AN ALGORITHM FOR OPTIMAL SHIP ROUTING
FOR SEISMIC DATA COLLECTION

By

EDWARD PAYSON WILLARD

Bachelor of Science
Southern Methodist University
Dallas, Texas
1962

Bachelor of Science
Southern Methodist University
Dallas, Texas
1963

Master of Science
Southern Methodist University
Dallas, Texas
1964

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 1970

AN ALGORITHM FOR OPTIMAL SHIP ROUTING

FOR SEISMIC DATA COLLECTION

Thesis Approved:

Marvin Palmer Terrell
Thesis Adviser

James E. Shamblin

Earl J. Ferguson

Wilson J. Bentley

D. Leroy Folks

N. Durham
Dean of the Graduate College

PREFACE

This dissertation develops an algorithm that can be used to schedule ships to collect seismic data at a given prospect containing N seismic lines. To collect the required data, a ship must leave a known port and travel to the prospect, traverse each of the N lines one time, then return to a known port. At present, geophysical companies are relying solely upon managerial judgment to schedule these specially equipped ships. Since all data associated with this problem are deterministic in nature at the time the decision is being made, the theoretical solution serves as the usable solution for alleviating the managerial decision difficulties.

The problem is formulated as a dynamic programming model composed of N stages and is programmed in the FORTRAN IV language. The proposed algorithm selects the minimum cost route through any configuration based on an input consisting of location co-ordinates and known parameters.

I wish to express my appreciation to Professors Wilson J. Bentley, Earl J. Ferguson, J. Leroy Folks, James E. Shamblin, and M. Palmer Terrell, who have served as members of my doctoral committee. An especially grateful acknowledgment should be given to Dr. Terrell for the personal interest, encouragement, and assistance he has rendered while serving as adviser of this research endeavor. I also wish to express my gratitude to Colonel Frank F. Groseclose, Director Emeritus of the School of Industrial Engineering at the Georgia Institute of Technology. Professor Groseclose has given me invaluable guidance and encouragement during my pursuance of doctoral studies.

Finally, I feel the knowledge and philosophies instilled in me during my tenure at this institution have had a very beneficial influence on me. I am greatly indebted to the Oklahoma State University for the opportunity to have studied under and worked with such an outstanding faculty.

Typing and processing services of this dissertation were rendered by Miss Velda Davis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Orientation to Problem

Marine seismic exploration is an international business of which the primary concern is the detection of oil reserves beneath the floor of the ocean. At present, there are approximately one hundred geophysical crews collecting seismic data at strategic locations around the world. Because of the high costs associated with maintaining a geophysical crew, decisions concerning the management of the crews are very critical.

Seismic data is collected from beneath an ocean by exploding an energy source in the water, then recording the magnitudes of the noise reflections through a series of sensing devices. These sensing devices are located at equal intervals within a seismic marine cable which is towed by a ship. This cable, usually from one-half of a mile to two miles in length, must be aligned with the ship's movement when data is being collected.

A prospect at which data is to be collected consists of a configuration of straight lines, the lines indicating the locations where data is to be gathered. An example of a seismic prospect is shown in Figure 1. The number of lines

