1984

Optimal stochastic paths

Robert James Arnold
Iowa State University

INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International
300 N. Zeeb Road
Ann Arbor, MI 48106

Arnold, Robert James

OPTIMAL STOCHASTIC PATHS

Iowa State University PH.D. 1984

Optimal stochastic paths

by

Robert James Arnold

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major: Statistics

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1984

TABLE OF CONTENTS

1. INTRODUCTION

The work in this dissertation is a sequel and compliment to the investigations of Chern-Tarng Lin (1983). That work, in turn, was motivated by an applied probability problem.

One formulation of that problem involves an operator watching a control panel, e.g. in the control room of a power plant. The operator must scan the meters and dials to locate the source of any possible difficulty.

For each meter, an estimate is available of the probability that the operation it monitors will fail, and these failures are assumed to be independent. The problem is to decide in what order to scan the meters so as to minimize the expected time it takes to locate the first dial indicating a failed operation.

Lin saw that this problem could be treated as a close relative of the well-known traveling salesman problem, and solved it using dynamic programming.

He then generalized the problem to include the possibility of a countably infinite number of sites (meters or cities). In this setting, any infinite schedule (infinite ordered list) of cities has the same cardinality as the whole set, and is considered to be a feasible solution if it minimizes the expected cost (or time) of travel.

In our Chapter 2, we discuss Lin's theorems for the infinite-city case and provide a different model for the problem, a model which leads to another method of solution for the original problem.

That other method involves what we will call Minty's algorithm. In their book on scheduling, Conway, Maxwell, and Miller (1967) present Minty's algorithm as a solution to shortest route problems. We will discuss the algorithm in that setting in Chapter 3.

The origin of the algorithm is obscure. Conway, Maxwell, and Miller refer to a paper by Minty (1957) which does not, in fact, mention the algorithm. (It is a one paragraph note on the "knotted string" solution.) There is a mention of the algorithm in another reference cited by Conway, Maxwell, and Miller. The review article of Pollack and Wiebenson (1960), "Solutions of the Shortest-Route Problem--A Review," mentions the "highly efficient" algorithm of Minty. Their reference is to a private communication from Minty. (No date is given.)

A bit of additional muddying of the historical waters comes from Dreyfus (1969). He mentions the Pollack and Wiebenson attribution, but gives credit for the origin of the algorithm to Ford and Fulkerson (1958). In fact, a statement of an algorithm very similar to Minty's is found in Dijkstra (1959).

In any case, as we said earlier, we will give Minty the credit for the algorithm. Its importance to us is in the structure it provides for network problems.

Actually, it is the structure elucidated by the proof which is important. We see the connection between the way we discussed the Lin problem and the proof we provide for Minty's algorithm.

The two proofs of Minty's algorithm in Chapter 3 are new. There do not seem to be any published proofs. Our proofs were the springboard to new applications, both stochastic and non-stochastic.

The work with Lin's countable problem, including capture probabilities, leads to the development of the hazard function model we present in Chapter 4. Travel through the network becomes travel along curves in the plane. The discrete set of cities is replaced by points in the plane (the whole plane). At each point we are given a local conditional capture probability, the value of the hazard function. There is no history for a problem formulation of this sort.

We state two possible optimality criteria, one based on the probability of traveling further than a certain given distance and the other using the comparison of expected lifetimes on given paths.

In each case, the analysis uses a path perturbation technique from the calculus of variations.

In Chapter 2 and Chapter 3 we will be discussing network problems. Since there is some lack of uniformity in the definition of a network we provide the following definitions. (See Burr, 1982.)

A graph is an ordered triple (V, X, f) where V is a set of points, also called vertices or cities, X is a set of edges (or lines), and f is a function which associates two-element subsets of V with elements of X. We will restrict attention to graphs without self-loops, i.e. no edge connects a vertex with itself. These graphs are called multigraphs.

If the edges in X are associated (under f) with ordered pairs of vertices, the graph is a digraph. Note that a digraph may have an edge associated with (v_i,v_j) and another associated with (v_j,v_i). In discussing digraphs, we will often refer to edges as arrows. We call the edge (v_i,v_j) an arrow from v_i to v_j and speak of v_i as the node at the tail of the arrow and v_j as the node at the head of the arrow.

A walk from vertex v_1 to vertex v_k in a digraph is an alternating sequence of vertices and edges, v_1, e_1, v_2, ... , e_{k-1}, v_k such that $f(e_i) = (v_i,v_{i-1})$. We will also refer to walks as paths. If there is a walk from v_1 to v_n we say v_n is accessible from v_1.

A network is a graph or digraph along with a function which maps X into the real numbers. I.e. a network is a graph with a weight (length, cost) on each edge. In all of our work, we will assume these weights are positive real numbers.

2. THE STOCHASTIC TRAVELING SALESMAN PROBLEM IN A COUNTABLY INFINITE NETWORK

In his dissertation (Lin, 1983), Lin introduces a stochastic traveling salesman problem for a countable number of cities. We will state some of his definitions and results, then add a few of our own.

The problem involves a network with a countable number of nodes, $S = \{C_0, C_1, \ldots, C_n, \ldots\}$. They may be thought of as "cities" located in the plane, but we prefer to think of the nodes as representing possible states of a system.

The edges of the network have weights (costs, distances) associated with them. The weight on the edge from C_i to C_j is called d_{ij}. Thus d_{ij} represents the directed distance from C_i to C_j or the cost of changing from state C_i to state C_j. We assume $d_{ij} > 0$ for all i, j. There need not be an edge corresponding to every pair of nodes.

The traveling salesman problem in a finite network asks for a path of minimum cost which includes each node exactly once. We know that there are only a finite number of paths in a finite network. If there is a path which contains each node exactly once then there is a path which achieves the minimum cost.

Lin's stochastic traveling salesman problem in a finite network includes a probability, P_i, associated with each node. It is the probability of leaving that node assuming one has arrived there. The requirement is now for a scheduled path with minimum expected value which includes each node exactly once. That is, we write down a "schedule", some permutation of $C_1, C_2, \ldots, C_n$ preceded by C_0, then as we move through the list, a random mechanism at each node tells us whether to continue or stop at that node.

When the number of nodes is allowed to be either finite or countably infinite, one may wish to change the conditions for a path to be a solution. Lin chooses to minimize the expected value over paths whose set of nodes has the same cardinality as S. It would be possible for a solution to consist of a path which contains the nodes $C_0, C_2, C_4, \ldots, C_{2k}, \ldots$ and omits all the odd-numbered ones.

As Lin points out, "the set of countable infinite trips will largely consist of trips that do not involve all cities." If we let A be the set of all permutations of the integers 0, 1, 2, . . . and let B be the set of all countable ordered sets of integers, then B contains A.

If we identify paths with the elements of A and B in the obvious manner, then, since B is a larger set, we know the minimum expected cost over B must be less than the minimum expected cost over A.

For example in figure 1, with $d_{01} = 10$, all other weights equal to one, and $P_i = 1/2$ for all i, the optimal countable path $C_0\ C_2\ C_3\ \ldots$ has a smaller expected cost than $C_0\ C_1\ C_2\ C_3\ \ldots$, the best permutation path.

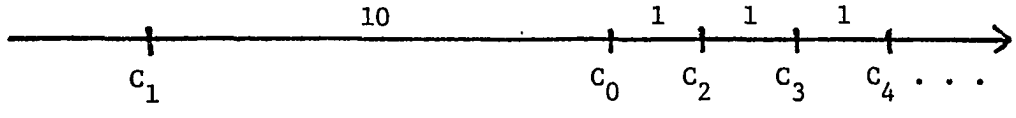

FIGURE 1. Comparing paths in A and B.

We would like to know the effect of using B for the set from which to select solutions, rather than A. In fact we can show, under conditions similar to those Lin used, that the minimum expected costs are the same over A as over B.

If we write $C_0, C_1^1, C_2^1, \ldots, C_k^1, \ldots$ for a particular path, designate the cost on the edge from C_i^1 to C_j^1 by d_{ij}^1, and write P_i^1 for the probability associated with node C_i^1, then we can write the formula for the expected cost as

$$E(\text{cost}) = \Sigma_{m=0}^{\infty}(\Pi_{i=0}^{m}P_i^1)d_{i,i+1}^1$$

We need to make some assumptions about the d_{ij}'s and P_i's:

1. $\inf d_{ij} \equiv d > 0$
2. $\sup P_i \equiv P^* < 1$
3. $d_{ij} < D$ for all i, j

Assumption 1 says the nodes do not get too close to each other in a cost sense, i.e. there is some lower limit to the cost of changing states. Assumption 2 implies that only nodes at which it is possible to be stopped ($1-P_i$ is the "capture" probability) are in the network. Assumption 3 is perhaps the most restrictive. It says there is an upper bound on the cost of changing from one state of the system to another. This assumption cannot hold if the weights on the edges are distances, all nodes are connected, and we assume $\inf d_{ij} > 0$.

Under these assumptions, the expected cost of any trip is finite:

$$\begin{aligned} E(\text{cost}) &= \Sigma_{m=0}^{\infty}(\Pi_{i=0}^{m}P_i)d_{i,i+1} \\ &\leq \Sigma_{m=0}^{\infty}(\Pi_{i=0}^{m}P_i)D \\ &= D\Sigma_{m=0}^{\infty}(\Pi_{i=0}^{m}P_i) \\ &\leq D\Sigma_{m=0}^{\infty}(P^*)^m \\ &= D(1 - P^*)^{-1}. \end{aligned} \tag{1}$$

Equation (1) holds for trips in B as well as trips in A.

Now we can show that the minimum expected cost over A is the same as over B, if all transitions between states are possible. Suppose a path with minimum expected cost in B is C_0, C_1^r, C_2^r, . . . , C_k^r . . . , which we refer to as trip r. Given $\varepsilon > 0$ we can find N such that the expected cost of any trip up to its Nth city is within ε of its total. That is, choose N so that

$$\begin{aligned} &\Sigma_{m=N+1}^{\infty}(\Pi_{i=1}^{m}P_i)d_{i,i+1} \\ &\qquad \leq D(P^*)^N(1-P^*)^{-1} \\ &\qquad < \varepsilon \end{aligned} \tag{2}$$

Given any integer N we can find a permutation trip which agrees with trip r up through node N. No matter which such permutation trip we select, the additional expected cost from N + 1 on cannot be greater than ε. Let p be a permutation trip which agrees with r through N. Then the expected cost of p is within ε of the expected cost of r.

We know the minimum over B is smaller than the minimum over A. The argument above shows that there are trips in A whose expected cost is within ε of the minimum in B. Since ε is arbitrary, we see the minimum over A must be the same as the minimum over B.

We showed that the expected cost of any trip is finite. Would that be a sufficient condition for the infimum over A to equal the infimum over B? Unfortunately no. What seems to be required is a uniform condition. We assumed a uniform bound on the d_{ij}'s and on the P_i's.

Lin assumes conditions 1 and 2 above, and limits feasible trips to those trips for which no two adjacent cities are more than D apart. Then he assumes that there exists a sequence of circular (in the cost metric) regions, Ω_n, centered at C_0, with radii tending to infinity, such that there is a feasible trip within each region.

Furthermore he specifies his "condition C". Condition C is said to hold for a sequence $\{\Omega_n\}$ of regions if for each Ω_k in $\{\Omega_n\}$ one may find an optimal feasible trip T_n which starts at some node C_i and contains all the cities of Ω_n except those in a set S_M. Furthermore some beginning part of T_n must be extendable to a trip in each Ω_k, $k > n$, and the number of cities in these extensions must tend to infinity.

Let $f_{\Omega_n}(C_i|S_M)$ denote the minimum expected cost for feasible trips starting at C_i and containing all cities in Ω_n except those in S_M.

With these definitions and assumptions in hand Lin proves his Lemma 3.1. Suppose the sequence of regions Ω_n, with radii r_n, satisfies condition C. Then given $\varepsilon > 0$ we can find a radius r_k, such that for any two larger radii, say $r_l > r_j \geq r_k$, the difference in the minimum expected costs in Ω_l and Ω_j is less than ε, i.e.

$$f_{\Omega_l}(C_i|S_M) - f_{\Omega_j}(C_i|S_M) < \varepsilon.$$

This result is similar to our equation (2).

As an alternative to Lin's expanding-circular-region approach one might instead compare the first n steps along each path, letting n go to infinity.

Fix attention on a particular trip, call it r. In order to make the notation more convenient, we assume the cities on this trip are C_0, C_1, C_2, . . . , i.e. renumber the cities if necessary.

Define an n step objective function.

$$f_n(r) = \Sigma_{m=0}^{n}(\Pi_{i=0}^{m}P_i)d_{i,i+1}$$

That is, $f_n(r)$ is the sum of the additional expected costs of the edges up to the nth edge. We also define

$$f(r) = \Sigma_{m=0}^{\infty}(\Pi_{i=0}^{m}P_i)d_{i,i+1}$$

i.e. f(r) is the expected cost of the path.

Our goal is to find a path which minimizes f(r). We might hope to do this by finding for each n, the path which minimizes $f_n(r)$. Then one might hope that

$$\inf\{f(r)\} = \lim \inf f_n(r) \tag{3}$$

where the infimum is taken over all feasible routes.

Suppose the set of feasible routes is R and on each r in R

$$\sup P_i = P^* < 1$$

$$d_{ij} < D \text{ for all } i, j$$

$$\inf f(r) = M < \infty$$

where the infimum is over all r in R. With these hypotheses, we can show (3) holds.

