A Polynomial-Time Algorithm to Find Shortest Paths with Recourse

J. Scott Provan
Department of Operations Research
University of North Carolina
Chapel Hill, NC 27599-3180

December 2, 2002

Abstract

The Shortest Path with Recourse Problem involves finding the shortest expected-length paths in a directed network each of whose arcs have stochastic traversal lengths (or delays) that become known only upon arrival at the tail of that arc. The traveler starts at a given source node, and makes routing decisions at each node in such a way that the expected distance to a given sink node is minimized. We develop an extension of Dijkstra's algorithm to solve the version of the problem where arclengths are nonnegative and reset after each arc traversal. All known no-reset versions of the problem are NP-hard. We make a partial extension to the case where negative arclengths are present.

Keywords: stochastic network; shortest path; recourse; Dijkstra

1 Introduction

The *stochastic shortest path problem* has as input a directed network $G = (N, A)$ with node set N and arc set A of cardinalities n and m, respectively. Each arc (i, j) of G has length L_{ij} which is a random variable taking on r_{ij} finite values $l_{ij}^1 < \ldots < l_{ij}^{r_{ij}}$. The traveler starts at a source node s, and is interested in reaching a destination node t using a path having minimum expected length.

There are two popular versions of the stochastic shortest path problem that differ in the amount of information that a traveler has about the arclengths as the network is traversed. In the *expected shortest path problem*, the traveler knows *all* arclengths in the network before starting the traversal, and hence always chooses the deterministic shortest (s, t)-path for that particular realization of the network. This problem is surveyed in [4], Section 6. Although determination of the path knowing each particular realization is a simple shortest path computation, actually computing the expected pathlengths over all realizations is NP-hard even in the case where the arcs take on only 0-1 values. In the *shortest expected path problem*, the traveler is aware of *no* arclengths while traversing the network, and thus must take the predetermined (s, t)-path that has shortest expected length. This problem involves simply finding the shortest (s, t)-path in G using expected arclengths for each arc, and the expected value here is just the expected length of this path.

The *shortest path with recourse (SPR) problem* provides an intermediate version that is more realistic in routing situations where only local traffic information is available. In this version the traveler is aware of the length of an arc only upon arrival at the tail of that arc. Choices are made at each node of the path based on knowledge of both the overall stochastic structure of the network and the immediate realizations of the lengths of the arcs pointing away from the node currently visited, in such a way as to reach the destination in the minimum expected time. We call the corresponding traversal a *recourse path.*

There are two ways the various SPR models differ in terms of the stochastic structure of the network. The first is in the dependence of the individual arclength values.

Arc-independent arclengths: Associated with each arclength l_{ij}^k is a probability p_{ij}^k. Arc (i,j) independently takes random length L_{ij}, with probabilities

$$P[L_{ij} = l_{ij}^k] = p_{ij}^k,\ k = 1, \ldots, r_{ij}.$$

Node-independent arclengths: Associated with each node i is a set of r_i arclength vectors for the forward star arcs of that node: $\boldsymbol{l}_i^k = (l_{ij}^k :\ (i,j) \in A)$ and probabilities p_i^k, $k = 1, \ldots, r_i$. Each node i independently gives the arcs pointing out of it a set of arclength realizations $\boldsymbol{L} = (L_{ij} :\ (i,j) \in A)$ with probabilities

$$P[L_{ij} = l_{ij}^k, (i,j) \in A] = p_i^k,\ k = 1, \ldots, r_i.$$

Dependent arclengths: There is a collection of r arclength vectors for the entire network $\boldsymbol{l}^k = (l_{ij}^k :\ (i,j) \in A)$ and probabilities p_k, $k = 1, \ldots, r$, and the network takes on random arclengths $\boldsymbol{L} = (L_{ij} :\ (i,j) \in A)$ with probabilities

$$P[L_{ij} = l_{ij}^k, (i,j) \in A] = p_k,\ k = 1, \ldots, r.$$

Markov arclengths: The arclengths are associated with a Markov process $\mathcal{M}$ denoted by states $k = 1, \ldots, r$ and an $r \times r$ transition probability matrix M. Each state k is associated with a set $\boldsymbol{l}^k = (l_{ij}^k :\ (i,j) \in A)$ of arclength realizations for that state. The process $\mathcal{M}$ starts in some initial state σ_0 — usually determined from the steady-state probabilities of the network — and after each arc traversal the system moves from state σ_i to σ_{i+1} according to the transition probabilities given by M. One can likewise define this process for node-independent or arc-independent models, by giving independent Markov processes $\mathcal{M}_i$ or $\mathcal{M}_{ij}$ on each node i or arc (i, j) of the network, respectively.

The other way SPR problems are distinguished is by whether the arclengths are static or change as the network is traversed.

No reset: The arclengths remain fixed throughout the traversal, that is, once the arclengths at a node are known they will look the same upon further revisits to that node.

Reset: The arclengths take on new independent (and unknown) values from the same distribution each time an arc is traversed.

We make several remarks at this point:

- With the reset assumption on the stochastic network behavior the dependent and node-independent models become identical. Thus we will consider only the arc- and node-independent models when we analyze the reset version of the problem.

- The Markov chain model has properties of both the reset and no-reset models. While there is clearly an arclength reset occurring, it is a state-dependent reset. As we shall see below, the complexity classification of Markov models fits more appropriately into the no-reset class of SPR problems. An interesting exception will be analyzed in Section 5.

- Technically, a recourse "path" may well have repeated nodes and arcs. Take the example given in Figure 1. Arcs $(1, 2)$, $(2, 3)$, and $(3, 1)$ have deterministic length 1 and arc $(2, 4)$ has length 0 with probability $\frac{1}{2}$ and 10 with probability $\frac{1}{2}$. Then the optimal strategy is to take the cycle $(1, 2)$, $(2, 3)$, $(3, 1)$ as long as arc $(2, 4)$ has the value 10 and $(2, 4)$ when it has the value 0, for an expected length of 4 for an (s, t)-recourse path. In this case the path could repeat the cycle $(1, 2)$, $(2, 3)$, $(3, 1)$ any number of times. Paths can repeat arcs in the no-reset versions of the problem as well, although the number of times this is done in an optimal recourse path will be bounded by the number of arcs in the network (since the repeat traversal of any arc is useless unless at least one new arclength is discovered in the interim).

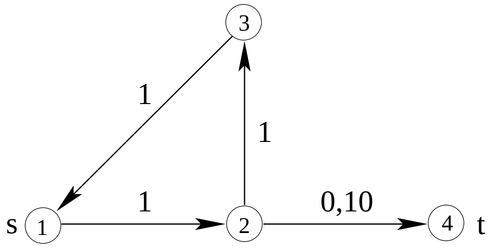

Figure 1: An SPR solution with cycles

There have been a number of studies of recourse path problems, using a wide variety of models. Croucher [3] was one of the first to consider such a problem. In his model the network has randomly failing weighted arcs, and at each node one specifies a predetermined arc to be traversed unless it fails, in which case one of the remaining arcs is traversed at random. He gives a polynomial-time algorithm when the network is acyclic and arcs fail independently. Andreatta and Romeo [1] also consider an arc-failure model, but when an arc fails a fixed recourse path to t is traversed. They give a polynomial algorithm for general networks when arcs fail independently. Both of these problems can be reduced to the appropriate model given here, and the complexity classifications will be the same as those for these models. Cheung [2] considers the general arc-independent reset model, and gives label-correcting iterative approximation methods for solving the problem. Polychronopoulos and Tsitsiklis [8] consider both the dependent and arc-independent models in a no-reset context, giving complexity results, non-polynomial algorithms, and heuristics. Psaraftis and Tsitsiklis [9] study the node-independent Markov model, giving a polynomial-time algorithm for acyclic networks with loops — or equivalently, waiting states — at nodes. Hall [6] and Miller-Hooks [7] consider slightly different models, where traversal times and routing decisions are based on time-of-arrival at a node, and stochastic behavior of an arc is known only upon traversal of that arc. We will not consider their models in this paper.

