorithms (such as the Dijkstra algorithm) do not find the minimum expected travel time path on such a network, then proposes a method which does find the minimum path. Next, this paper shows that the optimal "route choice" is not a simple path but an adaptive decision rule. The best route from any given node to the final destination depends on the arrival time at that node. Because the arrival time is not known before departing the origin, a better route can be selected by deferring the final choice until later nodes are reached. A method for finding the optimal adaptive decision rule is proposed.

Path finding algorithms are used to find the optimal routes (i.e., "shortest paths") between nodes in many types of networks. They are used in transportation to direct patrons phoning into telephone information lines, route freight shipments and predict the demand over road links. In all of these applications, the objective of computer programs is generally to find the quickest route between specific travel origins and destinations.

Many algorithms exist for finding the least expected travel time path through networks with deterministic time-dependent travel times (e.g., DREYFUS[5]) and for finding the least expected travel time path through networks with random non-time-dependent travel times (e.g., see DIJKSTRA[4]). The question of how to find the least expected travel time path through a network with travel times that are both random and time-dependent has not been addressed in the literature (for example, the problem is absent from the reviews in Dreyfus,[5] LAWLER[12] and PIERCE[15]).

Random time-dependent travel times occur on networks where vehicles are dispatched according to a schedule and incur random delays (e.g., truck, rail, air and bus networks). For example, the travel time for a bus passenger between a boarding stop and destination stop depends on his arrival time at the boarding stop relative to the bus schedule. This travel time is also random, depending on when the bus actually arrives, and traffic congestion, vehicle breakdowns and demand en route.

The purpose of this paper is to show how the problem of finding the least expected travel time path through a network with travel times that are both random and time-dependent might be approached. It first proves that standard shortest path algorithms (such as the Dijkstra algorithm) do not find the least expected travel time path on networks with random time-dependent travel times. Branch-and-bound is proposed for finding the least expected travel time path on this type of network. This approach differs from standard methods in that it utilizes travel time probability functions rather than expected travel times alone.

This paper next describes an alternative to choosing a simple path. Instead of selecting an entire route before departing the origin, the traveler might adapt his route as information is acquired. For networks with random time-dependent travel times, the best route from any given node to the final destination depends on the arrival time at that node. Because the arrival time is not known before departing the origin, a better route can be selected by deferring the final choice until later nodes are reached. With an adaptive decision rule, an optimal successor node is defined for each node *as a function of the arrival time* at the node. Dynamic programming is proposed for finding the optimal time-adaptive decision rule.

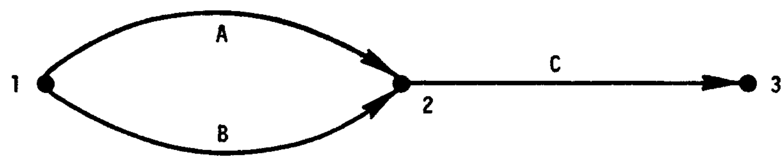
LEAST EXPECTED TRAVEL TIME PATH

TO SEE that standard shortest path algorithms do not identify optimal routes on networks with random time-dependent travel times, consider the example in

Transportation Science
Vol. 20, No. 3, August 1986

0041-1655/86/2003-0182 $01.25

STANDARD PATH : A - C (200 MINUTES)
FASTEST PATH : B - C (170 MINUTES)



TRAVEL TIMES

A	100 MINUTES
B	90 MINUTES (.5 PROBABILITY) 120 MINUTES (.5 PROBABILITY)
C	30 MINUTES, ARRIVAL TIME AT NODE 2 BEFORE 3:35 100 MINUTES, OTHERWISE

Fig. 1. Standard path versus fastest path on network with random time-dependent travel times.

Figure 1. Suppose that a traveler wishes to go from node 1 to node 3 by the minimum expected travel time path and leaves node 1 at 2:00. Arc A has a certain travel time of 100 minutes, while arc B has a random travel time of 90 minutes with 0.5 probability and 120 minutes with 0.5 probability. Now, suppose that travel time over arc C is time-dependent: 30 minutes when the arrival time at node 2 is before 3:35, and 100 minutes otherwise.

