NBS TECHNICAL NOTE 787

Heuristic Cost Optimization of the Federal Telpak Network

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of a Center for Radiation Research, an Office of Measurement Services and the following divisions:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Nuclear Sciences [2] — Applied Radiation [2] — Quantum Electronics [3] — Electromagnetics [3] — Time and Frequency [3] — Laboratory Astrophysics [3] — Cryogenics [3].

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials and the following divisions:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations leading to the development of technological standards (including mandatory safety standards), codes and methods of test; and provides technical advice and services to Government agencies upon request. The Institute consists of a Center for Building Technology and the following divisions and offices:

Engineering and Product Standards — Weights and Measures — Invention and Innovation — Product Evaluation Technology — Electronic Technology — Technical Analysis — Measurement Engineering — Structures, Materials, and Life Safety [4] — Building Environment [4] — Technical Evaluation and Application [4] — Fire Technology.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus within the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

Information Processing Standards — Computer Information — Computer Services — Systems Development — Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal Government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data — Office of Technical Information and Publications — Library — Office of International Relations.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.

[2] Part of the Center for Radiation Research.

[3] Located at Boulder, Colorado 80302.

[4] Part of the Center for Building Technology.

Heuristic Cost Optimization of the Federal Telpak Network

R.G. Saltman,
G.R. Bolotsky,
and
Z.G. Ruthberg

Systems Development Division
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Frederick B. Dent, *Secretary*
NATIONAL BUREAU OF STANDARDS, Richard W. Roberts, *Director*
Issued June 1973

National Bureau of Standards Technical Note 787

Nat. Bur. Stand. (U.S.), Tech. Note 787, 52 pages (June 1973)

CODEN: NBTNAE

For sole by the Superintendent of Documents, U.S. Government Printing Office, Woshington, D.C., 20402.

(Order by SD Catolog No. (13.46:787).

Price $0.80 domestic postpoid or $0.55 G.P.O. Bookstore.

CONTENTS

TABLES | Page

We would like to acknowledge the efforts of Mr. T. Zois for his work in obtaining and processing the initial network, and Mr. J. Grossmon for his programming contributions while a summer student employee at NBS in 1970.

HEURISTIC COST OPTIMIZATION OF THE FEDERAL TELPAK NETWORK*

R. G. Saltman

G. R. Bolotsky

Z. G. Ruthberg

A heuristic method of optimizing the design of a very large communications network is described. The procedure is employed to configure the routes of 5552 communications service requests involving 1633 nodes. A FORTRAN IV program was developed to solve for actual needs of the Defense Communications Agency for leased-line service employing the Telpak tariff structure.

Key words: Communications network; computer program; heuristic; minimum cost; network configuration; optimization; Telpak rate structure.

1.0 INTRODUCTION

1.1 The Network Problem

The Federal Government leases, on a monthly basis, a large number of interstate voice-grade circuits for its own use under the Telpak tariff. This tariff offers savings for bulk requirements, basing costs on

---

*Sponsored by Defense Communications Agency, Reimbursable Order Nos. 70-19,71-36,72-78.

per-mile of circuits leased rather than number of calls made. In 1969, leasing costs for these communications services, managed by the DOD Defense Commercial Communications Office (DECCO), were $3,773,000/month. In that same year, the Institute for Computer Sciences and Technology, National Bureau of Standards, was asked by the Defense Communications Agency, the parent organization of DECCO, to explore the use of a computer to automatically configure and optimize the DECCO DOD Telpak Network, which was being optimized manually by DECCO personnel. This report summarizes the results of this effort.

The DECCO DOD network consisted at that time, of 1633 rate-centers with sufficient circuit linkage to satisfy the needs of 552 "requests". A "request" is a requirement for continuous, leased service submitted by using agencies. It consists of two items: (1) a unique node pair which constitutes the end points (terminals) of the service and (2) the number of circuits desired to connect these end points. All requirements submitted by different agencies for the same node pair are summed to determine the circuit requirements of a request. The routing of all requests for minimum cost is the essential substance of this problem. The outputs of the solution to the routing problem are (1) the sequence of nodes through which each request passes between its end points, (2) the listing and costing of the individual node-to-node links, and (3) the summarized cost of all links. The total requests for Telpak service which this R&D effort was to solve consisted of approximately 35,000 voice-grade lines.

1.2 The Telpak Problem's Value

The advantage of the Telpak bulk rate structure can be seen from Table 1. Whereas the cheapest single circuit lease (Ixc)* rate was 75¢/mile/month in 1969, the Telpak rate in that year brought the single circuit cost down to 47¢/mile/month for complete C bundles of 60 circuits and 25¢/mile/month for complete D bundles of 240 circuits. At present all costs are proportionately higher. The essential fact about the C and D trunks is that once a trunk is leased for whatever number of circuits less than maximum, any empty space up to its maximum capacity can be filled at zero additional cost.

---

*Ixc refers to the Interexchange tariff which is the mileage rate per single leased private line as opposed to Telpak bulk rates.

DECCO's current handling of the network involves the following manually performed operations: keeping records on all request routings; reconfiguring the network in an evolutionary way by inserting configuration changes at frequent intervals. If a computerization and optimization of these procedures could be accomplished, then the following could occur:

(1) there might be significant cost-savings to the Government, even if only a few percent of the total cost were saved;

(2) re-optimizations, due to changes in tariffs, could be carried out rapidly;

(3) the effects of proposed tariff changes could be analyzed rapidly;

(4) the locations of inefficient sections or connections of the network could be determined more systematically.

Table 1 Rate Structure for Ixc* and Telpak

a) Ixc Rates

Distance in Miles	Cost/Mile/Month
first 25 (1-25)	$3.00
next 75 (26-100)	2.10
next 150 (101-250)	1.50
next 250 (251-500)	1.05
over 500 (501-)	.75

b) Telpak Rates

Trunks	Maximum No. of Circuits In Trunk	Trunk Cost/Mile/Month	Circuit Cost/Mile/Month (filled trunk)
C	60	$28	$0.47
D	240	$60	$0.25

1.3 The Solution Method

Because of the nature of the Telpak tariff, direct terminal-to-terminal connections could not minimize the cost of routing requests. There are significant bulk savings when request routes or sections of routes are combined into groups of 60 and 240. In addition, since cost as a function of number of combined circuits is not in a closed form and, moreover, is non-linear, the optimization problem could not be handled by presently available mathematical programming methods. Lastly, the size of the network compounded the complexity of the problem. In view of these factors, the following general decisions were made:

a) Heuristic methods would be used throughout.

b) The resulting program would be tested on smaller subproblems that could be more easily manipulated at reasonable computer cost.

The specific method of solution used the following steps:

1) Choose an initial subproblem composed of the 250 cities (nodes) and 2394 requests that satisfy 87% of the required circuit-miles (see 3.1 for definition) in the entire problem. (PREPRO).

2) Find the shortest spanning tree* of this initial set of 250 cities and route the 2394 requests through it. (PASS1)

3) Decrease cost by decreasing the detour ratios** of these requests via a heuristic link-adding re-routing algorithm. (PASS1, cont.)

4) Configure, at minimum cost, the remaining 1383 cities and 3159 requests into this partially optimized network. (ADDREQ)

5) Decrease cost by introducing express trunks (long-distance, filled or very nearly filled D trunks) via another heuristic re-routing algorithm. (BYPASS)

6) Decrease cost by consolidating uneconomical links via still another heuristic re-routing algorithm, done in two passes. (PASS2, PASS3)

---

* See Appendix A for definitions and properties.

** Detour Ratio of a Request $= \dfrac{\text{no. of mi. in request rt. through network}}{\text{airline distance of request}}$

1.4 Problem Outputs

The main outputs containing the problem solution are as follows: (See Appendix D for sample printouts.)

1. The network links, their respective circuit fills, and their Telpak breakdown (C's, D's, Ixc's). A link is a unique direct connection between any two nodes and is assumed to travel in a straight line between the nodes.

2. The route of each request, listed by means of the links used. Links are identified by their two nodal end points.

3. The total network cost (TOTCST), calculated from the individual link costs.

Additional useful constants of the solution are discussed in Section 4.

1.5 Summary of Results

The complete problem containing 1633 cities and 5552 requests was configured by a FORTRAN IV computer program developed and run on IBM 360/91 and /95 equipment. (Initial development runs were made on a UNIVAC 1108.) The complete program took about 23 hours of CPU time accumulated over 44 separate runs. A maximum of 1,234,040 bytes of core storage were required for any one run. (See 5.0 for complete results.)

The number of required wire-miles configured were 9,345,396, and cost of the configuration was $3,528,318 per month rental charge. This yields a cost per required wire mile of $0.3775. The miles travelled by the circuits in the configuration were 11,314,383 which yields a cost per travelled wire-mile of $0.3118. The overall detour ratio (the travelled wire-miles divided by the required wire-miles) was 1.211. This means that the average circuit travelled a 21 percent greater distance than the direct point-to-point distance.

Although the cost per <u>required</u> wire-mile is clearly the best indicator of the effectiveness of the algorithm, the data now collected under the present manual system does not permit that number to be calculated for the manually-obtained results. The only computed data available from the present manual system that is approximately equivalent are that 11,965,323 <u>travelled</u> circuit miles were configured at cost per travelled wire-mile of $0.3153. This number is approximately the same as that achieved by the computed configuration.

The cost of the approximately equivalent manual configuration was $3,772,701.25 or somewhat more than the computed configuration. However, the manual configuration included a small number of multi-point circuits (circuits with more than two terminations) which were not included in the computed configuration.

In addition, the total manual configuration includes an extra 7,308,729 travelled miles of GSA circuits not configured in the computer problem (and not included in the cost above). The total configuration problem thus is about 60 percent larger than actually configured by computer. It may be that the least economical circuits configured by the computer program might have had the benefit of the use of bulk trunks had the GSA portion of the network been included.

In summary, an heuristic optimization program has been successfully run for a network of an extremely large size. A program for this size network has rarely if ever been attempted before. The inclusion of this program in the operational activities of DECCO is possible, but a systems analysis of DECCO data flow and operations must be accomplished, and a step-by-step conversion procedure designed, before installation can be achieved.