We next turn to the intractability results for SPR. (The reader is referred to [5] for definitions of NP, NP-hardness and NP-completeness.) It turns out that the primary model assumption that makes the SPR problem NP-hard is the no-reset assumption. Polychronopoulos and Tsitsiklis [8] give a reduction for the dependent model, for which the NP-hardness result applies to the arc- and node-independent models as well. The reduction is from Directed Hamiltonian Path. Briefly, we start with directed network $G = (N, A)$ for which it is desired to find a directed path going through all nodes of G. We attach a new source node s to each node of G, and in turn attach each node of G to a new sink node t. The arcs to t are given stochastic lengths 0 or M for some large number M (e.g., greater than n) with one-half probability each, and every other arc is given deterministic length 1. Then it is clear that the optimal recourse path from s to t is one that is prepared to visit every node of G so as to maximize the probability of finding an arc to t with length 0. That is, the optimal strategy is to follow a Hamiltonian path if one exists.

Note, by the way, that even in the reset model, if we insist that the optimal solution is a bona fide *path* — that is, no repeated vertices are allowed — then the above construction establishes the NP-hardness of this model as well. We also observe the interesting fact that the node- and arc-independent no-reset models are not known to be in NP, for it is not clear that there is a polynomial-length description of the optimal policies in either case.

Many of the papers mentioned above give algorithms for recourse problems that rely on the underlying network G being *acyclic*. For arc- and node-independent models, the reset and no-reset versions of SPR are essentially equivalent in the context of acyclic networks. Thus the solution methods given in this paper also work for no-reset models on acyclic networks with arc- and node-independence.

It is tempting from this discussion to assume that the acyclic property is what makes SPR easier. We give a new complexity result to show that this is not true in general. Define SPRADBN to be the SPR problem whose input is an acyclic network with dependent binary (0-1) arclengths having no reset, and for which we wish to determine whether there exists a recourse path with length 0.

Lemma 1 SPRADBN *is NP-complete.*

Proof That SPRADBN is in NP was proved in [8], Theorem 3. To prove the NP-hardness we use a reduction from 3SAT. Input for 3SAT is a set $\mathcal{E} = \{C_1, \ldots, C_r\}$ of clauses in the Boolean variables $x_1, \ldots, x_l$ with each clause C_i consisting of variables y_{i1}, y_{i2}, y_{i3}, where each y_{ij} is either x_w or $\bar{x}_w$ (= complement of x_w) for some variable x_w. The expression $\mathcal{E}$ is satisfiable if there exists a true-false assignment to the variables $x_1, \ldots, x_l$ that results in each clause having at least one true term.

We construct the network G and the stochastic model as follows: The nodes of G are u_i, v_i, and $\bar{v}_i$ for $i = 1, \ldots, l$, and additional node t. We set $s = u_1$, and associate each v_i and $\bar{v}_i$ with the symbol x_i and $\bar{x}_i$. The arcs of G are (u_i, v_i), $(u_i, \bar{v}_i)$, (v_i, t), and $(\bar{v}_i, t)$ for $i = 1, \ldots, l$, and (v_i, u_{i+1}), and $(\bar{v}_i, u_{i+1})$ for $i = 1, \ldots, l-1$. Figure 2 gives the construction. There are r states, with state i giving the arcs (v_w, t) or $(\bar{v}_w, t)$ the value 0 corresponding to each x_w or $\bar{x}_w$ in clause C_i, giving all other arcs adjacent to t the value 1, and giving all arcs not adjacent to t the value 0. Each state is realized with the same probability $1/r$.

Now any path from s to t must alternately go through u_i and one of v_i or $\bar{v}_i$ until at some point it takes its final arc from some v_w or $\bar{v}_w$ to t. This path will have length 0 if and only if the network state corresponds to a clause C_i having as one of its members a variable x_w or $\bar{x}_w$ corresponding to this final arc. It follows that the only way to guarantee that the path will eventually go through a node v_i or $\bar{v}_i$ having this property is to choose a sequence of v_i or $\bar{v}_i$ corresponding to a satisfying assignment for $\mathcal{E}$. Thus the

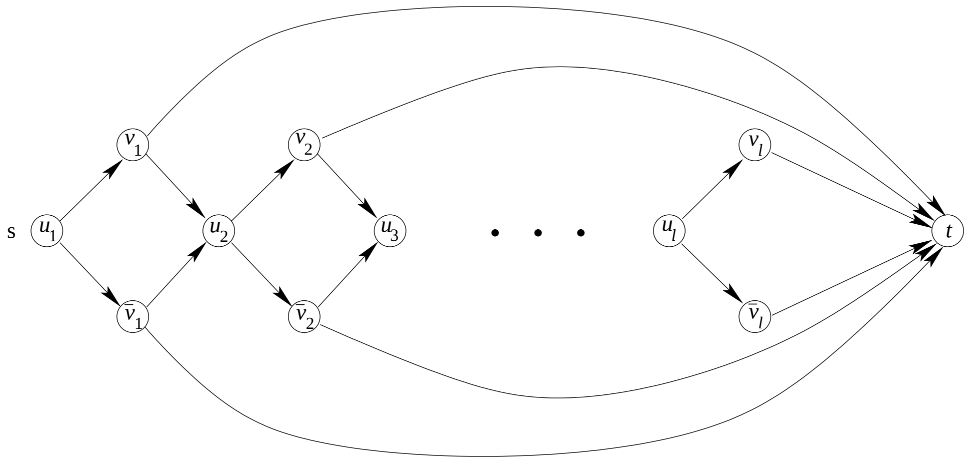

Figure 2: The reduction from 3SAT

problem of determining whether G has a recourse path of expected length 0 is NP-complete. □

All the above SPR problems using Markov-chain models turn out to be NP-hard as well. We can use the same reductions as we gave for the non-Markov models: simply construct the Markov chain that has two states (for the arc- and node-independent reduction given above) or r states (for the dependent reduction given above) with equal but very small probability of moving from state to state, and large probabilities of remaining in the same state. Then the Markov chain — using the equilibrium probabilities as the initial probabilities — essentially simulates the associated non-Markov no-reset model, and so the reductions work here as well.

In Section 3 we show that if the arclength realizations l_{ij}^k are all nonnegative, and reset occurs, then the SPR problem has a polynomial algorithm. This algorithm is based on the primal-dual method of solving deterministic shortest path problems, and yields a Dijkstra-like algorithm to find SPRs. Table 1 now summarizes the complexity status of the problems of SPR for the different combinations of dependence, reset, and acyclicness. In Section 4 we discuss a partial extension of our algorithm to networks with negative arclengths, and in Section 5 we show how a modification of the Psaraftis and Tsitsiklis Markov model can be solved using the method given in this paper.

	no reset	reset
arc ind.	P^3	P^1
node ind.	P^3	P^1
depend.	NPC^5	P^4

a. Acyclic networks

	no reset	reset
arc ind.	NPH^6	P^2
node ind.	NPH^6	P^2
depend.	NPC^6	P^4

b. Networks with cycles

P=polynomial, NPC=NP-complete, NPH=NP-hard
1=Section 2, 2=Section 3 (nonnegative weights only)
3=reduces to reset problem, 4=reduces to node-independent case
5=Lemma 1, 6=proved in [8]

Table 1: Complexity results for SPR

2 A linear programming formulation for the SPR problem

Let network G together with nonnegative arclength realizations l_{ij}^k and associated probabilities p_{ij}^k or p_k (for the independent and dependent models, respectively) be given and suppose we want to find the SPR from node s to node t. We assume that there exist paths from every node to t, since we can delete any irrelevant parts of the network. For each node i we use $\mathcal{A}(i) = \{j : (i,j) \in A\}$ to denote the *forward star* of i, and $\mathcal{B}(i) = \{j : (j,i) \in A\}$ to denote the *backward star* of i. We begin by giving conditions under which an optimal solution can be found for SPR. They generalize the Bellman equations for the deterministic shortest path problem.

