Online Routing with Nonlinear Disutility Functions with Arc Cost Dependencies

Steve Boyles, S. Travis Waller
The University of Texas at Austin
Presented at TRISTAN VI, Phuket, Thailand

June 10-15, 2007

1 Introduction

The importance of routing and shortest path problems in the study of transportation networks is well-known. For instance, these are used to describe user behavior, to find equilibrium states of large-scale networks, and to evaluate the impact of network improvements and broader transportation policies. Thus, the development of efficient, behaviorally realistic algorithms is of paramount interest. Although research in this area dates back over a half-century, continued refinements are being made to account for the specific nature of congested transportation systems.

In particular, researchers have realized that accounting for the fundamentally uncertain day-to-day nature of transportation networks is crucial for correctly modeling travelers' preferences. This operational uncertainty can be traced to a number of phenomena, both on the demand side (such as daily fluctuations in travel patterns), as well as the supply side (such as capacity reductions due to adverse weather or incidents). In turn, the presence of stochasticity affects user behavior in multiple ways. If the travel time (or cost) along a route cannot be predicted absolutely, the assumption that travelers choose the least-cost path is ill-defined. Indeed, it may not even be enough to simply say that travelers choose the path with least *average* cost; rather, depending on the trip type and the traveler's attitudes, one might expect some degree of risk aversion or other, more complicated preferences regarding trip costs. Additionally, uncertain conditions require assumptions to be made about the information or beliefs that travelers have regarding network conditions. The advent of real-time traffic information provision makes this question especially relevant, as travelers can change their chosen path *en route* in response to new information, a phenomenon known as *recourse*.

Within this context, Boyles [2006] developed an algorithm for one specific type of information and cost dependency structure, but a number of alternate structure exist as well. Thus, the intent of this paper is to develop additional algorithms that simultaneously account for general nonlinear user preferences and recourse decisions in the presence of various assumptions on cost structure and information. To simplify the presentation, we will assume that users are only concerned with travel time, although extending these algorithms to account for any additive, nonnegative travel attributes is straightforward. These travel times are both dynamic and stochastic, and exhibit limited forms of dependency.

That link travel times are dependent is natural, although introducing this mathematically leads to additional complications. To simplify, we consider three specific types of dependence and develop a shortest path algorithm for each, using the nomenclature from Waller and Ziliaskopoulos [2002].

1. *Spatial dependence* implies that the probability distribution for any arc is completely determined by the state of the preceding arc, as with a Markov process. Thus, the cost distribution differs according to what the previous arc was, and the experienced cost on it. This can be extended to account for multiple-step history dependence, although this is not undertaken here to keep the notation simple.

2. *Temporal dependence* implies that the cost of any arc is known with complete certainty once its tail node is reached. That is, upon reaching a node, the traveler realizes the costs of all outgoing arcs, and chooses among them accordingly. This structure might represent, for instance, a variable message sign indicating the current state of nearby arcs.

3. *Temporal-spatial dependence* implies both of the above; namely, that upon reaching a node, travelers learn the cost of all outgoing arcs. This knowledge then provides additional information about arcs one step further downstream.

In Section 2, we discuss previous research efforts related to this topic, before presenting our modeling approach in earnest in Section 3. This is followed by algorithms for each of the three types of dependency in Section 4. When applying these algorithms in cyclic networks that exhibit non-first-in-first-out (non-FIFO) arc cost structures, a technical issue arises; this is addressed in Section 5. Finally, Section 6 concludes the paper and summarizes the key contributions.

2 Literature Review

As mentioned in the previous section, the primary contribution of this work is to account for both nonlinear traveler preferences and recourse decisions in a single algorithmic framework, in the presence of certain types of arc cost dependency.

Nonlinear preferences essentially seek to quantify how travelers value reliability, or specific risk attitudes regarding travel time. Reliability has been modeled in a number of different ways. For instance, Sivakumar and Batta [1994] solve a shortest path problem that constrains the variance of the solution, while Sen et al. [2001] use a multiobjective routing approach when arc costs are normally distributed. Robust formulations, such as that in Yu and Yang [1998] or Montemanni and Gambardella [2004], solve a minimax shortest path problem when arc costs are only known to lie within a given interval, with no further information on their distribution. Fan et al. [2005], on the other hand, find a routing policy that minimizes the probability of arriving at the destination later than a specified arrival time. Another approach involving a desired arrival time is found in Gao [2005], where a weighted sum of expected arrival time before and after this target is minimized.

Other authors develop models incorporating non-linear preferences. Loui [1983] and Eiger et al. [1985] develop procedures for linear and exponential utility functions which allow an efficient dynamic programming-like solution, while Murthy and Sarkar [1996] present an algorithm for decreasing quadratic utility functions. Gabriel and Bernstein [2000] provide a heuristic method for finding the non-additive shortest path, and Tsaggouris and Zaroliagis [2004] present such an algorithm for monotone and convex disutility functions.

However, these models assume that disutility increases with travel time. This may not always be true; for instance, arriving much earlier than expected might cause the traveler to feel as though time has been wasted, as departure could have been postponed while still allowing on-time arrival. Indeed, Redmond and Mokhtarian [2001] indicate that a limited amount of commuting time may actually have positive utility: commuters find value in time spent alone while traveling, to transition between the work and home environments. In such cases, an extremely short commute is in fact less desirable than one of moderate length.

