AD 639602

MEMORANDUM
RM-4945-PR
MAY 1966



# AN INTERDICTION MODEL OF HIGHWAY TRANSPORTATION

Eugene P. Durbin



PREPARED FOR:

UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM
RM-4945-PR
MAY 1966

# AN INTERDICTION MODEL OF HIGHWAY TRANSPORTATION

Eugene P. Durbin

This research is sponsored by the United States Air Force under Project RAND—Contract No. AF 49(638)-1700—monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff. Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

The RAND Corporation
1700 MAIN ST • SANTA MONICA • CALIFORNIA • 90406

# PREFACE

This Memorandum describes a computer program that demonstrates the effect of denying the use of a portion of a highway transportation network. This application utilizes previous RAND research on network flows and on highway capacity.* The program is being used in the continuing RAND research on the effective employment of air power, including the employment of tactical air forces.

The model was originally intended for internal use at RAND, but a number of other agencies have indicated that it might prove useful in their research, including the Weapons Systems Evaluation Group (WSEG) and the Operations Analysis Office, Hq USAF (AFGOA). It should be of interest both to those concerned with targeting strikes against road networks and to those concerned with allocating road repair and improvement efforts.

---

*L. R. Ford, Jr., and D. R. Fulkerson, Flows in Networks, The RAND Corporation, R-375-PR (DDC No. AD 287499), August 1962 (also published by Princeton University Press, 1962); R. D. Wollmer, Some Methods for Determining the Most Vital Link in a Railway Network, The RAND Corporation, RM-3321-ISA, April 1963; L. P. Holliday, A Method for Estimating Road Capacity and Truck Requirements (U), The RAND Corporation, RM-3331-ARPA, November 1963 (Confidential).

## SUMMARY

This Memorandum describes a computer program designed to evaluate the capability of a transportation network to deliver supplies to destinations as road segments or arcs making up the network are successively destroyed and repaired. The program, written in FORTRAN IV, can be adapted easily for use on any of several large-scale computers.

As inputs, the program requires a description of the transportation road system under consideration, and of the cargo-carrying vehicles operating on the network. Given the basic data, the program furnishes a profile of maximum cargo flow as a function of the number of vehicles made available to the system, and then selects and destroys that vulnerable link in the network which reduces the cargo flow rate most severely. The program repeats these steps until flow on the network is totally stopped or the predesignated number of links have been destroyed. The program then steps to the next "day" or "period," restores to service all previously destroyed links that have been repaired by this date, and repeats the process of profile generation and link removal.

This model uses the Fulkerson Out-of-Kilter Algorithm* to generate the cargo flow profile as a function of vehicles in the system, and uses an algorithm of Wollmer** to determine the single most critical link in a network.

The program presently will accept a network of up to 1000 links, but this number may be modified to suit the capacity of a particular computer system. By properly describing the network, some combinations of rail, road, and water transportation can be analyzed. The program should be a useful tool in targeting, in logistics system analysis, in allocating funds for expansion of transportation systems, and allocating road-repair efforts.

---

*Ford and Fulkerson, op cit.; D. R. Fulkerson, An Out-of-Kilter Method for Minimal Cost Flow Problems, The RAND Corporation, P-1825, April 1960 (also published in J. Soc. Ind. Appl. Math., Vol. 9, March 1961, pp. 18-27).

**Wollmer, op. cit.

## CONTENTS

## I. INTRODUCTION

The general tactical interdiction problem is to select target elements from among the vehicles, road links, rail nets, waterways, and air lines supplying a combat force in order to effectively reduce the combat capability of that force. In order to select an optimal set of targets to attack, a general interdiction model would presumably consider both the weapons and alternatives available to the interdictor, and the current configuration and recuperative ability of the transportation system. This selection would change over time as lucrative concentrations of vehicles were observed, as critical segments of the network were repaired and returned to service, and as the weapons available to the interdictor changed. The obvious information one might require from such a model would be the ability of the transportation system to support the combat force at various levels of attack, or the level of attack required to degrade combat performance to a specified value.

The more restricted situation considered by the model described here explicitly includes only a single type of cargo-carrying vehicle moving over a highway network. The restriction to a single type of cargo vehicle was imposed since it allows the use of very efficient algorithms to calculate maximum flow through a network, and it was decided to consider highway rather than water, air, or rail transportation since research by L. P. Holliday* allows the network concept of arc capacity to be applied to highways. The key assumption in this model is that the maximum speed with which vehicles can flow across an arc is independent of the volume of flow on the arc.

Given this assumption, the highway transportation system is described as a network of nodes and directed arcs together with the vehicles that move on these arcs. Nodes may be towns, intersections, or any points at which it is convenient to distinguish between the road characteristics on either side of the node. Each arc in the network

---

*Holliday, op. cit.

is characterized by the name of the node at which it originates, the name of the node at which it terminates, the maximum flow capacity on the arc in vehicles per time-unit, the number of time-units required for a vehicle to move across the arc, and the number of time-units required to repair the arc if it should be rendered impassable. If an arc is assigned a repair time of zero, it is considered invulnerable and will not be cut. Arcs may have minimum flow requirements assigned in tons per day, permitting demands at various destinations to be specified.

Other inputs to the model are the number of vehicles in the inventory, vehicle tonnage capacity, vehicle in-commission rate, en route stop factor (to convert en route time to total time), the number of arcs that can be cut per day, the number of days the model is to consider, and the conversion factors between time-units, hours, and operating days. Holliday* discusses the methodology for determining road capacities, maximum road speeds, and time adjustments required by one-way use of the roads, waiting time, and convoying.

Although the actual number of vehicles in the inventory is specified to the model, the problem solved is that of determining throughput capability, measured in vehicles per unit time or, equivalently, tons per day, as a function of the number of usable arcs in the network as the number of vehicles in the system varies from zero up to network saturation level. This is done so that when use of an arc is denied, the resulting profile of throughput as a function of vehicles available to the system can be compared directly to the previous profile to determine the "vehicle-worth" of the newly removed arc.

This model makes repeated use of the Fulkerson Out-of-Kilter algorithm,** which is an efficient algorithm for constructing minimal cost flows in networks in which the arcs have costs, and both upper and lower bounds on permissible flow.

In selecting the most critical arc for removal, the model uses an algorithm developed by R. D. Wollmer.*** The essence of this algorithm is that in considering a network with a maximal flow, F,

---

*Ibid.

**Ford and Fulkerson, op. cit.; Fulkerson, op. cit.

***Wollmer, op. cit.

given the usable arcs of the network and the actual number of vehicles available to the system, there is no need to consider those arcs with zero flow as candidates for removal. Removing such arcs would contribute nothing toward the goal of identifying that arc which, if removed, would most drastically reduce throughput.

## II. DESCRIPTION OF THE MODEL

The problem of determining maximum flow through a network as a function of the number of vehicles in the network is seen clearly when formulated as a parametric linear programming problem.

Let $t_{ij}$ = time-units required for a vehicle to move from Node i to Node j

$u_{ij}$ = maximum flow capacity in vehicles per unit time on the arc (i,j)

$\ell_{ij}$ = minimum flow requirement in vehicles per unit time on the arc (i,j)

$x_{ij}$ = number of vehicles per unit time passing over the arc (i,j)

$d_i$ = minimum number of vehicles per unit time required at Destination i

T = number of vehicles available to the system.

Connect an artificial source, Node 0, to all true sources, and an artificial destination, Node n+1, to all true destinations. Also introduce an arc directed from the artificial destination to the artificial source. Arcs connected to artificial nodes are generally assigned no cost and infinite capacity. The problem is then

(1)
$$\underset{x_{ij}}{\text{maximize}}\ x_{n+1,0}$$

subject to

(2)
$$\begin{cases} \sum_i x_{ij} - \sum_i x_{ji} = 0 \text{ for each } j \\ \\ \ell_{ij} \le x_{ij} \le u_{ij} \text{ for all arcs } (i,j) \end{cases}$$

where

$$\ell_{ij} = d_i \text{ for all destinations } i, \text{ and}$$

(3) $$\sum_i t_{ij} x_{ij} \le T.$$

While the problem is seen clearly when formulated in this way, the presence of Constraint (3) precludes solution by standard capacitated transportation algorithms, while the size of realistic networks precludes solution by standard linear programming algorithms. The model therefore considers the inverse problem -- that of plotting the number of vehicles required by the system as a function of the number of vehicles flowing through the system. In the previous notation this is:

(4) $$\underset{x_{ij}}{\text{minimize}} \quad \sum t_{ij} x_{ij} ,$$

with

(5) $$\begin{cases} \sum_i x_{ij} - \sum_i x_{ji} = 0 \text{ for each } j \\ \ell_{ij} \le x_{ij} \le u_{ij} \text{ for all arcs } (i,j) , \end{cases}$$

where flow in the system is successively incremented by incrementing the lower bound on the return flow, $\ell_{n+1,0}$ . Solving a sequence of such problems yields points on the profile of vehicles required by the system as a function of vehicles flowing through the system. This is termed a flow profile. The number of such profile points to be generated is specified as an input to the program.

