Constrained shortest paths in networks
by Chi-Hung Shek

A thesis submitted in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in Electrical Engineering
Montana State University

Abstract:
Shortest path problems in networks constitute a branch of optimization theory wherein system network models with paths which are characterized by path lengths are of concern.

This work is an extension of the classical shortest path problem; here additional cost constraints are imposed on the network model.

A dynamic programming solution approach is first formulated. It is solved by an iterative scheme in which the cost bound variable is incremented by a unit at the end of each iteration where the shortest paths with path costs not exceeding the cost bound variable are found. The concept of "stepping cost" is introduced representing the path cost at which a change of shortest path length occurs. A theorem is proved on the procedure to locate the "immediate stepping cost" and the associated stepping-cost path length; thus, an acceleration of the iterative scheme is feasible. As a result, a combinatorial algorithm is developed with computational effort being the order of the square of the number of stepping costs.

Further extensions to networks with more than one path cost constraint are presented.

The constrained shortest path algorithms are applied to flow networks and to networks with probabilistic weighted branch costs. FORTRAN computer programs are given to implement the algorithms.

CONSTRAINED SHORTEST PATHS IN NETWORKS

by

CHI-HUNG SHEK

A thesis submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

Approved:

Donald A. Pierre
Chairman, Examining Committee

Paul E. Ulrich
Head, Major Department D.N.M

Henry L Parsons
Graduate Dean

MONTANA STATE UNIVERSITY
Bozeman, Montana

June, 1975

ACKNOWLEDGMENTS

The author would like to thank the Electrical Engineering Department of Montana State University for the financial support throughout his entire Ph.D. program.

Also, the author would like to thank Dr. Donald A. Pierre for his guidance, moral support, and patience without which the preparation of this thesis would have been impossible. Thanks are expressed to Dr. D. A. Rudberg and Dr. R. M. Johnson for their review of material and their many constructive suggestions for the thesis.

The author would like to thank Peggy Humphrey for typing the manuscript and for her suggestions in the arrangement of this thesis.

Finally, the author would like to thank his parents and the rest of his family for their encouragement and support.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd.)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (cont'd)

LIST OF FIGURES (cont'd)

ABSTRACT

Shortest path problems in networks constitute a branch of optimization theory wherein system network models with paths which are characterized by path lengths are of concern.

This work is an extension of the classical shortest path problem; here additional cost constraints are imposed on the network model.

A dynamic programming solution approach is first formulated. It is solved by an iterative scheme in which the cost bound variable is incremented by a unit at the end of each iteration where the shortest paths with path costs not exceeding the cost bound variable are found. The concept of "stepping cost" is introduced representing the path cost at which a change of shortest path length occurs. A theorem is proved on the procedure to locate the "immediate stepping cost" and the associated stepping-cost path length; thus, an acceleration of the iterative scheme is feasible. As a result, a combinatorial algorithm is developed with computational effort being the order of the square of the number of stepping costs.

Further extensions to networks with more than one path cost constraint are presented.

The constrained shortest path algorithms are applied to flow networks and to networks with probabilistic weighted branch costs. FORTRAN computer programs are given to implement the algorithms.

CHAPTER I

INTRODUCTION AND PRELIMINARY CONCEPTS

1.1 Introduction

Because of the existence and availability of digital computers, the past few decades have seen a rapid growth in the application of scientific methods to solve problems for which there was virtually no hope of solution before. Optimization theory has emerged as a very important discipline common to many diverse problem areas.

Loosely speaking, optimization can be defined as finding the best way to carry out an action or to produce a result. More accurately, it can be said that one wishes to define a system of some type, to identify its variables and the conditions they must satisfy, to define a measure of effectiveness of the system, and then to seek the state of the system that gives the most desirable measure of effectiveness.

Problems in optimal control usually involve a measure of effectiveness in the form of a time integral, and their solution requires that a set of boundary-value equations be solved, in general. With appropriate modifications, the control problem can often be posed such that the scope of solution methods includes dynamic programming. The problem of finding a minimal value of a function of several variables requires the application of mathematical programming. When only linear algebraic equations are involved in the optimizing function and constraints, linear programming is used; otherwise, nonlinear programming is applicable.