2.0 THE KEY TABLES

In carrying out the solution method outlined in Section 1.3 and generating the outputs of Section 1.4, four very important tables of information are initiated and manipulated. The City and Request Tables contain the bulk of the input data. During the course of the program the Request, Link, and Route Tables collect the current network solution information. The final values in these last three tables contain the optimal route and link information. The final network cost is always calculated from the final Link Table circuit fills and the rate structure. (See Table 1.) These tables will now be described.

2.1 The City Table (LOCTAB)

The given information for each city (node) consists of a four letter identification code (devised and used by DECCO) and a pair of integer geographic coordinates expressed in miles. Each city record in the table is five words long and contains:

1. City A - code name (given)
2. H_A - horizontal coordinates in miles (given)
3. V_A - vertical coordinates in miles (given)
4. Working word used during the program
5. " " " " " "

2.2 The Request Table (REQLST or WREQ)

The given information for each request consists of the coded names for the two terminal cities and the number of wires* (sub-voice grade circuits) required. Each request record is eight words long and contains:

1. Terminal City A - code name (given)

2. Terminal City B - code name (given)

3. Number of wires* required (given)

4. D_{AB} - Airline distance between Cities A and B, calculated by the Pythagorean Theorem and rounded up twice according to the procedure spelled out in the F.C.C. tariff regulations.

5. Detour Ratio (Calculated)

6. Working word used during the program

7. Number of links in the request route (from program)

8. Pointer to first word of request route in Route Table (from program)

Note that REQLST is an integer array equivalenced to the real array WREQ so that the proper mode (integer or real) could be used at various points in the FORTRAN program. Also note that words 7 and 8 of a record contain the final output information necessary for reading the Route Table (See Figure 1 and Section 2.4).

2.3 The Link Table (LNKTAB)

The Link Table is one of the two key output tables of the problem. It always contains the current information on the links chosen for the network routing configuration. The record for each link contains the following seven words:

1. End City A - code name

2. End City B - code name

3. Number of wires in use (fill)

4. Airline distance between end cities A and B

---

*In the final version of the program there are 12 wires per voice-grade circuit. This constant is stored in NWIRE.

5. Temporary added fill (for rerouting)

6. Working word used during program

7. " " " " "

The Link Table is alphabetized on Cities A and B for purposes of program searches of this table.

2.4 The Route Table (ROUTAB)

The Route Table is the second of the two key outputs of the problem. It consists entirely of pointers to the links in the Link Table that are in each route of a request. This indirect addressing scheme uses word 7 in the Request Table for the length of the request route (in links) and word 8 of that table for a pointer to the first word of that route in the Route Table (See Figure 1).

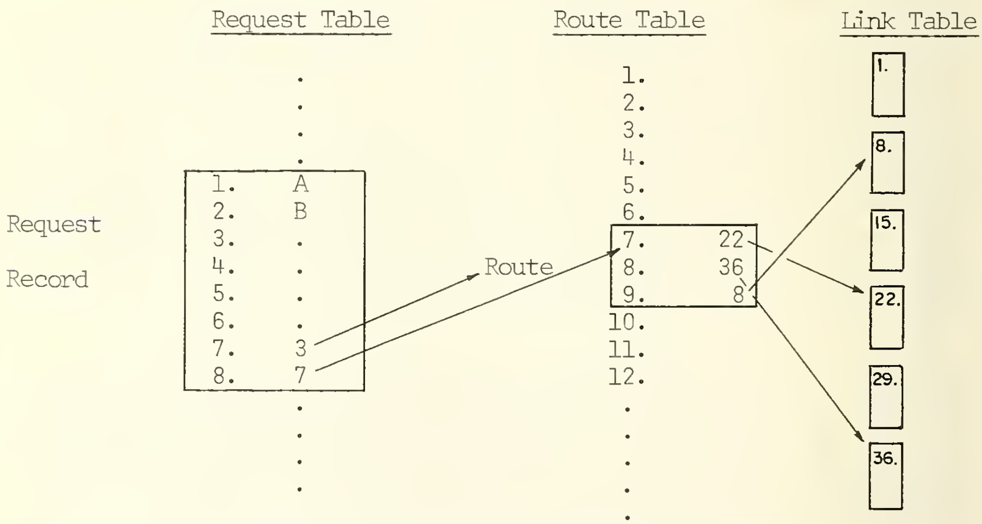


Figure 1 Indirect Addressing Scheme for Route of a Request

3.0 THE ALGORITHMS

The solution to this problem involves the sequential running of six programs. The first one (PREPRO) is a program that selects from the 1633 cities and 5552 requests a sub-problem of 250 cities. The five remaining programs (PASS1, ADDREQ, BYPASS, PASS2, and PASS3) are linked by a simple MAIN program. These five programs call upon an additional twenty-one subroutines to perform all the computations needed. A run-time generated restart capability was added to accommodate the long running time needed (22.8 hours of running time for the entire problem on the 360/91,95). Whenever a restart was implemented, magnetic tapes were used to store the current key tables.

3.1 Choosing a Subproblem (PREPRO)

The programs PASS1, BYPASS, PASS2, and PASS3 were developed and tested on subproblems of the large problem. In order to form meaningful subproblems, the 1633 cities were ordered according to their required number of circuit-miles. This number was determined for each city (node) by calculating the circuit-miles (number of circuits x airline distance) for each request and then summing the required circuit-miles for all requests terminating at that city. Then for a problem with some number of chosen cities, only those requests were picked that had both terminal cities chosen. Table 2 shows the percent of total circuit-miles considered in each subproblem. PREPRO accomplished Step 1 in the solution method of Section 1.3.

As a measure of the fraction of the network contained in each subproblem, the following quantity was calculated:

$$\text{\% Circuit-Miles} = \frac{\text{\# of Circuit-miles of Included Requests}}{\text{\# of Circuit-miles of All Requests}} \times 100$$

It includes the circuit-miles of only those requests which can be routed entirely within the subproblem network. It was decided to use a network of 250 cities as a subproblem to start with in the final solution.

Table 2 Subproblem Selection by Circuit-Miles

No. of Cities	No. of Requests	% Circuit-Miles
50	390	45
100	958	66
150	1520	76
200	2068	83
250	2394	87
1633	5552	100

3.2 The Linking Program (MAIN)

The MAIN program initializes the City and Request Tables by reading in the given city and request data. It also reads in four variable constants required by the programs. These constants are: XMDR, NOKC, XSIZE, and NWIRE. Their functions will be described later. The rest of MAIN is used to sequentially invoke the programs PASS1, BYPASS, PASS2, and PASS3.

3.3 First Network and First Optimization (PASS1)

PASS1 contains three distinct segments to accomplish Steps 2 and 3 in the solution method of Section 1.3. These segments are:

1. Constructions of the shortest spanning tree.

2. Routing of requests over the spanning tree.

3. Addition of links by decreasing detour ratios of requests.

3.3.1 Construction of the Shortest Spanning Tree: The initial network is established by linking the 250 city subproblem by a shortest distance spanning tree* formed from the complete graph. The method used is essentially the one outlined in reference [1], Construction B, and uses the property that every node is connected to its closest neighbor node. One starts the construction by marking an arbitrary city (the first city in the City Table) 'in' the network and the rest of the cities 'out' of the network. At each iteration the closest 'out' city to the set of 'in' cities is found and is connected to its closest 'in' city. This newly established link is then sorted into the Link Table alphabetically and the 'out' city marked 'in'. This procedure terminates when all the cities have achieved an 'in' status. Searching through the 'in' cities was minimized by a judicious use of the minimum distance values found in each iteration. See Figure 2 for a sample construction.

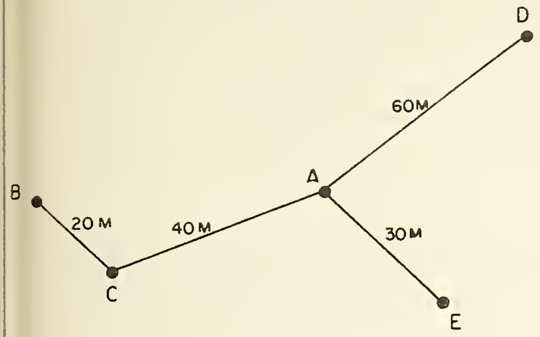
3.3.2 Routing Requests Over the Spanning Tree: This part of PASS1 uses the shortest spanning tree properties that (a) there is only one path between any pair of cities and that (b) every node is reachable from every other node. (See Appendix A) Requests are routed over the entire tree by repeated tracings of chains** in the tree, each such tracing starting from a different root city. A sufficient number of chains are considered to have been traced when all requests have been given paths through the tree. The particular root city chosen for each tracing is always the one with the largest number of unrouted requests for which this city is an end point. This choice maximizes the possible number of requests that can be routed via each tracing.

Starting at a root city, the program finds all chains in the spanning tree originating from that root city. This procedure finds the only route through the spanning tree between any two cities on a chain.

---

*See Appendix A for definition and properties.

**The word 'chain' is used interchangeably with 'path' throughout this paper. A chain may include a route or be included by a route.



Iteration	"in" Cities	"out" Cities	Added Link
0	A	B,C,D,E	
1	A,E	B,C,D	AE
2	A,E,C	B,D	AC
3	A,E,C,B	D	BC
4	A,E,C, B,D		AD

Figure 2 Shortest Spanning Tree Construction

Table 3 Tracings of Shortest Spanning Tree of Figure 2

Requests	Root City	Tracing	Requests Routed
1. A to B	A	Chain 1. AC, CB	A tc B, A to C, B to C
2. A to C		Chain 2. AD	none
3. B to C		Chain 3. AE	none
4. D to E	D	Chain 4. DA, AC, CB	none
		Chain 5. DA, AE	D to E

If any route so found is associated with a real request for service as defined by its terminal points, then the initial routing of the request has been determined. The program goes to a different root city when all chains from that root city have been traced. (See Table 3 for the tracings of Figure 2's shortest spanning tree.) Note that not only is the Route Table initialized here, but the initial detour ratio for each request and the initial accumulated circuit fill of each link is computed and stored in the proper key tables.

3.3.3 Decreasing Detour Ratios of Requests: Before actually proceeding with this phase of the program, the COST subroutine is called in to establish the initial cost of this initial network. This is done by breaking down the link fill values (found in the Link Table) into Telpak units (Ixc, C, D) that minimize the cost per link. This calculation uses the Telpak rate structure (See Table 1). The sum of these link costs is then considered the initial cost of the network and is the quantity to be reduced by any optimization procedure.

