AN INTERDICATION MODEL OF HIGHWAY TRANSPORTATION

Eugene P. Durbin

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation

SANTA MONICA • CALIFORNIA
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.
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.

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.


**Wollmer, op. cit.
## CONTENTS

PREFACE ......................................................... iii
SUMMARY ......................................................... v

Section
  I.  INTRODUCTION ............................................. 1
  II. DESCRIPTION OF THE MODEL ............................... 4
  III. PROGRAM SEQUENCE OF OPERATION ....................... 7
  IV.  INPUT DESCRIPTION AND PREPARATION ................... 8

Appendix
  A. INTERDICTION PROGRAM OUTPUT ............................ 15
  B. INTERDICTION PROGRAM ................................. 17
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 \((i,j)\)

\( l_{ij} \) = 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

\[
\text{maximize } x_{n+1,0}^{n+1,0}
\]

subject to

\[
\begin{align*}
\sum_i x_{ij} - \sum_j x_{ji} &= 0 \text{ for each } j \\
\ell_{ij} &\leq x_{ij} \leq u_{ij} \text{ for all arcs } (i,j)
\end{align*}
\]

where

\( \ell_{ij} = d_i \) for all destinations \( i \), and
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:

\[
\text{(4)} \quad \min \sum_{ij} t_{ij} x_{ij},
\]

with

\[
\begin{align*}
\sum_{i} x_{ij} - \sum_{i} x_{ji} &= 0 \text{ for each } j \\
\ell_{ij} &\leq x_{ij} \leq u_{ij} \text{ for all arcs } (i,j),
\end{align*}
\]

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 "FLOW TOTALLY STOPPED ON 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 SOME DESTINATION CANNOT 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 (*** map) 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 \( K \cdot \text{FSTOP}/\text{FINCOM} \) vehicles are required to deliver \( F \cdot \text{TNSTRK}\cdot \text{TUPRHR}\cdot \text{HRPRDA} \) 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.
The word "READY"

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

The word "ARCS"

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, $l_{ij}$ (I)
- Col 61-63 Air Time of Arc, $r_{ij}$ (I)

The word "END"

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

c = cost (time units to transition arc)
u = upper capacity (vehicles per time unit)
Double line indicates invulnerable arc.

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

<table>
<thead>
<tr>
<th>ARCS</th>
<th>COST</th>
<th>UPPER</th>
<th>LOWER</th>
<th>TIME TO REPAIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARTIFS</td>
<td>A</td>
<td>0</td>
<td>999</td>
<td>-0</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>4</td>
<td>30</td>
<td>-0</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>15</td>
<td>8</td>
<td>-0</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>11</td>
<td>15</td>
<td>-0</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>20</td>
<td>30</td>
<td>-0</td>
</tr>
<tr>
<td></td>
<td>F</td>
<td>8</td>
<td>15</td>
<td>-0</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>11</td>
<td>10</td>
<td>-0</td>
</tr>
<tr>
<td>ARTIFS</td>
<td>-0</td>
<td>999</td>
<td>4</td>
<td>-0</td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COMPUTE THIS PROBLEM HAS 10 ARCS AND 8 NODES

DAY 1
FLOW PROFILE PRIOR TO FIRST CUT

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY IN INVENTORY INVENTORY</th>
<th>TRUCKS</th>
<th>TONS/DAY THROUGHPUT</th>
<th>TRUCKS</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>366</td>
<td>720</td>
<td>ARTIFS</td>
<td>104</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>345</td>
<td>810</td>
<td>ARTIFS</td>
<td>205</td>
<td>205</td>
<td></td>
</tr>
<tr>
<td>366***</td>
<td>1080</td>
<td>ARTIFS</td>
<td>362</td>
<td>362</td>
<td></td>
</tr>
<tr>
<td>626</td>
<td>1440</td>
<td>ARTIFS</td>
<td>678</td>
<td>678</td>
<td></td>
</tr>
<tr>
<td>678</td>
<td>1530</td>
<td>ARTIFS</td>
<td>1046</td>
<td>1046</td>
<td></td>
</tr>
<tr>
<td>1105</td>
<td>2250</td>
<td>ARTIFS</td>
<td>1105</td>
<td>1105</td>
<td></td>
</tr>
</tbody>
</table>