Other researchers have studied shortest path problems that allow users to update their path choice *en route* in response to learning information about network conditions. Waller and Ziliaskopoulos [2002] present an algorithm for networks with limited temporal and spatial dependency, and Provan [2003] develops a Dijkstra-like procedure for a more general dependency structure with nonnegative arc costs. Gao and Chabini [2006] also present algorithms for general dependency. This case is more difficult; in fact, Polychronopoulos and Tsitsiklis [1996] and Provan [2003] show that this problem is NP-complete. When arc costs are independent, however, Miller-Hooks [2001] develops a polynomial algorithm for online routing in stochastic, time-dependent networks.

Although considerable literature exists concerning the value of travel time reliability, and concerning recourse decisions, relatively little work combines the two into a more general treatment of uncertainty in routing problems. Gao [2005] presents a heuristic to find a policy with minimum variance, and an exact algorithm for minimizing the weighted sum of three linear penalties for expected travel time and expected arrival before and after a target arrival time. Boyles [2006] provides an algorithm to find optimal online routing policies with piecewise polynomial disutility functions, but only for one particular cost structure (limited spatial dependence). The intent of this paper is to provide similar algorithms for additional types of arc dependency.

3 Model Development

Table 1 lists the notation used in this paper, which we define in this section. Let $G = (N, A, \Pi)$ be a probabilistic directed network with finite node and arc sets N and A of cardinality n and m, respectively. Π is a set of probability distributions for arc costs, described below in greater detail. Let the origin and destination nodes be denoted O and D, respectively. Assume that a directed path from each node to D exists. For each arc $a = (a_1, a_2) \in A$ there is an associated finite set of $S(a,t)$ integer travel times, which depend on the current time t; denote this set $C_{a,t} = \{c^1_{a,t}, c^2_{a,t}, \dots, c^{S(a,t)}_{a,t}\}$. Let M be the highest travel time among all arcs. For each node i, the forward star $FS(i)$ and the reverse star $RS(i)$ respectively denote the set of incoming and outgoing arcs, and $\Gamma^{-1}(i)$ represents the set of predecessor nodes.

The form of the arc cost probability distributions Π varies according to the dependency structure as mentioned in Section 1; however, they can all be described by an expression of the form $p(b, s, t, \theta)$, which indicates the probability that arc b is in state s at time t. θ indicates any information relevant to the particular dependency structure of the problem. For the case of spatial dependency, the state r of the predecessor arc a is needed to specify the probability distributions; thus $\theta = (a, r)$ and the probabilities for arc b are given by $p(b, s, t, a, r)$. For temporal dependency, the costs of all outgoing arcs are known once the traveler arrives at the tail node, but the costs of arcs are assumed to be independent; effectively, there is no useful information ($\theta = \varnothing$) and $p(b, s, t)$ suffices to describe arc cost probabilities. Finally, as temporal-spatial dependency incorporates elements of both structures, the information is again given by $\theta = (a, r)$ and the probabilities by $p(b, s, t, a, r)$, but these describe arcs that are *two* steps downstream, rather than the adjacent arc (one step downstream) which is known with certainty once its tail node is reached.

The general approach for all of these cases is modeled on the classical label correcting all-to-one shortest path algorithm (see, for instance, Ahuja et al. [1993]). One begins at the destination, where trip length, and thus trip disutility, is known with absolute certainty. Then, by proceeding backwards through the network, one can inductively determine strategies and expected disutilities until these are known for all nodes.

The same fundamental formula is used in order to perform these calculations regardless of the dependency structure. Specifically, one needs to know how to calculate the expected disutility that results from choosing to travel on a particular arc, assuming that all relevant information is known downstream. This is done by storing a number of labels. In particular, the dynamic and dependent nature of the network requires that separate labels be stored at each node i, at each possible arrival time at that node t, and for each possible information set θ for that node at that time. These form a *node-time-information combination* or NTIC $\phi = (i, t, \theta)$, which are the fundamental points at which the traveler makes decisions. Further, for each NTIC, a label needs to be stored representing the expected disutility. However, to calculate this disutility, more information is needed.

If the polynomial disutility function is of degree d (for now assume it only has one piece), the expected disutility can be calculated if one knows the first d moments of total travel time X:

$$E[P(X)] = E\left[\sum_{i=0}^{d} a_i X^i\right] = \sum_{i=0}^{d} a_i E[X^i]$$

To accomplish this, we can store d labels at each NTIC, one for each of the necessary moments. Now, consider the general case when the traveler is at a specific NTIC (i, t, θ), and needs to calculate the expected disutility obtained by choosing a particular arc $b = (i, q)$. The total travel time can be divided into three parts: the time elapsed since departing (t); the time incurred when traveling on arc b (denote this by C);

Table 1: Notation used in this paper.

