TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77843

THEMIS

(12)

TEXAS A&M UNIVERSITY
PROJECT THEMIS

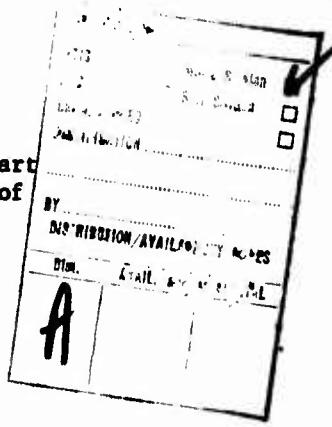
Technical Report No. 50

STATISTICAL PERT: DECOMPOSING A PROJECT NETWORK

by

R. L. Sielken Jr. and N. E. Fisher

Texas A&M Research Foundation
Office of Naval Research
Contract N00014-68-A-0140
Project NR047-700



ATTACHMENT I

STATISTICAL PERT: DECOMPOSING A PROJECT NETWORK

by

R. L. Sielken Jr. and N. E. Fisher

THEMIS OPTIMIZATION RESEARCH PROGRAM
Technical Report No. 50

INSTITUTE OF STATISTICS
Texas A&M University
November 1975

Research conducted through the
Texas A&M Research Foundation
and sponsored by the
Office of Naval Research
Contract N00014-68-A-0140
Project NR047-700

ATTACHMENT II

TABLE OF CONTENTS

ABSTRACT

Statistical PERT is a new procedure for obtaining information about the distribution of a project's completion time when the project is comprised of a large number of activities and the time required to complete an individual activity once it can be begun is a random variable. The project is represented as an acyclic network whose arcs correspond to the project activities. This network is simplified by replacing various activity configurations by single equivalent activities and then decomposed into several subnetworks. The distribution and moments of each subnetwork's completion time are bounded and approximated on the basis of two points from each activity's completion time distribution by using some mathematical programming techniques and a new result in the theory of networks. The project's completion time distribution is then approximated by combining the approximate subnetwork distributions.

This report documents two computer programs. The first program BREAKUP decomposes a project network into several subnetworks which are connected in either series or parallel in the project network. The second program LOOP checks a given project network for loops (cycles) since any loop would contradict the assumed acyclic structure of the project network.

BREAKUP

General Description;

This program "breaks up" a network into a set of subnetworks which can be linked together either in series or parallel to yield the given network. This breakup is complete in the sense that none of the subnetworks in the set can be further broken up.

The basic breakup procedure involves two main subroutines, BUNDLE and CUT. BUNDLE partitions the activities in a given network or subnetwork into parallel subnetworks connecting the network's source and sink. CUT identifies the cut nodes in a given network or subnetwork and then identifies the sets of activities between each of the consecutive cut nodes. This series of activity sets represents a breakup of the given network or subnetwork into subnetworks in series. The complete breakup of the given network is the following sequential procedure:

(1) Use BUNDLE to identify the parallel subnetworks connecting the source and sink of the network..

(2) Use CUT separately on each parallel subnetwork identified in the previous step - (1) or (3) -, and breakup the parallel subnetwork into subnetworks in series. If no such breakup is possible for a parallel subnetwork, that subnetwork is not considered again. If no new series subnetwork is identified in this entire step, stop. Otherwise go to (3).

(3) Use BUNDLE separately on each series subnetwork identified in (2). If BUNDLE cannot breakup a series subnetwork, that subnetwork is not considered again. If no new parallel subnetwork is identified in this entire step,

A schematic example of this procedure accompanies the sample problem.

Specific Input Instructions:

Card 1. Col. 1-3: The number of arcs in the network, Format (I3).

Col. 4-6: The number of the node which is the source node, Format (I3).

Col. 7-9: The number of the node which is the sink node, Format (I3).

Col. 10-12: The largest node number in the network, Format (I3).

For each activity one card with:

Col. 1-3: The activity's number, Format (I3).

Col. 4-6: The activity's origin node number, Format (I3).

Col. 7-9: The activity's terminal node number, Format (I3).

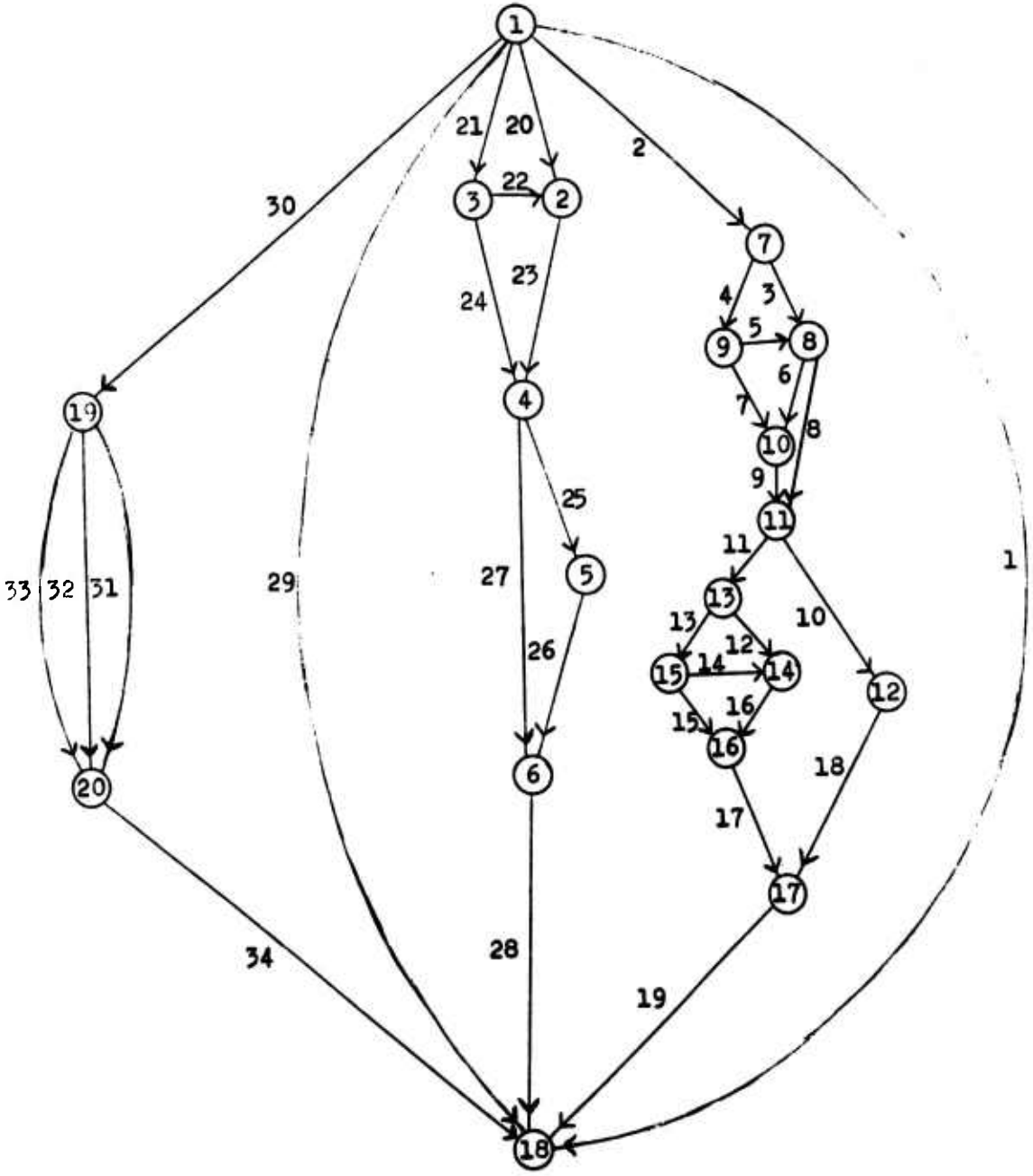
The activities and nodes may be numbered in any way and may be read in in any order.

Dimension Restrictions:

Currently this program will accomodate a network that has a maximum of 300 nodes and a maximum of 300 arcs. It can store a maximum of 100 subnetworks with each subnetwork having a maximum of 300 arcs.

This program is written in FORTRAN G.

SAMPLE NETWORK



SAMPLE INPUT

```
34  1 18 20
 2  1  7
15 15 16
12 13 14
27  4  6
 3  7  3
28  6 18
29  1 13
 4  7  9
 5  9  8
13 13 15
 6  8 10
15 14 16
17 16 17
14 15 14
 7  9 10
18 12 17
30  1 19
19 17 18
 8  8 11
20  1  2
31 19 20
21  1  3
 9 10 11
10 11 12
22  3  2
23  2  4
32 19 20
33 19 20
34 20 18
24  3  4
25  4  5
26  5  6
 1  1 18
11 11 13
```

INPUT STAGE

THE INITIAL NETWORK HAS 34 ARCS

THE SOURCE IS NODE NUMBER 1

THE SINK IS NODE NUMBER 18

THE LARGEST NODE IS NODE NUMBER 20

THE INITIAL NETWORK AS READ IN IS:

ARC NUMBER	ORIGIN NODE	TERMINAL NODE
2	1	7
15	15	16
12	13	14
27	4	6
3	7	8
28	6	18
29	1	18
4	7	9
5	9	8
13	13	15
6	8	10
16	14	16
17	16	17
14	15	14
7	9	10
18	12	17
33	1	19
19	17	18
8	8	11
20	1	2
31	19	20
21	1	3
9	10	11
10	11	12
22	3	2
23	2	4
32	19	20
33	19	20
34	20	18
24	3	4
25	4	5
26	5	6
1	1	18
11	11	13

SUBNETWORK 1 IS COMPOSED OF SUBNETWORKS:

2, 3, 4, 5, 6,

1 PARALLEL

```
STAGE  1 BREAKUP

SUBNETWORK   2 IS COMPOSED OF SUBNETWORKS:

  7,  8,  9, 10,

IN SERIES

SUBNETWORK   3 IS COMPOSED OF SUBNETWORKS:

 11, 12, 13,

IN SERIES


          SUBNETWORK    4 IS A MINIMUM NETWORK
          IT IS COMPOSED OF:

              SOURCE NODE =   1

              SINK NODE   =  18

              ARC  S(ARC)  T(ARC)
               29      1      18


SUBNETWORK   5 IS COMPOSED OF SUBNETWORKS:

 14, 15, 16,

IN SERIES


          SUBNETWORK    6 IS A MINIMUM NETWORK
          IT IS COMPOSED OF:

              SOURCE NODE =   1

              SINK NODE   =  18

              ARC  S(ARC)  T(ARC)
                1      1      18
```

SUBNETWORK 7 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 7

ARC	S(ARC)	T(ARC)
2	1	7

SUBNETWORK 8 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 7

SINK NODE = 11

ARC	S(ARC)	T(ARC)
3	7	8
4	7	9
6	8	10
8	8	11
5	9	8
7	9	10
9	10	11

SUBNETWORK 9 IS COMPOSED OF SUBNETWORKS:

17, 18,

IN PARALLEL

SUBNETWORK 10 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 17

SINK NODE = 18

ARC	S(ARC)	T(ARC)
19	17	18

SUBNETWORK 11 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 4

ARC	S(ARC)	T(ARC)
20	1	2
21	1	3
23	2	4
22	3	2
24	3	4

SUBNETWORK 12 IS COMPOSED OF SUBNETWORKS:

19, 20,

IN PARALLEL

SUBNETWORK 13 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 6

SINK NODE = 18

ARC	S(ARC)	T(ARC)
28	6	18

SUBNETWORK 14 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 1

SINK NODE = 19

ARC	S(ARC)	T(ARC)
30	1	19

SUBNETWORK 15 IS COMPOSED OF SUBNETWORKS:

21, 22, 23,

IN PARALLEL

SUBNETWORK 16 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 20

SINK NODE = 18

ARC	S(ARC)	T(ARC)
34	20	18

STAGE 3 BREAKUP

SUBNETWORK 17 IS COMPOSED OF SUBNETWORKS:

24, 25,

IN SERIES

SUBNETWORK 18 IS COMPOSED OF SUBNETWORKS:

26, 27, 28,

IN SERIES

SUBNETWORK 19 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 4

SINK NODE = 6

ARC	S(ARC)	T(ARC)
27	4	6

SUBNETWORK 20 IS COMPOSED OF SUBNETWORKS:

29, 30,

IN SERIES

SUBNETWORK 21 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 19

SINK NODE = 20

ARC	S(ARC)	T(ARC)
31	19	20

SUBNETWORK 22 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 19

SINK NODE = 20

ARC	S(ARC)	T(ARC)
32	19	20

```
SUBNETWORK  23 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

     SOURCE NODE =  19

     SINK NODE   =  20

     ARC  S(ARC)  T(ARC)
      33    19      20
```

SUBNETWORK 24 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 11

SINK NODE = 12

ARC	S(ARC)	T(ARC)
10	11	12

SUBNETWORK 25 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 12

SINK NODE = 17

ARC	S(ARC)	T(ARC)
18	12	17

SUBNETWORK 26 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 11

SINK NODE = 13

ARC	S(ARC)	T(ARC)
11	11	13

SUBNETWORK 27 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

SOURCE NODE = 13

SINK NODE = 16

ARC	S(ARC)	T(ARC)
12	13	14
13	13	15
16	14	16
15	15	16
14	15	14

SUBNETWORK 28 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

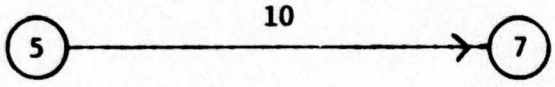
SOURCE NODE = 16

```
     SINK NODE    =  17

     ARC  S(ARC)  T(ARC)
      17    16      17


SUBNETWORK  29 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

     SOURCE NODE =   4

     SINK NODE   =   5

     ARC  S(ARC)  T(ARC)
      25    4       5


SUBNETWORK  30 IS A MINIMUM NETWORK
IT IS COMPOSED OF:

     SOURCE NODE =   5

     SINK NODE   =   6

     ARC  S(ARC)  T(ARC)
      26    5       6
```

SAMPLE PROBLEM: SCHEMATIC REPRESENTATION

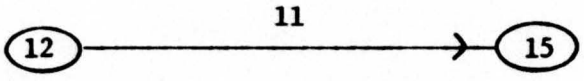
BREAKUP

Arcs are designated as follows:

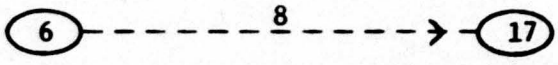


The arc number is written above or to the left of the arc activity line. The arrowhead indicates the direction of the activity. The circled numbers are node numbers. In this case the arc depicted is arc number 10, flowing from left to right with beginning node 5 and terminating node 7.

Subnetworks are designated as follows:



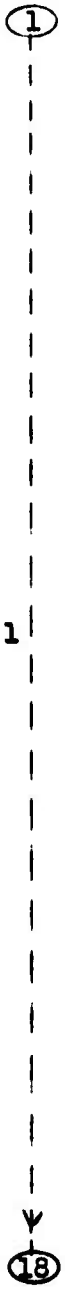
The subnetwork number is written above or to the left of the network activity direction line. The arrowhead indicates the direction of the subnetwork activity. The numbers enclosed in ovals are source and sink nodes. In this case the subnetwork depicted is subnetwork 11, flowing from left to right with source node 12 and sink node 15. A solid activity direction line indicates the subnetwork is a minimum network. A broken line indicates the subnetwork must be considered by at least one more subroutine of the BREAKUP program.



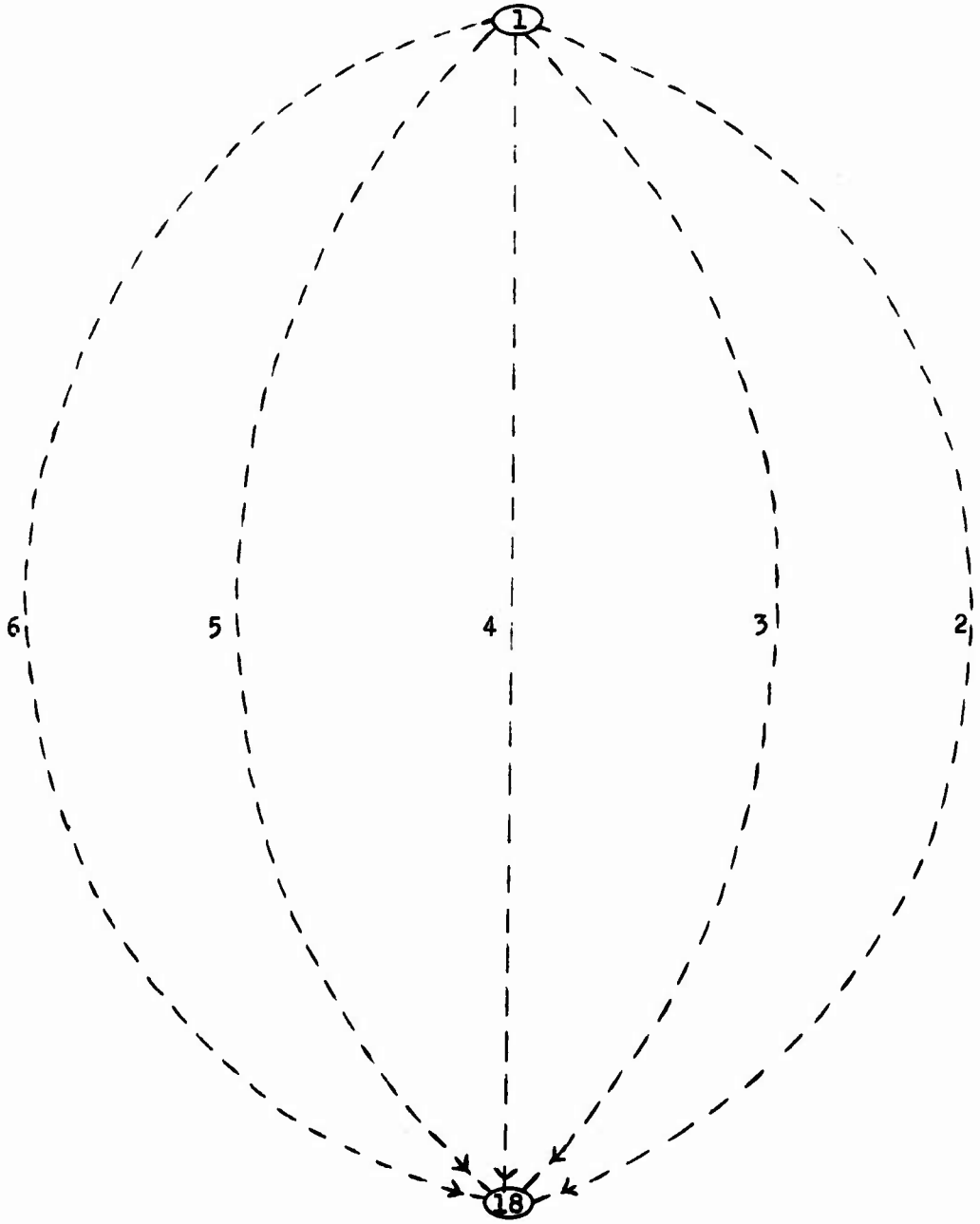
The above figure indicates subnetwork 8, flowing from left to right with source node 6 and sink node 17, is not yet a minimum network.

In the BREAKUP diagrams, the decomposition of the initial network is finished when all subnetworks are minimum networks, e.g. all lines are solid.