***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.

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY IN INVENTORY INVENTORY</th>
<th>TRUCKS</th>
<th>TONS/DAY THROUGHPUT</th>
<th>TRUCKS</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>195</td>
<td>270</td>
<td>ARTIFS</td>
<td>200</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>310</td>
<td>360</td>
<td>ARTIFS</td>
<td>310</td>
<td>310</td>
<td></td>
</tr>
<tr>
<td>361</td>
<td>630</td>
<td>ARTIFS</td>
<td>471***</td>
<td>471***</td>
<td></td>
</tr>
<tr>
<td>471</td>
<td>810</td>
<td>ARTIFS</td>
<td>530</td>
<td>530</td>
<td></td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY IN INVENTORY INVENTORY</th>
<th>TRUCKS</th>
<th>TONS/DAY THROUGHPUT</th>
<th>TRUCKS</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>180</td>
<td>ARTIFS</td>
<td>195</td>
<td>195</td>
<td></td>
</tr>
<tr>
<td>206</td>
<td>270</td>
<td>ARTIFS</td>
<td>206</td>
<td>206</td>
<td></td>
</tr>
<tr>
<td>258</td>
<td>450</td>
<td>ARTIFS</td>
<td>310</td>
<td>310</td>
<td></td>
</tr>
<tr>
<td>413</td>
<td>720</td>
<td>ARTIFS</td>
<td>530</td>
<td>530</td>
<td></td>
</tr>
</tbody>
</table>
### DAY 2

**FLOW PROFILE PRIOR TO FIRST CUT**

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY</th>
<th>TRUCKS</th>
<th>TONS/DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN INVENTORY</td>
<td>THROUGHPUT</td>
<td>ACTIVE</td>
<td>THROUGHPUT</td>
</tr>
<tr>
<td>200</td>
<td>630</td>
<td>161</td>
<td>630</td>
</tr>
<tr>
<td>306</td>
<td>720</td>
<td>184</td>
<td>720</td>
</tr>
<tr>
<td>400***</td>
<td>1170</td>
<td>299</td>
<td>1170</td>
</tr>
<tr>
<td>536</td>
<td>1260</td>
<td>322</td>
<td>1260</td>
</tr>
<tr>
<td>575</td>
<td>1350</td>
<td>345</td>
<td>1350</td>
</tr>
<tr>
<td>885</td>
<td>1890</td>
<td>531</td>
<td>1890</td>
</tr>
<tr>
<td>988</td>
<td>2070</td>
<td>593</td>
<td>2070</td>
</tr>
</tbody>
</table>

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

**CUT NUMBER 1 REDUCED CAPACITY ON ARC B TO D FROM 1350 TO 0.**

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY</th>
<th>TRUCKS</th>
<th>TONS/DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN INVENTORY</td>
<td>THROUGHPUT</td>
<td>ACTIVE</td>
<td>THROUGHPUT</td>
</tr>
<tr>
<td>103</td>
<td>180</td>
<td>62</td>
<td>180</td>
</tr>
<tr>
<td>199</td>
<td>270</td>
<td>93</td>
<td>270</td>
</tr>
<tr>
<td>206</td>
<td>360</td>
<td>124</td>
<td>360</td>
</tr>
<tr>
<td>258</td>
<td>450</td>
<td>155</td>
<td>450</td>
</tr>
<tr>
<td>310</td>
<td>540</td>
<td>186</td>
<td>540</td>
</tr>
<tr>
<td>413</td>
<td>720</td>
<td>264</td>
<td>720</td>
</tr>
</tbody>
</table>

**CUT NUMBER 2 REDUCED CAPACITY ON ARC A TO C FROM 720 TO 0.**

**FLOW TOTALLY STOPPED ON DAY 2.**

### DAY 3

**FLOW PROFILE PRIOR TO FIRST CUT**

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY</th>
<th>TRUCKS</th>
<th>TONS/DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN INVENTORY</td>
<td>THROUGHPUT</td>
<td>ACTIVE</td>
<td>THROUGHPUT</td>
</tr>
<tr>
<td>306</td>
<td>720</td>
<td>184</td>
<td>720</td>
</tr>
<tr>
<td>345</td>
<td>810</td>
<td>207</td>
<td>810</td>
</tr>
<tr>
<td>460***</td>
<td>1080</td>
<td>276</td>
<td>1080</td>
</tr>
<tr>
<td>626</td>
<td>1440</td>
<td>376</td>
<td>1440</td>
</tr>
<tr>
<td>678</td>
<td>1530</td>
<td>407</td>
<td>1530</td>
</tr>
<tr>
<td>1046</td>
<td>2160</td>
<td>628</td>
<td>2160</td>
</tr>
<tr>
<td>1105</td>
<td>2290</td>
<td>663</td>
<td>2290</td>
</tr>
</tbody>
</table>

