JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 127, No. 3, pp. 497–513, December 2005 (© 2005)
DOI: 10.1007/s10957-005-7498-5

Arriving on Time[1]

Y. Y. FAN,[2] R. E. KALABA,[3] AND J. E. MOORE, II[4]

Abstract. This research proposes a procedure for identifying dynamic routing policies in stochastic transportation networks. It addresses the problem of maximizing the probability of arriving on time. Given a current location (node), the goal is to identify the next node to visit so that the probability of arriving at the destination by time t or sooner is maximized, given the probability density functions for the link travel times. The Bellman principle of optimality is applied to formulate the mathematical model of this problem. The unknown functions describing the maximum probability of arriving on time are estimated accurately for a few sample networks by using the Picard method of successive approximations. The maximum probabilities can be evaluated without enumerating the network paths. The Laplace transform and its numerical inversion are introduced to reduce the computational cost of evaluating the convolution integrals that result from the successive approximation procedure.

Key Words. Optimal routing, stochastic shortest path problems, dynamic programming, convolution integrals.

1. Introduction

Shortest path problems have been studied extensively in the fields of computer science, operations research, and transportation engineering.

[1]We are grateful to the colleagues who responded to this work with questions and comments during the Transportation Science Section session on Urban Transportation Planning Models II at the 2002 Meeting of the Institute for Operations Research and Management Science (INFORMS) in San José, California.
[2]Assistant Professor, Department of Civil and Environmental Engineering, University of California, Davis, California.
[3]Professor, Departments of Biomedical, Economics, and Electrical Engineering, University of Southern California, Los Angeles, California.
[4]Professor, Daniel J. Epstein Department of Industrial and System Engineering, Department of Civil and Environmental Engineering, and School of Policy, Planning, and Development, University of Southern California, Los Angeles, California.

Consider a network with N nodes and M arcs (links) connecting these nodes. Given the arc weights, the classical shortest path problem is to determine the path from a given origin to the desired destination with minimum total arc weights. Arc weights can be expressed in different terms depending on the applications. In this paper, we specify the arc weights as the link travel times. However, the model is general and can be extended easily to other types of networks.

Previous studies on shortest path problems include a variety of both deterministic and stochastic applications (see e.g. Refs. 1–7). The deterministic version of the problem is well understood, but how should an optimal path be defined and identified when the link travel times are defined by a probability distribution? Loui (Ref. 8) reports that the standard procedure for addressing the time independent stochastic shortest path problem is to identify the least expected time (LET) path, an approach that is computationally equivalent to the deterministic problem. However, the path identified in this way can be risky. A path with minimum expected travel time can still have a high variance in travel time. Such a path is a suboptimal choice for travelers who are sufficiently risk averse with respect to delay. For example, travelers may want to arrive at their destination as quickly as possible, but also no later than a defined target time.

Consequently, it is not clear how to best extend the deterministic shortest path problem or its variants to the stochastic context. The definition of an optimal path is not obvious in the stochastic context, because it incorporates notions of both expected time and reliability. Frank (Ref. 9) proposed defining the optimal path to be the path that maximizes the probability of realizing a travel time less than a constant k. Sigal et al. (Ref. 10) suggested defining the optimal path to be the path that has the greatest probability of realizing the least travel time. However, neither provided a means of identifying such paths.

In the deterministic problem, the knowledge of the current node and the network topology provide the conditioning information needed to frame the decision of which successor node to visit next. Indeed, this information identifies not just the optimal successor node, but the optimal path to the destination. Introducing the notion of a time budget (i.e., the need to arrive at the destination node at or before a specific time) introduces the opportunity to further condition the choice of successor node on how much time is remaining.

Perhaps, the key difference between the deterministic problem and this version of the stochastic shortest path problem is that, in the stochastic problem, the state resulting from the choice of the successor node is not defined completely until after the link leading to the successor node has been traversed. If the traveler faces a time budget, a random link travel time means that the nature of the traveler's problem changes in ways that are not deterministically foreseeable. Unlike the deterministic problem, an optimal path cannot be identified a priori, because the problem changes with changes in the random time available to the traveler to complete the on-time arrival at the destination. Instead, what is required is a sequence of decisions (choices of successor nodes) that incorporates conditioning information in an optimal way as this information becomes available.

This point of view provides a prototypical example of the stochastic on-time arrival problem (SOTA). Label the nodes in a network as $1, 2, \ldots, N$. Given the known stationary independent probability density function $p_{ij}(t)$ for the link travel times on any link ij and given a traveler current location at node i, what is the next node to visit to maximize the probability of arriving at the destination node N within time t? Arriving at the destination node at any point in time prior to the time t is a success; arriving at any point in time after the time t is a failure. There are no penalties for early arrival. The traveler objective is to pick a successor node that maximizes the probability of success, given his current location and the remaining time budget. Problems of this sort arise routinely when punctuality of arrival or delivery is important. Routing aircraft and freight flows provide good examples, particularly if the commodities are perishable.