Arc A clearly has the lower expected travel time to node 2 (100 minutes as opposed to 105 minutes on B), and would be chosen by a standard shortest path algorithm. Yet it is inferior to arc B when continuing on to node 3. Because arc B has the higher probability of arriving at node 2 before 3:35 minutes, the total expected travel time to node 3 is lower (170 minutes as opposed to 200 minutes).

Standard shortest path algorithms are based on BELLMAN's[2] equations, which assume that the best path to a node k is also best when the path continues on to another node j. The above example proves that Bellman's equations do not apply when travel times are *both* random and time-dependent.

To find the least expected travel time path on a network with random time-dependent travel times, an algorithm must utilize the travel time probability distributions for the links in the network. For example, in transit networks, the algorithm must account for the fact that some lines might give a low expected travel time to a bus stop, yet have poor likelihood of arriving in time for a transfer connection. Other lines with higher expected travel times might be preferred if the likelihood of arriving in time for the transfer connection is better.

The method presented in Table I is one approach for solving this problem. It can be used when arc travel times are boundable from below and calculating the expected travel time on any path is difficult (as is the case for networks like the one in Figure 1). It combines a branch-and-bound technique with an M shortest paths algorithm (that is, an algorithm which finds the shortest path between two nodes, the second shortest, . . . , up to the Mth shortest path; e.g., Dreyfus[5] and MURTY[14]).

TABLE I

Method for Finding Least Expected Travel Time Path

Step		
0	Set $m = 1$, $t_u = \infty$ Find the shortest path from node 1 (origin) to all other nodes, based on minimum possible travel times over arcs $(l_{i,j})$ Call the shortest path from node 1 to node N (destination) P_1	
1	Set $m = m + 1$ Calculate the expected travel time on path P_{m-1}, denoted T_{m-1}	
2	Based on minimum possible travel times $(l_{i,j})$, find the mth shortest path from node 1 to all other nodes Call the mth shortest path from node 1 to node N P_m Set t_l equal to the minimum possible travel time over P_m	
3	If $T_{m-1} < t_u$:	$t_u = T_{m-1}$ $P = P_{m-1}$
	If $t_u \le t_l$:	P is optimal t_u is minimum expected travel time
	If $t_u > t_l$:	Return to Step 1

The method iteratively evaluates the expected travel times of selected paths, and stops when one of the evaluated paths is found to have the minimum expected travel time. At each iteration an upper bound on the expected travel time of the best possible path and a lower bound on the expected travel times of the paths not yet considered are found. For the mth iteration, the upper bound equals the minimum expected travel time of the $m - 1$ paths evaluated so far, t_u. The lower bound equals the minimum possible travel time of the mth path, t_l. The minimum possible travel time over a path equals the sum of the minimum possible travel times of its arcs (denoted by $l_{i,j}$ for arc (i, j)), possibly the scheduled vehicle travel time with zero waiting time).

At any iteration, the expected travel time of all paths not yet evaluated must be greater than t_l. If it turns out that $t_l > t_u$, then the minimum expected travel time of all paths not yet evaluated must be greater than t_u. Since t_u is both the minimum of the first $m - 1$ paths, and less than the expected travel time of all other possible paths, it must then be the overall minimum.

It should be clear from the above description that finding the minimum expected travel time path on a network with random time-dependent travel times is considerably more complicated than solving the simple shortest path problem. Whether or not the added computational effort in implementing the above method (or any alternative) is justified depends on the application.