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

**CUT NUMBER 1 REDUCED CAPACITY ON ARC B TO D FROM 1350 TO 0.**

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY</th>
<th>TRUCKS</th>
<th>TONS/DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN INVENTORY</td>
<td>THROUGHPUT</td>
<td>ACTIVE</td>
<td>THROUGHPUT</td>
</tr>
<tr>
<td>155</td>
<td>270</td>
<td>93</td>
<td>270</td>
</tr>
<tr>
<td>206</td>
<td>360</td>
<td>124</td>
<td>360</td>
</tr>
<tr>
<td>310</td>
<td>540</td>
<td>186</td>
<td>540</td>
</tr>
<tr>
<td>361</td>
<td>630</td>
<td>217</td>
<td>630</td>
</tr>
<tr>
<td>471***</td>
<td>810</td>
<td>283</td>
<td>810</td>
</tr>
<tr>
<td>530</td>
<td>900</td>
<td>318</td>
<td>900</td>
</tr>
</tbody>
</table>

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

**CUT NUMBER 2 REDUCED CAPACITY ON ARC B TO E FROM 2700 TO 0.**

<table>
<thead>
<tr>
<th>TRUCKS</th>
<th>TONS/DAY</th>
<th>TRUCKS</th>
<th>TONS/DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN INVENTORY</td>
<td>THROUGHPUT</td>
<td>ACTIVE</td>
<td>THROUGHPUT</td>
</tr>
<tr>
<td>103</td>
<td>180</td>
<td>62</td>
<td>180</td>
</tr>
<tr>
<td>199</td>
<td>270</td>
<td>93</td>
<td>270</td>
</tr>
<tr>
<td>206</td>
<td>360</td>
<td>124</td>
<td>360</td>
</tr>
<tr>
<td>258</td>
<td>450</td>
<td>155</td>
<td>450</td>
</tr>
<tr>
<td>310</td>
<td>540</td>
<td>186</td>
<td>540</td>
</tr>
<tr>
<td>413</td>
<td>720</td>
<td>248</td>
<td>720</td>
</tr>
</tbody>
</table>
INTERDICTION PROGRAM

CIRSEA

COMMON /RTRK/ IRFLO
COMMON /MARKR/ IT(1001)
COMMON /FACTR/ FACTOR
COMMON /RTIME/ Intime(1001)
COMMON /NODRE/ ISINK
COMMON /DENOM/ IDENOM
COMMON /ORIG/@M, /*KB(12)* INIT NOT NN, NP, II, IJ, KU, LB, KX, NL, KV
COMMON /FACT/ FINCOM, FSTOP, TNSTRK, HRPDA, TUPRHR
COMMON /KDAY/ KPD$L, LINK, LINK1
COMMON /LBUG/ BUGWRD

DIMENSION @I(1001), IJ(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, KU(1), KUS(1)
LOGICAL BUGWRD

CRAO (5.2)

READ (5,2) NTRks, NPDS, NLINKS, NPPTS, IORuCE, ISINK, BUGWRD
READ (15,20) TNSTRK, FINCOM, FSTOP, HRPDA, TUPRHR

20 FORMAT (3F5.0/2F5.0)

WRITE (6,3) NTRKs, NPDS, NLINKS, NPPTS, IORuCE, ISINK, BUGWRD

X TNSTRK, FINCOM, FSTOP, HRPDA, TUPRHR

C INITIALIZATION
CALL QKINPT

SAVE THE ORIGINAL DATA
DO 5000 I = 1,N
KUS(I) = KU(I)
5000 LBS(I) = LB(I)

HERE FIND THE DESTINATION LINK.
DO 5002 I = 1,N
N2 = I, J(I)
IF (NN(N2) .EQ. IORUC) GO TO 5002
5002 CONTINUE

ISAVE = I
FIND THE SUM OF THE DEmand RATIOs
IDENOM = 0
DO 5003 I = 1,N
N2 = I, J(I)
IF (NN(N2) .NE. ISINK) GO TO 5003
IDENOM = IDENOM + LB(I)
5003 CONTINUE

IDENOM = MAX(1, IDENOM)

MAIN ROUTINE

ONCE FOR EACH DAY
DO 999 KPOS = 1, NPDS
KCUT = .FALSE.
THE NETWORK ISNT CUT YET
WRITE (6,1990) KPOS
1990 FORMAT (INI., 23X, 7HD A Y .12 / I)
WRITE (6,2001)

FLOW PROFILE PRIOR TO FIRST CUT
RESTORE ALL REPAIRED LINKS
DO 5004 I = 1, N
IF (INTIME(I) .LE. KPDS) KUS(I) = KU(I)
5004 LBS(I) = 0
NLINK1 = NLINKS + 1