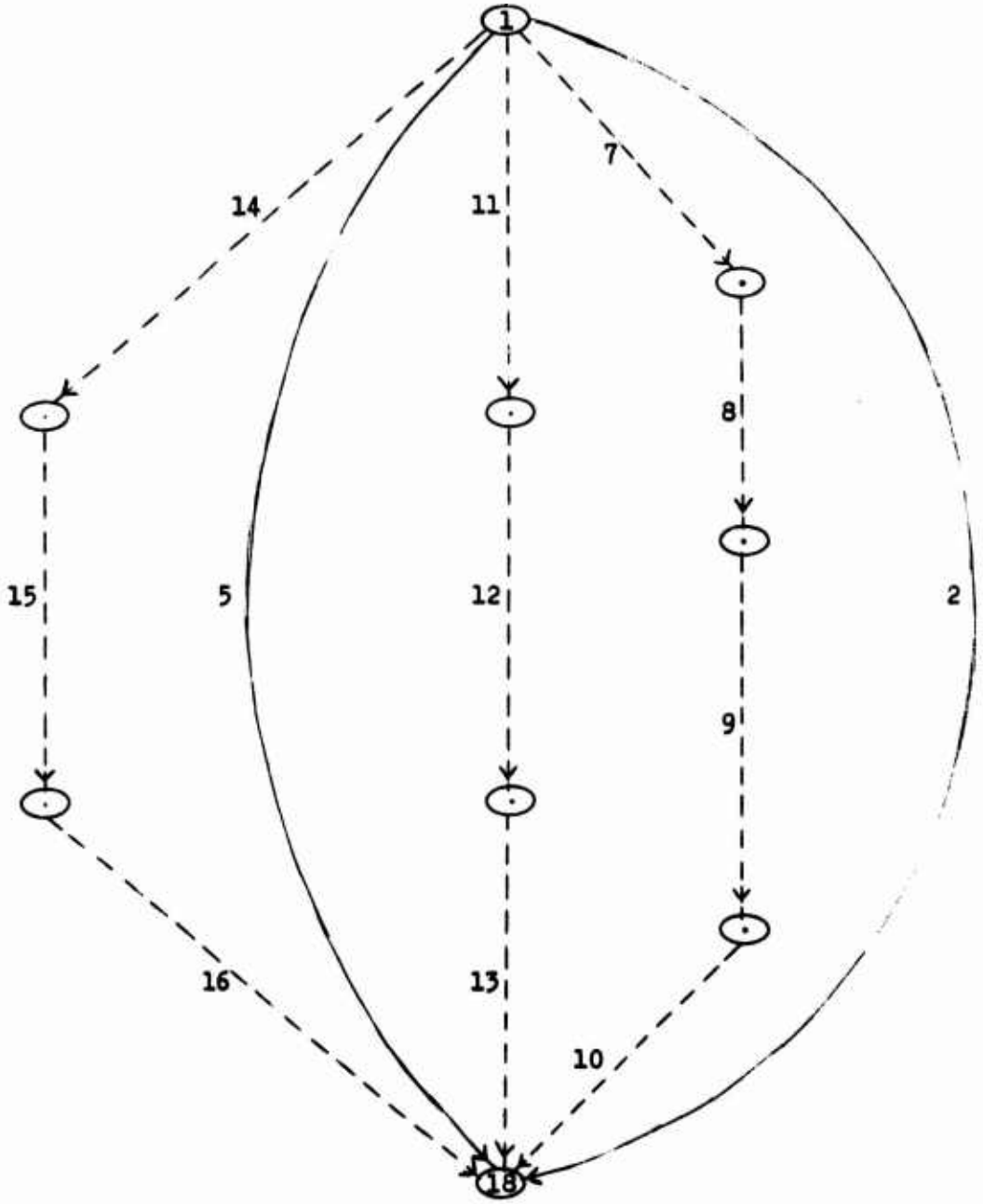
INITIAL NETWORK



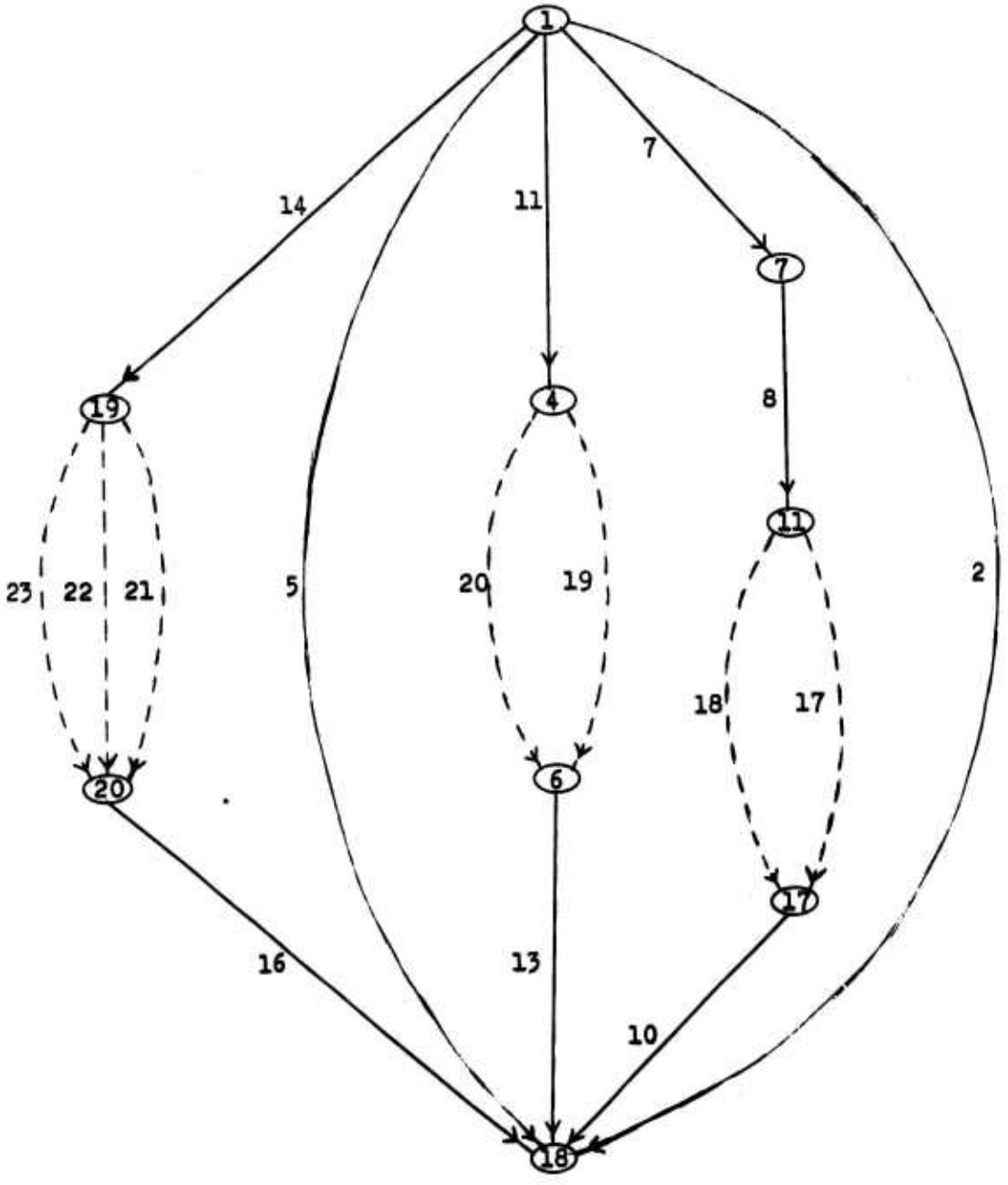
INPUT STAGE



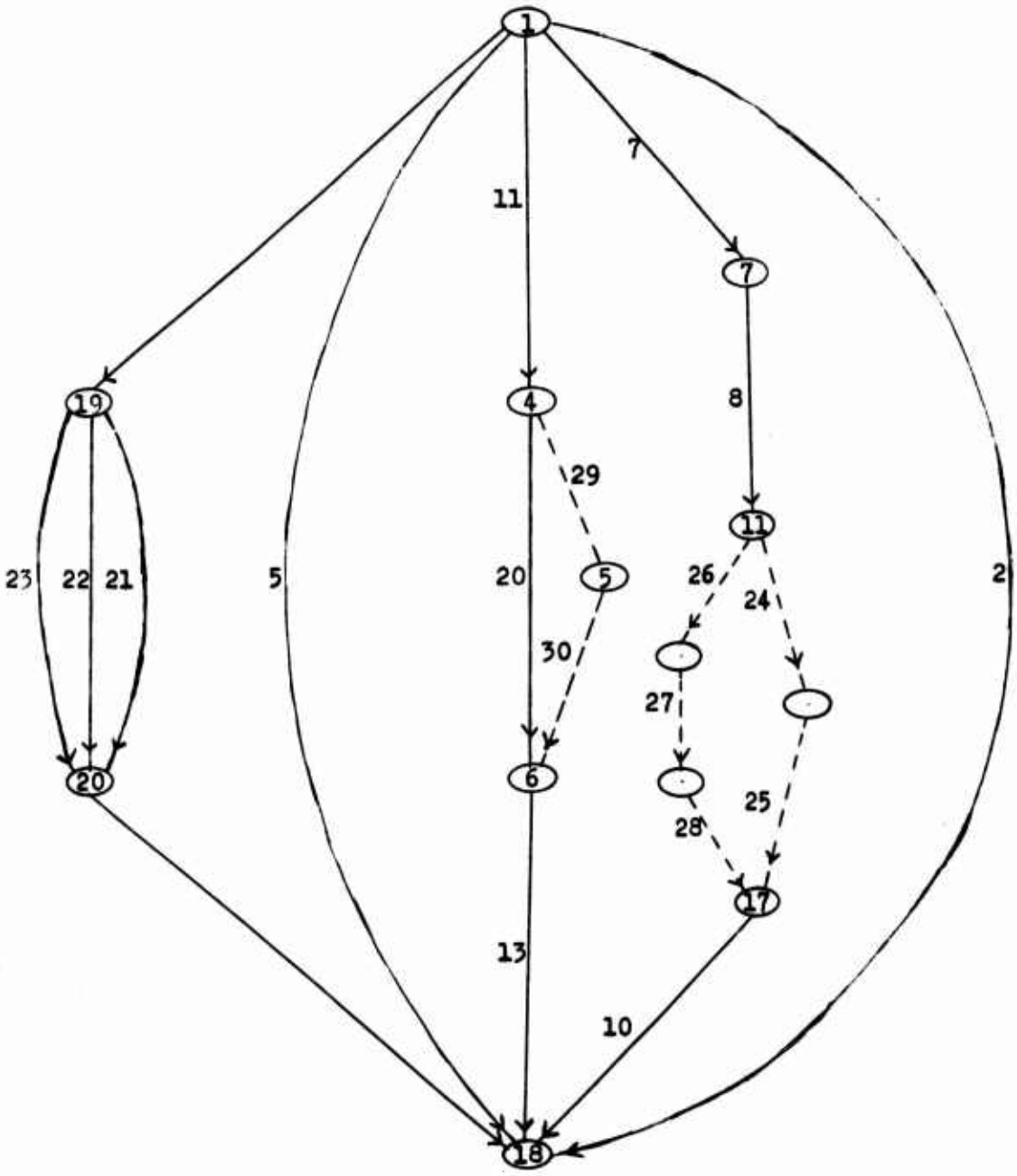
STAGE 1



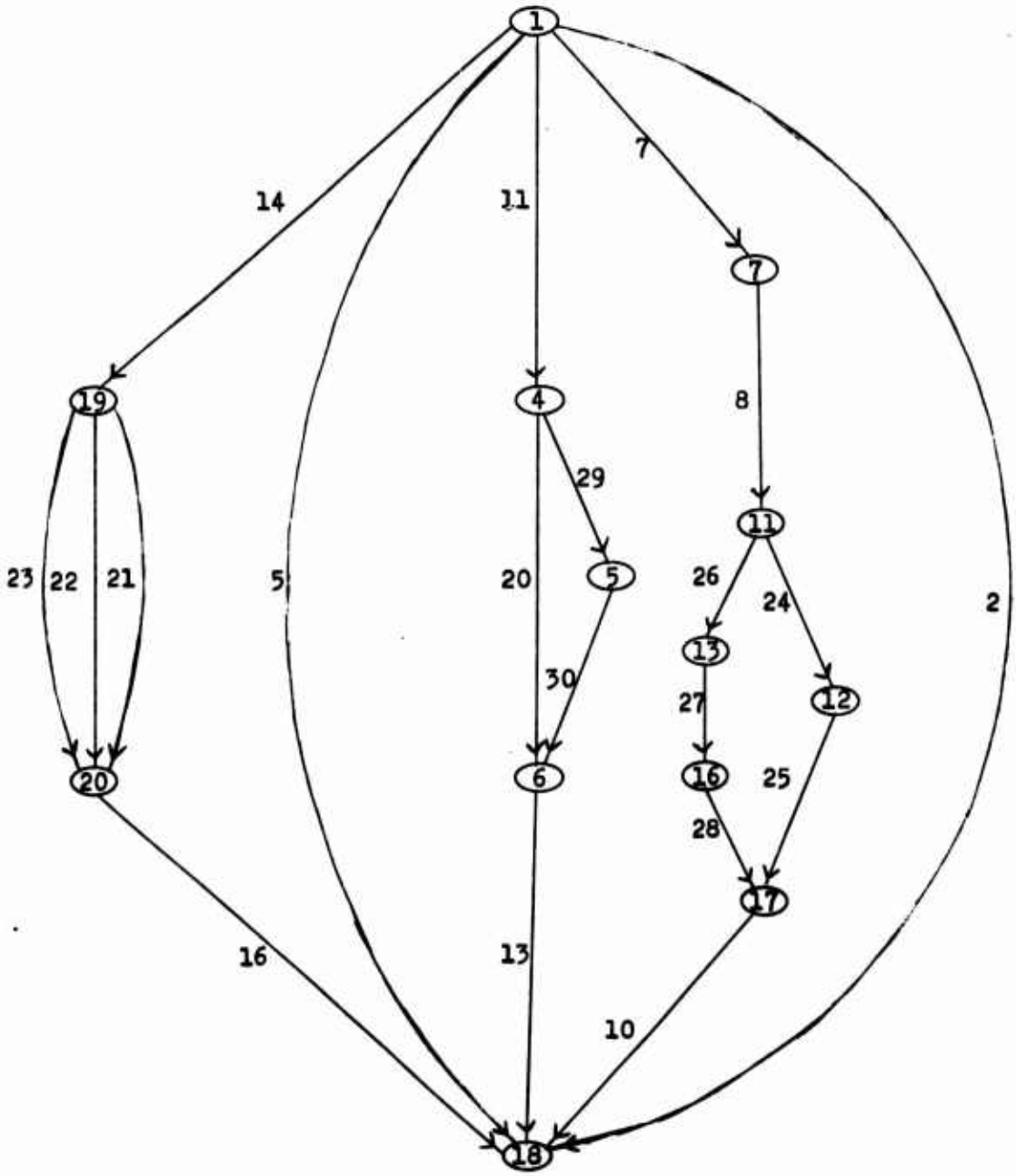
STAGE 2



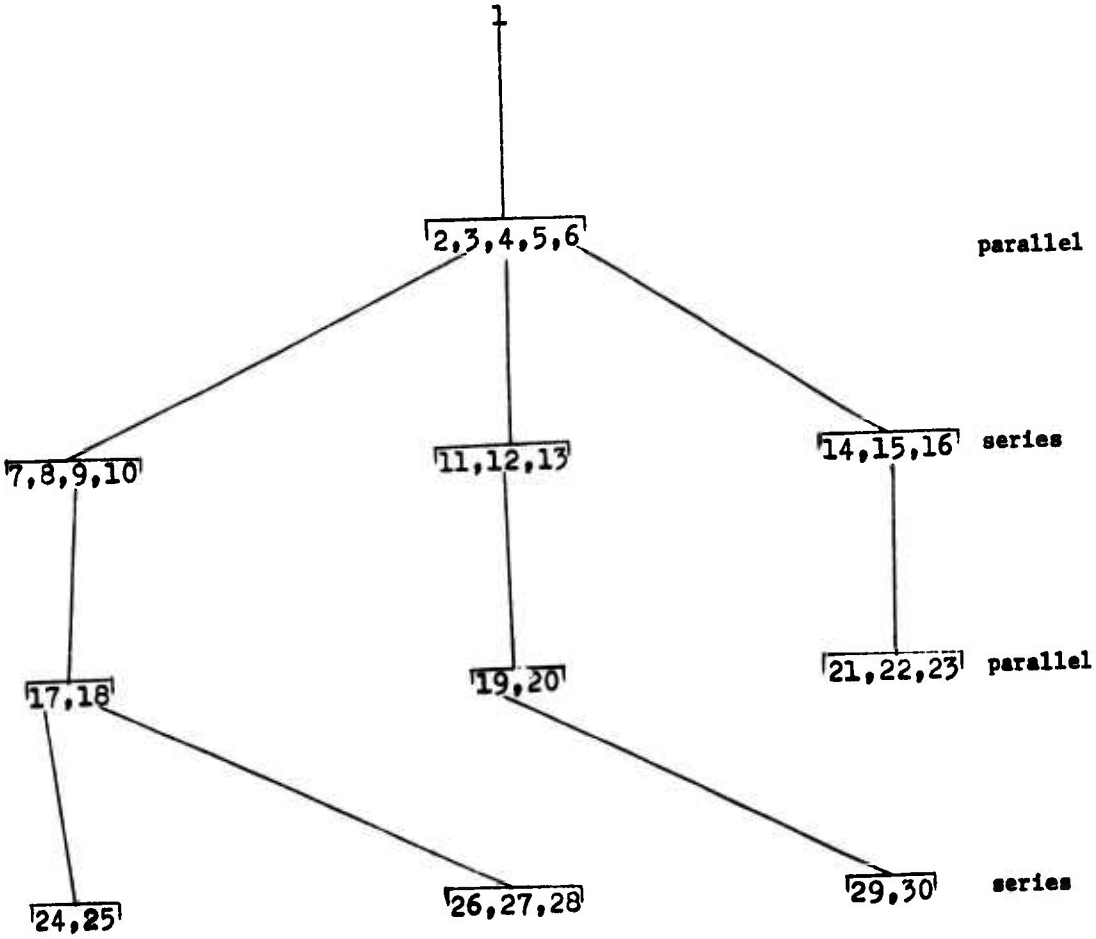
STAGE 3



STAGE 4


1
14
11
7
19
4
7
8
29
11
23
22
21
5
20
5
26
24
2
13
30
12
27
20
6
16
25
28
17
16
13
10
18