The initial configuration of the network is extremely inefficient. Many requests have very high detour ratios, meaning that they travel an extremely circuitous route from one end point to the other. The purpose of this first cost reduction program is to reduce the detour ratio by adding links that shorten the mileage travelled by requests. In order to investigate only that part of the network in the vicinity of the request being shortened (the primary request) (See 3.3.3.1) an ellipse is employed whose foci coincide with the end points of the primary request. Only those nodes of the network that fall on or within the ellipse are considered for this subproblem. Initially, the ellipse* is established with a relatively wide minor axis. If this results in too many cities within the ellipse than can be easily handled by the subproblem, the minor axis is reduced incrementally until an acceptable number of cities is obtained. The acceptable number of cities is established by NOKC, which is set at 20. In addition, the ellipse must be small enough so that at least one city of the route of the primary request falls outside the ellipse. This fact is necessary if the link-addition program is to work properly.

3.3.3.1 Choosing Candidate Links. The primary request to be re-routed is always one that has not yet been a candidate for rerouting, and which has the largest detour ratio larger than XMDR = 1.15. This value of XMDR was found by trial and error on the basis of network cost saved versus computer time used.

For the cities contained within the ellipse initially, a

*The size of the ellipse used can be identified by its own detour ratio. For an ellipse, the detour ratio is defined as the sum of the distances from the foci to any point on the ellipse divided by the distance between the foci. For any ellipse, the sum of distances to any point on the ellipse from the foci is constant regardless of the point chosen.

completely connected network is formed. This means that fictitious links are added temporarily to those already connecting the cities within the ellipse. A completely-connected network of 20 cities contains 190 links. A larger number of cities would begin to produce an unacceptably large number of links. Real links in this network are, for the purposes of this subproblem, given a distance value of zero. Fictitious links are assigned their real mileage. Then, using the algorithm of Dantzig [2] for finding the shortest route through a network, the subroutine MINPA finds the shortest path in the subnetwork between the terminals of the primary request. The method consists of tracing multiple paths from the starting terminal city of the request and extending these paths at each step with a link or links that result in a minimum cumulative length path (or paths). The first path to reach the second terminal city of the request is the minimum length route sought.

The device of assigning real links a length zero provides the answer in terms of the minimum extra real mileage that re-connects the terminals of the primary request. This procedure takes advantage of any links in the route of the primary request that fall within the ellipse, as they have zero length for this problem. Thus, the links in the minimum length path need not include the direct connection between the foci cities. (See Figure 3a and 3b for illustration). The subnetwork's fictitious links that lie on the new shortest path for the primary request are then the candidate links being sought.

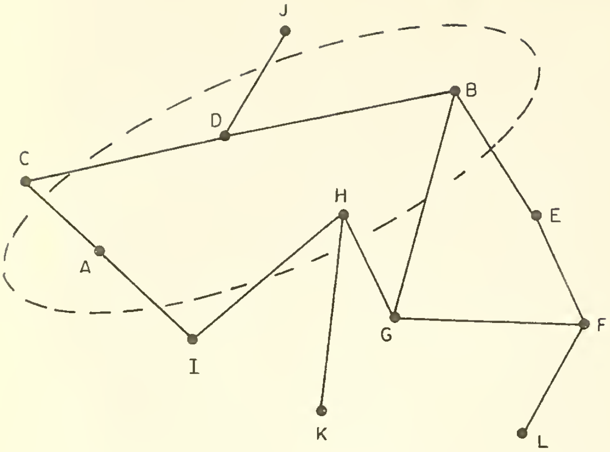
3.3.3.2 Testing Candidate Links. The shortest path between the terminal nodes in the primary request establishes the fictitious links on this path as candidate links for adding to the network. The next step is to re-route the primary request and other secondary requests passing through the ellipse, through the candidate links to determine if a reduced network cost will result. The re-routing results, in most cases, in reduced detour ratios and therefore in reduced cost. The basic subnetwork used for this test is the same as the subnetwork of 3.3.3.1 with the following two changes:

1. All fictitious links that are not candidate links are removed.

2. All primary request links that are not in the above ellipse are added.

All requests having at least two links in this basic subnetwork are called secondary requests and are candidates for rerouting and reduction of their detour ratios. These requests include the primary request. Since true distances are now involved, all link lengths are restored to their true values.

For each secondary request a true shortest route is now sought via subroutine MINPA. The subnetwork used consists of the basic subnetwork plus any of the secondary request route links not in the basic subnetwork. If this shortest route reduces the detour ratio of



Network A,B,C,D,E,F,G,H,I,
J,K,L

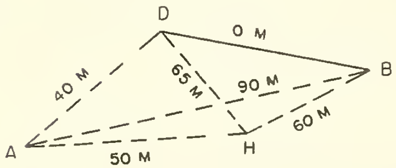
Primary Request (A,B)

Route [A,B] = AC,CD,DB

Ellipse has foci A,B

and eccentricity $e > \frac{1}{1.2}$

Figure 3a Network with Ellipse about nodes A,B



Real links: DB

Fictitious links: AD,AB,AH,DH,BH

Shortest route for [A,B] : AD,DB

Figure 3b Subnetwork for Finding Shortest Route for Primary Request (A,B) that secondary request, the new route is temporarily accepted (see Figure 4a and 4b for illustration). After all secondary requests are tested (including the primary request), and the appropriate new routes temporarily accepted, the final new network cost is compared to the old one. If a net saving is achieved, the candidate links and their accompanying secondary request route changes are accepted permanently.

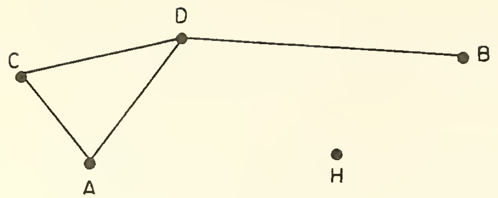
After all requests with detour ratio greater than 1.15 (see 3.2.1) are put through this two-part process, PASS1 ends. Note that no primary request is tried a second time after having once failed to produce a saving, even though the network changes after each successful test of candidate links. The reason for this decision was to shorten the execution of this time-consuming part of the program. In most cases, each successful test of candidate links adds just one new link to the network.

3.4 Configuring the Network for the Remaining Requests (ADDREQ)

Immediately after PASS1, the remaining 1383 cities and 3158 requests are read into the memory and configured into the partially optimized 250 city network, one request at a time. These requests, which have at least one city not in the network, are taken in descending order of their circuit-miles (see 3.1). Every such request is routed via a Minimum Cost Route Algorithm which, for each request, adds at least one new city and one new link with a minimal increase in the cost of the network. If the resulting request route ends in a dangling link (a link ending in a city with no other links connected at that city), the route is further optimized by a Deflection Link Algorithm that eliminates the dangling link while reducing the network cost. It was decided to use ADDREQ after PASS1 because the Route Table, which is the largest of the key tables, first reaches small enough size at this point.

3.4.1 <u>Minimum Cost Route Algorithm</u>: For each request to be added to the network, an appropriate subnetwork is established. Each of the two terminal cities of the request is linked to its five closest cities in the current network. These cities (all real) and links (both fictitious and real) form part of the subnetwork. Next, the direct link between the request terminal cities is added to the subnetwork. Then an ellipse, with a detour ratio XSIZE<1.25, is drawn using the two terminal cities of the request as foci. (See Page 12) Requests lying in the sparser northwest area of the network use a wider ellipse with XSIZE<1.6. (These values were obtained by experimentation.) All additional cities in the network that are in or on this ellipse are added to the subnetwork. Finally, all real links of the network that contain at least one city in this subnetwork are added to the subnetwork. (See Figure 5a and 5b for illustration.)

Using cost rather than length as the pertinent link characteristic, the program now finds a minimum <u>cost</u> path for the request in the above subnetwork. This minimum path algorithm is again based on the



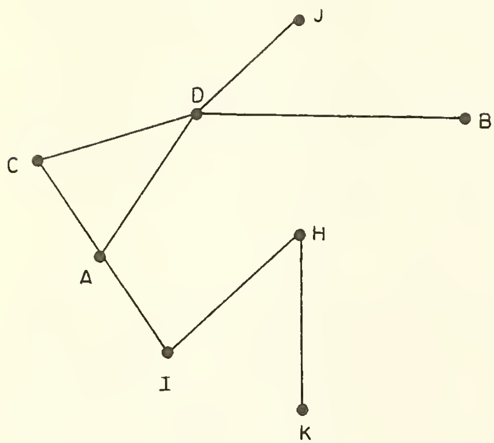
Primary Request Links:

AC, CD, DB

Shortest Route Link:

AD

Figure 4a Basic Subnetwork Used for Rerouting Secondary Requests



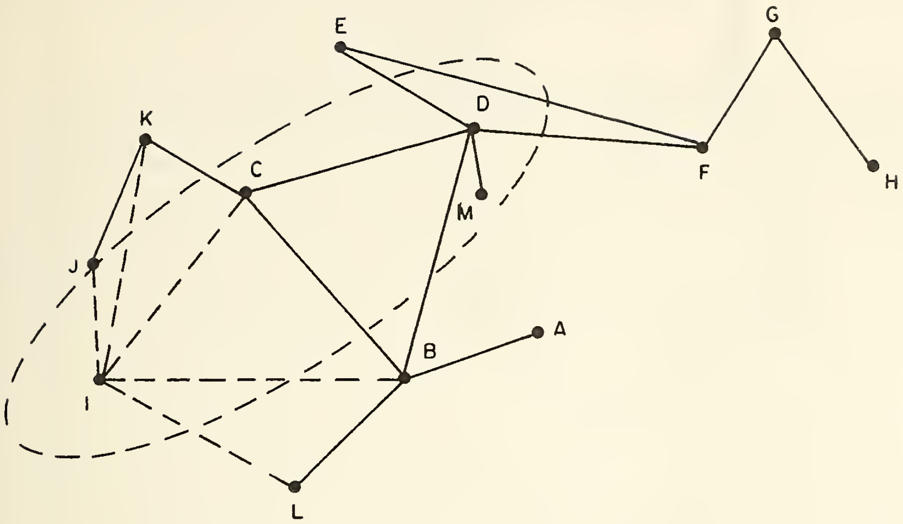
Original Route of (J,K):

[JD,DC,CA,AI,IH,HK]

New Route of (J,K):

[JD,DA,AI,IH,HK]

Figure 4b Subnetwork Used for Rerouting Secondary Request (J,K)