Demonstration and Network Construction

A transit network is used as an example to illustrate how the method can be applied. Most of the previous research on path choice on transit networks has been directed toward traffic assignment (e.g., Dial,[3] Lamb and Havers,[10] Last and Leek,[11] Le Clerq,[13] and Spiess[16]). Because of their intended application, these papers did not account for the time dependency of travel time on transit networks. That is, instead of incorporating a detailed bus schedule, waiting times were estimated from vehicle headways. A paper by Gilsinn et al.[6] was directed toward the application of providing traveler directions, and did incorporate actual bus schedules. However, the paper did not account for travel time variability.

An important issue in all of these papers is how to treat transfers between bus lines. Suppose that arcs (1, 3) and (3, 4) represent two segments of one bus line, and arc (2, 3) represents one segment of another bus line in a hypothetical network. If arc (3, 4) is traversed after arc (1, 3), then the traveler incurs no waiting time. However, if arc (3, 4) is traversed after (2, 3), then the traveler must wait for a transfer. Therefore, the arc can have two different travel times, depending on how the arc is reached.

Previous authors have devised special purpose algorithms to deal with this issue, and other issues that are specific to transit networks. Although somewhat less efficient than these special purpose algorithms, more general algorithms can also be used on

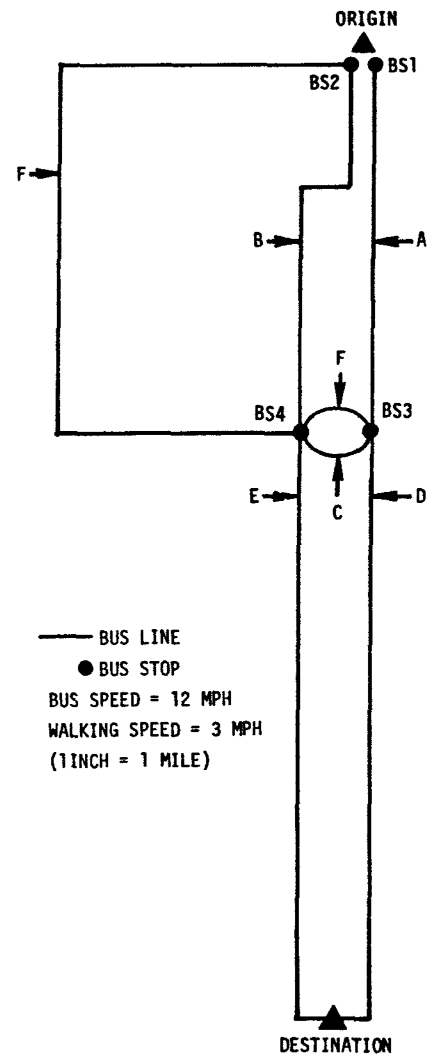


Fig. 2. Bus lines.

transit, provided that the transit network is suitably reconstructed. Figure 2 shows a conventional representation of a transit network and Figure 3 shows the same network in reconstructed form. Figure 3 is constructed so that there is no more than one arc between any pair of nodes. Each node represents the origin, destination or a place where a specific bus line can be boarded (if two bus lines can be boarded at the same place, there are two nodes, one for each line). The travel time on each arc accounts for waiting and riding time for the corresponding line (the method used to construct this network is described in detail in Hall[7]).

Although Figure 3 represents a transit system, the network is in fact general. The network is defined by a set of nodes and a set of arcs connecting nodes. And no more than one arc connects each pair of nodes. Figure 3 also has the properties of randomness and time dependency. Travel times are time-dependent