We approach the same research question as in the Frank paper (Ref. 9), but from a different viewpoint. The routing problem is treated as a multistage decision process and formulated based on the Bellman principle of optimality. The Picard method of successive approximation is used to solve the nonlinear system of equations involved in our formulation. This model involves the evaluation of convolution integrals at each successive approximation iteration. Relying on the convolution theorem of Laplace transforms reduces the computational cost associated with evaluating these integrals. The results are converted numerically from the transform domain to the time domain by applying generalized inverses to solve linear algebraic equations in a least square sense. Examples and validation tests reported here indicate that this numerical approach of solving the arriving-on-time problems is efficient and reliable.

2. Formulation of the Model

Suppose that a traveler is now at the decision node i. The traveler objective is to select the next node to visit in a way that maximizes the probability of arriving at the destination node N on time or earlier.

Suppose that the traveler chooses to go from node i directly to node j. As noted above, the time left for completing the remaining journey is known only after the traveler arrives at node j. The formulation of the problem to be solved at node j is the same as the formulation at node i, except for the change in the starting node and the reduction in the remaining allowable travel time. Thus, at each new decision node, the allowable remaining time must be updated, and the resulting problem must be solved for an optimal choice of the next node.

The Bellman principle of optimality states that an optimal sequence of decisions has the property that, whatever the initial state and decision are, the remaining decisions must be optimal with respect to the state resulting from the initial decision (Ref. 11).

Define $u_i(t)$ to be the maximum probability of arriving at the destination N from node i within time t or less. This is the optimal return function at node i. If a traveler who is at node i chooses to visit node j the next, the probability that the traveler spends time on the interval $\omega + d\omega$ on link ij is $p_{ij}(\omega)d\omega$ by definition. The remaining time left for the journey from node j is thus $t-\omega$. Based on the Bellman principle, no matter which node j the traveler elects to visit, the traveler must achieve the best value of the return function traveling from that node j to the destination within the remaining time $t-\omega$.

There may be several nodes j that can be visited from the current node i. The traveler should choose the node j that provides the maximum probability of on-time arrival at the destination, beginning the journey from node i. Applying the Bellman principle in the context of the SOTA problem provides the following system of nonlinear, convolution integral equations

$$u_i(t) = \max_{j \neq i} \int_0^t p_{ij}(\omega) u_j(t-\omega) d\omega, \quad i = 1, 2, \ldots, N-1, \tag{1}$$

$$u_N(t) = 1, \tag{2}$$

where $p_{ij}(\omega)d\omega$ is the probability of traversing the direct link ij within the times ω and $\omega + d\omega$ and $u_i(t)$ is the probability that, starting from node i, the traveler arrives at node N by time t when an optimal sequence of choices is made with $i = 1, 2, \ldots, N$ and $0 \leq t < \infty$. The functions $u_i(t), i = 1, 2, \ldots, N$ and $0 \leq t < \infty$, and the sequence of next nodes to visit from each node i are to be determined.

The probability density function $p_{ij}(\omega)$ is presumed to be continuous, but if it were discrete, then Equation (1) would consist of a sum of probabilities $u_j(t-\omega)$ weighted by the discrete probabilities $\mathrm{P}_{ij}(\omega)$, where ω is the set of possible travel times on link ij and t is the time defining on-time arrival at the destination node N. However, if the function $P_{ij}(\omega)$ is continuous, then Equation (1) is not a sum, but a convolution integral with the unknown function $u_j(t)$ in the integrand.

3. Numerical Solution

3.1. Solving a System of Nonlinear Equations: Picard Method of Successive Approximations. How are Equations (1)–(2) to be solved for the unknown probabilities $u_1(t), u_2(t), \ldots, u_N(t)$? How is the optimal choice of the successor nodes identified for each potential origin i? The Picard method of successive approximation is one possible approach to solving this simultaneous system of nonlinear equations. This fixed-point method begins with initial approximations to the solution and then refines these approximations by successive iterations. Since the travel time distribution for each link is known, we begin with the initial, simple approximations

$$u_i^0(t) = \int_0^t p_{iN}(\omega)d\omega, \quad i = 1, 2, \ldots N-1, \quad t \geq 0, \tag{3}$$

$$u_N^0(t) = 1, \quad t \geq 0. \tag{4}$$

These approximations are based on the distributions of travel times over direct links between node i and node N. These first approximations will usually be quite poor because, in a spatially defined network, most of the probabilities identified in Equation (3) are necessarily zero. Most nodes i have no direct link to node N.