BREAKUP FLOW



```
C       PROGRAM BREAKUP                                                    1
C                                                                          2
C       BREAKUP IDENTIFIES PARALLEL SUBNETWORKS AND SERIES SUBNETWORKS     3
C                                                                          4
C       MORE THAN ONE NETWORK MAY BE DECOMPOSED DURING A RUN.              5
C       SEPARATE THE DATA FOR EACH SUBNETWORK BY A BLANK CARD              6
C                                                                          7
C                                                                          8
C       THE FOLLOWING IS AN ALPHABETICAL LISTING OF THE VARIABLES AND      9
C       ARRAYS THAT ARE USED IN THIS MAIN PROGRAM AND ITS SUBROUTINES      10
C                                                                          11
C                                                                          12
C       ARCS = THE NUMBER OF ARCS IN THE SUBNETWORK                        13
C       BONUM(I) = THE BUNDLE NUMBER TO WHICH NODE I IS ASSIGNED           14
C       CHECK = ARRAY USED TO STORE ARCS HAVING THE SOURCE AND THE         15
C               SINK AS THEIR ONLY NODES                                   16
C       CTNSUB = THE TEMPORARY NUMBER OF SUBNETWORKS FOUND IN THE          17
C                PREVIOUS STEP                                             18
C       LNODEN = THE LARGEST NODE NUMBER BEING READ IN                     19
C       MAXND = THE LARGEST NODE NUMBER THAT HAS ALREADY BEEN ASSIGNED     20
C            AT LEAST TEMPORARILY TO A BUNDLE                              21
C       NARCSS(I) = THE NUMBER OF ARCS IN SUBNETWORK I                     22
C       NSUB = THE TOTAL OF SUBNETWORKS THUS FAR                           23
C       NTARC = THE NUMBER OF ARCS IN THIS SUBNETWORK                      24
C       NUMBD = THE NUMBER OF BUNDLES CREATED                              25
C       S(I) = THE STARTING NODE FOR ARC I                                 26
C       SINK    = THE NODE NUMBER CORRESPONDING TO THE SINK                27
C       SINKS(I) = THE SINK IN SUBNETWORK I                                28
C       SOURC(I) = SOURCE NODE IN SUBNETWORK I                             29
C       SOURCE = THE NODE NUMBER CORRESPONDING TO THE SOURCE               30
C       STEP = STAGE NUMBER                                                31
C       SUBNET(I,J) = THE ITH ARC IN THE JTH SUBNETWORK                    32
C       SUMARC = THE CURRENT NUMBER OF ARCS IN SUBNETWORK NSUB             33
C       T(I) = THE TERMINATING NODE FOR ARC I                              34
C       TARC = THE SUBNETWORK WITHOUT THE ARCS INVOLVING NODE K            35
C       TLNDEN = TEMPORARY LARGEST NODE NUMBER                             36
C       TNSUB = THE NUMBER OF SUBNETWORKS CREATED IN THE CURRENT STAGE     37
C       TSNSUB = THE TEMPORARY SUBNETWORK BEING USED IN STAGE              38
C                SUBROUTINE                                                39
C       TSUBN = THE NUMBER OF THE SUBNETWORK CURRENTLY BEING CONSIDERED    40
C       TYPESN = THE TYPE OF SUBNETWORK BEING CONSIDERED:                  41
C            1 = BUNDLE SUBNETWORK  2 = CUT SUBNETWORK                     42
C                                                                          43
C       A MINIMUM SUBNETWORK IS ONE THAT IS NOT COMPOSED OF SMALLER        44
C       SUBNETWORKS                                                        45
C                                                                          46
      IMPLICIT INTEGER*2 (A-Z)                                             47
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS                      48
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)                   49
      DIMENSION SINKS(100),NARCSS(100)                                     50
C                                                                          51
C       THIS PROGRAM WILL ACCOMODATE A NETWORK THAT HAS A MAXIMUM OF       52
C       300 NODES, A MAXIMUM OF 300 ARCS AND CAN STORE A MAXIMUM OF        53
C       100 SUBNETWORKS IN THE BREAKUP PROCESS                             54
C                                                                          55
C       NA = THE NUMBER OF ARCS IN THE NETWORK TO BE CONSIDERED            56
C       NS = THE TOTAL NUMBER OF POSSIBLE SUBNETWORKS IN THE BREAKUP       57
C            PROCESS                                                       58
C       IN THE CASE WHERE NO GOOD ESTIMATE CAN BE MADE OF NS, USE NA       59
```

```
C         FOR ALL DIMENSIONS                                                    60
C         THE ARRAY DIMENSIONS ARE:                                             61
C         S(NA),T(NA),SUBNET(NA,NS),SOURC(NS),SINKS(NS),NARCSS(NS)              62
C                                                                               63
C         IN SUBROUTINE BUNDLE THE ARRAY DIMENSIONS ARE:                        64
C         BDNUM(NS),CHECK(NS)                                                   65
C                                                                               66
C         IN SUBROUTINE CUT THE ARRAY DIMENSIONS ARE:                           67
C         ORIGIN(NS),POST(NS),TARC(NS),RCUT(NA)                                 68
C                                                                               69
C         READ THE NETWORK IN                                                   70
C                                                                               71
6000   CALL NETIN (LNODEN)                                                      72
       STEP=1                                                                   73
       TYPESN=1                                                                 74
       TSUBN=1                                                                  75
       CALL BUNDLE (LNODEN,TSUBN)                                               76
C                                                                               77
C         PRINT OUT THIS STAGE OF THE BREAKUP                                   78
C                                                                               79
       CALL STAGE (TYPESN,TSUBN)                                                80
       CTNSUB=TNSUB                                                             81
       GO TO 70                                                                 82
C                                                                               83
C         FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS                          84
C                                                                               85
10     CTNSUB=0                                                                 86
       TSUBN=NSUB-LOOP                                                          87
       TYPESN=1                                                                 88
       DO 20 I=1,LOOP                                                           89
C                                                                               90
C         FIND THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED                     91
C                                                                               92
       TSUBN=TSUBN+1                                                            93
C                                                                               94
C         FIND THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN                      95
C                                                                               96
       CALL NODER (TLNDEN,TSUBN)                                                97
C                                                                               98
C         FIND THE BUNDLE SUBNETWORKS                                           99
C                                                                              100
       CALL BUNDLE (TLNDEN,TSUBN)                                              101
C                                                                              102
C         PRINT OUT THIS STAGE OF THE BREAKUP                                  103
C                                                                              104
       CALL STAGE (TYPESN,TSUBN)                                               105
C                                                                              106
C         IF THERE IS ONLY ONE BUNDLE FOUND IN SUBNETWORK TSUBN, WE ARE        107
C         FINISHED.  PRINT OUT ITS COMPONENT ARCS                              108
C                                                                              109
       IF (TNSUB.EQ.1) CALL ENDSNT (TSUBN)                                     110
C                                                                              111
C         COUNT THE NEW NUMBER OF SUBNETWORKS CREATED                          112
C                                                                              113
20     CTNSUB=TNSUB+CTNSUB                                                     114
C                                                                              115
C         IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM, WE ARE FINISHED       116
C                                                                              117
       IF (CTNSUB.EQ.0) GO TO 90                                               118
       GO TO 70                                                                119
```

```
80    LOOP=CTNSUB                                                          120
      CTNSUB=0                                                             121
      TSUBN=NSUB-LOOP                                                      122
C                                                                          123
C        FIND THE NUMBER OF NEWLY CREATED SUBNETWORKS                      124
C                                                                          125
      TYPESN=2                                                             126
      DO 30 I=1,LOOP                                                       127
C                                                                          128
C        FIND THE NEXT SUBNETWORK TO BE FURTHER SUBDIVIDED                 129
C                                                                          130
      TSUBN=TSUBN+1                                                        131
      IF (TSUBN.EQ.1) GO TO 55                                             132
C                                                                          133
C        FIND THE LARGEST NODE NUMBER IN SUBNETWORK TSUBN                  134
C                                                                          135
      CALL NODER (TLNDEN,TSUBN)                                            136
C                                                                          137
C        FIND THE CUT SUBNETWORKS                                          138
C                                                                          139
55    IF (TSUBN.EQ.1) TLNDEN=LNODEN                                        140
      CALL CUT (TSUBN,TLNDEN)                                              141
C                                                                          142
C        PRINT OUT THIS STAGE OF THE BREAKUP                               143
C                                                                          144
      CALL STAGE (TYPESN,TSUBN)                                            145
C                                                                          146
C        IF THERE ARE NO CUTS FOUND IN SUBNETWORK TSUBN, WE ARE FINISHED   147
C        PRINT OUT ITS COMPONENT ARCS                                      148
C                                                                          149
      IF (TNSUB.EQ.1) CALL ENDSNT (TSUBN)                                  150
C                                                                          151
C        COUNT THE NEW NUMBER OF SUBNETWORKS CREATED                       152
C                                                                          153
30    CTNSUB=TNSUB+CTNSUB                                                  154
C                                                                          155
C        IF ALL SUBNETWORKS ARE IN THEIR SMALLEST FORM, WE ARE FINISHED    156
C                                                                          157
      IF (CTNSUB.EQ.0) GO TO 90                                            158
      GO TO 70                                                             159
85    LOOP=CTNSUB                                                          160
      GO TO 10                                                             161
70    WRITE (6,900) STEP                                                   162
900   FORMAT (1H1,5X,'STAGE',I3,' BREAKUP')                                163
      STEP=STEP+1                                                          164
C                                                                          165
C        LET'S GO BACK TO THE APPROPRIATE LOOP FOR THE NEXT STAGE          166
C                                                                          167
      GO TO (80,85),TYPESN                                                 168
90    CONTINUE                                                             169
      READ (5,100,END=666)                                                 170
100   FORMAT (I3)                                                          171
      GO TO 6000                                                           172
666   CONTINUE                                                             173
      WRITE (6,9000)                                                       174
9000  FORMAT (1H1)                                                         175
      RETURN                                                               176
      END                                                                  177

      SUBROUTINE NETIN (LNODEN)                                            178
```

```
      IMPLICIT INTEGER*2 (A-Z)                                          179
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS                   180
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)                181
      DIMENSION SINKS(100),NARCSS(100)                                  182
C                                                                       183
C        ZEROIZE SOURC ARRAY                                            184
C                                                                       185
      DO 20 I=1,50                                                      186
20    SOURC(I)=0                                                        187
C                                                                       188
C        READ IN THE INITIAL NETWORK LIMITS                             189
C                                                                       190
      READ (5,100) ARCS,SOURCE,SINK,LNODEN                              191
100   FORMAT (4I3)                                                      192
      WRITE (6,200)                                                     193
200   FORMAT (1H1,5X,'INPUT STAGE')                                     194
      WRITE (6,210) ARCS,SOURCE,SINK,LNODEN                             195
210   FORMAT (1H0,/6X,'THE INITIAL NETWORK HAS',12X,I4,' ARCS',//6X,    196
     *'THE SOURCE IS NODE NUMBER',11X,I3,//6X,'THE SINK IS NODE NUMBER',  197
     *13X,I3,//6X,'THE LARGEST NODE IS NODE NUMBER',5X,I3)              198
      WRITE (6,220)                                                     199
220   FORMAT (1H0,5X,'THE INITIAL NETWORK AS READ IN IS:',///6X,        200
     *'ARC NUMBER',5X,'ORIGIN NODE',5X,'TERMINAL NODE')                 201
C                                                                       202
C        READ IN EACH ARC AND ITS STARTING AND TERMINATING NODES        203
C        THE ARCS AND NODES MAY BE NUMBERED ANY WAY AND READ IN IN ANY  204
C        ORDER                                                          205
C        I = ARC NUMBER                                                 206
C        S = THE NODE NUMBER FOR THE START OF AN ARC                    207
C        T = THE TERMINAL NODE OF AN ARC                                208
C                                                                       209
      DO 10 J=1,ARCS                                                    210
      READ (5,100) I,S(I),T(I)                                          211
      WRITE (6,240) I,S(I),T(I)                                         212
240   FORMAT (1H ,8X,I3,13X,I3,14X,I3)                                  213
C                                                                       214
C        CREATE THE FIRST SUBNETWORK                                    215
C                                                                       216
10    SUBNET (J,1)=I                                                    217
      SOURC(1)=SOURCE                                                   218
      SINKS(1)=SINK                                                     219
      NARCSS(1)=ARCS                                                    220
      NSUB=1                                                            221
      RETURN                                                            222
      END                                                               223

      SUBROUTINE NODER (TLNDEN,TSUBN)                                   224
C                                                                       225
C        FINDS LARGEST NODE NUMBER IN THE SUBNETWORK TSUBN              226
C                                                                       227