Since each n-city trip is part of an infinite trip

$$f(r) \geq f_n(r)$$

so that

$$\inf f(r) \geq \inf f_n(r)$$

and

$$\inf f(r) \geq \lim_{n\to\infty} f_n(r).$$

On the other hand, for each r,

$$\begin{aligned} f(r) &= f_n(r) + \Sigma_{m=n+1}^{\infty}(\Pi_{i=0}^{m} P_i)d_{i,i+1} \\ &\leq f_n(r) + D(P^*)^n(1-P^*)^{-1} \end{aligned}$$

Thus

$$\inf f(r) \leq \inf f_n(r) + D(P^*)^n(1-P^*)^{-1}$$

and

$$\begin{aligned} \lim\inf f(r) &= \inf f(r) \\ &\leq \lim\,[\inf f_n(r) + D(P^*)^n(1-P^*)^{-1}] \\ &= \lim\inf f_n(r) \end{aligned}$$

as was to be shown.

There are several ways one might wish to weaken the conditions under which we proved (3). Could we, for example, use the assumption $d_{ij} > d$ for all i, j rather than bounding the d_{ij}'s from above? In figure 2 we give an example which shows $d < d_{ij}$ is not sufficient to

ensure (3). For this network all the probabilities are 1/2. The d_{ij}'s are bounded below but not above. We set $d_{0,1} = 1$ and $d_{ij} = 1$ if i and j are both odd. The allowed routes are $r_1 = (C_0\ C_1\ C_2)$, $r_2 = (C_0\ C_1\ C_3\ C_4)$, . . . , $r_k = (C_0, C_1, C_3, \ldots, C_{2k-1}, C_{2k})$, . . . and the value of $d_{2k-1,2k}$ is chosen so that each route has expected cost 2.

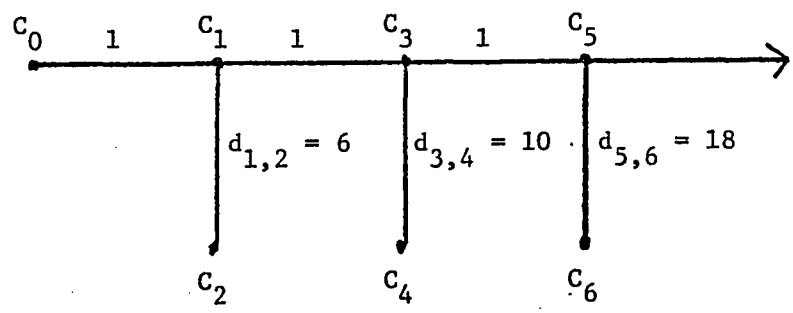

FIGURE 2. Example showing necessity of conditions on distances and probabilities.

Every allowed route ends at an even-numbered city and has expected cost 2. Thus inf f(r) = 2. If we fix n at, say, 1000 we see inf

$f_{1000}(r) < 1$ since routes like the one which ends at C_{4000} will contain only cities with odd subscripts through the first 1000 steps. The expected cost of those first thousand steps will thus be less than 1. In fact $\inf f_n(r) \leq 1$ for any n so $\lim \inf f_n(r) \leq 1$. Thus

$$\inf f(r) \neq \lim \inf f_n(r)$$

and one cannot approach the solution to the infinite problem through a sequence of finite problems.

One might think that bounding the d_{ij}'s above (as well as below) might lead to $\inf f(r) = \lim \inf f_n(r)$, but an upper bound is not enough, at least it is not sufficient if the P_i's are allowed to be 1. We modify the above example as in figure 3.

In this example each $d_{ij} = 1$. For odd-numbered cities $P_i = 1/2$ while for every other city $P = 1$. We add enough cities to the string below the odd-numbered cities so that the expected value of each route is again 2. The allowed routes go along through the odd-numbered cities until they descend along one even-numbered group.

Once again, for any n, there is a route for which the expected value along the first n steps is less than 1.

In Lin's objective function, $f_\Omega(c_i|S_M)$, the set S_M contains the cities one need not visit in the trip from C_i. They are the ones already included in an earlier portion of the trip.

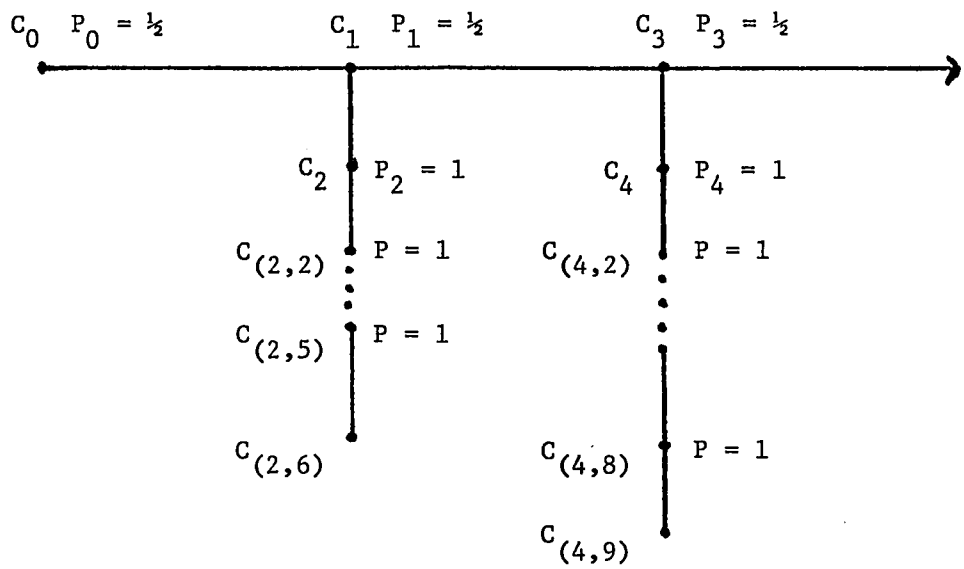

FIGURE 3. Second example of conditions on distances and probabilities.

We will think of this concept the other way around: which cities are available to visit next from C_i? This is the basis for our discussion of the set of "next nodes", N(n), in the next chapter.

3. STOCHASTIC NETWORK PROBLEMS AND MINTY'S ALGORITHM

3.1. Early Work with the Algorithm

As we mentioned in the introduction, there is a difference of opinion about who was first into print.

We will present the algorithm and show how it works. Then we give our proof and explain how the proof elucidates the fundamental structure of the network problem.

We will see that the proof (and hence the algorithm) applies to a much larger category of problems than the one we started with. At one stage, we have broad leeway in deciding which nodes come next. This leads to the realization that the Minty algorithm may be used to solve traveling salesman (visit each node once) as well as the shortest-route (minimize the cost of going from A to B) problem for which it was intended.

One more major step involves solving path-dependent problems. In the usual case, the weight or cost of an edge is fixed. We will solve problems in which the cost of an edge may depend on the route by which one arrived at that edge. This allows us to solve the type of stochastic network problems we pose.

In order to make our discussion more concrete, we will give an example of a shortest-route problem. Then we will give the algorithm and see how it works.

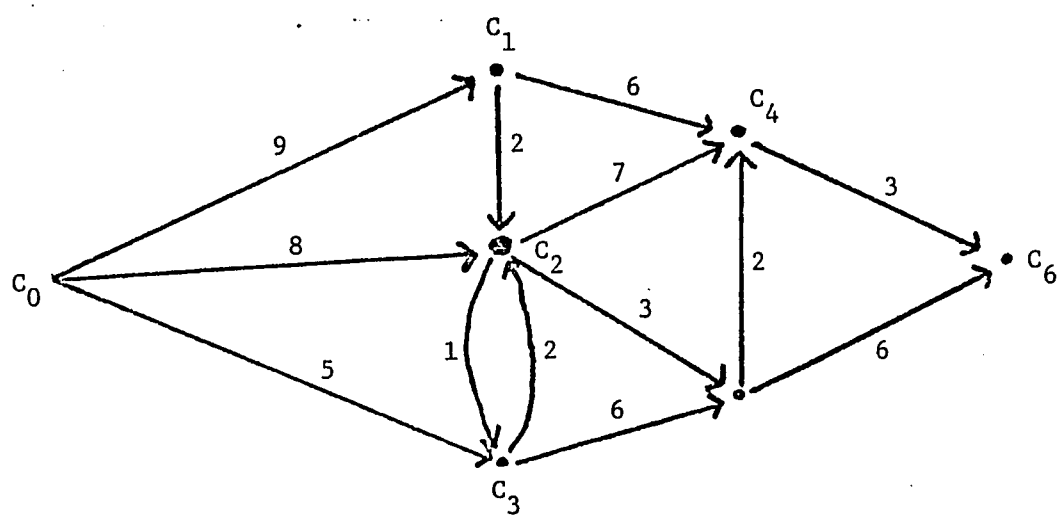

FIGURE 4. A simple network problem.

Consider the network in figure 4. We will think of the nodes as representing possible states of a system, e.g. location of a traveler. The arrows indicate which transitions are allowed: the state may change from C_1 to C_4 but not from C_4 to C_1. The weights on the edges are the costs of changing states, e.g. the cost of going from "traveler located in city 3" to "traveler located in city 5" is 6.

Notice that the cost of changing from C_i to C_j may not be the same as going from C_j to C_i. Perhaps it is "uphill" from C_3 to C_2 so travel in that direction costs more.

We start with a shortest-route problem in the network of figure 4. Our objective is to go from C_0 to C_6. As we go from one state to another we accumulate the cost of the edge which joins them. We want to minimize the sum of the costs, i.e. find a path from C_0 to C_6 which minimizes the total cost.

This is the problem Minty's algorithm was designed to solve. The algorithm may be stated as follows:

1. Label C_0 with zero and go to step 2.
2. Consider each edge whose tail node has been assigned a value but whose head node has no assigned value. For each such edge, add the weight on the edge to the value on the node.
3. Among all the sums formed in 2, select the minimum. That minimum value is associated with an edge, and a valueless node. Assign that minimum sum to the valueless node.

4. Repeat steps 2 and 3 unless the node to which a value was assigned in 3 is the target (destination), in which case, stop.

In order to see how the algorithm works, we will go through the steps for the network in figure 4.

First, C_0 is assigned the value zero (labeled with 0). Then, writing $i \rightarrow j$ for the edge from C_i to C_j, we consider $0 \rightarrow 1$, $0 \rightarrow 2$, and $0 \rightarrow 3$ in step 2. The sums are $0 + 9$, $0 + 8$, and $0 + 5$. Since the minimum, 5, goes with $0 \rightarrow 3$, we assign the value 5 to state C_3.

Now repeat step 2. The edges in the competition are $0 \rightarrow 1$, $0 \rightarrow 2$, $3 \rightarrow 2$, and $3 \rightarrow 5$. The corresponding sums are 9, 8, $5 + 2 = 7$, $5 + 6 = 11$. The minimum is 7, so C_2 is given the value 7.

Note that when we return to step 2, we need no longer consider edge $0 \rightarrow 2$ since both C_0 and C_2 have values. The edges to consider are $0 \rightarrow 1$, $2 \rightarrow 4$, $2 \rightarrow 5$, and $3 \rightarrow 5$, with sums 9, 14, 10, and 11. The minimum is 9, so C_1 is assigned a value of 9.

In a sense, we have started over in a new direction. The algorithm may require going back to C_0 or it could ask for a branch at some intermediate point along a path. This illustrates an unfortunate feature of the algorithm. We could go far out along one path and then be forced to return to C_0 and start out in a new direction.

Repeating step 2 would lead to labeling C_5 with 10, C_4 with 12, then C_6 with 15. Thus, the optimal route is C_0 C_3 C_2 C_5 C_4 C_6 and the minimum cost is 15.

We provide some additional notation which will set the stage for a proof of the algorithm. The algorithm is iterative, so we let n count the number of times we have applied step 2. Each time we go through steps 2 and 3 there are three characters which play the major roles. The graph consisting of (a) the nodes which have been assigned values and (b) the edges used in computing those values is called S(n). In the example, S(3) contains C_0, C_3, C_2, C_1 and the edges $0 \rightarrow 3$, $3 \rightarrow 2$, and $0 \rightarrow 1$. In figure 5 we list S(1) through S(3), and in figure 6 we give S(4) through S(6).

In step 2 we consider edges whose "tails" are labeled, but whose "heads" are not labeled. These edges, along with their "head" nodes are designated N(n).

The set of nodes not in S(n) or N(n) is the third character in our cast. Actually, it will be useful to consider the possibility that the nodes in the network are not all accessible from C_0. Label the set of non-accessible nodes UR. The set of nodes which are not in S(n) or N(n) or UR is called R(n).

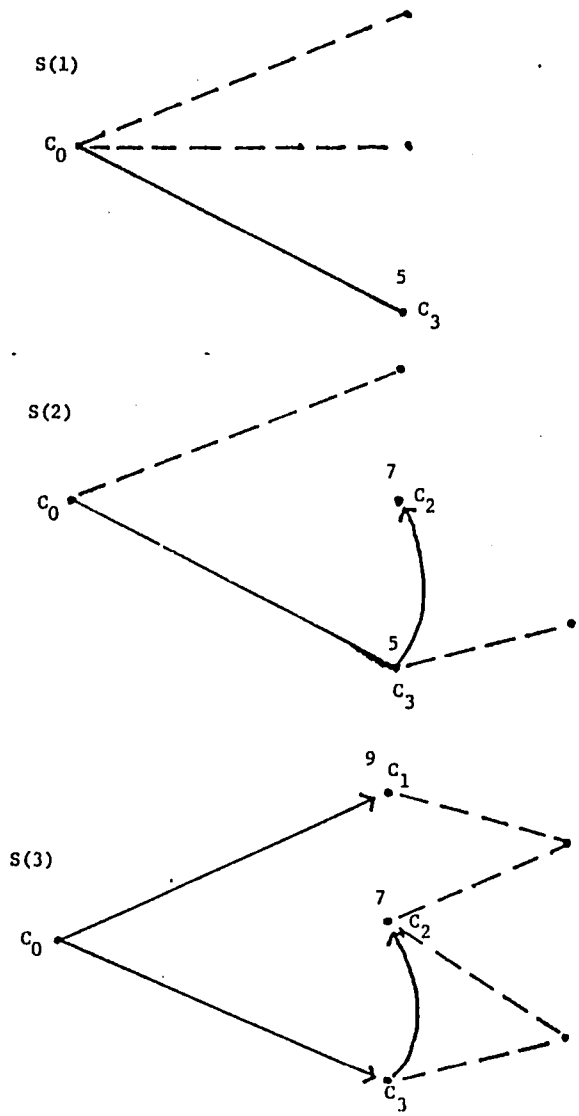

FIGURE 5. The graphs S(1) through S(3) for the sample problem.