a_{ji}	the coefficient on the term x^i for the j-th segment of P
$c^s_{a,t}$	cost of arc a in state s at time t
m	number of arcs
n	number of nodes
$p(b, s, t, \theta)$	probability that arc b is in state s at time t according to information θ.
$p_{\omega,\pi}$	probability that realization ω occurs when policy π is used
s	index for arc states
t	current time
A	set of arcs
A_j	the intervals defining each segment of P
$C_{a,t}$	set of possible costs for arc a at time t
C	a random variable indicating the cost of the next arc
D	destination node
$FS(i)$	the set of outgoing arcs from node i
$L(i, t, \theta, j, n)$	label for n-th moment of remaining travel time from node i at time t, given information θ and arrival during interval A_j
M	largest possible arc cost in the network
N	set of nodes
O	origin node
P	the piecewise polynomial disutility function
R	random variable for travel time remaining after the next node
$RS(i)$	the set of incoming arcs at node i
$S(a, t)$	the number of possible states of arc a at time t
$T(i)$	the set of possible arrival times at node i
T	latest possible arrival time
X	random variable indicating total travel time
X_ω	the travel time associated with realization ω
ϕ	a node-time-information combination
π	routing policy for spatial dependency, a function from Φ to A
θ	current information, depending on dependency structure
$\rho(i, t, \theta, j)$	probability of arriving during interval A_j when at node i at time t with information θ.
ω	a particular realization of the network
$\Gamma^{-1}(i)$	the set of predecessor nodes for node i
Φ	the set of all node-time-information combinations
Ξ	a constant such that $P(x)$ is increasing for $x > \Xi$
Ω	the set of all network realizations
$\varnothing$	null arc used to assign probabilities when a trip begins

and the remaining travel time R between the head node of b and the destination. For spatial dependency, C and R are both random variables; for temporal and temporal-spatial dependency, R is random but C is not. For all of these cases, the binomial theorem allows us to express expected disutility as the sum of specific moments which can then be stored as labels:

$$E[(t+C+R)^n] = \sum_{s=1}^{S(b,t)} \sum_{k=0}^{n} \sum_{k'=0}^{k} \binom{n}{k} \binom{k}{k'} p(b,s,t,\theta)(C_{a,s}^t)^{n-k} t^{k-k'} E[R^k|\iota]$$

where ι is the information attained after traveling on arc b ((b,s) for spatial and temporal-spatial dependency, $\varnothing$ for temporal). Also note that for temporal dependency, the state of b is known exactly, so $p(b,s,t,\theta)$ is zero for all states except for the one revealed. Since we assume that $E[R^k]$ is known for all downstream NTICs, this formula is enough to calculate all of the travel time moments at i, and therefore to calculate the expected disutility obtained from choosing arc b.

When the piecewise disutility function consists of multiple segments, additional labels are needed to store the probability that one arrives during each segment, and the travel time moments are replaced by *conditional* moments for each of these. If we let ρ_j represent the probability of arriving during time interval A_j, and again assume that all labels for downstream nodes have already been calculated, the extended set of formulas needed to calculate the labels at a particular node are

$$\rho(i,t,\theta,j) = \sum_{s=1}^{S} p(b,s,t,\theta)\rho(q, t+C_{a,s}^t, \iota, j)$$

$$E[(C+R)^n|X \in A_j] = \frac{1}{\rho(i,t,\theta,j)} \sum_{s=1}^{S(b,t)} \sum_{k=0}^{n} \binom{n}{k} p(b,s,t,\theta)\rho(q,t+C_{a,s}^t,\iota,j)(C_{a,s}^t)^{n-k} E[R^k|\iota, X \in A_j]$$

$$E[P(X)] \sum_{j=1}^{N} \sum_{n=0}^{d(j)} \sum_{k=0}^{n} \binom{n}{k} a_{ji}\rho(i,t,\theta,j) t^{n-k} E[(C+R)^k | X \in A_j]$$

The dependency structure affects the manner in which the optimal routing policy is defined. For spatial dependence, a policy necessarily prescribes an arc that the traveler should take whenever such a choice is available; in this context, a routing policy π is a function mapping Φ to A; that is, prescribing an arc choice for each fundamental decision point in the problem. Thus, for this case, the difficult issue is efficiently determining these arc choices in such a way that the expected travel disutility is minimal.

The issue is slightly different for the temporal and temporal-spatial dependency cases, because the cost of all outgoing arcs are known with exact certainty. Thus, if the moments of remaining travel time for all downstream nodes are known, one can easily calculate the moments of remaining travel time from the current node for each possible arc choice, and the best arc can be chosen. Thus, learning the costs of downstream arcs *a priori* obviates the need to develop and store a pre-defined routing policy. Rather, the routing policy is developed on the fly based on the revealed arc costs and the travel time moments which were calculated ahead of time. Again following the style of an all-to-one shortest path algorithm, one begins at the destination (where the remaining travel time moments are trivially known), and proceeds backwards through the network, inductively calculating the travel time moments at all other nodes and times, but without setting path pointers.

However, naive calculation leads to an exponential number of steps in the worst case. To circumvent this, Waller and Ziliaskopoulos [2002] develop a procedure that allows efficient calculation of expected remaining travel time. To see why this is needed, consider a node which has three outgoing arcs leading to nodes a, b, and c, each of which can exist in one of two states (c_{hi} and c_{lo}) with equal probability, as shown in Figure 1. Since this is temporal dependency, the traveler has just learned the costs of these three arcs, and must choose one of them accordingly. Let σ_{hi}^a indicate the strategy of choosing arc a when it has a 'high' cost, and $P(\sigma_{hi}^a)$ the corresponding expected disutility (which is assumed known); the same notation is adopted