Each time a flow profile is to be generated, the model first computes the maximum flow through the network constrained only by the arc capacities and the demands at the various destinations. This maximum flow is divided by the number of requested profile points to obtain the quantity of flow by which the system will be successively incremented. The vehicle minimization problem stated in Expressions (4) and (5) is actually solved at each level of flow, F, and also at F+1, which approximates determining the derivative of vehicle requirements at F, and facilitates plotting the profile. If the actual number of vehicles declared to be available to the system falls between the values computed at two successive profile points, a series of interpolations are carried out that yield a profile point within 50 vehicles of the actual number. If the flow that the model is attempting to force through the system is less than the sum of all the demands imposed on the destinations, the model reduces the demands to the proper proportion of the original demands. If the model determines that either the actual or the proportionately reduced demands cannot be satisfied, no further profiles are generated for the current day. The network is assumed to be interdicted.

After the profile is generated for a fixed network configuration, the program selects an arc to remove. First, a maximum flow capacity is imposed on the return arc (n+1,0), which prevents maximum flow in the network from exceeding that flow which can be sustained by the actual number of vehicles declared available to the system. Vulnerable arcs are then removed in turn while an attempt is made to maximize flow through the network subject to the new constraint on maximum flow. As flow is maximized with successive arcs removed for test, all arcs on which there is zero flow are removed from further consideration. That arc is finally selected for removal which causes maximum flow through the network -- constrained by the actual number of vehicles available -- to be minimized. If several arcs yield the same minimum flow value, f, that arc is removed which requires the greatest number of vehicles to be used at the flow, f. Arc removal is accomplished in the program by setting arc maximum capacity to zero. At the time the arc is removed, its return-to-service time is updated by the arc repair time. When the return-to-service time is reached, the arc is restored to full use.

## III. PROGRAM SEQUENCE OF OPERATION

The program operates in daily cycles for the number of days specified by the user. At the start of Day k, all arcs with appropriate return-to-service dates are restored to full capacity. An attempt is then made to generate the flow profile at the start of Day k. If an insufficient number of arcs have been repaired by this date, and no flow is possible, the message "FLØW TØTALLY STØPPED ØN DAY K" will be printed and the program will cycle to Day k+1. If flow through the network is possible, but not in a pattern appropriate to meet demands imposed at various destinations, the message "MIN DEMANDS AT SØME DESTINATIØN CANNØT BE SATISFIED" will be printed, and the program will cycle to Day k+1. If neither of these conditions occurs, the flow profile at the start of Day k will be printed.

Next, an arc will be selected for removal, its capacity will be set equal to zero, and the flow profile resulting from the new network configuration will be generated and printed. The cycle of arc removal and flow profile generation continues until either minimum demand cannot be met at some destination, a preassigned number of arcs have been removed, or flow is totally stopped. When any of these conditions occur, the model proceeds to Day k+1 and the sequence begins again. The program prints several other self-explanatory data-editing messages and statements describing the arcs that have been cut.

# IV. INPUT DESCRIPTION AND PREPARATION

## MODEL CONTROL DATA

1. Number of Vehicles in the Inventory (NTRKS). The model uses this information only to place a marker (***) at the appropriate point in the flow profile. The flow profile is continued until network saturation.

2. Length of System Evaluation (NDAYS). This is the number of days for which the model is to continue network evaluation. Since the present formulation contains no random elements, it is useless to continue evaluation beyond the point at which network behavior begins to repeat itself.

3. Number of Arcs to be Removed Each Day (NARCS). This is the maximum number of arcs that will be removed each day. If the network is interdicted prior to this maximum number of cuts, arc removal will cease.

4. Number of Profile Points (NPPTS). The profile of flow versus vehicles will be drawn through the plotted points. This number of points must be specified.

5. Name of Artificial Source and Destination (SOURCE,SINK). Two artificial nodes are required. One will be the artificial source and must be connected to all true sources. The other is the artificial destination and all true destinations must be connected to it. An arc must also connect the artificial destination to the artificial source.

## VEHICLE AND OPERATING DATA

1. Tons per Vehicle (TNSTRK). All vehicles carry the same load. The model actually evaluates vehicle flow per time-unit and then converts this to flow in tons per day.

2. Time-Units per Hour (TUPRHR). Arc data can be expressed in arbitrary time-units. TUPRHR is the factor that defines the number of such arbitrary time-units per hour.

3. Operating Hours per Day (HRPRDA). Vehicles operate only for the number of hours specified by this input.

4. In-Commission Factor (FINCOM). The in-commission factor is the fraction of the total inventory available for dispatch.

5. Stop Factor (FSTOP). This is a multiplicative factor that can be used to increase the total number of vehicles in the system to account for vehicle stops, refueling, etc. If K vehicles are actually required to provide a flow of F vehicles per unit time at the final destination, the model reports that $\lceil K \cdot FSTOP/FINCOM \rceil$ vehicles are required to deliver $F \cdot TNSTRK \cdot TUPRHR \cdot HRPRDA$ tons per day to the destination.

ARC DATA

Arcs are directed from Node i to Node j. The names of the nodes at the beginning and end of each arc are required. If flow is possible in both directions between i and j, an arc must be entered from Node j to Node i. When an arc is cut between i and j, the reverse arc from j to i is also cut. If there are multiple arcs in both directions between i and j, the appropriate reverse arc is found by searching for the reverse arc with the identical repair time.

The program will accept up to 1000 arcs and up to 500 nodes.

The cost of obtaining a flow of one vehicle per time-unit from i to j is required. In the ground-transportation problem this cost has been the number of time-units required for one vehicle to traverse the arc (i,j) in both directions, given that the vehicle is moving at the most efficient speed for the arc. The cost should include waiting time required by single-lane operation on narrow roads.

Maximum capacity for each arc is required in terms of vehicles per time-unit passing an average point on the arc.

Minimum demand at each destination can be specified by placing a minimum required flow in tons per day on the arc connecting the destination and the artificial destination.

Repair times of the arcs are entered in time-units. If a repair time is zero, the arc will never be selected for removal.

The artificial arcs connecting artificial to true sources, true to artificial destinations, and artificial destinations to artificial sources should have very large upper capacities, zero repair times, and minimum flow requirements where appropriate.

## INPUT DATA FORMAT

I denotes an integer right-justified in its field.

D denotes a decimal number anywhere in the field. If no decimal point is explicitly entered, the decimal point is understood to be to the right of the entire field.

A denotes any combination of alphabetic and numeric data.

### First Card

| | | |
|---|---|---|
| Col 1-10 | Number of Vehicles (NTRKS) | (I) |
| Col 11-20 | Number of Days (NDAYS) | (I) |
| Col 21-30 | Number of Links (NARCS) | (I) |
| Col 31-40 | Number of Profile Points (NPPTS) | (I) |
| Col 43-48 | Name of Artificial Source | (A)* |
| Col 51-56 | Name of Artificial Destination | (A)* |

### Second Card

| | | |
|---|---|---|
| Col 1-5 | Tons per Vehicle (TNSTRK) | (D) |
| Col 6-10 | In-Commission Factor (FINCOM) | (D) |
| Col 11-15 | Stop Factor (FSTOP) | (D) |

### Third Card

| | | |
|---|---|---|
| Col 1-5 | Operation Hours per Day (HRPRDA) | (D) |
| Col 6-10 | Time-Units per Hour (TUPRHR) | (D) |

---

*Node names are read as six characters, including blanks. Therefore node names of less than six characters must be identically placed in their field whenever written.

Fourth Card

Col 1-5 The word "READY"

Fifth Card

Any arbitrary title information in Cols 1-72. This will be printed on the output.