      IMPLICIT INTEGER*2 (A-Z)                                          228
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS                   229
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)                230
      DIMENSION SINKS(100),NARCSS(100)                                  231
      ARCS=NARCSS(TSUBN)                                                232
      TLNDEN=0                                                          233
      DO 20 J=1,ARCS                                                    234
      A=SUBNET(J,TSUBN)                                                 235
      M=S(A)                                                            236
      N=T(A)                                                            237
```

```
      MAXND=N                                                          238
      IF (M.GT.N) MAXND=M                                              239
      IF (MAXND.GT.TLNDEN) TLNDEN=MAXND                                240
20    CONTINUE                                                         241
      RETURN                                                           242
      END                                                              243

      SUBROUTINE ENDSNT (TSUBN)                                        244
C                                                                      245
C        PRINTS SMALLEST BREAKDOWN OF SUBNETWORK TSUBN                 246
C                                                                      247
      IMPLICIT INTEGER*2 (A-Z)                                         248
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS                  249
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)               250
      DIMENSION SINKS(100),NARCSS(100)                                 251
      SOURCE=SOURC(TSUBN)                                              252
      SINK=SINKS(TSUBN)                                                253
      WRITE (6,100) TSUBN                                              254
100   FORMAT (1HO,//16X,'SUBNETWORK ',I3,' IS A MINIMUM NETWORK',/16X,  255
     *'IT IS COMPOSED OF:')                                            256
      WRITE (6,200) SOURCE,SINK                                        257
200   FORMAT (1HO,19X,'SOURCE NODE = ',I3,//20X,'SINK NODE   = ',I3)   258
      M=NARCSS(TSUBN)                                                  259
      WRITE (6,400)                                                    260
400   FORMAT (1HO,19X,'ARC',2X,'S(ARC)',2X,'T(ARC)')                   261
      DO 10 I=1,M                                                      262
      N=SUBNET(I,TSUBN)                                                263
10    WRITE (6,300) N,S(N),T(N)                                        264
300   FORMAT (1H ,19X,I3,3X,I3,5X,I3)                                  265
C                                                                      266
C        SET TNSUB=0 SO THAT THE REMAINING NUMBER OF SUBNETWORKS DOESN'T  267
C        INCLUDE THIS MINIMUM SUBNETWORK                               268
C                                                                      269
      TNSUB=0                                                          270
      RETURN                                                           271
      END                                                              272

      SUBROUTINE STAGE (TYPESN,TSUBN)                                  273
C                                                                      274
C        PRINTS OUT THE CURRENT STAGE OF BREAKUP                       275
C                                                                      276
      IMPLICIT INTEGER*2 (A-Z)                                         277
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS                  278
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)               279
      DIMENSION SINKS(100),NARCSS(100)                                 280
C                                                                      281
C        TNSUB=THE NUMBER OF NEW SUBNETWORKS RESULTING FROM THE BREAKUP  282
C        IN THIS STAGE                                                 283
C                                                                      284
C        TNSUB=1 IMPLIES NO BREAKUP OCCURRED IN THIS STAGE             285
C                                                                      286
      IF (TNSUB.EQ.1) GO TO 600                                        287
      WRITE (6,100) TSUBN                                              288
100   FORMAT (1HO,/6X,'SUBNETWORK ',I3,' IS COMPOSED OF SUBNETWORKS: ')  289
      TSNSUB=NSUB-TNSUB                                                290
      M=TSNSUB+1                                                       291
      WRITE (6,300)(I,I=M,NSUB)                                        292
300   FORMAT (1HO,5X,20(I3,','))                                       293
      IF (TYPESN.EQ.1) GO TO 60                                        294
      WRITE (6,400)                                                    295
```

```
  400   FORMAT (1HO,5X,'IN SERIES')                                          296
        GO TO 600                                                            297
   60   WRITE (6,500)                                                        298
  500   FORMAT (1HO,5X,'IN PARALLEL')                                        299
  600   RETURN                                                               300
        END                                                                  301

        SUBROUTINE BUNDLE (LNODEN,TSUBN)                                     302
C                                                                            303
C          BUNDLE IDENTIFIES PARALLEL SUBNETWORKS CONNECTING DESIGNATED      304
C          SOURCE AND SINK                                                   305
C                                                                            306
        IMPLICIT INTEGER*2 (A-Z)                                             307
        COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS                      308
        DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100)                   309
        DIMENSION SINKS(100),NARCSS(100),BDNUM(300),CHECK(300)               310
C                                                                            311
C          GROUP NODES INTO BUNDLES                                          312
C                                                                            313
         SOURCE=SOURC(TSUBN)                                                 314
         SINK=SINKS(TSUBN)                                                   315
        NUMBD=1                                                              316
C                                                                            317
C          ZEROIZE THE BDNUM ARRAY                                           318
C                                                                            319
        DO 10 I=1,LNODEN                                                     320
    10  BDNUM(I)=0                                                           321
        K=SUBNET(1,TSUBN)                                                    322
        M=S(K)                                                               323
        N=T(K)                                                               324
        BDNUM(M)=1                                                           325
        BDNUM(N)=1                                                           326
        MAXND=N                                                              327
        IF (M.GT.N) MAXND=M                                                  328
        ARCS=NARCSS(TSUBN)                                                   329
        IF (ARCS.EQ.1) GO TO 515                                             330
        DO 1 K=2,ARCS                                                        331
        I=SUBNET(K,TSUBN)                                                    332
        BDNUM(SOURCE)=0                                                      333
        BDNUM(SINK)=0                                                        334
        M=S(I)                                                               335
        N=T(I)                                                               336
        IF(M.GT.MAXND) MAXND=M                                               337
        IF(N.GT.MAXND) MAXND=N                                               338
C                                                                            339
C          IF AT LEAST 1 NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A BUNDLE   340
C          GO TO 2                                                           341
C                                                                            342
        IF(BDNUM(M).EQ.0) GO TO 2                                            343
C                                                                            344
C          IF ONLY THE TERMINAL NODE ON THE ARC HAS NOT BEEN ASSIGNED TO A   345
C          BUNDLE, GO TO 3                                                   346
C                                                                            347
        IF(BDNUM(N).EQ.0) GO TO 3                                            348
C                                                                            349
C          IF BOTH NODES ON THE ARC HAVE BEEN ASSIGNED TO THE SAME BUNDLE    350
C          EVERYTHING IS OKAY, GO TRY ANOTHER ARC                            351
C                                                                            352
        IF(BDNUM(N).EQ.BDNUM(M)) GO TO 1                                     353
C                                                                            354
```

```
C          IF THE NODES ON THE ARC ARE ASSIGNED TO DIFFERENT BUNDLES,          355
C          THEN THESE TWO BUNDLES SHOULD BE POOLED                             356
C                                                                              357
       IF(BDNUM(N).LT.BDNUM(M)) GO TO 6                                        358
C                                                                              359
C          POOL BUNDLES                                                        360
C          THE BUNDLE WITH THE LARGER BUNDLE NUMBER IS POOLED INTO THE         361
C          BUNDLE WITH THE SMALLER BUNDLE NUMBER                               362
C          THE BUNDLE NUMBERS OF ALL BUNDLES ARE ALL ADJUSTED                  363
C                                                                              364
       MAXBD=BDNUM(N)                                                          365
       MINBD=BDNUM(M)                                                          366
       GO TO 7                                                                 367
     6 MAXBD=BDNUM(M)                                                          368
       MINBD=BDNUM(N)                                                          369
     7 DO 5 J=1,MAXND                                                          370
       B=BDNUM(J)                                                              371
       IF (B.EQ.MAXBD) BDNUM(J)=MINBD                                          372
       IF (B.GT.MAXBD) BDNUM(J)=BDNUM(J)-1                                     373
     5 CONTINUE                                                                374
       NUMBD=NUMBD-1                                                           375
       GO TO 1                                                                 376
C                                                                              377
C          IF BOTH NODES ON THE ARC ARE UNASSIGNED, GO TO 4 WHERE A NEW        378
C          BUNDLE IS CREATED                                                   379
C                                                                              380
     2 IF(BDNUM(N).EQ.0) GO TO 4                                               381
C                                                                              382
C          ASSIGN THE ORIGIN NODE OF THE ARC TO THE BUNDLE CONTAINING THE      383
C          TERMINAL NODE                                                       384
C                                                                              385
       BDNUM(M)=BDNUM(N)                                                       386
       GO TO 1                                                                 387
C                                                                              388
C          ASSIGN THE TERMINAL NODE OF THE ARC TO THE BUNDLE CONTAINING        389
C          THE ORIGIN NODE OF THE ARC                                          390
C                                                                              391
     3 BDNUM(N)=BDNUM(M)                                                       392
       GO TO 1                                                                 393
C                                                                              394
C          CREATE A NEW BUNDLE                                                 395
C                                                                              396
     4 NUMBD=NUMBD+1                                                           397
       BDNUM(M)=NUMBD                                                          398
       BDNUM(N)=NUMBD                                                          399
     1 CONTINUE                                                                400
   515 CONTINUE                                                                401
       BDNUM(SINK)=0                                                           402
C                                                                              403
C          IF WE ONLY HAVE 1 BUNDLE FROM THE SUBNETWORK, WE ARE FINISHED       404
C                                                                              405
       IF (NUMBD.EQ.1) GO TO 219                                               406
C                                                                              407
C          ZEROIZE CHECK ARRAY                                                 408
C                                                                              409
       DO 290 I=1,ARCS                                                         410
   290 CHECK(I)=0                                                              411
       L=0                                                                     412
C                                                                              413
C          THE NODES ARE IN BUNDLES.  PUT THE ASSOCIATED ARCS INTO             414
```

```
C          APPROPRIATE PARALLEL SUBNETWORKS                                415
C                                                                          416
      DO 33 I=1,NUMBD                                                      417
      SUMARC=0                                                             418
      NSUB=NSUB+1                                                          419
      DO 34 K=1,ARCS                                                       420
      M=SUBNET(K,TSUBN)                                                    421
      N=S(M)                                                               422
C                                                                          423
C          SOURCE AND SINK HAVE BUNDLE NUMBER 0                            424
C                                                                          425
      IF (N.EQ.SOURCE) N=T(M)                                              426
      IF (BDNUM(N).EQ.I) GO TO 239                                         427
      IF (N.EQ.SINK) GO TO 229                                             428
      GO TO 34                                                             429
C                                                                          430
C          SPECIAL CASE:  BUNDLE HAS ONLY 2 NODES: SOURCE, SINK.           431
C          PUT ALL ARCS THAT ARE PARALLEL SUBNETWORKS BY THEMSELVES INTO    432
C          THE CHECK ARRAY                                                 433
C                                                                          434
229   DO 291 J=1,K                                                         435
      W=CHECK(J)                                                           436
      IF (W.EQ.0) GO TO 292                                                437
      IF (M.EQ.W) GO TO 34                                                 438
291   CONTINUE                                                             439
292   CHECK(J)=M                                                           440
      GO TO 34                                                             441
C                                                                          442
C          THIS ARC IS IN THE BUNDLE I, HENCE IT IS IN THE ITH NEW         443
C          SUBNETWORK                                                      444
C                                                                          445
239   SUMARC=SUMARC+1                                                      446
      SUBNET(SUMARC,NSUB)=M                                                447
34    CONTINUE                                                             448
C                                                                          449
C          CREATE NEW SUBNETWORKS                                          450
C                                                                          451
C                                                                          452
C          IF THIS BUNDLE HAS NO NODES, PUT AN ARC SUBNETWORK INTO         453
C          SUBNET (1,NSUB)                                                 454