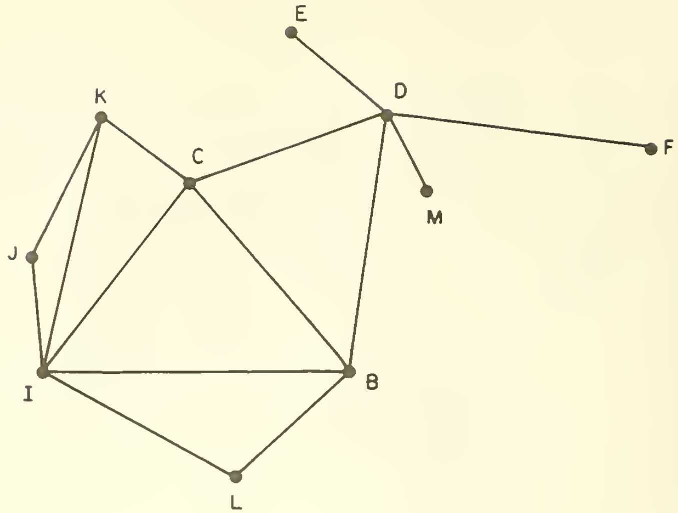
'OUT' city = I

'OUT' Request = R_{ID}

Fictitious Links = L_{IJ}, L_{IK}, L_{IC}, L_{IB}, L_{IL}

Ellipse Foci = I and D

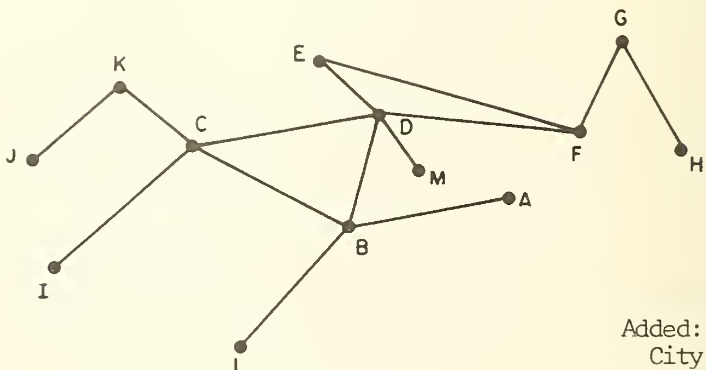
Figure 5a Network with Fictitious Links, 'Out' City I, and Subnetwork Ellipse for Minimum Cost Route Algorithm



Minimum Cost
Route of $R_{ID} = RT_{ID} = [L_{IC}, I_{CD}]$

Figure 5b Subnetwork Used For Finding a Minimum

Cost Route for Request from I to D



Added:
City I
Link IC
Route [IC, CD]

Figure 5c Reconfigured Network with Request I to D

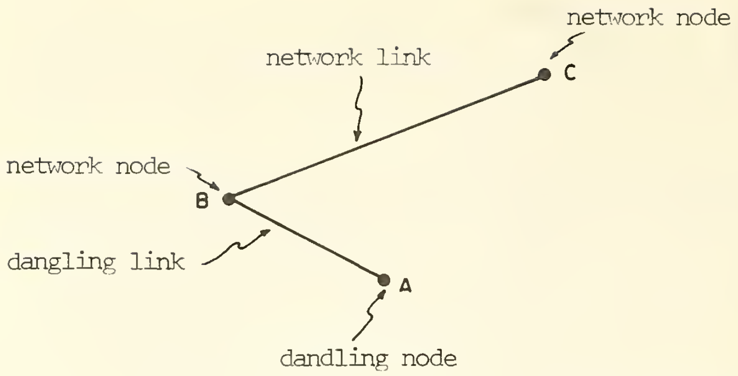
Dantzig algorithm [2] used in PASS1 (see 3.3.3.1) and is contained in subroutine MINPA1. After finding the minimum cost route for the new request, the four key tables are appropriately updated with the new city, request, link, and route data. (See Figure 5c for illustration.) The total network cost is also updated.

3.4.2 <u>Deflection Link Algorithm</u>: If the minimum cost route subroutine makes either terminal city of the new request into a dangling (terminal) node of the network, a second optimization procedure occurs before finalizing the route of the request. The subnetwork used for this algorithm always consists of the dangling node A, the corresponding dangling link AB, the network interior node B, and a network link BC out of B. (See Figure 6a.) An attempt is then made to convert the dangling node into an interior node of the network by deflecting the traffic in BC into a new link AC via a route consisting of BA, AC. (See Figure 6b.) The deflection link AC and the corresponding route changes which produced the largest cost saving are then accepted into the network, as well as the final route of the new request (which may or may not use that deflection link AC). If no cost saving can be achieved by this process, the original minimum cost route of the new request is accepted and, in addition, the dangling node is connected to its two closest network nodes by links with zero fill. This last step allows future network changes to possibly make the dangling node into an interior node and thus achieve more economic bundling of circuits. (See Appendix B for description of network changes accompanying the application of this algorithm.)

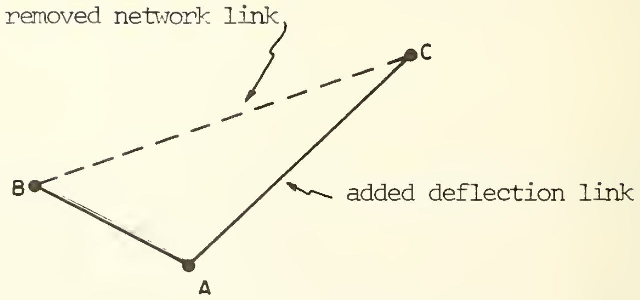
3.5 "Express" Links (BYPASS)

Further reduction in the cost of the network is attained by substituting "express" links (direct links) for chains of links containing the same number of D Telpak bundles. The cost reduction occurs because the length of the express link is less than the length of the sequence of links that it replaces. The object of BYPASS is to find a candidate chain, to identify the routes that use the chain, and to test for a cost reduction when bypassing the chain's traffic through a direct link. Every successful bypass either introduces a new link with an appropriate number of D Telpaks or adds that number of D Telpaks to an existing express link.

3.5.1 <u>Finding the First Candidate Chain</u>: The program starts by determining the number L, which equals the largest number of D Telpaks in any one link of the entire network. If at least one C Telpak is also present in any link having L 'D Telpaks' this value of L is increased by 1. The route of each request is then examined to find the longest sequence of links with fills $\geq$ 240L. Any request that has only one link in its route is automatically eliminated from the search. The longest sequence of links with fills $\geq$ 240L is the candidate chain being sought. A table of chains that failed to produce a cost saving is developed in FCHAIN and checked against a current chain being tested (via function



(a) Before Deflection



(b) After Deflection

Figure 6 Adding a Deflection Link

FTEST) so that no chain is found and run through this program more than once.

3.5.2 Finding Routes that Contain a Candidate Chain: A search is then made for all the request routes that completely contain the candidate chain (via subroutine CNTAIN). Each time such a request is found, it is placed with its request fill, in a table FILTAB. This table is arranged in descending order of fill requirements (in wires). The required fills in this table are also accumulated and stored in a word SUMC.

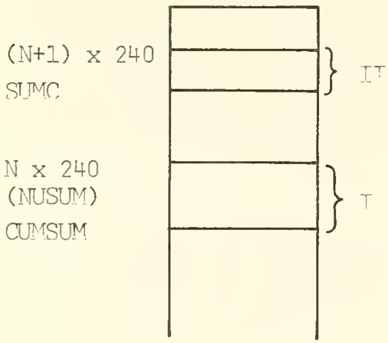
3.5.3 Determining the Bypass Fill and Its Corresponding Requests: If SUMC contains an integer multiple of 240 circuits, the bypass link is given a fill of SUMC and all requests in FILTAB are rerouted. If SUMC does not contain an integer multiple of 240 circuits, a choice must be made between the two integral multiples of 240 that bound SUMC. The word NUSUM is set to 240 times the lower bound.

An additional requirement that must be satisfied in this process is that all circuits of a request must be bypassed completely or not bypassed at all. Requests cannot be split. The word CUMSUM accumulates the fill requirements of requests in FILTAB in descending order of magnitude until the largest CUMSUM is obtained such that CUMSUM $\leq$ NUSUM. Then CUMSUM and SUMC are examined to find the one closest to an integer multiple of 240 circuits. If CUMSUM is the closer one, the program attempts to bypass NUSUM of the D Telpaks and alters routes of those requests in FILTAB that contributed to the value of CUMSUM. If SUMC is the closer one, the program tries to bypass NUSUM+1 of the D Telpaks and reroutes all requests in FILTAB. (See Figure 7)

3.5.4 Testing for Cost Savings: Using the end points of the candidate chain as the terminals of the bypass link and the fill just determined in 3.5.3, the cost of the network with the changed routes of requests is compared to the old network cost. If a saving is achieved, the bypass link is accepted and the route changes are made permanent. If no saving is achieved in bypassing the candidate chain, a subset of that chain is chosen as a possible candidate chain and the procedure just outlined is repeated.

3.5.5 Choosing a Chain Subset: The scheme for choosing subsets of a candidate chain consists of removing links from the ends of a chain in all possible ways so that the chain length is sequentially diminished by one each time. (See Figure 8 for details.) When a subset of a primary chain is bypassed, the largest of the unused chains on either side of the subset is selected for the next trial chain and is treated like the first candidate chain.

3.5.6 Finding Other Candidate Chains: After the first candidate chain has been processed by the program, all other candidate chains with fill $\geq$ 240L are sought for and checked out. When no more chains are found for this value of L, L is reduced by 1 and the whole procedure



I < II → Bypass N 'D Telpaks'

II < I → Bypass (N+1) 'D Telpaks'

Figure 7 Bypass Fill Selection

Iteration	Length of Chain	Beginning Link	Ending Link
1	N	1	N
2	N-1	1	N-1
3	N-1	2	N
4	N-2	1	N-2
5	N-2	2	N-1
6	N-2	3	N
7	N-3	1	N-3
8	N-3	2	N-2
9	N-3	3	N-1
10	N-3	4	N
.	.	.	.
.	.	.	.
.	.	.	.
$\frac{(N-1)(N)}{2}$	2	N-1	N

Figure 8 Scheme for Choosing Chain Segments

is repeated. The BYPASS program terminates when L = 1.

