PERGAMON

Transportation Research Part B 35 (2001) 749–765

TRANSPORTATION
RESEARCH
PART B

www.elsevier.com/locate/trb

An adaptive routing algorithm for in-vehicle route guidance systems with real-time information

Liping Fu [*]

Department of Civil Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1

Received 21 June 1999; received in revised form 17 January 2000; accepted 7 February 2000

Abstract

This paper examines the problem of routing a given vehicle through a traffic network in which travel time on each link can be modeled as a random variable and its realization can be estimated in advance and made available to the vehicle's routing system before it enters the link. The underlying problem is formulated as the closed-loop adaptive shortest path routing problem (CASPRP) with the objective of identifying only the immediate link, instead of a whole path, to account for the future availability of travel time information on individual links. Having formulated the problem as a dynamic program and identified the associated difficulties, we apply an approximate probabilistic treatment to the recurrent relations and propose a labeling algorithm to solve the resultant equations. The proposed algorithm is proved theoretically to have the same computational complexity as the traditional label-correcting (LC) algorithm for the classic shortest path problems. Computational experiments on a set of randomly generated networks and a realistic road network demonstrate the efficiency of the proposed algorithm and the advantage of adaptive routing systems. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Shortest path problem; Traffic network; Adaptive routing; Intelligent transportation systems; Route guidance systems

1. Introduction

In-vehicle route guidance systems (RGS) have gained significant popularity worldwide due to their promising potential to reduce many transportation-related problems such as driving-induced stress, congestion, traffic accidents, and air pollution. By applying advanced surveillance, telecommunication and computer technologies, RGS aims to provide drivers with turn-by-turn

[*] Tel.: +1-519-885-1211 ext. 3984; fax: +1-519-888-6197.
E-mail address: lfu@uwaterloo.ca (L. Fu).

PII: S0191-2615(00)00019-9

guidance on optimal routes to their desired destinations based on real-time traffic information collected over the underlying road network through loop detectors, probe vehicles and video surveillance systems.

One of the crucial components in RGS is the routing system that is used to identify optimal routes from a current vehicle position to a specific destination in the underlying network. A common routing strategy employed in many RGS that have been field tested or simulated in the past works as follows: travel times on individual links in the network are first updated based on real-time information; a shortest path algorithm is then utilized to compute a complete path from the current vehicle position to the destination with a common routing objective of minimizing the expected travel time; lastly, the computed path is suggested to the driver to be followed. This routing strategy is essentially a priori optimization that does not take into account the future availability of information on possible realizations of link travel times, and therefore may generate suboptimal solutions. A routing scheme (policy) that specifies the next link to travel, as opposed to a fixed path, and defers route choices until later nodes are reached may become appropriate. The objective of this research is to develop an optimal adaptive routing algorithm that potentially can be used in RGS to which real-time information on link travel times is available.

This paper is organized as follows. We begin with an overview of various shortest path problems under three different routing schemes with a specific focus on the availability and the use of real-time information, and provide a review of the related literature. In Section 3, we present a formal definition of the closed-loop adaptive shortest path routing problem (CASPRP) and a dynamic programming formulation. The difficulties arising in the formulation are discussed and demonstrated using a simple network problem. In Section 4, we propose a method to approximate the recurrent equations, based on which we have designed a labeling algorithm to solve the CASPRP. Section 6 presents the results of a computational experiment using a set of randomly generated networks and a realistic road network from the city of Edmonton, Alberta, Canada.

2. Background and related work

Routing a vehicle through a road network intuitively calls for the selection of paths from an origin node to a destination node that optimizes a given objective. The underlying problem is the well-known shortest path problem (SPP) that has been studied extensively for many years, resulting in a large number of models and algorithms (Deo and Pang, 1984). The following review primarily focuses on the availability and the use of information in modeling and solving the shortest path problems. Without loss of generality, we assume that travel time minimization is the only routing criterion.

Based on the availability and the use of real-time information, the following three routing schemes may be applied: non-adaptive routing rule (NAR), open-loop adaptive routing rule (OAR) and closed-loop adaptive routing rule (CAR), as shown in Fig. 1. Under the NAR, a complete, fixed path is to be identified on the basis of a priori or historical travel time information. Such paths are usually computed before a trip starts and no en-route, adaptive diversion is considered either due to lack of real-time information or because there is no RGS available. Consequently, real-time information is irrelevant.

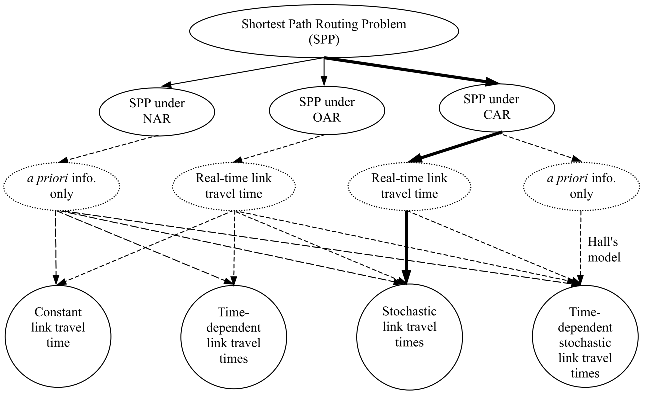

Shortest Path Routing Problem (SPP)
SPP under NAR
SPP under OAR
SPP under CAR
a priori info. only
Real-time link travel time
Real-time link travel time
a priori info. only
Hall's model
Constant link travel time
Time-dependent link travel times
Stochastic link travel times
Time-dependent stochastic link travel times

Fig. 1. Shortest path problems under different routing schemes and information.

Research on the shortest path problems under NAR has constituted the major efforts in the past, mainly focusing on seeking for more efficient algorithms to solve the problem and its variations (Deo and Pang, 1984). Depending on the type of a priori information available on link travel times (and thus how they can be modeled), the shortest path problems under NAR can be further classified into four categories (Fig. 1). The most basic model assumes that link travel times are constant, that is, both deterministic and time-independent. Problems of this type are referred as to the classic shortest path problems and can be solved using efficient labeling algorithms (Dijkstra, 1959; Gallo and Pallottino 1984; Moore, 1959). The same types of algorithms can also be used to solve the shortest path problems with time-dependent link travel times under the minor first-in-first-out (FIFO) condition (Chabini, 1997; Cooke and Hasley, 1966; Dreyfus, 1969; Kaufman and Smith, 1993).

