OPTIMUM SEISMIC PROTECTION FOR NEW BUILDING CONSTRUCTION IN EASTERN METROPOLITAN AREAS

NSF GRANTS GK-27955 and GI-29936

Internal Study Report No. 38

NETWORK CONNECTIVITY

George Panoussis

March 1974

Department of Civil Engineering

Massachusetts Institute of Technology

Cambridge, Massachusetts

50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. NSF-RA-E-74-531	2.	3. Recipient's Accession No. PB 80 116700

4. Title and Subtitle
Network Connectivity (Optimum Seismic Protection for New Building Construction in Eastern Metropolitan Areas, Internal Study Report 38)

5. Report Date
March 1974

6.

7. Author(s)
G. Panoussis

8. Performing Organization Rept. No.
No. 38

9. Performing Organization Name and Address
Massachusetts Institute of Technology
Department of Civil Engineering
Cambridge, Massachusetts 02139

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C)
(G) GK27955
GI29936

12. Sponsoring Organization Name and Address
Engineering and Applied Science (EAS)
National Science Foundation
1800 G Street, N.W.
Washington, D.C. 20550

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

A FORTRAN computer program involving network connectivity is described as part of a study on seismic protection for new building construction. The program has been developed to trace all possible paths between two points of a given network, defined as a set of sites (points) interconnected with branches. Presented are: requirements necessary for input and output; the algorithm, including an example diagram and description of a subroutine alter; and the computer program itself, accompanied by a sample problem.

17. Document Analysis a. Descriptors

Earthquake resistant structures
Computer applications
Programs
Network flows
Linear programming

b. Identifiers/Open-Ended Terms

FORTRAN

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
NTIS		15
	20. Security Class (This Page)	22. Price A02 - A01

(See ANSI-Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4-77)
(Formerly NTIS-35)
Department of Commerce

INTRODUCTION

A FORTRAN computer program has been developed to trace all possible paths between two points of a given network. The motivation for writing such a program came from a predictable need of the SDDA of lifelines.

DEFINITION OF A NETWORK

A network is a set of sites (points) interconnected with branches.

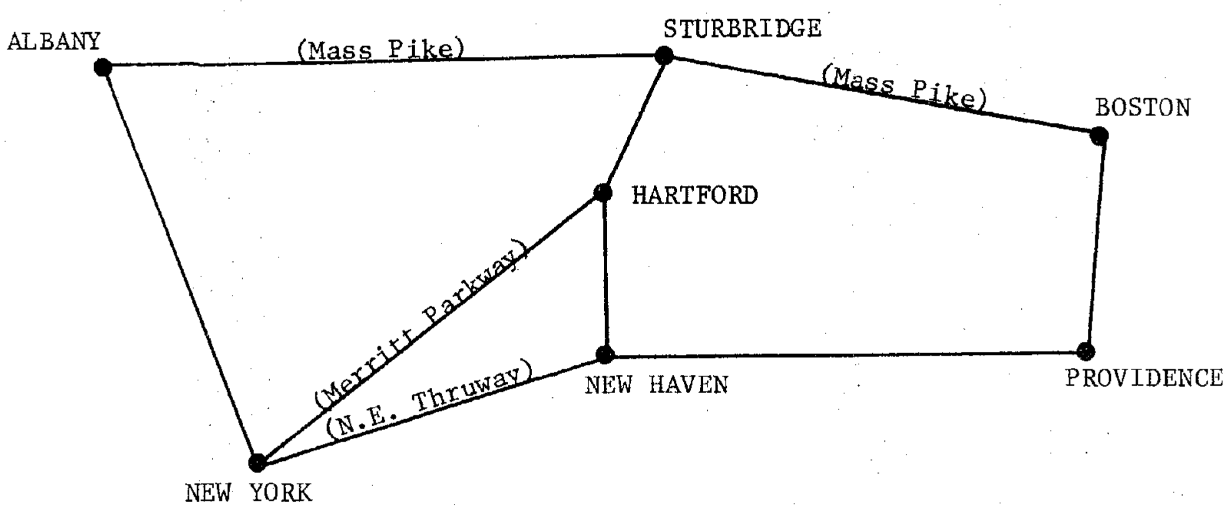


Figure 1

For example, sites, per Figure 1, are New York, Hartford, Boston, etc. Branches are the New England Thruway, Mass Pike, etc.

A minimum tie set is a set of branches and sites connecting two specified points without going over the same site more than once.