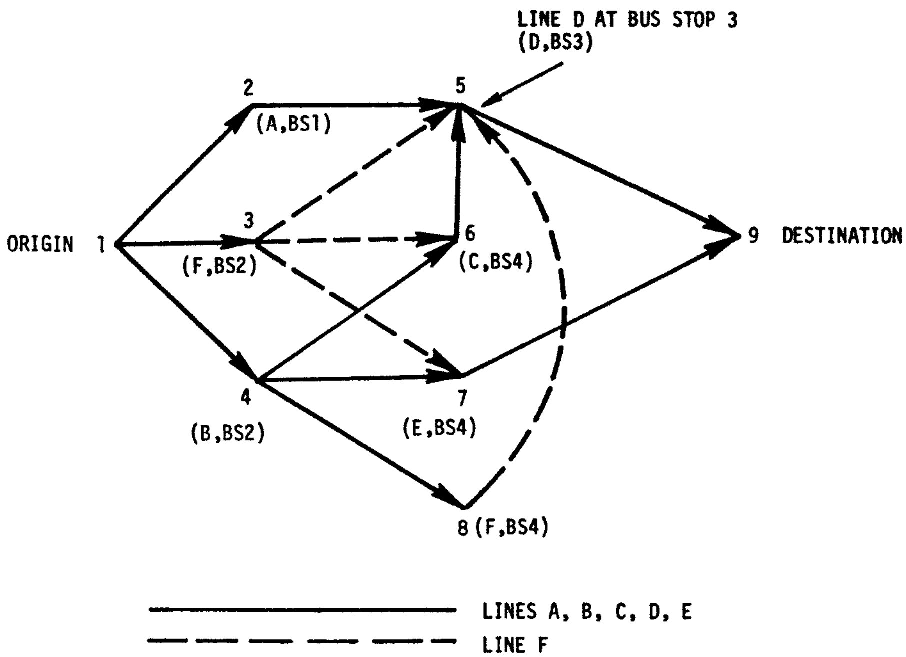


Fig. 3. Bus network in graph form.

TABLE II
Travel Time Distributions

Line A	$\mu_2 + T_{2,5}$	$\sim$	EXP (15, 5)	$0 \le \mu_2 \le 5$
			EXP (45, 5)	$5 \le \mu_2 \le 35$
			EXP (75, 5)	$35 \le \mu_2 \le 65$
Line B	$\mu_4 + T_{4,6}$	$\sim$	EXP (15, 5)	$0 \le \mu_4 \le 3$
	$\mu_4 + T_{4,7}$		EXP (45, 5)	$3 \le \mu_4 \le 33$
	$\mu_4 + T_{4,8}$		EXP (75, 5)	$33 \le \mu_4 \le 63$
Line C	$\mu_6 + T_{6,5}$	$\sim$	EXP $(3 + \mu_6, 1)$	$\forall \mu_6$
Line D	$\mu_5 + T_{5,9}$	$\sim$	EXP (45, 5)	$0 \le \mu_5 \le 25$
			EXP (75, 5)	$25 \le \mu_5 \le 55$
			EXP (105, 5)	$55 \le \mu_5 \le 85$
Line E	$\mu_7 + T_{7,9}$	$\sim$	EXP (50, 5)	$0 \le \mu_7 \le 30$
			EXP (80, 5)	$30 \le \mu_7 \le 60$
			EXP (110, 5)	$60 \le \mu_7 \le 90$
Line F	$\mu_3 + T_{3,5}$		EXP (120, 5)	$0 \le \mu_3 \le 80$
	$\mu_3 + T_{3,6}$		EXP (115, 5)	$0 \le \mu_3 \le 80$
	$\mu_3 + T_{3,7}$		EXP (115, 5)	$0 \le \mu_3 \le 80$
	$\mu_8 + T_{8,5}$		EXP(120, 5)	$0 \le \mu_8 \le 115$
Walking	$T_{1,2}$	$= 5$		
	$T_{1,3}$	$= 3$		
	$T_{1,4}$	$= 3$		

$$\text{EXP}(M, \lambda) = \begin{cases} 1/\lambda \, e^{-(t-M)/\lambda}, & t \ge M \\ 0, & \text{elsewhere} \end{cases}$$