Taking the Picard approach, the iterative relationships for successive approximations are

$$u_i^{k+1}(t) = \max_{j \neq i} \int_0^t p_{ij}(\omega) u_j^k(t-\omega)d\omega, \ i = 1, 2, \ldots, N-1, \ 0 \leq t < \infty, \tag{5}$$

$$u_N^{k+1}(t) = 1, \tag{6}$$

where the superscript k is the index of iteration. The function $u_i^k(t)$ has a useful interpretation. It is the probability of arriving on time if optimal choices are made and no more than k intermediate nodes are allowed between an origin node i and the destination. The results can always be improved, or will at least remain unchanged, if the k-intermediate-nodes constraint is relaxed to some degree. Also, because the values $u_i^k(t)$ are probabilities, they are bounded below by 0 and above by 1. Therefore,

$$0 \leq u_i^k(t) \leq u_i^{k+1}(t) \leq 1.$$

Thus, this sequence of ever improving approximations is bounded, monotone, and converges to a limiting value as k increases. There are N nodes in the network, including the origin and destination nodes. Thus, an optimal path can have no more than $N-2$ intermediate nodes in acyclic networks. This means that the maximum number of iterations needed to compute exact probabilities is $N-2$. However, in cyclic networks, the optimal paths may include loops. Therefore, k may exceed $N-2$ for the sequence to converge in cyclic networks. The open question on the convergence of similar problems in the context of infinite-horizon Markovian processes (Refs. 12–13) suggests that the exact complexity of the successive approximation scheme in handling the SOTA problem is still unclear.

3.2. Evaluating the Convolution Integrals: Laplace Transforms. Equation (5) includes a convolution integral that must be evaluated at each successive approximation of $u_i^k(t)$. This requirement is analytically burdensome. Fortunately, there are several methods available for dealing with convolution integrals (Ref. 14). Recall the convolution theorem of Laplace transforms: The transform of a convolution integral is given by the product of the transforms of the functions in the integrand.

This theorem can be applied to reduce greatly the analytical requirements of each iteration. If the Laplace transforms of the function $p_{ij}(t)$ and the function $u_i^k(t)$ are known, then the Laplace transform of the function $u_i^{k+1}(t)$ in Equation (5) can be obtained simply. How do we obtain the Laplace transform of a given function? Given a function in the transform domain, how do we invert it into the time domain?

3.3. Numerical Evaluation of a Finite Integral: Gaussian Quadrature. We begin with the definition of the Laplace transform of a given function $f(t)$,

$$F(s)=L(f(t))=\int_0^{\infty} f(t)e^{-st}dt. \tag{7}$$

The infinite interval in Equation (7) can be reduced to a finite interval between 0 and 1 by substituting τ for e^{-t}. Proceeding, we have

$$t=-\log\tau, \tag{8}$$

$$dt=-d\tau/\tau. \tag{9}$$

Substituting, Equation (7) becomes

$$F(s)=\int_0^1 \tau^{s-1} f(-\log\tau)d\tau. \tag{10}$$

Now, the integral over this finite interval can be approximated by a finite sum. This leads to the relationship

$$F(s) = \sum_{i=1}^{n} \tau_i^{s-1} f(-\log \tau_i) w_i, \tag{11}$$

where $\tau_1, \tau_2, \ldots, \tau_n$ are the points at which $f(-\log \tau)$ is to be evaluated and $w_1, w_2, \ldots, w_n$ are the weights to be attached to these values.

All of the standard quadrature formulas have the form of Equation (11), including the rectangular rule, the trapezoidal rule, and the Simpson rule. Gaussian quadrature is more sophisticated than these equal interval rules. Gaussian quadrature picks τ_i and w_i in such a way that the value of the finite sum in Equation (11) achieves the true value of the integral if the integrand is a polynomial of degree up to $2n-1$ or less (Ref. 15).

As the time available to complete the journey decreases, the probability of arriving on time also decreases for all potential origins i. Thus, the unknown probabilities in this problem are expected to be smooth, monotone functions of t. Any such function is similar to a polynomial of low degree. This limits the number of quadrature points needed for the finite sum in Equation (11) to estimate the integral in Equation (10). Consequently, Gaussian quadrature is employed here to evaluate the convolution integrals over a finite interval. This replaces the analytical requirements with modest computational costs, while simultaneously guaranteeing the quality of the approximation.