The Simplex Method, originated by G. B. Dantzig [1], is a widely used linear programming method. Recent research has shown that large problems are likely to be decomposable into a number of smaller sized linear optimization problems; problems with several thousand variables and constraints have been solved successfully.

Nonlinear programming solutions are found through various search techniques. Concepts such as steepest descent, the Lagrange multiplier, Newton-Raphson search, and conjugate-directions are incorporated in some of the popularly used methods [35].

Dynamic programming, originated by R. Bellman [2,3], is applicable when the system can be modelled as a multistage decision process where a recurrence relation is obtained to relate the optimal decision of one stage to that of the next, so that the remaining decision is expressible in terms of the previous decision and the state of the system. Whereas both linear and nonlinear programming are suitable for solving systems described by well-behaved functions of the system state, dynamic programming is well suited for problems with discontinuities because of the discrete nature of the technique.

Of all the system optimization problems, network optimization has been considered as a highly important subdiscipline. In general, all problems in the field have the following in common: They comprise components and parts which are interconnected, and the interconnection can be described by a graph. The constituents or variables in the

networks are interdependent yet some of them are loosely coupled, thus allowing simplified solution. Certain constraints and goals are prescribed. The objective is, by and large, the optimization of such items as speed, cost, or reliability.

In transportation networks where commodities are transported through the network between specified pairs of source and destination nodes, the maximum flow problem [4] has an objective to find the optimal flow pattern so that as much flow as possible can be shipped without causing an overflow of the network capacity; the minimum cost flow problem [5] concerns the cost of shipping as the measure of effectiveness when a set of flow requirements between the source and destination pairs is prescribed. The placement of power lines [6], pipelines [7,8] are special kinds of Steiner's problem where the objective is to minimize the total interconnection length (or cost) subject to a limited budget.

As small size products are in continuous demand, the placement of logic units, the interconnection of wires [9,10], and the ordering of logic boards [11] of complicated electronic devices to minimize total wire length become nontrivial problems for which only heuristic solution methods exist at present.

Computer networks, which link data processors at different locations to form communication networks, are used to facilitate high speed information handling and transmission in addition to resource sharing. Rapid advances [12,13,14,15,16,17,18,19,20] have been seen in optimiza-

tion objective areas such as flow control, routing, message switching, delay, and reliability.

The problem of finding shortest paths in networks is a type of multistage decision process where the dynamic programming solution approach has been most efficient [21,22]. The problem first appears as a commuting problem [23] in which given a road map the shortest route from one city to another is sought. Under appropriate definitions, the shortest path has been used as an analytic tool to relate network performance; for example, packets or messages are forwarded to their destinations along the "shortest" route [24], "shortest" in the sense of minimal time-delay [16], maximal reliability [25,26], or maximal utilization of network capability [27]. The ordering of logic boards in a complicated electronic system, so that the total interconnection wire length is minimized, turns out to be a problem of finding the shortest path between a unique pair of source and sink nodes of a network model [11].

Shortest paths with special features have been studied in some works. In [28,29], consideration is given to the shortest path distribution in a probabilistic weighted network; in [30], the shortest path problem is formulated in a network with time-dependent weights; in [31], a study that corresponds to an optimal traveling scheme through a transportation network with time restrictions for the accessibility and parking on certain nodes is treated; the problem of finding shortest

paths that must pass through a given subset of nodes is discussed in [32]; and in [33], the optimal route to traverse a road network with penalties or prohibitions in making a turn is found as an application of the shortest path problem.

In this thesis, shortest paths that satisfy a constraint are treated. Section 1.2 introduces the problem of finding the constrained shortest path in a network, with the dynamic programming formulation stated in Section 1.3. Section 1.4 cites some of the basic graph definitions. Pertinent history and some of the solution approaches for the unconstrained shortest path problem are listed in Sections 1.5 and 1.6. The contribution of this research work is outlined in Section 1.7.