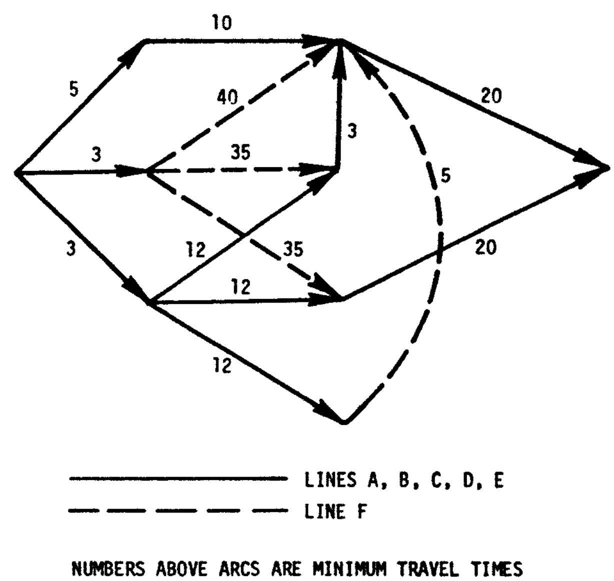


Fig. 4. Minimum travel time across arcs.

because waiting time varies with the traveler's arrival time at bus stops. Travel times are also random because the bus may not arrive on time, or because there may be delay en route. Therefore, transit systems can be represented by general networks with travel times that are both random and time-dependent.

The travel time probability distributions for the arcs in Figure 3 are provided in Table II. The symbol μ_i represents the traveler's arrival time at node i, and $T_{i,j}$ represents the actual (random) travel time over arc (i, j). These data reflect a situation where buses on lines A, B, D, E and F leave on schedule every 30 minutes, but are randomly delayed before arriving at the destination according to an exponential distribution. Travel time on line C (arc (6, 5)) does not depend on arrival time (the line does not follow a schedule) and walking times (arcs (1, 2), (1, 3) and (1, 4)) are deterministic. The minimum possible travel times ($l_{i,j}$) are provided in Figure 4.

The method is executed in Table III for a departure

TABLE III

Execution of Method for Finding Least Expected Travel Time Path

m	P_m	$T(P_m)$	t_u	t_l	P
1	(1, 2, 5, 9)	54.1	∞	35	—
2	(1, 4, 7, 9)	56.5	54.1	35	(1, 2, 5, 9)
3	(1, 4, 6, 5, 9)	54.3	54.1	38	(1, 2, 5, 9)
4	(1, 4, 8, 5, 9)	125.0	54.1	40	(1, 2, 5, 9)
5	(1, 3, 7, 9)	—	54.1	58	(1, 2, 5, 9)

$T^* = 54.1$

$P^* = (1, 2, 5, 9)$

time of $\mu_0 = 0$. The path (1, 2, 5, 9) is optimal, and the expected travel time equals 54.1 minutes. Referring to Figure 3, the optimal route begins by walking to Bus Stop 1 (BS1) and boarding line A. At BS3, the traveler exits and transfers to line D. Depending on the traveler's arrival time at BS3, he might either board the first scheduled bus on line D (departing at time 25), or the second scheduled bus on line D (departing at time 55).

TIME-ADAPTIVE ROUTE CHOICE

EACH TIME a node is reached in a network, the traveler acquires a new piece of information—the arrival time at the node. On a network with random time-dependent travel times, this information can be used in selecting the next arc in the path. Because this information is acquired after leaving the origin, expected travel time can be reduced over the best simple path (i.e., selecting the entire path before leaving the origin) by adapting the route at each node.

To see that the optimal "route choice" for a network with random time-dependent travel times is not a simple path but an adaptive decision rule, consider the network of Figure 5. Suppose that a traveler wishes to go from node 1 to node 3, leaving at 2:00. Let the travel time on arc A be 30 minutes with 0.5 probability and 45 minutes with 0.5 probability. Now suppose that the vehicle on arc B takes 15 minutes to reach the destination and departs from node 2 at 2:35 and 3:05. Suppose that the vehicle on arc C also takes 15 minutes to reach the destination, but departs from node 2 at 2:50 and 3:20.