3.6 Rerouting Uneconomical Link Fills (PASS2 and PASS3)

The object of PASS2 and PASS3 is to reroute some link fills so that the maximum number of network links take advantage of the Telpak bulk savings. In order to do this it is necessary to determine which links have an uneconomical "overflow"* of circuits, to find and reroute the requests that create the "overflow", and to test the rerouting for cost savings. The rerouting is done within a subnetwork via the minimum cost route subroutine (MINPA1). The major difference between PASS2 and PASS3 is that PASS2 considers an overflow fill less than 30 circuits (one half of a C trunk) to be uneconomical while PASS3 uses a value less than 92 circuits (about one and one-half C trunks). PASS2 attempts to eliminate Ixc's and uneconomical (half-filled) C trunks. PASS3 attempts to eliminate full C trunks. Both attempt to fill more D trunks. A minor difference between the two passes lies in the searching of the requests for choosing those that cause the overflow.

3.6.1 <u>Choosing the Uneconomical Link and Its Overflow Fill Value</u>: The overflow value F of each link is computed (via subroutine COST1) and stored in the last word of each link record. In PASS2 the F value for a link is the number of circuits remaining after removing all full D and C Telpaks ($0 < F < 30$). ** PASS3 uses an F value obtained by removing all full D trunks and all but the last full C trunk ($0 < F < 92$). The untried link that has the largest F value is then tagged the uneconomical one and its corresponding F value becomes the maximum number of circuits to be rerouted (FMAX).

3.6.2 <u>Choosing the Rerouting Subnetwork</u>: Two intersecting circles are drawn whose centers are the uneconomical link's end nodes and whose radii are the link's length (in miles) plus 500. The subnetwork then consists of all nodes in or on these two circles and all links which have at least one node in this node set. The current uneconomical link is deliberately deleted from the subnetwork so that it will not be used in the rerouting. If the subnetwork's allowable node table (OKCTYS) or link table (OKLNKS) is exceeded (250 nodes, 2,600 links), the circle radii are reduced by 20 miles and the procedure repeated until a satisfactory size network is achieved (see Figure 9).

3.6.3 <u>Choosing Requests for Rerouting in PASS2</u>: In rerouting requests that contribute to the overflow fill FMAX of the uneconomical link, the largest number of circuits that is removed is FMAX+4. Therefore all requests containing the uneconomical link and requiring

---

*An 'overflow' fill is the number of circuits in a link, modulo 240.

**Note that COST1 assigns another C Telpak to any link having an overflow greater than 30 circuits.

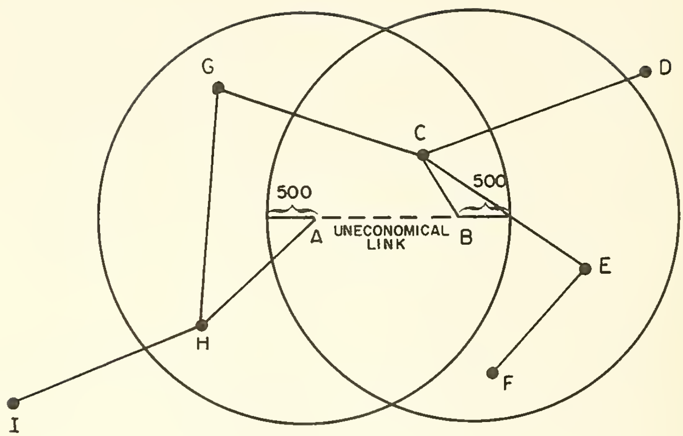


Figure 9 Subnetwork for Rerouting Unecomical Links

≤FMAX+4 are stored in a try-table in descending order of number of circuits. The program attempts to reroute a single request from the try-table with an overflow fill, FTRY, ranging from FMAX to FMAX+4. Once an FTRY value fails to achieve a saving, no other request with that FTRY value is tested. If none of these reroutings produces a saving, a combination of requests in the try-table, whose requirements have a combined fill <FMAX, is tested. (See Appendix C for scheme.) If a request or a combination of requests produces a saving, the new route or routes are accepted.

3.6.4 Testing for Cost Savings in PASS2: To determine if a saving is achieved, the cost of the network with the new route of a request is compared to its cost with the present route. This is done by comparing the cost of adding the request with the new route (in CUMCST) to the cost of adding the request with the present route (in RESAVE). Whenever CUMCST is smaller than RESAVE, the new route of the request is producing a saving, and is thus accepted permanently.

3.6.5 Choosing Requests for Rerouting and Testing for Cost Savings in PASS3: In this last program of the Telpak problem, all requests whose routes contain the uneconomical link and whose requirement is less than FMAX are stored in a try-table in descending order of fill requirements. A set of requests is now sought in the try-table such that the sum of their requirements is minimally greater than FMAX. If no such set exists, a single request minimally greater than FMAX is used. This request set or the single request is then rerouted and tested for cost savings just as in PASS2. (See 3.6.4.)

3.6.6 Terminating PASS2 and PASS3: Every time a rerouting is permanently accepted, new overflow values, F, are calculated for the network links. If the new F value of a link is greater than its old value, and that link has already been tried, it is allowed to be tried again by marking its record 'untried'. The program terminates when no more untried links have overflow fill values greater than zero.

4.0 THE COMPUTER RUNS

In order to understand the runs that were made for this problem, it is necessary to discuss the computer programs and subprograms, the organization of the runs, the operating environments and the Input/Output used. Whenever possible the pertinent information has been placed in tables to summarize this information.

4.1 Programs, Subroutines, and Storage Requirements

The computer programs embodying the algorithms of Section 3 are all written in Fortran IV for ease of transfer from one machine to another. The program PREPRO (see 3.1) ordered the 1633 cities and 5552 requests into nested subproblems of 50, 150, 200, and 250 cities. The solution to the entire DECCO network configuration involves the six programs MAIN, PASS1, ADDREQ, BYPASS, PASS2, and PASS3 described in

Section 3, and twenty-one subprograms whose purpose will now be described very briefly. The number of statements and the storage requirements for these will be found in Table 4.1. The subprograms used in each major program segment are shown in Table 4.2, and the core storage requirements for each major program segment are shown in Table 4.3.

4.1.1 Subroutine CDISS: This routine calculates the distance, in miles, between two cities. It uses the city's coordinate values for the calculation.

4.1.2 Subroutine CLOCK: This routine is used at predetermined restart points in the program. It obtains the remaining CPU time allowed the program for the current run and decides whether to continue running or to dump the vital information on tape and stop.

4.1.3 Subroutine CNTAIN: This routine is called by BYPASS to examine a request route for containment of a candidate chain.

4.1.4 Function COST: This function finds the total cost of the network at any given time.

4.1.5 Function COST1: This function is similar to function COST but has some different options.

4.1.6 Subroutine DEFLCT: This routine is called by ADDREQ to attempt to convert an end city of a newly added request from a peripheral to an internal city of the network.

4.1.7 Function FTEST: This function examines a variable length table of failed chains for containment of a given chain. It is similar subroutine CNTAIN.

4.1.8 Subroutine GARBAG: This routine condenses the route table whenever the program finds itself at the end of that table. Discarded routes are marked by negative numbers and must be periodically cleared out.

4.1.9 Function IXC: This function computes the cost of one circuit, using the Ixc cost structure (see Table 1).

4.1.10 Function LKCOST: This function computes the cost of one link by using the Telpak rate structure of Table 1 and dividing the fill among D's, C's, and Ixc's so as to minimize the cost.

4.1.11 Subroutine MINLNK: This routine finds the closest 'out' city to a given 'in' city of a network. It is used in constructing the minimum spanning tree.

4.1.12 Subroutine MINPA: This routine finds the shortest path length in miles between any two cities in a given network.

Table 4.1 Size of the Programs and Subprograms

No.	Program or Subprogram	No. of Source Statements (Fortran IV)	Storage Required (Bytes)
1.	MAIN	103	2,368
2.	PASS1	732	30,168
3.	ADDREQ	421	8,172
4.	BYPASS	359	5,792
5.	PASS2	288	5,216
6.	PASS3	323	7,060
1.	CDISS	8	470
2.	CLOCK	6	288
3.	CNTAIN	30	684
4.	COST	39	912
5.	COST1	70	1,376
6.	DEFLCT	382	27,300
7.	FTEST	37	596
8.	GARBAG	57	1,162
9.	IXC	24	662
10.	LKCOST	43	1,014
11.	MINLNK	20	386
12.	MINPA	89	1,234
13.	MINPA1	101	1,164
14.	NULINK	60	980
15.	PROUT	103	3,728
16.	PROUT3	77	3,008
17.	SAVEIT	23	812
18.	TEMPLK	24	350
19.	TESTLK	46	816
20.	TRY	75	1,366
21.	TRY1	65	1,246
	Total	3,605	108,330

4.2 Subprograms Employed

Run Type	Programs Used*	Subprograms Used*	Storage (bytes)
PASS1	1, 2	1, 2, 4, 8, 9, 10, 11, 12, 14, 15, 17	44,184
ADDREQ	1, 3	1, 4, 6, 8, 9, 10 13, 14, 15, 16, 17 18, 19	52,918
BYPASS	1, 4	1, 2, 3, 4, 7, 8, 9, 10, 14, 16, 17	19,468
PASS2	1, 5	1, 2, 4, 5, 8, 9, 10, 13, 15, 17, 20	20,538
PASS3	1, 6	1, 2, 4, 5, 8, 9, 10, 13, 15, 17, 21	22,262

* The numbers refer to the programs and subprograms listed in Table 4.1.

Table 4.3 Maximum Core Storage Required (in bytes)

	PASS1	ADDREQ	BYPASS, PASS2 OR PASS3
Program	44,184	52,918	22,262
System Routines	21,078	21,078	21,078
Common Blocks	586,936	938,832	1,190,700
Total	652,198	1,012,828	1,234,040

4.1.13 Subroutine MINPA1: This routine finds the minimum cost path in dollars, between any two cities in a given network. It is similar to subroutine MINPA.

4.1.14 Subroutine NULINK: This routine finds the location of a new link in the alphabetized link table and, if desired, inserts the new link in the table at that position.

4.1.15 Subroutine PROUT: This routine is the general print-out routine for all the programs and contains numerous options for printing results of the computer runs.

4.1.16 Subroutine PROUT3: This routine is a print-out routine, especially designed for the ADDREQ program.

4.1.17 Subroutine SAVEIT: This routine dumps the contents of the common blocks on magnetic tape when a check on the computer clock shows that the allowable time for a run has almost expired.