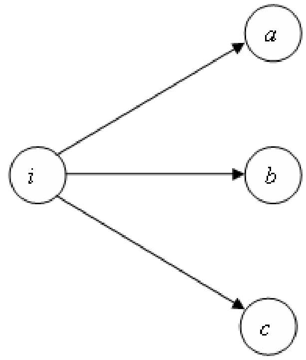

Figure 1: Each of the three arcs will have one of two possible costs.

for all possible choices of arc and state. Since there are $2^3 = 8$ possible combinations of states for the three arcs, one might conclude that there are eight possible costs, as seen in the probability matrix

$$\begin{array}{c} 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \\ 1/8 \end{array} \begin{bmatrix} \min\{P(\sigma^a_{hi}), P(\sigma^b_{hi}), P(\sigma^c_{hi}) \\ \min\{P(\sigma^a_{hi}), P(\sigma^b_{hi}), P(\sigma^c_{lo}) \\ \min\{P(\sigma^a_{hi}), P(\sigma^b_{lo}), P(\sigma^c_{hi}) \\ \min\{P(\sigma^a_{hi}), P(\sigma^b_{lo}), P(\sigma^c_{lo}) \\ \min\{P(\sigma^a_{lo}), P(\sigma^b_{hi}), P(\sigma^c_{hi}) \\ \min\{P(\sigma^a_{lo}), P(\sigma^b_{hi}), P(\sigma^c_{lo}) \\ \min\{P(\sigma^a_{lo}), P(\sigma^b_{lo}), P(\sigma^c_{hi}) \\ \min\{P(\sigma^a_{lo}), P(\sigma^b_{lo}), P(\sigma^c_{lo}) \end{bmatrix}$$

However, regardless of the values of these disutilities, there are at most six unique values that can occur in this matrix: $P(\sigma^a_{hi})$,$P(\sigma^b_{hi})$,$P(\sigma^c_{hi})$,$P(\sigma^a_{lo})$,$P(\sigma^b_{lo})$, and $P(\sigma^c_{lo})$. Thus, when these are realized, this matrix may be reduced to six elements while preserving all relevant information by adding the probabilities of rows that are combined. In general, this process reduces the number of possible scenarios from exponential ($O(S^n)$) to polynomial($O(Sn)$). This is important for developing an efficient algorithm which must calculate the expected value by summing the product of the probability of each scenario with its corresponding disutility.

4 Algorithms

Based on the above discussion, we now present three algorithms (SD-MEDR, TD-MEDR, and TSD-MEDR) to solve the online routing problem with nonlinear disutility functions for the spatial, temporal, and temporal-spatial dependency cases, respectively. The procedure REDUCE called by TD-MEDR and TSD-MEDR is listed as Algorithm 4. The SEARCH(a, b) function called by REDUCE performs a search on vector a for value b and returns its location (if found), and the first available location otherwise. The correctness of these algorithms should be apparent from the previous section; the reader desiring greater rigor is referred to Boyles [2006] which contains a detailed proof of correctness for SD-MEDR (named MPPR-MC in that work). Similar proofs for TD-MEDR and TSD-MEDR are omitted here for reasons of brevity.

As mentioned above, SD-MEDR calculates conditional expected disutility labels for each NTIC, while TD-MEDR and TSD-MEDR operate by calculating a single expected disutility value for each node and arrival time. To find this single value for the latter, each possible combination of outgoing arc states is examined and REDUCEd to produce a polynomial-sized vector of choices and resulting disutilities, depending on the realized arc states. The expectation over these probabilities is then taken to calculate the expected disutility.

Boyles [2006] demonstrates that SD-MEDR has complexity $O(NMn^3mD^2S^2)$ in acyclic networks, where N, M, n, m, D, and S are the number of segments in the disutility function, the greatest arc cost, the number of nodes, the number of arcs, the largest degree of any segment of the disutility function, and the maximum number of states an arc can exist in, thus showing that this algorithm has pseudopolynomial complexity.

For TD-MEDR, additional time is needed at each iteration to combine and REDUCE the probability matrices. As discussed in the previous section, the size of this matrix after considering k successor nodes is bounded by kS. At worst, when considering each successor node, a matrix of this size is combined with S additional states for the new arc, leading to nS^2 operations of REDUCE. An implementation of REDUCE as described in Waller and Ziliaskopoulos [2002] can perform this step in $O(nS^2 \log(nS))$. Thus, the overall complexity of TD-MEDR is $O(NMn^3mD^2S^2 \log(nS))$.

A similar argument for TSD-MEDR yields a complexity of $O(NMn^4mD^2S^3 \log(nS))$ for that algorithm in acyclic networks.