Sixth Card

Col 1-4 The word "ARCS"

Arc Data Cards

Each arc data card has the following format.

| | | | |
|---|---|---|---|
| Col | 7-12 | Name of First Node, i | (A) |
| Col | 13-18 | Name of Second Node, j | (A) |
| Col | 21-30 | Cost of One Unit of Flow, $t_{ij}$ | (I) |
| Col | 31-40 | Maximum Capacity of Arc, $u_{ij}$ | (I) |
| Col | 41-50 | Minimum Flow on Arc, $\ell_{ij}$ | (I) |
| Col | 61-63 | Repair Time of Arc, $r_{ij}$ | (I) |

Next Card

Col 1-3 The word "END"

Next Card

Col 1-7 The word "COMPUTE"

Figure 1 illustrates a hypothetical network on which arc capacities and costs have been entered. Node A is the source, the double-lined arcs are invulnerable, and the demand at Node F is 440 tons per day. Artificial nodes and arcs are first introduced as shown, and it is assumed that each vehicle in the system carries 3 tons, that the in-commission factor is 0.75, that the stop factor is 1.25, that vehicles operate 15 hours per day, and that all arc data are expressed in terms

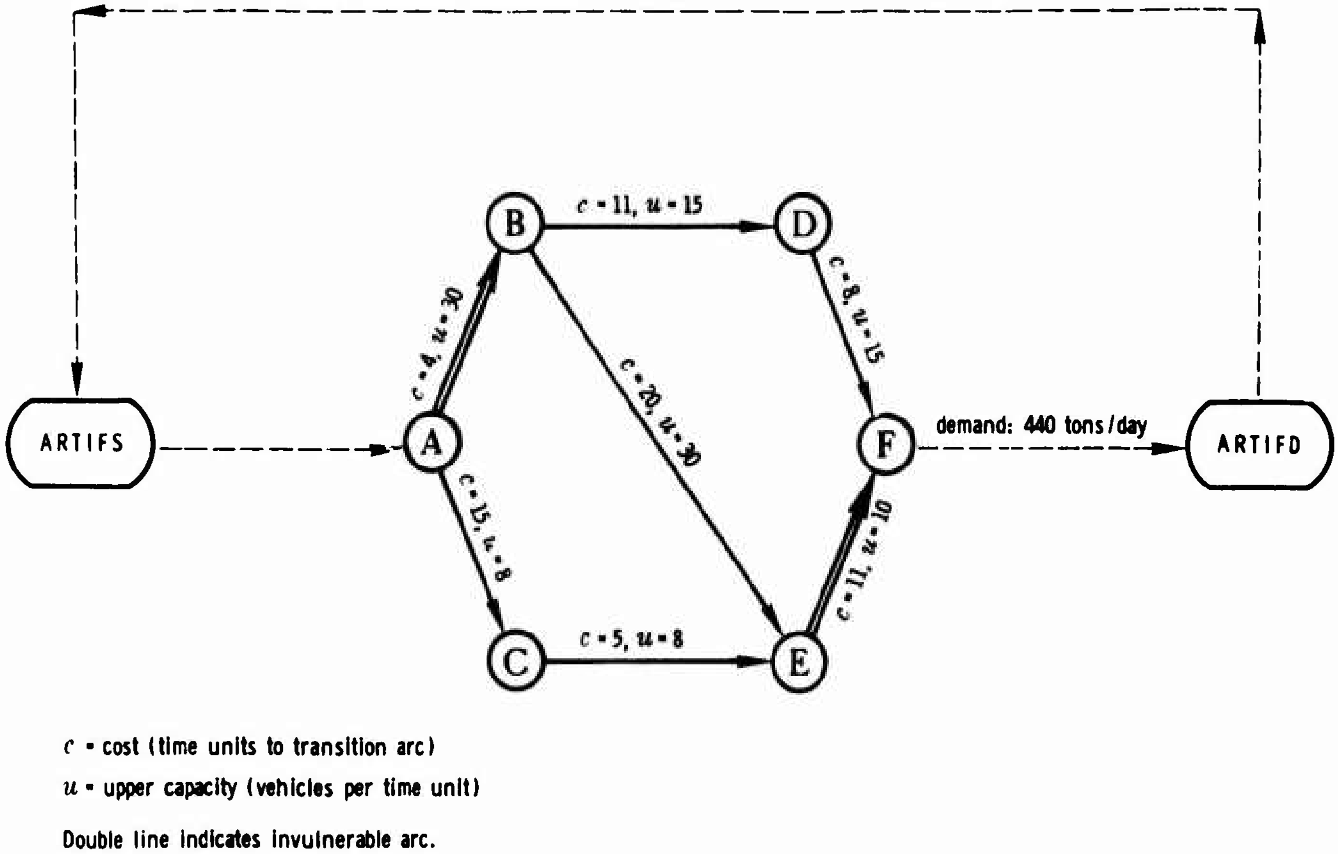
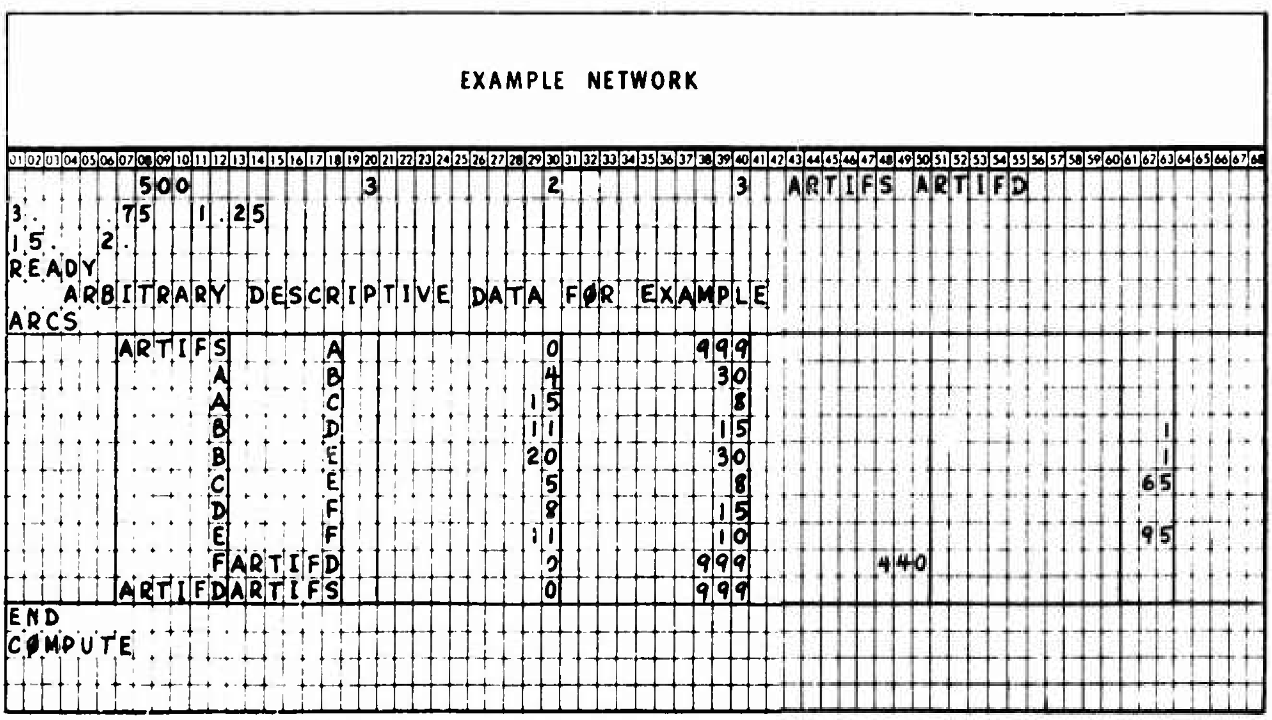



Fig. 1 -- A Hypothetical Network



EXAMPLE NETWORK

```
       500         3         2         3  ARTIFS ARTIFD
3.   .75  1.25
15.  2.
READY
   ARBITRARY DESCRIPTIVE DATA FØR EXAMPLE
ARCS
      ARTIFS     A           0       999
           A     B           4        30
           A     C          15         8
           B     D          11        15                      1
           B     E          20        30                      1
           C     E           5         8                     65
           D     F           8        15
           E     F          11        10                     95
           FARTIFD           0       999       440
      ARTIFDARTIFS           0       999
END
CØMPUTE
```

Fig. 2 -- Input Data Describing Example Network

of 30-minute time-units. A three-day evaluation is desired, with two arcs to be removed per day, and three points generated on each flow profile. Figure 2 indicates how these data would be entered. The Appendixes depict the interdiction program and its output.

# Appendix A

## INTERDICTION PROGRAM OUTPUT

ARBITRARY DESCRIPTIVE DATA FOR EXAMPLE

ARCS

| ARCS FROM | TO | COST | UPPER | LOWER | TIME TO REPAIR |
|---|---|---|---|---|---|
| ARTIFS | A | 0 | 999 | -0 | 0 |
| A | B | 4 | 30 | -0 | -0 |
| A | C | 15 | 8 | -0 | 1 |
| B | D | 11 | 15 | -0 | 1 |
| B | E | 20 | 30 | -0 | 65 |
| C | E | 5 | 8 | -0 | -0 |
| D | F | 8 | 15 | -0 | 95 |
| E | F | 11 | 10 | -0 | -0 |
| F | ARTIFD | -0 | 999 | 4 | -0 |
| ARTIFD | ARTIFS | -0 | 999 | -0 | -0 |

END
COMPUTE
THIS PROBLEM HAS 10 ARCS AND 8 NODES

D A Y 1

FLOW PROFILE PRIOR TO FIRST CUT

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 306 | 720 | 184 |
| 345 | 810 | 207 |
| 460*** | 1080 | 276 |
| 626 | 1440 | 376 |
| 678 | 1530 | 407 |
| 1046 | 2160 | 628 |
| 1105 | 2250 | 663 |

***APPROXIMATES THE ACTUAL NUMBER OF TRUCKS, 500.

CUT NUMBER 1 REDUCED CAPACITY ON ARC B TO D FROM 1350. TO 0.
ARC WILL BE RESTORED ON DAY 1.