Lemma 2 *Let G have arclength realizations satisfying the property that for every directed cycle C, $\sum_{(i,j)\in C} l_{ij}^1 > 0$, and let v_i, $i \in N$, be an assignment of real numbers to the nodes of G. Then the v_i represent the expected cost of an optimal recourse path from i to t for every i if and only if they satisfy the following set of equations:*

$$v_t = 0 \tag{1}$$

$$v_i = E\left[\min_{j\in\mathcal{A}(i)} \{L_{ij} + v_j\}\right] \quad i \neq t. \tag{2}$$

Further, for any set of values satisfying (1) and (2), the optimal recourse path can be found using the traversal strategy that consists of choosing, at each node $i \neq t$, any arc (i, j) for which $L_{ij} + v_j$ is the minimum value indicated by (2).

Proof That the set of expected recourse-path values satisfy (1) and (2) follows immediately from the definition. Conversely, suppose we have a set of v_i that satisfy (1) and (2). Note that $v_i < \infty$, since any path from i to t will have finite expected length, which in turns bounds v_i above. Consider any strategy chosen according to the second part of the lemma. This is a standard optimizing strategy for the associated Markov decision process, with states corresponding to the pair (i, R), $i \in N$ and R is a realization of arclengths in $\mathcal{A}(i)$, decisions corresponding to the associated arc to traverse, and the penalties and transition probabilities corresponding to the arclengths and distribution of the network. Further, any solution to (1) and (2) results in values v_i that represent the minimum expected loss from this strategy when starting from node i (see, for example, [10], Chapter III, Theorem 1.2). It only remains to show that this strategy actually results in a recourse path to t, that is, t is reached with probability 1. Suppose not. Then there is some positive probability that the path will reach some node i from which the strategy will never leave a subset S of nodes not containing t. But this means that, starting at node i, the strategy will, with probability 1, pass through an infinite number of cycles, each of which by assumption will have positive weight. Thus v_i will have value ∞, a contradiction. □

Lemma 2 gives us a compact way of presenting the complete set of optimal strategies, and so the optimal policy can be described in polynomial space. To illustrate the lemma, consider the example in Figure 3. Each of the arcs has two states with arclength realizations as marked, and each of the arclengths is realized independently with probability $\frac{1}{2}$. The numbers at the nodes are the values of v_i in the solution to (1) and (2). The optimal path can be read off from these numbers as follows:

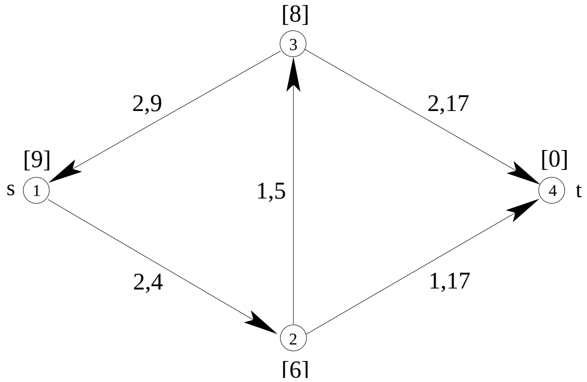

Figure 3: An SPR example

At node 1: Choose arc (1,2) under any realization.

At node 2: Given realizations L_{23} and L_{24} for arcs (2,3) and (2,4), choose the arc that has the smaller of the values $L_{23} + 8$ and L_{24}.

At node 3: Given realizations L_{31} and L_{34} for arcs (3,1) and (3,4), choose the arc that has the smaller of the values $L_{31} + 9$ and L_{34}.

It will be instructive for later discussion to give an explicit procedure, based on an idea by Cheung [2], for computing v_i using Equation (2). To do this, let (i, j) be an arc and let $\boldsymbol{k} = (k_{iw} : (i, w) \in A)$ be a set of state indices on the arcs of the forward star of i, with $0 \leq k_{iw} < r_{iw}$, $w \neq j$, and $0 < k_{ij} \leq r_{ij}$. Define the event

$$\mathcal{E}_{ij}(\boldsymbol{k}) = [\mathbf{L} : L_{ij} = l_{ij}^{k_{ij}},\ L_{iw} \geq l_{iw}^{k_{iw}+1}, w \in \mathcal{A}(i) \setminus \{j\}]. \tag{3}$$

In the independent case the probability of this event is

$$P[\mathcal{E}_{ij}(\boldsymbol{k})] = p_{ij}^{k_{ij}} \prod_{w \in \mathcal{A}(i) \setminus \{j\}} \sum_{\kappa=k_{iw}+1}^{r_{iw}} p_{iw}^{\kappa}$$

and in the dependent case it is

$$P[\mathcal{E}_{ij}(\boldsymbol{k})] = \sum_{r\,:\,l^r\,\in\,\mathcal{E}_{ij}(\boldsymbol{k})} p_r.$$

We can now define the procedure to compute the right-hand-sides in (2).

```
**Procedure** BELLMAN($\hat{v}, i$)

    **Input:** Node values $\hat{v}_j$, $j \in N$, and node $i \neq t$

    **Output:** $V = E\left[\min_{j \in \mathcal{A}(i)} \{L_{ij} + \hat{v}_j\}\right]$

    **Procedure:** Let $M_1 \leq \ldots \leq M_a$ be an ordering of the values
        $\{l_{ij}^{\kappa} + \hat{v}_j : \; j \in \mathcal{A}(i), \; \kappa = 1, \ldots, r_{ij}\}$ in nondecreasing order, where
        $a = \sum_{j \in \mathcal{A}(i)} r_{ij}$. Note that the arclength values of an individual arc
        will appear in increasing order in this sequence.

        Set $V = 0$ and initialize index vector $\boldsymbol{k} = (k_{ij} : j \in \mathcal{A}(i)) \equiv 0$.

        *for* $\xi = 1, \ldots, a$ *do*

            Let $l_{ij}^{\kappa}$ be the arclength associated with the value $M_\xi$. Set
            $k_{ij} = \kappa$ and define
                $\rho_{ij}^{\kappa} = P[\mathcal{E}_{ij}(\boldsymbol{k})]. \qquad (4)$

            Add $\rho_{ij}^{\kappa} M_\xi$ to $V$.
```

To summarize: For each arc (i, j) and each index $\kappa = 1, \ldots, r_{ij}$, the number ρ_{ij}^{κ} represents the probability that arc (i, j), with length l_{ij}^{κ}, will be chosen as the arc from which we leave node i. This happens precisely when the associated value $l_{ij}^{\kappa} + \hat{v}_j$ is minimum among the values $L_{iw} + \hat{v}_w$, $w \in \mathcal{A}(i)$, and the associated event $\mathcal{E}_{ij}(\boldsymbol{k})$ describes the set of realizations for which this happens. Procedure BELLMAN simply allocates each realization to the appropriate event $P[\mathcal{E}_{ij}(\boldsymbol{k})]$ in nondecreasing order of the expected length $l_{ij}^{\kappa} + \hat{v}_j$ of the traversed path from i, with the components of $\boldsymbol{k}$ keeping track of the last index for which the arc (i, j) had the minimum value $l_{ij}^{k_{ij}} + \hat{v}_j$ in this ordering.