NOW BEGIN THE STRIKES FOR THIS CURRENT DAY
DO 999 KLINKS = 1, NLINK1
FIND MAX FLOW
KU(ISAVE) = KU(ISAVE)
LBS(ISAVE) = 0
KCSI(SAVE) = -9999999
CALL MNCF(M, N, IJ, KU, LB, KX, NP, NL, INFEAS)
KCSI(SAVE) = 0
IF (BUGWRD) CALL OJTPUT(I I, IJ, KU, LB, KX, NP, INFEAS, KV)
RESTORE ALL TRUE DEMANDS
DO 5004 I = 1, N
I
5005 LB(11) = LB(11)
IF ((INFEAS*NEQ) .AND. (KKX(ISAVE) .NE. 0)) GO TO 5005
IF (KKX(ISAVE) .EQ. 0) GO TO 10117
WRITE (6, 10114)
10114 FORMAT (1HO, 5HM) DEMANDS AT SOME DESTINATION CANNOT BE SATISFI
ED )
GO TO 999
10117 WRITE (6, 004) KPD
004 FORMAT (1H 5X, 27H TOTALLY STOPPED ON DAY, 13)
GO TO 999
5005 INC = MAXO(I, KKX(ISAVE)/NPPTS)
KSAVE = KKX(ISAVE)
WRITE (6, 2000) KSAVE

