Finding Minimum-Cost Dynamic Routing Policies in Stochastic-State Networks with Link Failures

David Fajardo and S. Travis Waller

The focus of this research is to develop minimum-cost dynamic routing policies that can identify connecting paths between nodes in a stochastic-state network. In this context, the stochastic element of the network is the network structure, that is, the set of links that exist under each realization of the network state. It is assumed that information about the true network state can be gathered only endogenously through the routing decisions themselves. As such, the objective becomes finding a dynamic policy that accounts for information gathered en route that minimizes the cost of detection of a viable path between a given origin and destination. An exact solution method, based on a Markovian decision process, is presented, and then a heuristic based on an aggregating function of the network is developed.

Link failures in the context of networks imply a disconnection or a lack of connectivity between two previously connected components. On the basis of the context of the problem, this may involve physical or virtual failures. In the context of transportation networks, a link failure may represent the structural failure of a bridge or the blockage of a road segment because of an accident. In the context of communications networks, a link failure may represent the inability of two points in the network to send and receive information because of weather or technical difficulties.

The motivation for this research arises from the field of emergency routing. Although in regular routing operations the network topology is assumed to be known a priori, in circumstances in which the topology of the network may be affected in stochastic ways, for example, in the case of natural or human-caused disasters, algorithms that can account for this uncertainty in network structure must be developed. Furthermore, when the routing in question is of a sensitive type, for example, routing involving injured individuals or emergency resources, the time savings that can be obtained from proper exploitation of knowledge of the probability distributions governing the failure of links could be significant.

A secondary source of motivation for this problem came from the field of telecommunications, in which link failures are much more common and proper planning of information routing policies is necessary in many cases to maintain connectivity throughout the network. One main assumption in the problem presented in this research that was taken from the field of communications is the need for the rerouting of units from an origin, that is, the routing representation of a lost routing unit or, in the case of telecommunication networks, information packets.

School of Civil and Environmental Engineering, University of New South Wales, Sydney, New South Wales 2052, Australia. Corresponding author: D. Fajardo, davidfajardo2@gmail.com.

Transportation Research Record: Journal of the Transportation Research Board, No. 2283, Transportation Research Board of the National Academies, Washington, D.C., 2012, pp. 113–121.
DOI: 10.3141/2283-12

LITERATURE REVIEW

This section surveys the literature of shortest-path problems most relevant to the problem studied in this research, namely, stochastic shortest-path problems and, in particular, those problems for which stochasticity affects the network structure.

Stochastic Shortest-Path Problems

Deterministic shortest-path problems seek a path or sequence of nodes that minimizes the generalized cost of traveling between a given origin and destination. Stochastic shortest-path problems introduce uncertainty in the problem parameters so that the stochastic nature of such parameters can be modeled.

In the classical deterministic shortest-path problem, all problem parameters are constant and no negative cost cycles exist. In this specific case, it can be shown that at least one deterministic policy exists (*1*). Several variations of the problem have been developed since the original work by Dijkstra, including stochastic (*2–7*), time-dependent (*5, 8, 9*), and recourse-based (*10, 11*) problems.

Frank considered one of the earliest versions of a stochastic shortest path (*7*). The focus of the research was to establish the probability distribution of the length of the shortest path in a network in which link costs are stochastic. Further work that focused on measures of performance on the basis of probabilistic statements includes that of Sigal et al. (*2*), Kulkarni (*6*), and Corea and Kulkarni (*3, 4*).

Stochastic shortest-path problems in which the objective is to minimize a given utility function of flows have received considerable attention in the research of Loui (*12*), Eiger et al. (*13*), Mirchandani and Soroush (*14*), Murthy and Sarkar (*15*), and Mirchandani (*16*).

Bertsekas and Tsitsiklis first considered a Markovian decision process (MDP) approach to the case of a stochastic shortest path and used classic results from the MDP literature to establish the existence of an optimal stochastic policy, as well as a policy iteration solution methodology (*17*). Although the approach is mathematically sound, the concern with the modeling of network problems as MDPs is the size of the state and policy spaces. Because of the combinatorial nature of routing problems, enumeration of all possible states is computationally expensive, and as such, classical MDP methods

such as policy and value iteration are usually not computationally tractable for realistically sized routing problems. The problem considered in this research exploits the solution structure to reduce the set of decisions that needs to be considered at each stage to develop a more efficient solution procedure to the problem.

Shortest-Path Problems with Recourse

Shortest-path problems with recourse introduce a sequence of decision-making stages into the routing process during which information acquired up to that point can be used to make such a decision.

Although the literature on shortest-paths algorithms is extensive, the focus of this section is on shortest-path problems in which the desired solution is not a single route per se but, rather, a routing policy. The need for a routing policy arises in general as a result of introduction of stochasticity into the problem, whether it be in the link costs, link existence, or link travel times (in the case in which travel costs are different from link travel times).

Pretolani examined a shortest-path problem in which link costs are both stochastic and time dependent (*9*). The problem is reformulated in a time-expanded hypergraph in which the objective becomes to find the hyperpath (that is, a family of paths connecting two nodes) of least cost. Waller and Ziliaskopoulos (*10*) and Polychronopoulos and Tsitsiklis (*11*) consider versions of the shortest-path problem in which recourse is available on the basis of information gathered en route; that is, the routing policy is dependent on the particular realization of link costs that are encountered. Polychronopoulos and Tsitsiklis (*11*) assume that once a link cost is observed, it does not subsequently change, whereas Waller and Ziliaskopoulos (*10*) assume that each visit to a link is a new random trial. Fu and Rilett consider a dynamic and stochastic shortest-path problem in which the link costs are modeled as continuous-time stochastic processes (*18*).