When link travel times are modeled as random variables with known probability distribution, the resulting problem could have a large number of variations, depending on what utility or cost function is used to model the traveler's risk attitude (Frank, 1969; Loui, 1983; Mirchandani, 1976; Mirchandani and Soroush, 1986; Murthy and Sarkar, 1996). If the routing objective is to identify the expected shortest path (i.e., risk neutral, linear utility function), then the problem simply reduces to a deterministic shortest path problem in a network where the random link travel times are replaced by their expected values and can be solved using the efficient labeling algorithms.

The shortest path problem in networks where link travel times are both time-dependent and stochastic was first studied by Hall (1986). In examining the passenger route choice problem in a transit network, Hall observed that the standard shortest path algorithm might fail to find the expected shortest path in these networks. This observation was further proved by Fu and Rilett (1998), when they studied the same problem but in general road traffic networks.

The OAR is similar to NAR in the sense that a complete path is to be computed. However, OAR assumes that real-time information on link travel times is available as the routed vehicle advances and better paths may be identified en-route with this information. As a result, only the first link of the identified path needs to be committed by the vehicle and re-optimization can be invoked whenever the vehicle approaches a new decision point (intersection) and a diversion from the path previously identified is possible with real-time information up to that point of time. Note that the shortest path problem to be solved under OAR can be considered as one under NAR and solved efficiently by the traditional labeling algorithms. Similar to the SPP under NAR, various subproblems may be defined under OAR on the basis of how link travel times are modeled, as shown in Fig. 1.

Unlike OAR, an optimal adaptive routing system should be one of CAR, which suggests only immediate action, taking into account current as well as future availability of travel time information and thus the future opportunities to divert to different routes. While the concepts of adaptation and learning have been the focal point of research and development in the areas of system control, communication and dynamic programming since the 1960s (Bellman 1959; Sworder 1966), their application in transportation network routing problems was quite recent and scant (Hall, 1983; 1986). One explanation could be that information provision to motorists, which is the key to adaptation and learning, is just a recent possibility.

Based on a set of travel experiments on transit users' route choice under different information scenarios and route choice rules, Hall (1983) found that adaptive route choice could potentially lower travel time as compared to non-adaptive route choice with its effectiveness depending on the number of routes that have comparable travel times. Hall (1986) further explored this concept in a model of transit networks where travel times on links (transit lines) are random and time-dependent with known probability distributions as functions of a traveler's arrival time. Since the arrival time will be made known to travelers (passengers) when they arrive at a node of the network (bus stops or transfer points), so will the travel time distribution. Hall (1986) pointed out that, by accounting for the future availability of this information and making route choices accordingly, more efficient travel can be made by deferring route choice until later points are reached, as opposed to selecting a whole path at the beginning of or during the journey. A solution method based on dynamic programming technique was proposed to solve the so-called time-adaptive route choice problem. It should be pointed out that in Hall's adaptive routing model, the "real-time information" used is the arrival times at individual nodes or links, and the time-dependent travel time on each link is assumed to be known a priori and real-time information on link travel times is irrelevant.

In this paper, we consider a routing scheme similar to Hall's, but assume that real-time information on link travel times is available to the routed vehicle during the course of its travel and this information allows an estimate of the realization of the travel time on each link being made available to the vehicle before it enters that link. An algorithmic scheme is proposed to take into account the time-dependency of link travel times.

3. Problem definition and properties

A traffic network represented by a directed graph $G = (N, A)$ consists of a set of $|N|$ nodes and a set of $|A|$ links. Let $B(i)$ represent the outgoing link adjacent list of node i, that is, $B(i) = \{(i,j) \in A : j \in N\}$. Consider a hypothetical situation in which a RGS-equipped vehicle is currently traveling on the link (i,s) towards the destination node r and its in-vehicle routing system has been requested to make a recommendation about on which link of $(s,j) \in B(s)$ the vehicle should enter next so as to minimize the expected travel time to the destination node.

It is assumed that the routing system has available complete information on the topology of the network G and current estimates of travel times on individual links. Travel time on a link, or link travel time, is assumed to be a random variable with its mean and standard deviation being predictable on the basis of a priori historical travel time information and real-time data. Furthermore, it is assumed that the uncertainty of the travel time on a link would gradually dissolve to the routing system of a given vehicle as the vehicle gets closer to that link, and that the actual link travel time – a realization of the random link travel time can be observed or precisely estimated before the vehicle enters the link. Let X_a denote the link travel time on link a and by μ_a and σ_a its mean and standard deviation, respectively.

It is important to point out that both μ_a and σ_a should be considered as the current estimates representing the possible travel time that a vehicle would experience if it were guided to that link. Thus, the mean and standard deviation of link travel time on a specific link are vehicle-specific, depending on how far away a vehicle is positioned from that link. This is especially true in time-dependent traffic networks. An estimation method similar to Koutsopoulos and Xu (1993) may be used to estimate μ_a and σ_a based on real-time data and historical travel time information. In this way, the time-dependency of link travel times can be considered implicitly and heuristically.