Algorithm 1 SD-MEDR

{Initialization}
for all a, s, t, j, n such that $a \in RS(D)$, $s \in S(a)$, $t \in T(d)$, $j \in \{1, \ldots, N\}$, $n \in \{1, \ldots, d_j\}$ **do**
  $L(D, t, a, s, j, n) \leftarrow 0$
  $L(D, t, a, s, j, 0) \leftarrow 1$
  if $t \in A_j$ **then**
    $\rho(D, t, a, s, j) \leftarrow 1$
  else
    $\rho(D, t, a, s, j) \leftarrow 0$
  end if
end for
for all i, t, a, s, j, n such that $i \in N - D$, $t \in T(j)$, $a \in RS(i)$, $s \in S(a)$, $j \in \{1, \ldots, N\}$, $n \in \{1, \ldots, d_j\}$ **do**
  $L(i, t, a, s, j, n) \leftarrow \infty$
end for
$SEL \leftarrow \{i : i \in \Gamma^{-1}(D)\}$
{Iteration}
while $SEL \neq \varnothing$ **do**
  Select $i \in SEL$ and set $SEL \leftarrow SEL - i$
  for all t, a, s, b such that $(i, t, a, s) \in \Phi$, $b = (i, q) \in FS(i)$ **do**
    for all $j \in \{1, \ldots, N\}$, $n \in \{1, \ldots, d_j\}$ **do**
      $\rho_b(i, t, a, s, j) \leftarrow \sum_{s'=1}^{S(b,t)} p(b, s', t, a, s)\rho(q, t + c_{a,s}^t, b, s', j)$
      $L_b(i, t, a, s, j, n) \leftarrow \sum_{s'=1}^{S(b,t)} \sum_{k=0}^{n} \binom{n}{k} \frac{p(b,s',t+c_{a,s}^t,a,s)}{\rho_b(i,t,a,s,j)} \rho(q, t + c_{a,s}^t, b, s', j)(c_{a,s}^t)^{n-k} L(q, t + c_{a,s}^t, b, s', j, k)$
    end for
  end for
  $P_b(i, t, a, s) \leftarrow \sum_{j=1}^{N} \sum_{n=0}^{d} \sum_{k=0}^{n} \binom{n}{k} a_{ji}\rho_b(i, t, a, s, j)t^{n-k} L_b(i, t, a, s, j, k)$
  if $P_b(i, t, a, s) < P(i, t, a, s)$ **then**
    for all $j \in \{1, \ldots, N\}$, $n \in \{1, \ldots, d_j\}$ **do**
      $L(i, t, a, s, j, n) \leftarrow L_b(i, t, a, s, j, n)$
      $\rho(i, t, a, s, j) \leftarrow \rho_b(i, t, a, s, j)$
      $P(i, t, a, s) \leftarrow P_b(i, t, a, s)$
      $\pi(i, t, a, s) \leftarrow b$
      $SEL \leftarrow SEL \cup h : h \in \Gamma^{-1}(i)$
    end for
  end if
end while

Algorithm 2 TD-MEDR

for all t, j, n such that $t \in T(d)$, $j \in \{1, \ldots, N\}$, $n \in \{1, \ldots, d_j\}$ **do**
    $L(D, t, j, n) \leftarrow 0$
    $L(D, t, j, 0) \leftarrow 1$
    if $t \in A_j$ **then**
        $\rho(D, t, j) \leftarrow 1$
    else
        $\rho(D, t, j) \leftarrow 0$
    end if
end for
for all i, t, j, n such that $i \in N - D$, $t \in T(j)$, $j \in \{1, \ldots, N\}$, $n \in \{1, \ldots, d_j\}$ **do**
    $L(i, t, j, n) \leftarrow \infty$
end for
$SEL \leftarrow \{i : i \in \Gamma^{-1}(D)\}$
while $SEL \neq \varnothing$ **do**
    Select $i \in SEL$ and set $SEL \leftarrow SEL - i$
    $\mu \leftarrow [\infty]$
    $\nu \leftarrow [1]$
    for all $t \in T(i)$ **do**
        for all b such that $b = (i, q) \in FS(i)$ **do**
            $temp\mu \leftarrow [\varnothing]$
            $temp\nu \leftarrow [\varnothing]$
            $r \leftarrow 1$
            for $s = 1$ to $S(b, t)$ **do**
                for $l = 1$ to $|\mu|$ **do**
                    $temp\nu_r \leftarrow \nu_r p(b, s, t)$
                    for all $j \in \{1, \ldots, N\}$, $n \in \{1, \ldots, d_j\}$ **do**
                        $\rho_r(i, t, j) \leftarrow \rho(q, t + c^t_{a,s}, j)$
                        $L_r(i, t, j, k) \leftarrow \sum_{k=0}^{n} \binom{n}{k} \frac{1}{\rho_b(i,t,j)} \rho(q, t + c_s, j)(c_s)^{n-k} L(q, t + c_s, j, k)$
                    end for
                    $P_r \leftarrow \sum_{j=1}^{N} \sum_{n=0}^{d} \sum_{k=0}^{n} \binom{n}{k} a_{ji} \rho_r(i, t, j) t^{n-k} L_r(i, t, j, k)$
                    $temp\mu_r \leftarrow \min\{P_r, \mu_l\}$
                    $r \leftarrow r + 1$
                end for
            end for
            $[\mu, \nu] = \textsc{Reduce}(temp\mu, temp\nu)$
        end for
        $tempP = \sum_{l=1}^{|\mu|} \mu_k \nu_k$
        if $tempP < P(i, t)$ **then**
            $P(i, t) \leftarrow tempP$
            $SEL \leftarrow SEL \cup h : h \in \Gamma^{-1}(i)$
        end if
    end for
end while