1.2 Fundamental Definitions

A network G(V,B) of n nodes and m branches is characterized by the set of nodes V, and the set of branches B, where each member of B connects a distinct pair of nodes in V. Let v_i be the i^{th} node, and b_j be the j^{th} branch, so that $V = \{v_1, v_2, \ldots, v_n\}$, $B = \{b_1, b_2, \ldots, b_m\}$.

Let (i,j) denote a branch b_q if v_i is the initial node of b_q, and v_j is the terminal node of b_q. In a directed network, b_q is said to be directed from v_i to v_j; if the network is undirected, b_q is said to be incident at v_i and v_j.

A weighted network is one in which numbers, called weights, are associated with the branches of the network. Throughout this work,

each branch (i,j) of B is assumed to be assigned a pair of weights, d_{ij} and h_{ij}, represented by $W_{ij} = (d_{ij},h_{ij})$, which we call the length and cost of (i,j), respectively. In real networks, the weights may represent things such as distance (wire, road), amount of flow (commodity), cost, capacity (road, communication channel), probability measure (reliability), or delay (time to traverse the branch).

A chain is a sequence of nodes and branches, say $v_{i_1}, b_1, v_{i_2}, b_2, \ldots, v_{i_{k-1}}, b_{k-1}, v_{i_k}$, where each branch b_j incidents at $v_{i_{j+1}}$ and v_{i_j}, $j = 1, 2, \ldots, k-1$. A loop is an unidirectional chain where $i_1 = i_k$. A path π is an unidirectional chain where the nodes are distinct. $|\pi|$ is used to denote the length of π which is the sum of the lengths of the constituent branches. $C(\pi)$ is used to denote the cost of π which is the sum of the costs of the constituent branches.

An $i \to j$ path, denoted as $\pi_{i,j}$, is a path from node i to node j. When paths from a common initial node, say node 1, are under consideration, π_j represents a $1 \to j$ path.

A network G(V,B) is said to be connected if, for any node pair v_i and v_j, an $i \to j$ path exists.

A tree of undirected network is a subset of branches that connects the set of nodes V and that has the property that one and only one path exists between each pair of nodes.

A cut $(X,\bar{X})$ is the set of branches (i,j) where $i \in X$, $j \in \bar{X}$, and X is a subset of V with $\bar{X}$ denoting its node complement V-X.

In Fig. 1.1, a directed network G with 5 nodes, and 8 branches, is shown. G is connected. The branches (1,5), (4,5), (5,2) are undirected. The set of 1→5 paths are {(1,5)}, {(1,2),(2,5)}, {(1,2),(2,3),(3,5)}, {(1,2),(2,3),(3,4),(4,5)}. If $X = \{v_1\}$, $\bar{X} = \{v_2,v_3,v_4,v_5\}$, $(X,\bar{X}) = \{(1,2),(1,5)\}$. If $X = \{v_1,v_4\}$, $(X,\bar{X}) = \{(1,2),(1,5),(4,5)\}$.

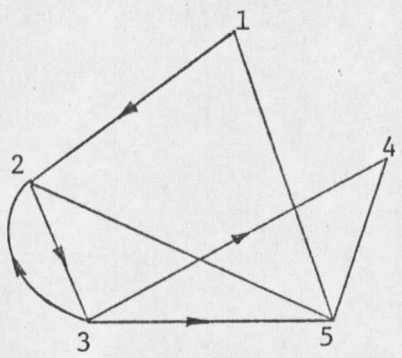

Fig. 1.1. Illustration of graph fundamentals.

1.3 The Problem Statement

Given a directed network G(V,B), a set of length weights $\{d_{ij}\}$,† a set of nonnegative cost weights $\{h_{ij}\}$, and a nonnegative constant c, find the set of shortest 1→i paths π_i, i = 2, 3, ..., n, if they exist, such that $C(\pi_i) \leq c$, for all $i \neq 1$.

If $\{\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_q}\}$ represents the set of 1→i paths, and π_i is the shortest 1-i path such that $C(\pi_i) \leq c$, then the length $|\pi_i|$ of π_i

---

†The branch lengths may be negative. However, a practical assumption is that all loop lengths must be nonnegative.

is such that

$$|\pi_i| = \underset{j}{\text{minimum}}\{|\pi_{i_j}|, \text{ such that } C(\pi_{i_j}) \leq c\} \qquad (1\text{-}3\text{-}1)$$

Equation (1-3-1) contains a minimizing operation that selects the minimum value of $|\pi_{i_j}|$ for π_{i_j} that satisfies the path cost requirement to be the shortest $1 \to i$ path length.