4.1.18 Subroutine TEMPLK: This routine is called by ADDREQ to link the five closest cities to an end city of a new request.

4.1.19 Subroutine TESTLK: This routine is called by ADDREQ. It either computes the cost of a subnetwork link with permanent fill and places the fill requirement of the new request in a word of the link record, or, if the link is not in the permanent table sets its cost to zero and places the link record in the temporary block of the link table.

4.1.20 Subroutine TRY: This routine is used by PASS2 to find a cheaper route for a request that is presently routed via an uneconomical link.

4.1.21 Subroutine TRY1: This routine is used by PASS3 for a purpose similar to subroutine TRY.

4.2 Organization of the Runs

4.2.1 The Developmental Subproblems: The program was initially tested on small networks on the UNIVAC 1108 computer at the National Bureau of Standards before any production runs were attempted. Then, to eliminate further bugs and attain more confidence in the program's operation, the nested subproblems of the DECCO network were run on various IBM 360 machines (see Section 4.3). At this point all the programs and subprograms except ADDREQ and its required subprograms DEFLCT, PROUT3, TEMPLK, and TESTLK were developed and debugged. The smaller developmental subproblems (50 and 150 cities) were completed in one run and used the entire program as developed to this point. The larger developmental subproblems (200 and 250 cities) used the entire program to this point but had to incorporate a restart feature in MAIN for the completion of PASS1.

Table 5 shows various parameters associated with developmental subproblem computer runs for 50, 150, 200, and 250 city problems. As can be seen, a complete 250 city problem run took about 82 minutes of computer time to complete. The 200 city problem was done with three restarts of PASS1 while the 250 city problem used four restarts. A restart in PASS1 always began at the point at which a new primary request was selected for re-routing.

4.2.2 Full Network: For the configuration of the entire DECCO network, the result of the 250 city problem after the completion of PASS1 was employed as a point of departure. The application of the program ADDREQ and its necessary subprograms were completed with seventeen restarts. (The number of restarts was determined by the availability of the computer.) An iteration here consists of the configuring of a new request into the network. When a restart occurs, it is caused to occur at the beginning of an iteration. BYPASS and its subprograms required one run. An iteration here is defined as the selection of a candidate chain and testing its segments for a bypass link substitution. A restart capability was programmed but was not called in by this execution. The application of PASS2 and its subprograms was accomplished with fourteen restarts while PASS3 used four restarts. PASS2 and PASS3 define an iteration as finding a link with overflow circuits and attempting to redistribute the overflow more economically. Note that the MAIN program contains all the restart logic and it is the restart verstion of MAIN that is included in the program size data of Table 4. Table 6 shows the computer time required to complete the full 1633 city configuration and optimization problem. Individual run times are given.

4.3 The Operating Environment

4.3.1 The Machines Used: The large core and time requirements of the DECCO network and its larger subproblems required the use of a large machine. Three different IBM 360's were used, depending on the availability of the machines. Table 7 lists the storage capabilities of the computers used while Table 5 includes a reference to the machine used for each run.

Table 5 Developmental Subproblem Computer Runs

Size of Network Cities	Requests	Machine Used°	Minimum Detour Ratio (XMDR)	Run No.	Program		C.P.U. Time (min.)
50	390	2	1.00	1	PASS1	(Spanning Tree (plus Routing	.07
						(Decr. Detour (Ratio	1.53
					BYPASS		0.05
					PASS2		0.03
					PASS3		0.03
						Total	1.71
150	1520	2	1.20	1	PASS1	(Spanning Tree (plus Routing	0.53
						(Decr. Detour (Ratio	7.80
					BYPASS		0.38
					PASS2		0.25
					PASS3		2.90
						Total	11.86
200	2028	2	1.15	1	PASS1	(Spanning Tree (plus Routing	2.62
						(Decr. Detour (Ratio	9.72
				2		"	11.52
				3		"	12.23
				4		"	10.00
				5	BYPASS		1.18
					PASS2		1.40
					PASS3		3.40
						Total	52.07
250	2394	2	1.10	1	PASS1	(Spanning Tree (plus Routing	3.95
						(Decr. Detour (Ratio	22.67
				2		"	12.25
				3		"	7.10
				4		"	11.38
				5		"	12.28
				6	BYPASS		1.75*
				7	PASS2		2.37
				8	PASS3		8.03*
						Total	81.78

° See Table 7 for code numbers for the machines used.
* Charged time. CPU time unavailable.

Table 6 Full Network Computer Runs

Size of Network			Machine Used°	Run. No.	Program	C.P.U. Time (min)
Cities	Added Requests	Total Requests				
250	2394	2394	2	1-5	(See Table 5 for breakdown)	Total 69.63
250+	162	2556	3	1	ADDREQ	7.51
	157	2713	"	2	"	7.50
	134	2847	"	3	"	7.51
	100	2947	4	4	"	4.35
	199	3146	"	5	"	6.47
	200	3346	"	6	"	7.12
	200	3546	"	7	"	7.08
	200	3746	"	8	"	7.29
	200	3946	"	9	"	7.14
	200	4146	"	10	"	7.60
	200	4346	"	11	"	7.97
	200	4546	"	12	"	8.69
	200	4746	2	13	"	9.30
	200	4946	"	14	"	9.70
	200	5146	"	15	"	9.70
	200	5346	"	16	"	10.10
	200	5546	4	17	"	8.54
	6	5552	"	18	"	3.55
						Total 137.12
1633		5552	"	19	BYPASS	8.06
			2	20	PASS2	13.1
			"	21	"	12.8
			"	22	"	12.8
			"	23	"	27.8
			"	24	"	27.8
			"	25	"	12.8
			"	26	"	43.0
			"	27	"	12.8
			"	28	"	12.7
			4	29	"	58.28
			"	30	"	58.28
			"	31	"	28.31
			"	32	"	118.50
			"	33	"	58.36
			"	34	"	6.90
			"	35	PASS3	13.28
			"	36	"	58.23
			"	37	"	178.62
			"	38	"	359.51
			2	39	"	39.84*
						Total 1161.77
						Grand Total 1368.52 or 22.81 hrs.

° See Table 7 for code numbers for machines used

* Charged Time. CPU time unavailable.

Table 7 Storage of the Machines Used

No.	Machine	Location	Available Core Memory
1.	UNIVAC 1108	NBS, Gaithersburg, Md.	50K words
2.	IBM 360/95	NASA, Greenbelt, Md.	4000K bytes
3.	IBM 360/91	NASA, Greenbelt, Md.	1500K bytes
4.	IBM 360/91	Applied Physics Lab., Columbia, Md.	1500K bytes

4.3.2 Execution Time and Storage: It should be noted that throughout this report the core storage and C.P.U. time values are approximate. This is due to differences between IBM 360 systems as well as variations in one system from run to run. The IBM 360/95 had variations in both time and storage which sometimes were and sometimes were not attributable to its multiprogramming environment. In this system the user has the option of calling for one of three FORTRAN compilers, each differing in the level of optimization and the consequent program storage allocation. This system also has two types of high speed memory, magnetic core and thin film core, which were assigned by the operating system. In one instance, for example, two identical runs had C.P.U. times of 2.37 minutes and 1.59 minutes, with the shorter time undoubtedly due to thin film storage allocation. It should be added, however, that thin film allocation was much less common.

There were also program factors contributing to the variation in time and storage needs. Changes, from problem to problem, in the input value of the minimum detour ratio (XMDR) were made in order to find an optimal choice for this constant. The value of XMDR determined the stopping point in PASS1 and thus influenced the running time. (The smaller XMDR, the longer the running time.) Also, the array that contributed most to the core requirements of the common block was the Route Table. A garbage collection routine was introduced to compact this table as necessary. The larger the table, however, the less garbage collecting was required and the faster the execution time.

The C.P.U. times for the runs that were made are included in Table 5 but, for the above reasons are considered only approximately comparable. The storage requirements for the different program combinations as well as the system and common block requirements are included in Table 4.2. Table 4.3 lists the maximum storage required for the three phases of the DECCO network solution. Clearly, a computer with about 1300K bytes available is necessary for solving the Telpak problem.

4.4 Input/Output

4.4.1 Input Data: The input data for a subproblem consists of the City Table (see 2.1), the Request Table (see 2.2), and the four variable constants XMDR (minimum detour ratio), NOKC (the maximum number of cities allowed in a subnetwork of PASS1), XSIZE (the ellipse detour ratio used to determine a subnetwork of PASS1), and NWIRE (the number of sub-voice telegraph/telephoto lines in a voice circuit). The input data is read in from cards, one card per city or request and one card for the constants. For the DECCO network the above input is used for PASS1. ADDREQ then reads the remaining requests from cards and adds the corresponding city or cities as each request is configured. After all requests have been configured, successive restarts of the program use the contents of the previous run's arrays as input (via magnetic tape).

4.4.2 Output of the Runs: As described in Section 1.4, the main outputs of the runs consist of the Link Table, the Route Table, and the final network cost (TOTCST). The following five quantities are also calculated and printed out. These are very important for the evaluation of the network cost and configuration (see Section 5). See Appendix D for sample outputs. See Table 8 for these values from the computer runs.

4.4.2.1 Required Wire-Miles (REQMI). This number is obtained from the sum over all requests of the wire-miles (number of wires required x airline distance) for each request. It is a measure of the total network communication requirement in wire-miles.

4.4.2.2 Total Path (TOTPTH). This number is the sum over all requests of the product for each request of the wire-miles with its detour ratio. It is a measure of the travelled wire-miles in the network.

4.4.2.3 Cost Per Required Wire-Mile (CPRWM). This number is the ratio of the total network cost (TOTCST) to the network's required wire-miles (REQMI). It is a measure of the average cost for each unit of communication requirement.

4.4.2.4 Cost Per Travelled Wire-Mile (CPTWM). This number is the ratio of the total network cost (TOTCST) to the total path travelled in the network (TOTPTH). It is a measure of the average cost for each unit of communication utilization.

4.4.2.5 Average Detour Ratio (ADR). This number is the ratio of the total path (TOTPTH) to the required wire-miles (REQMI). It is a weighted average of the detour ratios of all the network requests.