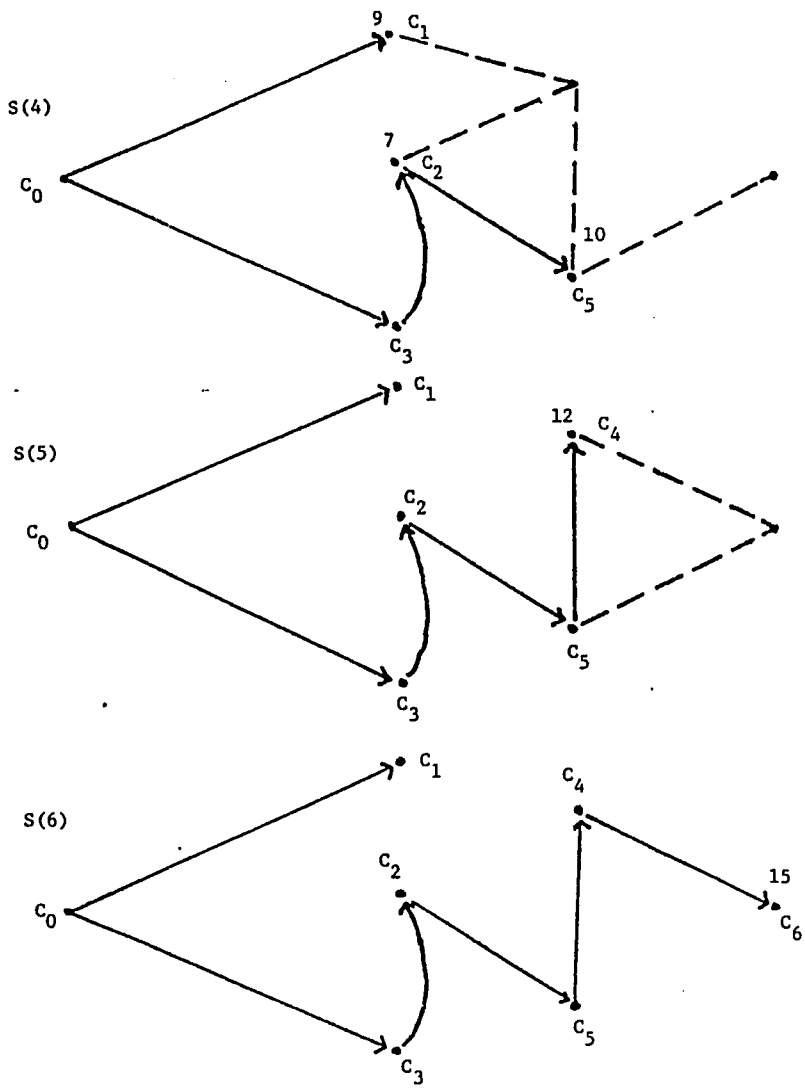

FIGURE 6. The graphs of S(4) through S(6) for the sample problem.

We are now ready to present our proof.

3.2. Proof of the Algorithm

We can paraphrase the operation of the algorithm in terms of our own notation. At each iteration of step 2 we select one edge (and its node) from N(n) to add to S(n).

We must show that algorithm selects edges appropriately, i.e. if the edge from C_i to C_j is selected. Then the value associated with C_j is the cost of a minimum cost path from C_0 to C_j (we would say "the" minimum cost path, but there may be ties).

There are two kinds of ties which may occur, each of which is easily dealt with. At stage n there may be two different arrows, say $i \rightarrow j$ and $k \rightarrow l$, each of which gives the minimizing value. We may randomly select one to enter S(n) at stage n. The other will be the minimizer at the next stage, and so will be added to S(n + 1).

The other kind of tie occurs when $i \rightarrow j$ and $k \rightarrow j$ both give a minimum total. In this case, we are indifferent as to which is added to S(n). Our only selection criterion is the total of the path weights. The value associated with C_j will be the same in either case. Once a

value is associated with C_j there is no effect on the future due to our choice of $i \to j$ or $k \to j$. Since we know how to deal with ties, we will not again mention tie-breaking procedures in what follows.

Note that there can only be one path from C_0 to any node in S(n). Once the "head" of an arrow is labeled, it is never again in N(n). There cannot be two arrows in S(n) whose heads are at the same node.

Two more notations will be needed. We need to refer to the sequence of cities on a particular path. We use a superscript to indicate the path and subscripts to indicate the order along that path, e.g. $C_1^0, C_2^0, \ldots, C_r^0, C_{r+1}^0$. We use d_{ij} for the weight on the edge $i \to j$ or d_{ij}^0 for the weight on the edge from C_i^0 to C_j^0.

The proof uses induction on the iteration counter n. Suppose that after n iterations the nodes in S(n) are labeled with minimum costs, and the arrows (edges) in S(n) are the components of minimizing paths. (We saw there could not be two arrows with heads at the same node of S(n); also, each node of S(n) is accessible from C_0, so from any node in S(n) there is a unique path to (and from) C_0.)

Now assume the edge $C_r^0 \to C_{r+1}^0$ is the one selected at this stage for addition to S(n). We must show that C_{r+1}^0 will be labeled with the minimizing value. One way to show this would be to compare the total cost of the (unique) path from C_0 to C_{r+1}^0 (through S(n) up to C_r^0) with the cost of any other path from C_0 to C_{r+1}^0.

We write $C_0^0 = C_0, C_1^0, C_2^0, \ldots, C_r^0, C_{r+1}^0$ for the route through S(n) (up to C_r^0) and then to C_{r+1}^0. Let $C_0^1 = C_0, C_1^1, C_2^1, \ldots, C_m^1, C_{r+1}^0$ be any other path from C_0 to C_{r+1}^0.

We must show

$$d_{0,1}^0 + d_{1,2}^0 + \ldots + d_{r,r+1}^0 \leq d_{0,1}^1 + d_{1,2}^1 + \ldots + d_{m,r+1}^1.$$

The route through S(n) is unique so the competitor must contain at least one node, other than C_{r+1}^0, which is not in S(n). But C_0 is in S(n), so we may proceed along the list of nodes in the competitor until we find the first one not in S(n), say C_{t+1}^1. This implies that the next node back along the path , C_t^1 is in S(n) (and all the nodes $C_0^1, C_1^1, \ldots, C_t^1$ are in S(n)). Furthermore C_{t+1}^1 must be in N(n): it is one edge away from a node in S(n).

The node C_{r+1}^0 must also be in N(n), in fact it was selected from among the nodes in N(n) to be the next one added to S(n). But that means C_{t+1}^1 and C_{r+1}^0 were both in the competition at iteration n. The algorithm specifies C_{r+1}^0 to add to S(n) so it must be the case that

$$d_{0,1}^0 + d_{1,2}^0 + \ldots + d_{r,r+1}^0 \leq d_{0,1}^1 + d_{1,2}^1 + \ldots + d_{t,t+1}^1. \tag{4}$$

The weights are all positive so equation (4) implies

$$d_{0,1}^0 + d_{1,2}^0 + \ldots + d_{r,r+1}^0 \leq d_{0,1}^1 + d_{1,2}^1 + \ldots + d_{t,t+1}^1 + \ldots + d_{m,r+1}.$$

The node C^0_{r+1} is assigned the value of the expression on the left-hand side of (4) and the edge and node are adjoined to S(n) to form S(n+1), which has the correct minimizing values for its nodes and contains the arrows forming minimum-cost paths.

To complete the induction we observe that the first iteration, from S(0) which contains only C_0, to S(1) which has one edge, must result in an S(1) which satisfies the induction hypothesis.

There is just one more detail needed to finish the proof. We must show that the target node is eventually labeled.

Of course if the target node is not accessible from C_0 it will never be labeled, so we assume it is accessible. Assume in addition that the set of nodes is finite. (In an infinite network, it would still be true that a labeled node would have its minimizing value.) Let us say there are k nodes and the target node is C_k.

Since C_k is accessible from C_0 there is a path from C_0 to C_k, say C_0, C^u_1, C^u_2, . . . , C_k. If, at iteration n, C_k is in S(n) then we are done. Suppose not; we will show the algorithm may be applied to add another node to S(n).

Just as before we may find the first node on the path from C_0 to C_k which is not in S(n). That node must be in N(n). As long as N(n) contains at least one node (and corresponding edge) the algorithm may be used to add a node and edge to S(n). Since the network contains a

finite number of nodes, every accessible node will eventually be given a value. This completes the proof of the algorithm.

As an immediate bonus from this proof, we see that the algorithm may be used to find the shortest route from C_0 to every other node.

3.3. A Stochastic Network Problem

The problem mentioned in the introduction, with an operator watching a control panel, may be modeled with a network which has probabilities associated with nodes.

The probability P_i associated with node C_i is the conditional probability that a trip which gets to C_i will continue on (in any direction).

Instead of finding the shortest (derministic) route from C_0 to C_k we now want the route with the minimum expected value. The difficulty is that if one is at node C_i, the additional expected cost of an arrow from C_i to C_j depends on the path by which one arrived at C_i.

For example consider the network in figure 7. We wish to find the path from C_0 to C_4 with minimum expected value.

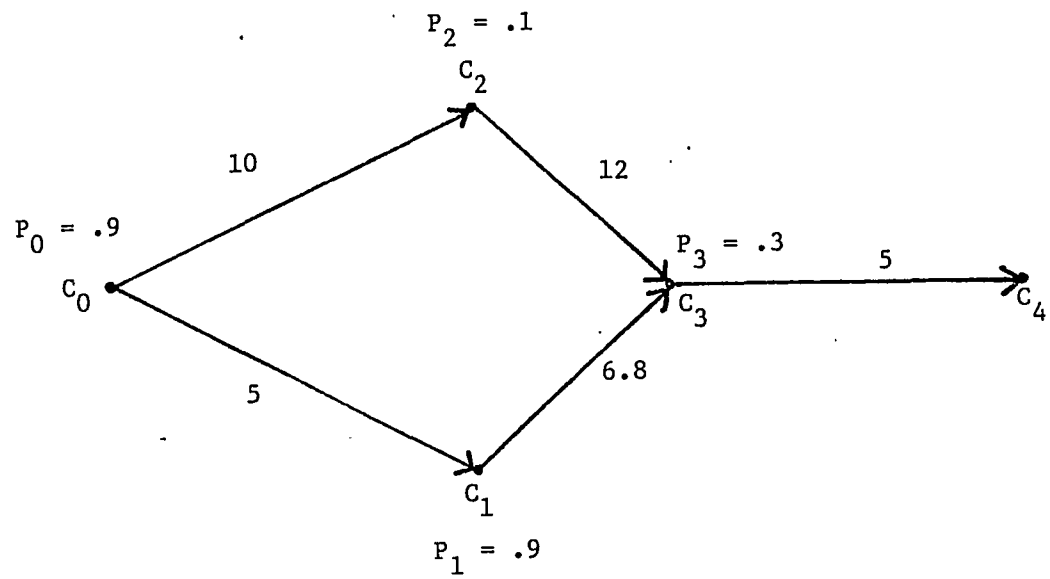

FIGURE 7. Path dependency in a stochastic network.

On the route C_0, C_2, C_3, C_4, the probability that we cross the edge $0 \rightarrow 2$ is 0.9 and its cost is 10, so the expected cost is 9. The probability that we leave C_2 given that we got there from C_0 is 0.1, so the probability of crossing the edge $2 \rightarrow 3$ is $(0.9)(0.1) = .09$. The expected value of the edge $2 \rightarrow 3$ on this path is $(.09)(12) = 1.08$. Similarly the additional expected cost of $3 \rightarrow 4$ is $(0.9)(0.1)(0.3)(5) = 0.135$.

On the route C_0, C_1, C_3, C_4 the additional expected cost of $3 \rightarrow 4$ is $(0.9)(0.9)(0.3)5 = 1.215$.

If we thought of using Minty here as we did in the deterministic case, but with expected costs replacing costs, we would fail to find the path with the minimum expected cost: S(1) would include C_0 and C_1, S(2) would include C_0, C_1, and C_2, and the next iteration would add edge $1 \rightarrow 3$. The path C_0, C_2, C_3 would not be in S(n). But C_0, C_2, C_3, C_4 is the path from C_0 to C_4 with minimum expected value.

We need to allow another path through C_3; there must be a way for C_3 to appear in N(n) at more than one iteration (or more than once in the same iteration as a component of different paths).

One way to solve this problem is to introduce the "multi-plane" idea. Suppose there are two ways of arriving at C_3. We make two copies of the network from C_3 on, and put them in two different planes, linked at C_3. See figure 8.

In general, if there are m paths from C_0 to C_i we think of m copies of the network from C_i "onward". See figure 9. As we iterate the algorithm we may have C_i^k in N(m) and C_i^l in N(r). When the target node is labeled, in any one of its aliases, we stop. In a listing of the path which gives the smallest expectation we do not, of course, need to mention the version number of a node.