Stochastic Shortest-Path Problems with Link Failures

The field of routing in the presence of link failures has received increasing attention as the requirements for the maintenance of large networks that are connected has increased in the telecommunications industry. Development of restoration procedures, that is, rerouting of information in the presence of random link failures, is an active area of research, as previously discussed. This section focuses on the literature on shortest paths under probabilistic link failures, which differ from those discussed earlier in that they explicitly account for the risk of link failure rather than just being a reaction to the failure of links.

Zhang et al. use a multiplication-to-summation method to transform failure probabilities into additive link cost functions by taking the logarithm of the link failure probabilities (*19*). Using this technique, Zhang et al. identify, on the basis of a minimum threshold of availability probability, which paths must be protected from failure (*19*). Modiano and Lee attempt to determine sets of disjointed paths with the lowest joint probability of failure in networks in which link failures are probabilistic and potentially correlated (*20*). Andreatta and Romeo consider a problem in which links have an explicit probability of failure (*21*). Link failures are learned when the upstream node is reached and the route must be continued to the destination.

Contribution

The majority of work on shortest-path problems in the presence of link failures has focused on detection of sets of paths that have an acceptable probability of inclusion of at least one path not affected by failures. The problem presented in this research focuses on the issue of routing in a network in which an available path is guaranteed to exist; however, as routing costs must be accounted for, a balance between the cost of a path and its survivability must be established. Intuitively, a path with a very high cost and a low probability of failure may be preferred over a path that is cheaper and yet has a very high probability of failure. Furthermore, the problem is not chance constrained in nature: detection of a path is not sought in a certain fraction of problem instances but in every instance of the problem.

Although MDP formulations for shortest-path problems have been explored in the literature, the formulation for the problem presented in this research uses structural properties of the problem to reduce its complexity. In particular, deterministic shortest-path procedures are used to identify the set of paths that needs to be considered to identify nontrivial transitions. A heuristic procedure based on identification of feasible paths for aggregated network states is also proposed to generate feasible solutions in an efficient manner.

PROBLEM DEFINITION

The objective of this section is to provide an informal definition of the problem at hand, including the objective, decisions, and constraints of the problem, and a small illustrative example that helps motivate the problem, which leads to the formal definition of the problem later in the paper.

The problem addressed in this section is the following: given a single vehicle, determine the optimal a priori routing policy that guarantees detection of a feasible route between a given origin and destination with minimum routing cost in a stochastic-state network (SSN) in which the stochastic parameter is the binary variable representing the availability of each link and routing must restart from the origin if a link failure is encountered as part of a chosen path.

The requirement that units must be rerouted from the origin can be interpreted in several ways within specific application contexts:

- Loss of routing unit. Interpretation of loss of the routing unit is relevant to routing operations in which the unit is lost or destroyed if it encounters a link failure. Examples are routing of units in military operations and routing of information in telecommunication networks.
- Unit-based service of link. Interpretation of the unit-based service of a link is relevant to routing operations in which, in addition to establishment of connectivity, it is also desirable to repair network elements to restore the original state of the network.
- Lack of availability of rerouting mechanisms. The interpretation that rerouting mechanisms are unavailable is relevant to routing operations in which the routing units used do not have an available rerouting mechanism or in which communication of such a rerouting decision is not possible. Examples are routing of units with artificial intelligence not prepared to deal with the failure of paths or cases in which information about the system is available to a central planning unit but not the individual routing units.

An "online policy" is defined to be the set of routing decisions conditional on the information acquired at each stage of the problem.

MATHEMATICAL FORMULATION

To define the problem mathematically in a thorough manner, the network characteristics are first defined and the dynamic programming formulation is provided.

Stochastic-State Network

Define an SSN, $SSN_A(\Omega_A, P)$, where the set Ω_A is a finite set of elements ω equal to (N, A, c, d); where N is the set of nodes; A is the set of available links under scenario ω; c is the link cost vector; and d is the demand vector, which is equal to 1 for the source node and −1 for the destination node; and P is probability. Assume that N, c, and d are the same across all network states. The subscript A is meant to indicate stochasticity in the set of links only; that is, only the link set changes from state to state.

Given an origin r and destination t, further assume that for every scenario $\omega \in \Omega$ at least one path exists between r and t; that is, $\rho_\omega(r, t) \neq \varnothing$, where $\rho_\omega(r, t)$ is the set of paths between nodes r and t in scenario ω. Furthermore, define $\rho_H(r, t)$ for a set of scenarios $H \subseteq \Omega$. $\rho_H(r, t)$ is equal to $\cap_{\forall \omega \in H} \rho_\omega(r, t)$, that is, the set of paths between nodes r and t that exists under all network states $\omega \in H$. Further define $c(\rho)$ to be equal to $\sum_{\forall (i,j) \in \rho} c(i, j)$ as the cost of path ρ.

Markovian Decision Process

To model this routing problem as an MDP, four main concepts are defined: state, control policies, decisions, and transition costs. The problem can be defined as an MDP because of the stochastic-state network setup: it is such that, for a given control policy, the problem reduces to a Markov chain with absorbing states defined by the routing unit being positioned at the destination.