Algorithm 3 TSD-MEDR

for all a, s, t, j, n such that $a \in RS(D)$, $s \in S(a)$, $t \in T(d)$, $j \in \{1, \dots, N\}$, $n \in \{1, \dots, d_j\}$ **do**
  $L(D, t, a, s, j, n) \leftarrow 0$
  $L(D, t, a, s, j, 0) \leftarrow 1$
  if $t \in A_j$ **then**
    $\rho(D, t, a, s, j) \leftarrow 1$
  else
    $\rho(D, t, a, s, j) \leftarrow 0$
  end if
end for
for all i, t, a, s, j, n such that $i \in N - D$, $t \in T(j)$, $a \in RS(i)$, $s \in S(a)$, $j \in \{1, \dots, N\}$, $n \in \{1, \dots, d_j\}$ **do**
  $L(i, t, a, s, j, n) \leftarrow \infty$
end for
$SEL \leftarrow \{i : i \in \Gamma^{-1}(D)\}$
while $SEL \neq \varnothing$ **do**
  Select $i \in SEL$ and set $SEL \leftarrow SEL - i$
  for all $t \in T(i), a \in RS(i), s \in S(a, t)$ **do**
    $\mu \leftarrow [\infty]$
    $\nu \leftarrow [1]$
    for all $b = (i, q) \in FS(i)$ **do**
      $temp\mu \leftarrow [\varnothing]$
      $temp\nu \leftarrow [\varnothing]$
      $r \leftarrow 1$
      for $s = 1$ to $S(b, t)$ **do**
        for $l = 1$ to $|\mu|$ **do**
          $temp\nu_r \leftarrow \nu_r p(b, s', t, a, s)$
          for all $j \in \{1, \dots, N\}$, $n \in \{1, \dots, d_j\}$ **do**
            $\rho_r(i, t, a, s, j) \leftarrow \sum_{s'=1}^{S(b,t)} p(b, s', t, a, s)\rho(q, t + c_{a,s}^t, b, s', j)$
            $L_r(i, t, a, s, j, n) \leftarrow \sum_{s'=1}^{S(b,t)} \sum_{k=0}^{n} \binom{n}{k} \frac{p(b, s', t + c_{a,s}^t, a, s)}{\rho_b(i, t, a, s, j)} \rho(q, t + c_{a,s}^t, b, s', j)(c_{a,s}^t)^{n-k} L(q, t + c_{a,s}^t, b, s', j, k)$
          end for
          $P_r(i, t, a, s) \leftarrow \sum_{j=1}^{N} \sum_{n=0}^{d} \sum_{k=0}^{n} \binom{n}{k} a_{ji} \rho_b(i, t, a, s, j) t^{n-k} L_b(i, t, a, s, j, k)$
          $temp\mu_r \leftarrow \min\{P_r, \mu_l\}$
          $r \leftarrow r + 1$
        end for
      end for
      $[\mu, \nu] = \textsc{Reduce}(temp\mu, temp\nu)$
      $tempP = \sum_{l=1}^{|\mu|} \mu_k \nu_k$
    end for
    if $tempP < P(i, t, a, s)$ **then**
      $P(i, t, a, s) \leftarrow tempP$
      $SEL \leftarrow SEL \cup \{h : a = (h, i)\}$
    end if
  end for
end while

Algorithm 4 $\{\mu, \nu\} = \textsc{Reduce}(temp\mu, temp\nu)$

Require: A vector of disutilities $temp\mu$ and a vector of probabilities $temp\nu$.

$\mu \leftarrow [\varnothing]$
$\nu \leftarrow [\varnothing]$
for $k = 1$ to $|temp\mu|$ **do**
  $h = \textsc{Search}(\mu, temp\mu_k)$
  $\mu_h = temp\mu_k$
  $\nu_h = nu_h + temp\nu_h$
end for

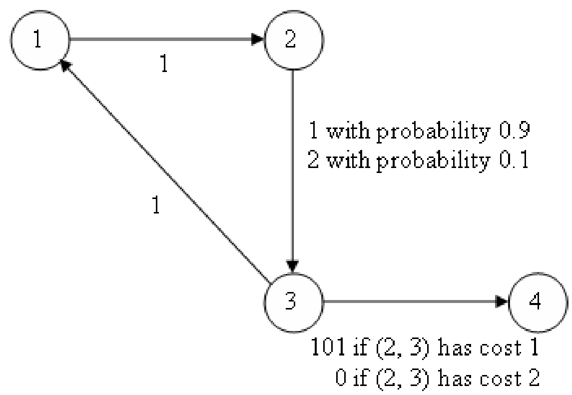

Figure 2: Infinite cycling in an online shortest path.

5 Cyclic Networks with Non-FIFO arcs

The presence of cycles can cause difficulties in online algorithms: for instance, in online shortest path problems, one cannot rule out paths that include cycles if there are arcs which have non-FIFO travel times, that is, when a traveler can arrive earlier by leaving an arc later. For instance, the optimal policy for the network in Figure 2, based on Waller and Ziliaskopoulos [2002], contains the possibility of cycling arbitrarily many times. In this graph, after observing the state of arc (2, 3) the state of (3, 4) is known. When arriving at node 3, if arc (2, 3) had cost 1, the traveler knows that the cost of arc (3, 4) is very high; thus, the optimal strategy is to travel back to node 1, hoping that after another traversal, the state of (3, 4) will be lower. However, at the second traversal, there is still no guarantee that (3, 4) will have low cost; and the cycle could be traversed a third time, and so on. Thus, there is no upper bound on the maximum number of cycles that will be present in the optimal solution, although the probability of a large number of traversals of a cycle is small. Although non-FIFO arcs may not be common in transportation networks, one example where such behavior might appear is at a toll or customs plaza when a new service lane opens. If lane changing is restricted, newly arriving travelers using this new lane can overtake earlier-arriving vehicles that are still waiting in a long queue.