C IF VM Odds have been interpolated on the true number of trucks yet for the comin
C PROFILE
KINTERP = .TRUE.
C
C FOR THIS NETWORK CONFIGURATION PLOT THE PROFILE
DO 99990 KPPTS = 1, NPPTS
IF (INC .EQ. KPPTS) WRITE (6, 10134) NTRKS
10134 FORMAT (48H0 »»» APPROXIMATES THE ACTUAL NUMBER OF TRUCKS. .16. 
X 1H, )
99991 IF (KLINKS .GT. NLINKS) GO TO 9990
C NOW CHOOSE A LINK TO CUT
CALL WOLMR(KY, KU, NN, II, IJ, NV, KL, NAVN, LB, KC, NP, NL, ISAVE, LBS)
9990 CONTINUE
999 CONTINUE
CALL EXIT
C FORMAT STATEMENTS
2 FORMAT (4I10.2X.A6.2X.A6.23X.L1)
003 FORMAT(10H TRUCKS = .I6.1X.8HLINKS » .A6.1X.7HSINK » .A6/ 
X 10H TONS/TRK. .F8.4.2X.6HTU/HR .F8.4 )
2000 FORMAT (1HO INVENTORY.7X TROUGHPUT.16X ACTIVE)
END
S1BFTC
WOLMR
SUBROUTINE WOLMR(KY, KU, NN, II, IJ, NV, KL, NAVN, LB, KC, NP, NL, ISAVE, 
X LBS)
COMMON /RTMK/ IRFLO
COMMON /NODEM/ ISINK
COMMON /DENOM/ IDEMOM
COMMON /FACTR/ FACTOR
COMMON /FACT/ FINCOM, FSTOP, TNSTRK, HRPRDA, TUPRHR
COMMON /MARKR/ IT(1001)
COMMON /LBUG/ BUGWRD
COMMON /TIME/ INTIME(1001)
COMMON /KDAY/ KPDSD, LINX, LINK
DIMENSION LB(11), KC(11), LBS(11)
DIMENSION KY(11), KU(11), NN(11), II(11), IJ(11), NV(11)
DIMENSION LOC(2)
LOGICAL BUGWRD
WRITE (6, 10017)
10017 FORMAT (1HO)
LINK = 0
MFLO = 999999
ITRO = 0
LB(ISAVE) = 0
KC(ISAVE) = -999999
KU(ISAVE) = IRFLO
IF (IRFLO .EQ. 0) KU(ISAVE) = 999999
DO 1 I = 1, N
IT(I) = MINO(KV(I)) .OR. KU(I)
N2 = IJ(I)
IF (MIN(N2, NE) .EQ. ISINK) GO TO 1
KLB = LBS(I1) * IRFLO / IDEMOM
LB(I1) = MINO(LRS(I1) .AND. KLB)
LB(ISAVE) = LB(ISAVE) + LB(I1)
CONTINUE
CALL MNCF(M,N+1+J+K+L+P+Q+N+L+INF+AS)
DO 2 IMAX = 1,N
     IF ( IT(IMAX) EQ. 0) GO TO 2
     IEX = 0
     DO 3 I = 1,N
        IF ( KY(I) EQ. 0) IT(I) = 0
        MARK OUT OF CONSIDERATION ALL ARCS WITH ZERO FLOW
        IF ( KY(I) EQ. 0) IT(I) = 0
        IEX = IEX + IT(I)
    END DO
    2 MARK OUT OF CONSIDERATION ALL ARCS WITH ZERO FLOW
    IF ( IEX EQ. 0) GO TO 35
    ITEMPT = KU(IMAX)
    KUMAX = 0
    KCLISAVE = -9999999
    FIND MAX FLOW IN NETWORK WITH LINK IMAX OUT
    CALL MNCF(M,N+1+J+K+L+P+Q+N+L+INF+AS)
    KCLISAVE = 0
    KUMAX = ITEMPT
    IF ( INFAS NE. 1) GO TO 6
    LINK = IMAX
    GO TO 35
    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.
    IF THEY ARE LESS FORGOT IT.
    DO 36 I = 1,N
       IF ( KY(ISAVE) GT. MFLO) GO TO 2
       CALL INPRD(KY,KC,NP,TRQ)
       IF ( KY(ISAVE) LT. MFLO) GO TO 7
       IF INTRQ LT. TRQ) GO TO 2
       IF ( KY(ISAVE) EQ. MFLO) GO TO 7
       IF ( KY(ISAVE) EQ. MFLO) GO TO 7
       MFLO = KY(ISAVE)
       C IF RESULTING FLOW IS GREATER THAN TEST FLOW SKIP THIS LINK, IF LESS KEEP THE
       C IF THEY ARE LESS FORGOT IT.
       IF THEY ARE LESS FORGOT IT.
       006 CONTINUE
       035 ZK = KU(LINK)
       IF (LINK NE. 0) GO TO 10116
       WRITE (6,10117)
       10117 FORMAT(25H NO LINKS ARE VULNERABLE.)
       RETURN
       10116 KULINK = 0