C                                                                          455
      IF (SUMARC.EQ.0) GO TO 333                                           456
343   NARCSS(NSUB)=SUMARC                                                  457
      SOURC(NSUB)=SOURCE                                                   458
      SINKS(NSUB)=SINK                                                     459
      GO TO 33                                                             460
C                                                                          461
C          STORE THE SUBNETWORKS THAT HAVE ONLY SOURCE AND SINK NODES      462
C                                                                          463
333   SUMARC=1                                                             464
      L=L+1                                                                465
      SUBNET(1,NSUB)=CHECK(L)                                              466
      GO TO 343                                                            467
33    CONTINUE                                                             468
219   TNSUB=NUMBD                                                          469
      RETURN                                                               470
      END                                                                  471

      SUBROUTINE CUT (TSUBN,LNODEN)                                        472
C                                                                          473
```

```
C          CUT IDENTIFIES CUT NODES EXCLUDING THE DESIGNATED SOURCE AND          474
C          SINK                                                                  475
C                                                                                476
C          CUT ALSO IDENTIFIES THE CUT GROUPS; THAT IS, THE SUBNETWORKS          477
C          WHICH ARE IN SERIES AND CONNECTED BY THE CUT NODES                    478
C                                                                                479
      IMPLICIT INTEGER*2 (A-Z)                                                   480
      COMMON NSUB,TNSUB,S,T,SUBNET,SOURC,SINKS,NARCSS                            481
      DIMENSION S(300),T(300),SUBNET(300,100),SOURC(100),TARC(300)               482
      DIMENSION SINKS(100),NARCSS(100),ORIGIN(300),POST(300),RCUT(100)           483
C                                                                                484
C          FIND THE CUT NODES                                                    485
C          NCUT IS THE NUMBER OF CUT NODES FOUND THUS FAR                        486
C                                                                                487
      NCUT=0                                                                     488
      ARCS=NARCSS(TSUBN)                                                         489
      SOURCE=SOURC(TSUBN)                                                        490
      SINK=SINKS(TSUBN)                                                          491
C                                                                                492
C          THE DO LOOP DOWN TO STATEMENT NUMBER 1 DETERMINES THE CUT NODES       493
C                                                                                494
      DO 1  K=1,LNODEN                                                           495
C                                                                                496
C          CHECK TO SEE IF NODE K IS ACTUALLY IN THE SUBNETWORK                  497
C                                                                                498
      DO 20 J=1,ARCS                                                             499
      Z=SUBNET(J,TSUBN)                                                          500
      IF(S(Z).EQ.K) GO TO 21                                                     501
      IF (T(Z).EQ.K) GO TO 21                                                    502
20    CONTINUE                                                                   503
C                                                                                504
C          NODE K IS NOT IN THIS SUBNETWORK                                      505
C                                                                                506
      GO TO 1                                                                    507
21    CONTINUE                                                                   508
C                                                                                509
C          NODE K IS IN THIS SUBNETWORK                                          510
C                                                                                511
      IF(K.EQ.SOURCE) GO TO 1                                                    512
      IF(K.EQ.SINK) GO TO 1                                                      513
      NTARC=0                                                                    514
      DO 2 J=1,ARCS                                                              515
      Z=SUBNET(J,TSUBN)                                                          516
      IF (S(Z).EQ.K) GO TO 2                                                     517
      IF (T(Z).EQ.K) GO TO 2                                                     518
      NTARC=NTARC+1                                                              519
      TARC(NTARC)=SUBNET(J,TSUBN)                                                520
2     CONTINUE                                                                   521
C                                                                                522
C          TARC IS THE SUBNETWORK WITHOUT THE ARCS INVOLVING NODE K              523
C          IF TARC CONTAINS A PATH FROM THE SOURCE TO THE SINK, THEN NODE        524
C          K IS NOT A CUT NODE                                                   525
C          OTHERWISE, K IS A CUT NODE                                            526
C                                                                                527
      ORIGIN(1)=SOURCE                                                           528
      NORIG=1                                                                    529
11    CONTINUE                                                                   530
      NPOST=0                                                                    531
C                                                                                532
C          IF THERE ARE NO ARCS IN THE TARC ARRAY, K IS A CUT NODE               533
```

```
C                                                                       534
      IF (NTARC.EQ.0) GO TO 44                                          535
C                                                                       536
C        FIND ALL NODES WHICH COME AFTER AN ORIGIN; PUT THEM IN POST     537
      DO 4 I=1,NORIG                                                    538
      DO 5 J=1,NTARC                                                    539
      Y=ORIGIN(I)                                                       540
      Z=TARC(J)                                                         541
      U=S(Z)                                                            542
      V=T(Z)                                                            543
      IF (U.NE.Y) GO TO 5                                               544
      IF (V.EQ.SINK) GO TO 1                                            545
C                                                                       546
C                                                                       547
C        IF WE'VE REACHED THE SINK, NODE K IS NOT A CUT NODE            548
C                                                                       549
C                                                                       550
      IF(NPOST.GE.1) GO TO 8                                            551
      NPOST=NPOST+1                                                     552
      POST(NPOST)=V                                                     553
      GO TO 5                                                           554
8     DO 9 L=1,NPOST                                                    555
C                                                                       556
C        IF THIS TERMINAL NODE IS ALREADY A POST, LET'S IGNORE IT       557
C                                                                       558
      IF (POST(L).EQ.V) GO TO 5                                         559
9     CONTINUE                                                          560
      NPOST=NPOST+1                                                     561
      POST(NPOST)=V                                                     562
5     CONTINUE                                                          563
4     CONTINUE                                                          564
C                                                                       565
C        IF THERE ARE NOW NO POSTS, NODE K IS A CUT NODE                566
C                                                                       567
44    IF(NPOST.NE.0) GO TO 13                                           568
      NCUT=NCUT+1                                                       569
      RCUT(NCUT)=K                                                      570
      GO TO 1                                                           571
13    NORIG=NPOST                                                       572
      DO 14 L=1,NORIG                                                   573
C                                                                       574
C        THESE ARE NOW OUR NEW ORIGINS                                  575
C                                                                       576
14    ORIGIN(L)=POST(L)                                                 577
C                                                                       578
C        CHECK THE NEW ORIGINS FOR THEIR POSTS                          579
C                                                                       580
      GO TO 11                                                          581
1     CONTINUE                                                          582
      IF (NCUT.EQ.0) GO TO 32                                           583
      NSUB=NSUB+1                                                       584
      SOURC(NSUB)=SOURCE                                                585
33    ORIGIN(1)=SOURCE                                                  586
C                                                                       587
C        NOW WE NEED TO FIND THE COMPONENTS OF THE SERIES SUBNETWORKS   588
C        THAT ARE SEPARATED BY THE CUT NODES                            589
C                                                                       590
39    SUMARC=0                                                          591
      NORIG=1                                                           592
23    NPOST=0                                                           593
```

```
C                                                                       594
C          ZEROIZE POST ARRAY                                           595
C                                                                       596
      DO 55 I=1,LNODEN                                                  597
55    POST(I)=0                                                         598
      DO 24 I=1,NORIG                                                   599
      Y=ORIGIN(I)                                                       600
      DO 25 J=1,ARCS                                                    601
      Z=SUBNET(J,TSUBN)                                                 602
C                                                                       603
C          ALL ARCS BEGINNING AT THIS ORIGIN GO INTO THE NEW SUBNETWORK  604
C                                                                       605
      IF (S(Z).NE.Y) GO TO 25                                           606
      SUMARC=SUMARC+1                                                   607
      SUBNET (SUMARC,NSUB)=Z                                            608
C                                                                       609
C          T(Z) WILL BE A NEW ORIGIN IF IT ISN'T A REPEAT OF A CURRENT  610
C          ORIGIN                                                       611
C                                                                       612
C          CHECK TO SEE IF IT IS A REPEAT                               613
C                                                                       614
      DO 30 K=1,NORIG                                                   615
      X=T(Z)                                                            616
C                                                                       617
C          IF T(Z) IS A REPEAT OF A CURRENT ORIGIN, LET'S IGNORE IT     618
C                                                                       619
      IF(X.EQ.ORIGIN(K)) GO TO 25                                       620
30    CONTINUE                                                          621
      NPOST=NPOST+1                                                     622
      POST(NPOST)=X                                                     623
C                                                                       624
C          CHECK TO SEE IF ANY POST IS A CUT NODE                       625
C          IF IT IS, REPLACE IT WITH A ZERO                             626
C                                                                       627
      DO 52 W=1,NCUT                                                    628
      D=RCUT(W)                                                         629
      IF (X.NE.D) GO TO 52                                              630
      POST(NPOST)=0                                                     631
      NPOST=NPOST-1                                                     632
C                                                                       633
C          THIS CUT NODE IS THE SINK OF THE SUBNETWORK UNDER            634
C          CONSIDERATION AND THE SOURCE OF THE NEXT SUBNETWORK TO BE    635
C          CONSIDERED.                                                  636
C                                                                       637
      SINKS(NSUB)=D                                                     638
      SOURC(NSUB+1)=D                                                   639
C                                                                       640
C          CHECK TO SEE IF ANY POSTS ARE REPEATED IN THE POST ARRAY.  IF 641
C          THEY ARE, REDUCE THE NUMBER OF POSTS TO WHERE THERE ARE NO   642
C          REPEATS                                                      643
C                                                                       644
52    CONTINUE                                                          645
C                                                                       646
C          IF WE HAVE ONE OR LESS POSTS, THERE ARE NO ADJUSTMENTS OF THE 647
C          POST ARRAY TO BE MADE; LET'S CONTINUE                        648
C                                                                       649
53    IF (NPOST.LE.1) GO TO 25                                          650
      POSTCK=NPOST-1                                                    651
      DO 7 K=1,POSTCK                                                   652
      IF (POST(K).EQ.POST(NPOST)) GO TO 61                              653
```

```
7       CONTINUE                                                        654
        GO TO 25                                                        655
61      POST(NPOST)=0                                                   656
        NPOST=NPOST-1                                                   657
25      CONTINUE                                                        658
24      CONTINUE                                                        659
C                                                                       660
C          IF WE HAVE NO POSTS LEFT, WE HAVE FOUND ALL OF THIS SUBNETWORK 661
C                                                                       662
        IF (NPOST.EQ.0) GO TO 34                                        663
        NORIG=NPOST                                                     664
        DO 28 L=1,NORIG                                                 665
28      ORIGIN(L)=POST(L)                                               666
        GO TO 23                                                        667
34      NARCSS(NSUB)=SUMARC                                             668
        NSUB=NSUB+1                                                     669
        X=SOURC(NSUB)                                                   670
C                                                                       671
C          IF THE SOURCE OF NSUB IS NOT A CUT NODE, WE NEED TO ADJUST NSUB 672
C          AND GO BACK TO THE MAIN PROGRAM FOR THE NEXT STAGE OF THE    673
C          BREAKUP                                                      674
C                                                                       675
        IF (X.EQ.0) GO TO 31                                            676
        ORIGIN(1)=X                                                     677
        GO TO 39                                                        678
31      NSUB=NSUB-1                                                     679
        SINKS(NSUB)=SINK                                                680
32      TNSUB=NCUT+1                                                    681
        RETURN                                                          682
        END                                                             683
```