State

Define a state s^k, which is equal to (i^k, Ω_{s^k}), where i^k is the current location of the routing unit and $\Omega_{s^k} \subseteq \Omega$ is the set of possible scenarios in state s^k. Once again, the distinction between a state in the MDP formulation of the problem and a network state, which refers to the SSN representation of the network, is emphasized. This definition of state is such that the transition probabilities from state to state can be defined a priori for a given control policy and such that the events that occurred prior to the current state that are relevant to the decision process are all accounted for in the definition of the state itself. It is not necessary to know the path taken to the current node; only the implications of the survivability of the path taken for the set of remaining network states need to be known.

Control Policies

Define a control policy μ^k to be a mapping from s^k into a path between i^k and t, where μ^{s^k} is the path to be followed under state s^k. Denote $\mu_i^{s^k}$ to be the ith link in path μ^{s^k}.

Transition Costs and Probabilities

For a generic state s^k and control policy μ: such that $\mu(s^k) = (i, j)$, transitions to two different states are possible. These transitions correspond to whether link (i, j) has failed under each of the network states in Ω_{s^k}. Let A^{ω_0} and A^{ω_f} be the set of links that exist in network states ω_0 and ω_f, respectively:

- The state in which link (i, j) failed is $s_f^{k+1} = (r, \Omega_f^{k+1})$, where $\Omega_f^{k+1} = \{\omega_f \in \Omega_{s^k}: (i, j) \notin A^{\omega_f}\}$ is the set of network states in which link (i, j) failed. This transition occurs with transition probability $P(\Omega_f^{k+1})$ and cost $c(i, j)$.
- The state in which link (i, j) did not fail is $s_o^{k+1} = (j, \Omega_o^{k+1})$, where $\Omega_o^{k+1} = \{\omega_o \in \Omega_{s^k}: (i, j) \in A^{\omega_o}\}$ is the set of network states in which link (i, j) did not fail. This transition occurs with transition probability $P(\Omega_o^{k+1})$ and cost $c(i, j)$.

Problem Formulation

Given the states, control policies, transition costs, and transition probabilities defined above, the problem can be formulated given the following recursive function:

$$\min_\mu V\left(s^o = \{0, \Omega\}\right) \tag{1}$$

where

$$V\left(s^k\right) = \min_{\mu \in A: i = i^k} c(\mu) + P\left(\Omega_f^{k+1}\right) V\left(s_f^{k+1}\right) + P\left(\Omega_o^{k+1}\right) V\left(s_o^{k+1}\right) \tag{2}$$

$$V\left(\left\{t, \Omega^f\right\}\right) = 0 \tag{3}$$

The beginning recursion is the recursive function evaluated when the routing vehicle is at the origin and the full set of network states Ω is possible. The recursive function consists of three terms: the transition cost, which is equal to the link cost; the cost of rerouting of a unit from the origin in the case in which a link failure is encountered, weighted by the probability of such an event; and the cost of routing of a unit to the origin from the next node, in the case in which the link has not failed, weighted by the probability of such an event.

The terminal states for which the recursive function value is 0 are defined by the location variable being equal to the destination and Ω^f, which is the set of network states under which the path followed is valid. This set Ω^f can be any set of network states.

Given this formulation, the solution method is developed in the next section.

SOLUTION ALGORITHM

An algorithm for the problem defined in this paper is present on the basis of the development of upper and lower bounds based on the properties outlined in the previous section within a dynamic programming framework. To provide a formal definition of the algorithm, the recursive function presented above is again used:

$$V\left(s^k\right) = \min_{\mu = (i,j) \in A: i = i^k} c(i, j) + P\left(\Omega_f^{k+1}\right) V\left(s_f^{k+1}\right) + P\left(\Omega_o^{k+1}\right) V\left(s_o^{k+1}\right) \tag{4}$$

The main concept used in the development of the algorithm is the fact that a subset of links is transition links, that is, links that will partition the set of network states, and as such, an encounter with any particular transition link requires the solution of two subproblems corresponding to the outcomes of a link failure and

a link survival. A formal definition of a transition link is provided as follows:

Definition 1. Let a link (i, j) be a transition link under state $s^k = (i^k, \Omega^k)$ if $(i, j) \notin \cap_{\omega \in \Omega^k} A^\omega$ and $(i, j) \in \cup_{\omega \in \Omega^k} A^\omega$, that is, if link (i, j) does not exist in at least one network state and it exists in at least one network state.

The definition is motivated by the fact that a nontransition link transitions into a single state with a probability of 1 as either $P(\Omega_f^{k+1})$ or $P(\Omega_o^{k+1})$ will be equal to 0. The transition that occurs when a nontransition link is chosen as a trivial transition is further defined. A sequence of trivial transitions beginning at state s^k comprises a deterministic path between i^k and the tail node of either a transition link or a link that arrives at the destination. As such, all outgoing links at a particular state do not need to be considered; rather, the focus can be on the least-cost paths that lead to either a transition link or the destination. This set of paths is defined as follows:

Definition 2. Let the closest transition path set (CTPS) of a state s^k, $\mu \in \text{CTPS}(s^k)$, where $\mu = (\mu^0 = i^k, \mu^1, \mu^2, \ldots, \mu^{N-1} = j^\mu, \mu^N = l^\mu)$, be the set of least-cost paths that uses only nontransition links between i^k and either the destination or the downstream node of each transition link that can be reached from the initial node, i^k, where the link (j^μ, l^μ) is the link arriving at the destination or the transition link reached, respectively.