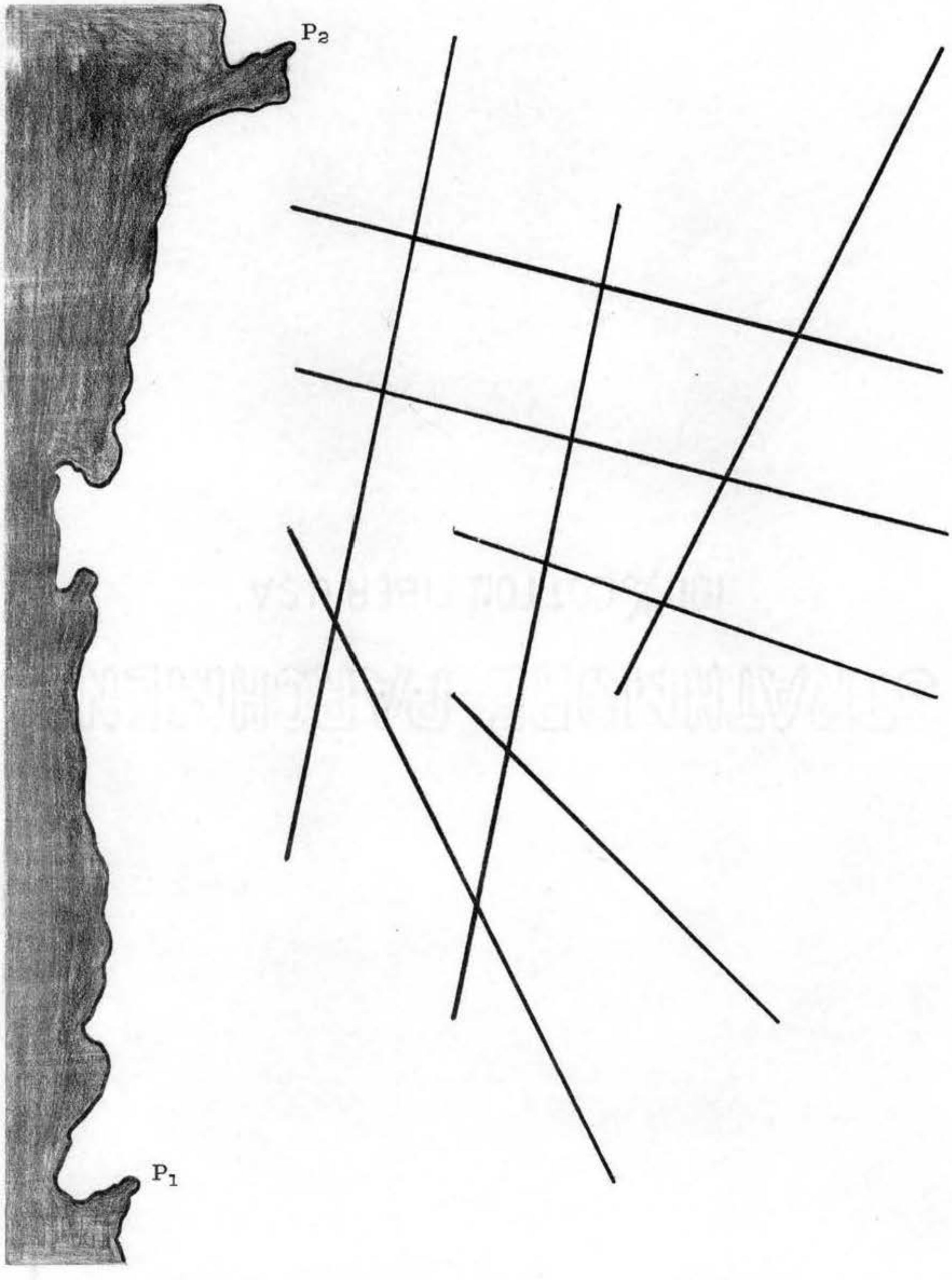


Figure 1. Seismic Prospect Configuration 1

at a given prospect might be as great as two dozen, with individual line lengths ranging from perhaps five miles to one hundred miles in length. To collect the required data, a ship must leave a known port P_1 and travel to the prospect, traverse each of the lines one time collecting data, then return to a known port P_2.

It is the responsibility of the management of a geophysical company to route a ship through a given configuration so as to collect the required data efficiently. Ships are presently being scheduled through seismic prospects by managerial discretion with a limited number of calculations. Because of the extremely large number of feasible paths that could be selected and the great expense of maintaining a marine crew, a decision based on subjective judgment is highly vulnerable to costly error.

The objective of this investigation is to develop an algorithm which, with the aid of a digital computer, will select the minimum cost route through any given seismic prospect configuration.

Mathematical Statement of Problem

By joining the departation and termination ports with a line, the configuration is changed from one of N lines to one having N+1 lines. The problem now becomes selecting the minimum circuit that covers the N+1 lines. Let the interchange portion of a circuit which traverses the N+1 lines be denoted by D.

This problem can be stated as follows:

given

$d(i,j,m,n)$ = distance from line i, end j to line m, end n

$x(i,j,m,n) = 1$ if line i, end j is linked to line m, end n

$x(i,j,m,n) = 0$ otherwise

where

$i = 1,2,3, \ldots, N+1 \qquad j = 1,2$

$m = 1,2,3, \ldots, N+1 \qquad n = 1,2$

minimize

$$D = \sum_{i=1}^{N+1} \sum_{j=1}^{2} \sum_{m=1}^{N+1} \sum_{n=1}^{2} d(i,j,m,n)\ x(i,j,m,n)$$

subject to

$$\sum_{\substack{m=1 \\ m \neq i}}^{N+1} \sum_{n=1}^{2} x(i,1,m,n) + \sum_{\substack{p=1 \\ p \neq i}}^{N+1} \sum_{q=1}^{2} x(i,2,p,q) = 1$$

$$\sum_{\substack{m=1 \\ m \neq i}}^{N+1} \sum_{n=1}^{2} x(m,n,i,j) + \sum_{\substack{p=1 \\ p \neq i}}^{N+1} \sum_{q=1}^{2} x(i,j,p,q) = 1$$

where

$i = 1,2,3, \ldots, N+1 \qquad j = 1,2$

and

$x(i,j,m,n) = 0$ or 1

where

$$i = 1,2,3, \ldots, N+1 \qquad j = 1,2$$

$$m = 1,2,3, \ldots, N+1 \qquad n = 1,2.$$

Literature Survey

A thorough search of the literature reveals that very little research has been published pertinent to this specific problem. An analogy can be made, however, with the proposed problem and the classical traveling salesman problem. The objective of the traveling salesman problem is to select the route that will minimize the total distance traveled in visiting N cities once and only once and returning to the starting city. A seismic configuration consisting of N lines is comparable to a 2N+2 city traveling salesman problem which is constrained such that cities are visited in specified pairs. The unconstrained traveling salesman problem has been treated by a number of persons using a variety of techniques. Several of the more important contributions will be discussed in this chapter.

One of the earliest investigations was made by Dantzig, Fulkerson, and Johnson (9) in 1954. Their publication outlines a linear programming approach to the problem. Their approach starts with an arbitrary solution, then employs the standard simplex method to improve the basis. A link in the basis is replaced by a new link in each iteration. Since a link which has been removed can be re-introduced at a later iteration, this approach is highly inefficient. Because of the additional constraints that would need to be imposed, a linear programming formulation of the proposed problem would be extremely large. Since the above original research was performed, other attempts have been made to use linear programming to solve the traveling salesman problem. Because of the nature of the problem, however, little success has been achieved.

Miller, Tucker, and Zemlin (17) formulated the traveling salesman problem as an integer programming problem. Using this technique, an N-city problem required N^2+N constraints and N^2 variables. The authors concluded that the integer programming procedure was highly inefficient.

In 1962, Bellman (3) employed dynamic programming to obtain the optimal route a salesman should travel. Using this approach, the traveling salesman problem was formulated as a multi-stage decision problem. The optimal path segments obtained from a particular stage are retained and used in obtaining the optimal segmental routes in subsequent stages. The author points out that, although this method of attack is highly efficient, the algorithm is faced with a storage problem for an arbitrarily large number of cities.

A year later, Little, Murty, Sweeney, and Karel (16) developed a "branch-and-bound" algorithm for the traveling salesman problem. This algorithm divides all paths into two categories. The first category includes all paths containing a directional link connecting two particular cities, whereas the second category includes all remaining paths that exclude the selected link. At every stage where this separation of paths or "branching" occurs, a lower bound is calculated for each of the sets of paths within each of the above categories. At each branching stage, the directional link is selected in such a manner that the lower bound for the set of paths not containing the link in question will be as large as possible. The optimal route is determined once a circuit is found where the total distance required to be traveled is smaller than the lower bound of each of the other path segments, respectively. Using both the execution time and memory storage requirements as criteria, this algorithm has been shown to be the most efficient method to date for solving the traveling salesman problem (5, p. 555).

CHAPTER II

OPTIMIZATION MODEL

General Statement of Problem

The following notation will be used in formulating the multi-stage model of the problem:

N = Number of lines in the configuration

Distance from P_1 to line i, end j = $\alpha(i,j)$

Distance from line i, end j to line m, end n
= $\delta(i,j,m,n)$

Distance from line m, end n to P_2 = $\beta(m,n)$.