As an example of the procedure, consider the evaluation of Bellman$(\hat{v}, 2)$, where the $\hat{v}$ are as given in Figure 3. The M_ξ's are ordered 1+0,1+8,5+8,17+0. Applying Bellman$(\hat{v}, i)$ produces the following set of $\boldsymbol{k}$, ρ^κ_{ij}, and cumulative values of V:

iteration	k_{23}	k_{24}	**arc**	ρ^κ_{ij}	V
1	0	1	$(2,4)$	$\frac{1}{2}$	$\frac{1}{2}(1+0) = \frac{1}{2}$
2	1	1	$(2,3)$	$\frac{1}{4}$	$\frac{1}{2} + \frac{1}{4}(1+8) = \frac{11}{4}$
3	2	1	$(2,3)$	$\frac{1}{4}$	$\frac{11}{4} + \frac{1}{4}(5+8) = 6$
4	2	2	$(2,4)$	0	6

Most SPR papers have used formulae analogous to (1) and (2) for finding optimal recourse paths. Bellman's Procedure will actually yield a polynomial-time solution for SPR — with the single exception of the dependent no-reset version — whenever G is *acyclic*, since then (1) and (2) can be solved in reverse topological order. Bellman's Procedure can also be used to *verify the correctness* of a set of expected distance values, as this just involves checking if the value computed by the procedure matches the actual value. It does not immediately give an efficient method for *finding* these values in non-acyclic networks, since the values in a cycle cannot be unambiguously computed. We can, however, use the constructs of Bellman's Procedure to build a Dijkstra-like algorithm for solving SPR on networks with cycles.

Henceforth we will deal exclusively with the reset model. In this context, we can give a linear programming formulation for SPR motivated by the characterization in Lemma 2. Let $\mathcal{R}$ be the set of realizations of arclengths for G, and for each state $R \in \mathcal{R}$ let p_R be the probability of state R being realized. For node i, let v_i represent the expected distance of an optimal recourse path from i to t, and for node i and state $R \in \mathcal{R}$, let u_i^R represent the part of the expectation in (2) corresponding to seeing state R upon reaching node i. Consider the following linear program for finding the v_i and

u_i^R values:

$$
SPRLP: \quad \begin{array}{rcll}
\max\ z & = & v_s & \\
v_t & = & 0 & \\
v_i & = & \displaystyle\sum_{R\in\mathcal{R}} u_i^R p_R & i \in N \setminus \{t\} \\
u_i^R & \leq & l_{ij}^R + v_j & (i,j) \in A,\ i \neq t,\ R \in \mathcal{R}
\end{array}
$$

where l_{ij}^R is the length of arc (i,j) under realization R. We will in fact find a solution to $SPRLP$ that simultaneously maximizes v_i for *every* $i \neq t$, and therefore satisfies (1) and (2).

$SPRLP$ has an exponential number of both variables and constraints, so that it cannot be solved directly as an LP — even using delayed column or constraint generation — without requiring exponential storage. We can, however, solve $SPRLP$ by a Dijkstra-like algorithm based on the primal-dual method of solving LPs.

The dual to $SPRLP$ has variables α_i, $i \in N$, corresponding to the expectation equations, and x_{ij}^R, $(i,j) \in A$, $R \in \mathcal{R}$, corresponding to the inequalities on the u_i^R.

$$
SPRD: \quad \begin{array}{rcll}
\min\ w & = & \displaystyle\sum_{(i,j)\in A}\sum_{R\in\mathcal{R}} l_{ij}^R x_{ij}^R & \\
\alpha_j & = & \displaystyle\sum_{i\in\mathcal{B}(j)}\sum_{R\in\mathcal{R}} x_{ij}^R \ + & \left\{\begin{array}{ll} 1 & j = s \\ 0 & j \in N \setminus \{s,t\} \end{array}\right. \\
p_R\alpha_i & = & \displaystyle\sum_{j\in\mathcal{A}(i)} x_{ij}^R & i \in N \setminus \{t\},\ R \in \mathcal{R} \\
x_{ij}^R & \geq & 0 & (i,j) \in A,\ R \in \mathcal{R}.
\end{array}
$$

The interpretation of the variables in $SPRD$ is instructive. The variable α_i represents the expected number of times the recourse path encounters node i, and the variable x_{ij}^R represents the expected number of times the path passes across arc (i,j) with the current state being R.

The complementary slackness conditions relate the x_{ij}^R to the inequality constraint in $SPRLP$:

$$x_{ij}^R > 0 \text{ only when } u_i^R = l_{ij}^R + v_j$$

holds for all $(i, j) \in A$, $R \in \mathcal{R}$. We will call such x_{ij}^R *admissible.*

3 A primal-dual algorithm for SPR with reset and nonnegative arclengths

Although the linear programming formulation given in Section 2 cannot be solved explicitly, we can construct a polynomial-time solution algorithm by generalizing Dijkstra's algorithm using a primal-dual format. To start the algorithm, we first note that when arclengths are nonnegative, $SPRLP$ always has a feasible starting solution obtained by setting all v_i and u_i^R variables equal to zero. The algorithm proceeds by raising all of the u_i^R variables whose R is not associated with an admissible variable simultaneously by the same amount, adjusting the v_i's so that the expectation equations still hold, and creating admissible variables x_{ij}^R whenever one of the inequalities $u_i^R \leq l_{ij}^R + v_j$ becomes an equality. A general stage of the algorithm is defined by a set of feasible values $\hat{v}_i$, $i \in N$. The admissible x_{ij}^R are obtained by applying $\textsc{Bellman}(\hat{v}, i)$ to each node. Each ξ in the iterative loop corresponds to an arc (i, j), an index κ associated with an arclength realization of (i, j), and an event $\mathcal{E}_{ij}(\boldsymbol{k})$ that represents the set of states R for which arc (i, j) will be taken from node i when in state R. The variables x_{ij}^R, $R \in \mathcal{E}_{ij}(\boldsymbol{k})$ are the admissible variables associated with the pair (i, j) and κ, and the value of u_i^R for events $R \in \mathcal{E}_{ij}(\boldsymbol{k})$ is set to $l_{ij}^\kappa + \hat{v}_j$. Since the $\hat{v}_i$ values are dual feasible but not necessarily optimal, then the inequality $\hat{v}_i \leq E\left[\min_{j \in \mathcal{A}(i)}\{L_{ij} + \hat{v}_j\}\right]$ may not hold at equality. If this is the case, then the cumulative value of V computed in the iterative loop will eventually exceed $\hat{v}_i$. Let $\boldsymbol{k}$ be the set of indices obtained in the algorithm immediately before this occurs. Then the remaining u_i^R will be set to the same value $\hat{u}_i$ chosen such that

$$\hat{v}_i = \sum_{j\in\mathcal{A}(i)} \sum_{\kappa=1}^{k_{ij}} (l_{ij}^\kappa + \hat{v}_j)\rho_{ij}^\kappa + \hat{u}_i\eta_i, \tag{5}$$

where ρ_{ij}^κ is given by (4) and

$$\eta_i = 1 - \sum_{j\in\mathcal{A}(i)} \sum_{\kappa=1}^{k_{ij}} \rho_{ij}^\kappa. \tag{6}$$

We note that the $\mathcal{E}_{ij}(\boldsymbol{k})$'s *partition* the set of events R for which some x_{ij}^R is admissible, so that the ρ_{ij}^κ and η_i are all nonnegative. The value of η_i is 0 whenever every state is associated with some admissible x_{ij}^R, $j \in \mathcal{A}(i)$.