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 155 | 270 | 93 |
| 206 | 360 | 124 |
| 310 | 540 | 186 |
| 361 | 630 | 217 |
| 471 | 810 | 283 |
| 471*** | 810 | 283 |
| 530 | 900 | 318 |

***APPROXIMATES THE ACTUAL NUMBER OF TRUCKS, 500.

CUT NUMBER 2 REDUCED CAPACITY ON ARC B TO E FROM 2700. TO 0.
ARC WILL BE RESTORED ON DAY 3.

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 103 | 180 | 62 |
| 155 | 270 | 93 |
| 206 | 360 | 124 |
| 258 | 450 | 155 |
| 310 | 540 | 186 |
| 413 | 720 | 248 |

D A Y 2

FLOW PROFILE PRIOR TO FIRST CUT

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 268 | 630 | 161 |
| 306 | 720 | 184 |
| 498*** | 1170 | 299 |
| 536 | 1260 | 322 |
| 575 | 1350 | 345 |
| 885 | 1890 | 531 |
| 988 | 2070 | 593 |

***APPROXIMATES THE ACTUAL NUMBER OF TRUCKS, 500.

CUT NUMBER 1 REDUCED CAPACITY ON ARC B TO D FROM 1350. TO 0.
ARC WILL BE RESTORED ON DAY 2.

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 103 | 180 | 62 |
| 155 | 270 | 93 |
| 206 | 360 | 124 |
| 258 | 450 | 155 |
| 310 | 540 | 186 |
| 413 | 720 | 248 |

CUT NUMBER 2 REDUCED CAPACITY ON ARC A TO C FROM 720. TO 0.
ARC WILL BE RESTORED ON DAY 2.
FLOW TOTALLY STOPPED ON DAY 2

D A Y 3

FLOW PROFILE PRIOR TO FIRST CUT

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 306 | 720 | 184 |
| 345 | 810 | 207 |
| 460*** | 1080 | 276 |
| 626 | 1440 | 376 |
| 678 | 1530 | 407 |
| 1046 | 2160 | 628 |
| 1105 | 2250 | 663 |

***APPROXIMATES THE ACTUAL NUMBER OF TRUCKS, 500.

CUT NUMBER 1 REDUCED CAPACITY ON ARC B TO D FROM 1350. TO 0.
ARC WILL BE RESTORED ON DAY 3.

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 155 | 270 | 93 |
| 206 | 360 | 124 |
| 310 | 540 | 186 |
| 361 | 630 | 217 |
| 471 | 810 | 283 |
| 471*** | 810 | 283 |
| 530 | 900 | 318 |

***APPROXIMATES THE ACTUAL NUMBER OF TRUCKS, 500.

CUT NUMBER 2 REDUCED CAPACITY ON ARC B TO E FROM 2700. TO 0.
ARC WILL BE RESTORED ON DAY 5.

| TRUCKS IN INVENTORY | TONS/DAY THROUGHPUT | TRUCKS ACTIVE |
|---|---|---|
| 103 | 180 | 62 |
| 155 | 270 | 93 |
| 206 | 360 | 124 |
| 258 | 450 | 155 |
| 310 | 540 | 186 |
| 413 | 720 | 248 |