       AK = KU(LINK)
       KOUT = AK/ITUPHR*HRPRDA*+0.5
       INTIME(LINK) = KPD5 + KOUT
       A = ZE/FACTOR
       OK = KU(LINK)
       B = OK/FACTOR
       N1 = I(LINK)
       N2 = J(LINK)
       WRITE (6,9) KLINKS,NN(N1),NN(N2),A,B,INTIME(LINK)
       009 FORMAT(12H CUT NUMBER 12X25H REDUCED CAPACITY ON ARC X A6 X M TO
       XA6 X M FROM X F7.014H TO X F7.015X28H ARC WILL BE RESTORED ON DAY
       X X(I3+1)M+1)
       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 = I(I)
          M2 = J(I)
          IF (NN(M1) EQ. NN(N2) AND NN(M2) EQ. NN(N1) AND KV(I) EQ. KU(LINK))
             X GO TO 37
            END DO
         36 CONTINUE
         RETURN
         365 CONTINUE
         LINK = ITTMP
         RETURN
         37 ITTMP = LINK
         LINK = 1
         LINK1 = 1
         IEX = 9999999
         GO TO 35
      END
SUBROUTINE PROFIL(ISAVE,ISINK,IFLO,NTRKS,LBS,IDESAVE,IEND,ISAVE)
COMMON /LFLAG/ KINTRP,KCUT
COMMON /ORIG/MN,NX,KB12,NIT,NOT,NN,NP,II,J,K,KU,LB,KX,NL,KV
COMMON /LBUG/ BUGWRD
DIMENSION II(1001),IJ(1001),K(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 100 I = 1,N
N2 = IJ(I)
IF (NN(N2).NE.ISINK) GO TO 1000
KLB = LBS(I)*IFLO/IDESAVE
LB(I) = MINO(LBS(I)+KLB)
1000 CONTINUE
LB(ISAVE) = IFLO
CALL MNCF(M,N,II,IJ,KC,KU,LB,KX,NN,NL,INESAS)
IF (INESAS.NE.1) GO TO 1010
WRITE (6,2010)
2010 FORMAT (1H0,51HMIN DEMANDS CANNOT BE SATISFIED AT SOME DESTINATION)
XNOM 1
IF (BUGWRD) XCALL OUTPUT(M,IJ,KC,KU,LB,KX,NL,INESAS,KV)
KCU = .TRUE.
RETURN
1010 CALL INPRD(KX,KC,N,ITRQ)
IF (BUGWRD) CALL OUTPUT(M,IJ,KC,KU,LB,KX,NN,NL,INESAS,KV)
CALL JADJ(ITRQ,IFLO,JTRQ,JFLO)
IF (JTRQ.GE.NTRKS) AND (KINTRP)
X CALL INTRP(JTRQ,NTRKS,IFLO)
X LTRQ,JFLO,INESAS,IDESAVE,LBS,ISINK)
LTRQ = JTRQ
JFLO = IFLO
WRITE (6,2000) JTRQ,JFLO,ITRQ
2000 FORMAT (1H0,5X,16*10X,18*15X,19)
IF (INESAS.NE.0) GO TO 1020
IF (IFLO.EQ.ISAVE) GO TO 1015
IFLO = ISAVE
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
SUBROUTINE INTRP(ITRQ,NTRKS,IFLO,LTRQ,JFLO,INESAS,IDESAVE,LBS,ISINK)
COMMON /RTRK/ IRflo
COMMON /LBUG/ BUGWRD
COMMON /LFLAG/ KINTRP,KCUT
COMMON /ORIG/MN,NX,KB12,NIT,NOT,NN,NP,II,J,K,KU,LB,KX,NL,KV
DIMENSION II(1001),IJ(1001),K(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 OLDIFLO
IF (JTRQ.EQ.ITRQ) GO TO 2003
IFLO = 0
NUFLO = LFLO+IFLO+1
LTRQ = 0
NUFLO = NUFLO + 1
500 DO 1000 I = 1,N
N2 = IJ(I)
IF (NN(N2).NE.ISINK) GO TO 1000
KLB = LBS(I)*NUFLO/IDESAVE
LB(I) = MINO(KLB,LBS(I))
1000 CONTINUE
  NUTRK = 0
  LB(ISAVE) = NUFL0
  CALL MNCF(M,N,II,J,K,CX,KU,LB,XX,NN,NL,INFEAS)
  CALL IMPROD(KX,KC,N+NUTRK)
  CALL JADJ(NUTRK,NUFL0,JTRK,JFL0)
  IFLO = NUFL0
  IF (JTRK.GE.NTRK-50) GO TO 2001
  WRITE (*,2001) JTRK,JFL0,NUTRK
  FORMAT (1H,5X,16X,10X,18X,15X,19)
  NUTEM1 = NUFL0 + (IFLO-NUFL0)/3
  NUTEM2 = NUFL0 + 1
  NUFL0 = MAX0(NUTEM1,NUTEM2)
  GO TO 500
  2001 FORMAT (1H,5X,16X,10X,18X,15X,19)
  KINTRP = *FALSE*
  RETURN
END