Given

$$\alpha(i,j),\ \delta(i,j,m,n),\ \beta(m,n) \qquad i = 1,2,3,\ \ldots,\ N \quad j = 1,2$$
$$m = 1,2,3,\ \ldots,\ N \quad n = 1,2$$

determine the sequence containing N+1 elements that minimizes D with D being defined as follows:

$$D = \alpha(L_1,E_1) + \delta(L_1,E_2,L_2,E_3) + \delta(L_2,E_4,L_3,E_5) + \ldots +$$
$$\delta(L_{N-1},E_{2N-2},L_N,E_{2N-1}) + \beta(L_N,E_{2N})$$

where

$L_1, L_2, L_3, \ldots, L_N$ is a permutation of the integers 1 through N

$E_i = 1,2 \qquad i = 1,2,3,\ \ldots,\ 2N$

$E_i + E_{i+1} = 3 \qquad i = 1,3,5,\ \ldots,\ 2N-1.$

In general, there are $2^N N!$ possible routes through an N-line configuration that would have to be considered if an exhaustive enumeration is to be performed. To reduce the required number of paths to be considered, Bellman's "principle of optimality" is employed, which states:

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision (2, p. 83).

To utilize this principle, the problem will be formulated as a dynamic programming model composed of N stages.

Stage 1

Let subscript ξ_1 denote the line to be covered immediately prior to returning to port P_2. Prior to approaching line ξ_1, the ship will change lines from a second line. Let this second line be designated by the subscript i. The last line covered (ξ_1) can be traversed in either one of two possible directions. Cumulative penalties of these two path segments from line i, end j to port P_2 can be calculated as follows:

$$f_1(i,j,\xi_1) = \delta(i,j,\xi_1,1) + \beta(\xi_1,2)$$

$$f_1'(i,j,\xi_1) = \delta(i,j,\xi_1,2) + \beta(\xi_1,1).$$

Let $f_1^*(i,j,\xi_1)$ denote the shortest path segment from line i, end j to P_2, covering line ξ_1. If $f_1(i,j,\xi_1) <$ $f_1'(i,j,\xi_1)$, then $f_1^*(i,j,\xi_1) = f_1(i,j,\xi_1)$. Line ξ_1 would then always be attacked in the order of $i,j \rightarrow \xi_1,1 \rightarrow \xi_1,2 \rightarrow$

P_2, should line ξ_1 be in reality the immediate predecessor to P_2 and the immediate successor to line i.

If $f_1(i,j,\xi_1) > f_1'(i,j,\xi_1)$, then $f_1^*(i,j,\xi_1) = f_1'(i,j,\xi_1)$. In this case, line ξ_1 would always be traversed in a sequence of $i,j \rightarrow \xi_1,2 \rightarrow \xi_1,1 \rightarrow P_2$. If $f_1(i,j,\xi_1) = f_1'(i,j,\xi_1)$, line ξ_1 could be traversed in either direction.

The above procedure is extended by calculating the values of $f_1^*(i,j,\xi_1)$ using the relationship

$$f_1^*(i,j,\xi_1) = \underset{\substack{n=1,2 \\ \bar{n}=3-n}}{\text{minimum}} \{\delta(i,j,\xi_1,n) + \beta(\xi_1,\bar{n})\}$$

for the following states:

$$i = 1,2,3, \ldots, N$$

$$j = 1,2$$

$$\xi_1 = 1,2,3, \ldots, N \quad \xi_1 \neq i.$$

This procedure composes STAGE 1 of the optimization algorithm. It is apparent that one-half of the feasible paths in the N-line configuration are eliminated from further consideration in this stage.

Stage 2

Since the "principle of optimality" is being utilized, the optimal values determined in STAGE 1 are used in obtaining the optimal values for STAGE 2. The recurrence relationship between STAGE 2 and STAGE 1 is:

$$f_2(i,j,m,n,\xi_1) = \delta(i,j,m,n) + f_1^*(m,\bar{n},\xi_1)$$

$$n = 1,2 \qquad \bar{n} = 3-n.$$

Subscript ξ_1 is an index denoting the last line in the configuration to be traversed, and $i \rightarrow m \rightarrow \xi_1$ is the line sequence in which the ship is routed prior to returning to port P_2.

The minimum distance from line i, end j to port P_2 that traverses two lines can be determined using the following expression:

$$f_2^*(i,j,\xi_2) = \underset{\substack{k = m,\xi_1 \\ n = 1,2}}{\text{minimum}} \; f_2(i,j,k,n,\bar{k})$$

where

$m = 1,2,3, \ldots, N \qquad m \neq i$

$\xi_1 = 1,2,3, \ldots, N \qquad \xi_1 \neq i,m$

if $k = m$, then $\bar{k} = \xi_1$

if $k = \xi_1$, then $\bar{k} = m$

ξ_2 = an index denoting a unique combination of lines m and ξ_1.

Since k and n each assume two values respectively, four feasible paths are considered when determining a value for a particular $f_2^*(i,j,\xi_2)$. Only the minimum of these four routings is retained for the optimal policy, hence permanently eliminating all paths containing the other three possible path segments from future consideration.

This elimination procedure is repeated with values of

$f_2^*(i,j,\xi_2)$ being determined for the following states:

$i = 1,2,3, \ldots, N$

$j = 1,2$

ξ_2 = a unique combination of two lines (neither = i) taken from the N lines.

For given values of i and j, ξ_2 will assume values representing all combinations of two lines taken from the N lines of the configuration, line i excluded.

STAGE 2 thus eliminates seventy-five per cent of the remaining feasible paths from further consideration. This reduction, coupled with the fifty per cent reduction in STAGE 1, reduces the number of paths as candidates for the minimum to 12.5% of the original number as STAGE 3 is entered.

Stage K (K = 3,4,5, ..., N-1)

The optimization procedure described for STAGE 2 can be generalized to be applicable for the K^{th} stage. The cumulative distance traveled from line i, end j to port P_2 can be calculated by using the following recurrence equation:

$$f_K(i,j,m,n,\xi_{K-1}) = \delta(i,j,m,n) + f_{K-1}^*(m,\bar{n},\xi_{K-1})$$

where

$i = 1,2,3, \ldots, N$

$m = 1,2,3, \ldots, N \qquad m \neq i$

$j = 1,2$

$n = 1,2$

$\bar{n} = 3-n$.