3.4. Numerical Inversion of Laplace Transforms: Linear Algebra and Generalized Inverses. The remaining requirement is to obtain a function in the time domain, given the Laplace transform of the function. Suppose that the Laplace transform values $F(s)$ are obtained for a number of discrete value of s. The relationship between a single transform at a value s and the original function in the time t domain is given in Equation (11). The relationships between a set of values $F(s)$ and the values of the original function in the time t domain can be represented by a system of linear algebraic equations. These are

$$F(s) \cong \sum_{i=1}^{n} \tau_i^{s-1} x_i, \quad s = s_1, s_2, \ldots, s_L, \tag{12}$$

where, from Equation (11),

$$x_i = f(-\log \tau_i) w_i, \tag{13}$$

over L observations of s.

The system of equations (12) can be more easily presented in the matrix format

$$F = T\ X, \tag{14}$$

where

$$F = \begin{bmatrix} F(1) \\ F(2) \\ \vdots \\ F(L) \end{bmatrix}_{L\times 1}. \tag{15}$$

The quantity L is the total number of observed points to be evaluated for the function $F(s)$. The number of quadrature points is n. Thus, the matrix T is of the form

$$T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \tau_1 & \tau_2 & \cdots & \tau_n \\ \vdots & \vdots & \vdots & \vdots \\ \tau_1^{L-1} & \tau_2^{L-1} & \cdots & \tau_n^{L-1} \end{bmatrix}_{L\times n}. \tag{16}$$

The matrix T is the well-known, ill-conditioned Vandermonde matrix. The vector X is of the form

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}. \tag{17}$$

The desired values $f(-\log \tau_i)$ can be obtained as

$$f(-\log \tau_i) = x_i / w_i, \quad i = 1, 2, \ldots, n, \tag{18}$$

once the values of $x_1, x_2, \ldots, x_n$ have been determined.

If we have the same number of independent conditions as we have unknown variables in the system of algebraic Equations (14) and if the matrix T is nonsingular, then the unknown vector X can be obtained by the formula

$$X = T^{-1}\ F, \tag{19}$$

where T^{-1} is the inverse of matrix T. However, it will not generally be true that the number of observations $F(s)$ and the number of quadrature points are equal. This is an important point. Since the unknown probabilities, $u_1(t), u_2(t), \ldots, u_N(t)$ are expected to be well represented by polynomials of a low degree, only a small number of quadrature points is needed. Obtaining good-quality results from the numerical inversion of the Laplace transform requires evaluating the Laplace transform at more discrete values of s. Thus, finding X requires generally the solution of a system of algebraic equations with more conditions than unknowns. Fortunately, nonsingularity does not have to hold for this system of algebraic equations to have a solution in the least squares sense. If the matrix T is singular, then the unknown vector X can be obtained by

$$X = T^+ \; F, \tag{20}$$

where T^+ is the Moore-Penrose inverse of the matrix T. Methods to obtain the Moore-Penrose inverse are widely available in the applied mathematics literature. Fan and Kalaba (Ref. 16) give an efficient dynamic programming approach to computing the Moore-Penrose inverse, including a procedure for treating matrices with nearly linearly dependent columns.

The computational cost of evaluating the convolution integrals depends on the number of time points at which the unknown functions $u_i(t)$ are to be evaluated. These time points correspond to the quadrature points in the Gaussian quadrature procedure. The values of these Gaussian quadrature points and their associated weights have already been tabulated and are available in the literature for a wide number of different quadrature points. See for example Bellman and Kalaba (Ref. 15). This reduces the burden of evaluating a convolution integral to that of solving two sets of linear equations

$$Ax = b,$$

where the matrices A remain unchanged throughout the entire successive approximation procedure.

4. Numerical Examples

The accuracy of our algorithms for obtaining the Laplace transform of a function and the transform numerical inversion via solution of a system of linear equations were easily validated by applying the procedure to a few smooth, monotone functions. We proceeded with Laplace transforms that can be obtained analytically so that our numerical results can be compared with exact solutions. For the sake of simplicity, the matrix T was defined to be square of dimension[8]. This permits the inverse of T to be computed as a standard matrix inverse, rather than requiring the calculation of the generalized inverse T^+. Calculations were executed with double precision. Agreement between numerical and analytical values is excellent.