LOOP

This short program will determine whether a given network is acyclic or contains loops (cycles). The program examines each node and indicates whether or not the node is part of a loop. If a node is part of a loop, the number of activities in the loop is also indicated.

The basic steps in determining whether or not the INODE-th node is part of a loop are as follows:

(1) Identify all activities whose termanal node is the INODE-th node. Let A be the set of all origin nodes for these activities.

(2) If INODE is in A, the INODE-th node is part of a loop and stop.

(3) Identify all activities whose terminal node is in A. Redefine A to be the set of origin nodes for these activities. If A is now empty, stop and the INODE-th node is not part of a loop. If A is not empty, return to step 2.

Specific Input Instructions:

Card 1. Col. 1-3: The number of activities in the network, Format I3).
Col. 4-6: The largest node number in the network, Format (I3).

For each activity one card with:

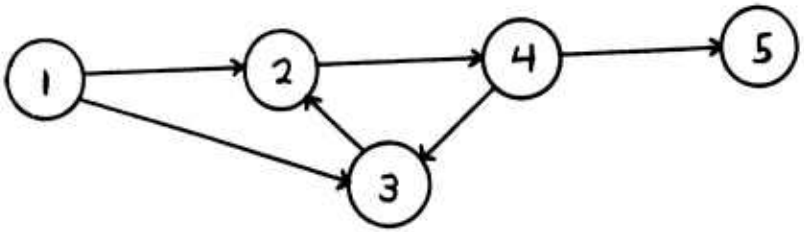
Col. 1-3: The activity's origin node number, Format (I3).
Col. 4-6: The activity's terminal node number, Format (I3).

The activities may be inputted in any order. The nodes may be numbered in any manner; however, the program is more efficient if the set of node numbers contains only the numbers 1 through N where N is the number of nodes in the network.

Dimension Restrictions:

This program is written in FORTRAN G. The current dimensions will allow a network with 300 activities and 200 nodes to be considered.

SAMPLE NETWORK



SAMPLE INPUT

```
6   5
1   2
1   3
4   3
3   2
2   4
4   5
```

SAMPLE OUTPUT

Activity	Tail	Head
1	1	2
2	1	3
3	4	3
4	3	2
5	2	4
6	4	5

```
Node    1 is not looped
Node    2 is looped.   There are 3 activities in the loop.
Node    3 is looped.   There are 3 activities in the loop.
Node    4 is looped.   There are 3 activities in the loop.
```

```
C         PROGRAM LOOP                                                          1
C                                                                               2
C         THIS PROGRAM DETERMINES WHETHER THE INPUTTED NETWORK IS ACYCLIC       3
C         OR CONTAINS NODES WHICH ARE PART OF LOOPS (CYCLES)                    4
C                                                                               5
      IMPLICIT INTEGER*2 (A-Z)                                                  6
      DIMENSION A( 200),B( 200),HEAD(300),TAIL(300)                             7
C                                                                               8
C         THE ARRAY DIMENSIONS ARE: A(N),B(N),HEAD(M),TAIL(M)                   9
C         WHERE                                                                10
C                   N = THE NUMBER OF NODES IN THE NETWORK                     11
C                   M = THE NUMBER OF ACTIVITIES IN THE NETWORK                12
C                                                                              13
      ILOOP=0                                                                  14
      READ (5,100) M,NMM                                                       15
  100 FORMAT (2I3)                                                             16
      READ(5,101) (TAIL(I),HEAD(I),I=1,M)                                      17
101   FORMAT(2I3)                                                              18
      WRITE(6,2001)                                                            19
 2001 FORMAT(1H1)                                                              20
      WRITE(6,2000) (I,TAIL(I),HEAD(I),I=1,M)                                  21
 2000 FORMAT(' ACTIVITY      TAIL      HEAD'/,(4X,I3,7X,I3,6X,I3))             22
      WRITE(6,2001)                                                            23
C     FORM THE 1ST HIERARCHY                                                   24
      INODE = 0                                                                25
   80 HIER = 2                                                                 26
      INODE = INODE + 1                                                        27
      J = 0                                                                    28
      DO 1 I= 1,M                                                              29
      IF(HEAD(I).NE.INODE) GO TO 1                                             30
      J = J+1                                                                  31
      A(J)=TAIL(I)                                                             32
      IF (TAIL(I).EQ.INODE) GO TO 998                                          33
    1 CONTINUE                                                                 34
      IF (J.EQ.0) GO TO 997                                                    35
      IA=J                                                                     36
      J=0                                                                      37
C     FORM THE SUBSEQUENT HIERARCHIES                                          38
  102 CONTINUE                                                                 39
      DO 2 II=1,IA                                                             40
      DO 3 I=1,M                                                               41
      IF (HEAD(I).NE.A(II)) GO TO 3                                            42
      IF(TAIL(I).EQ.INODE) GO TO 998                                           43
      IF(J.EQ.0) GO TO 40                                                      44
      DO 10 K=1,J                                                              45
      IF(TAIL(I).EQ.B(K)) GO TO 11                                             46
10    CONTINUE                                                                 47
40    CONTINUE                                                                 48
      J=J+1                                                                    49
      B(J)=TAIL(I)                                                             50
11    CONTINUE                                                                 51
    3 CONTINUE                                                                 52
    2 CONTINUE                                                                 53
      IF (J.EQ.0) GO TO 997                                                    54
      HIER = HIER+1                                                            55
      IA=J                                                                     56
      J = 0                                                                    57
      DO 20 I=1,IA                                                             58
20    A(I)=B(I)                                                                59
```

```
      GO TO 102                                                          60
  997 CONTINUE                                                           61
      WRITE(6,2002) INODE                                                62
 2002 FORMAT(' NODE',I5,' IS NOT LOOPED' )                               63
      IF (INODE.NE.NMM ) GO TO 80                                        64
      IF(ILOOP.EQ.1) GO TO 50                                            65
      WRITE(6,1000)                                                      66
 1000 FORMAT (' THERE ARE NO LOOPS IN THIS NETWORK')                     67
      GO TO 999                                                          68
   50 WRITE(6,51)                                                        69
   51 FORMAT(' THERE ARE NO OTHER LOOPS IN THIS NETWORK')                70
      GO TO 999                                                          71
  998 WRITE (6,1001) INODE,HIER                                          72
 1001 FORMAT (' NODE ' ,I5,' IS LOOPED. THERE ARE ',I3,' ACTIVITIES IN T 73
     *HE LOOP.')                                                         74
      ILOOP=1                                                            75
      INDDD=INODE+1                                                      76
      IF(INDDD.NE.NMM) GO TO 80                                          77
  999 CONTINUE                                                           78
      WRITE(6,2001)                                                      79
      STOP                                                               80
      END                                                                81
```