Subscripts m and n are the identification of the end of the line to be traversed immediately after having traversed line i and departed line i at end j. Index ξ_{K-1} identifies a unique combination of K-1 lines taken from N, none of which are m or i, to be traversed between line m and port P_2.

The shortest path segment to P_2 from line i, end j having covered K lines can be determined using the following relationship:

$$f_K^*(i,j,\xi_K) = \underset{\substack{m=1,2,\ldots,N \quad m\neq i \\ n=1,2 \\ \text{all } \xi_{K-1}}}{\text{minimum}} f_K(i,j,m,n,\xi_{K-1})$$

where

$i = 1,2,3, \ldots, N$

$j = 1,2.$

Index ξ_{K-1} denotes a unique combination of K-1 lines taken from the N lines of the configuration, excluding lines m and i. Index ξ_K denotes the set of lines identified by ξ_{K-1} plus line m.

Stage N

STAGE N is the last stage considered in developing the minimum path in a N-line configuration. This final stage compares the path lengths of each of the 2N remaining paths, one of which is the optimum, which were generated in STAGES 1 through N-1. The recurrence relationship for this stage is:

$$f_N(i,j) = \alpha(i,j) + f_{N-1}(i,\bar{j},\xi_{N-1})$$

where

$i = 1,2,3, \ldots, N$

$j = 1,2$

$\bar{j} = 3-j.$

The index ξ_{N-1} denotes all lines in the configuration with the exception of line i.

The minimum route through the configuration can be determined as follows:

$$f_N^*(P_1) = \underset{\substack{i=1,2,\ldots,N \\ j=1,2.}}{\text{minimum}} f_N^*(i,j).$$

The first position (L_N^*, E_N^*) to which the ship will be routed after leaving port P_1 will be that combination of i and j associated with $f_N^*(N_1)$.

Backtrack Routine for Optimal Path

Having obtained the values for L_N^*, E_N^*, and $f_N^*(P_1)$ in STAGE N, a backtrack procedure can be employed to obtain the optimal route through the prospect configuration. Line L_N^* and end E_N^* identify the first position to which the ship will travel after leaving port P_1. The ship will then traverse line L_N^* to the end opposite E_N^*. Let this opposite end be designated as $\bar{E}_N^*$. The successor to $L_N^*, \bar{E}_N^*$ on the optimal route can be found by examining STAGE N-1. This point will be the L_{N-1}^*, E_{N-1}^* associated with $f_{N-1}^*(L_N^*, \overline{E}_N^*, \xi_{N-1})$, where ξ_{N-1} is an index that identifies the particular combination of all lines of the configuration with the exception of L_N^*.

The above procedure can be repeated to determine the third line of the optimal sequence and the direction it will be traversed. Let $\overline{E}_{N-1}^*$ denote the end of line L_{N-1}^* from which the ship will depart as it changes lines. The next position to which the ship will be routed will be the L_{N-2}^* and E_{N-2}^* associated with $f_{N-2}^*(L_{N-1}^*, \overline{E}_{N-1}^*, \xi_{N-2})$, where ξ_{N-2} identifies the combination of the N lines excluding lines L_N^* and L_{N-1}^*.

Continuing this reverse movement through the N stages of the model, the complete sequence of the minimum path can be determined. The optimal route is determined as being the following sequence:

$$P_1 \rightarrow L_N^*, E_N^* \rightarrow L_N^*, \overline{E}_N^* \rightarrow L_{N-1}^*, E_{N-1}^* \rightarrow L_{N-1}^*, \overline{E}_{N-1}^* \rightarrow L_{N-2}^*, E_{N-2}^* \rightarrow \ldots$$

$$\ldots\ L_2^*, E_2^* \rightarrow L_2^*, \overline{E}_2^* \rightarrow L_1^*, E_1^* \rightarrow L_1^*, \overline{E}_1^* \rightarrow P_2.$$

CHAPTER III

PROGRAMMED ALGORITHM

The optimization algorithm was programmed in the FORTRAN IV language. The programmed instructions (Figure 2) implement the theory presented in the preceding chapter. The programmed algorithm will select the optimal route for a configuration of ten lines or less and requires a computer memory capacity of approximately 250,000 bytes. As shown in Figure 3, the execution time required for selecting the optimal route ranges from 0.01 minute for a three line configuration to approximately 0.44 minute for a ten line configuration using an IBM 360/50 computer, "G" LEVEL. The optimization procedure can be easily extended for larger configurations, however, by using auxiliary equipment such as tapes or discs for temporary storage of the required matrices.

A transformation routine was employed for identifying unique line combinations represented by the ξ_K index described in Chapter II. Values of this index were generated for permanent identification of the lines being identified by first assigning a unique numerical value (θ_1) to each of the lines, then using a series of nested DO LOOPS for each stage to generate the transformation