We improve the current solution by simultaneously raising all u_i^R associated with inadmissible variables x_{ij}^R, while updating $\hat{v}$ to maintain dual feasibility and equality of constraints corresponding to admissible variables. In particular, we compute improvement direction $y = (y_i : i \in N)$ such that $\hat{v} + \Delta y$ will continue to satisfy the equalities $u_i^R = l_{ij}^R + v_j$ for all admissible variables x_{ij}^R, and so that the values of all other u_i^R will increase by the same amount Δ. Now for all admissible x_{ij}^R we have

$$u_i^R = l_{ij}^R + \hat{v}_j + y_j\Delta,$$

and so we must maintain

$$\hat{v}_i + y_i\Delta = \sum_{j\in\mathcal{A}(i)} \left[\sum_{R:\ x_{ij}^R \text{ admissible}} (l_{ij}^R + \hat{v}_j + y_j\Delta)p_R + \sum_{R:\ x_{ij}^R \text{ inadmissible}} (\hat{u}_i + \Delta)p_R \right].$$

By combining the events in this equation using the $\mathcal{E}_{ij}(\boldsymbol{k})$ determined by (3) we get

$$\hat{v}_i + y_i\Delta = \sum_{j\in\mathcal{A}(i)} \sum_{\kappa=1}^{k_{ij}} (l_{ij}^\kappa + \hat{v}_j + y_j\Delta)\rho_{ij}^\kappa + (\hat{u}_i + \Delta)\eta_i.$$

Subtracting Equation (5) leaves only the Δ terms. Considering the associated coefficients, we get the following set of equations:

$$\begin{aligned} y_i &= \textstyle\sum_{j \in \mathcal{A}(i)} \rho_{ij} y_j \quad + \quad \eta_i, \quad i \in N \setminus \{t\} \\ y_t &= 0 \end{aligned} \tag{7}$$

where

$$\rho_{ij} = \sum_{\kappa=1}^{k_{ij}} \rho_{ij}^{\kappa} \tag{8}$$

and η_i is defined by Equation (6). Intuitively, ρ_{ij} is the probability that arc (i, j) will be the arc traversed when node i is reached, considering only the set of events processed so far, and η_i is the remaining probability not yet assigned to traversals from node i.

To give the next lemma, we define $G_\rho = (V, A_\rho)$, associated with the current ρ_{ij} values, as the subnetwork whose arcs are those (i, j) with $\rho_{ij} > 0$. We call G_ρ *t-reachable* if every arc of G_ρ is on some directed path to t, and define V_ρ to be the set of nodes (including t) that can reach t through arcs in G_ρ.

Lemma 3 *Let ρ have the property that G_ρ is t-reachable. Then the set of equations (7) always admits a nonnegative solution $y = (y_i : i \in N \setminus \{t\})$ having the property that $y_i < 1$ if and only if $i \in V_\rho$.*

Proof It is more convenient to write system (7) as

$$\begin{aligned} y_i - \sum_{j=1}^{n} \rho_{ij} y_j \quad - \quad \eta_i \mu &= 0, \quad i \in N \setminus \{t\} \\ y_t &= 0 \\ \mu &= 1 \end{aligned} \tag{9}$$

where $\rho_{ij} = 0$ whenever $j \notin \mathcal{A}(i)$. For convenience renumber the nodes in $V_\rho \setminus \{t\}$ as $1, \ldots, q$ and $t = n$. Since G_ρ is t-reachable, then ρ_{ij} must be 0 if either i or j is not in V_ρ. The matrix associated with system (9) is therefore

$$A = \left(\begin{array}{ccccccccc} 1 & -\rho_{12} & \cdots & -\rho_{1,q} & 0 & \cdots & 0 & -\rho_{1,t} & -\eta_1 \\ -\rho_{21} & 1 & \cdots & -\rho_{2,q} & 0 & \cdots & 0 & -\rho_{2,t} & -\eta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ -\rho_{q,1} & -\rho_{q,2} & \cdots & 1 & 0 & \cdots & 0 & -\rho_{q,t} & -\eta_q \\ 0 & 0 & \cdots & 0 & 1 & \cdots & 0 & 0 & -1 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & & 0 & \cdots & 1 & 0 & -1 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 1 \end{array} \right).$$

The definitions of ρ_{ij} and η_i imply that in the first q rows all off-diagonal elements of A are nonpositive and all rows sum to 0. Suppose we start to row reduce the system (9) by pivoting on the first q diagonal elements, and let $A^1, A^2, \ldots, A^q$ be the corresponding sequence of matrices. It is clear that each matrix in the sequence will continue to have the two properties given above, and in addition the values of ρ_{ij}^w, $i \neq j$ will always be nondecreasing as long as i and j are greater than w. Further, for any such i and j, if $\rho_{iw}^{w-1} > 0$ and $\rho_{wj}^{w-1} > 0$ then $\rho_{ij}^w > 0$. By induction it follows that $\rho_{ij}^w > 0$ if and only if there is a directed path from i to j in G_ρ all of whose intermediate vertices have index less than or equal to w. Since every node in $V_\rho \setminus \{t\}$ must have at least one path to t, it also follows that row w of A^w must have at least one $j > w$ with $\rho_{wj} > 0$. Therefore the w^{th} diagonal element of A^w is also positive, and so the pivot can always be performed.

From the above discussion we get that for the final matrix A^q, $\rho_{it}^q > 0$ for $i = 1, \ldots, q$, and since the row sums are 0 it follows that $0 \leq \eta_i^q < 1$ for $i = 1, \ldots, q$. By setting $y_i = \eta_i^q$, $i = 1, \ldots, q$, $y_i = 1$, $i = q+1, \ldots, n-1$, $y_t = 0$ and $\mu = 1$ we obtain a y that satisfies the conditions given by the lemma. □

We can now proceed with the dual update. This involves finding the largest Δ for which the set of node values $\hat{v}_i + \Delta y_i$ is still feasible to $SPRLP$, and requires in turn that for all inadmissible variables x_{ij}^R,

$$\hat{u}_i + \Delta \leq l_{ij}^R + \hat{v}_j + y_j \Delta.$$

In particular, for every inadmissible x_{ij}^R for which $y_j < 1$ we must have

$$\Delta \leq \frac{l_{ij}^R + \hat{v}_j - \hat{u}_i}{1 - y_j}.$$

We need not check every R of this sort, however, as the only thing that differentiates these inequalities is the arc (i, j) and the values l_{ij}^R. Further, from the procedure for determining admissible variables given at the beginning of the section, we need only check the values $l_{ij}^{k_{ij}+1}$ corresponding to the smallest arclength associated with an inadmissible variable. Now an arc (i, j) can only be inadmissible if $\eta_i > 0$, and this in turn implies that all arcs pointing out of i have $k_{ij} < r_{ij}$. Further, from Lemma 3 we have $y_j < 1$ if and only if $j \in V_\rho$. The value of Δ can thus be computed as

$$\Delta = \min \left\{ \frac{l_{ij}^{k_{ij}+1} + \hat{v}_j - \hat{u}_i}{1 - y_j} \ : \ (i, j) \in A \text{ with } \eta_i > 0 \text{ and } j \in V_\rho \right\}. \qquad (10)$$

Since there are paths from every node to t, it follows that if there is not already an arc $(i, j) \in E_\rho$ with $\eta_i > 0$ then any arc (i, j) coming into V_ρ from $V \setminus V_\rho$ will have $\eta_i = 1$. Thus as long as at least one $\eta_i > 0$, we have $\Delta < \infty$. We also note that since any new arc added to G_ρ in this process must have its head in G_ρ, then G_ρ continues to be t-reachable and thus Lemma 3 applies through all iterations. For every arc (i, j) minimizing (10) we have the associated event $\mathcal{E}_{ij}(\boldsymbol{k})$, which we update one by one — adjusting $\boldsymbol{k}$ accordingly — as indicated by Bellman$(\hat{v}, i)$. We then update $\hat{v}_i = \hat{v}_i + \Delta y_i$ for every $i \in N \setminus \{t\}$, $\hat{u}_i = \hat{u}_i + \Delta$, identify the added sets of admissible variables, and continue the process.