4.1. Small Network. Testing our procedure on a small network validates the basic accuracy of this solution scheme and provides a useful insight into the nature of the results. Consider the 100-node network in Figure 1. This is a fairly representative spatial network in which the nodes are connected to all of their nearest neighbors. Link travel times are drawn from gamma distributions,

$$p(t, n, \alpha) = \alpha^n e^{-\alpha t} t^{n-1} / \Gamma(n), \tag{21}$$

in which Γ(n) is the gamma function of n. Gamma distributions are a flexible class of probability density functions convenient for modeling a wide range of random processes. The mean μ and variance σ^2 of the gamma distribution are

$$\mu = n/\alpha, \tag{22}$$

$$\sigma^2 = n/\alpha^2. \tag{23}$$

The parameters of the gamma distribution for each of the links in the network in Figure 1 are defined using the following rules:

$$n = 0.4 \ \log_{10}(10 + 0.8i + 0.7j), \tag{24}$$

$$\alpha = 2 \ \log_{10}(12 + 1.2i + 0.8j), \tag{25}$$

where i and j are the starting and ending nodes for link ij. This produces values of n on the approximate interval 0.4 to 0.75, values of α on the approximate interval 2 to 4, and link travel time functions of relatively low variance. Use of equations (24)–(25) is a matter of programming convenience. Other parameter ranges are admissible. The number of quadrature points and the number of observations on the Laplace transform $F(s)$ are both taken to be eight so that the matrix T is defined square.

4.2. Numerical Results. Applying this procedure yields the numerical values of the probabilities defined in Equations (1)–(2). The procedure converges quickly. Results for examples of this size remain unchanged after 8 iterations of the Picard method. The probabilities of arriving at the destination node $N = 100$ in time t or less, given an optimal direction of departure from each potential origin, are summarized graphically in Figure 2.

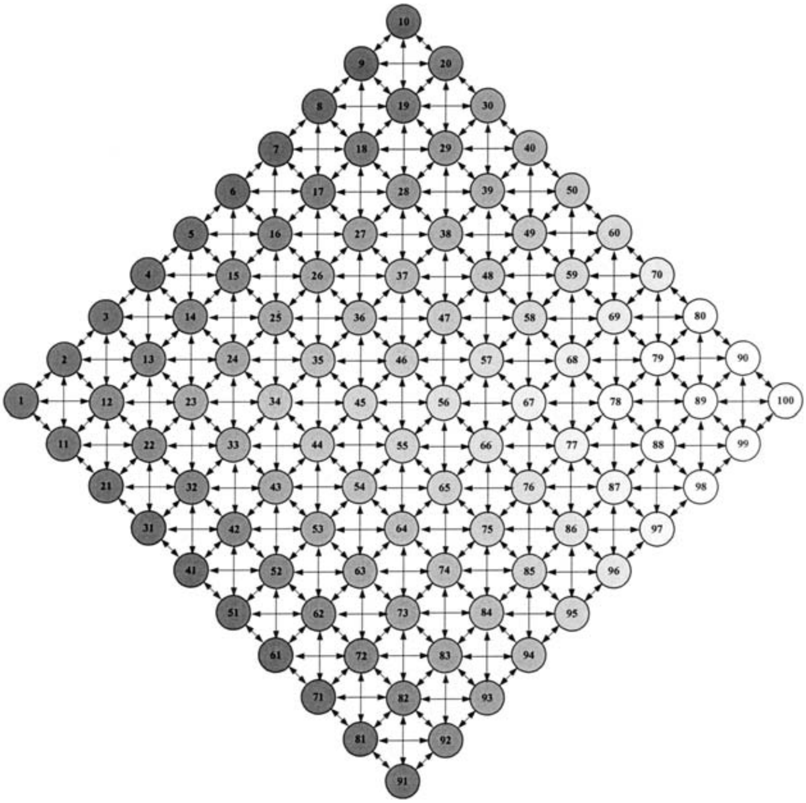

Fig. 1. A 100-node network with gamma-distributed link travel times.

As expected, the probability of arriving at the destination node N on time decreases monotonically for all origins as the time available for completing the trip is reduced. The procedure fails to identify consistently the optimal successor node for the smallest values of t, when the probability of arriving on time is very close to zero for all origins. The computer program implementing the SOTA algorithm is initialized by designating the destination node N as the first estimate of the optimal successor node for all nodes. As the probability of arriving on time approaches zero, the numerical implementation of the procedure can fail by never successfully updating this initial designation. Numerical resolution for very small values of t can be improved by setting the time required for traversing nonexistent arcs to sufficiently large integers when the program is initialized.