INPUT AND OUTPUT

Three sets of information are necessary for input:

1) The number and names of all sites. There is no limit to the number of sites, as long as dimensioning is adjusted accordingly. The names are alphanumerically input using any combination of up to four letters, numbers or punctuation.

2) The number of branches connecting two sites and the name of each branch, if given; the two sites connected by the branch and specification of whether it is a "one way" or a "two way" branch.

3) The two sites whose connecting paths we are interested in. We shall call them ORIGIN and FIN.

The output consists of two sets of information:

1) A connectivity matrix, $[cm_{ij}]$, such that

$$cm_{ij} = \begin{cases} 0 & \text{if } \underline{\text{no}} \text{ branch exists from i to j} \\ 1 & \text{if } \underline{\text{a}} \text{ branch exists from i to j} \end{cases}$$

cm_{ij} equals cm_{ji} if and only if the branch connecting i and j is a two way branch.

2) All paths (nodes and branches) going from ORIGIN to FIN.

ALGORITHM

The procedure the program follows is one of jumping from one site to the next possible one. If the program comes to a site it has come across before (i.e., the path is going over itself), this path is eliminated. If the prgoram arrives at FIN (i.e., the pass is completed), this path is printed as output, and another path is searched for, until all possibilities are examined.

This procedure therefore prints out all minimum tie sets.

<u>Example</u>

Given Figure 2, find all ways to go from site 1 to site 4.

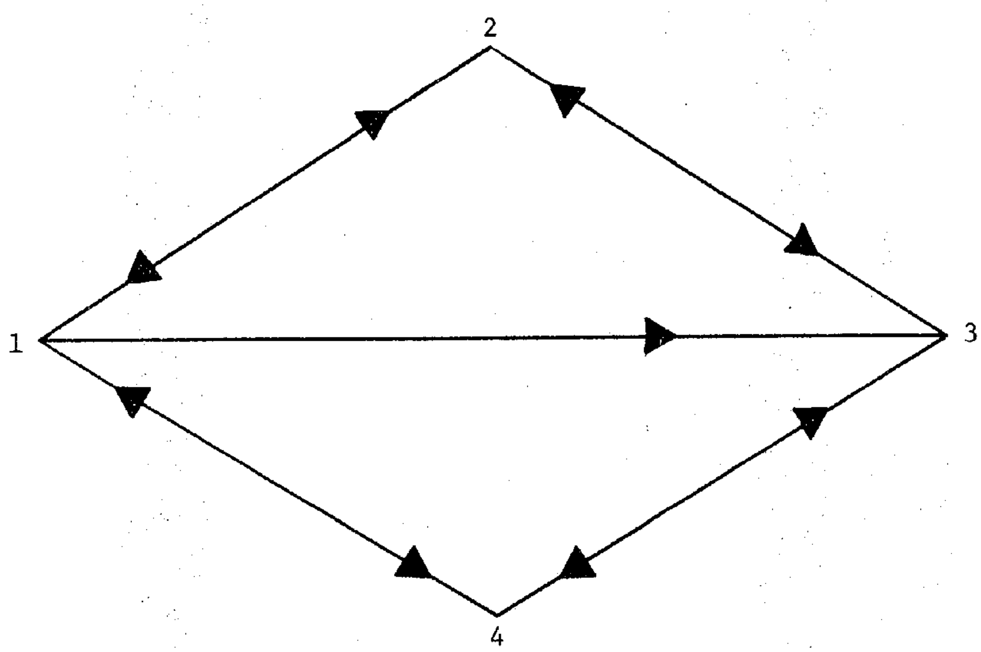


Figure 2

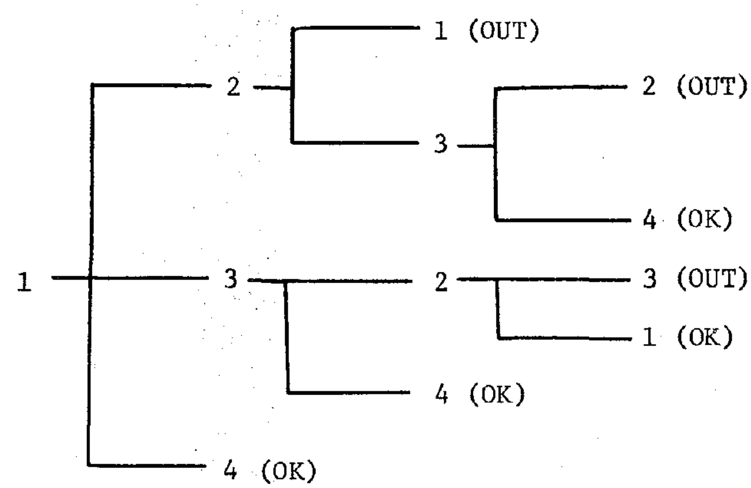


Figure 3

The computer finds all possible sites it can go to from site 1, which are 2, 3, and 4. The path 1 - 4 is complete and therefore output.