Clearly, arc B is optimal if travel time on A is 30 minutes, and arc C is optimal if travel time on A is 45 minutes. Thus the traveler would be better off to choose between arcs B and C after reaching node 2 (and noting arrival time) than to choose his entire path before leaving the origin. This adaptive decision rule has an expected travel time of 57.5 minutes, 7.5 minutes less than either path A–B or A–C by itself. Thus, travel time can be reduced over the best simple path by choosing routes adaptively.

On a general network, the traveler can potentially

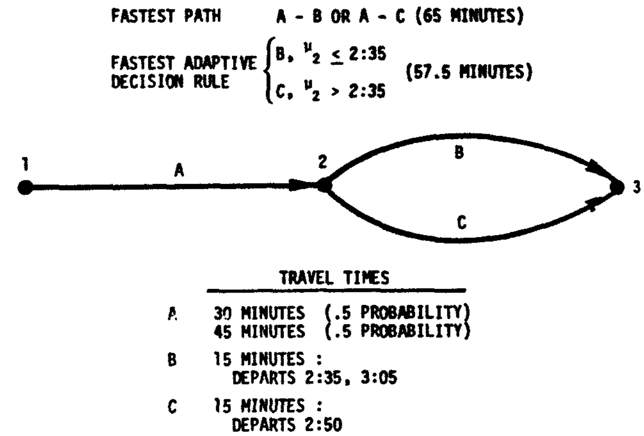


Fig. 5. Fastest path versus adaptive decision rule.

change his route at any node. The traveler can note his arrival time upon arriving at the node, and respond to this learned information by selecting the best possible arc to continue his path. This process is called time-adaptive route choice. For a transit network, time-adapative route choice could mean that the traveler selects the next bus in his path upon exiting his last bus and noting his arrival time.

Dynamic programming (BELLMAN[1] and HOWARD[9]) can be applied to the problem of finding the optimal time-adaptive decision rule. However, applying dynamic programming requires that the decision problem be divided into a finite number of stages. The route choice problem can be divided into K stages by limiting the number of arcs in any path to K. While this approach is not guaranteed to identify the optimal solution, with K sufficiently large the solution should be very close to the optimum.

One way to construct a graph for time-adaptive route choice is to let each node represent a point where the traveler can change from one vehicle to another. For example, in transit networks, a node would represent a bus stop, and the number of stages in the dynamic program (K) would equal the maximum number of buses used in the trip. Each of the K stages would then correspond to a choice of bus line and bus stop destination: at the first stage (numbered $k = 0$), the traveler chooses the first bus (and destination) in his trip; at stage $k = 1$ the traveler chooses the second bus; and at stage $k = K - 1$ the traveler chooses the last bus. Because very few people are willing to transfer more than twice, a value of $K = 3$ is reasonable for transit networks. Different values may be appropriate for other types of networks (freight, air, etc.).

A dynamic program is defined by a boundary condition and recurrence equation. Let $T_k^*(n, \mu_n)$ equal the minimum expected arrival time at the destination,