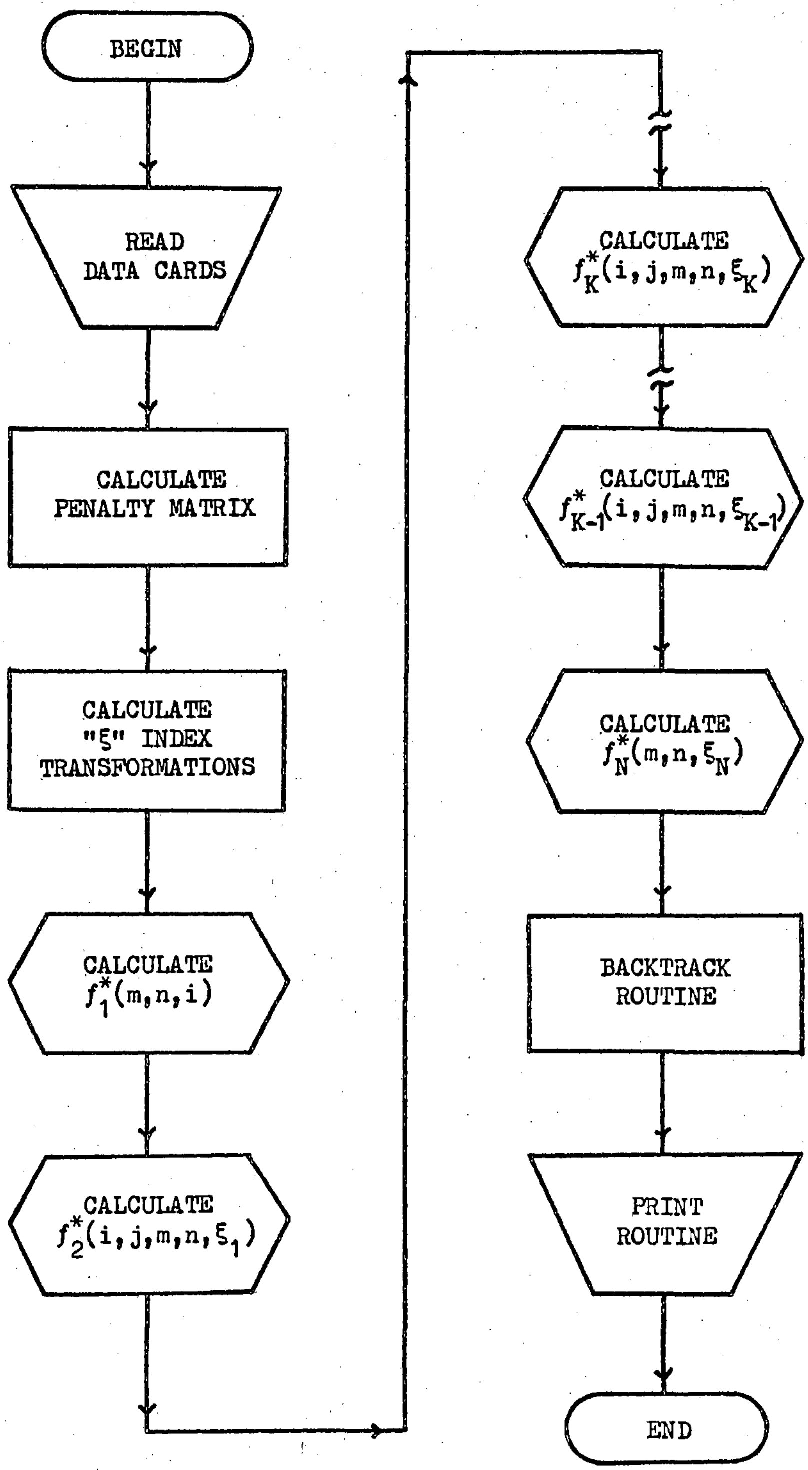


Figure 2. Summary Flow Chart

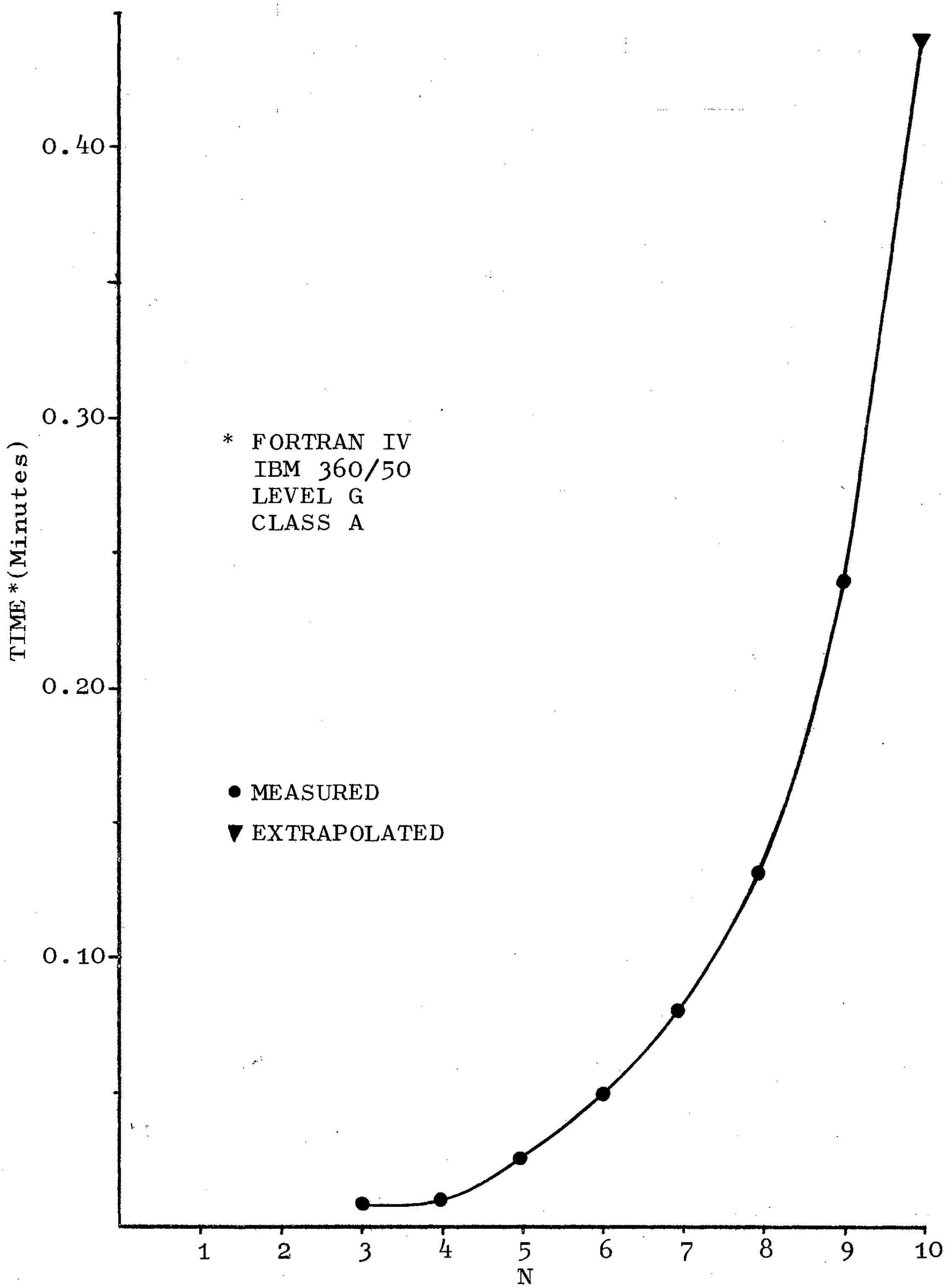


Figure 3. Execution Time for N-Line Configuration

identifications. The numerical values that were assigned to the lines needed to have the property that the sum of the individual θ_i values (ξ_K) would also be a unique number within each stage. This index is defined as follows:

$$\xi_K = f_K\left[\sum_{i \in k} \theta_i\right] \qquad K=1,2,3,\ \ldots,\ N-1$$

where k denotes a particular combination of K lines that composes a ξ_K index.