SIBFCT OKINPT
SUBROUTINE OKINPT
COMMON /FACTR/ FACTOR
COMMON /FACT/ FINCOM,FSTOP, TNSTRK,HRPRDA, TUPRHR
COMMON /OR10/M,N,KB(12), NIT, NOT, NN, NP, 11, IJ, KC, KU, LB, KX, NL, KV
DIMENSION NL(501), NN, NP, 11, IJ(1001), KO, KU, LB, KX, NL, KV
C   INPUT TAPE
  NIT = 9
C   OUTPUT TAPE
  NOT = 6
C   MAXIMUM ARC5
  MAXM = 1000
C   MAXIMUM NODES
  MAXN = 500
  FACTOR = 1/(TNSTRK*TUPRHR*HRPRDA)
C   PREPARE TO READ DATA
  100 READ (NIT,90) (KA(I), I=1,12)
  CALL NUMK(4,36) PAUSE SAVE READY ARCS PUNCH PUNCH N KKK)
  IF (KKK EQ 2 .OR. KKK EQ 3) GO TO 102
  WRITE (NOT,91) (KA(I), I=1,12)
  IF (KKK EQ 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
  102 CALL EXIT
C   PUNCHING
  160 PUNCH 79*(KA(I), I=1,12)
  DO 161 J = 1,N
    N1 = IJ(J)
    N2 = IJ(J)
    PUNCH 75*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 MODE 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
210 WRITE(6,10126)
10126 FORMAT(5HOARCS6X*4H*CO*ST6X*5HUPPER6X*5HLOWER,10X*4HTIME/
X 5H FROM3X*2HT0*4X*9HTO REPAIR/I1X)
 DO 201 I=1,10000
 97 FORMAT(3A6.2X.4ll0.mREAD(NIT,9?)
READ(NIT,9?) KD=II(I1),IJ(I1),KC(I1),XI(I1),JLBT,JX(I1),KV(I1)
CALL NUM(KD,24HFDN NODES COMPUT,KKK)
IF (KKK.NE.4 ) GO TO 209
ALBT = JLBT
LB(I1) = ALBT*FACTOR
WRITE (6,10135)11111111111
10135 FORMAT(2A7.3lll.
201 CONTINUE
209 WRITE(NOT,93) KD,II(I1),IJ(I1)
710 N = I-1
220 IF(KKK,NE.5) GO TO 300
799 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,KX,KP
CALL NUM(KD,18HEND NODES COMPUT,KKK)
IF (KKK,NE.2) GO TO 410
N1 = NODENO(KX)
IF (N1,GT,M) GO TO 400
420 NP(N1) = NPK
GO TO 400
410 WRITE (NOT,82) KD,KP
IF (KKK,NE.4) GO TO 299
GO TO 500
C READER
500 IF (KKK,NE.3) GO TO 500
501 READ (NIT,83) KA(I1),II(I1),I1(I1)
CALL NUM(KA,24HALER COMPUT NODES,KKK)
IF (KKK,NE.2) GO TO 515
WRITE (NOT,93) KA(I1),II(I1)
560 IF (KKK,NE.2) GO TO 600
IF (KKK,NE.5) GO TO 501
IF (KKK,NE.4) GO TO 400
515 WRITE (NOT,84) (KA(I1),II(I1),I1(I1))
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.1) OR(N2,NE.IJ(I1)) GO TO 520
KA(4) = KA(4)-1
IF (KA(4),NE.0) GO TO 530
520 CONTINUE
WRITE (NOT,85)
N = N+1
I = N
530 II(I1) = N1
I1(I1) = N2
KC(I1) = KA(5)
KU(I1) = KA(6)
LB(I1) = KA(7)
XX(I1) = KA(8)+XX(I1)
GO TO 501
600 WRITE (NOT,94) N,M
C TESTS
  IF (N.LE.MAXNI) GO TO 601
  CALL EXIT
601 IF (N.LE.MAXNI) GO TO 320
  CALL EXIT
C LOOK FOR DEAD NODES
320 DO 303 I=1,N
    DO 304 L=1,N
      IF (K(J,J).GE.LB(J)) GO TO 305
      CONTINUE
    305 WRITE (NOT,99) N(I)
    306 CONTINUE
  303 CONTINUE
  DO 611 I=1,M
    NL(I) = 0
  611 CONTINUE
C CALCULATE FLOWS
DO 612 J=1,N
  IF (KU(J,J).GE.LB(J)) GO TO 609
  LUP = KU(J)
  KU(J) = LB(J)
  LB(J) = LUP
  609 N1 = I(J)
  N2 = J(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) N(I),NL(I)
  605 CONTINUE
C DO THE ALGORITHM
RETURN
END
$IBFC
SUBROUTINE MNCF(NODES, ARCS, I, J, COST, HI, LO, FLOW, PI, NA, INFEAS)
  DIMENSION LOC(2)
END
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 AA,N,SRC,SNK,DEL,INF,C,AOK,COK,N1,N2,INC,LABEL
INF = 34399738367
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 SSWITCH(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))) GO TO 50
   IF (FLOW(AA).GT.HI(AA).OR.(C.GT.0.AND.FLOW(AA).GT.LO(AA))) GO TO 60
   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) = ABS(I(A))
270 CONTINUE
C IF NO LABELING DONE ON LAST PASS, GO TO INCREASE PIE
C IF LABEL = EQ = 0 GO TO 400
C BREAKTHRU, FIND THE INCREMENT
300 INC = INF
C FOLLOW PATH BACK FROM SOURCE
N * SRC
110 A = IABS(N(A))
IF (NA(N) = LT(n) ) GO TO 315
N2 = J(A)
C = COST(A) - Pi(N) + Pi(N2)
IF (C = GT(0) ) INC = MIN0(INC-L0(A) + FLOW(A))
IF (C = LE(0) ) INC = MIN0(INC-HI(A) - FLOW(A))
GO TO 340
315 N2 = I(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(N(A))
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(A) = EQ = LO(A) ) OR = FLOW(A) = EQ = HI(A) ) GO TO 425
C INFEASIBLE SOLUTION
INFEAS = 1
RETURN
C INCREASE PI
425 DEL = IABS(C0K)
430 DO 450 N = 1,NOODES
IF (NA(N) = EQ = 0 ) Pi(N) = Pi(N) + DEL
450 CONTINUE
C GO BACK TO KILTER TEST
GO TO 100
END
$1BFCT OUTPUT
SUBROUTINE OUTPUT(I I . I J . KC . KU . LB . XX . NL . NP . INFEAS . KV)
COMMON /RTIME/ INTIME(1001)
COMMON /LBUG/ BUGWRD
COMMON /MARKR/ IT(1001)
COMMON /ORIG/. N*KB(12) . NIT . NOT . NN
DIMENSION NL(1001) . NP(1001) . NN(1001)
DIMENSION I(l(1001) , 1J(I(1001) , KC(1001) . KU(3001) . LB(3001) . XX(1001)
DIMENSION KV(1001)
LOGICAL BUGWRD
CARCS COST UPPER LOWER X CBAR
C
C KILTER NUMBER
C 1
C 92 FORMAT(5HARCS , 4HCOST , 6HUPPER , 5HLOWER , 7H1X , 8H2X , 4HCBAR , 3X.
C 18 \HINKT , 8HINTIME , 2X , 2HIT / 1 X )
C 93 FORMAT(2A7 , 4I10 , , 13HCONVERGENCE = 5X . 4HYLVLN , 4X . 6HINTIME , 2X , 2HIT / 1 X )
C
WRITE (NOT*92)
MOUT = 0
KILTER = 0
MA = 0
MB = 0
DO 901 J = 1, N
   N1 = I(1, J)
   N2 = I(J, J)
   KCK = KC(J) + NP(N1) - NP(N2)
   IF (KCK .GT. 0) KIL = IABS(KC(J)-LB(J))
   IF (KCK .LT. 0) KIL = IABS(LB(J)-KC(J))
   IF (KIL .EQ. 0) NOUT = NOUT + 1
   KILTER = KILTER + KIL
   CALL MAOD(MA, MR, KC(J), KX(J))
   IF (INFEAS .EQ. 1 .AND. (NL(N1) .GT. NL(N2))) EQ.0 .AND. (NL(N1) .EQ. NL(N2))
   IF (KIL .EQ. 0) GO TO 910
   WRITE (NOT*93)
   NN(N1) = N1(2, J)
   NN(N2) = N2(2, J)
   KC(J) = N2(2, J)
   KX(J) = N1(2, J)
   X = KCK + KIL + KX(J) + INTIME(J) + IT(J)
   GO TO 901
910 WRITE (NOT*98) NN(N1), NN(N2), KC(J), KX(J)
   X = KCK + KIL + KX(J) + INTIME(J) + IT(J)
901 CONTINUE
ME = IE*09
MD = MB*ME
MB = MB - MD*ME
M1 = MA + MD
IF (M1 .LT. 0 .OR. (ISIGN(1, M1) .EQ. ISIGN(1, MB))) GO TO 922
M2 = MB + ISIGN(MB, MB)
IF (M2 .LT. 0 .OR. (ISIGN(1, M2) .EQ. ISIGN(1, MB))) GO TO 922
922 M1 = MIN(M1, MB)
923 M2 = MB + ISIGN(MB, MB)
94 CALL IMPR(KX, KC, M, M, MT, M1, M2)
WRITE (NOT*94)
99 WRITE (NOT*99)
94 CONTINUE
99 CONTINUE
RETURN
END