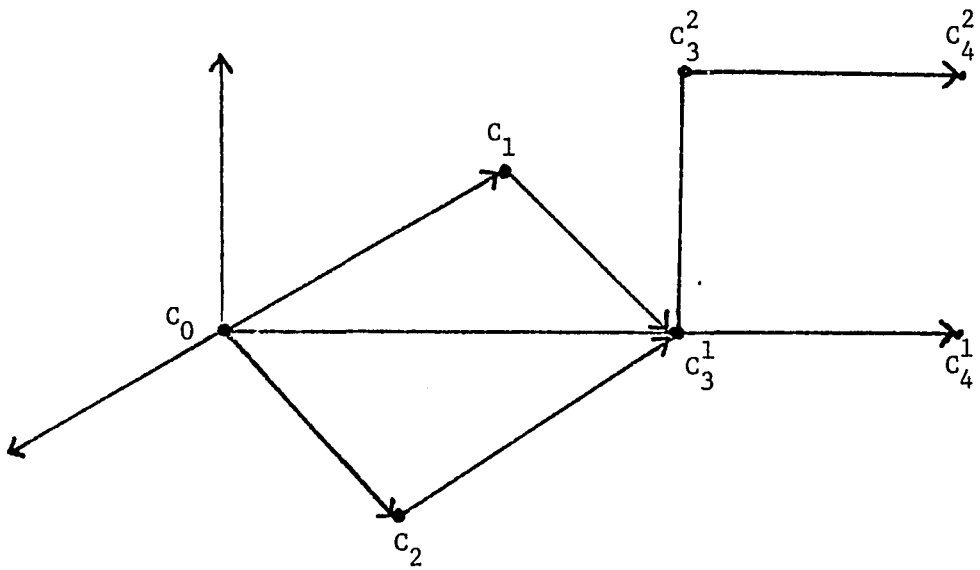

FIGURE 8. Two paths to C_3.

In applying the algorithm, the only new feature is the bookkeeping.

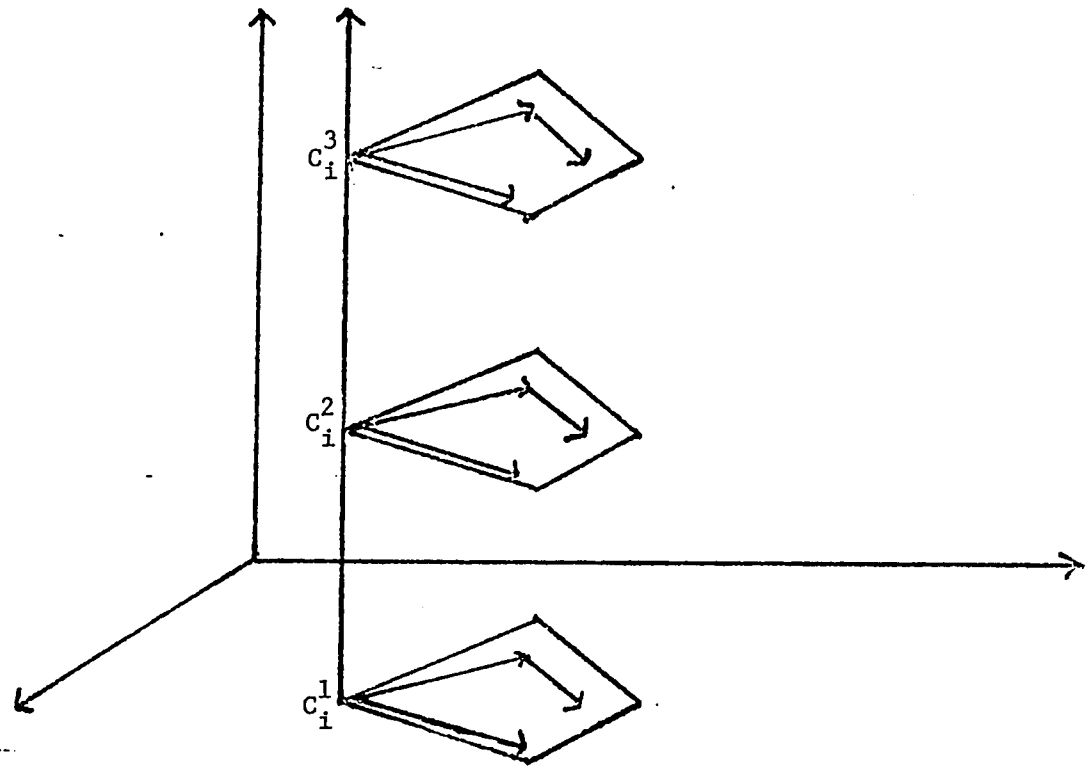

FIGURE 9. Multiple copies.

We need to keep track of the version number of a node to put in N(n) depending on the path. Otherwise, thinking of our multi-plane picture, and considering the different versions of a node to be different nodes, the algorithm works as before. The proof is the same too.

The multi-plane idea would seem to require a huge amount of additional work, mapping out all possible routes to each node. Indeed, it may. But we need not start out with all of the possible connections (or routes) listed, and typically a small portion of the possible connections s used in the computation. All we need to know is what to put in N(n) at the next step. The amount of branching actually required will depend on how dominant the winning route is.

The proof given earlier still works in this expanded, stochastic setting, but we will provide another proof in the next section which uses a different optimality criterion at each step.

3.4. Another Proof

In the proof given earlier, we showed that if C^0_{r+1} were the node to be added to S(n) at iteration n, then any route to C^0_{r+1} other than the one through S(n) must have higher cost.

Alternatively, we may construct a proof based on a different optimality criterion. Assume C^0_{r+1} is added to S(n) at iteration n. Then the (unique) path from C_0 to C^0_{r+1} in S(n) contains r+1 edges. Consider all other paths starting at C_0 which contain r+1 edges but

which are not entirely in S(n). No path in this set can have smaller cost than the path in S(n) to C^0_{r+1}.

If C^0_{r+1} is added to S(n) and completes a path, C_0, C^0_1, . . . , C^0_r, C^0_{r+1} of length r+1, then for every k, $0 < k \leq r+1$ we can find a sequence of arrows in S(n), of length k, starting at C_0.

Consider another path of length r+1, not completely contained in S(n): C_0, C^1_1, C^1_2, . . . , C^1_r, C^1_{r+1}. We show it has greater cost than C_0, C^0_1, . . . , C^0_r, C^0_{r+1}.

Now the argument is the same as before. On the competing path there is a first node not in S(n), say C^1_{m+1}. The cost of C_0, C^1_1, . . . , C^1_m, C^1_{m+1} is greater than the cost of C_0, C^0_1 , . . . , C^0_{r+1} since C^0_{r+1} was selected to enter S(n) at this stage.

So we know the path to C^0_{r+1} has a smaller cost than any other path of length r+1 or less. With this method of proof it is harder to see that we are finished when the target node enters S(n), but that is still the case.

This proof will also make it easier to show how to change the algorithm to solve traveling salesman problems.

3.5. Other Network Problems

Minty's algorithm was designed to solve shortest-route problems and that's how we have used it so far. But when we developed the multi-plane idea we saw that the algorithm could solve path-dependent problems.

Now we will show how to solve traveling salesman problems using Minty and the multi-plane idea. The key is simply to choose the set of successors, N(n), correctly. The traveling salesman problem in a finite network is as follows: find a path of minimum cost which contains each node exactly once.

When we presented the multi-plane concept in Section 3.3 we thought of each plane containing an identical copy of the connections from C_k "onward". But there is actually no need for the versions in the different planes to be identical. The connections in one plane may be different from the connections in another; in fact the weights on the edges may be different too.

Suppose C_k is in S(n). We want to see which of the edges connected to C_k should be included in N(n). There is a unique path from C_0 to C_k, say C_0, C_1^1, . . . , C_r^1, C_k. Our scheme is to include in N(n) each node whose tail is at C_k and whose head is not at one of the nodes C_0, C_1^1, . . . , C_r^1.

At iteration n we will consider each node of S(n) and select edges for inclusion in N(n) by the above scheme. Of course there is path dependency, so there may be many versions of a node in S(n).

The picture here is similar to the first multi-plane picture, except that the connections in one plane containing a version of C_k may be different from those in another version.

In figure 10 we show the diagram of a network. We picture the planes corresponding to various versions of C_2. There are six possible paths for the traveling salesman problem in this network:

1. $C_0\ C_2\ C_1\ C_3$
2. $C_0\ C_2\ C_3\ C_1$
3. $C_0\ C_1\ C_2\ C_3$
4. $C_0\ C_3\ C_2\ C_1$
5. $C_0\ C_1\ C_3\ C_2$
6. $C_0\ C_3\ C_1\ C_2$

If S(n) contains path 5 or 6 we are done. There are three paths leading to C_2 for which we need to consider its successors: C_0, $C_0\ C_1$, and $C_0\ C_3$. In the figure we see the three versions of C_2 and the connections in the planes corresponding to those versions.

The second proof of the algorithm is better suited to this situation than is the first. Suppose there are k nodes in the network. The first time there is a path k nodes long in S(n) we are done. The

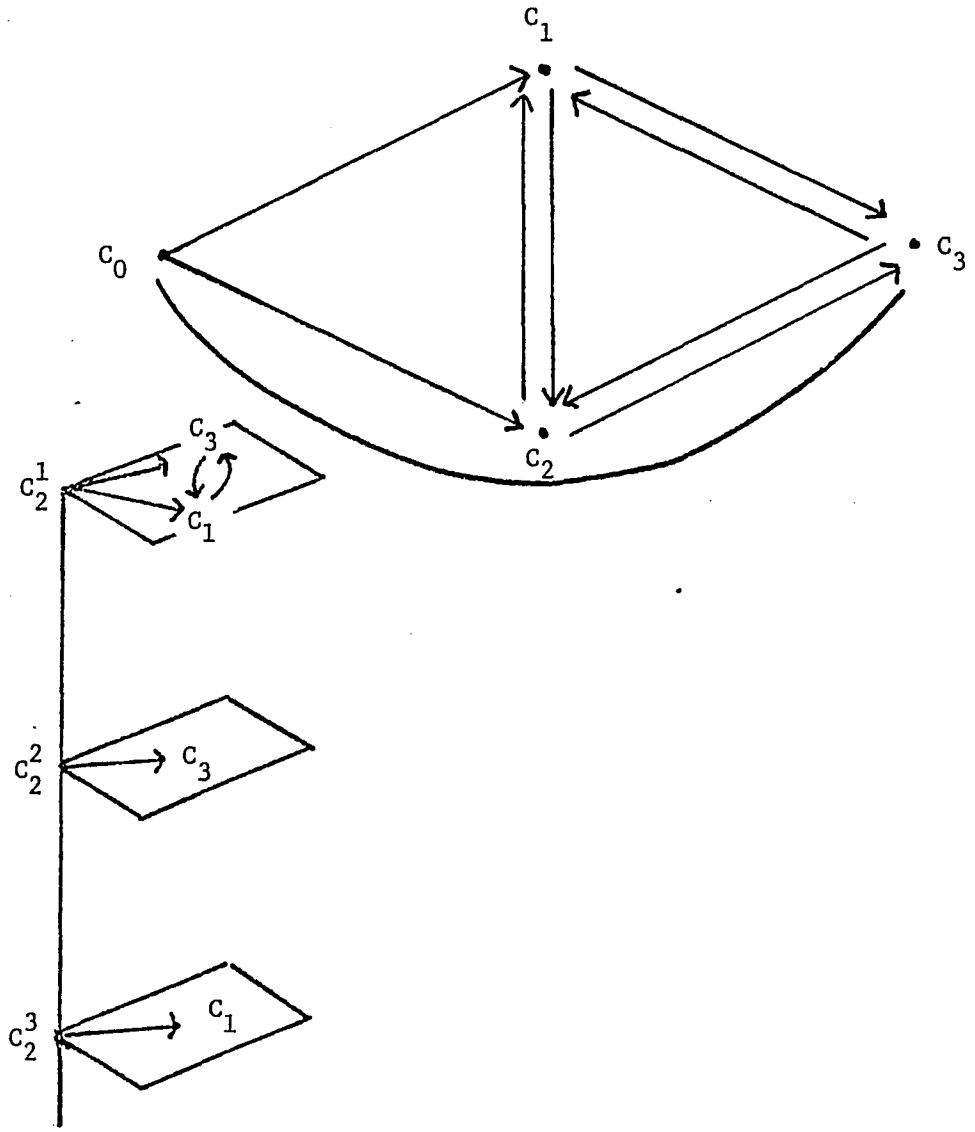

FIGURE 10. Paths from C_2 onward.

optimality condition of the second proof tells us at that point that there is no path of k or fewer nodes which has smaller cost. That is, we have a solution to the traveling salesman problem.

We now have enough machinery available to solve the problem mentioned in the introduction. Given a network with capture probabilities assigned to nodes, find the path which visits each node exactly once and has the minimum expected value.

Our traveling salesman formulation already contains path dependency so we need only assign the probabilities to the nodes and compute expected values of edges. Each version of a node is associated with a particular path from C_0 so there is a unique conditional probability of continuation associated with a particular version of a node.

Associating capture probabilities with nodes can radically change the solution. For the simple network in figure 11 the solution to the traveling salesman problem is $C_0\ C_2\ C_1\ C_3$.
If we assign capture probabilities of 0.9 to all nodes, i.e. $P_i = 0.1$ for all i, then the minimizing path is $C_0\ C_1\ C_3\ C_2$. The additional expected cost of the edge from C_3 to C_2 is $(0.1)(0.1)(0.1)100 = 0.1$. The cost of that edge is heavily discounted since it is so far from C_0. Thus in the stochastic problem we are quite willing to schedule large costs for edges far from C_0 since those costs are not likely to be realized.