Both Waller and Ziliaskopoulos [2002] and Boyles [2006] develop probabilistic bounds to allow one to find an approximately optimal solution in finite time, even if the optimal routing policy involves inifinitely many cycles. The bound in Waller and Ziliaskopoulos [2002] is based on the error arising from only applying through a finite number of passes of the scan eligible list SEL, while the bound in Boyles [2006] is based on constraining trips to arrive before a certain time, and bounding the error resulting from this additional constraint. The latter is more flexible, since it can be applied regardless of the particular polynomial disutility function chosen; in fact, for a linear disutility function (for which this problem reduces to the online shortest path problem of Waller and Ziliaskopoulos [2002]), the two bounds are asymptotically equivalent. In this section, the bound from Boyles [2006] is briefly restated in terms of the broader dependency structures introduced in this paper.

To derive this bound, we assume that the disutility function P is strictly increasing when arrival times are sufficiently large (let Ξ denote some time for which $x_1 > x_2 > \Xi \Rightarrow P(x_1) > P(x_2)$). Now, we are interested in the error that is induced by constraining all trips to finish before some pre-announced time T, which ensures a finite set of NTICs and, therefore, a finite running time for the algorithm. To assist with this, we define a *network realization* ω to be a subset of NTICs representing a particular path from the origin to the destination, with Ω the set of all network realizations and $p_{\omega,\pi}$ the proability that the path a particular network realization ω represents occurs when routing policy π is followed.

This constraint forces a slight change in the definition of routing policies for the temporal and temporal-spatial dependent cases. Previously, there was no pre-determined policy, and arc choices were made in real time as these costs were made known. However, this constraint may require overriding this default behavior, as trips longer than T are strictly prohibited. One way to accomplish this is to set any disutility labels corresponding to a time greater than T to infinity. Alternately, one may prescribe a policy like that for spatial dependence for selected NTIC's to ensure arrival before T.

Since the destination can be reached from any node in at most Mn time, and since P is increasing for $x > \Xi$, if all arcs are FIFO, clearly $\Xi + Mn$ is an upper bound on arrival time. For non-FIFO arcs, the case is not as simple, as the example in Figure 2 demonstrated. We start by deriving a bound on the probability that the total travel time X is greater than any time $\tau > \Xi + Mn$.

Lemma 1. *For $\tau > \Xi + Mn$, the probability that the traveler completes a trip after time τ when following an optimal (unconstrained) policy π^* is at most $P(\Xi + Mn)/P(\tau)$.*

Proof: Consider a policy in which the traveler deterministically moves from the origin to any node that is part of a cycle, traverses this cycle repeatedly until the current time is greater than Ξ, and then any acyclic path is followed to the destination. Thus the traveler arrives at the destination no later than $\Xi + Mn$; therefore, the disutility from this policy is at most $P(\Xi+Mn)$, which forms an upper bound on the disutility for the optimal policy. Thus

$$\sum_{\omega: X^\omega > \tau} p_{\omega,\pi^*} P(\tau) \leq \sum_{\omega: X^\omega > \tau} p_{\omega,\pi^*} P(X^\omega) \leq \sum_{\omega \in \Omega} p_{\omega,\pi^*} P(X^\omega) \leq P(\Xi + Mn)$$

$$\sum_{\omega: X^\omega > \tau} p_{\omega,\pi^*} \leq \frac{P(\Xi + Mn)}{P(\tau)} \quad \blacksquare$$

The next step is to bound the error created by forcing trips to end by a given time $T > \Xi + Mn$. To do this, modify the optimal policy for all NTICs whose time component is greater than $T - Mn$ so that the traveler will deterministically arrive before T. Again, the disutility from this modified policy forms an upper bound on the disutility of the optimal policy.

Theorem 1. *For $T > \Xi + Mn$, the difference between the disutility of an optimal policy, and that of an optimal policy which guarantees arrival before T, is at most $P(\Xi + Mn)\left[\frac{P(T)}{P(T-Mn)} - 1\right]$.*

Proof: Consider the constrained policy π_T described above. Since it is identical to the optimal unconstrained policy for network realizations in which the travel time is less than $T - Mn$, all of the difference between the expected disutilities of these two policies is derived from trips ending after this time:

$$P(\pi_T) - P(\pi^*) \leq \sum_{\omega: X^\omega > T - Mn} p_{\omega,\pi^*}[P(T) - P(X^\omega)] \leq \sum_{\omega: X^\omega \in [T-Mn, T]} p_{\omega,\pi^*}[P(T) - P(X^\omega)]$$

the latter because $P(T) - P(X^\omega) < 0$ for $X^\omega > T$. Continuing,

$$P(\pi_T) - P(\pi^*) \leq \sum_{\omega: X^\omega \in [T-Mn, T]} p_{\omega,\pi^*}[P(T) - P(X^\omega)] \leq \frac{P(\Xi + Mn)}{P(T - Mn)}[P(T) - P(T - Mn)]$$

by Lemma 1 and because $P(T-Mn) \leq P(X^\omega)$. Rearranging the last result, and knowing that the expected disutility of the optimal constrained policy π_T^* is no greater than $P(\pi_T)$, we also have $P(\Xi + Mn)\left[\frac{P(T)}{P(T-Mn)} - 1\right]$. ■

Using l'Hôpital's Rule, one can show that this last bound approaches zero as $T \to \infty$. Although this last result gives the error induced by adding the time constraint as a function of T, in practice it is more useful to find T as a function of the maximum allowable error or tolerance ϵ. Solving the last bound for T analytically is difficult (and, in fact, impossible in general if P has degree five or greater), although numerical solution is possible. However, the asymptotic properties of the inverse function $T(\epsilon)$ can be studied.