The CTPS can be found efficiently by running of a shortest-path algorithm in a network consisting only of links that exist in all possible network states, that is, nontransition links. The resulting solution will be a set of paths to each node that is reachable. The cost to reach each transition link will be equal to the cost required to reach its head node.

Equation 4 is reformulated to account for the fact that, at every state, only paths in CTPS need to be considered:

$$V(s^k) = \min_{\mu \in \text{CTPS}(s^k)} c(\mu) + P(\Omega_f^{k+1}) V(s_f^{k+1}) + P(\Omega_o^{k+1}) V(s_0^{k+1}) \quad (5)$$

where $c(\mu)$ is the cost of path μ, where $\mu \in \text{CTPS}(s^k)$.

To provide an interpretation of the formulation, at each state the optimal decision can be either (*a*) a path to a transition link such that the routing cost to and including that transition link plus the expected cost of the resulting states is minimized or (*b*) the least-cost path between the current node and the destination that uses only nontransition links. Because the path followed to a transition link $(i, j) \in \text{CTPS}(\Omega^k)$ is of no importance because, by definition, it cannot include any transition links, only the shortest path to each such transition link must be considered.

An outline of the general algorithm is provided next.

Main recursion:
$V(i^k, \Omega)$
Inputs: current node, current set of link-failure scenarios,
optimal value: $z^* \leftarrow \infty$, optimal solution: μ^*
find closest transition path set = CTPS
for all $\mu \in$ CTPS do
  if j^μ = destination then
    $z^{\text{temp}} \leftarrow c(\mu)$
    else
      Partition Ω:
      Set Ω_f = scenarios in which link (j^μ, l^μ) failed
      Set Ω_o = scenarios in which link (j^μ, l^μ) did not fail
      $z^{\text{temp}} \leftarrow c(\mu) + P(\Omega_f) V(0, \Omega_f) + P(\Omega_f) V(j, \Omega_o)$
    end if
    if $z^{\text{temp}} <$ optimal value then
      $\mu^* \leftarrow \mu$
      $z^* \leftarrow z^{\text{temp}}$
  end if
Output: z^*, μ

During each iteration, the set of reachable transition links is identified, and for each transition link, the value of the recursive function is calculated. The worst-case complexity of the algorithm can be determined with consideration of the fact that at most $|\Omega| - 1$ transition links will need to be visited in any one instance of the problem to establish the true network state, and as such, the depth of the recursion will be $O(|\Omega|)$. Because the set of reachable transition links will need to be calculated within each iteration, for a procedure that has complexity $O(n^2)$ and a set of at most m recursive function evaluations, the overall complexity of the algorithm will be $O(m^{|\Omega|-1} n^2)$.

HEURISTIC

This section presents a heuristic developed for the problem that allows tractability in much larger networks. The heuristic is based on the concept of determination of sets of network states under which a single-path strategy is feasible.

Definition 3. Define a set of states $T = \{\omega_1, \omega_2, \omega_3, \ldots, \omega_\gamma\}$ to be consistent if, for a source r and sink t, $\rho_T(r, t) \neq \varnothing$, that is, if at least one path connects r and t in all scenarios in T.

The heuristic presented in this section focuses on the use of consistent subsets of network states in a sequential aggregation procedure and by use of guaranteed paths in such subsets to obtain feasible solutions.

An intuitive view of the main concept behind the heuristic and then a rigorous definition of the algorithm are provided.

Motivation

The main idea behind the heuristic is that, for a given set of network states Ω, a feasible solution is a family of paths such that for each possible network state, at least one such path is feasible. As such, any feasible solution can be thought of as a partitioning of the set of possible scenarios into consistent subsets, with one path being assigned to each partition and the paths being followed in a specific order. The critical issue is how both the sequence of paths and the partitions are determined.

The MDP formulation presented in this section seeks the sequence of paths that minimizes the expected cost but addresses the partitioning only implicitly: the paths effect the partition. The main concept behind this heuristic is use of the procedure in the exact opposite way: the partitions are first determined, and then path policies are created to generate feasible solutions. The issue then becomes systematic generation of these partitions so that (*a*) the generation of partitions is efficient and (*b*) the generation of paths given the partitions is efficient. The focus of the heuristic presented in this

section is to choose the partitions so that the paths can be generated according to a simple shortest-path procedure. To guarantee this, the way in which the set of feasible network states is chosen to be partitioned is important. Specifically, the objective is to generate partitions such that the individual partitions form consistent sets, which allows the determination of simple paths by use of a shortest-path procedure.

Rather than an evaluation of possible families of paths that result in a feasible solution, the focus is on the family of static partitions (that is, partitions that do not depend on the path followed) that can be generated. Each partitioning implies that one path that is feasible for each partition will be determined and that the paths will be followed in the order of the partitions if link failures are encountered. Although this is not optimal, this procedure provides feasible solutions.

Heuristic Algorithm

This section describes the elements of the heuristic on the basis of the concepts described in the previous section and builds toward the full heuristic algorithm.

The algorithm can be divided into three parts: the state-partitioning algorithm, the path sequencing procedure, and the reoptimization of the path sequence algorithm. Because this is a heuristic and, as such, is meant to be significantly more tractable that the optimal algorithm, this focus is on a state-partitioning procedure that is computationally simple but that produces good results.