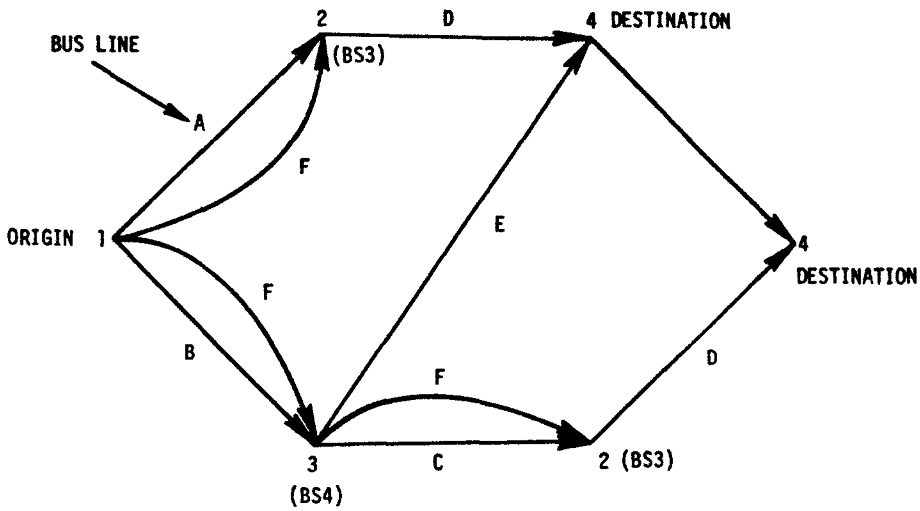


Fig. 6. Graph for time-adaptive route choice.

given that the traveler's arrival time at node n is μ_n, and k arcs have been used so far. Then the boundary condition for the destination node N is:

$$T_k^*(N, \mu_N) = \mu_N, \qquad \forall \mu_N.$$

Let

$f_{ij}(t \mid \mu_i)$ = the probability density function for the arrival time at node j given that the traveler arrived at node i at time μ_i.

The recurrence equation is defined by integrating the probability density function multiplied by the associated objective value for each successor node j, and selecting the minimum:

$$T_k^*(i, \mu_i) = \min_{j \in N(i)} \int f_{ij}(t \mid \mu_i) T_{k+1}^*(j, t)\, dt,$$

where $N(i)$ is the set of successor nodes accessible from node i. Therefore, this process finds the best arc to continue the path as a function of the arrival time at node i (μ_i).

To find the optimal decision rule, calculations can be performed iteratively beginning from the final stage K and ending at the initial stage 0. The problem is solved when $T_k^*(1, \mu_0)$ is evaluated, where μ_0 is the departure time and node 1 is the origin.

The above method is easily generalized to allow more than two arcs between any pair of nodes. Figure 6 shows one way to represent the bus network in Figure 2 in graph form. To execute the dynamic program, the data in Table II can be translated to match the graph in Figure 6. Table IV summarizes the calculations for the dynamic program. The resulting expected travel time is 53.45 minutes, 0.65 minute less than the minimum expected travel time of any simple path.

TABLE IV

Derivation of Time Adaptive Decision Rule

Stage (k)	Node (i)	μ_i	$T_k^*(i, \mu_i)$	Successor Node and Line
3	4	μ_4	μ_4	—
2	4	μ_4	μ_4	—
	2	$0 \le \mu_2 \le 25$	50	Node 4, Line D
		$25 < \mu_2 \le 55$	80	Node 4, Line D
		$55 < \mu_2 \le 85$	110	Node 4, Line D
1	3	$0 \le \mu_3 \le 20.2$	$50 + 30e^{-(22-\mu_3)}$	Node 2, Line C
		$20.2 < \mu_3 \le 30$	55	Node 4, Line E
		$30 < \mu_3 \le 50.2$	$80 + 30e^{-(52-\mu_3)}$	Node 2, Line C
		$50.2 < \mu_3 \le 80$	85	Node 4, Line E
	2	$0 \le \mu_2 \le 25$	50	Node 4, Line D
		$25 < \mu_2 \le 55$	80	Node 4, Line D
		$55 < \mu_2 \le 85$	110	Node 4, Line D
0	1	0	53.45	Node 3, Line B

Referring to Table IV and Figure 6, the trip begins by walking to BS2 and boarding line B. The traveler exits line B at BS4. Depending on his arrival time there, the traveler might then either board line C or line E. Line E takes the traveler straight to the destination. However, if line C is boarded, the traveler transfers a second time at BS3 to line D.

The time adaptive decision rule differs from the optimal simple path. Instead of beginning with line A, the traveler now begins the trip with line B. Line B has become more attractive because of the opportunity to adapt the route at BS4. Without adapting, line B is inferior to line A.

CONCLUSIONS

THIS paper has shown that:

1. Standard shortest path algorithms do not identify the least expected travel time path on networks with travel times that are both random and time-dependent; and

2. The optimal "route choice" on such a network is not a simple path but an adaptive decision rule, conditioned to the times nodes are actually reached.