Now the computer continues from the left over sites (2 and 3) and continues trying to end the paths. If a certain path goes over the same site for a second time (e.g., 1 - 2 - 1), this path is eliminated.

SUBROUTINE ALTER

This subroutine is called to store all paths and routes generated for future reference and manipulation. In particular, it may be desirable in further computer analyses of damaged systems to eliminate a set of branches or nodes. The answers for the connectivity problem under the new circumstances can be obtained by simply eliminating all paths that involve the eliminated branches or nodes.

COMPUTER PROGRAM

```
 205 FORMAT( 6X,20(2X,A4))
 400 FORMAT(20(3X,A4))
 499 FORMAT('   CONNECTIVITY MATRIX'//)
 500 FORMAT(1X,A4,20I6)
 501 FORMAT(//' THE PATHS CONNECTING ',A4,' AND ',A4,' ARE :')
 600 FORMAT(8(6X,A4))
     INTEGER ONE,TWO
     ONE=1
     TWO=2
     INTEGER ORIGIN,END,FIN
     INTEGER PATH(20,20),WAY,ROAD(50)
     DIMENSION MATRIX(20,20) ,NAME(20),NOM(30)
     DIMENSION LINK1(30),LINK2(30),LINK(30)
     COMMON NOM,LINK
     KK=0
     READ(5,100)NSITES
     WRITE(6,499)
     READ(5,600)(NAME(K),K=1,NSITES)
     WRITE(6,205)(NAME(K),K=1,NSITES)
     DO 10 I=1,NSITES
     DO 10 J=1,NSITES
  10 MATRIX(I,J)=0
     READ(5,100) NPATHS
     DO 20 N=1,NPATHS
     READ(5,200)ROAD(N),LINK1(N),LINK2(N),WAY
 200 FORMAT(3(6X,A4),9X,I1)
     DO 29 K=1,NSITES
     IF(LINK1(N).EQ.NAME(K)) GO TO 32
  29 CONTINUE
  32 I=K
     DO 33 K=1,NSITES
     IF(LINK2(N).EQ.NAME(K)) GO TO 34
  33 CONTINUE
  34 MATRIX(I,K)=1
  20 IF(WAY.EQ.TWO) MATRIX(K,I)=1
     DO 25 J=1,NSITES
  25 WRITE(6,500) NAME(J),(MATRIX(J,I),I=1,NSITES)
  97 READ(5,200) ORIGIN,FIN
     WRITE(6,501) ORIGIN,FIN
     DO 37 K=1,NSITES
     IF(ORIGIN.EQ.NAME(K)) GO TO 38
  37 CONTINUE
  38 PATH(1,1)=K
     DO 39 K=1,NSITES
     IF(FIN.EQ.NAME(K)) GO TO 41
  39 CONTINUE
  41 END=K
 100 FORMAT(I3)
     IMAX=1
     I=1
     J=1
 110 DO 30 N=1,NSITES
     IF(MATRIX(PATH(I,J),N).NE.1) GO TO 30
```