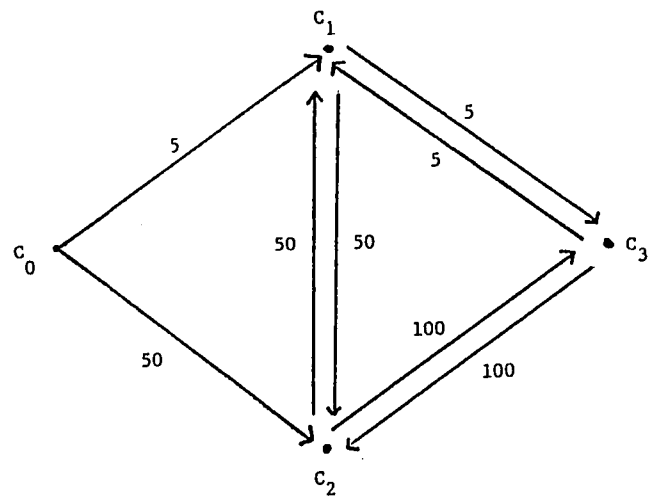

FIGURE 11. Discounting due to probabilities.

The network in figure 12 is adapted from an example in Lin (1983). The solution requires four iterations. The paths in S(n) at each stage are listed along with their expected costs. A minimum expected value of 81 is associated with the path $C_0\ C_2\ C_1\ C_3$.

The solution requires five additions and seven multiplications, compared to nine additions and nine multiplications for a dynamic programming solution.

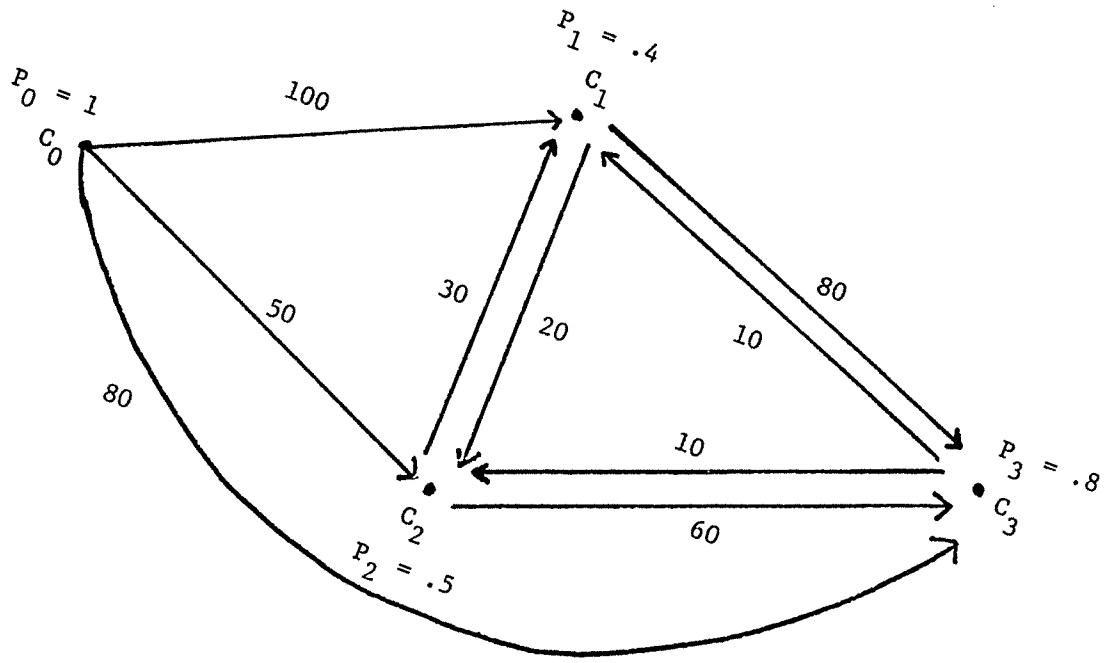

FIGURE 12. A stochastic traveling salesman problem.

We make no claims about the efficiency of Minty's algorithm in solving traveling salesman problems. (The term "efficiency" is used here as in the theory of complexity Karp (1975) and Cook (1971).) But the connection between shortest-route and traveling salesman problems becomes clear when the same basic algorithm is used to solve both. As we mentioned earlier, the difference in solutions is simply the way one decides what nodes to put in N(n) when working through the algorithm.

We have presented the traditional traveling salesman problem which requires visiting each node exactly once. Of course no real-life salesman would feel that that was an appropriate model. A salesman would be happier with a path of minimum cost which visits each node at least once.

Suppose, for example, that there is only one road to a certain city (i.e., it is at the end of a cul de sac). The stipulation that one may not backtrack could lead to excessive costs. Certainly a formulation of the problem which allows visiting cities more than once would be desireable.

With our algorithm, once we know how to put nodes in N(n) we will have a solution. We want to be allowed to visit cities more than once, but we need to avoid looping. If we put no restrictions on travel and try to apply Minty we may find ourselves going many times around a loop of low cost. This would not lead to an optimal solution.

This difficulty disappears if we allow return to nodes, but do not allow the same arrow to enter N(n) twice. One may go from C_i to C_j, then return to C_i (if there is an arrow) as long as the arrow $j \to i$ has not been traversed (on the unique path from C_0 to C_j).

4. CONTINUOUS MODELS

4.1. The Hazard Function Model

In Chapter 3, we dealt exclusively with discrete networks. Discrete probabilities were assigned to the nodes. It is natural to search for a continuous analog of that discrete problem.

This chapter presents a continuous analog based on the concept of a hazard function. We will describe the components of the model in sections 4.1 and 4.2, then derive conditions on optimal paths in section 4.3 and 4.4.

Instead of discrete nodes, we now consider points in the plane. Each point plays the role of a city in Lin's model. The idea of a path containing a countable number of cities corresponds with a path in the plane of infinite arc length. In the discrete network a value of $1 - P_i$ told us how likely we were to stop at a particular node. In our new model, we use the idea of a hazard function to assign a probability of stopping near a point.

4.1.1. The Hazard Function on the Plane

If we think of movement through the networks of Chapters 2 and 3 as corresponding to the life of a system in time, we might think of the capture probabilities, $1 - P_i$, as probabilities of system failure.

This reminds one of the terminology of reliability. If the time at which a system fails is governed by some random mechanism, we can define a lifetime random variable, T, for the system. (See Lawless, 1982.) We could specify the distribution of T by giving its distribution function or its density if the density exists. A third possibility, which is preferred in applications, is to specify a hazard function. It gives the instantaneous failure rate at any time t, i.e. the probability of failure in the next small increment, Δ, of time, given there has not been a failure up to t is approximately $h(t)\Delta$ where

$$h(t) \equiv f(t)[1-\int_0^t f(x)dx]^{-1}.$$

We want to define a hazard function as a space function rather than a time function, i.e. a function of (x,y,) rather than t. We use the hazard function h(x,y) at a point (x,y) as follows: the probability of failure in the next small segment ds of arc length on any curve through (x,y), given no failure along the curve from the origin up to (x,y), is approximately h(x,y)ds.

Consider a particle moving in the plane starting at (0,0) at time 0. We use the hazard function h(x,y) in the following way: given the hazard function in the plane, and any curve Γ starting at the origin, we define a stopping time T as a non-negative random variable such that the distribution function is given by

$$P(T \leq s) = 1 - \exp\{-\int_0^s h_\Gamma(t)dt\}.$$

Note that this agrees with the probabilistic interpretation of the hazard function given earlier.

We will consider the following two families of path functionals: (1) for each s_0, the distribution function $P(T \leq s_0)$ and (2) the expected value of T. In section 4.3 we discuss the first family and in section 4.4 the second. The problem is to choose the path, Γ, to extremeize the appropriate path functional. The optimal curve Γ is characterized by equations (18) and (19) of section 4.3.1 for (1) and by equations (26) and (27) of section 4.4.1 for (2).

4.2. The Perturbation Technique for Path Functionals

4.2.1. Paths in the Plane

In this section we develop the perturbation technique for path functionals (Ewing, 1969). This comes from the calculus of variations. We apply it to our two particular functionals in sections 4.3 and 4.4.

In the following sections, when we refer to a curve or path Γ, we mean a curve which "starts" at the origin. That is, any parameterization of Γ, say $x(t)$, $y(t)$, $0 \leq t < \infty$, will satisfy $x(0) = 0$, $y(0) = 0$. We restrict attention to curves such that $x(t)$ and $y(t)$ have continuous derivatives on $(0,\infty)$. It is possible to extend our arguments to "admissible" curves (Dreyfus, 1965), i.e. continuous curves made up of a finite or countable number of segments on each of which the tangent turns continuously.

We will typically assume that a curve Γ is parameterized in terms of its arc length (Ewing, 1969). This will simplify several expressions which occur frequently. Also we see that if Γ has arc length parameterization $x(s)$, $y(s)$ then, using $Dx(s)$ for the derivative of $x(s)$,

$$[Dx(s)]^2 + [Dy(s)]^2 = 1$$

which implies

$Dx(s) \leq 1$

$Dy(s) \leq 1$ for all s

and

$x(s) \leq s$

$y(s) \leq s$ for all s.

4.2.2. Perturbed Curves

Suppose we wish to show that the curve Γ_0 minimizes the value of some path functional $F(\Gamma)$ over a class C of curves Γ. One way to procede is to perturb Γ_0 with another curve Γ_1.

Assume Γ_0 has parameterization $x_0(t)$, $y_0(t)$, and Γ_1 has parameterization $x_1(t)$, $y_1(t)$. For each t we add ε times the coordinates on Γ_1 to the coordinates on Γ_0.

Call the resulting curve Γ^*. We write $\Gamma^* = \Gamma_0 + \varepsilon\Gamma_1$ for

$$x^*(t) = x_0(t) + \varepsilon x_1(t)$$

$$y^*(t) = y_0(t) + \varepsilon y_1(t)$$

With Γ_0 and Γ_1 fixed, one may consider $F(\Gamma^*)$ to be a one variable function of ε. If Γ_0 gives a minimum of F, then $dF/d\varepsilon$ must be 0 at $\varepsilon = 0$. Notice that this is the directional derivative of the functional $F(\Gamma)$ at Γ_0 in the direction of Γ_1. If Γ_0 is to be optimal then this

must hold for all Γ_1. In many cases of interest this leads to a unique solution for Γ_0.

4.2.3. The Arc Length Function

We define a function which gives arc length on $\Gamma^* = \Gamma_0 + \varepsilon\Gamma_1$ as a function of s, arc length on Γ_0. Both Γ_0 and Γ_1 are parameterized in terms of their arc lengths. If we write

$$x^*(s) = x_0(s) + \varepsilon x_1(s)$$

$$y^*(s) = y_0(s) + \varepsilon y_1(s)$$

then $x^*(s)$, $y^*(s)$ is not necessarily a parameterization of Γ^* in terms of its own arc length.

For example, suppose Γ_0 is the half of the line $y = x$ in the first quadrant:

$$x_0(s) = s/2^{1/2}$$

$$y_0(s) = s/2^{1/2} \qquad 0 \le s \le \infty$$

Let Γ_1 be the positive x - axis:

$$x_1(s) = s$$

$$y_1(s) = 0 \qquad 0 \le s \le \infty$$

See figure 13. Each curve is parameterized in terms of its own arc length. We have used the same symbol, s, for the parameter in each

case. Thus, when we form Γ^*, we go out a distance s along Γ_0 and compute the coordinates $x_0(s)$, $y_0(s)$. Then go out the same distance s along Γ_1, and compute the coordinates $x_1(s)$, $y_1(s)$. Thus, $(x_0(s), y_0(s))$ is perturbed by ε times the coordinates of a point which is also an arc length s from the origin. This is not a restriction on the class of competitors or on the curves Γ^*. Returning to our example, we see

$$x^*(s) = x_0(s) + \varepsilon x_1(s) = s/2^{1/2} + \varepsilon s$$

$$y^*(s) = y_0(s) + \varepsilon y_1(s) = s/2^{1/2}$$

This parameterization is not in terms of arc length. If $s = 1$, then $x^* = 2^{-1/2} + \varepsilon$, $y^* = 2^{-1/2}$ and this point is at arc length $(1 + \varepsilon 2^{1/2} + \varepsilon^2)^{1/2}$.