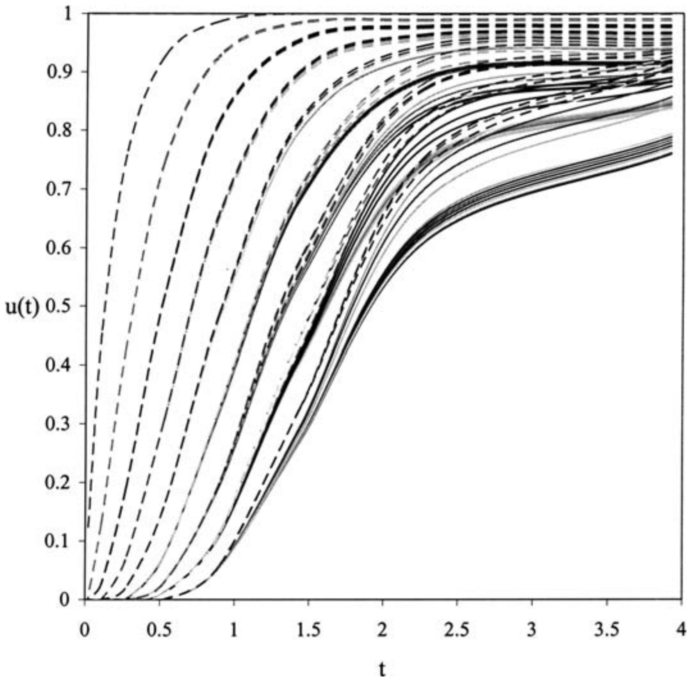

Fig. 2. Maximum probabilities of arriving on time at node 100 in the optimal solution to the stochastic on-time arrival problem, lower link travel time variances.

Given the similarities between the link travel time distributions selected for this example, the larger the minimum number of intervening nodes between the origin and the destination node N, the smaller the corresponding probability of arriving on time for any given value of t. The numerical results bear this out and the 99 curves in Figure 2 cluster predictably into nine sets defined by the minimum number of arcs that must be traversed to arrive at the destination node. Trips originating further away from the destination node have a relatively lower probability of on time arrival.

Figure 3 compares the results from the SOTA and LET solutions. Nodes for which the SOTA and LET solutions are different are shown in gray. Links connecting nodes to optimal successor nodes in the LET solution are shown in black. Links pointing from nodes for which the SOTA solution prescribes a different successor node than does the LET solution for $t_1 = 3.9193$ are shown in gray. Figure 3 is drawn for the maximum value of t from among the eight quadrature points. This is the value for

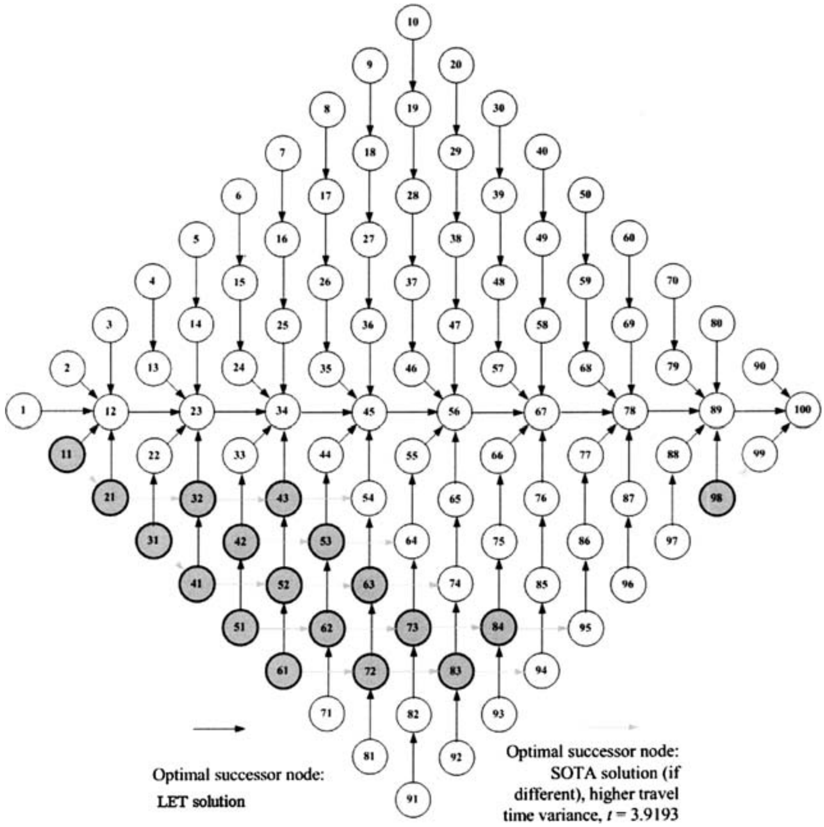

Fig. 3. Comparison of optimal successor nodes for the solutions to the least expected time path and stochastic on-time arrival problem: 100-node network, lower link travel time variances.

which the SOTA and LET solutions diverge to the greatest degree. As the value of t decreases, the gray links in Figure 3 disappear and the SOTA solution approaches the LET solution. Consistent results are provided by numerical experiments with a 100-node network for which the gamma distribution parameters have been selected to produce distributions with the same means as before, but only half the variance of the previous example. As the variance in the link travel times diminishes, the disparity between the SOTA and LET solutions is reduced for all values of t and the SOTA solution converges to the LET solution in all cases.