FUNCTION NOENO
COMMON /ORIG/M, N, KB(12), N1, NOT, NN(1000)
NOENO = M + 1
IF (M .EQ. 0) RETURN
DO 1001 I = 1, M
   IF (I .EQ. NN(I)) NOENO = I
1001 CONTINUE
RETURN
END

SUBROUTINE NUM (KA, KB, L)
DIMENSION KB(1)
DO 1 I = 1, 100
   IF (KB(I) .EQ. 143959738367) GO TO 2
   IF (KA .EQ. KB(I)) GO TO 2
1 CONTINUE
I = 1
2 L = 1
RETURN
END
SITEMAP MADD
ENTRY MADD
MADD SAVE 4
LAC MADD+4
LDG* 4+4
MPY* 5+4
STO MA
STO MB
CLA MB
ADD* 3+4
STO* 3+4
ARS 35
ADD MA
ADD* 2+4
STO* 2+4
RETURN MADD
M: BSS 1
MB BSS 1
END
$IBFTC INPRO
SUBROUTINE INPRO(MZA,NZA,N,ITEMP)
DIMENSION MZA(1),NZA(1)
ITEMP = 0
DO 1 I = 1,N
  1 ITEMP = ITEMP + MZA(I)*NZA(I)
RETURN
END
$IBFTC JADJ
SUBROUTINE JADJ(ITRQ,IFLO,JTRQ,JFLO)
COMMON /FACT/, FINCOM, FSTOP, TNSTRK, HPRD*A, TUPRHR
A = ITRQ
ATRUCK = A*FSTOP/FINCOM
C ATRUCK IS THE ACTUAL NUMBER OF TRUCKS REQUIRED
B = IFLO
BFLO = B*TUPRHR*HPRD*A*TNSTRK
JTRQ = ATRUCK
JFLO = BFLO
RETURN
END
ENTRY CIRSFA
AN INTERDICTION MODEL OF HIGHWAY TRANSPORTATION

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.