Methods were proposed for finding the least expected travel time path and the least expected travel time adaptive decision rule. The computation time for either method is much greater than that of standard shortest path algorithms (at least 10 times as large; Hall[7]). Therefore, they should only be applied when the potential savings are great.

As discussed in Spiess,[16] transit riders sometimes use other "strategies" (specific to transit networks) that might be viewed as adaptive route choice. Simple strategies, such as boarding the first bus to arrive among several alternative lines, can be effective when the lines have identical, or nearly identical, routes. Unfortunately, complicated forms of adaptive route choice, despite their theoretical appeal, are not very practical on transit networks. To be effective, travelers must be willing to transfer between lines, and there must be several alternative routes between each origin and destination. Travelers must also be willing to change their routes from day to day. Previous research (HALL[8]) has indicated that time-adaptive route choice would reduce travel time by less than 1%.

The methods developed here are most useful in analyzing the phenomena of route adaptation and travel time variability. They can be used to identify networks (such as motor carrier, rail, etc.) where route adaptation might offer substantial savings, and to develop adaptation strategies. They can also be used to identify situations where standard shortest path algorithms do not identify least expected travel time paths.

ACKNOWLEDGMENTS

THIS PAPER benefited from the advace of Carlos Daganzo, Stuart Dreyfus and Gordon Newell. Research was supported by a grant from the National Science Foundation.

REFERENCES

1. R. E. BELLMAN, *Dynamic Programming*, Princeton University Press, Princeton, N.J., 1957.
2. R. E. BELLMAN, "On a Routing Problem," *Quart. Appl. Math.* **16,** 87–90 (1958).
3. R. B. DIAL, "Transit Pathfinder Algorithm," *Highway Res. Rec. No. 205*, 67–85 (1967).
4. E. W. DIJKSTRA. "A Note on Two Problems on Connexion with Graphs," *Numer. Math.* **1,** 395–412 (1959).
5. S. E. DREYFUS, "An Appraisal of Some Shortest-Path Algorithms," *Opns. Res.* **17,** 395–412 (1969).
6. J. GILSINN, P. SAUNDERS AND M. PEARL, Path Finding Algorithm and Data Structure for Point-to-Point Trip Management, U.S. National Bureau of Standards, NBSIR 75-676, 1975.
7. R. W. HALL, Traveler Route Choice Under Six Information Scenarios: Transit Level of Service Implications, Ph.D. thesis, Institute of Transportation Studies, University of California, Berkeley, 1982.
8. R. W. HALL, "Traveler Route Choice: Travel Time Savings from Improved Information and Adaptive Decisions," *Trans. Res.* **17A,** 201–214 (1983).
9. R. W. HOWARD, "Dynamic Programming," *Mgmt. Sci.* **12,** 317–348 (1966).
10. G. M. LAMB AND G. HAYERS, "Introduction to Transportation Planning. 2. Treatment of Networks," *Traffic Eng. Control* **11,** 486–489 (1970).
11. A. LAST AND S. E. LEAK, "Transept: A Bus Model," *Traffic Engrg. Control* **17,** 14–20 (1976).
12. E. L. LAWLER, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart & Winston, New York, 1976.
13. F. LE CLERCQ, "A Public Transport Assignment Method," *Traffic Eng. Control* **13,** 91–96 (1971).
14. K. G. MURTY, "An Algorithm for Ranking all Assignments in Increasing Order of Cost," *Opns. Res.* **16,** 682–687 (1968).
15. A. R. PIERCE, "Bibliography on Algorithms for Shortest Path, Shortest Spanning Tree, and Related Circuity Touring Problems (1956–1974)," *Networks* **5,** 129–149 (1975).
16. H. SPIESS, On Optimal Route Choice Strategies in Transit Networks, Centre de Recherche Sur Les Transports, Publication #286, Montreal, 1986.