This procedure can be illustrated by using a simple example configuration composed of four lines. The values of θ_i and ξ_K for the respective combinations are defined as follows:

STAGE 1			STAGE 2			STAGE 3		
Line	θ_i	ξ_1	Combination	$\Sigma\theta_i$	ξ_2	Combination	$\Sigma\theta_i$	ξ_3
A	1	1	AB	3	1	ABC	7	1
B	2	2	AC	5	2	ABD	10	2
C	4	3	AD	8	3	ACD	12	3
D	7	4	BC	6	4	BCD	13	4
			BD	9	5			
			CD	11	6			

The advantage of this transformation is that it uniquely identifies a particular combination of K lines for computation purposes by simply adding the θ_i values for the K particular lines under consideration. As an example, consider the minimum path segment from C_2 to P_2, having covered lines B and D. Since two lines are traversed, the stage under consideration is STAGE 2.

$$\theta_B + \theta_D = 2 + 7 = 9 \qquad\qquad \xi_{BD} = f_2[9] = 5.$$

Hence, this path segment would be designated as $f_2^*(3,2,5)$ within the computer.

The above procedure allows the ξ_K index value to be as small in magnitude as is possible, therefore allowing the memory storage requirements for the necessary matrices associated with $f_K^*(i,j,\xi_K)$ to be a minimum. To further reduce the core requirements, the smallest integers that satisfied the uniqueness property stated earlier in this chapter were selected as the numerical values of the θ_i's.

The programmed algorithm will select the optimum path based upon any given initial penalty matrix containing the values of $\alpha(i,j)$, $\beta(m,n)$, and $\delta(i,j,m,n)$. The problem was originally attacked with the objective of minimizing the total distance required to be traveled to traverse the N lines of a given configuration. It is believed that in practice a more appropriate criterion for generating the initial penalty matrix is travel time.

When the recording ship is in port, the seismic streamer is wound on a reel aboard the ship. When collecting data on a line, this streamer must be laid out in the water and towed by the ship. The axis of the streamer must lie in line with the line being traversed when data is being gathered. To lay out or pull in a streamer requires approximately one to three hours, depending on the streamer length and the mechanical equipment installed on the ship. With the streamer aboard the ship, the average ship can travel approximately ten to fifteen knots per hour. If the streamer is towed, the ship's speed is reduced to approximately five knots per hour because of the severe drag.

Because of the difference in speeds with the streamer in or out, there is a break-even distance where it is equally advantageous to leave the cable in the water and change lines at the slower speed or pull in the cable and travel to the next line faster. To allow the streamer to be always in the correct position when data is being gathered, the computer calculates a new set of co-ordinates (B_2') for the approached end of each line.

The following is a summary of how the penalty matrix is calculated to depict more accurately the costs actually incurred when changing lines.

Define

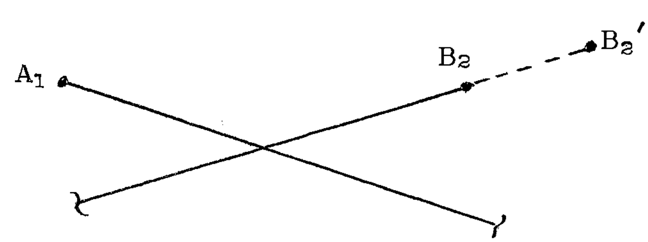
T_{in} = time required to pull cable in

T_{out} = time required to lay cable out

L = streamer length

S_{in} = speed w/cable in

S_{out} = speed w/cable out.



The calculation of the penalty for changing lines from A_1 to B_2 is as follows:

1) Calculate new co-ordinates (B_2') for B_2.

2) Calculate distance (D) from A_1 to B_2'.

3) Calculate break-even travel distance (D^*).

$$D^* = (T_{in} + T_{out})/[(1/S_{out}) - (1/S_{in})]$$

Finally, if $D < D^*$, then $\delta(A,1,B,2) = D/S_{out}$

If $D \geq D^*$, then $\delta(A,1,B,2) = [D/S_{in}] + T_{in} + T_{out}$.

Also, $\alpha(B,2) = [(\text{distance from } P_1 \text{ to } B_2')/S_{in}] + T_{out}$,

and $\beta(B,2) = [(\text{distance from } B_2 \text{ to } P_2)/S_{in}] + T_{in}$.

A representative example of a seismic prospect configuration is shown in Figure 3. By calculating and using a penalty matrix as just described, the optimal route a ship should follow is shown in Table II. Also included in the output is the position of the streamer during each line change that minimizes the line change time. Mileages and times are printed for each of the individual path segments and for the total prospect. The total times and distances are divided into productive and non-productive portions. Production time is the time when the crew is actually collecting the seismic data, whereas the latter is the elapsed time going both to and from the prospect and changing lines. Although the identification of the optimal route is the information that is of primary importance, the additional information aids both the party manager aboard ship and the office executive management in effectively utilizing the ship.