Based on the definition and assumption stated above, we define the CASPRP as follows: *Find the link $a^* \in B(s)$ that minimizes the expected travel time from the node s to the destination node r under CAR.* The problem can be formally stated as

$$\text{(CASPRP)} \quad a^* = \operatorname{argmin}\{g(i) + \mu_a |\ \forall a = (s,i) \in B(s)\}, \tag{1}$$

where $g(i)$ is the expected travel time to the destination node r given that the vehicle arrives at the node i and continues to be routed under CAR. Clearly, a solution to the CASPRP requires that the expected travel time, $g(i)$, for every node i be solved, which in turn calls for a formulation of a dynamic recursive equation for a general node i in the network. Consider an arbitrary node i in the network and denote by the random variable Y_i the travel time from node i to the destination node r given the vehicle arrives at the node i and continues to be routed under CAR, then

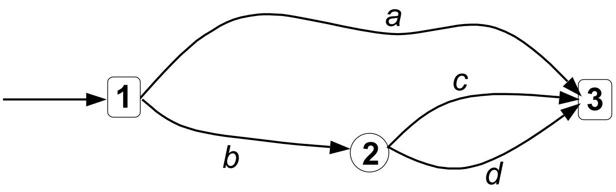
$$g_{(i)} = E[Y_i], \tag{2}$$

and

$$Y_i = \min\{g(j) + X_a | \forall a = (i,j) \in B(i)\}. \tag{3}$$

Eqs. (2) and (3), coupled with the boundary condition $g(r) = 0$, can be considered as the recurrent relation that in theory can be solved using the dynamic programming technique (Bellman, 1965). Two difficulties arise when applying the dynamic programming technique. First, it requires constructing an acyclic layered network from the current node to the destination node out of the original network, which is not a trivial task in generic traffic networks containing cycles. Second, solution to Eqs. (2) and (3) requires that the probability density function (PDF) of link travel times be known, and that the PDF as well as the expected value of Y_i be decided iteratively. These difficulties, along with some other properties related to the CASPRP, are illustrated using an example network in the following section.

As shown in Fig. 2, the example network consists of five links with the a priori travel time on each link being listed in the same figure. A vehicle is currently traveling towards node 1 and the current estimates (a posteriori) of link travel times are assumed to be available, as shown in Fig. 2. The problem is to find the optimal link to travel next that will ultimately minimize its expected travel time to node 3. The three routing schemes described in Section 2 as well as Hall's time-



Travel Time Table:

Link	*A Priori* Information	Estimate Based on Real-time Information (*Posteriori*)
a	N{10, 2}	10 minutes
b	N{5, 1}	5 minutes
c	N{4, 1}	N{5.1, 0.5}
d	N{6, 2}	N{5.1, 0.5}

Note: * The notation $N\{\mu,\sigma\}$ represents a normal distribution with the mean μ and the standard deviation σ;

** Travel times between individual links are statistically independent

Fig. 2. A simple network to illustrate the difference among different routing schemes.

adaptive method are considered. First, if the driver has no access to real-time information, as in the case of NAR, then a priori information is used in identifying the optimal routes. Consequently, path $\{b, c\}$ is selected as it has the minimum expected travel time (9 min).

Alternatively, if the driver has access to real-time information, he must identify a complete path from node 1 to node 3, as it would be implemented under OAR. Three possible paths – $\{a\}, \{b, c\}$ and $\{c, d\}$ are available, which, based on the real-time (a posteriori) link travel time information, have the expected travel times of 10, 10.1 and 10.1 min, respectively. As a result, path $\{a\}$ is the optimal decision at node 1.

If Hall's time-adaptive routing scheme is followed, the optimal link to travel next at node 1 is link *b* and the optimal routing decision when the vehicle reaches node 2 is link *c*, regardless of when the vehicle will arrive at node 2 because link travel times in this case are time-unvaried. This example also shows that the adaptivity of Hall's routing model is in effect only when link travel times are both time-dependent and random.

The last scenario, which is the focus of this paper, assumes that the CAR rule is followed, that is, only the next link to travel needs to be identified, and that the actual travel times on link *c* and *d* will be made known (with real-time information) to the routing system when the vehicle arrives at node 2. Under this routing model, the link with lower travel time when the vehicle reaches node 2 can be selected. Travel time from node 2 to node 3 under CAR, Y_2, can therefore be obtained from Eq. (3).

$$Y_2 = \min\{g(3) + X_c, g(3) + X_d\} = \min\{X_c, X_d\}. \tag{4}$$

The probability density function of Y_2, denoted by $f_Y(y)$, can be determined from the probability density functions of X_c and X_d. Since X_c and X_d have the same normal distribution, $f_Y(y)$ can be expressed as Eq. (5) (Ang and Tang, 1984).

$$f_Y(y) = 2[1 - F_X(x)]f_X(x), \tag{5}$$

where $f_X(x)$ and $F_X(x)$ are, respectively, the PDF and the cumulative distribution function of the travel times on link c or d. The expectation of Y_2 is therefore

$$E[Y_2] = \int_{-\infty}^{\infty} 2[1 - F_X(x)]f_X(x)x \, \mathrm{d}x. \tag{6}$$

As no analytical solution could be obtained for Eq. (6), we resorted to a numerical process that yielded a value of 4.8 min. The total expected travel time from node 1 to node 3 by taking link b and then deciding which link to take next at node 2 would be 9.8 min. The optimal decision is therefore taking link b and deferring the decision on which link to take next until the vehicle arrives at node 2, by which time the information on the actual travel time on the links c and d is available and the link with a lower travel time can be selected accordingly.

Two observations can be made from this example. First, a routing system based on the closed-loop adaptive descision rule (CAR) would result in better decisions as compared to a system with the open-loop decision rule (OAR) if the expected travel time is to be minimized. The reason, as illustrated by this example, is that the OAR scheme may miss the paths that are better in terms of the opportunity to divert as the vehicle transverses through the network and new information is made available.

Second, the example shows that it is practically impossible to obtain an analytical solution of closed form to Eqs. (2) and (3) and thus the CASPRP, even for a very simple network. Approximation is clearly required to make the recurrent relation (Eqs. (2) and (3)) operational and solvable using a dynamic programming technique. We propose an approximate scheme to resolve this issue, based on which we subsequently prove that the CASPRP can be solved efficiently using an algorithm similar to the traditional labeling algorithm.