# Appendix B

## INTERDICTION PROGRAM

```
$IBFTC CIRSEA
      COMMON  /RTRK/  IRFLO
      COMMON  /MARKR/  IT(1001)
       COMMON /FACTR/  FACTOR
      COMMON  /RTIME/  INTIME(1001)
      COMMON  /NNODE/  ISINK
      COMMON  /DENOM/  IDENOM
      COMMON  /ORIG/M,N,KB(12),NIT,NOT,NN,NP,II,IJ,KC,KU,LB,KX,NL,KV
      COMMON /LFLAG/ KINTRP,KCUT
      COMMON  /FACT/  FINCOM, FSTOP,TNSTRK,HRPRDA,TUPRHR
      COMMCN  /KDAY/  KPDS,LINK,LINK1
      COMMON  /LBUG/  BUGWRD
     DIMENSION II(1001),IJ(1001),KC(1001),KU(1001),LB(1001),KX(1001)
     DIMENSION NL( 501),NP( 501),NN( 501),KV(1001)
      DIMENSION  KCS(1001),KUS(1001),LBS(1001)
      DIMENSION  LOC(2)
      LOGICAL KINTRP,KCUT
      LOGICAL  BUGWRD
C
     READ (5,2) NTRKS,NPDS,NLINKS,NPPTS,IOURCE,ISINK,BUGWRD
      READ (5,20) TNSTRK,FINCOM,FSTOP,HRPRDA,TUPRHR
   20 FORMAT ( 3F5.0/2F5.0)
     WRITE (6,3) NTRKS,NPDS,NLINKS,NPPTS   ,IOURCE,ISINK,
    X  TNSTRK,FINCOM,FSTOP,HRPRDA,TUPRHR
C
C         INITIALIZATION
     CALL QKINPT
C    SAVE THE ORIGINAL DATA
      DO 5000 I = 1,N
      KUS(I) = KU(I)
 5000 LBS(I) = LB(I)
C    HERE FIND THE DESTINATION LINK.
 5001  DO 5002 I = 1,N
     N2 = IJ(I)
      IF (NN(N2).EQ. IOURCE)  GO TO 50002
 5002  CONTINUE
50002  ISAVE = I
C         FIND THE SUM OF THE DEMAND RATIOS
      IDENOM = 0
      DO 5003  I = 1,N
      N2 = IJ(I)
      IF ( NN(N2).NE.ISINK) GO TO 5003
      IDENOM = IDENOM + LB(I)
 5003  CONTINUE
      IDENOM = MAX0(1,IDENOM)
C
C                MAIN ROUTINE
C
C   ONCE FOR EACH DAY
     DO 999 KPDS = 1,NPDS
      KCUT = .FALSE.
C  THE NETWORK ISNT CUT YET
       WRITE (6,1990)  KPDS
 1990  FORMAT ( 1H1, 23X,7HD A Y  ,I2 /   )
      WRITE (6,2001)
 2001 FORMAT (1H ,12X,  32H FLOW PROFILE PRIOR TO FIRST CUT
C   RESTORE ALL REPAIRED LINKS
      DO 5004  I = 1,N
      IF ( INTIME(I) .LE.KPDS) KU(I) = KUS(I)
 5004  LB(I) = 0
      NLINK1 = NLINKS + 1
C
C     NOW BEGIN THE STRIKES FOR THIS CURRENT DAY
      DO 9990  KLINKS = 1,NLINK1
C     FIND MAX FLOW
      KU(ISAVE) = KUS(ISAVE)
       LB(ISAVE) = 0
      KC(ISAVE) =-9999999
      CALL MNCF(M,N,II,IJ,KC,KU,LB,KX,NP,NL,INFEAS)
      KC(ISAVE) = 0
     IF (BUGWRD) CALL OUTPUT(II,IJ,KC,KU,LB,KX,NL,NP,INFEAS,KV)
C  RESTORE ALL TRUE DEMANDS
      DO 50004  I = 1,N
```

```
50004  LB(I) = LBS(I)
      IF ((INFEAS.NE.1).AND.(KX(ISAVE).NE.0)) GO TO 5005
       IF ( KX(ISAVE).EQ.0) GO TO 10117
       WRITE (6,10114)
10114  FORMAT ( 1H0,51HMIN DEMANDS AT SOME DESTINATION CANNOT BE SATISFI
     XED     )
       GO TO 999
10117  WRITE (6,4) KPDS
   004  FORMAT ( 1H ,5X,27HFLOW TOTALLY STOPPED ON DAY,I3)
       GO TO 999
 5005 INC = MAX0(1,KX(ISAVE)/NPPTS)
       KSAVE = KX(ISAVE)
       WRITE (6,2000)
C   WE HAVENT INTERPOLATED ON THE TRUE NUMBER OF TRUCKS YET FOR THE COMIN
C    PROFILE
       KINTRP = .TRUE.
C
C           FOR THIS NETWORK CONFIGURATION PLOT THE PROFILE
              DO 99990 KPPTS = 1,NPPTS
       IFLO = INC*KPPTS
C   IEND IS TO TEST THE LAST ITERATION WITH ONLY ONE WORD
       IEND = NPPTS-KPPTS
       CALL PROFIL(ISAVE,ISINK,IFLO,NTRKS,LBS,IDENOM,IEND,KSAVE)
       IF (KCUT) GO TO 999
99990           CONTINUE
      IF ( .NOT.KINTRP) WRITE (6,10134) NTRKS
10134 FORMAT  ( 48H0   ***APPROXIMATES THE ACTUAL NUMBER OF TRUCKS, ,I6,
     X  1H.)
99991  IF (KLINKS.GT.NLINKS) GO TO 9990
C   NOW CHOOSE A LINK TO CUT
        CALL WOLLMR(KX,KU,NN,II,IJ,N,KV,KLINKS,LB,KC,M,NP,NL,ISAVE,LBS)
 9990   CONTINUE
  999 CONTINUE
      CALL EXIT
C       FORMAT STATEMENTS
    2 FORMAT (4I10,2X,A6,2X,A6,23X,L1)
  003 FORMAT(10H1 TRUCKS =   ,I6,1X,10H PERIODS =   ,I6,1X,8HLINKS =   ,
     X  I6,1X,8H NPPTS =   ,I6,1X,9HSOURCE = ,A6,1X,7HSINK = ,A6/
     X 10H TONS/TRK  ,F8.4,2X,6HINCOM , F8.4,2X,5HSTOP ,F8.4,2X,
     X  8HHRS/DAY ,F8.4,2X,6HTU/HR ,F8.4 )
 2000 FORMAT ( 1H0,6X,6HTRUCKS,11X,8HTONS/DAY,17X,6HTRUCKS/
     X 4X,12HIN INVENTORY,7X,10HTHROUGHPUT,16X,6HACTIVE   )
      END
$IBFTC WOLLMR
       SUBROUTINE WOLLMR(KY,KU,NN,II,IJ,N,KV,KLINKS,LB,KC,M,NP,NL,ISAVE,
     X  LBS)
       COMMON  /RTRK/ IRFLO
       COMMON  /NNODE/  ISINK
       COMMON  /DENOM/  IDENOM
       COMMON  /FACTR/  FACTOR
        COMMON  /FACT/  FINCOM,FSTOP,TNSTRK,HRPRDA,TUPRHR
       COMMON  /MARKR/ IT(1001)
       COMMON  /LBUG/  BUGWRD
       COMMON  /RTIME/  INTIME(1001)
       COMMON  /KDAY/  KPDS,LINK,LINK1
       DIMENSION LB(1),KC(1),LBS(1)
       DIMENSION KY(1),KU(1),NN(1),II(1),IJ(1),KV(1)
       DIMENSION  LOC(2)
       LOGICAL BUGWRD
       WRITE (6,10017)
10017  FORMAT ( 1H0)
      LINK = 0
        MFLO = 999999
       ITRQ = 0
       LB(ISAVE) = 0
       KC(ISAVE) = -9999999
       KU(ISAVE) = IRFLO
       IF(IRFLO.EQ.0) KU(ISAVE) = 9999999
       DO 1 I = 1,N
       IT(I) = MIN0(KV(I),KU(I))
       N2 = IJ(I)
       IF(NN(N2).NE.ISINK) GO TO 1
       KLB = LBS(I)*IRFLO/IDENOM
        LB(I) = MIN0(LBS(I),KLB)
       LB(ISAVE) = LB(ISAVE) + LB(I)
```

```
  001  CONTINUE
       CALL MNCF(M,N,II,IJ,KC,KU,LB,KY,NP,NL,INFEAS)
       DO 2 IMAX = 1,N
  004   IF ( IT(IMAX).EQ.0) GO TO 2
       IEX = 0
        DO 3 I = 1,N
C  MARK OUT OF CONSIDERATION ALL ARCS WITH ZERO FLOW
       IF ( KY(I).EQ. 0) IT(I) = 0
  003  IEX = IEX + IT(I)
C
       IF ( IEX.EQ.0) GO TO 35
       ITEMP = KU(IMAX)
       KU(IMAX) = 0
        KC(ISAVE) = -9999999
C  FIND MAX FLOW IN NETWORK WITH LINK IMAX OUT
       CALL MNCF(M,N,II,IJ,KC,KU,LB,KY,NP,NL,INFEAS)
       KC(ISAVE) = 0
       KU(IMAX) = ITEMP
       IF ( INFEAS.NE.1) GO TO 6
       LINK = IMAX
       GO TO 35
C
C  IF RESULTING FLOW IS GREATER THAN TEST FLOW SKIP THIS LINK.  IF LESS KEEP THE
C   LINK, AND IF EQUAL FIND OUT THE TOTAL NUMBER OF TRUCKS REQUIRED.
C  IF THEY ARE LESS FORGET IT.
  006  IF ( KY(ISAVE).GT.MFLO) GO TO 2
       CALL INPRD(KY,KC,N,NUTRQ)
       IF ( KY(ISAVE).LT.MFLO) GO TO 7
       IF(NUTRQ.LT.ITRQ) GO TO 2
  007  LINK = IMAX
       MFLO = KY(ISAVE)
       ITRQ = NUTRQ
  002  CONTINUE
  035   ZK = KU(LINK)
      IF (LINK.NE.0) GO TO 10116
      WRITE (6,10117)
10117 FORMAT(25H NO LINKS ARE VULNERABLE.)
      RETURN
10116   KU(LINK) = 0
        AK = KV(LINK)
        KOUT = AK/(TUPRHR*HRPRDA)+.5
       INTIME(LINK) = KPDS + KOUT
       A = ZK/FACTOR
       QK = KU(LINK)
       B = QK/FACTOR
       N1 = II(LINK)
       N2 = IJ(LINK)
      WRITE (6,9) KLINKS,NN(N1),NN(N2),A,B,INTIME(LINK)
  009 FORMAT(12H CUT NUMBER ,I2,25H REDUCED CAPACITY ON ARC ,A6,4H TO ,
     XA6, 6H FROM ,F7.0,4H TO  ,F7.0/15X,28H ARC WILL BE RESTORED ON DAY
     X,I3,1H.)
C  THE NEXT SECTION OF CODE IS TO TAKE OUT LINKS IN THE OPPOSITE DIRECTION IF
C  THEY EXIST
       IRFLO = 0
       IF ( IEX.EQ.999999) GO TO 365
       DO 36  I = 1,N
       M1 = II(I)
       M2 = IJ(I)
      IF( NN(M1).EQ.NN(N2).AND. NN(M2).EQ.NN(N1).AND.KV(I).EQ.KV(LINK))
     X  GO TO 37
  036  CONTINUE
       RETURN
  365   CONTINUE
       LINK = ITTMP
      RETURN
  037  ITTMP = LINK
       LINK = I
       LINK1 = I
       IEX = 999999
       GO TO 35
C
      END
```

```
$IBFTC PROFIL
      SUBROUTINE PROFIL(ISAVE,ISINK,IFLO,NTRKS,LBS,IDENOM,IEND,KSAVE)
       COMMON /LFLAG/ KINTRP,KCUT
       COMMON  /ORIG/M,N,KB(12),NIT,NOT,NN,NP,II,IJ,KC,KU,LB,KX,NL,KV
       COMMON  /LBUG/  BUGWRD
      DIMENSION II(1001),IJ(1001),KC(1001),KU(1001),LB(1001),KX(1001)
      DIMENSION NL( 501),NP( 501),NN( 501),KV(1001)
       DIMENSION  LBS(1001)
       DIMENSION  LOC(2)
       LOGICAL KINTRP,KCUT
       LOGICAL  BUGWRD
       JT = 1
  010  LB(ISAVE) = 0
       DO 1000 I = 1,N
       N2 = IJ(I)
        IF ( NN(N2).NE.ISINK) GO TO 1000
         KLB = LBS(I)*IFLO/IDENOM
       LB(I) = MINO(LBS(I),KLB)
 1000 CONTINUE
       LB(ISAVE) = IFLO
       CALL MNCF(M,N,II,IJ,KC,KU,LB,KX,NP,NL,INFEAS)
       IF ( INFEAS.NE.1) GO TO 1010
       WRITE (6,2010)