State Partitioning and Path Generation

The procedure developed to generate the state partitioning is a greedy, sampling-based algorithm that attempts to generate consistent subsets by sequential use of an aggregated network. The procedure starts with a subset consisting of a randomly chosen, feasible network state; other feasible network states are randomly added; and an aggregated network state is generated by use of an aggregating function. To ensure that a consistent subset is maintained, a check that a path exists in such a network state is performed. If no network state can be added such that the subset remains consistent, a new subset is constructed and this procedure is repeated until every feasible network state belongs to one subset. This ensures that the subsequent path-generating procedure will generate a feasible path for each network state.

With formalization of the procedure, the algorithm is described below.

State partitioning:
SP(Ω)
Inputs: set of feasible network states Ω
List of partitions Q, list of paths Θ
$k \leftarrow 0$
while $\Omega \neq \emptyset$ do
  $q^k = \{i\}$ where i is a random element chosen from Ω
  $\theta^k \leftarrow \emptyset$
    Remove i from Ω
    Set temp = Ω
    while temp $\neq \emptyset$ do
      Choose random network state $j \in$ temp
      Remove j from temp
      $\theta^k \leftarrow$ shortest path in $q^k \cup j$
      if path exists, then
        $q^k \leftarrow q^k \cup j$
      end if
    end while
    Add q^k to Q, and add θ^k to Θ
    Remove elements of q^k from Ω
    $k \leftarrow k + 1$
end while
Output: list of partitions Q, list of shortest paths Θ

The algorithm takes as input the set of feasible states Ω and outputs a partition, that is, a group of subsets q^k, where $\cup_{\forall k} q^k = \Omega$ and $q^i \cap q^j = \emptyset$, and an associated feasible path for such a partition. Once this sequence of partitions and paths is available, the sequence in which these paths will be followed must then be determined.

Path Sequencing

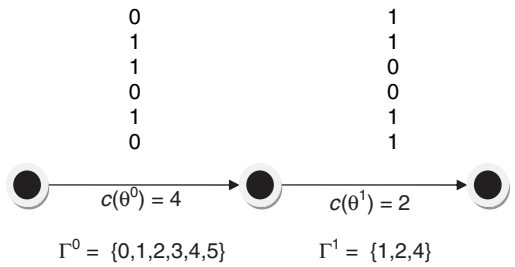
The three main properties of each partition are as follows:

1. Cardinality of the partition. In intuitive terms, a partition that has a large cardinality will provide a feasible path in a larger number of network states and will thus have a lower probability of route failure.
2. Failure profile of the associated path. Define the failure profile of path θ^k as the frequency with which each link in the path will be the first to fail under scenarios that are not q^k. A path in which its earlier links are the first to fail more frequently is desirable, as a shorter portion of the path will need to be traveled before it is recognized that a rerouting from the origin will be needed.
3. Path cost. Because the objective is to minimize expected costs, paths of lower cost are preferred.

The path sequencing heuristic presented in this section attempts to account for these three factors in the determination of a sequence of partitions and paths. The heuristic is based on estimation of the cost of a route failure on the basis of the cost of alternative paths and number of partitions, and then the path with the lowest cost of route completion plus route failure is chosen. When this procedure is performed one routing stage at a time, a path sequence can be developed.

Consider a routing stage u and an associated list of candidate partitions and paths Q^u such that the set of feasible network states in $\Omega^u \subseteq \Omega$, as some network states may have been ruled out in earlier routing stages. The estimate of the failure cost of each path is used as the cost of the worst-case scenario, that is, the sum of the costs of all other paths in the list of possible paths. This represents the event such that if a route failure is encountered, each subsequent path will be followed until its last link before a failure, which constitutes an upper bound, is encountered.

To determine the failure profile of a path θ^k, create a set Γ^i of network states corresponding to all network states in which link θ_i^k will be reached. To calculate this set for each i, initialize the set $\Gamma^0 = \Omega$; and in each iteration, record the magnitude of Γ^i, weight it by the cost of the link $c(\theta_i^k)$, and determine the set Γ^{i+1} by removal of the network states in which link θ_i^k does not exist. The sum of all these terms divided by the magnitude of Ω^u gives an expected cost of routing before failure over all states. An example of this procedure is shown in Figure 1.



$fp = \sum_{i=0}^{|\theta^q|} \Gamma^i c(\theta_i^q) = \Gamma^0 c(\theta_0^q) + \Gamma^1 c(\theta_1^q) = 6(4) + 3(2) = 30$

FIGURE 1 Example of failure profile estimation.

An estimate of the cost of each path can now be generated, and as such, the partition–path combination with the minimum heuristic cost, $\eta(u)$, can be chosen:

$$\eta(u) = \min_{q \in Q^u} \left\{ \frac{|q| c(\theta^q)}{|\Omega^u|} + \left(1 - \frac{|q|}{|\Omega^u|}\right) \sum_{j \in Q^u, j \neq q} c(\theta^j) + \sum_{i=0}^{|\theta^q|} \frac{r^i c(\theta_i^q)}{|\Omega^u|} \right\} \quad (6)$$

The formulation presented above is rewritten to provide a clearer picture of the terms:

$$\eta(u) = \min_{q \in Q^u} \left(\frac{|q| c(\theta^k) + (|\Omega^u| - |q|)(c_{\text{total}} - c(\theta^q)) + fp}{|\Omega^u|} \right) \quad (7)$$

$$fp = \sum_{i=0}^{|\theta^q|} \Gamma^i c(\theta_i^q) \quad (8)$$

$$c_{\text{total}} = \sum_{j \in Q^u} c(\theta^j) \quad (9)$$

The first term in the expression in Equation 6 is the cost of the path weighted by the probability that the path will be successfully traversed. The second term represents an upper bound on the remaining cost of routing, weighted by the probability that a link failure will be encountered. The final term represents the expected cost of routing before a link failure is encountered.