The element of arc length on Γ^* is

$$ds^* = [(Dx^*)^2 + (Dy^*)^2]^{1/2}\, ds.$$

We define $u(s,\varepsilon)$ to be the function which gives the arc length on Γ^* corresponding to an arc length s on Γ_0 (or Γ_1). (Consider Γ_0 and Γ_1 to be given.) Thus,

$$\begin{aligned} u(s,\varepsilon) &= \int_0^s \{[Dx^*(t)]^2 + [Dy^*(t)]^2\}^{1/2}\, dt \\ &= \int_0^s \{[Dx_0(t) + \varepsilon Dx_1(t)]^2 \\ &\qquad + [Dy_0(t) + \varepsilon Dy_1(t)]^2\}^{1/2} dt \end{aligned} \tag{5}$$

In our example we have

$$u(s,\varepsilon) = \int_0^s \{(2^{-1/2} + \varepsilon)^2 + (2^{-1/2})^2\}^{1/2} dt$$

or

$$u(s,\varepsilon) = s(1 + \varepsilon 2^{1/2} + \varepsilon^2)^{1/2}.$$

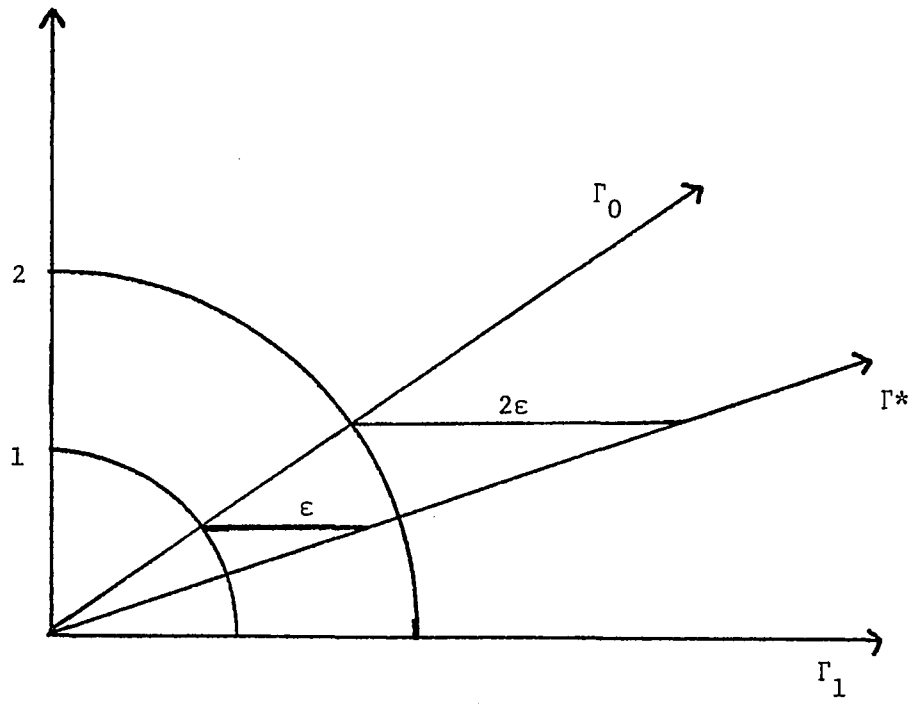

FIGURE 13. The perturbed curve.

We will often need to go the other direction too, i.e., find the arc length to the point on Γ_0 which corresponds with the point at arc length s^* on Γ^*. We will write $v(s^*,\varepsilon)$ for this inverse function. That is, $v(s^*,\varepsilon)$ is the function satisfying

$$u(v(s^*,\varepsilon),\varepsilon) = s^*.$$

In the example we have $v(s^*,\varepsilon) = s^*(1 + \varepsilon 2^{1/2} + \varepsilon^2)^{-1/2}$

4.2.4. Derivatives of the Arc Length Function

We will need to know partial derivatives of $u(s,\varepsilon)$ when we discuss optimization. The derivative operator will be denoted by D, e.g. Dx_0 is the derivative of x_0 (with respect to the parameter, usually t or s.)

The curves Γ_0 and Γ_1 are fixed for the present. First we compute the partial of $u(s,\varepsilon)$ with respect to s:

$$\begin{aligned} \partial u(s,\varepsilon)/\partial s &\equiv u_1 \\ &= \partial/\partial s\{\int_0^s [(Dx^*)^2 + (Dy^*)^2]^{1/2} dt\} \\ &= \{[Dx^*(s)]^2 + [Dy^*(s)]^2\}^{1/2} \end{aligned} \tag{6}$$

If we evaluate at $\varepsilon = 0$ we have

$$\begin{aligned} u_1|_{\varepsilon = 0} &= \{[Dx_0]^2 + [Dy_0]^2\}^{1/2} \\ &= 1 \end{aligned} \tag{7}$$

since Γ_0 is parameterized in terms of its arc length. This partial evaluated at $\varepsilon = 0$ will equal one for any value of s, in particular for $s = v(s^*,\varepsilon)$:

$$u_1|_{(v(s^*,0),0)} = 1. \tag{8}$$

Next we find the partial of $u(s,\varepsilon)$ with respect to ε. Our curves were assumed to be "nice" enough that the interchange of derivatives and integrals is allowed. First we note that

$$\begin{aligned} \partial/\partial\varepsilon\{[Dx^*]^2 + [Dy^*]^2\} &= \partial/\partial\varepsilon\{[Dx_0 + \varepsilon Dx_1]^2 + [Dy_0 + \varepsilon Dy_1]^2\} \\ &= \partial/\partial\varepsilon\{(Dx_0)^2 + 2\varepsilon Dx_0 Dx_1 + \varepsilon^2 (Dx_1)^2 \\ &\quad + (Dy_0)^2 + 2\varepsilon Dy_0 Dy_1 + \varepsilon^2 (Dy_1)^2\} \end{aligned}$$

$$=2[Dx_0Dx_1 + Dy_0Dy_1 + \varepsilon(Dx_1)^2 + \varepsilon(Dy_1)^2].$$

Now we compute

$$\begin{aligned}
&\partial u(s,\varepsilon)/\partial\varepsilon \equiv u_2 \\
&\quad = \partial/\partial\varepsilon \int_0^s \{[Dx^*]^2 + [Dy^*]^2\}^{1/2} dt \\
&\quad = \int_0^s \partial/\partial\varepsilon \{[Dx^*]^2 + [Dy^*]^2\}^{1/2} dt \\
&\quad = \int_0^s 1/2\{[Dx^*]^2 + [Dy^*]^2\}^{-1/2} \partial/\partial\varepsilon\{[Dx^*]^2 + [Dy^*]^2\} dt \\
&\quad = 1/2 \int_0^s \{[Dx^*]^2 + [Dy^*]^2\}^{-1/2} 2[Dx_0Dx_1 + Dy_0Dy_1 \\
&\qquad + \varepsilon(Dx_1)^2 + \varepsilon(Dy_1)^2] dt. \qquad (9)
\end{aligned}$$

If this is evaluated at $\varepsilon = 0$, we have

$$u_2|_{\varepsilon = 0} = \int_0^s \{[Dx_0]^2 + [Dy_0]^2\}^{-1/2} \{Dx_0Dx_1 + Dy_0Dy_1\} dt.$$

Once again we make use of the parameterization of Γ_0 in terms of arc length: $[Dx_0]^2 + [Dy_0]^2 = 1$. So

$$u_2|_{\varepsilon = 0} = \int_0^s \{Dx_0Dx_1 + Dy_0Dy_1\} dt \qquad (10)$$

and also

$$u_2|_{(v(s^*,0),0) = 0} = \int_0^{s^*} \{Dx_0Dx_1 + Dy_0Dy_1\} dt \qquad (11)$$

Now it's easy to get $\partial^2 u/\partial\varepsilon\partial s$:

$$\begin{aligned}
&\partial^2 u/\partial\varepsilon\partial s = \partial/\partial s(\partial u/\partial\varepsilon) \equiv u_{12} \\
&\quad = \partial/\partial s \int_0^s \{[Dx^*]^2 + [Dy^*]^2\}^{-1/2} 2[Dx_0Dx_1 + Dy_0Dy_1 \\
&\qquad + \varepsilon(Dx_1)^2 + \varepsilon(Dy_1)^2] dt \\
&\quad = \{[Dx^*(s)]^2 + [Dy^*(s)]^2\}^{-1/2} [Dx_0Dx_1 + Dy_0Dy_1 \\
&\qquad + \varepsilon(Dx_1)^2 + \varepsilon(Dy_1)^2].
\end{aligned}$$

Evaluate at $\varepsilon = 0$:

$$u_{12}|_{\varepsilon = 0} = [Dx_0 Dx_1 + Dy_0 Dy_1] \tag{12}$$

We can relate the partial of $u(s,\varepsilon)$ with respect to ε to the partial of $v(s^*,\varepsilon)$. Since

$$u(v(s^*,\varepsilon),\varepsilon) = s^*$$

we have, taking partials with respect to ε,

$$(\partial u/\partial s)(\partial v/\partial \varepsilon) + (\partial u/\partial \varepsilon) = 0$$

or

$$u_1 v_2 + u_2 = 0.$$

Evaluate at $\varepsilon = 0$ and use (7) and (8) from page 50:

$$\partial v(s^*,\varepsilon)/\partial \varepsilon|_{\varepsilon = 0} = -u_2|_{(v(s^*,0),0)}$$
$$= -\int_0^{s^*} \{Dx_0 Dx_1 + Dy_0 Dy_1\} dt. \tag{13}$$

4.3. Stochastic Ordering

Suppose for the moment that we have only two allowed paths, Γ_i and Γ_j. We are thinking of a traveler starting at the origin and moving along a fixed, preselected path. The traveler is stopped at some random time in accordance with the given hazard function.

The random variable T is the distance (or time, since we assume travel at unit speed) the traveler goes. Recall that we want to select a path to minimize the time (or distance) traveled.

For some systems, one may know that some initial time segment is crucial, e.g., the first three minutes after a warning signal. In such a case we can reasonably choose between Γ_i and Γ_j based on the probability of traveling more than 3 units.

If on Γ_i the probability of traveling more than 3 units is less than the corresponding probability on Γ_j, then Γ_i is preferred to Γ_j. That is, if

$$P(T_{\Gamma_i} > 3) < P(T_{\Gamma_j} > 3)$$

(where T_{Γ_i} and T_{Γ_j} are the stopping times on Γ_i and Γ_j respectively), then Γ_i is preferred to Γ_j.

More generally, if we have some fixed time/distance, s_0, at which we wish to make comparisons, and if Γ_0 is such that

$$P(T_{\Gamma_0} > s_0) \leq P(T_{\Gamma} > s_0)$$

for every other curve Γ, then we will call Γ_0 optimal with respect to a stochastic ordering at s_0.

Our optimality criterion is that this must hold for each s_0. This optimality criterion also makes sense in relation to a theorem of Lin's (1983). Lin showed in his Lemma 3.1 (p.37) that one can find a distance beyond which the expected trip lengths cannot change by more than ε.

Essentially, this is due to probability accumulating as we go further along a path. In our hazard function model this kind of theorem would exist if $h(x,y) \geq h^* > 0$ (except possibly on some set whose intersection with every curve has arc length measure zero). We may well be able to specify an ε such that we would be happy with any trip whose expected length is not more than ε greater than the best (smallest) possible.

From our discussion of the hazard function in section 4.2 on p. 43 we know that

$$P(T_\Gamma > s) = \exp\{-\int_0^s h(\Gamma(z))dz\}.$$

We will use this function with the perturbation technique to find a curve which is stochastically dominant. For some hazard functions there will be a path Γ_0 which is stochastically optimal at every arc length. That is, Γ_0 will minimize

$$\exp(-\int_0^s h(\Gamma(z))dz)$$

for every value of $s = s_0$. In this case we may be able to derive a condition the optimal curve must satisfy. Let Γ^ν be the parameterization of Γ^* in terms of its arc length. Define

$$F_{s_0}(\varepsilon) \equiv \int_0^{s_0} h(\Gamma^\nu(z))dz$$

$$= \int_0^{v(s_0,\varepsilon)} h(x_0+\varepsilon x_1, y_0+\varepsilon y_1)u_1 dt. \qquad (14)$$

If Γ_0 is stochastically dominant at s_0 then $F_{s_0}(\varepsilon)$ has a maximum at $\varepsilon = 0$. This is a necessary condition for Γ_0 to be optimal.

4.3.1. Derivation of the Condition for Optimality

We will take the derivative of $F_{s_0}(\varepsilon)$ (equation 14), evaluate at $\varepsilon = 0$, and set it to zero. In this way we will develop a necessary condition for stochastic optimality.

The derivative is

$$dF_{s_0}/d\varepsilon = h(\Gamma^*(v(s_0,\varepsilon)))u_1|_{(v(s_0,\varepsilon),\varepsilon)}v_2 + \int_0^{v(s_0,\varepsilon)} [\{h_1(\Gamma^*(t))x_1(t) + h_2(\Gamma^*(t))y_1(t)\}u_1 + h(\Gamma^*(t))u_{12}]dt \quad (15)$$

Now evaluate at $\varepsilon = 0$. Recall

$$v(s_0,\varepsilon)|_{\varepsilon = 0} = s_0.$$

In the evaluation of (15) at $\varepsilon = 0$ we use equation (8) on page 50

$$u_1|_{(v(s^*,0),0)} = 1$$

and equation (13), page 52

$$\partial v(s^*,\varepsilon)/\partial\varepsilon|_{\varepsilon = 0} = -\int_0^{s^*}\{Dx_0Dx_1 + Dy_0Dy_1\}dt$$

along with equation (12) page 51:

$$u_{12}|_{\varepsilon = 0} = [Dx_0Dx_1 + Dy_0Dy_1].$$

Thus when we evaluate (15) at $\varepsilon = 0$ we have

$$dF/d\varepsilon|_{\varepsilon=0} = -h(\Gamma_0)\int_0^{s_0} Dx_0Dx_1 + Dy_0Dy_1dt + \int_0^{s_0} h_1(\Gamma_0)x_1 + h_2(\Gamma_0)y_1dt + \int_0^{s_0} h(\Gamma_0)[Dx_0Dx_1 + Dy_0Dy_1]dt \quad (16)$$

Set this equal to zero and rearrange:

$$h(\Gamma_0)\int_0^{s_0} Dx_0Dx_1 + Dy_0Dy_1 dt =$$

$$\int_0^{s_0} h_1(\Gamma_0)x_1 + h_2(\Gamma_0)y_1 dt$$

$$+ \int_0^{s_0} h(\Gamma_0)[Dx_0Dx_1 + Dy_0Dy_1]dt$$

Now consider this as an equation in s_0 and take derivatives with respect to s_0:

$$h(\Gamma_0)[Dx_0Dx_1 + Dy_0Dy_1] +$$

$$[h_1(\Gamma_0)Dx_0 + h_2(\Gamma_0)Dy_0]\int_0^{s_0} Dx_0Dx_1 + Dy_0Dy_1 dt$$

$$= h_1(\Gamma_0)x_1 + h_2(\Gamma_0)y_1 + h(\Gamma_0)[Dx_0Dx_1 + Dy_0Dy_1]$$

or

$$[h_1(\Gamma_0)Dx_0 + h_2(\Gamma_0)Dy_0]\int_0^{s_0} Dx_0Dx_1 + Dy_0Dy_1 dt =$$

$$h_1(\Gamma_0)x_1 + h_2(\Gamma_0)y_1. \qquad (17)$$

Thus if there is a path Γ_0 which is "best" for all arc lengths, equation (17) must hold for any competing curve Γ_1. We are free to choose particular curves as competitors and use them in (17).