 2010  FORMAT ( 1H0,51HMIN DEMANDS CANNOT BE SATISFIED AT SOME DESTINATI
     XON   )
      IF (BUGWRD)
     XCALL        OUTPUT(II,IJ,KC,KU,LB,KX,NL,NP,INFEAS,KV)
       KCUT =.TRUE.
       RETURN
 1010  CALL INPRD(KX,KC,N,ITRQ)
      IF (BUGWRD) CALL OUTPUT(II,IJ,KC,KU,LB,KX,NL,NP,INFEAS,KV)
       CALL JADJ(ITRQ,IFLO,JTRQ,JFLO)
       IF ((JTRQ.GE.NTRKS).AND.(KINTRP))
     X  CALL INTRP(JTRQ,NTRKS,IFLO,
     X LTRQ,LFLO,ISAVE,IDENOM,LBS,ISINK)
       LTRQ = JTRQ
       LFLO = IFLO
       WRITE (6,2000) JTRQ,JFLO,ITRQ
 2000 FORMAT (1H , 5X, I6,10X,I8,15X,I9)
      IF ( IEND.NE. 0) GO TO 1020
      IF (IFLO.EQ.KSAVE) GO TO 1015
      IFLO = KSAVE
      GO TO 10
 1015  LFLO = 0
       LTRQ = 0
       RETURN
 1020 IF ( JT.EQ. 1) GO TO 1030
      RETURN
 1030 JT = 2
       IFLO = IFLO + 1
       GO TO 10
       END
$IBFTC INTRP
      SUBROUTINE INTRP(ITRQ,NTRKS,IFLO,LTRQ,LFLO,ISAVE,IDENOM,LBS,ISINK)
       COMMON  /RTRK/  IRFLO
       COMMON  /LBUG/  BUGWRD
       COMMON /LFLAG/  KINTRP,KCUT
       COMMON  /ORIG/M,N,KB(12),NIT,NOT,NN,NP,II,IJ,KC,KU,LB,KX,NL,KV
      DIMENSION II(1001),IJ(1001),KC(1001),KU(1001),LB(1001),KX(1001)
      DIMENSION NL( 501),NP( 501),NN( 501),KV(1001)
       DIMENSION  KCS(1001),KUS(1001),LBS(1001)
         DIMENSION  LOC(2)
       LOGICAL KINTRP,KCUT
       LOGICAL BUGWRD
       INTEGER  OLDFLO
       IF ( ITRQ.EQ.NTRKS) GO TO 2003
       IRFLO = 0
      NUFLO = LFLO+(IFLO-LFLO)*(NTRKS-LTRQ)/(ITRQ-LTRQ)
       IF ( NUFLO.EQ.LFLO) NUFLO = NUFLO + 1
  500  DO 1000 I = 1,N
        N2 = IJ(I)
       IF ( NN(N2).NE.ISINK) GO TO 1000
         KLB = LBS(I)*NUFLO/IDENOM
       LB(I) = MINO(KLB,LBS(I))
```

```
 1000 CONTINUE
        NUTRK = 0
        LB(ISAVE) = NUFLO
        CALL MNCF(M,N,II,IJ,KC,KU,LB,KX,NP,NL,INFEAS)
        CALL IMPRD(KX,KC,N,NUTRK)
        CALL JADJ(NUTRK,NUFLO,JTRQ,JFLO)
        IRFLO = NUFLO
        IF ( JTRQ.GE.NTRKS- 50) GO TO 2003
        WRITE (6,2001) JTRQ,JFLO,NUTRK
 2001   FORMAT ( 1H ,5X,I6,10X,I8,15X,I9)
         NUTEM1 = NUFLO + (IFLO-NUFLO)/3
        NUTEM2 = NUFLO + 1
        NUFLO = MAXO(NUTEM1,NUTEM2)
        GO TO 500
 2003   WRITE ( 6,2000) JTRQ,JFLO,NUTRK
 2000 FORMAT ( 1H , 5X, I6,3H***,7X,I8,15X,I9)
        KINTRP= .FALSE.
        RETURN
        END
$IBFTC QKINPT
      SUBROUTINE QKINPT
       COMMON  /FACTR/  FACTOR
       COMMON  /FACT/  FINCOM,FSTOP, TNSTRK,HRPRDA,TUPRHR
       COMMON  /ORIG/M,N,KB(12),NIT,NOT,NN,NP,II,IJ,KC,KU,LB,KX,NL,KV
      DIMENSION NL( 501),NP( 501),NN( 501)
      DIMENSION II(1001),IJ(1001),KC(1001),KU(1001),LB(1001),KX(1001)
      DIMENSION KA(12),KV(1001)
C   INPUT TAPE
      NIT = 5
C   OUTPUT TAPE
      NOT = 6
C   MAXIMUM ARCS
        MAXN = 1000
C   MAXIMUM NODES
       MAXM = 500
       FACTOR = 1.0/(TNSTRK*TUPRHR*HRPRDA)
C   PREPARE TO READ DATA
  100 READ (NIT,90) (KA(I),I=1,12)
      CALL NUM(KA,36HPAUSE SAVE  READY ARCS  PUNCHAPUNCHN,KKK)
      IF(KKK.EQ.2 .OR. KKK.EQ.3)  GO TO 102
      WRITE(NOT,91) (KA(I),I=1,12)
      IF (KKK.GT.6) GO TO 100
      IF (KKK.EQ.4) GO TO 200
      IF (KKK.EQ.5)  GO TO 160
      IF (KKK.EQ.6)  GO TO 170
  180 CALL EXIT
C  PUNCHING
  160 PUNCH 79,(KA(I),I=1,11)
      DO 161 J = 1,N
        N1 = II(J)
        N2 = IJ(J)
        PUNCH 78,NN(N1),NN(N2),KC(J),KU(J),LB(J),KX(J)
  161 CONTINUE
      PUNCH 77
      GO TO 100
  170 PUNCH 76
      DO 171 I = 1,M
        PUNCH 75,NN(I),NP(I)
  171 CONTINUE
      PUNCH 77
      GO TO 100
  102 WRITE (NOT,97) (KA(I),I=1,12)
      IF (KKK.EQ.2)  GO TO 101
C  ZERO NODE PRICES
  110 DO 111 I=1,MAXM
        NP(I) = 0
  111 CONTINUE
C  READ TITLE CARD
  101 READ (NIT,90)  (KB(I),I=1,12)
      WRITE (NOT,91) (KB(I),I=1,12)
      IF (KKK.EQ.2) GO TO 501
      GO TO 100
```

```
C  READ ARCS
  200 M = 0
       WRITE (6,10126)
10126 FORMAT(5H0ARCS16X,4HCOST6X,5HUPPER6X,5HLOWER,          10X,4HTIME/
     X 5H FROM,3X,2HTO,46X,9HTO REPAIR/1X)
      DO 201 I=1,10000
   92    FORMAT(3A6,2X,4I10,I3)
         READ(NIT,92) KD,II(I),IJ(I),KC(I),KU(I),JLBT ,KX(I),KV(I)
         CALL NUM(KD, 24HEND   NODES COMPUT      ,KKK)
         IF (KKK.NE.4 )  GO TO 209
        ALBT = JLBT
        LB(I) = ALBT*FACTOR
        WRITE (6,10135)II(I),IJ(I),KC(I),KU(I),LB(I),      KV(I)
10135 FORMAT(2A7,3I11,          11X,I3)
  201 CONTINUE
  209 WRITE(NOT,93) KD,II(I),IJ(I)
  210 N = I-1
  220 IF(KKK.NE.5) GO TO 300
  299 WRITE (NOT,96)
      CALL EXIT
C  NUMBER THE NODES
  300 DO 301 L =1,N
         NN(M+1) = II(L)
         II(L) = NODENO(II(L))
         M = MAXO(M,II(L))
  301 CONTINUE
      DO 302 L=1,N
         NN(M+1) = IJ(L)
         IJ(L) = NODENO(IJ(L))
         M = MAXO(M,IJ(L))
  302 CONTINUE
C  SKIP NODE READING IF NODE CONTROL WASN'T READ
      IF (KKK.NE.2) GO TO 500
C  NODE READING
  400 READ (NIT,81) KD,KE,NPK
      CALL NUM(KD,   18HEND          COMPUT,KKK)
      IF (KKK.NE.2) GO TO 410
      N1 = NODENO(KE)
      IF ( N1.GT.M ) GO TO 400
  420 NP(N1) = NPK
      GO TO 400
  410 WRITE (NOT,82) KD,KE
      IF (KKK.EQ.4) GO TO 299
      GO TO 500
C  READER
  500 IF (KKK.EQ.3) GO TO 600
  501 READ (NIT,83) (KA(I),I=1,8)
      CALL NUM(KA,24HALTER COMPUT      NODES ,KKK)
      IF (KKK.EQ.1 .OR. KKK.EQ.3 )  GO TO 515
      WRITE (NOT,93) KA(1),KA(2),KA(3)
  560 IF (KKK.EQ.2) GO TO 600
      IF (KKK.EQ.5) GO TO 501
      IF (KKK.EQ.4) GO TO 400
  515 WRITE (NOT,84) (KA(I),I=1,8)
      KA(4) = MAXO(KA(4),1)
      N1 = NODENO(KA(2))
      NN(N1) = KA(2)
      M = MAXO(M,N1)
      N2 = NODENO(KA(3))
      NN(N2) = KA(3)
      M = MAXO(M,N2)
      DO 520 I =1,N
        IF (N1.NE.II(I).OR.N2.NE.IJ(I) ) GO TO 520
        KA(4) = KA(4) - 1
        IF (KA(4).EQ.0) GO TO 530
  520 CONTINUE
      WRITE (NOT,85)
      N = N+1
      I = N
  530 II(I) = N1
      IJ(I) = N2
      KC(I) = KA(5)
      KU(I) = KA(6)
      LB(I) = KA(7)
      KX(I) = KA(8) + KX(I)
```

```
      GO TO 501
  600 WRITE (NOT,94) N ,M
C   TESTS
      IF (N.LE.MAXN) GO TO 601
      WRITE (NOT,95)
      CALL EXIT
  601 IF (M.LE.MAXM) GO TO 320
      WRITE (NOT,98)
      CALL EXIT
C   LOOK FOR DEAD NODES
  320 DO 303 I=1,M
        DO 304 L =1,N
          IF(II(L).EQ.I) GO TO 305
  304   CONTINUE
        WRITE(NOT,99) NN(I)
  305   DO 306 L =1,N
          IF(IJ(L).EQ.I) GO TO 303
  306   CONTINUE
        WRITE(NOT,80) NN(I)
  303 CONTINUE
      DO 611 I=1,M
        NL(I) = 0
  611 CONTINUE
C   CALCULATE FLOWS
      DO 612 J=1,N
        IF (KU(J).GE.LB(J) ) GO TO 609
        LUP = KU(J)
        KU(J) = LB(J)
        LB(J) = LUP
  609   N1 = II(J)
        N2 = IJ(J)
        NL(N1) = NL(N1) - KX(J)
        NL(N2) = NL(N2) + KX(J)
  612 CONTINUE
      DO 605  I=1,M
        IF (NL(I).NE.0) WRITE (NOT,86) NN(I),NL(I)
  605 CONTINUE
C   DO THE ALGORITHM
       RETURN
   75 FORMAT(6X,A6,I18)
   76 FORMAT(6HNODES )
   77 FORMAT(6HEND    )
   78 FORMAT(6X,2A6,2X,4I10)
   79 FORMAT(6HARCS  ,11A6)
   80    FORMAT(24H NO ARC ENDS AT NODE    A6)
   81 FORMAT(2A6,I18)
   82 FORMAT(1X,2A6)
   83 FORMAT(3A6,I2,4I10)
   84 FORMAT(1X,3A6,I4,4I10)
   85 FORMAT(26H AUGMENT ARCS BY ABOVE ARC)
   86    FORMAT(6H NODE ,A6,28H NON-CONSERVATIVE, NET FLOW=I14)
   90 FORMAT(12A6)
   91 FORMAT(1H1,12A6)
   93 FORMAT(1X,3A6)
   94 FORMAT(18H THIS PROBLEM HAS I12,9H ARCS AND,I12,6H NODES )
   95 FORMAT(14H TOO MANY ARCS)
   96 FORMAT(36H ILLEGAL DATA CARD OR CONTROL CARD)
   97 FORMAT(1H0,12A6)
   98 FORMAT(15H TOO MANY NODES)
   99    FORMAT(24H NO ARC BEGINS AT NODE  A6)
      END
$IBFTC MNCF
      SUBROUTINE MNCF(NODES,ARCS,I,J,COST,HI,LO,FLOW,PI,NA,INFEAS)
      INTEGER NODES,ARCS,I,J,COST,HI,LO,FLOW,PI,NA,INFEAS
      DIMENSION I(3000),J(3000),COST(3000),HI(3000),LO(3000),FLOW(3000)
       DIMENSION  LOC(2)
      DIMENSION PI(1000),NA(1000)
C
C
C     DEFINITION OF CALLING SEQUENCE
C
C
C   NAME    USE
C
```

```
C    NODES   NUMBER OF NODES