Start with u equal to 0 and Ω^0 equal to Ω. Then, calculate this value for each stage, fix the path, and then perform the same procedure on the next stage with a reduced number of partitions and paths. In the worst case, this will take $|\Omega|$ iterations, in which each iteration can be performed with polynomial worst-case complexity.

The formal algorithm is as follows:

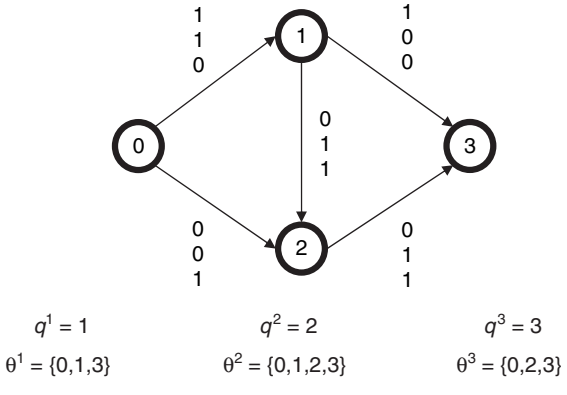
Path sequencing:
PS(Ω, Q, Θ)
Inputs: network states Ω, partitions Q, and paths Θ
$Q^0 \leftarrow Q$
$\Omega^0 \leftarrow \Omega$
Sequence of paths $\xi \leftarrow \varnothing$
for $u = 0$; $u < |Q^u|$; $u++$ do
  min = ∞, tempPath $\leftarrow \varnothing$
  sum $\leftarrow$ 0
  for $q \in Q^u$ do
    sum += $c(\theta^q)$
  end do
  for $q \in Q^u$ do
    $fp \leftarrow 0$
    for i to $|\theta^q|$ do
      $fp = fp + |\Omega| \, c(\theta_i^q)$
      $\Omega^{u+1} = \omega \in \Omega^u$: $\theta_i^q \in A^\omega$
    end for
    if $|q|c(\theta^q) + (|\Omega^u| - |q|)(\text{sum} - \theta^q) + fp$ then
      min = $|q|c(\theta^q) + (|\Omega^u| - |q|)(\text{sum} - \theta^q) + fp$
      tempPath $\leftarrow \theta^q$
    end if
  end for
end for
Output: ordered Q and Θ

Once this sequence of paths has been determined, the last step of the heuristic, namely, the reoptimization of the path sequence with an accounting for the real transitions given a specific path sequence, can be performed.

Path Sequence Reoptimization

To further optimize the previously developed path sequence, path sequence reoptimization is used with the aim to remove unnecessary paths from the sequence, that is, paths corresponding to partitions that will not be reached under the sequence of link failures established by the path sequence chosen. The reoptimization procedure is a recursive function that determines which links along each path will result in transitions and ensures that only paths with a positive probability of completion are used. An illustrative example is presented in Figure 2.

This step is required because the sequencing of paths presented in the previous section allows earlier paths to be feasible in more network states than just the partition for which the path was determined. The original partition–path combinations are sequenced in an order such that earlier paths cannot be feasible for network states in later partitions; but as this order is changed, this property no longer



If link {0,1} has failed, then the probability of being in partition 2 is 0. Therefore, path θ^2 is skipped, and path θ^3 is followed instead.

FIGURE 2 Example of path sequence reoptimization procedure.

holds and a possibility to improve the solution by elimination of paths that have a success probability of 0 (that is, paths that were meant for a specific partition, the network states for all of which are no longer feasible) exists. The algorithm is shown below:

Path sequencing reoptimization:
PSRO(s, Q, Θ, π)
Inputs: input state $s = (\Omega, \iota)$, network states Ω,
Inputs: partitions Q, paths Θ, current partition π
flag = false, UB $\leftarrow \infty$
if $\Omega \cap Q_\pi = \varnothing$ then
  UB = PSRO(s^k, Q, Θ, $\pi + 1$)
else
  $L = \Omega$
  $\rho \leftarrow \varnothing$
    for each link $(i, j) \in \Theta_\pi$ do
      UB+ = $c(i, j)|\Omega|$
        failureScenarios = scenarios in L where link (i, j) fails
        successScenarios = scenarios in L where link (i, j) does not fail
        if failureScenarios = $\varnothing$ then
          add link (i, j) to path ρ
          if j = destination then
            $\mu(s) \leftarrow \rho$
          end if
      else
        $\mu(s) \leftarrow \rho$
        UB+ = PSRO((failureScenarios, rootNode), Q, Θ, $\pi + 1$)
        $L \leftarrow$ successScenarios
        clear ρ UB/Ω
      end if
    end for
end if
Outputs: routing policy, routing policy cost

Full Algorithm

The advantage of the heuristic algorithm proposed here is that it can be implemented as a subroutine within the original MDP formulation. As the heuristic provides both an upper bound and a feasible solution to the problem, the authors chose to approximate both the solution vector and the value of the recursive function at any point in time by the value of the heuristic.