Increasing the variances in the link travel time distributions causes the SOTA and LET solutions to diverge. Also, this higher level of uncertainty

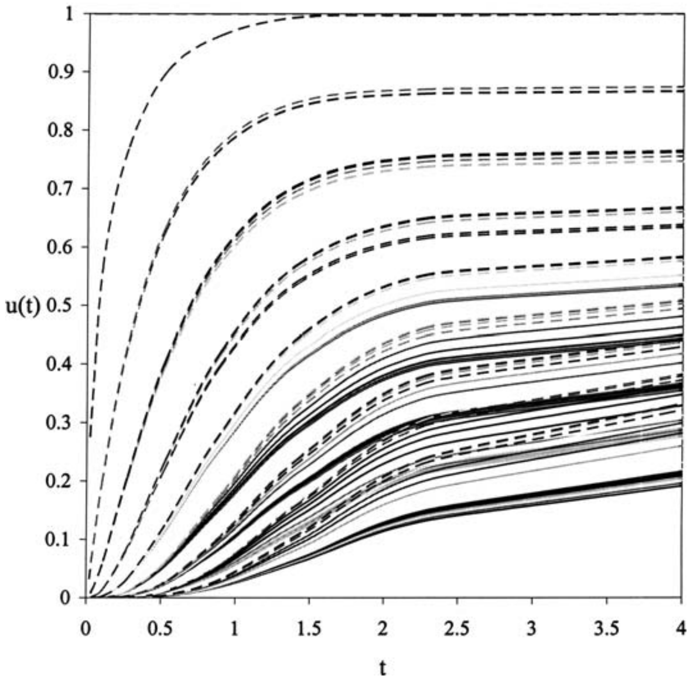

Fig. 4. Maximum probabilities of arriving on time at node 100 in the optimal solution to the stochastic on-time arrival problem, higher link travel time variances.

with respect to travel times results in an overall decrease in the probability of on-time arrival for relatively large time budgets. When the time budget is below a certain threshold, a higher level of uncertainty provides a better chance for on-time arrival than a lower level of uncertainty. The value of this threshold depends on the shapes of these two distributions being compared. Figure 4 gives the probabilities of arriving at the destination node $N=100$ in time t or less given an optimal direction of departure from each origin for a version of the network in which the gamma parameters n and α have been selected to approximately double the variances in the travel times. This effect becomes more pronounced as the variances in the link travel time distributions increase further. Doubling variances again produces consistent results. In this last case, only those trips originating in the immediate vicinity of the destination node have a nontrivial probability of arriving on time.

One important aspect of this approach is that it permits the probabilities $u_i(t)$ to be evaluated without enumerating alternative paths between each origin and the destination. The number of alternative paths in a network is combinatorically large, so any procedure that requires enumeration of paths is too computationally burdensome to be relevant, other than as a conceptual tool. Fortunately, it is not necessary to know the path that produces each optimal $u_i(t)$ and this SOTA procedure does not identify paths.

5. Conclusions and Extensions

This research solves a difficult network problem via applied mathematical techniques. We have shown that stochastic on-time arrival problems can be formulated using the Bellman principle of optimality and solved via the Picard method of successive approximation. Relying on the convolution theorem of Laplace transforms reduces the analytical burden associated with evaluating the convolution integrals that result from this approach. Numerical results in the transform domain are converted back to the time domain by applying generalized inverses to solve linear algebraic equations in the least squares sense. Systematic numerical examples and other validation tests indicate that this numerical approach for solving the stochastic on time arrival is sufficiently efficient and reliable to justify further investigation. The solutions achieved to date are encouraging because they were achieved quickly with desktop computing resources, but a very few results were not computable because of the difficulty of inverting ill-conditioned matrices. Further, several methodological questions remain.

5.1. Numerics. The integer values s at which the Laplace transforms are evaluated were selected arbitrarily. It is possible that the quality of the results may be improved by choosing these values more systematically. Note that the value of s should not be too large. Refer to the definition of Laplace transform in Equation (7). If s becomes large, e^{-st} approaches zero when t is large. In this case, no matter what value of $u(t)$, the product of e^{-st} and $u(t)$ will be close to 0. Therefore, only values of $u(t)$ for t close to 0 contribute to the Laplace transform $F(s)$.

We know from computational experience that increasing the number of quadrature points does not necessarily improve the quality of the results. However, this relationship has not been studied closely and there may be circumstances in which increasing the number of quadrature points improves sufficiently the quality of the results to justify the additional computational expense involved.