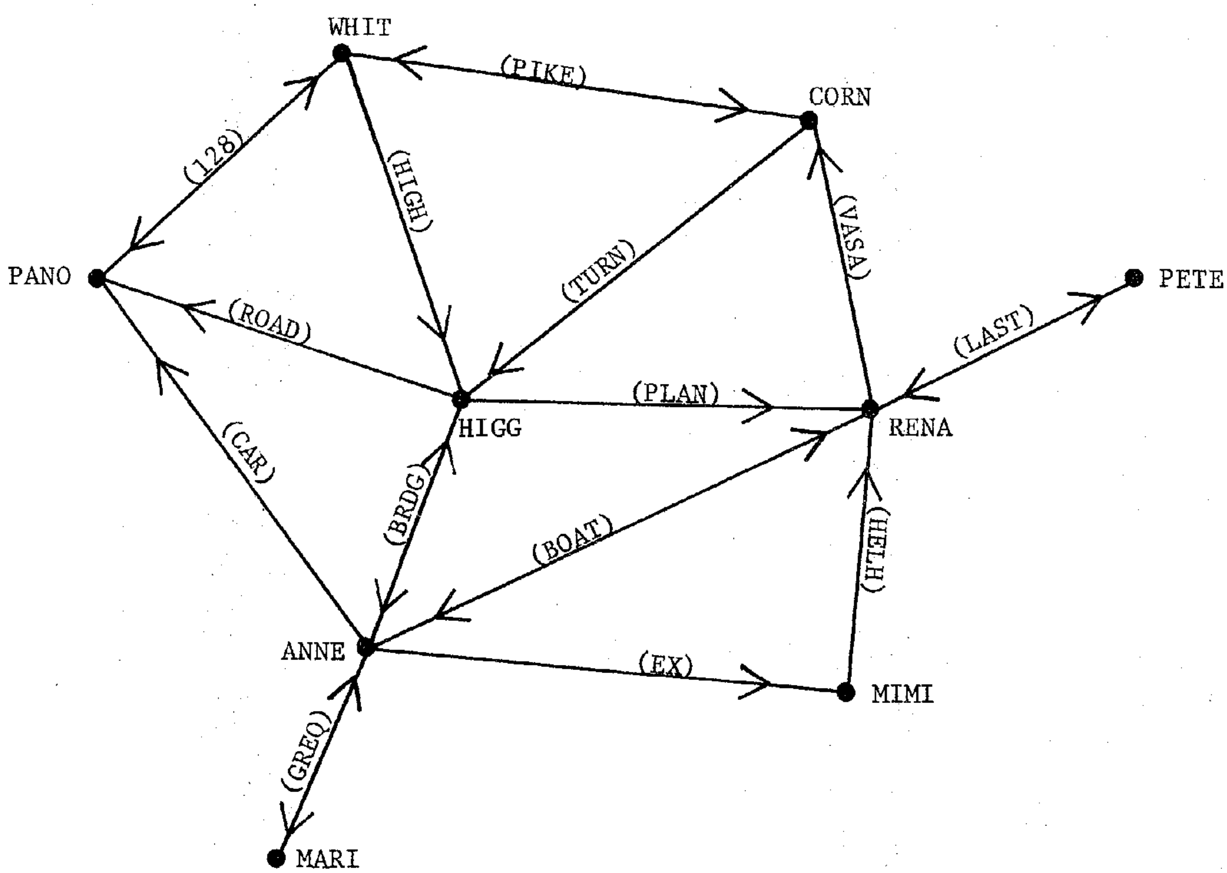
```
          K1=IMAX
       70 DO 40 K2=1,J
      40  PATH(K1+1,K2)=PATH(K1,K2)
          IF(K1.EQ.I) GO TO 60
          K1=K1-1
          GO TO 70
       60 IMAX=IMAX+1
          PATH(I,J+1)=N
          I=I+1
       30 CONTINUE
          IF(I-IMAX)80,90,90
       80 IF(PATH(I,J).EQ.PATH(I-1,J)) GO TO 82
       85 I=I+1
          GO TO 110
       82 DO 88 MI=I,IMAX
          DO 88 MJ=1,J
       88 PATH(MI,MJ)=PATH(MI+1,MJ)
          IMAX=IMAX-1
          IF(I-IMAX)110,90,90
       90 J=J+1
          I=1
      170 IF(PATH(I,J).EQ.END) GO TO 120
          JJ=J-1
          DO 140 MI=1,JJ
          MM=MI+1
          DO 140 MJ=MM,J
          IF(PATH(I,MI).EQ.PATH(I,MJ))GO TO 130
      140 CONTINUE
          I=I+1
          GO TO 190
      120 DO 121 K=1,J
      121 NOM(K)=NAME(PATH(I,K))
          WRITE(6,400)(NOM(K),K=1,J)
      401 FORMAT(' VIA THE FOLLOWING ROUTES: ',20(3X,A4)///)
          DO 122 K=1,J
          DO 123 N=1,NPATHS
          IF(NOM(K).EQ.LINK1(N).AND.NOM(K+1).EQ.LINK2(N)) GO TO 122
          IF(NOM(K).EQ.LINK2(N).AND.NOM(K+1).EQ.LINK1(N)) GO TO 122
      123 CONTINUE
      122 LINK(K)=ROAD(N)
          JJ=J-1
          WRITE(6,401)(LINK(K),K=1,JJ)
          KK=KK+1
          CALL ALTER(KK,J)
      130 IMAX=IMAX-1
          IF(IMAX.EQ.0) GO TO 180
          DO 160 MI=I,IMAX
          DO 160 MJ=1,J
      160 PATH(MI,MJ)=PATH(MI+1,MJ)
      190 IF(I.NE.IMAX) GO TO 170
          I=1
          GO TO 110
      180 GO TO 97
```

```
   99 CALL EXIT
      END
```



```
      SUBROUTINE ALTER(KK,J)
      DIMENSION NOM(30),LINK(30),NODE(30,30)
      INTEGER BRANCH(30,30)
      COMMON NOM,LINK
      DO 1 K=1,J
    1 NODE(KK,K)=NOM(K)
      J=J-1
      DO 2 K=1,J
    2 BRANCH(KK,K)=LINK(K)
      J=J+1
      RETURN
      END
```