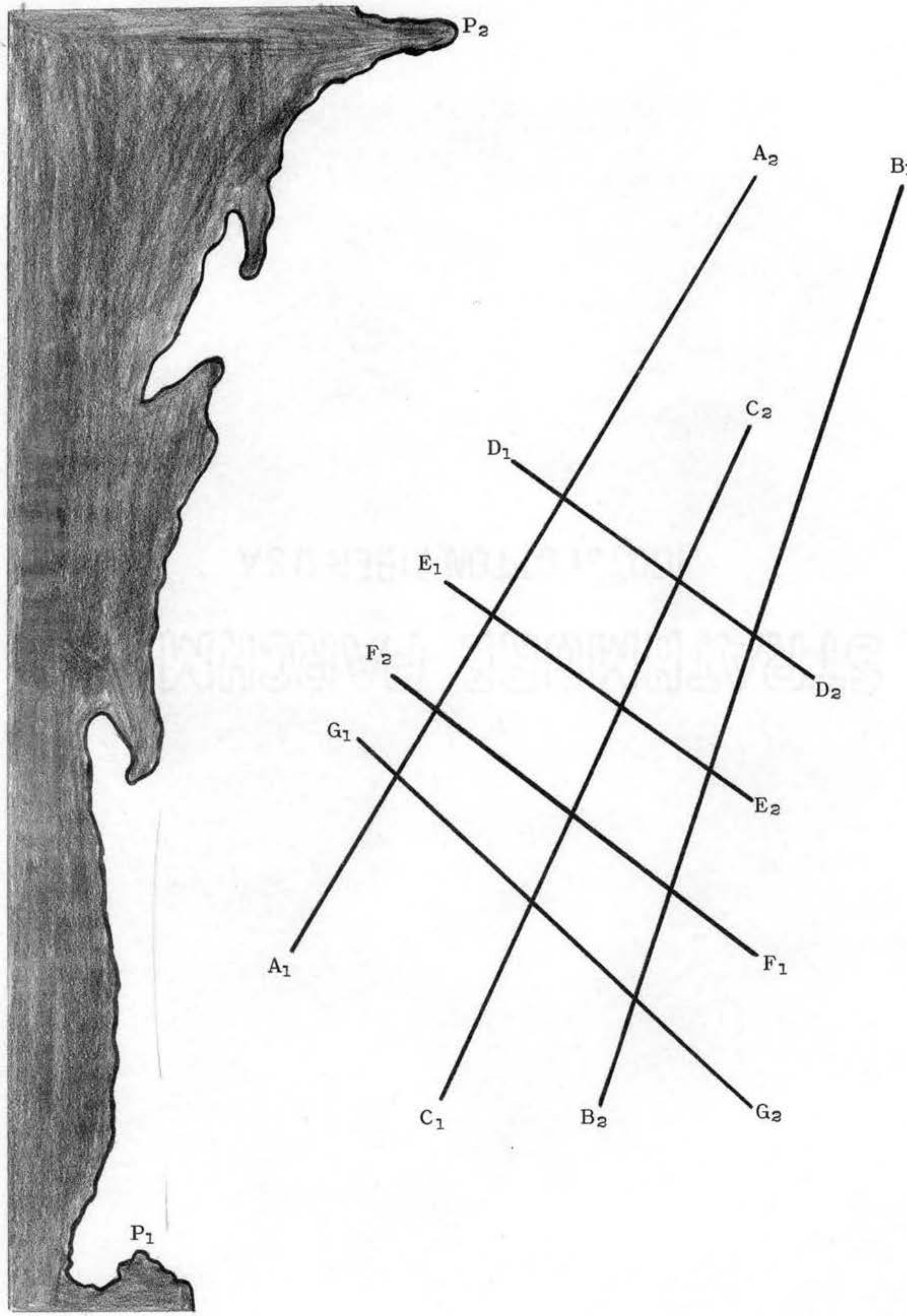


Figure 4. Seismic Prospect Configuration 2

TABLE I

INPUT DATA - CONFIGURATION 2

PORT AND LINE CO-ORDINATES

	X1	Y1	X2	Y2
P	5.	5.	15.	45.
A	10.	15.	25.	40.
B	30.	40.	20.	10.
C	15.	10.	25.	32.
D	17.	31.	27.	24.
E	15.	27.	25.	20.
F	25.	15.	13.	24.
G	12.	22.	25.	10.

PARAMETERS

CABLE LENGTH = 7874 FEET
SCALE = 0.16 UNITS/MILE

SHIP SPEED (CABLE OUT) = 6.00 MPH
SHIP SPEED (CABLE IN) = 15.00 MPH

CABLE HANDLING TIME (LAY OUT) = 1.25 HOURS
CABLE HANDLING TIME (PULL IN) = 1.75 HOURS

TABLE II

OUTPUT DATA - CONFIGURATION 2

OPTIMUM PATH INFORMATION

FROM	TO	CABLE	MILES	HOURS
P2	A1	IN	70.	5.7
A1	A2	OUT	182.	30.4
A2	B1	IN	35.	5.2
B1	B2	OUT	198.	32.9
B2	G2	IN	36.	5.2
G2	G1	OUT	111.	18.4
G1	F2	OUT	18.	3.0
F2	F1	OUT	94.	15.6
F1	C1	IN	75.	7.8
C1	C2	OUT	151.	25.2
C2	E1	IN	74.	7.8
E1	E2	OUT	76.	12.7
E2	D2	OUT	31.	5.1
D2	D1	OUT	76.	12.7
D1	P2	IN	151.	11.8

	HOURS	MILES
PRODUCTIVE	148.0	888.
NON-PRODUCTIVE	42.4	464.
TOTAL	190.4	1352.

CHAPTER IV

SUMMARY AND CONCLUSIONS

The algorithm described in this dissertation selects the optimal route through any configuration based upon the given input consisting of location co-ordinates and known parameters. Programmed in FORTRAN IV, the algorithm requires a computer memory capacity of approximately 250,000 bytes for a ten line configuration, due to the large matrices which must be stored for each stage. Although all arrays must be kept to execute the backtrack routine, only the matrices for the predecessor stage are required when developing the matrices for any given stage. Because of this desirable feature, the algorithm presented in Chapter II is readily adaptable to using either discs or tapes for temporary storage of matrices while they are not needed. The time required to obtain the optimal path through an N-line prospect configuration is approximately $(1.84)^N(0.001)$ minutes when executed on an IBM 360/50 computer.

The proposed algorithm should be a valuable and powerful tool for enhancing the managerial decision-making of geophysical companies. Although a search of the literature reveals no previous research on this specific problem that could be used for comparison, the proposed dynamic programming formulation appears to be highly efficient relative to the amount of computation required but is limited due to the large storage space required. Since most geophysical companies have access to a large memory computer and a sophisticated communication system, the storage problem is not too critical. The optimal route a ship should follow can be determined in advance, using a large memory computer on the mainland and dispatched to the ship. Should the ship be forced to deviate from this pre-determined optimal route, a new schedule can be calculated, based on the current data and transmitted to the ship through the communication system.