5.0 RESULTS

5.1 Costs

Table 8.1 includes a summary of the final cost results for the full 1633 node network and its four subproblems. Several differences are worth noting. The final problem introduces on the average, about one new city for every two requests added over the 250 city problem. The effect of these many extra cities is that unit costs go up. The cost per required wire-mile rises more than 12% over the result for the 250 city problem, although only 13% additional mileage has been configured. The average detour ratio also rises, as might be expected, but not as much as the unit costs.

5.2 Connectivity

As shown in Table 8.2, the number of links per route is much higher in the full network than in the smaller ones, and the connectivity is much lower. The connectivity is the ratio of the number of used links in the configuration to the number of links in the minimum-distance spanning tree. In any n-node network, a completely connected configuration has n(n-1)/2 links while the minimum spanning tree has (n-1) links. Thus the maximum value of the connectivity is n/2. From Table 8.2, it can be seen that all the networks are sparsely connected, and the full network is the least dense.

5.3 Computing Time

Table 9 summarizes the time taken to compute different subprograms for different size problems. It can be seen, that, in general, the dollar savings in the solution per minute of computing time decreases as the problem size grows. The programs PASS2 and PASS3 were relatively inefficient on the full size problem, although they still saved in monthly rentals considerably more than the cost of their computing time. However, it was not expected that PASS2 and PASS3 would take as long as they did. If that had been known, a better strategy would have been to re-run PASS1 on the largest problem following the running of ADDREQ. This conceivably would have provided the input to PASS2 with a more optimized configuration. Time and funding constraints prevented further experimentation.

Table 8.1 - Summary of Results

Network Cities	Network Requests	Total Cost (TOTCST)($)	Required Wire-Miles (REQMI)	Cost per Required Wire-Mile (CPRWM)($)	Travelled Wire-Miles (TOTPTM)	Cost per Travelled Wire-Mile (CPTWM)($)	Av. Detour Ratio (ADR)
50	390	1,366,150	4,208,093	.3246	4,851,984	.2816	1.153
150	1520	2,405,106	7,116,866	.3379	8,297,539	.2899	1.166
200	2028	2,662,264	7,724,914	.3395	9,257,560	.2833	1.198
250	2394	2,719,790	8,098,960	.3358	9,742,655	.2792	1.203
1633	5552	3,528,318	9,345,396	.3775	11,314,383	.3118	1.211

Table 8.2 - Summary of Network Parameters

Network Cities	Network Requests	Av.No.of Links/Route	Tot.No. of Links	Connec-tivity Number
50	390	4	83	1.69
150	1520	5	256	1.72
200	2028	6	328	1.65
250	2394	7	391	1.57
1633	5552	15	1990	1.22

Table 9 Computing Time Effectiveness

Network Size

		50 City	150 City	200 City	250 City	1633 City
PASS1	Compute Time (Min.)	1.53	7.80	43.47	65.68	65.68*
	Dollar Reduction	$ 436,099	$1,950,468	$2,009,453	$2,531,690	$2,531,690*
	Savings Per Minute	$ 285,032	$ 250,060	$ 46,226	$ 38,545	$ 38,545*
BYPASS	Compute Time (Min.)	0.05	0.38	1.18	1.75	8.06
	Dollar Reduction	$ 6,780	$ 18,733	$ 29,532	$ 39.866	$ 37.838
	Savings Per Minute	$ 135,600	$ 49,297	$ 25,027	$ 22,781	$ 4,695
PASS2	Compute Time (Min.)	0.03	0.25	1.40	2.37	504.30
	Dollar Reduction	$ 39,761	$ 102,045	$ 109,261	$ 94,112	$ 237,541
	Savings Per Minute	$1,325,367	$ 408,180	$ 78,044	$ 39,710	$ 471
PASS3	Compute Time (Min.)	0.03	2.90	3.40	8.03	649.48
	Dollar Reduction	$ 59,786	$ 12,754	$ 53,139	$ 67,961	$ 81,274
	Savings Per Minute	$1,992,867	$ 4,398	$ 15,629	$ 8,463	$ 125

* 250 City Configuration Employed

REFERENCES

[1] Kruskal, J. B., Jr., On the Shortest Spanning Subtree of a Graph and the Travelling Salesmen Problem, Proceedings of the American Mathematical Society, Vol. 7, No. 1, (Feb. 1956), pp. 48-50.

[2] Dantzig, G. B., On the Shortest Route Through a Network, Management Science, Vol. 6 (1960), pp. 187-190.

APPENDIX A

PROPERTIES OF A SHORTEST SPANNING TREE

The properties of a shortest spanning tree network for a finite connected graph with a positive number (length)* associated with each edge are summarized below:

1. Every node is connected to its closest node.

2. Every node is reachable from every other node.

3. There is a unique path between any two given nodes.

4. The sum of the lengths of the edges is a minimum.

5. The tree is unique if the lengths of the edges are distinct.**

6. The tree contains the minimum number of edges for connecting all nodes. If n is the number of nodes, the number of edges is (n-1).

7. The tree contains no loops.

---

* The positive number associated with an edge of a graph can be any characteristic possessed by all edges. In this problem, the cost of an edge is used in this way.

** Since the lengths of the edges in the DECCO network are not all distinct, the graph formed is not unique but is a shortest spanning tree.

APPENDIX B

NETWORK CHANGES PRODUCED BY DANGLING LINK ALGORITHM

Each of the links in a path of two links can be thought of as having a fill composed of two disjoint sets. These are:

1. the fill of those requests whose routes contain both links.

2. the fill of those requests whose routes contain one link or the other.

Using the subnetwork of Figure 6 one can translate this fill information into set theoretic relations.

If one denotes the fill of a link by FL, the fill of a set of requests by FR and then subscripts these quantities with letters denoting the link or route, one can say, before deflection:

$$FL_{AB} = FR_{ABC} + FR_{AB} \tag{B.1a}$$

$$FL_{BC} = FR_{ABC} + FR_{BC} \tag{B.1b}$$

The cost of link AB is then a known function of its fill (FL_{AB}) and its length (D_{AB}) and similarly for link BC. Thus the total cost of the two links is C, where

$$C = f(FL_{AB}, D_{AB}) + f(FL_{BC}, D_{BC}) \tag{B.2}$$

After deflection one has the relation:

$$FL'_{AB} = FR_{BC} + FR_{AB} \tag{B.3a}$$

$$FL'_{AC} = FR_{BC} + FR_{ABC} \tag{B.3b}$$

where the notation FL' denotes the link fill after deflection. Combining equations (B.1) and (B.3) one then finds

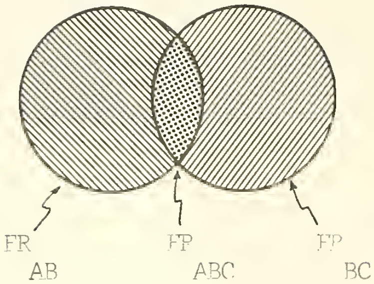
$$FL'_{AB} = FL_{AB} + FL_{BC} - 2FR_{ABC} \tag{B.4a}$$

$$FL'_{AC} = FL_{BC} \tag{B.4b}$$

Correspondingly the new cost, C', is a known function of the new fills, FL'_{AB} and FL'_{AC}, and the link lengths.

$$C' = f(FL'_{AB}, D_{AB}) + f(FL'_{AC}, D_{AC}) \qquad (B.5)$$

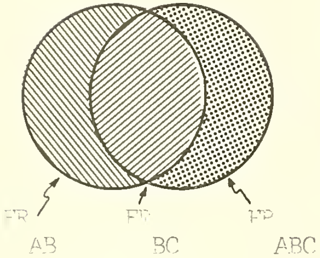
Figure B.1 illustrates these set theoretic relations. Note that two cases arise. One occurs when the deflected link BC is in the route of the new request and the other occurs when BC is <u>not</u> in the route of the new request.



Left Circle = FL_{AB}

Right Circle = FL_{BC}

a) Before Deflection



Left Circle = FL'_{AB}

Right Circle = FL'_{AC}

b) After Deflection

Figure B.1 Set Theoretic Relations for Deflection Algorithm

APPENDIX C

REROUTING SEARCH SCHEME FOR UNECONOMICAL LINKS

The sequence for attempting reroutings of requests in PASS2 is determined by establishing a value for the maximum number of circuits to be rerouted (FMAX) and a try-table (TRYTAB) of requests in descending order of fill requirements. After unsuccessfully trying to find a single request whose fill lies between FMAX and (FMAX + 4), the region from (FMAX - 1) to 1 is tested for a set of requests. If a request is successful and the fill number I is less than FMAX, a new value of FMAX is computed from (FMAX - I). If this new FMAX is greater than I-4 the search for requests continues downwards. If this new FMAX is less than (I-4) the direction of the search is reversed and the range is (FMAX + 4) to 1. This technique continuously offers the opportunity for finding a request that will cover the remaining range.