C    ARCS    NUMBER OF ARCS
C    I       LIST OF FIRST NODES
C    J       LIST OF SECOND NODES
C    COST    UNIT COST OF FLOW ON ARCS
C    HI      UPPER BOUNDS FOR ARCS
C    LO      LOWER BOUNDS FOR ARCS
C    FLOW    AMOUNT OF FLOW IN ARCS
C    PI      NODE PRICES
C    NA      SCRATCH LIST FOR NODE LABELING
C    INFEAS   FLAG DENOTING THE CONDITION OF OUTPUT
C
C  BEGIN
      INTEGER A,AA,N,SRC,SNK,DEL,INF,C,AOK,COK,N1,N2,INC,LABEL
      INF = 34359738367
      AOK = 0
C LOOK FOR AN OUT OF KILTER ARC
      DO 90 AA=1, ARCS
  100   N1 =  I(AA)
        N2 =  J(AA)
        C  =  COST(AA) +PI(N1) - PI(N2)
C  EXIT IF SENSE SWITCH 5 IS DOWN
   30   CALL SSWTCH(5,LABEL)
        IF (LABEL.EQ.2) GO TO 40
        INFEAS = 2
        RETURN
   40   IF(FLOW(AA).LT.LO(AA).OR.(C.LT.0.AND.FLOW(AA).LT.HI(AA)))GOTO 50
        IF(FLOW(AA).GT.HI(AA).OR.(C.GT.0.AND.FLOW(AA).GT.LO(AA)))GOTO 60
   90 CONTINUE
C  NO OUT OF KILTER ARCS LEFT
      INFEAS = 0
      RETURN
C    OUT OF KILTER ARC FOUND
   50   SRC = J(AA)
        SNK = I(AA)
      LABEL = +AA
        GO TO 200
   60   SRC = I(AA)
        SNK = J(AA)
      LABEL = -AA
C  SAVE LABELS IF LAST OPERATION WAS INCREASING NODE PRICES ON THIS ARC
  200 IF ( AA.EQ.AOK.AND.NA(SRC).NE.0)   GO TO 205
      DO 201 N = 1,NODES
        NA(N) = 0
  201 CONTINUE
      AOK =AA
  205 COK = C
      NA(SRC) = LABEL
C LABEL
  210 LABEL = 0
      DO 250 A =1,ARCS
        N1 = I(A)
        IF (N1.LT.0) GO TO 250
        N2 = J(A)
        IF  (NA(N1).EQ.0.AND.NA(N2).EQ.0) GO TO 250
        IF (NA(N1).NE.0.AND.NA(N2).NE.0)  GO TO 245
        C = COST(A) + PI(N1) - PI(N2)
        IF (NA(N1).EQ.0)  GO TO 220
        IF(FLOW(A).GE.HI(A).OR.(FLOW(A).GE.LO(A).AND.C.GT.0)) GO TO 245
        NA(N2) = A
        GO TO 240
  220   IF(FLOW(A).LE.LO(A).OR.(FLOW(A).LE.HI(A).AND.C.LT.0)) GO TO 245
        NA(N1) = -A
  240   LABEL = 1
C  NODE LABELED, TEST FOR BREAKTHRU
        IF (NA(SNK).NE.0) GO TO 260
  245   I(A) = -N1
  250 CONTINUE
C  GO BACK AND DO MORE LABELING IF SOME NODE WAS LABELED ON LAST PASS
      IF (LABEL.NE.0)  GO TO 210
C  RESTORE POSITIVE SIGNS TO FIRST NODE LIST
  260 DO 270 A = 1,ARCS
        I(A) = IABS(I(A))
  270 CONTINUE
```

```
C   IF NO LABELING DONE ON LAST PASS, GO TO INCEASE PIE
       IF (LABEL.EQ.0) GO TO 400
C   BREAKTHRU,  FIND THE INCREMENT
  300 INC = INF
C   FOLLOW PATH BACK FROM SOURCE
       N     = SRC
  310 A = IABS(NA(N))
       IF (NA(N).LT.0)   GO TO 315
       N2 = I(A)
       C = COST(A) - PI(N) + PI(N2)
       IF (C.GT.0) INC = MIN0(INC,LO(A)-FLOW(A) )
       IF (C.LE.0) INC = MIN0(INC,HI(A)-FLOW(A) )
       GO TO 340
  315 N2 = J(A)
       C = COST(A) + PI(N) - PI(N2)
  320 IF (C.LT.0) INC = MIN0(INC,FLOW(A)-HI(A) )
       IF (C.GE.0) INC = MIN0(INC,FLOW(A)-LO(A) )
  340 N  = N2
       IF   (N.NE.SRC) GO TO 310
C   INCREMENT ARCS
  350 A = IABS(NA(N))
       IF (NA(N).LT.0)   GO TO 360
       FLOW(A) = FLOW(A) + INC
       N = I(A)
       GO TO 370
  360 FLOW(A) = FLOW(A) - INC
       N = J(A)
  370 IF ( N.NE.SRC ) GO TO 350
C   FLOW INCREMENTED, RETURN TO KILTER TEST
       NA(N) = 0
       GO TO 100
C   CHANGE PI
  400 DEL = INF
C   FIND INCREMENT
       DO 420 A=1,ARCS
         N1 = I(A)
         N2 = J(A)
         IF    (NA(N1).EQ.0 .AND. NA(N2).EQ.0)  GO TO 420
         IF    (NA(N1).NE.0 .AND. NA(N2).NE.0)  GO TO 420
         C   = COST(A) + PI(N1) - PI(N2)
         IF (NA(N2).EQ.0 .AND. FLOW(A).LT.HI(A) ) DEL=MIN0(DEL,C)
         IF (NA(N2).NE.0 .AND. FLOW(A).GT.LO(A) ) DEL=MIN0(DEL,-C)
  420 CONTINUE
       IF (DEL.NE.INF)   GO TO 430
       IF  (FLOW(AA).EQ.LO(AA)  .OR.  FLOW(AA).EQ.HI(AA))  GO TO 425
C   INFEASIBLE SOLUTION
       INFEAS = 1
       RETURN
C   INCREASE PI
  425 DEL = IABS(COK)
  430 DO 450 N =1,NODES
         IF (NA(N).EQ.0)   PI(N) = PI(N) + DEL
  450 CONTINUE
C   GO BACK TO KILTER TEST
       GO TO 100
       END
$IBFTC OUTPUT
       SUBROUTINE OUTPUT(II,IJ,KC,KU,LB,KX,NL,NP,INFEAS,KV)
        COMMON  /RTIME/  INTIME(1001)
        COMMON  /LBUG/  BUGWRD
        COMMON  /MARKR/ IT(1001)
        COMMON  /ORIG/M,N,KB(12),NIT,NOT,NN
       DIMENSION NL(1001),NP(1001),NN(1001)
       DIMENSION II(3001),IJ(3001),KC(3001),KU(3001),LB(3001),KX(3001)
       DIMENSION KV(1001)
        LOGICAL BUGWRD
CARCS                COST       UPPER       LOWER           X           CBAR
CAAAAAA AAAAAA       ...I        ...I        ...I        ...I           ...I
C
C     KILTER NUMBER
C           I
C
   92 FORMAT(5H0ARCS16X,4HCOST6X,5HUPPER5X,5HLOWER 7X,1HX 8X,4HCBAR3X,
     1          13HKILTER NUMBER,5X,4HVULN,4X,6HINTIME,2X,2HIT/1X)
   93 FORMAT(2A7,4I10,I8 ,I7,5X,I3,4X,I6,2X,I3 )
```

```
  098 FORMAT(2A7,4I10,I8 ,I7,5X,I3,4X,I6,2X,I3 ,3HCUT)
      WRITE (NOT,92)
      NOUT =0
      KILTER = 0
      MA = 0
      MB = 0
      DO 901 J=1,N
        N1 = II(J)
        N2 = IJ(J)
        KCK = KC(J) + NP(N1) - NP(N2)
        IF (KCK.GT.0) KIL = IABS(KX(J)-LB(J))
        IF (KCK.LT.0) KIL = IABS(KX(J)-KU(J))
        IF (KCK.EQ.0) KIL = MAX0(MAX0(0,KX(J)-KU(J)),LB(J)-KX(J))
        IF (KIL.NE.0) NOUT = NOUT + 1
        KILTER = KILTER + KIL
        CALL MADD(MA,MB,KC(J),KX(J))
        IF (INFEAS.EQ.1.AND.(NL(N1)*NL(N2)).EQ.0.AND.(NL(N1)+NL(N2))
    C   .NE.0)  GO TO 910
       IF (KX(J).GT.0)
     XWRITE(NOT,93)     NN(N1),NN(N2),KC(J),KU(J),LB(J),KX(J)
     X ,KCK,KIL,KV(J),INTIME(J),IT(J)
        GO TO 901
  910   WRITE (NOT,98) NN(N1),NN(N2),KC(J),KU(J),LB(J),KX(J)
     X ,KCK,KIL,KV(J),INTIME(J),IT(J)
  901 CONTINUE
      ME = 1E+09
      MD = MB/ME
      MB = MB -MD*ME
      M1 = MA + MD
      IF ( M1.EQ.0.OR.(ISIGN(1,M1).EQ.ISIGN(1,MB)) ) GO TO 922
      MB = MB + ISIGN(ME,M1)
      M1 = M1 - ISIGN(1,M1)
  922 IF (M1.EQ.0) M1 = ISIGN(M1,MB)
      M2 = MB + ISIGN(ME,MB)
       CALL INPRD(KX,KC,N,MT)
      WRITE(NOT,94) NOUT,KILTER,MT,M1,M2
   94 FORMAT(1H0,I4,41H ARCS OUT OF KILTER, TOTAL KILTER NUMBER=I8,
     X2X,17HTRUCKS REQUIRED= ,3(I6,1X))
      RETURN
       WRITE (NOT,95)
  095  FORMAT ( 1H1,12H NODE PRICES/1X)
      MINI = NP(1)
      DO 940 I = 2,M
        MINI = MIN0(NP(I),MINI)
  940 CONTINUE
      DO 902 I=1,M
        NP(I) = IABS(NP(I) - MINI   )
        IF (INFEAS.NE.1 .OR. NL(I).EQ.0 )  GO TO 903
        WRITE (NOT,99) NN(I),NP(I)
   99   FORMAT(A7,I13,8H LABELED)
        GO TO 902
  903  WRITE (NOT,96) NN(I),NP(I)
   96   FORMAT(A7,I13)
  902 CONTINUE
      RETURN
      END
$IBFTC NODENO
      FUNCTION NODENO  (III)
       COMMON  /ORIG/M,N,KB(12),NIT,NOT,NN(1000)
      NODENO = M + 1
      IF (M.EQ.0) RETURN
      DO 1001 I = 1,M
        IF (III.EQ.NN(I) ) NODENO = I
 1001 CONTINUE
      RETURN
      END
$IBFTC NUM
      SUBROUTINE NUM (KA,KB,L)
      DIMENSION KB(1)
      DO 1 I = 1,100
        IF (KB(I).EQ.-34359738367)   GO TO 2
        IF(KA.EQ.KB(I)) GO TO 2
    1 CONTINUE
      I = 1
    2 L=I
      RETURN
      END
```

```
$IBMAP MADD
       ENTRY   MADD
MADD   SAVE    4
       LAC     MADD,4
       LDQ*    4,4
       MPY*    5,4
       STO     MA
       STQ     MB
       CLA     MB
       ADD*    3,4
       STO*    3,4
       ARS     35
       ADD     MA
       ADD*    2,4
       STO*    2,4
       RETURN  MADD
MA     BSS     1
MB     BSS     1
       END
$IBFTC INPRD
       SUBROUTINE INPRD(MZA,NZA,N,ITEMP)
       DIMENSION MZA(1),NZA(1)
       ITEMP = 0
       DO 1 I = 1,N
  001  ITEMP = ITEMP + MZA(I)*NZA(I)
       RETURN
       END
$IBFTC JADJ
      SUBROUTINE JADJ(ITRQ,IFLO,JTRQ,JFLO)
       COMMON  /FACT/  FINCOM, FSTOP,TNSTRK,HRPRDA,TUPRHR
       A = ITRQ
       ATRUCK = A*FSTOP/FINCOM
C  ATRUCK IS THE ACTUAL NUMBER OF TRUCKS REQUIRED
       B = IFLO
       BFLO = B*TUPRHR*HRPRDA*TNSTRK
       JTRQ = ATRUCK
       JFLO = BFLO
       RETURN
      END
$ENTRY         CIRSFA
```