The full heuristic consists of performance of a predetermined number of iterations of the cycle of procedures listed in this section in which each iteration corresponds to a specific partition. Because the partitions are developed by use of a random choice of nodes, each iteration can potentially yield a new partition, a partition that may result in a better solution. The full algorithm is presented below:

Full heuristic algorithm:
FH(s^k)
Inputs: state s^k
UB $\leftarrow \infty$
for $j <$ numberIterations do
  Q, $\Theta \leftarrow$ SP(Ω^k)
  Q, $\Theta \leftarrow$ PS(Ω^k, Q, Θ)
  $z^{temp} \leftarrow$ PSRO(s^k, Q, Θ, 0)
  if $z^{temp} <$ UB then
    UB $\leftarrow z^{temp}$ UB
  end if
end for
return UB
Output: upper bound, routing policy

Heuristic as Upper Bound

The heuristic procedure developed in the previous section can easily be implemented within the MDP-based algorithm developed earlier. This can be done by determination of a specific number of transitions α before the evaluation of future recursions is stopped and instead an upper bound and policy generated by use of the heuristic are used. This variation of the algorithm is presented below:

Main recursion:
$V(s^k, \text{lev})$
Inputs: current node, current set of link-failure scenarios,
optimalValue: $z^* \leftarrow \infty$, optimalSolution: μ^*
if lev $< \alpha$ do
  find closest transition path set = CTPS
  for all $\mu \in$ CTPS do
    if j^μ = destination then
      $z^{temp} \leftarrow c(\mu)$
    else
      Partition Ω:
      Set Ω_f = scenarios where link (j^μ, l^μ) failed
      Set Ω_o = scenarios where link (j^μ, l^μ) did not fail
      $z^{temp} \leftarrow c(\mu) + P(\Omega_f)V((0, \Omega_f), \text{lev} + 1) + P(\Omega_f)V((j, \Omega_o), \text{lev} + 1$
    end if
    if $z^{temp} <$ optimalValue then
      $\mu^* \leftarrow \mu$
      $z^* \leftarrow z^{temp}$
    end if
  end for
else
  z^*, $\mu^* \leftarrow$ FH((i^k, Ω^k))
end if
Output: z^*, μ

NUMERICAL RESULTS

A comparison of numerical results is shown for three sets of randomly generated networks:

- $N = 25$ and $|\Omega| = 10$,
- $N = 25$ and $|\Omega| = 15$, and
- $N = 50$ and $|\Omega| = 5$.

The networks were randomly generated by use of a probability of 0.2 that a link exists between any two nodes and a probability of 0.2 that each link will exist under each particular scenario. The results for each network are shown in Table 1 and Figure 3. The number of iterations used for the heuristic was set at the number of feasible network states squared. The number of new partitions developed after that number was found to decrease significantly and did not improve the results significantly.

TABLE 1 Results for Three Sets of Randomly Generated Networks

Network Size	Statistic	Exact Solution Time (s)	Heuristic							
			$\alpha = 1$		$\alpha = 2$		$\alpha = 3$		$\alpha = 4$	
			Δ	Time (s)	Δ	Time (s)	Δ	Time (s)	Δ	Time (s)
$N = 25, \lvert\Omega\rvert = 10$	Mean	56.5	14.45	3.54	6.68	5.26	0.85	11.04	0.078	25.88
	SD	54.3	11.8	2.1	9.1	3.7	2.9	7.0	0.6	18.2
$N = 25, \lvert\Omega\rvert = 15$	Mean	694.40	16.58	9.04	4.87	14.60	1.00	24.84	—	—
	SD	1,465.61	14.92	4.67	7.05	9.62	2.80	19.16	—	—
$N = 50, \lvert\Omega\rvert = 5$	Mean	151.74	5.36	21.14	0.08	45.42	—	—	—	—
	SD	87.73	10.09	6.23	0.42	19.52	—	—	—	—

NOTE: Time represents the run time (in seconds), Δ represents the percentage of the value from the optimal solution, and α represents the number of steps before the heuristic is used to approximate the recursive function value. SD = standard deviation; — = not tested.

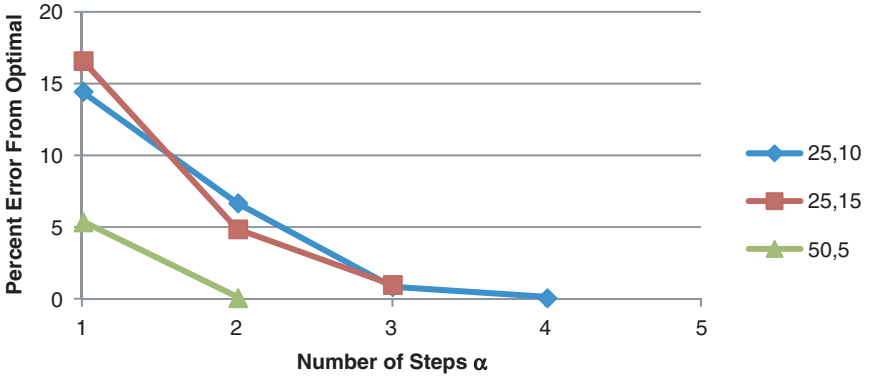


FIGURE 3 Results of numerical testing for different sets of randomly generated networks and heuristic parameter α.

The numerical results confirm that the heuristic not only requires significantly shorter computational times but also can provide extremely accurate results for relatively low values of the number of steps used in the heuristic. In particular, when the number of steps of the heuristic, α, is set at values of 2 or higher, the percentage from optimality approaches 0 in most cases.