4. Algorithm

4.1. Approximation of the fundamental recurrent relation

Consider a specific node i in the network G defined in Section 3, from which there are $|B(i)|$ links, represented by $a_1, a_2, \ldots, a_{|B(i)|}$, emanating from the node, as shown in Fig. 3. Eq. (3) for this node can be equivalently transformed into the following set of $|B(i)|$, (Eqs. (7) and (8)) that can be solved sequentially

$$Y_i^k = \min\left\{Y_i^{k-1}, g(j_k) + X_{a_k}\right\}, \quad k = 2, 3, \ldots, |B(i)|, \tag{7}$$

and

$$Y_i^1 = g(j) + X_{a_1}, \tag{8}$$

where j_k is the end node of the link a_k, and Y_i^k is the minimum travel time from node i to the destination node r, considering the travel times of the links: $a_1, a_2, \ldots, a_k$. Clearly, $Y_i = Y_i^{|B(i)|}$.

The expected travel time from node i to the destination node $r, g(i)$, can be therefore determined by iteratively solving the following equations:

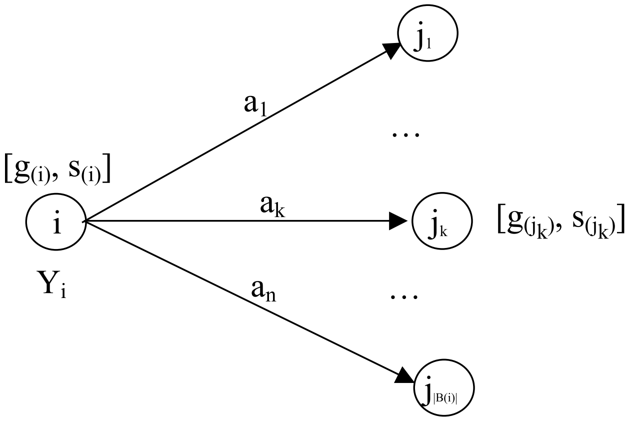


Fig. 3. Routing decision at a specific node.

$$g^k_{(i)} = E\left[Y^k_i\right] = E\left[\min\left\{Y^{k-1}_i, g(j_k) + X_{a_k}\right\}\right], \quad k = 2, 3, \ldots, |B(i)| \tag{9}$$

with

$$g^1_{(i)} = E\left[Y^1_i\right] = E[g(j_1) + X_{a_1}] = g_{(j_1)} + \mu_{a_1}, \tag{10}$$

and $g_{(i)} = E[Y_i] = g^{|B(i)|}_{(i)}$.

We now apply an approximate procedure to make Eqs. (9) and (10) operational without having to determine the PDF of Y^k_i and related integrals. The approximation method was originated by Rosenblueth (1975) with the idea of estimating the first three moments of a function of random variables based on the first three moments of the random variables (instead of the PDFs of the random variables). Appendix A provides a detailed derivation on how this technique can be used to approximate Eqs. (9) and (10). To manage the level of complexity, we assume that the skewness of travel times is negligible and thus consider only the mean and standard deviation of travel times. Let $s(i)$ denote the estimate of the standard deviation of Y_i. Eqs. (9) and (10) can be approximated by Eqs. (11)–(14)

$$g^k_{(i)} = \frac{1}{4}\sum_{m=1}^{2}\sum_{n=1}^{2} \min\left\{g^{k-1}_{(i)} + (-1)^m \cdot s^{k-1}_{(i)},\ g_{(j_k)} + \mu_{a_k} + (-1)^n \cdot \sigma_{a_k}\right\},$$
$$k = 2, 3, \ldots, |B(i)|, \tag{11}$$

$$\left(s^k_{(i)}\right)^2 = \frac{1}{4}\sum_{m=1}^{2}\sum_{n=1}^{2}\left(\min\left\{g^{k-1}_{(i)} + (-1)^m \cdot s^{k-1}_{(i)},\ g_{(j_k)} + \mu_{a_k} + (-1)^n \cdot \sigma_{a_k}\right\}\right)^2 - \left(g^k_{(i)}\right)^2,$$
$$k = 2, 3, \ldots, |B(i)| \tag{12}$$

with

$$g^1_{(i)} = g(j_1) + \mu_{a_1}, \tag{13}$$

$$s^1_{(i)} = \sigma_{a_1}. \tag{14}$$

The above Eqs. (11)–(14) fundamentally represent the recurrent relation between the optimal value at node $i, g(i)$, and the optimal value at node $j, g(j)$, with $(i,j) \in B(i)$, similar to the recurrent equation for the classic shortest path problems. Based on Eqs. (11)–(14), we designed an efficient labeling algorithm to solve the CASPRP, as described in Section 4.2.

4.2. Adaptive labeling (AL) algorithm

The proposed algorithm falls fundamentally into the category of label-correcting (LC) algorithms. The algorithm maintains two labels, $g(i)$ and $s(i)$, for each node i, representing the mean and standard deviation of the travel time from node i to the destination node r under CAR. The key part of the algorithm starts at the destination node r and iteratively updates the labels of individual nodes based on Eqs. (11)–(14). The labels of a specific node are calculated based on the travel times (mean and standard deviation) of links emanating from that node and the labels at the end nodes of these links.

All the nodes that are eligible to be examined are kept in a scan-eligible node set (Q). The Q is maintained by a bucket-type data structure (Dial, 1969) similar to a radix proposed by Ahuja et al. (1990), except that nodes in each bucket are not sorted. While the advantage of using a bucket data structure in maintaining a sorted list for a label-setting algorithm has been well described elsewhere (Ahuja et al., 1993), the reason we use this structure is that the nodes with lower labels would have a higher possibility to update the label at the origin node s (or to be used by the vehicle) and therefore should be given a higher order of priority. The reason we use unsorted buckets is that in our algorithm, a node that has been selected for examination may be inserted into the list again in later stages, which therefore makes it unnecessary to maintain rigorous ordered list – sorted buckets as in a label-setting algorithm. We note that while the efficiency of the proposed algorithm depends on what data structure is used to maintain the Q, the correctness of algorithm always holds as proved by the following theorem.

The proposed algorithm involves the following steps:

0. (*Preprocessing*) Apply a LC algorithm (forward) to compute the minimum paths from the node s to all other nodes, and at the same time estimate μ_a and $\sigma_a \forall a \in A$ based on historical travel time and real-time data.