Other computational and analytical methods for handling convolution integrals are available. The methods applied here demonstrate the feasibility of this approach and provide a useful indication of performance. However, other approaches may offer advantages and these should be investigated.

5.2. Solution Properties. We evaluate the unknown functions $u_1(t)$, $u_2(t), \ldots, u_N(t)$ describing the maximum probabilities of arriving on time and the optimal choice of the next node to visit for discrete points in time. The uniqueness and continuity of these functions have not been proved here. However, computing can be used as an experimental tool to explore the nature of more general solutions. Our extensive numerical experiments suggest that these functions should be unique and continuous, though perhaps not everywhere differentiable. Therefore, the value of these functions for other times t can be approximated likely via interpolation. However, it is not clear how to identify the optimal successor nodes for the interpolated values of t. In addition, we have shown that an optimal solution to the SOTA formulation exists through successive approximation. However, the complexity of this approximation scheme in handling SOTA problem in general cyclic networks is still an open question. If we decide to stop the successive approximation procedure before it converges, how can we identify systematically the approximation error $|u_i^k(t) - u_i(t)|$ for each given i and t? More analytical and numerical work is required for answering these questions.

5.3. Correlation. The stochastic on-time arrival problem has been formulated based on the assumption that the travel time distributions for any two links are independent. Further research is needed to accommodate the possibility of correlated travel times on adjacent links. Such correlation might be a prominent aspect of large-scale applications, such as rerouting a commercial airline to avoid weather delays and emergency vehicles through networks under natural disasters.

5.4. Network Size. This procedure combines finite and continuous mathematics; it has complexity characteristics that are defined both combinatorially and as a result of the numerical analysis procedures. How the performance of this approach will scale with the size of the problem is unknown. The 100-node examples described here are too small to be relevant to the decision support requirements associated with, for example, centralized and vehicle-based route guidance applications in advanced traveler management information systems (ATMIS). A larger scale application is needed, but whether a much larger network can be accommodated by this approach is unclear.

References

1. Bellman, R. E., *On a Routing Problem*, Quarterly of Applied Mathematics, Vol. 16, pp. 87–90, 1958.
2. Dreyfus, S. E., *An Appraisal of Some Shortest-Path Algorithms*, Operations Research, Vol. 17, pp. 395–412, 1969.
3. Howard, R. A., *Dynamic Probabilistic Systems*, Vol. 2, John Wiley and Sons, New York, NY, 1971.
4. Hall, R. W., *The Fastest Path through a Network with Random Time-Dependent Travel Times*, Transportation Science, Vol. 20, pp. 182–188, 1986.
5. Fu, L., and Rilett, L. R., *Expected Shortest Paths in Dynamic Stochastic Traffic Networks*, Transportation Research, Vol. 32B, pp. 499–516, 1998.
6. Waller, S. T., and Ziliaskopoulos, A. K., *On the Online Shortest Path Problem with Limited Arc Cost Dependencies*, Networks, Vol. 40, pp. 216–227, 2002.
7. Miller-Hooks, E. D., and Mahmassani, H. S., *Least Expected Time Paths in Stochastic, Time-Varying Transportation Networks*, Transportation Science, Vol. 34, pp. 198–215, 2000.
8. Loui, P., *Optimal Paths in Graphs with Stochastic or Multidimentional Weights*, Communications of the ACM, Vol. 26, pp. 670–676, 1983.
9. Frank, H., *Shortest Paths in Probabilistic Graphs*, Operations Research, Vol. 17, pp. 583–599, 1969.
10. Sigal, C. E., Pritsker, A. A. B., and Solberg, J. J., *The Stochastic Shortest Route Problem*, Operations Research, Vol. 28, pp. 1122–1129, 1980.
11. Bellman, R. E., and Kalaba, R. E., *Dynamic Programming and Modern Control Theory*, McGraw-Hill, New York, NY, 1965.
12. White, D. J., *Minimizing a Threshold Probability in Discounted Markov Decision Processes*, Journal of Mathematical Analysis and Applications, Vol. 173, pp. 634–646, 1993.
13. Wu, C. and Lin, Y., *Minimizing Risk Models in Markovian Decision Processes with Policies Depending on Target Values*, Journal of Mathematical Analysis and Applications, Vol. 231, pp. 47–67, 1999.
14. Srivastava, H. M., and Buschman, R. G., *Theory and Applications of Convolution Integral Equations*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1992.
15. Bellman, R. E., and Kalaba, R. E., *Numerical Inversion of the Laplace Transform*, American Elsevier Publishing Company, New York, NY, 1966.
16. Fan, Y. Y., and Kalaba, R. E., *Dynamic Programming and Pseudo-Inverses*, Applied Mathematics and Computation, Vol. 139, 323–342, 2003.