Since many seismic ships in the near future will be equipped with small memory digital computers, it is proposed that formulations other than dynamic programming be made of this problem to reduce the storage requirements. Of the published methods reviewed in the literature, the "branch-and-bound" algorithm appears to have the greatest potential (16).

Although this algorithm was developed for the primary objective of alleviating the managerial decision difficulties involved in scheduling geophysical ships, there are other applications where the algorithm can be used. Probably the most apparent of these other applications is finding the solution to any constrained traveling salesman problem. An N-line seismic configuration is equivalent to a traveling salesman problem composed of N+1 sets of cities, where a set is defined as an ordered sequence of one or more cities to be visited. The two ends of a particular seismic line would be analogous to the two end cities of a constraining sequence. The other major application of this algorithm would involve scheduling jobs that could be produced by a machine using one of two possible methods. In this latter application, there would be N jobs to be performed by one machine, where the machine set-up cost for a given job and method is dependent on the previous job and method performed by the machine. The penalty matrix would include the costs incurred by changing from job i, method j to job m, method n.

BIBLIOGRAPHY

(1) Barachet, B. L. "Graphic Solution of the Traveling Salesman Problem." Operations Research, Vol. 5 (1957), 841-45.

(2) Bellman, R. E. Dynamic Programming. Princeton: Princeton University Press, 1957.

(3) Bellman, R. E. "Dynamic Programming Treatment for the Traveling Salesman Problem." Association for Computing Machinery, Vol. 9 (1962), 61-63.

(4) Bellman, R. E. "On an Optimal Routing Problem." Quarterly Applied Mathematics, Vol. 16 (1958), 87-90.

(5) Bellmore, M., and G. L. Nemhauser. "The Traveling Salesman Problem: A Survey." Operations Research, Vol. 16 (1968), 538-58.

(6) Croes, G. A. "A Method for Solving Traveling Salesman Problems." Operations Research, Vol. 6 (1958), 791-812.

(7) Dantzig, G. B. "On the Shortest Route Through a Network." Management Science, Vol. 6 (1960), 187-90.

(8) Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson. "On a Linear Program - Combinatorial Approach to the Traveling Salesman Problem." Operations Research, Vol. 7 (1959), 58-66.

(9) Dantzig, G. B., D. R. Fulkerson, and S. M. Johnson. "Solution of a Large-Scale Traveling Salesman Problem." Operations Research, Vol. 2 (1954), 393-410.

(10) Dreyfus, S. E. "An Appraisal of Some Shortest-Path Algorithms." Operations Research, Vol. 17 (1969), 395-412.

(11) Flood, M. M. "The Traveling Salesman Problem." Operations Research, Vol. 4 (1956), 61-75.

(12) Hu, T. C. "A Decomposition Algorithm for Shortest Paths in a Network." Operations Research, Vol. 16 (1968), 91-102.

(13) Isaac, A. M., and E. Turban. "Some Comments on the Traveling Salesman Problem." Operations Research, Vol. 17 (1969), 543-46.

(14) Lin, S. "Computer Solutions of Traveling Salesman Problem." Bell System Technical Journal, Vol. 44 (1965), 2245-69.

(15) Lin, S. "Found: A Rapid Route to the Shortest Path." Journal of Engineering Education (1966), 89.

(16) Little, J. D., K. G. Murty, D. W. Sweeney, and C. Karel. "An Algorithm for the Traveling Salesman Problem." Operations Research, Vol. 11 (1963), 972-89.

(17) Miller, C. E., A. W. Tucker, and R. A. Zemlin. "Integer Programming and the Traveling Salesman Problem." Association of Computing Machinery Journal, Vol. 7 (1960), 326-29.

(18) Nicholson, T. A. "Finding the Shortest Route Between Two Points in a Network." Computer Journal, Vol. 9 (1966), 275-80.

(19) Obruca, A. K. "Spanning Tree Manipulation and the Traveling Salesman Problem." Computer Journal, Vol. 10 (1968), 374-77.

(20) Peart, R. M., R. M. Randolph, and T. E. Bartlett. "The Shortest Route Problem." Operations Research, Vol. 8 (1960), 866-68.

(21) Perko, A. "Some Computational Notes on the Shortest Route Problem." Computer Journal, Vol. 8 (1965), 19.

(22) Pollack, M., and W. Weibenson. "Solutions of the Shortest Route Problem - A Review." Operations Research, Vol. 8 (1960), 224-30.

(23) Rossman, M. J., R. J. Twery, and F. D. Stone. "A Solution to the Traveling Salesman Problem by Combinatorial Programming." Operations Research, Vol. 6 (1958), 897.

(24) Rothkopf, M. "Traveling Salesman Problem: On the Reduction of Certain Large Problems to Smaller Ones." Operations Research, Vol. 14 (1966), 532-33.

(25) Saksena, J. P., and S. Human. "Routing Problem With K Specified Nodes." <u>Operations Research</u>, Vol. 14 (1966), 909-13.

VITA

Edward Payson Willard

Candidate for the Degree of

Doctor of Philosophy

Thesis: AN ALGORITHM FOR OPTIMAL SHIP ROUTING FOR SEISMIC DATA COLLECTION

Major Field: Engineering

Biographical:

Personal Data: Born in Commerce, Texas, October 31, 1939, the son of Dr. and Mrs. E. P. Willard.

Education: Graduated from Highland Park High School, Dallas, Texas, in May, 1957; received Bachelor of Science in Mechanical Engineering from Southern Methodist University in 1962; received Bachelor of Science in Industrial Engineering from Southern Methodist University in 1963; received Master of Science in Industrial Engineering from Southern Methodist University in 1964; enrolled in doctoral program at the Georgia Institute of Technology, 1964-66; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1970.

Professional Experience: Co-operative Engineer, Johnson Controls, 1959-63; graduate teaching assistant, School of Industrial Engineering, Georgia Institute of Technology, 1964-66; Senior Industrial Engineer, Texas Instruments Incorporated, 1966-68; evening instructor, Southern Methodist University, 1967; graduate teaching assistant, Oklahoma State University, School of Industrial Engineering, 1968-69; member of Technical Staff, Texas Instruments Incorporated, 1970.