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Texas A&M University

2a. REPORT SECURITY CLASSIFICATION

unclassified

2b. GROUP

unclassified

3. REPORT TITLE

(6) STATISTICAL PERT: DECOMPOSING A PROJECT NETWORK

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Progress Report

5. AUTHOR(S) (First name, middle initial, last name)

(10) R. L. Sielken, Jr. and N. E. Fisher

(12) 44 p.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
(11) Nov[illegible]975	37	0

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
(15) N00014-68-A-0140	Number 50
b. PROJECT NO. NR047-700	(14) THEMIS-TR-50
c. (16)	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research

13. ABSTRACT

See page following Table of Contents

(9) THEMIS optimization research program,

347 380

ATTACHMENT III

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Statistical PERT						
Project Scheduling						
Network Decomposition						
ATTACHMENT III (Continued)						

BASIC DISTRIBUTION LIST FOR UNCLASSIFIED TECHNICAL REPORTS

	Copies
Operations Research Office of Naval Research (Code 434) Arlington, Virginia 22217	3
Director, Naval Research Laboratory Attn: Library, Code 2029 (ONRL) Washington, D.C. 20390	2
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Defense Logistics Studies Information Exchange Army Logistics Management Center Fort Lee, Virginia 23801	1
Technical Information Center Naval Research Laboratory Washington, D.C. 20390	6
Office of Naval Research New York Area Office 207 West 24th Street New York, New York 10011 Attn: Dr. J. Laderman	1
Director, Office of Naval Research Branch Office 1030 East Green Street Pasadena, California 91101 Attn: Dr. A.R. Laufer	1
Lt. Dennis Johnston - ED3 Manned Spacecraft Center NASA Houston, Texas 77058	1
Mr. William Dejka Code 360 Naval Electronics Laboratory Center San Diego, California 92132	1
Professor Lucien A. Schmit, Jr. Room 6731, Boelter Hall School of Engrng. & Applied Sci. University of California Los Angeles, California 90024	1

	Copies
Director Office of Naval Research Branch Office 495 Summer Street Boston, Massachusetts 02210 Attn: Dr. A.L. Powell	1
Director Office of Naval Research Branch Office 219 South Dearborn Street Chicago, Illinois 60604 Attn: Dr. A.R. Dawe	1
Office of Naval Research San Francisco Area Office 760 Market St. - Room 447 San Francisco, California 94103	1
Technical Library Naval Ordnance Station Indian Head, Maryland 20640	1
Bureau of Naval Personnel Navy Department Technical Library Washington, D.C. 20370 STOP 82	2
Library, Code 0212 Naval Postgraduate School Monterey, California 93940	1
Naval Ordnance Station Louisville, Kentucky 40214	1
Library Naval Electronics Laboratory Center San Diego, California 92152	1
Naval Ship Engineering Center Philadelphia Division Technical Library Philadelphia, Pennsylvania 19112	1
Dr. A.L. Slafkosky Scientific Advisor Commandant Marine Corps (Code AX) Washington, D.C. 20380	1

	Copies
Purdue University Graduate School of Ind. Admin. Lafayette, Indiana 47907 Attn: Prof. Andrew Whinston	1
Department of Economics Tisch Hall - 5th Floor Washington Square, NY Univ. New York City, NY 10003 Attn: Prof. O. Morgenstern	1
Dept. of Statistics Syntex Research 3401 Hillview Palo Alto, CA 94304 Attn: Prof. Stuart Bessler	1
University of California Department of Engineering Los Angeles, California 90024 Attn: Prof. R.R. O'Neill	1
Maritime Transportation Research Board National Academy of Sciences 2101 Constitution Avenue Washington, D.C. 20418 Attn: RADM J.B. Oren USCG (RET)	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. A.F. Veinott, Jr.	1
Texas A&M Foundation College Station, Texas 77843 Attn: Prof. H.O. Hartley	1
Case Western Reserve University Cleveland, Ohio 44106 Attn: Prof. B.V. Dean	1
University of California Center for Research in Management Science Berkeley, California 94720 Attn: Prof. W.I. Zangwill	1
U.S. Naval Postgraduate School Department of Operations Research and Economics Monterey, California 93940 Attn: Prof. C.R. Jones	

	Copies
Dr. Del Cilmer Naval Weapons Center (Code 607) China Lake, California 93555	1
Dr. Paul Murrill Project Themis Department of Chemical Engineering Louisiana State University Baton Rouge, Louisiana 70803	1
Director Army Materials & Mechanics Research Center Attn: Mr. J. Bluhm Watertown, Massachusetts 02172	1
Commanding Officer U.S. Army Ballistic Research Laboratories Attn: Dr. John H. Giese Aberdeen Proving Ground, Maryland 21005	1
University of Iowa Department of Mechanics and Hydraulics Iowa City, Iowa 52240 Attn: Prof. W.F. Ames	1
Office of Naval Research Resident Representative 1105 Guadalupe Lowich Building Austin, Texas 78701 Attn: Mr. Francis M. Lucas	1
Dr. Jerome Bracken Institute of Defense Analyses 400 Army-Navy Drive Arlington, Virginia 22202	1
Professor Richard L. Fox School of Engineering Case Western Reserve University Cleveland, Ohio 44106	1
Director, National Security Agency Ft. George G. Meade, Maryland 20755 Attn: Dr. Joseph Blum, R44	1

	Copies
Naval Applied Science Laboratory Technical Library, Code 222 Flushing & Washington Avenues Brooklyn, New York 11251	1
Naval Undersea Warfare Center 3202 E. Foothill Boulevard Pasadena, California 91107 Technical Library	1
University of Chicago Statistical Research Center Chicago, Illinois 60637 Attn: Prof. W. Kruskal	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. G.J. Lieberman	1
Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. I.R. Savage	1
Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. J.W. Tukey	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. T.W. Anderson	1
University of California Department of Statistics Berkeley, California 94720 Attn: Prof. P.J. Bickel	1
University of Washington Department of Mathematics Seattle, Washington 98105 Attn: Prof. Z.W. Birnbaum	1
Cornell University Department of Mathematics White Hall Ithaca, New York 14850 Attn: Prof. J. Kiefer	1

	Copies
Harvard University Department of Statistics Cambridge, Massachusetts 02139 Attn: Prof. W.G. Cochran	1
Columbia University Department of Industrial Engineering New York, New York 10027 Attn: Prof. C. Derman	1
New York University Institute of Mathematical Sciences New York, New York 10453 Attn: Prof. W.M. Hirsch	1
University of North Carolina Statistics Department Chapel Hill, North Carolina 27515 Attn: Prof. W.L. Smith and Prof. M.R. Leadbetter	1
Purdue University Division of Mathematical Sciences Lafayette, Indiana 47079 Attn: Prof. H. Rubin	1
University of California, San Diego Department of Mathematics P.O. Box 109 La Jolla, California 92038 Attn: Prof. M. Rosenblatt	1
Florida State University Department of Statistics Tallahassee, Florida 32306 Attn: Prof. R.A. Bradley	1
New York University Department of Industrial Engineering & Operations Research Bronx, New York 10453 Attn: Prof. J.H.K. Kao	1
University of Wisconsin Department of Statistics Madison, Wisconsin 53706 Attn: Prof. G.E.P. Box	1

	Copies
Logistics Research Project The George Washington University 707 - 22nd Street, N.W. Washington, D. C. 20037 Attn: Dr. W. H. Marlow	1
International Business Machines Corporation P.O. Box 218, Lamb Estate Yorktown Heights, New York 10598 Attn: Dr. Alan Hoffman	1
University of California Management Sciences Research Project Los Angeles, California 90024 Attn: Dr. J.R. Jackson	1
Harvard University Department of Economics Cambridge, Massachusetts 02138 Attn: Prof. K.J. Arrow	1
Cowles Commission for Research in Economics Yale University New Haven, Connecticut 06520 Attn: Prof. Martin Shubik	1
Carnegie-Mellon University Graduate School of Industrial Administration Pittsburgh, Pennsylvania 15213 Attn: Prof. G. Thompson	1
University of California Department of Economics Berkeley, California 94720 Attn: Prof. R. Radner	1
University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. D. Gale	1
University of California Graduate School of Business Administration Los Angeles, California 90024 Attn: Prof. J. Marschak	1
The University of Michigan Department of Mathematics, W.E. Ann Arbor, Michigan 48104 Attn: Prof. R.M. Thrall	1
Princeton University Department of Mathematics Princeton, New Jersey 08540 Attn: Prof. A.W. Tucker	1
Case Western Reserve University Systems Research Center Cleveland, Ohio 44106 Attn: Prof. M. Mesarovic	1
University of Texas Department of Mathematics Austin, Texas 78712 Attn: Dr. A. Charnes	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Dr. D.L. Iglehart	1
University of Delaware Department of Mathematics Newark, Delaware 19711 Attn: Prof. R. Remage, Jr.	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. F.S. Hillier	1
Dr. Claude-Alain Burdet Asst. Prof. Industrial Admin. Carnegie-Mellon University Pittsburgh, Pennsylvania 15213	1
Stanford University Department of Operations Research Stanford, California 94305 Attn: Prof. G. B. Dantzig	1
Chief of Naval Research (Code 436) Department of the Navy Arlington, Va. 22217	1
Science Librarian Kresge Library Oakland University Rochester, Michigan 48063	1

	Copies
University of Connecticut Department of Statistics Storrs, Connecticut 06268 Attn: Prof. H.O. Posten	1
Prof. Emanuel Parzen Statistics Department State Univ. New York at Buffalo 4230 Ridge Lea Road Amherst, New York 14226	1
ARCON Corporation Lakeside Office Park North Avenue at Route 128 Wakefield, Massachusetts 01880 Attn: Dr. A. Albert	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Chernoff	1
Yale University Department of Statistics New Haven, Connecticut 06520 Attn: Prof. L.A. Savage	1
Rutgers-The State University Statistics Center New Brunswick, New Jersey 08903 Attn: Prof. H.F. Dodge	1
Yale University Department of Statistics New Haven, Connecticut 06520 Attn: Prof. F. J. Anscombe	1
Purdue University Division of Mathematical Sciences Lafayette, Indiana 47907 Attn: Prof. S.S. Gupta	1
Cornell University Department of Industrial Engineering Ithaca, New York 19850 Attn: Prof. R.E. Bechhofer	1
Mrs. Barbara Baudi Univ. Program Coordinator, B.E. NASA Johnson Space Center Houston, TX 77058	1
Professor Geoffrey S. Watson, Chairman Department of Statistics Princeton University Fine Hall Princeton, N. J. 08540	1
Stanford University Department of Mathematics Stanford, California 94305 Attn: Prof. S. Karlin	1
University of Sheffield Department of Probability and Statistics Sheffield 10, ENGLAND Attn: Prof. J. Gani	1
University of California Operations Research Center Institute of Engineering Research Berkeley, California 94720 Attn: Prof. R.E. Barlow	1
Stanford University Department of Statistics Stanford, California 94305 Attn: Prof. H. Solomon	1
Applied Mathematics Laboratory Naval Ships Research Development Center Washington, D.C. 20007	1
Systems Analysis Division Room BE760, Pentagon Washington, D.C. 20350 Attn: Mr. A.S. Rhodes, Op-964	1
Department of Statistics University of North Carolina Chapel Hill, North Carolina 27515 Attn: Prof. M.R Leadbetter	1
Southern Methodist University Department of Statistics Dallas, Texas 75222 Attn: Prof. D.B. Owen	1
Israel Institute of Technology Technion Haifa, ISRAEL Attn: Prof. P. Naor	

Office of Gifts & Exchanges	Copies
Library	
Texas A&M University	
College Station, Texas 77843	2
Archives	
Texas A&M University	
College Station, TX 77843	1