Now suppose $\eta_i = 0$ for all i. We claim the solution $\hat{v}$ is optimal. To verify this, we give the *restricted dual problem* $SPRD(\hat{v})$, consisting of those solutions to $SPRLP$ that are complementary with respect to $\hat{v}$:

Find $\alpha_i,\ x_{ij}^R$ such that

$$
SPRD(\hat{v}):\quad
\begin{array}{rcll}
\alpha_j & = & \displaystyle\sum_{i\in\mathcal{B}(j)}\sum_{R\in\mathcal{R}} x_{ij}^R \ + & \left\{\begin{array}{ll} 1 & j=s \\ 0 & j\in N\setminus\{s,t\}\end{array}\right. \\
p_R\alpha_i & = & \displaystyle\sum_{j\in\mathcal{A}(i)} x_{ij}^R & i\in N\setminus\{t\},\ R\in\mathcal{R} \\
x_{ij}^R & \geq & 0 & (i,j)\in A,\ R\in\mathcal{R} \\
\text{and} & & & \\
x_{ij}^R & = & 0 & \text{if } x_{ij}^R \text{ is inadmissible.}
\end{array}
$$

Lemma 4 *Let $\hat{v}$ be a solution to $SPRLP$, with ρ_{ij}^{κ} and η_i computed as above, and suppose that $\eta_i = 0$ for all i. Then $SPRD(\hat{v})$ has a solution for any $s \neq t$.*

Proof If x_{ij}^R is an admissible variable, then R will be in exactly one $\mathcal{E}_{ij}(\boldsymbol{k})$, so that the second equation in $SPRD(\hat{v})$ reduces to

$$p_R\alpha_i = x_{ij}^R,\ \text{where } R\in\mathcal{E}_{ij}(\boldsymbol{k}),$$

so that

$$\sum_{R\,\in\,\mathcal{E}_{ij}(\boldsymbol{k})} x_{ij}^R = \sum_{R\,\in\,\mathcal{E}_{ij}(\boldsymbol{k})} p_R\alpha_i = \rho_{ij}^{k_{ji}}\alpha_i.$$

Summing over all $\mathcal{E}_{ij}(\boldsymbol{k})$ associated with the arc (i,j) and using Equation (8) we get

$$\sum_{R\in\mathcal{R}} x_{ij}^R = \sum_{\kappa=1}^{k_{ij}} p_{ij}^{\kappa}\alpha_i = \rho_{ij}\alpha_j$$

The equations in $SPRD(\hat{v})$ now reduce to

$$\alpha_j = \sum_{i\in\mathcal{B}(j)} \rho_{ij}\alpha_i \quad + \left\{\begin{array}{ll} 1 & j=s \\ 0 & j\in N\setminus\{s,t\}\end{array}\right. \tag{11}$$

Let B be the corresponding matrix of α_i coefficients. We will show that B^T has a nonnegative inverse, and so the system (11) will have a nonnegative solution for any $s \neq t$. B^T is identical to the first $n-1$ rows and columns of the matrix A used in Lemma 3. In particular, since every node can reach t by a path in G_ρ then pivoting down the diagonal of B^T will always result in a positive pivot element, and the pivot eta-matrices will always be nonnegative. Thus the final inverse of B^T will also be nonnegative, and hence so will the solution to system (11). This provides the required solution to $SPRD(\hat{v})$. □

It follows that the final $\hat{v}$ values will be optimal for all starting nodes s, and so each v_i represents the length of a shortest recourse path from i to t. This completes the discussion of the procedure for finding shortest recourse paths. The full algorithm is presented below.

Procedure RECOURSE

Input: Network $G = (N, A)$, arclength realizations l_{ij}^k and associated probabilities p_{ij}^k for $(i, j) \in A$ and $k = 1, \ldots, r_{ij}$, and sink node t.

Output: Node values $v^* = (v_i^* : i \in N)$ such that v_i^* is the expected length of a shortest recourse path from i to t. The path is found by choosing, at each $i \neq t$, an arc (i, j) to traverse that has the minimum value of $L_{iw} + v_w^*$ over all $w \in \mathcal{A}(i)$.

Procedure:

```
Initialize $\hat{v}_i = \hat{u}_i = 0$ for $i \in N$, $\eta_i = 1$ for $i \in N \setminus \{t\}$ and $\eta_t = 0$,
    and $k_{ij} = \rho_{ij} = 0$ for $(i, j) \in A$.
do while $\eta_i > 0$ for some $i \in N$
    solve the system (7) with respect to the current $\rho$ and $\eta$
        values, and find $\Delta$ using (10).
    for each $i \in N$, set $\hat{v}_i = \hat{v}_i + y_i \Delta$ and $\hat{u}_i = \hat{u}_i + \Delta$.
    for each $(i, j)$ with $\hat{u}_i = l_{ij}^{k_{ij}+1} + \hat{v}_j$ and $\eta_i > 0$
        add 1 to $k_{ij}$.
        add $P[\mathcal{E}_{ij}(\boldsymbol{k})]$ to $\rho_{ij}$.
        subtract $P[\mathcal{E}_{ij}(\boldsymbol{k})]$ from $\eta_i$.
    end for
end do while

Set $v_i^* = \hat{v}_i$ for all $i \in N$.
```

We demonstrate the procedure for the example given in Figure 3. Table 2 gives the set of $\hat{v}$, $\hat{u}$, y, and Δ values, the event $\mathcal{E}_{ij}(\boldsymbol{k})$ of all states R such that x_{ij}^R becomes admissible, and the associated updated k_{ij}, ρ_{ij} and η_i values. Since the $\hat{u}_i$ all start off at value 0, then their values will all be the same, and therefore we can represent them by a single number. The indices for an added event at node i give only the sequence of k_{ij} for j in the forward

	$\hat{v}_i$				$\hat{u}_i$	y_i					added	new values		
iteration	1	2	3	4	(common)	1	2	3	4	Δ	event $\mathcal{E}_{ij}(\boldsymbol{k})$	k_{ij}	ρ_{ij}	η_i
1	0	0	0	0	0	1	1	1	0	1	$\mathcal{E}_{24}(0,1)$	1	$\frac{1}{2}$	$\frac{1}{2}$
2	1	1	1	0	1	1	$\frac{1}{2}$	1	0	1	$\mathcal{E}_{34}(0,1)$	1	$\frac{1}{2}$	$\frac{1}{2}$
3	2	1.5	2	0	2	1	$\frac{1}{2}$	$\frac{1}{2}$	0	2	$\mathcal{E}_{23}(1,1)$	1	$\frac{1}{4}$	$\frac{1}{4}$
4	4	2.5	3	0	4	1	$\frac{3}{8}$	$\frac{1}{2}$	0	$\frac{4}{5}$	$\mathcal{E}_{12}(1)$	1	$\frac{1}{2}$	$\frac{1}{2}$
5	4.8	2.8	3.4	0	4.8	$\frac{11}{16}$	$\frac{3}{8}$	$\frac{1}{2}$	0	$\frac{16}{5}$	$\mathcal{E}_{12}(2)$	2	1	0
6	7	4	5	0	8	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{2}$	0	$\frac{8}{5}$	$\mathcal{E}_{31}(1,1)$	1	$\frac{1}{4}$	$\frac{1}{4}$
7	7.6	4.6	5.8	0	9.6	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{9}{5}$	$\mathcal{E}_{23}(2,1)$	2	$\frac{1}{2}$	0
8	8.2	5.2	6.4	0	11.4	$\frac{1}{7}$	$\frac{1}{7}$	$\frac{2}{7}$	0	$\frac{28}{5}$	$\mathcal{E}_{34}(2,1)$	2	$\frac{1}{2}$	0
9	9	6	8	0										