If we take Γ_1 to be the positive x-axis, i.e. $x_1(s) = s$, $y_1(s) = 0$, $0 \le s < \infty$, then $Dx_1(s) = s$, $Dy_1(s) = 0$ and (17) becomes

$$[h_1(\Gamma_0)Dx_0 + h_2(\Gamma_0)Dy_0]\int_0^{s_0} Dx_0 dt =$$

$$h_1(\Gamma_0)s_0$$

or

$$x_0[h_1Dx_0 + h_2Dy_0] = s_0\, h_1 \qquad (18)$$

Similarly we may take Γ_1 to be the positive y-axis: $x_1(s) = 0$, $y_1(s) = s$, $0 \leq s < \infty$, so $Dx_1(s) = 0$, $Dy_1(s) = 1$. The resulting equation is

$$y_0[h_1(\Gamma_0)Dx_0 + h_2(\Gamma_0)Dy_0] = h_2(\Gamma_0)s_0 \tag{19}$$

4.3.2. Discussion of the Solution

We may combine (18) and (19), writing the condition as

$$x_0h_2 = y_0h_1. \tag{20}$$

This is a pointwise condition. Our original assumption was that Γ_0 was parameterized in terms of its arc length, but (20) does not depend on the parameterization. Thus we are free to use it to find a solution without regard to parameterizations.

Equation (20) is a local condition. That is, we now see that requiring the existence of a curve which is best for every arc length is equivalent to the optimality of a "myopic" procedure: one need not consider what could happen in the future in order to decide what to do now.

4.3.3. Examples

There need not be a single path which is stochastically dominant for every arc length. Consider the following hazard function:

$$h(x,y) = \begin{cases} x^2 + y^2 & x < 0 \\ x^3 + y^3 & x \geq 0. \end{cases}$$

We select two paths for consideration,

$$\Gamma_a: \quad x_a(s) = s, \; y_a(s) = 0$$

and

$$\Gamma_b: \quad x_b(s) = -s, \; y_b(s) = 0.$$

We can easily compute the probability of continuing past arc length 1 on each path.

$$\begin{aligned} P_a(T > 1) &= \exp\{-\int_0^1 h(\Gamma_a)ds\} \\ &= \exp\{-\int_0^1 s^2 ds\} \\ &= \exp\{-1/3\} \\ &= 0.72 \end{aligned}$$

and

$$\begin{aligned} P_b(T > 1) &= \exp\{-\int_0^1 h(\Gamma_b)ds\} \\ &= \exp\{-\int_0^1 s^3 ds\} \\ &= \exp\{-1/4\} \\ &= 0.78 \end{aligned}$$

One is less likely to have to go beyond arc length 1 on Γ_a, so it is preferred. On the other hand,

$$P_a(T > 2) = \exp\{-\int_0^2 h(\Gamma_a)ds\}$$
$$= \exp\{-\int_0^2 s^2 ds\}$$
$$= 0.07$$

while

$$P_b(T > 2) = \exp\{-\int_0^2 h(\Gamma_b)ds\}$$
$$= \exp\{-\int_0^2 s^3 ds\}$$
$$= 0.02$$

One is less likely to go past arc length 2 on Γ_b, so it is better at length 2. It is easy to see that Γ_a and Γ_b are optimal curves for these two arc lengths.

As another example we take for the hazard function

$$h(x,y) = x^2 + y^2$$

so $h_1 = 2x$ and $h_2 = 2y$. Any ray from the origin should be stochastically optimal. Fix θ and let

$$x_0(s) = s(\cos\theta)$$
$$y_0(s) = s(\sin\theta) \qquad 0 \le s < \infty.$$

Now the necessary condition is

$$s(\cos\theta)\{2s(\sin\theta)\} = s(\sin\theta)\{2s(\cos\theta)\}.$$

So if one takes as Γ_0 any ray from the origin, the necessary conditions are satisfied.

As our next example we use

$$h(x,y) = \exp\{-1/2[(x - 2)^2 + (y - 2)^2]\}.$$

Now

$$h_1(x,y) = h(x,y)[-(x - 2)]$$

and

$$h_2(x,y) = h(x,y)[-(y - 2)]$$

We expect the optimal curve to be the 45-degree line, at least out to (2,2). Let

$$x_0(s) = s2^{-1/2}$$

$$y_0(s) = s2^{-1/2}$$

The necessary conditions are

$$xh(x,y)[-(x - 2)] = yh(x,y)[-(y - 2)]$$

or

$$s2^{-1/2}h[-(s2^{-1/2} - 2)] = s2^{-1/2}[-(s2^{-1/2} - 2)]$$

so Γ_0 satisfies the necessary condition.

Given a hazard function we may write down the necessary condition and try to solve for Γ_0. It may then be possible to verify that Γ_0 provides a minimum. For example, let

$$h(x,y) = \exp\{-1/2[(x - 1)^2 + (y - 3^{1/2})^2]\}.$$

Then

$$h_1 = -(x - 1)h(x,y)$$

$$h_2 = -(y - 3^{1/2})h(x,y)$$

so the necessary condition is

$$-x(y - 3^{1/2})h(x,y) = -y(x - 1)h(x,y).$$

For $0 < x < 1$, $0 < y < 3^{1/2}$, this simplifies to

$$x3^{1/2} = y$$

This is the line from the origin through $(1,3^{1/2})$, which is in fact stochastically optimal up to that point.

The computation in the previous example was simplified by the fact that the hazard function was circularly symmetric around $(1,3^{1/2})$. We will do one more example with a slightly more complicated hazard function. Let

$$h(x,y) \equiv h^{*} = \exp\{-1/2[1/2(x - 1)^2 + 1/4(y - 3^{1/2})^2]\}.$$

The necessary condition is

$$x(1/2)(y - 3^{1/2}) = y(x - 1)$$

or

$$x = 2y(y + 3^{1/2})^{-1}$$

The curve determined by this equation is plotted in figure 14.

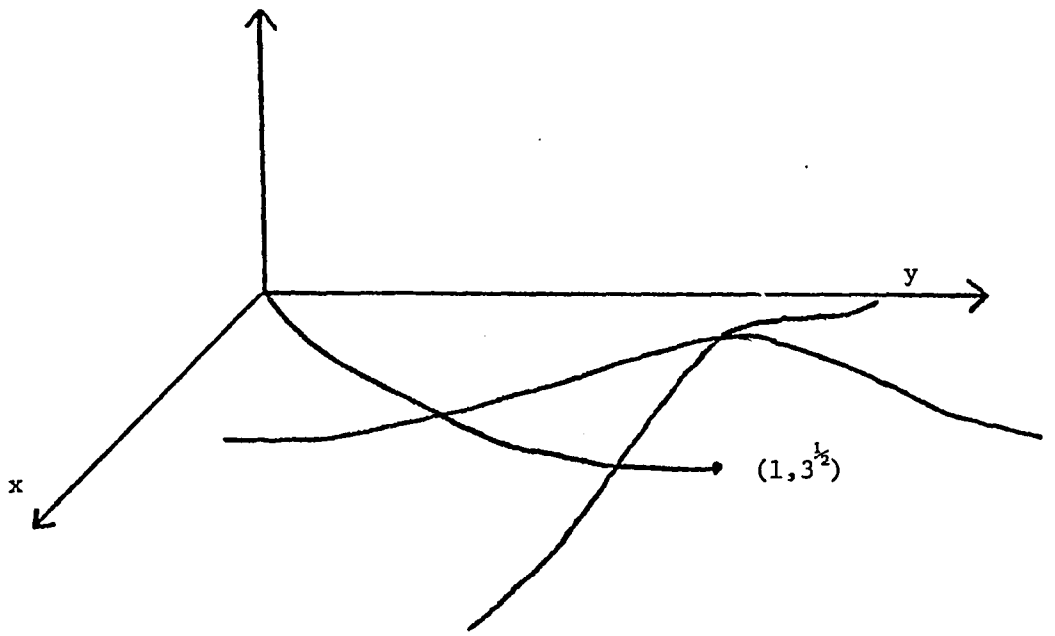

FIGURE 14. An optimal path for h*.

4.4. Minimum Expected Value

4.4.1. Discussion of the Criterion

In our discussion of finite network problems, we compared paths on the basis of their expected costs. Now, in the hazard function model, we have replaced costs with arc lengths, i.e. we assume it costs one unit to travel a distance of one unit.

In this section we employ the optimality criterion of minimum expected length. Our plan is to use the perturbation technique. The expression for the expected length of Γ^* is a function of ε, and we again take a derivative and set it equal to 0.

4.4.2. Derivation of the Optimality Equations

From our discussion in 4.1.1 we know the cdf of the stopping time on Γ is

$$F_\Gamma(s) = 1 - \exp\{\int_0^s h_\Gamma(t)dt\}.$$

Thus

$$\exp\{-\int_0^s h_\Gamma(t)dt\} = 1-F_\Gamma(s)$$

which we can integrate to find the expected value of the stopping time random variable on Γ:

$$E_\Gamma(T) = \int_0^\infty 1 - F_\Gamma(t)dt$$
$$= \int_0^\infty \exp\{-\int_0^s h_\Gamma(t)dt\}ds \qquad (21)$$

In the case of finite stochastic networks we sought a curve with minimum expected value. We will do the same here. Our tool will again be the perturbation technique. The expression for the expected value of the lifetime random variable on Γ^* is considered to be a function of ε. If Γ_0 is optimal, the derivative of that function must be 0 at $\varepsilon = 0$.

Some of the work of evaluating the derivative of the expected value function has already been done in the section on stochastic ordering.

Define

$$\Psi(s^*,\varepsilon) \equiv \int_0^{s^*} h(\Gamma^\nu(z))dz.$$
$$= \int_0^{v(s^*,\varepsilon)} h(x_0 + \varepsilon x_1, y_0 + \varepsilon y_1)(\partial u/\partial s)ds$$

(Recall Γ^ν is the arc length parameterization of Γ^*.) Then using equation (16) on page 55

$$d\Psi/d\varepsilon\big|_{\varepsilon = 0} = -h(\Gamma_0(s^*))\int_0^{s^*} Dx_0Dx_1 + Dy_0Dy_1dt$$
$$+ \int_0^{s^*}[\{h_1(\Gamma_0(s))x_1(s) + h_2(\Gamma_0(s))y_1(s)\}ds$$
$$+ \int_0^{s^*} h(\Gamma_0(s))\{Dx_0Dx_1 + Dy_0Dy_1\}]ds$$

So the derivative of the expected value function is

$$dE_{\Gamma^*}(T)/d\varepsilon = d/d\varepsilon\int_0^\infty \exp\{-\int_0^s h(\Gamma^\nu(z))dz\}ds$$
$$= d/d\varepsilon\int_0^\infty \exp\{-\Psi(s^*,\varepsilon)\}ds$$
$$= \int_0^\infty \exp\{-\Psi(s^*,\varepsilon)\}d/d\varepsilon\{-\Psi(s^*,\varepsilon)\}ds$$

$$= -\int_0^\infty \exp\{-\Psi(s^*,\varepsilon)\}$$
$$h(\Gamma^*(v(s^*,\varepsilon)))u_1|_{(v(s^*,\varepsilon),\varepsilon)}\partial v(s^*,\varepsilon)/\partial\varepsilon$$
$$+ \int_0^{v(s^*,\varepsilon)}[\{h_1(\Gamma^*(t))x_1(t) + h_2(\Gamma^*(t))y_1(t)\}u_1$$
$$+ h(\Gamma^*(t))u_{12}dt\}$$

Now set ε to zero:

$$dE_{\Gamma^*}(T)/d\varepsilon|_{\varepsilon=0} = -\int_0^\infty \exp\{-\Psi(s,0)\}$$
$$\{-h(\Gamma_0)\int_0^s [Dx_0Dx_1 + Dy_0Dy_1]dt$$
$$+ \int_0^s [h_1(\Gamma_0)x_1 + h_2(\Gamma_0)y_1]dt$$
$$+ \int_0^s h(\Gamma_0)[Dx_0Dx_1 + Dy_0Dy_1]dt\}ds.$$

We break this into three pieces to make it easier to manipulate

$$I = -\int_0^\infty \exp\{-\Psi(s,0)\}h(\Gamma_0(s))\int_0^s Dx_0Dx_1 + Dy_0Dy_1dt\ ds$$

$$II = \int_0^\infty \exp\{-\Psi(s,0)\}\int_0^s h_1(\Gamma_0(t)x_1(t) + h_2(\Gamma_0(t))y_1(t)dt\ ds$$

$$III = \int_0^\infty \exp\{\Psi(s,0)\}\int_0^s h(\Gamma_0(t))[Dx_0Dx_1 + Dy_0Dy_1]dt\ ds$$

In part I we interchange the order of integration

$$I = -\int_0^\infty\int_0^s \exp\{-\Psi(s,0)\}h(\Gamma_0(s))[Dx_0(t)Dx_1(t)$$
$$+ Dy_0(t)Dy_1(t)]dtds$$
$$= -\int_0^\infty\int_t^\infty \exp\{-\Psi(s,0)\}h(\Gamma_0(s))[Dx_0(t)Dx_1(t)$$
$$+ Dy_0(t)Dy_1(t)]dsdt$$
$$= \int_0^\infty [Dx_0Dx_1 + Dy_0Dy_1]\int_t^\infty \exp\{-\Psi(s,0)\}\{-h(\Gamma_0(s))\}dsdt$$