1. (*Initialization*) Apply a LC algorithm (backward) to compute the minimum paths from all nodes to the destination node r based on the estimated $\mu_a, \forall a \in A$. The resulting label at each node, $g(i)$, is the upper bound of the optimal travel time from that node to the destination node under CAR. If $g_{(s)} = \infty$, then there is no feasible solution, stop. Otherwise, insert the destination node r into the scan eligible node set $Q = \{r\}$.

2. (*Updating*) Select and remove node i from Q and:

   2.1 For every link (k,i) entering node i, insert node k into Q.

   2.2 Calculate the new labels at node i $(g_{\text{new}}, s_{\text{new}})$ based on Eqs. (11)–(14): set $g_{\text{new}} = \infty$ and $s_{\text{new}} = 0$, and iterate all the links emanating from node i, for link $a = (i,j) \in B(i)$

$$g_{\text{new}} = \frac{1}{4} \sum_{m=1}^{2} \sum_{n=1}^{2} \min \left\{ g_{\text{new}} + (-1)^m \cdot s_{\text{new}}, g_{(j)} + \mu_a + (-1)^n \cdot \sigma_a \right\},$$

$$s_{\text{new}} = \sqrt{\frac{1}{4}\sum_{m=1}^{2}\sum_{n=1}^{2}\left(\min\left\{g_{\text{new}} + (-1)^m \cdot s_{\text{new}}, g_{(j)} + \mu_a + (-1)^n \cdot \sigma_a\right\}\right)^2 - g_{\text{new}}^2}.$$

2.3 If $g_{(i)} > g_{\text{new}}$, then $g_{(i)} = g_{\text{new}}$ and $s_{(i)} = s_{\text{new}}$.

3. (*Terminating iteration*) If $Q = \emptyset$, then go to Step 4. Otherwise, go to Step 2.

4. (*Solution*) Examine all the links emanating from node s and find the link $a^* = (s, j)$ which has the minimum $g(j) + \mu_a$. Stop.

Theorem. *Upon termination of the AL algorithm, $g(s)$ is either an infinite number, indicating that there is no solution to the CASPRP, or a finite number that represents the expected travel time from the node s to the destination node r under CAR.*

Proof. Our proof includes the following four parts:

First, we prove that the AL algorithm terminates or converges in a finite number of iterations. This can be proved by contradiction. Suppose that the scan-eligible list Q is not exhausted in a finite number of iterations, this means that there is at least one node that is repeatedly inserted into the list, which in turn implies that the label of this node is infinitely improved at each step. This will eventually lead to a negative label at this node, which contradicts the fact that all labels must be positive due to positive link travel times.

Second, we prove that, after Step 1, an infinite label for a node means there is no path from that node to the destination node r and that a finite label for a node is the upper bound of the label at that node under the CAR rule. The first assertion is self-evident due to the correctness of the LC algorithm. The second assertion can be proved as follows. A label of finite value associated with a node after Step 1 is the expected travel time from that node to the destination node on the path with minimum expected travel time. Since there may exist other alternative paths that have equal or shorter travel time than the expected minimum path, the use of those paths under the CAR rule would therefore lower the expected travel time.

Next, we prove that the label $g(i)$ for every node i either stays unchanged or decreases and the inequality $g_{\text{new}} \leqslant g(i)$ holds at each iteration, where g_{new} is computed at Step 2.2. The proof on the first part of the statement can be derived from Step 2.3, in which $g(i)$ is replaced by g_{new} only when $g_{\text{new}} < g(i)$. That means that the $g(i)$ value would never increase during the iterations. It can be observed from Eqs. (11)–(14) that g_{new} is a monotonic function of the label at each successor node $j, g(j)$ with $(i, j) \in B(i)$. The non-increasing feature of $g(j)$ guarantees that the g_{new} value calculated based on $g(j)$ would always be less or equal to the $g(i)$ which is calculated based on $g(j)$ at the previous iteration. Therefore, the relation $g_{\text{new}} \leqslant g(i)$ holds at each iteration.

Lastly, we prove that, upon termination, the relation $g(i) = g_{\text{new}}$ holds for every node i. Since we have proved that $g_{\text{new}} \leqslant g(i)$, we only need to contradict the inequality $g_{\text{new}} < g(i)$. The proof can be simply derived from Step 2 of the algorithm. Suppose there exists a node i such that $g_{\text{new}} < g(i)$, then node i has not been scanned (once a node is scanned, Eqs. (11)–(14) holds) and thus is still in Q. This contradicts the statement that the algorithm has terminated. □

Proposition. *The AL algorithm has computational complexity $O(|A||N|)$.*

Proof. We prove this when all the associated variables are positive integers [1]. The LC algorithm used in Step 0 and 1 has computation complexity $O(C|A||N|)$ (Ahuja et al., 1993), where C is the maximum link travel time. We have shown that the expected travel time from node i to the destination node r, $g(i)$, is bounded from above by the expected travel time from that node to the destination node on the path with minimum expected travel time (the $g(i)$ value after Step 1), which is also bounded by $C|A|$. $g(i)$ is also bounded from below by 0. Since each update of $g(i)$ (Step 3) decreases the label by at least one unit, the number of possible updates for $g(i)$ is utmost $C|A|$. As there are a total of $|N|$ nodes whose labels may need to be updated, the maximum total number of updates (or steps) is therefore $C|A||N|$. Observe that in a general road traffic network ($G = \{N, A\}$), the maximum link length C can be considered as a value that is independent of the network size ($|N|\mathrm{or}|A|$). Consequently, the AL algorithm has computational complexity $O(|A||N|)$. Note that this also proves that the algorithm terminates in a finite number of steps. □

5. Computational analysis

The objective of this section is to demonstrate the computational efficiency of the proposed algorithm and to provide some empirical evidence on the potential benefit of the closed-loop adaptive routing strategy. Both the LC algorithm and the AL are coded in C++. The LC algorithm uses a deque data structure for maintaining the scan-eligible data set (Pape, 1974). The bucket structure used in AL uses a bucket width of 60 s and a total of 1000 buckets. Note that the parameters associated with the bucket structure may be optimized for a specific network to further improve the computational efficiency of the AL algorithm. The algorithms are executed under the Microsoft Windows operating environment on a Pentium III with 450 MHz speed and 128 MB RAM. The experiment was performed on a set of hypothetical networks and a real road network described as follows:

1. The hypothetical networks are randomly generated grid networks with all the neighboring nodes (intersections) connected by two links, one in each direction. Each link has a length of 400 m and a speed that is randomly selected in a uniform distribution from 20 to 60 km/h. The mean travel time on a link is calculated as the ratio of the link length to the link speed. The standard deviation of the travel time on a specific link is estimated by multiplying the mean link travel time with a coefficient of variation (COV) that is randomly distributed in the interval [0.05, 0.25]. Six grid networks are considered: $50 \times 50, 50 \times 100, 100 \times 100, 100 \times 200, 200 \times 200$ and 400×400, where the first number is the rows and the second number is the columns.
2. The real road network used in this analysis is from the city of Edmonton, Alberta. The network consists of 3800 links and 1400 nodes, representing the arterial network system. Due to the lack of real travel time data, the real-time estimates of stochastic travel time patterns in the network were created based on a hypothetical change pattern in travel time similar to the hypothetical networks. The mean link travel time was first calculated for each link based on link length and posted travel speed, which are available as part of the network database. The standard devia-

[1] Note that the assumption of integer labels and travel times are reasonable when we use seconds as the unit of time. However, if real numbers are to be used, then the updating condition at Step 2.3 should be changed into $g(i) - g_{\mathrm{new}} < \varepsilon$, where ε is the minimum allowable error in the label. The algorithm then has a computational complexity of $(C|A||N|/\varepsilon)$.

tions of link travel times were generated in the same way as for the random networks with COV uniformly distributed within a given range. Three ranges of COV were considered: [0.1, 0.2], [0.1, 0.5] and [0.1, 0.8], representing different levels of variability of traffic conditions.

We first examine the computational efficiency of the AL algorithm. Table 1 lists the average CPU time required by the LC algorithm and the AL algorithm to solve the corresponding routing problem for a set of randomly generated trip pairs in each network. Note that since our test did not involve real-time data fusion and estimation, the AL algorithm for this test excluded the preprocessing step (Step 0). It can be observed from Table 1 that the CPU time ratio between LC and AL is quite constant (2.1–2.5) across different networks, which confirms our theoretical evaluation. We can therefore argue that the computational efficiency of the AL algorithm is comparable to the LC algorithm.

The next analysis focuses on the difference in routing results between the two routing strategies – OAR and CAR. The Edmonton network was used for this analysis. For a given trip of known origin and destination, the LC algorithm was first used to identify a complete path from the origin node to the destination node and the expected travel time, and the first links on the identified path were recorded, representing the routing results under OAR. The AL was then used to find the expected travel time and optimal link under CAR for the same trip. The results are summarized in Fig. 4, where each point represents the result of at least 50 randomly generated trip pairs for the corresponding trip distance range (measured along the shortest paths).

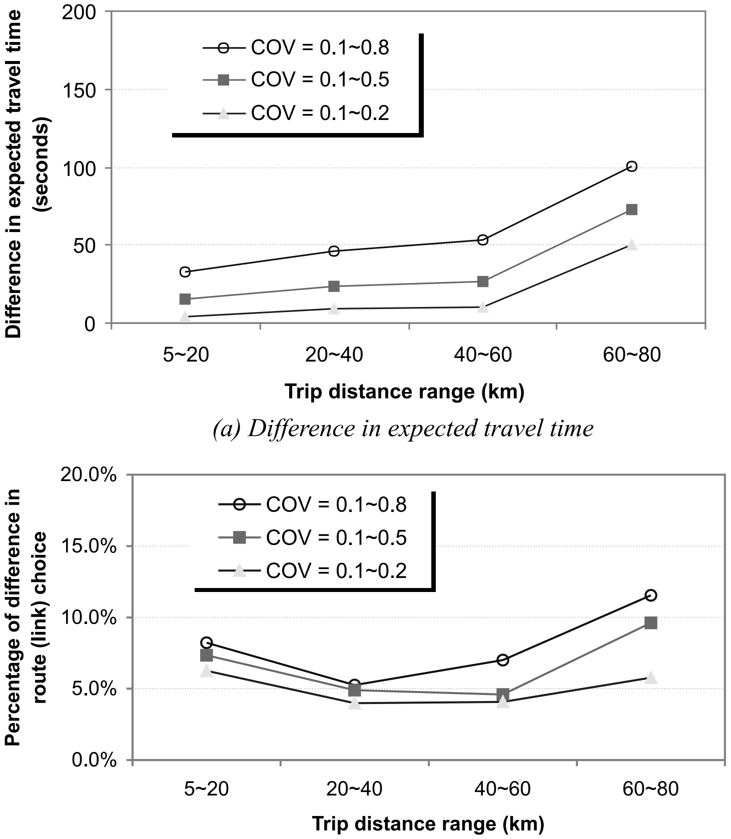
Two general observations can be made. First, as it would be expected, the differences in routing results between the two routing schemes become more pronounced under higher levels of travel time variability (or COV). This suggests that the scheme of adaptive routing may attain higher benefits in road networks with higher traffic variability and uncertainty. Second, the difference between the two routing schemes is quite significant: travel time saving with CAR reaches as high as 100 s (a relative travel time saving of approximately 5%) and the difference in route choice ranges from 4% to 12%.

The benefit of reduced expected travel time under adaptive routing scheme increases as the length of the trip increases, as shown in Fig. 4(a). Such trend implies that adaptive routing strategy or real-time information is more valuable to longer trips than to shorter trips. This phenomenon is expected because the longer trips have more alternative routes of comparable travel times and thus more en-route diversion opportunities at the future points of time.