Intuitively, one may find the optimal path by straightforward path enumeration. However, when the size of the network is large, which is quite usual for practical network models, a large number of feasible paths exist, and naive path enumeration may become a tedious task.†

1.4 The Dynamic Programming Formulation

The principle of optimality, when applied to the problem in Section 1.3, yields a set of equations,††

$$|\pi_s(c)| = \underset{t \neq s, h_{ts} \leq c}{\text{minimum}} \{d_{ts} + |\pi_t(c-h_{ts})|\}, \text{ for } s = 2, 3, \ldots, n \qquad (1\text{-}4\text{-}1)$$

$$|\pi_1(c)| = 0 \qquad (1\text{-}4\text{-}2)$$

subject to the condition,

$$|\pi_s(c)| \leq |\pi_s(c')|, \text{ for } c' \leq c, \; s = 2, 3, \ldots, n \qquad (1\text{-}4\text{-}3)$$

---

†For a network with n nodes, the number of paths between a pair of nodes with n-2 intermediate nodes may be as large as (n-2)!.

††A derivation of equation (1-4-1) is presented in Chapter II.

where c is a nonnegative constant, and $\pi_s(c)$ is the shortest 1-s path with a path cost that is no bigger than c.

Equation (1-4-1) is a set of n-1 equations which relate each of the optimal 1→s paths to the rest of the optimal paths as a function of the cost bound c . Equation (1-4-2) simply points out the fact that the optimal path from node 1 to itself is a trivial path with zero path length. Inequality (1-4-3) is a statement of the fact that the optimal 1→s paths are nonincreasing functions of the cost bound variable c.

The multistage decision process of finding the shortest paths is such that whatever the initial cost bound c (state) and the last intermediate node (decision) are, the remaining decisions of intermediate nodes must contribute an optimal policy with regard to the cost bound resulting from the decision of the last intermediate node.

If all branch costs of the network assume zero value, a degenerate problem results: The unconstrained 1→s shortest paths π_s's are sought. Equations (1-4-1) and (1-4-2) become

$$|\pi_s| = \underset{t \neq s}{\text{minimum}}\{d_{ts} + |\pi_t|\}, \text{ for } s \neq 1 \qquad (1\text{-}4\text{-}4)$$

$$|\pi_1| = 0 \qquad (1\text{-}4\text{-}5)$$

The variable c is eliminated, since any feasible 1→s path has zero path cost. The n-1 unknown variables are sought. A large number of solution methods exist in the literature to solve (1-4-4); some of them are presented in the latter part of this chapter.

1.5 A Historical Review

One of the most efficient procedures to find the unconstrained shortest paths was described by Dijkstra [34], and by Whiting and Hillier [23] for networks with nonnegative branch lengths. For networks with possible negative branch lengths, a modified network can be found by simply adding a fixed positive amount to each of the branch lengths in the original network, so that all branch lengths become nonnegative, and Dijkstra's method is then applicable. Bellman [21], Ford [36], and Moore [37] proposed a set of functional equations by the application of the principle of optimality to relate the set of unconstrained shortest paths from a common initial node to all other terminal nodes. The equations are then solved by an iterative scheme. Yen [38,39] modified the above method, reducing computational effort by an integral factor.

While the above methods are able to find the set of $1 \to s$ shortest paths, for all $s \neq 1$, Floyd [40], Shimbel [41] published a matrix procedure, based on the work of Warshall [42], to find the unconstrained shortest paths between all node pairs. Yen [43] reduced the amount of computational effort by repetitive application of Dijkstra's algorithm. Winograd and Hoffman [44] suggested a pseudomultiplication method through network partitioning. Hu [45], Land *et al.* [46], Mills [47], taking advantage of the loosely-connected nature of many large order networks, suggested a decomposition method which reduces the problem to a set of problems of smaller size.