But since

$$d/ds\{\Psi(s,0)\} = h(\Gamma_0(s))$$

and

$$\exp\{-\int_0^\infty h(\Gamma_0(t))dt\} = 0$$

we see

$$I = \int_0^\infty [Dx_0 Dx_1 + Dy_0 Dy_1][-\exp\{-\Psi(t,0)\}]dt$$
$$= -\int_0^\infty [Dx_0 Dx_1 + Dy_0 Dy_1][\exp\{-\Psi(t,0)\}]dt.$$

Now interchange the order of integration in II:

$$II = \int_0^\infty \{h_1(\Gamma_0(t))x_1(t) + h_2(\Gamma_0(t))y_1(t)\}\{\int_t^\infty \exp\{-\Psi(s,0)\}ds\}dt$$

and in III:

$$III = \int_0^\infty h(\Gamma_0(t))[Dx_0(t)Dx_1(t)$$
$$+ Dy_0(t)Dy_1(t)]\{\int_t^\infty \exp\{-\Psi(s,0)\}ds\}dt.$$

Now define

$$a(t) \equiv \int_t^\infty \exp\{-\Psi(s,0)\}ds$$

write Ψ for $\Psi(t,0)$ and rewrite

$$I = -\int_0^\infty [Dx_0 Dx_1 + Dy_0 Dy_1]\exp(-\Psi)dt$$
$$= -\int_0^\infty Dx_1[Dx_0 \exp(-\Psi)]dt - \int_0^\infty Dy_1[Dy_0 \exp(-\Psi)]dt$$
$$= \int_0^\infty Dx_1[-Dx_0 \exp(-\Psi)]dt + \int_0^\infty Dy_1[-Dy_0 \exp(-\Psi)]dt$$

and

$$II = \int_0^\infty \{h_1(\Gamma_0(t))x_1(t) + h_2(\Gamma_0(t))y_1(t)\}a(t)dt$$
$$= \int_0^\infty x_1(t)[h_1(\Gamma_0(t))a(t)]dt + \int_0^\infty y_1(t)[h_2(\Gamma_0(t))a(t)]dt$$

and

$$III = \int_0^\infty h(\Gamma_0(t))[Dx_0 Dx_1 + Dy_0 Dy_1]a(t)dt$$
$$= \int_0^\infty Dx_1[Dx_0 h(\Gamma_0(t))a(t)]dt + \int_0^\infty Dy_1[Dy_0 h(\Gamma_0(t))a(t)]dt.$$

Our aim is to group together terms involving x_1 and y_1. Define (associated with Dx_1)

$$f_1(t) \equiv [Dx_0 h(\Gamma_0)a(t) - Dx_0 \exp(-\Psi)]$$
$$= Dx_0[h(\Gamma_0)a(t) - \exp(-\Psi)]$$

and (associated with x_1)

$$f_2(t) \equiv h_1(\Gamma_0(t))a(t).$$

Also define (associated with Dy_1)

$$g_1(t) \equiv [Dy_0 h(\Gamma_0)a(t) - Dy_0 \exp(-\Psi)]$$
$$= Dy_0[h(\Gamma_0)a(t)) - \exp(-\Psi)]$$

and (associated with y_1)

$$g_2(t) \equiv h_2(\Gamma_0(t))a(t).$$

To recapitulate, we have expressed the derivative as

$$dE_{\Gamma^*}(T)/d\varepsilon\,|_{\varepsilon = 0}$$
$$= \int_0^\infty Dx_1(t)f_1(t)dt + \int_0^\infty x_1(t)f_2(t)dt$$
$$+ \int_0^\infty Dy_1(t)g_1(t)dt + \int_0^\infty y_1(t)g_2(t)dt. \qquad (22)$$

The functions f_1, f_2, g_1, g_2 involve only x_0, y_0 and not x_1, y_1.

Consider the first integral on the right-hand side of (22). We integrate by parts with

$$dv(t) = Dx_1(t)$$
$$u(t) = f_1(t) = Dx_0[h(\Gamma_0)a(t) - \exp(-\Psi)]$$

so that

$$u(t)v(t) = x_1(t)Dx_0[h(\Gamma_0)a(t) - \exp(-\Psi)]$$

Recall that the parameterization in terms of arc length implies

$$x_1(t) \leq t$$

and

$$Dx_0(t) \leq 1 \qquad 0 \leq t < \infty.$$

Also $x_1(0) = 0$ and $\exp\{-\Psi(0,0)\} = 1$, while $a(0) = E_{\Gamma_0}(T)$. We assume the expectation is finite. So we have

$$u(0)v(0) = 0.$$

Now we assume that the stopping time, T, has a finite variance on Γ_0. If we write F_T for the cdf of T on Γ_0,

$$t\int_t^\infty x dF_T(x) < \int_t^\infty x^2 dF_T(x) < \infty$$

and

$$\lim_{t\to\infty} t\int_t^\infty x dF_T(x) = 0.$$

Also assume that h on Γ_0 is such that $t[\exp\{-\int_0^t h(\Gamma_0)ds\}] \to 0$ as $t \to \infty$. This would be true, for example, if, for t large, h is bounded away from 0 on Γ_0. If h is also bounded above, then, as $t \to \infty$,

$$u(t)v(t) \to 0.$$

The first integral in (22) is

$$\int_0^\infty Dx_1(t)f_1(t)dt = 0 - \int_0^\infty x_1(t)Df_1(t)dt$$

Similarly,

$$\int_0^\infty Dy_1(t)g_1(t)dt = -\int_0^\infty y_1(t)Dg_1(t)dt.$$

The structure of the above results becomes clearer if we switch to inner product notation, $\int_0^\infty f(t)g(t)dt = (f,g)$. We may summarize the above as

$$dE_{\Gamma^*}(T)/d\varepsilon\big|_{\varepsilon = 0} = (Dx_1,f_1) + (x_1,f_2) + (Dy_1,g_1) + (y_1,g_2)$$

$$= (x_1,-Df_1) + (x_1,f_2) + (y_1,-Dg_1) + (y_1,g_2)$$

$$= (x_1,f_2-Df_1) + (y_1,y_2-Dg_1) \tag{23}$$

since $(Dx_1,f_1) = (x_1,-Df_1)$ and $(Dy_1,g_1) = (y_1,-Dg_1)$.

We have an expression for the derivative of the expected value function, and we are ready to set it equal to zero and derive conditions Γ_0 must satisfy:

$$(x_1,f_2-Df_1) + (y_1,g_2-Dg_1) = 0.$$

Recall that f_1, f_2, g_1, g_2 involve only Γ_0. We are free to choose any competing curve Γ_1 we wish, so it must be the case that

$$f_2-Df_1 = 0$$

$$g_2-Dg_1 = 0 \cdot \tag{24}$$

These are the necessary conditions.

It will be easier to work with them if we write them in terms of h and Γ_0:

$$h_1(\Gamma_0(t))a(t) - D\{Dx_0[h(\Gamma_0)a(t) - \exp(-\Psi)]\} = 0$$

$$h_2(\Gamma_0(t))a(t) - D\{Dy_0[h(\Gamma_0)a(t)) - \exp(-\Psi)]\} = 0 \tag{25}$$

But

$$Da(t) = D\int_t^\infty \exp\{-\Psi(s,0)\}ds$$

$$= -D\int_0^t \exp\{-\Psi(s,0)\}ds$$

$$= -\exp\{-\Psi(t,0)\}$$

$$= -\exp\{-\int_0^t h(\Gamma_0(s))\}ds$$

$$= -\exp\{-\Psi(t,0)\}$$

and

$$D\{Dx_0[h(\Gamma_0)a(t) - \exp(-\Psi)]\}$$
$$= D^2x_0[h(\Gamma_0)a(t) - \exp(-\Psi)]$$
$$+ Dx_0\{[h_1(\Gamma_0(t))Dx_0 + h_2(\Gamma_0(t))Dy_0]a(t)$$
$$-h(\Gamma_0)\exp(-\Psi) + \exp(-\Psi)D\Psi\}$$

and

$$D\{Dy_0[h(\Gamma_0)a(t) - \exp(-\Psi)]\}$$
$$= D^2y_0[h(\Gamma_0)a(t) - \exp(-\Psi)]$$
$$+ Dy_0\{[h_1(\Gamma_0(t))Dx_0 + h_2(\Gamma_0(t))Dy_0]a(t)$$
$$-h(\Gamma_0)\exp(-\Psi) + \exp(-\Psi)D\Psi\}$$

But

$$D\Psi = D\int_0^t h(\Gamma_0)ds = h(\Gamma_0)$$

so (25) may be written

$$h_1(\Gamma_0)a(t) - D^2x_0[h(\Gamma_0)a(t) - \exp(-\Psi)]$$
$$-Dx_0\{h_1(\Gamma_0)Dx_0 + h_2(\Gamma_0)Dy_0\}a(t) = 0 \qquad (26)$$

and

$$h_1(\Gamma_0)a(t) - D^2y_0[h(\Gamma_0)a(t) - \exp(-\Psi)]$$
$$-Dy_0\{h_1(\Gamma_0)Dx_0 + h_2(\Gamma_0)Dy_0\}a(t) = 0. \qquad (27)$$

As a simple example we take $h(x,y) = x^2 + y^2$. Then any ray should be optimal. Let Γ_0 be

$$x_0(s) = s(\cos\theta)$$
$$y_0(s) = s(\sin\theta)$$

Then

$$h(\Gamma_0) = s^2$$

$$h_1(\Gamma_0) = 2s(\cos\theta)$$

$$h_2(\Gamma_0) = 2s(\sin\theta)$$

$$\Psi(t,0) = \int_0^t s^2 ds = t^3/3$$

$$a(t) = \int_t^\infty \exp(-s^3/3)ds.$$

$$Dx_0 = \cos\theta \qquad D^2x_0 = 0$$

$$Dy_0 = \sin\theta \qquad D^2y_0 = 0$$

Now we evaluate (26):

$$2t(\cos(\theta))a(t) - \cos\theta\{2t(\cos\theta)\cos\theta + 2t(\sin\theta)\sin\theta\}a(t)$$

$$= 2t(\cos\theta)a(t) - 2t(\cos\theta)a(t)$$

$$= 0$$

Any ray from the origin satisfies the necessary conditions.

5. BIBLIOGRAPHY

Burr, Stefan A., ed. The Mathematics of Networks. Providence, Rhode Island: American Mathematical Society, 1982.

Conway, Richard W.; Maxwell, William L.; and Miller, Louis W. Theory of Scheduling. Reading, Massachusetts: Addison-Wesley Publishing Company, 1967.

Cook, S. The complexity of theorem-proving procedures. Proceedings of the Third ACM Symposium on Theory of Complexity, New York (1971):151-158.

Dijkstra, E. A note on two problems in connection with graphs. Numerische Mathematik 1 (1959):269-271.

Dreyfus, Stuart E. Dynamic Programming and the Calculus of Variations. New York: Academic Press Inc., 1965.

Dreyfus, S. E. An appraisal of some shortest-path algorithms. Operations Research 17 (1969):395-412.

Dreyfus, Stuart E., and Law, Averill M. The Art and Theory of Dynamic Programming. New York: Academic Press Inc., 1977.

Ewing, George M. Calculus of Variations with Applications. New York: W. W. Norton and Company, Inc., 1969.

Ford, L. R., and Fulkerson, D. R. Constructing maximal dynamic flows from static flows. Operations Research 6 (1958):419-433.

Gelfand, I. M., and Fomin, S. V. Calculus of Variations. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1963.

Harary, F. Graph Theory. Reading, Massachusetts: Addison-Wesley, 1969.

Henley, Ernest J., and Williams, R. A. Graph Theory in Modern Engineering. New York: Academic Press Inc., 1973.

Karp, Richard M. Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. Mathematics of Operations Research 2 No. 3 (1977):209-224.

Karp, R. On the complexity of combinatorial problems. Networks 5 (1975):45-68.

Kruskal, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society 7 (1956):48-50.

Lawless, Jerald F. Statistical Models and Methods for Lifetime Data. New York: John Wiley and Sons, 1982.

Leipala, T. On the solutions of stochastic traveling salesman problems. European Journal of Operational Research 2 No. 4 (1978):291-297.

Lin, Cherng-Tarng (Tony). Waiting times for target detection models. Ph.D. dissertation, Iowa State University, 1983.

Lin, S. Computer solutions of the traveling salesman problem. Bell System Technical Journal 44 (1965):2245-2269.

Minieka, E. Optimization Algorithms for Networks and Graphs. New York and Basel: Marcel Dekker, Inc., 1978.

Minty, George J. A comment on the shortest-route problem. Operations Research 5 No. 5 (1957):724.

Papadimitriou, C. H., and Steiglitz, K. Some examples of difficult traveling salesman problems. Operations Research 26 No. 3 (1978):434-443.

Pollack, Maurice, and Wiebenson, Walter. Solutions of the shortest-route problem-- a review. Operations Research 8 No. 2 (March 1960):224-230.

6. ACKNOWLEDGMENTS

I wish to thank my major professors, Dr. Krishna Athreya and Dr. Herbert T. David, for sharing their probabalistic and analytic expertise, and for demonstrating how a research project works. I also thank my committee members, Dr. Glen Meeden, Dr. Stephen Vardeman, and Dr. Justin Peters for their warm friendship. I also wish to express my appreciation to Kristen Keele for her excellent work deciphering formulas, drawing diagrams, and typing text efficiently.