APPENDIX D

SAMPLE COMPUTER OUTPUTS

The following three pages contain sample printouts from the TELPAK program for the 1633 city problem. The first printout (page D-2) contains the beginning of the Link Table. In addition to the two terminal city designations, the link distance and the link fill, the TELPAK breakdown (D, C, Ixc) and the cost of the link appears. For this particular printout, it was also found of value to list the requests using this link. The second printout (page D-3) contains the beginning of the Request Table. In addition to the city designations for the request pair, the number of circuits required, the request airline distance, and the detour ratio, the printout also has the route length (in links) and the node (city) sequence of the route. The third printout (page D-4) contains the end of the Request Table plus the vital statistics resulting from this last run of the problem. These figures appear in Table 8.1 for the 1633 city problem.

```
      FROM        TO          O;STANCE        FILL       O TRUNK       C TRUNK      IXC          LINK COST

 1    AAGM        8KHX          69            2880          1             0            0     0.414000000000 04
        FOLLOWING REQUESTS USE LINK     1
          6     175     177     181     182     187     189     727     728     729    1623    1932
       1934    1935    2008    2396    2973    4560    4987    5116
 2    AAGM        FSCD          10            2935          1             0           55     0.737500000000 03
        FOLLOWING REQUESTS USE LINK     2
          6     175     177     181     182     187     189     727     728     729    1043    1211
       1932    1934    1935    2008    2395    2397    2398    2419    2751    2805    2899    2903
       2973    3085    3173    3213    3661    3663    3669    3954    3955    4277    4560    4668
       4728    4849    4917    4959    4971    4987    5007    5116    5139    5250    5288    5293
       5357
 3    AAGM        KHET           5             543          0             1            0     0.140000000000 03
        FOLLOWING REQUESTS USE LINK     3
       1043    1211    1623    2419    2805    2899    2903    3085    3173    3213    3661    3663
       3669    3954    3955    4277    4668    4728    4849    4917    4959    4971    5007    5139
       5250    5288    5293    5357
 4    AAGM        WHRT          15              24          0             0           24     0.900000000000 02
        FOLLOWING REQUESTS USE LINK     4
       2751
 5    AAJM        GNYH          31              24          0             0           24     0.175200000000 03
        FOLLOWING REQUESTS USE LINK     5
       2399
 6    AALP        EKXN          19              12          0             0           12     0.570000000000 02
        FOLLOWING REQUESTS USE LINK     6
       2400
 7    AART        CSQC          13              74          0             0           74     0.240500000000 03
        FOLLOWING REQUESTS USE LINK     7
       3043    3044
 8    AART        MEJE         149            2944          1             0           64     0.105720000000 05
        FOLLOWING REQUESTS USE LINK     8
        114     115     120     187     426     616     625     627     646     936     937    1126
       1301    1325    1338    1380    1388    1392    1438    1523    1524    1528    1532    1535
       1539    1642    1643    1645    1646    1967    1995    2150    2375    2496    2855    3809
       3811    3812    4265    4323
 9    AART        WNWJ          40            5722          2             0            0     0.480000000000 04
        FOLLOWING REQUESTS USE LINK     9
         81     114     115     120     187     293     301     426     502     616     625     627
        646     687     709     836     914     930     934     935     936     937     941    1121
       1126    1301    1315    1325    1338    1369    1380    1388    1392    1438    1443    1444
       1517    1531    1533    1535    1538    1539    1540    1543    1578    1585    1642    1643
       1645    1646    1932    1966    1967    1995    2019    2106    2109    2150    2166    2293
       2309    2375    2380    2496    2499    2713    2850    2852    2855    3346    3621    3622
       3751    3787    3803    3808    3809    3811    3812    3816    4015    4016    4165    4265
       4323    4658    4675    4708    4786    5171
10    AART        YXNO         122            2880          1             0            0     0.732000000000 04
        FOLLOWING REQUESTS USE LINK    10
         81     293     301     502     687     709     836     914     930     934     935     941
       1121    1315    1369    1443    1444    1517    1521    1522    1525    1526    1527    1529
       1530    1538    1540    1543    1578    1585    1932    1966    2019    2106    2109    2166
       2293    2309    2380    2499    2713    2850    2852    3043    3044    3346    3621    3622
       3751    3787    3803    3808    3816    4015    4016    4165    4658    4675    4708    4786
       5171
```

REQUEST LIST

```
NO.   CITY 1    CITY 2    NO. REQ.CRCTS        DIST.     DETOUR RATIO     SATISFIED      NO. LINKS

   1  ADUK      ASAZ                12          812        1.366995                            15
RDUTE ADUK      YKLF      KAPF            EEHR       XKQU    QQSN      CHXN        QMFU      BQN[illegible]          JTUW
      RQMA      MCUS      AEML            NAZP       YTDX    ASAZ

   2  ADUK      DKGK                26          262        1.240458                             5
RDUTE ADUK      YKLF      KTLH            UZXF       BBJM    DKGK

   3  ADUK      EJYP                48          143        1.195803                             4
RDUTE ADUK      YKLF      GWFD            KAPF       EJYP

   4  ADUK      FCHZ                24          562        1.096085                            10
ROUTE ADUK      YKLF      KAPF            AQLV       PEQE    DLUL      VZHX        UERY      NELF          ZAFN
      FCHZ

   5  ADUK      GZMW                24          661        1.142208                             8
ROUTE ADUK      YKLF      GWFD            KAPF       NMYH    THCW      JDVZ        DFVC      GZMW

   6  ADUK      HEKP                 6          792        1.702020                            33
ROUTE ADUK      YKLF      GWFD            KAPF       AQLV    PEQE      DLUL        VZHX      UERY          NELF
      ZAFN      GCVV      DTVP            PKVY       ZBFU    GJHZ      WAAK        EFGV      NLZT          JSDT
      RRTJ      ZRVE      DXGL            JDVZ       MNTA    YGRT      YLMH        YLLR      SFVV          FSCD
      AAGM      BKHX      TAEZ            HEKP

   7  ADUK      HMAZ                 2          775        1.310967                            12
RDUTE ADUK      WRDQ      XLWU            CHXN       TFEJ    XXDC      AEML        VQHV      XLSN          GBNZ
      EZTC      GFPR      HMAZ

   8  ADUK      KYJX                16          475        1.221052                             9
RDUTE ADUK      MQUK      KSMD            LSFY       SLEN    XNHH      SLWA        QCEY      TCEF          KYJX

   9  ADUK      LSFY                44          172        1.267442                             3
ROUTE ADUK      MQUK      KSMD            LSFY

  10  ADUK      MDWS                14          503        1.427435                            19
RDUTE ADUK      MQUK      KSMD            LSFY       SLEN    XNHH      SLWA        EBMM      PWYL          XWMQ
      HJTH      WKDH      LWQN            YD8T       HGJC    QCEY      TCEF        KYJX      WLHH          MDWS

  11  ADUK      MQUK               324          132        1.000000                             1
ROUTE ADUK      MQUK
```

```
5548   YYSA      ZLJS           12           267        1.247190                  4
ROUTE  YYSA      FZJZ       YJC=       SFAG        ZLJS

5549   ZAFN      ZBFU          132            68        1.220588                  4
ROUTE  ZAFN      GCVV       DTVP       FKVY        ZBFU

5550   ZAPN      ZCFU           12            26        1.384615                  2
ROUTE  ZAPN      KFDS       ZCFU

5551   ZKSE      RQUP           24            28        1.107142                  2
ROUTE  ZKSE      SNHG       RQUP

5552   ZKSE      WBCK           36            49        1.122449                  2
ROUTE  ZKSE      QJNT       WBCK
TOTAL COST =   0.35263183000D 07
REQMIL =       0.53453964167D 07
TOTAL PATH =        0.11314382965D 08
COST PER TRAVELED WIRE MILE =      0.31184363398D 00
COST PER REQUIRED WIRE MILE =      0.37754613530D 00
AVERAGE DETOUR RATIO =       0.12106905326D 01

 LENGTH OF LINK TABLE =   28889
LENGTH OF ROUTE TABLE =   82044
```


☆ U. S. GOVERNMENT PRINTING OFFICE : 1973—511-327/285

FORM NBS-114A (1-71)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS TN-787	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE Heuristic Cost Optimization of the Federal Telpak Network			5. Publication Date 6. Performing Organization Code
7. AUTHOR(S) R. G. Saltman, G. R. Bolotsky, Z. G. Ruthberg			8. Performing Organization
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			10. Project/Task/Work Unit No. 640-2456 11. Contract/Grant No. 1-35859
12. Sponsoring Organization Name and Address Defense Communications Agency 8th Street & South Courthouse Road Arlington, Virginia 22204			13. Type of Report & Period Covered Final 14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)

A heuristic method of optimizing the design of a very large communications network is described. The procedure is employed to configure the routes of 5552 communications service requests involving 1633 nodes. A FORTRAN IV program was developed to solve for actual needs of the Defense Communications Agency for leased-line service employing the Telpak tariff structure.

17. KEY WORDS (Alphabetical order, separated by semicolons)
Communications network; computer program; heuristic; minimum cost; network configuration; optimization; Telpak rate structure.

18. AVAILABILITY STATEMENT	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
[X] UNLIMITED.	UNCLASSIFIED	52
[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NTIS.	20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price $.80 Domestic postpaid $.55 G.P.O. Bookstore

USCOMM-DC 66244-P71

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH reports National Bureau of Standards research and development in physics, mathematics, and chemistry. Comprehensive scientific papers give complete details of the work, including laboratory data, experimental procedures, and theoretical and mathematical analyses. Illustrated with photographs, drawings, and charts. Includes listings of other NBS papers as issued.

Published in two sections, available separately:

- **Physics and Chemistry (Section A)**

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, $17.00; Foreign, $21.25.

- **Mathematical Sciences (Section B)**

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, $9.00; Foreign, $11.25.

TECHNICAL NEWS BULLETIN

The best single source of information concerning the Bureau's measurement, research, developmental, cooperative, and publication activities, this monthly publication is designed for the industry-oriented individual whose daily work involves intimate contact with science and technology—*for engineers, chemists, physicists, research managers, product-development managers, and company executives.* Includes listing of all NBS papers as issued. Annual subscription: Domestic, $6.50; Foreign, $8.25.

NONPERIODICALS

Applied Mathematics Series. Mathematical tables, manuals, and studies.

Building Science Series. Research results, test methods, and performance criteria of building materials, components, systems, and structures.

Handbooks. Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications. Proceedings of NBS conferences, bibliographies, annual reports, wall charts, pamphlets, etc.

Monographs. Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

National Standard Reference Data Series. NSRDS provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated.

Product Standards. Provide requirements for sizes, types, quality, and methods for testing various industrial products. These standards are developed cooperatively with interested Government and industry groups and provide the basis for common understanding of product characteristics for both buyers and sellers. Their use is voluntary.

Technical Notes. This series consists of communications and reports (covering both other-agency and NBS-sponsored work) of limited or transitory interest.

Federal Information Processing Standards Publications. This series is the official publication within the Federal Government for information on standards adopted and promulgated under the Public Law 89–306, and Bureau of the Budget Circular A–86 entitled, Standardization of Data Elements and Codes in Data Systems.

Consumer Information Series. Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service (Publications and Reports of Interest in Cryogenics). A literature survey issued weekly. Annual subscription: Domestic, $20.00; foreign, $25.00.

Liquefied Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $20.00. Send subscription orders and remittances for the preceding bibliographic services to the U.S. Department of Commerce, National Technical Information Service, Springfield, Va. 22151.

Electromagnetic Metrology Current Awareness Service (Abstracts of Selected Articles on Measurement Techniques and Standards of Electromagnetic Quantities from D-C to Millimeter-Wave Frequencies). Issued monthly. Annual subscription: $100.00 (Special rates for multi-subscriptions). Send subscription order and remittance to the Electromagnetic Metrology Information Center, Electromagnetics Division, National Bureau of Standards, Boulder, Colo. 80302.

Order NBS publications (except Bibliographic Subscription Services) from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300