When the shortest path is not available because of possible network restrictions, the second shortest path is sought. Hoffman and Pavley [48] proposed a method to find the second-best path between a node pair by enumerating all of the possible deviation paths of the shortest path, and by selecting the best to be the second shortest path. Bellman and Kalaba's [50] functional formulation is a generalized expression. Dreyfus [51] suggested a general expression for the k^{th} best path. Based on the work of Murty [52] and Yen [53], Lawler's enumeration method [54] for the k^{th} best path is obviously more efficient.

Some of the unconstrained shortest path methods are presented in the following section as they contribute to the solution of the problem in Section 1.3.

1.6 Some Solution Methods

1.6.1 The Dynamic Programming Approach

The principle of optimality states that: Whatever the initial state and the initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the initial decision.

In finding the set of 1→s shortest paths, according to the principle of optimality, if the initial decision is to take branch (t,s) to be the last branch in the 1→s path, then the remaining portion of

the path, namely, the 1→t path, must be the shortest 1→t path. The system of equations

$$\begin{aligned} |\pi_s| &= \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|\}, \text{ for } s \neq 1 \\ |\pi_1| &= 0 \end{aligned} \tag{1-6-1-1}$$

thus constitute a necessary condition for the set of path lengths $\{|\pi_s|\}$ to be optimal. In fact, it is shown in Chapter II as a special case that the solution to equation (1-6-1-1) is also unique, so that the equation is indeed both a necessary and a sufficient condition for the set $\{|\pi_s|\}$ to be the shortest 1→s path lengths.

Although these equations are interlinked in such a fashion that they cannot be solved recursively, they may be solved by defining a new sequence $\{|\pi_s|^{(k)}\}$ in the following iterative scheme,

$$\begin{aligned} |\pi_s|^{(0)} &= d_{1s}, \text{ for } s \neq 1 \\ |\pi_1|^{(0)} &= 0 \end{aligned} \tag{1-6-1-2}$$

and

$$\begin{aligned} |\pi_s|^{(k+1)} &= \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|^{(k)}\}, \text{ for } s \neq 1 \\ |\pi_1|^{(k+1)} &= 0 \end{aligned} \tag{1-6-1-3}$$

for k = 0, 1, ... Then the sequence $\{|\pi_s|^{(k)}\}$ converges in a monotonically decreasing fashion to the sequence $\{|\pi_s|\}$ as $k\to\infty$. In fact, k need never be determined beyond the value n-2. Observe that, for $s \neq 1$,

$$|\pi_s|^{(1)} = \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|^{(0)}\} \leq d_{1s} = |\pi_s|^{(0)} \qquad (1\text{-}6\text{-}1\text{-}4)$$

since t = 1 is an admissible choice; and that

$$|\pi_s|^{(2)} = \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|^{(1)}\} \leq \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|^{(0)}\}$$

$$= |\pi_s|^{(1)} \qquad (1\text{-}6\text{-}1\text{-}5)$$

Hence, by induction, the sequence $\{|\pi_s|^{(k)}\}$ is such that $|\pi_s|^{(k)} \leq |\pi_s|^{(k-1)}$ for $k \geq 1$ and $s \neq 1$.

From equation (1-6-1-1), the shortest 1→s path length is

$$|\pi_s| = \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|\} \leq d_{1s} = |\pi_s|^{(0)} \qquad (1\text{-}6\text{-}1\text{-}6)$$

and

$$|\pi_s| = \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|\} \leq \underset{t\neq s}{\text{minimum}}\{d_{ts} + |\pi_t|^{(0)}\}$$

$$= |\pi_s|^{(1)}, \text{ for } s \neq 1 \qquad (1\text{-}6\text{-}1\text{-}7)$$

Through induction, the sequence $\{|\pi_s|^{(k)}\}$

$$|\pi_s|^{(0)} \geq |\pi_s|^{(1)} \geq \cdots \geq |\pi_s|^{(k)} \geq |\pi_s| \qquad (1\text{-}6\text{-}1\text{-}8)$$

The fact that the sequence in equation (1-6-1-3) will terminate for $k \leq n-2$ follows from the fact that $|\pi_s|^{(k)}$ is the shortest 1→s path