DISCUSSION OF RESULTS AND CONCLUSIONS

The focus of this paper was to develop an MDP-based formulation for a shortest-path problem with link failures to show that specific properties of the SSN formulation allow a more efficient solution procedure. The paper presents a heuristic based on aggregation of network states to solve problems of more realistic size.

The numerical results show that the heuristic does indeed perform well and much more efficiently than exact solution of the dynamic programming formulation. Furthermore, a hybrid version of the algorithm, in which the heuristic is used to bound evaluation of subproblems along the dynamic program, can be adjusted to provide increasing levels of accuracy while still performing more efficiently than the exact method.

Through the development of the research presented in this paper, it became apparent that several natural extensions of the problem could pose interesting research problems.

This research assumed the availability of routing units such that no decision on the number of routing units to be deployed or the number of units to be allowed to fail from a cost standpoint must be made. A more realistic scenario is one in which a fixed cost is used to add a unit to the routing procedure and one in which an associated cost is used with a vehicle experiencing a link failure.

This research assumes a sequential routing of vehicles. A more complex, though interesting, problem is the parallel routing of multiple units. Although an MDP formulation of the problem can be seen to be an extension of that presented in this paper, the problem quickly becomes intractable. Whether variations of the heuristic presented in this paper are suitable for the multivehicle setting is an interesting research question.

REFERENCES

1. Dijsktra, E. W. A Note on Two Problems in Connection with Graphs. *Numerische Mathematik,* Vol. 1, 1959, pp. 269–271.
2. Sigal, C. E., A. A. B. Pritsker, and J. J. Solberg. The Stochastic Shortest Route Problem. *Operations Research,* Vol. 28, 1980, pp. 1122–1129.
3. Corea, G. A., and V. G. Kulkarni. Minimum Cost Routing on Stochastic Networks. *Operations Research,* Vol. 38, No. 3, 1990, pp. 527–536.
4. Corea, G. A., and V. G. Kulkarni. Shortest Paths in Stochastic Networks with Arc Lengths Having Discrete Distributions. *Networks,* Vol. 23, No. 3, 1993, pp. 175–183.

5. Miller-Hooks, E., and H. S. Mahmassani. Least Possible Time Paths in Stochastic, Time-Varying Networks. *Computers and Operations Research,* Vol. 25, No. 12, 1998, pp. 1107–1125.
6. Kulkarni, V. G. Shortest Paths in Networks with Exponentially Distributed Arc Lengths. *Networks,* Vol. 16, No. 3, 1986, pp. 255–274.
7. Frank H. Shortest Paths in Probabilistic Graphs. *Operations Research,* Vol. 17, No. 4, 1969, pp. 583–599.
8. Ziliaskopoulos, A. K., and H. S. Mahmassani. Time-Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications. In *Transportation Research Record 1408,* TRB, National Research Council, Washington, D.C., 1993, pp. 94–100.
9. Pretolani, D. A Directed Hypergraph Model for Random Time Dependent Shortest Paths. *European Journal of Operational Research,* Vol. 123, No. 2, 2000, pp. 315–324.
10. Waller, S. T., and A. K. Ziliaskopoulos. On the Online Shortest Path Problem with Limited Arc Cost Dependencies. *Networks,* Vol. 40, No. 4, 2002, pp. 216–227.
11. Polychronopoulos, G. H., and J. N. Tsitsiklis. Stochastic Shortest Path Problems with Recourse. *Networks,* Vol. 27, No. 2, 1996, pp. 133–143.
12. Loui, R. P. Optimal Paths in Graphs with Stochastic or Multidimensional Weights. *Communications of the Association of Computing Machinery,* Vol. 26, No. 9, 1983, pp. 670–676.
13. Eiger, A., P. B. Mirchandani, and H. Soroush. Path Preferences and Optimal Paths in Probabilistic Networks. *Transportation Science,* Vol. 19, No. 1, 1985, p. 75.
14. Mirchandani, P. B., and H. Soroush. Optimal Paths in Probabilistic Networks: A Case with Temporary Preferences. *Computers and Operations Research,* Vol. 12, No. 4, 1985, pp. 365–381.
15. Murthy, I., and S. Sarkar. A Relaxation-Based Pruning Technique for a Class of Stochastic Shortest Path Problems. *Transportation Science,* Vol. 30, No. 3, 1996, p. 220.
16. Mirchandani, P. B. Shortest Distance and Reliability of Probabilistic Networks. *Computers and Operations Research,* Vol. 3, No. 4, 1976, pp. 347–355.
17. Bertsekas, D. P., and J. N. Tsitsiklis. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research,* Vol. 16, No. 3, 1991, pp. 580–595.
18. Fu, L., and L. R. Rilett. Expected Shortest Paths in Dynamic and Stochastic Traffic Networks. *Transportation Research Part B: Methodological,* Vol. 32, No. 7, 1998, pp. 499–516.
19. Zhang, J., K. Zhu, B. Mukherjee, and H. Zang. Service Provisioning to Provide Per-Connection-Based Availability Guarantee in WDM Mesh Networks. *Proc., Optical Fiber Communications Conference, 2003.* IEEE, New York, 2004, pp. 622–624.
20. Modiano, E. H., and H. W. Lee. Diverse Routing in Networks with Probabilistic Failures. Proc., INFOCOM. IEEE, New York, 2009, pp. 1035–1043.
21. Andreatta, G., and L. Romeo. Stochastic Shortest Paths with Recourse. *Networks,* Vol. 18, No. 3, 1988, pp. 193–204.

The Transportation Network Modeling Committee peer-reviewed this paper.