It is interesting to observe from Fig. 4(b) that, unlike the expected travel time, the difference in route choice (or more precisely link choice) resulting from the two routing schemes has a non-

Table 1
Performance of the AL algorithm as compared to the LC algorithm on different networks

Network	Number of nodes	Number of links	Average CPU time per trip		
			LC (s)	AL (s)	AL/LC ratio
50 × 50	2500	9800	0.018	0.040	2.20
50 × 100	5000	19,700	0.070	0.164	2.34
100 × 100	10,000	39,600	0.325	0.742	2.29
100 × 200	20,000	79,400	15.781	35.211	2.32
200 × 200	40,000	159,200	72.714	167.630	2.31
400 × 400	160,000	638,400	1156.790	2574.120	2.23
Edmonton network	1400	3800	0.011	0.005	2.20



(a) Difference in expected travel time

(b) Difference in route choice

Fig. 4. Difference in routing results between CAR and OAR.

monotonic relationship with the trip length. Although a convincing explanation to this phenomenon has yet to be found, we believe it results from two forces that augment the importance of adaptive routing strategy: the number of en-route diversion opportunities and the level of necessity to consider diversion. The number of en-route diversion opportunities normally increases as the trip length increases, as discussed previously. However, the level of necessity to consider diversion at the current position (origin node) is expected to decrease as the trip length increases because of increased diversion opportunities. If the former takes dominance for long trips and the latter for short trips, then the combined effect of them would indeed result in the relationship shown in Fig. 4(b).

6. Concluding remarks

This paper has examined the adaptive routing problem in traffic networks in which link travel times are modeled as random variables with known mean and standard deviation, and their realizations can be estimated based on real-time information collected over the links. The underlying problem, referred as to the CASPRP, was formulated as a set of recurrent equations for a dynamic programming solution. As it was found difficult to obtain a direct solution of closed form to these equations, an approximate probabilistic method was applied to make them operational. A labeling algorithm was developed to solve the resultant equations. The correctness of the algorithm was theoretically proved and its computational complexity is evaluated to be $O(|N||A|)$, which is the same as the LC algorithm for the classic shortest path problems. The algorithm was tested on a set of randomly generated networks with size ranging from one thousand links to half-a-million links, and an actual road network. The experiment confirmed the theoretical evaluation and demonstrated the practical importance of the proposed algorithm.

It should be noted that since the proposed routing model and algorithm and the associated analysis are still limited in many aspects, further research is needed. The following is a list of potential extensions suggested for future research:

1. The time-dependency of link travel times was handled algorithmically in this paper. An explicit consideration of this aspect in the routing model would serve as valuable extension to this research.
2. The empirical conclusions were obtained based on experiments on a limited number of networks with hypothesized link travel time patterns. It is therefore necessary to conduct further studies based on more realistic networks with real travel time data before any general conclusions may be made.
3. The proposed models and algorithms may be extended to take into account the skewness of the link travel time distributions and the correlation of travel times between individual links. We note that the approximation method used in this paper can handle the skewness and correlation (Rosenblueth, 1975). The major challenge therefore lies in how to obtain the required data (e.g., correlation coefficients) for this type of model.
4. The proposed algorithm works in a fashion similar to a LC algorithm, that is, the scan-eligible node set must be exhausted before the solution can be obtained. We expect that more efficient algorithms may be developed, which would take into account the origin–destination locations and avoid examining all nodes in the network, as the A* algorithm.
5. The suggested routing scheme, that identifies the immediate link, instead of a complete path, to travel, may not be desirable to motorists from a behavior point of view. Further evaluation on its potential benefits is therefore necessary before it can be recommended for practical use.
6. This paper considers the problem of routing a single vehicle, which is suitable only for the case of low RGS market penetration. An important extension would therefore consider the interactions among individual driver's route choices as required for the situations of high market penetration. Under this extension, real-time information other than travel time such as origin–destination desires would then become important. This extension is however far from straightforward, likely requiring coupled consideration of dynamic traffic network loading and adaptive route choice (Friesz et al., 1999; Kaufman et al., 1998).

Acknowledgements

This research was funded through a Natural Science and Engineering Research Council of Canada Operating Grant and their assistance is gratefully acknowledged.

Appendix A

This appendix describes the derivation of Eqs. (11) and (12) from Eq. (9). For completeness, we rewrite Eq. (9) as follows (Eq. (15))

$$Y_i^k = \min\left\{Y_i^{k-1}, g_{(j_k)} + X_{a_k}\right\}. \tag{15}$$

Note that the above equation essentially represents a functional relation in which the random variable Y_i^k is a function of two random variables – Y_i^{k1} and X_{a_k}. Our objective is to determine the mean and standard deviation of Y_i^k based on the mean and standard deviation of Y_i^{k1} and X_{a_k}. Based on Rosenblueth's (1975) two point estimates method, the distribution of Y_i^{k1} and X_{a_k} are simplified as discrete distributes of two points. Under the assumption that link travel times are independent and the skewness of their distribution is negligible (or simply information on the skewness is not available), the simplified discrete distributions of Y_i^{k1} and X_{a_k} can be obtained by Eqs. (16)–(19)

$$P_1 = P\left(Y_i^{k-1} = g_{(i)}^{k-1} + s_{(i)}^{k-1}\right) = 1/2, \tag{16}$$

$$P_2 = P\left(Y_i^{k-1} = g_{(i)}^{k-1} - s_{(i)}^{k-1}\right) = 1/2, \tag{17}$$

$$P_3 = P(X_{a_k} = \mu a_k + \sigma_{a_k}) = 1/2, \tag{18}$$

$$P_4 = P(X_{a_k} = \mu a_k - \sigma_{a_k}) = 1/2, \tag{19}$$

where $g_{(i)}^{k1}$ and $s_{(i)}^{k1}$ are respectively the mean and standard deviation of Y_i^{k1}, μ_{a_k} and σ_{a_k} are the mean and standard deviation, respectively, of X_{a_k}. The mean and standard deviation of Y_i^k, $g_{(i)}^k$ and $s_{(i)}^k$, can then be decided based on the two-point discrete distribution as follows:

$$\begin{aligned} g_{(i)}^k &= P_1P_3 \min\left(g_{(i)}^{k-1} + s_{(i)}^{k-1}, g_{(j_k)} + \mu_{a_k} + \sigma_{a_k}\right) + P_1P_4 \min\left(g_{(i)}^{k-1} + s_{(i)}^{k-1}, g_{(j_k)} + \mu_{a_k} - \sigma_{a_k}\right) \\ &\quad + P_2P_3 \min\left(g_{(i)}^{k-1} - s_{(i)}^{k-1}, g_{(j_k)} + \mu_{a_k} + \sigma_{a_k}\right) + P_1P_3 \min\left(g_{(i)}^{k-1} - s_{(i)}^{k-1}, g_{(j_k)} + \mu_{a_k} - \sigma_{a_k}\right) \\ &= \frac{1}{4}\sum_{m=1}^{2}\sum_{n=1}^{2} \min\left\{g_{(i)}^{k-1} + (-1)^m \cdot s_{(i)}^{k-1}, g_{(j_k)} + \mu_{a_k} + (-1)^n \cdot \sigma_{a_k}\right\}, \end{aligned} \tag{20}$$

$$\begin{aligned} \left(s_{(i)}^k\right)^2 &= E\left[\left(Y_i^k\right)^2\right] - \left(g_{(i)}^k\right)^2 \\ &= \frac{1}{4}\sum_{m=1}^{2}\sum_{n=1}^{2}\left(\min\left\{g_{(i)}^{k-1} + (-1)^m \cdot s_{(i)}^{k-1}, g_{(j_k)} + \mu_{a_k} + (-1)^n \cdot \sigma_{a_k}\right\}\right)^2 - \left(g_{(i)}^k\right)^2. \end{aligned} \tag{21}$$

Finally we note that Rosenblueth (1975) suggested, based on a number of empirical analyses, that the method based on two-point estimates is "almost as satisfactory as those of a rigorous probabilistic treatment" under moderate coefficient of variation of independent variables.

References

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Upper Saddle River, NJ 07458.

Ahuja, R.K., Mehlhorn, K., Orlin, J.B., Tarjan, R.E., 1990. Faster algorithms for the shortest path problem. J. ACM 37, 213–323.

Ang, A.H-S., Tang, W.H., 1984. Probability Concepts in Engineering Planning and Design, vol. II, Decision, Risk and Reliability, Wiley, New York.

Bellman, R., 1959. Adaptive Control Process: A Guided Tour. Princeton University Press, Princeton, NJ.

Bellman, R., 1965. Dynamic Programming. Princeton University Press, Princeton, NJ.

Chabini, I., 1997. A new algorithm for shortest paths in discrete dynamic networks, in: Papageorgiu, M., Pouliezos, A., Proceedings of the Eighth International Federation of Automatic Control (IFAC) Symposium on Transportation Systems, vol. 2, Chania, Greece, June, 1997, pp. 551–556.

Cooke, K.L., Hasley, E., 1966. The shortest route through a network with time-dependent internodal transit times. J. Math. Anal. Appl. 14, 493–498.

Deo, N., Pang, C.Y., 1984. Shortest path algorithms: taxonomy and annotation. Networks 14, 175–323.

Dial, R.F., 1969. Algorithm 360 shortest path forest with topological ordering. Commun. ACM 12, 632–633.

Dijkstra, E.W., 1959. A note on two problems in connection with graphs. Numer. Math. 1, 269–271.

Dreyfus, S., 1969. An appraisal of some shortest path algorithms. Oper. Res. 17, 395–412.

Frank, H., 1969. Shortest paths in probability graphs. Oper. Res. 17, 583–599.

Fu, L., Rilett, L.R., 1998. Shortest Path Problems in Traffic Networks with Dynamic and Stochastic Link Travel Times. Transportation Research Part B: Methodological B 32, 7.

Friesz, T.L., Lall, S.V., Stough, R.R., 1999. Traffic Network Dynamics: Alternative Mathematical Formulations, Presented at the 78th Annual Meeting of Transportation Research Board, Washington, DC.

Gallo, G., Pallottino, S., 1984. Shortest path methods in transportation models. In: Florian, M. (Ed.), Transportation Planning models, Elsevier, Amsterdam, pp. 227–256.

Hall, R., 1983. Traveler route choice: travel time implications of improved information and adaptive decisions. Transportation Research A 17A (3), 201–214.

Hall, R., 1986. The fastest path through a network with random time-dependent travel time. Transport. Sci. 20 (3), 182–188.

Kaufman, D.E., Lee, J., Smith, R.L., 1993. Fastest paths in time-dependent networks for intelligent vehicle highway systems application. IVHS J. 1 (1), 1–11.

Kaufman, D.E., Smith, R.L., Wunderlich, K.E., 1998. User-equilibrium properties of fixed points in dynamic traffic assignment. Transport. Res. C 6, 1–16.

Koutsopoulos, H.N., Xu, H., 1993. An information discounting routing strategy for advanced travel information systems. Transport. Res. C 1 (3), 249–264.

Loui, P., 1983. Optimal paths in graphs with stochastic or multidimensional weights. Commun. ACM 26, 670–676.

Mirchandani, B.P., 1976. Shortest distance and reliability of probabilistic networks. Comput. Oper. Res. 3, 347–676.

Mirchandani, B.P., Soroush, H., 1986. Routes and flows in stochastic networks. In: Angrealtta, G., Mason F., Serafini, P., (Eds.), Advanced School on Stochastic in Combinatorial Optimization, CISM, Udine, Italy, pp. 129–177.

Murthy, I., Sarkar, S., 1996. A relaxation-based pruning technique for a class of stochastic shortest path problems. Transport. Sci. 30 (3), 220–236.

Moore, E.F., 1959. The shortest path through a maze. In: Proceedings of the International Symposium on Theory of Switching, Harvard University Press, Cambridge, pp. 285–292.

Pape, U., 1974. Implementation and efficiency of moore algorithms for the shortest route problem. Math. Progr. 7, 212–222.

Rosenblueth, E., 1975. Point estimates for probability moments. Proceedings, National Academy of Science. Mathematics 7 (10), 3812–3814.

Sworder, D., 1966. Optimal adaptive control systems, Mathematics in Science and Engineering, vol. 25, Academic Press, New York.