Table 2: Recourse applied to the SPR example

star of i, with components in order of increasing node number. We show the details of iteration 8 of the example. The k_{ij} and ρ_{ij} values at this stage are

(i,j)	$(1,2)$	$(2,3)$	$(2,4)$	$(3,1)$	$(3,4)$
k_{ij}	2	2	1	1	1
ρ_{ij}	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$

and $\eta = (0,0,\frac{1}{4},0)$. Equation (7) is

$$\begin{array}{rclcccccc}
y_1 & = & & & 1y_2 & & & & \\
y_2 & = & & & & \frac{1}{2}y_3 & + & \frac{1}{2}y_4 & \\
y_3 & = & \frac{1}{4}y_1 & & & & + & \frac{1}{2}y_4 & + \quad \frac{1}{4} \\
y_4 & = & 0 & & & & & &
\end{array}$$

which has solution $(y_1,y_2,y_3,y_4) = (\frac{1}{7},\frac{1}{7},\frac{2}{7},0)$. To find Δ, we note that node 3 is the only one with $\eta_i > 0$. Using Equation (10) we have

$$\Delta = \min\left\{\frac{9+8.2-11.4}{1-1/7}, \frac{17+0-11.4}{1-0}\right\} = 5.6$$

and so the new v values are $(8.2+.8, 5.2+.8, 6.4+1.6, 0+0) = (9,6,8,0)$.

The index k_{34} is increased to 2, and the value $P[\mathcal{E}_{34}(\boldsymbol{k})] = \frac{1}{4}$ is added to ρ_{34}. This makes all η_i values 0, and the current solution is optimal. We obtain the dual solution by solving system (11):

$$\begin{array}{rcllll} \alpha_1 & = & & & \frac{1}{4}\alpha_3 & + \quad 1 \\ \alpha_2 & = & 1\alpha_1 & & & \\ \alpha_3 & = & & \frac{1}{2}\alpha_2 & & \end{array}$$

Figure 4 gives the dual solution, with the value at node i being α_i = the expected number of visits to node i, and the values on arc (i,j) being $(\rho^1_{ij}, \rho^2_{ij})$ = the probability of traversing (i,j) under the policy given by the v_i values under each realization of that arc. The expected number of times arc (i,j) is traversed under arclength realization k (this is the sum of the x^R_{ij} values over all states R associated with traversing (i,j) under realization k) is then $\alpha_i p^k_{ij}$, and the dual objective function is

$$\sum_{(i,j)\in A}\sum_{k=1}^{r_{ij}} c^k_{ij}\alpha_i p^k_{ij} = 2\cdot\frac{8}{7}\cdot\frac{1}{2} + 4\cdot\frac{8}{7}\cdot\frac{1}{2} + 2\cdot\frac{4}{7}\cdot\frac{1}{4} + 9\cdot\frac{4}{7}\cdot 0 + 2\cdot\frac{4}{7}\cdot\frac{1}{2}$$
$$+17\cdot\frac{4}{7}\cdot\frac{1}{4} + 1\cdot\frac{8}{7}\cdot\frac{1}{4} + 5\cdot\frac{8}{7}\cdot\frac{1}{4} + 1\cdot\frac{8}{7}\cdot\frac{1}{2} + 17\cdot\frac{8}{7}\cdot 0 = 9 = v_1.$$

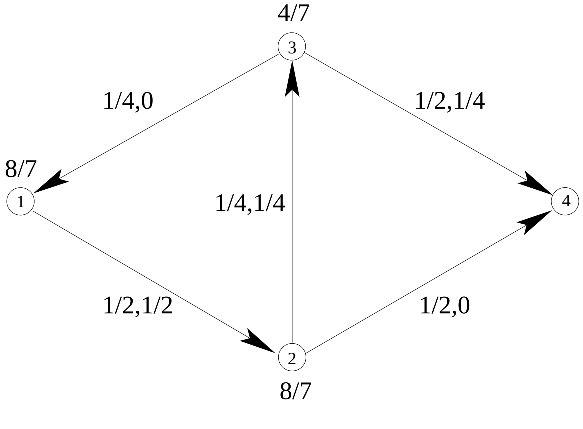

Figure 4: Expected number of visits and traversal probabilities

Theorem 1 *Procedure* Recourse *correctly finds the optimal values v^*_i in $O(rn^2)$ steps, where $r = \sum_{(i,j)\in A} r_{ij}$.*

Proof That the procedure terminates in an optimal solution follows from Lemma 4, since the corresponding feasible dual solution is complementary to the final $\hat{v}$ solution. For the complexity of the procedure, we have by the construction that in each iteration of Recourse every variable x_{ij}^R with R in the corresponding set $\mathcal{E}_{ij}(k)$ becomes admissible, and further, after this, no variable x_{ij}^R with $l_{ij}^R = k_{ij}$ is admissible. It follows that after at most r stages *all* of the variables must be admissible, and hence all $\eta_i = 0$ and Lemma 3 applies.

It remains to determine how much work is done in each iteration. The computation of the y variables can be done iteratively, by maintaining the inverse of each matrix A^w in Lemma 3. Updating the inverse can be done by changing the (i, j) and (i, t) entries in A^w corresponding to the change in ρ_{ij} and η_i, and performing two pivots. The whole update therefore takes $O(n^2)$ steps. Computing Δ takes $O(m)$ steps, and so the whole procedure takes $O(rn^2)$ time. □

4 The SPR problem in the presence of negative arclengths

A critical requirement for the procedure Recourse to find SPRs is that we have an initial feasible solution to *SPRLP*. When all arclengths are negative the zero solution is feasible, but the nonnegativity requirement can be relaxed if we can find some other way to produce a feasible solution to *SPRLP*. In the deterministic shortest path problem the requirement that no negative cycles exist in the network is sufficient to determine an optimal solution to the problem using one of the label-correcting techniques. This can be applied to SPR in the following sense. Suppose the network having each arclength equal to its *minimum* realization contains no negative length cycle. Then we can find the shortest path distances v_i^0 from each node i to t using these minimum lengths. If we then reset each arclength realization to

$$\tilde{l}_{ij}^r = l_{ij}^r - v_j^0 + v_i^0$$

then the $\tilde{l}$ values are all nonnegative, and so we can solve the SPR problem with the modified arclengths using RECOURSE. If the final optimal pathlength values are v_i^*, $i \in N$, then the optimal values for the problem with negative arclengths will be $v_i^* + v_i^0$, $i \in N$.