SAMPLE PROBLEM



Find paths between PANO & MIMI

MARI & MIMI

HIGG & CORN

```
  9
     PANO      WHIT      CORN      HIGG      ANNE      RENA      PETE      MIMI
     MARI
 14
     BOAT      RENA      ANNE         2
      128      PANO      WHIT         2
     PIKE      WHIT      CORN         2
     HIGH      WHIT      HIGG         1
     TURN      CORN      HIGG         1
     ROAD      HIGG      PANO         1
     BRDG      HIGG      ANNE         2
     PLAN      HIGG      RENA         1
      CAR      ANNE      PANO         1
       EX      ANNE      MIMI         1
     GREQ      ANNE      MARI         2
     VASA      RENA      CORN         1
     LAST      RENA      PETE         2
     HELH      MIMI      RENA         1
     PANO      MIMI
     MARI      MIMI
     HIGG      CORN
```

INPUT DATA

1st card: Number of nodes

2nd card(s): Identifiers of nodes

3rd card: Number of links

4th group of cards: Identifier of link; the two nodes it connects (from, to); one-way (1) or two-way (2).

5th group of cards: Input-output pairs for which paths are requested

CONNECTIVITY MATRIX

	PANO	WHIT	CORN	HIGG	ANNE	RENA	PETE	MIMI	MARI
PANO	0	1	0	0	0	0	0	0	0
WHIT	1	0	1	1	0	0	0	0	0
CORN	0	1	0	1	0	0	0	0	0
HIGG	1	0	0	0	1	1	0	0	0
ANNE	1	0	0	1	0	1	0	1	1
RENA	0	0	1	0	1	0	1	0	0
PETE	0	0	0	0	0	1	0	0	0
MIMI	0	0	0	0	0	1	0	0	0
MARI	0	0	0	0	1	0	0	0	0

```
THE PATHS CONNECTING PANO AND MIMI ARE :
  PANO   WHIT   HIGG   ANNE   MIMI
VIA THE FOLLOWING ROUTES:     128   HIGH   BRDG   EX
  PANO   WHIT   CORN   HIGG   ANNE   MIMI
VIA THE FOLLOWING ROUTES:     128   PIKE   TURN   BRDG   EX
  PANO   WHIT   HIGG   RENA   ANNE   MIMI
VIA THE FOLLOWING ROUTES:     128   HIGH   PLAN   BOAT   EX
  PANO   WHIT   CORN   HIGG   RENA   ANNE   MIMI
VIA THE FOLLOWING ROUTES:     128   PIKE   TURN   PLAN   BOAT   EX

THE PATHS CONNECTING MARI AND MIMI ARE :
  MARI   ANNE   MIMI
VIA THE FOLLOWING ROUTES:    GREQ    EX

THE PATHS CONNECTING HIGG AND CORN ARE :
  HIGG   RENA   CORN
VIA THE FOLLOWING ROUTES:    PLAN   VASA
  HIGG   PANO   WHIT   CORN
VIA THE FOLLOWING ROUTES:    ROAD   128   PIKE
  HIGG   ANNE   RENA   CORN
VIA THE FOLLOWING ROUTES:    BRDG   BOAT   VASA
  HIGG   ANNE   PANO   WHIT   CORN
VIA THE FOLLOWING ROUTES:    BRDG   CAR   128   PIKE
  HIGG   ANNE   MIMI   RENA   CORN
VIA THE FOLLOWING ROUTES:    BRDG   EX   HELH   VASA
  HIGG   RENA   ANNE   PANO   WHIT   CORN
VIA THE FOLLOWING ROUTES:    PLAN   BOAT   CAR   128   PIKE
```

COMPUTER RESULTS