Lemma 2. *$T(\epsilon)$ is $O\left((Mn)^{d(N)+1}/\epsilon\right)$ as $M, n \to \infty$ and $\epsilon \to 0$.*

Proof: Define $\epsilon(T)$ to be the maximum possible error associated with restricting trips to end before T (that is, the bound from Theorem 1. Expanding the polynomials P and rearranging yields

$$\epsilon(T) = P(\Xi + Mn)\left[\frac{a_{N,d(N)}(T^{d(N)-1}(Mn) - T^{d(N)-2}(Mn)^2 + T^{d(N)-3}(Mn)^3 - + \cdots) + \cdots}{a_{N,d(N)}(T-Mn)^{d(N)} + \cdots}\right]$$

which is the product of three terms, which are $O((Mn)^d(N))$, $O(MnT^{d(N)-1})$, and $O((T^d(N))^{-1})$, respectively; so $\epsilon(T)$ is $O((Mn)^{d(N)+1}/T)$. Therefore T is $O((Mn)^{d(N)+1}/\epsilon)$. ■

It bears repeating that the bounds in this section are only needed if the network is both acyclic and non-FIFO. In typical transportation networks, the latter property is almost never present in a macroscopic sense; in such cases, these bounds need not be used.

6 Conclusion

Accounting for the impacts of uncertain day-to-day network conditions is important for modeling route choice, performing traffic assignment, and other common transportation problems. Uncertainty manifests itself in multiple ways — in particular, it results in nonlinear "risk-averse" behavior, and allows the possibility of online response to travel information receieved *en route*. Most previous research focuses on only one of these aspects at a time; past work that considers both simultaneously only accounts for one specific type of arc cost dependency (spatial dependence). This paper contributes similar, pseudopolynomial algorithms for temporal and temporal-spatial dependency, in line with the structure established in Waller and Ziliaskopoulos [2002].

Although these algorithms can account for a wide range of user preferences (any piecewise polynomial disutility function), the question of which disutility functions are most applicable to particular trip and user types is still open. For instance, one would imagine that travelers embarking on a shopping trip, a home-to-work commute, and a delivery drop-off have distinctly different attitudes regarding uncertain travel times. Both numerical analysis and econometric techniques need to be applied to yield satisfactory answers to this question. Additionally, the task of describing dependent probability distributions for every link in a network is tedious and difficult. While loop detectors can provide such data for freeway links, estimating travel time distributions on arterials is somewhat harder, and automated estimation procedures are needed to make these algorithms practical for deployment.

References

R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows*. Prentice-Hall, Englewood, NJ, 1993.

S. Boyles. Reliable routing with recourse in stochastic, time-dependent transportation networks. Master's thesis, The University of Texas at Austin, 2006.

A. Eiger, P. B. Mirchandani, and H. Soroush. Path preferences and optimal paths in probabilistic networks. *Transportation Science*, 19(1):75–84, 1985.

Y. Fan, R. Kalaba, and J. Moore. Arriving on time. *Journal of Optimization Theory and Applications*, 127 (3):497–513, 2005.

S. A. Gabriel and D. Bernstein. Nonadditive shortest paths: subproblems in multi-agent competitive network models. *Computational and Mathematical Organization Theory*, 6(1):29–45, 2000.

S. Gao. *Optimal adaptive routing and traffic assignment in stochastic time-dependent networks*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2005.

S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research B*, 40(2):93–122, 2006.

R. P. Loui. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26(9):670–676, 1983.

E. D. Miller-Hooks. Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1):35–52, 2001.

R. Montemanni and L. Gambardella. An exact algorithm for the robust shortest path problem with interval data. *Computers and Operations Research*, 31(10):1667–1680, 2004.

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

G. H. Polychronopoulos and J. N. Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27 (2):133–143, 1996.

J. S. Provan. A polynomial-time algorithm to find shortest paths with recourse. *Networks*, 41(2):115–125, 2003.

L. Redmond and P. Mokhtarian. The positive utility of the commute: modeling ideal commute time and relative desired commute amount. *Transportation*, 28(2):179–205, 2001.

S. Sen, R. Pillai, S. Joshi, and A. K. Rathi. A mean-variance model for route guidance in advanced traveler information systems. *Transportation Science*, 35(1):37–49, 2001.

R. A. Sivakumar and R. Batta. The variance-constrained shortest path problem. *Transportation Science*, 28(4):309–316, 1994.

G. Tsaggouris and C. Zaroliagis. Non-additive shortest paths. In S. Albers and T. Radzhik, editors, *Proceedings of the 12th Annual European Symposium on Algorithms*, 2004.

S. Travis Waller and A. K. Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227, 2002.

G. Yu and J. Yang. On the robust shortest path problem. *Computers and Operations Research*, 25(6): 457–468, 1998.