The property of no negative minimum-value cycles in G is not a necessary condition for $SPRLP$ to have a feasible solution. Consider the network in Figure 5. The lengths of arcs (1,2), (3,1), (4,1), and (2,5) have deterministic lengths as marked, and the lengths of arcs (2,3) and (2,4) are 0 or –6 with probability $\frac{1}{2}$ each. Using the negative values on the arcs (2,3) and (2,4) gives a network with negative cycles, but the values $v_1 = 1$, $v_2 = -2$, $v_3 = v_4 = 3$, and $v_5 = 0$ gives a feasible — in fact optimal — solution to $SPRLP$. Using *expected* arclengths to determine negative cycles also does not work, for a different reason. Consider the same network Figure 5, but now give the negative lengths on arcs (2,3) and (2,4) the value -7. The expected length network has no negative cycles. The strategy that always takes at node 2 the choice of the smallest length arc among arcs (2,3) and (2,4), however, will result in expected gain going around the respective cycle of $\frac{3}{4}(-2) + \frac{1}{4}(5) < 0$. Thus the SPR problem is unbounded below, and so there is no feasible solution to $SPRLP$.

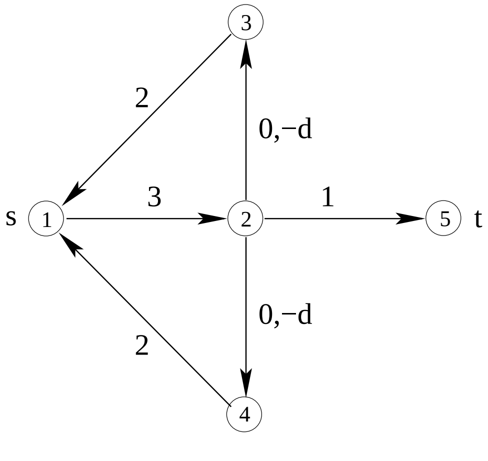

Figure 5: Example with negative arclengths

Precisely determining the feasibility of $SPRLP$ in the presence of negative arclengths is likely to be more complicated than reducing it to a deterministic problem, and to require extending one of the deterministic label-correcting techniques along similar lines to those taken in this paper. We conjecture that there exists a polynomial-time algorithm to solve SPR-reset problems for general networks with arbitrary arclengths, understanding that this includes the possibility of a $-\infty$-length solution consisting of indefinite cycling.

5 A Markov Chain Model

Although in general the results of this paper do not apply to Markov chain models, an exception can be made for a slight modification of the model presented by Psaraftis and Tsitsiklis [9]. They considered a node-independent Markov chain model with delays allowed at each node. We note that delays can be built into the standard SRP problems studied above, simply by placing loops at each of the nodes whose traversal represents a one-time-unit delay at that node. This is somewhat unsatisfactory, however, as it calls for a complete reset after each unit of delay. We allow a time-dependent reset to be made for this model by letting a Markov chain operate during the time spent waiting at a node, but having a complete reset occur — starting with the initial distribution — upon revisiting that node later in the traversal.

Let $\mathcal{M}_i$ be the Markov chain associated with node i, with r_i states. The state forward star lengths are $\boldsymbol{l}_i^k = (l_{ij}^k : \ j \in \mathcal{A}(i))$, $k = 1, \ldots, r_i$, the initial state distribution is chosen using probabilities p_k, $k = 1, \ldots, r_i$, and the state transitions occur using $r_i \times r_i$ transition probability matrix M^i. We make the important assumption that the state arclength vectors l^k are distinct, so that the traveler always knows exactly which state of the Markov process is occurring while at that node. Each time node i is encountered the state, and associated arclengths, are chosen from the initial distribution. If it is decided to wait at node i there is a delay charge of D, and the arclengths are reset to those associated with the new state using the Markov process.

It turns out that this process can be modeled using the appropriate modification of the original network. For each node i, we create $2r_i + 1$ nodes, labeled i_k for $k = 1, \ldots, r_i$ and i'_k for $k = 0, \ldots, r_i$. All arcs in G coming into node i now go into node i'_0, and each node i_k has arcs pointing from it to all of the nodes in $\mathcal{A}(i)$ (or more precisely their i'_0 counterparts), with arclengths equal to those given by l^k. Each node i_k has an arc from it to i'_k of length D. From i'_k there are arcs pointing to each of the nodes i_j, $j = 1, \ldots, r_i$, which will represent the initial and transition state determinations for the Markov chain. Figure 6 shows the replacement of a node i for which $\mathcal{M}^i$ has three states. The dynamics of the Markov chain at this node are modeled by assigning arclength vectors at i'_x by

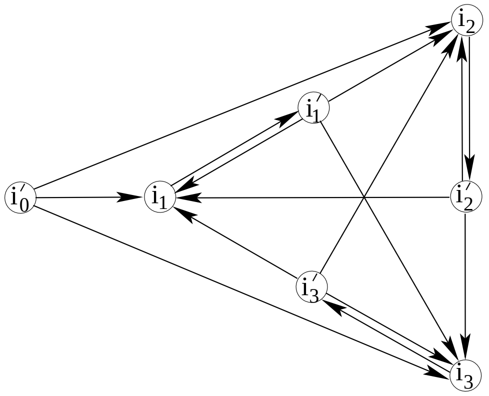

Figure 6: Transformation for a 3-state Markov chain

$$l^k_{i'_x i_w} = \begin{cases} 0 & w = k \\ \infty & w \neq k \end{cases}$$

with probabilities $p^k_{i'_0} = p_k$ or $p^k_{i'_x} = M^i_{x,k}$, $x = 1, \ldots, r_i$. This set of arclengths has the effect of requiring a traveler at node i'_x to take the single arc that goes to the node i_k as required by either the initial or transition probabilities. At each of the nodes i_x the traveler must decide whether to take one of the arcs going to a node of $\mathcal{A}(i)$, with length corresponding to that indicated by the Markov chain at state x, or to take the arc to i'_x, which corresponds to choosing to delay at that node. The transition nodes are arranged so that the traveler cannot see the new transition state until after the choice to delay is made. It follows that this Markov chain model can be transformed to a node-independent model, and therefore we can extend the Psaraftis/Tsitsiklis model (with the reset-upon-revisiting modification) to apply to networks with cycles. The following result follows from Theorem 1.

Theorem 2 *The optimal recourse path for the Psaraftis/Tsitsiklis Markov model, with the reset-upon-revisiting modification, can be found in $O(r(m + r)(n + r)^2)$ steps, where $r = \sum_{i=1}^{n} r_i$.*

One can define an arc-independent version of this model, where each of the individual arc Markov chains operates independently and resets independently at each revisit to the node. It would be an interesting problem to decide whether the techniques of this paper apply to that version as well.

References

[1] G. Andreatta and L. Romeo, Stochastic shortest paths with recourse, Networks 18 (1988), 193–204.

[2] R.K. Cheung, Iterative methods for dynamic stochastic shortest path problems, Naval Research Logistics 45 (1998), 769–789.

[3] J. Croucher, A note on the stochastic shortest route problem, Naval Research Logistics Quarterly 25 (1978), 729–732.

[4] M.O. Ball, C. Colbourn, and J.S. Provan, Network Reliability, Chapter 11 in Handbooks in OR & MS 7, Elsevier Science B.V., Amsterdam, 1995, pp. 673–762.

[5] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman, San Francisco, 1979.

[6] R. Hall, The fastest path through a network with random time-dependent travel times, Transportation Science 20 (1994), 182–188.

[7] E. Miller-Hooks, Adaptive least-expected time paths in stochastic, time-varying transportation and data networks, Networks 37 (2001), 35–52.

[8] G.H. Polychronopoulos and J.N. Tsitsiklis, Stochastic shortest path problems with recourse, Networks 27 (1996), 133–143.

[9] H.N. Psaraftis and J.N. Tsitsiklis, Dynamic shortest paths in acyclic networks with Markovian arc costs, Operations Research 41 (1993), 91–101.

[10] S. Ross, Introduction to Stochastic Dynamic Programming, Academic Press, New York, 1983.