DOCUMENT CONTROL DATA

| | |
|---|---|
| 1 ORIGINATING ACTIVITY<br>THE RAND CORPORATION | 2a REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED<br>2b GROUP |

3. REPORT TITLE

AN INTERDICTION MODEL OF HIGHWAY TRANSPORTATION

4. AUTHOR(S) (Last name, first name, initial)

Durbin, Eugene P.

| 5. REPORT DATE | 6a. TOTAL NO. OF PAGES | 6b. NO. OF REFS. |
|---|---|---|
| May 1966 | 34 | - - - |

| 7. CONTRACT or GRANT NO. | 8. ORIGINATOR'S REPORT NO. |
|---|---|
| AF 49(638)-1700 | RM-4945-PR |

| 9a AVAILABILITY/LIMITATION NOTICES | 9b SPONSORING AGENCY |
|---|---|
| DDC 1 | United States Air Force<br>Project RAND |

10. ABSTRACT

Description of a computer program to evaluate the capability of transportation networks to deliver supplies, as road segments or arcs of the network are successively destroyed and repaired. The program, written in FORTRAN IV, can be adapted for any of several large-scale computers. Required inputs are a description of the considered transportation road system and the cargo-carrying vehicles using it. The program furnishes a profile of maximum cargo flow as a function of the number of vehicles available to the system, then destroys the link in the network that reduces cargo flow rate most severely. These steps are repeated until network flow is stopped or predesignated links destroyed. The program then steps to the next "period," restores service to all previously destroyed links now repaired, and repeats the process of profile generation and link removal. The program will accept a network up to 1000 links.

11 KEY WORDS

Interdiction
Transport
Highways
Models
Trucks
Roads
Airpower
Networks
Tactical warfare
Logistics