A computer program using FORTRAN IV for determining all those paths in a reliability logic graph of the system, which allow the required capacity of the information contents to flow, from the source to the terminal node is appended here in this section.

The nomenclature used in this program is:

- NB Number of branches
- NP Number of paths from source to terminal
- BC Branch capacities
- SC Required system capacity

The source listing, including the data and print out for a test Example 3.4 is given. Keeping in view the size for this example, it is assumed that:

(i) Number of branches in the graph is less than 10; if not true, the dimensions of BC, P1, P2, P3 and P4 are changed accordingly.

(ii) Number of minimal s-t paths in the graph are less than 20; if not true, the dimensions of P1, P2, P3, P4 and COMB are also to be changed accordingly.

(iii) Capacity of each branch does not exceed 9; if not true, CP1(I) is changed accordingly.

These assumptions do not subscribe any loss of generality, since for any other example, these minor changes can be made.
C CREATING CAPACITIES OF SIMPLE PATHS AND TRANSFER VALID PATHS TO P4 AND C AND TO VALIDP TO P2

INTEGER RC(10)
P1(20,10),P2(20,10),P3(100,10),P4(100,10)
1 GO(20),CP2(20),CP3(100),CP4(100),CMH(10),POS,CAP,CAP1,SC,
1 PER
DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
1 DATA/265*/
C DATA MC, P1, P2, P3, P4, CP1, CP2, CP3, CP4, CMH, POS, CAP, CAP1,
IF (CAP1 .EQ. CAP) CAP = CAP1
GO TO 30 10 13 = 1, k3
POS = 0
DIF = P3(I3,J) - COMB(J)
IF (DIF .EQ. 0) POS = POS + 1
CONTINUE
IF (POS = MB) GO TO 300
CONTINUE
K3 = K3 + 1
DO 370 J = 1, MB
CP3(03, J) = COMB(J)
GO TO 300
CONTINUE
IF (K4 = 0) GO TO 2000
DO 350 I4 = 1, K4
POS = 0
DIF = P4(14, J) - COMB(J)
IF (DIF .EQ. 0) POS = POS + 1
CONTINUE
IF (POS .LE. 0) GO TO 300
CONTINUE
2000 CONTINUE
K4 = K4 + 1
DO 360 J = 1, MB
CP4(K4, J) = COMB(J)
GO TO 390
CONTINUE
K3 = K3 + K3
DO 390 I6 = 1, K3
CP3(I6) = CP3(I6 + K3)
DO 390 J = 1, MB
370 CP3(I6, J) = CP3(I6 + K3, J)
500 CONTINUE
600 CONTINUE
PRINT 77
FORM = 14, 20X, 'LISTING OF CAPACITIES OF PATHS AND TRANSFERS')
PRINT 56, 'CAP, IN, NP, K2, K3, K4, SC'
DO 90 I = 1, MB
PRINT 65, ((P1(I, J), J = 1, MB), CP1(I), I = 1, NP)
PRINT 56, ((P2(I, J), J = 1, MB), CP2(I), I = 1, K2)
PRINT 56, ((P3(I, J), J = 1, MB), CP3(I), I = 1, K3)
PRINT 56, ((P4(I, J), J = 1, MB), CP4(I), I = 1, K4)
STOP
END

Data of this listing refers to Example 3.4
APPENDIX - A-II

The program appended in this section uses FORTRAN IV to determine the optimal solution of a computer communication network having knowledge of the link costs, reliabilities and their locations with respect to the various nodes. The maximum permissible cost is also assumed to be known. The computer program does the following major tasks:

(i) Generation of path matrix.
(ii) Determines the cost of every path.
(iii) Determines the reliability of every path.
(iv) Determines the ratio of the path reliability to path cost and chooses the path with maximum of this ratio.
(v) Sequentially adds the various branches in the selected path with maximum reliability-cost ratio and determines every time the reliability of the augmented system and the cost consumed till the permissible cost is exhausted.

Most of the communication between the main program and the subroutines has been handled through COMMON declarations. The glossary of the various terms are provided in the computer listing itself.
MAIN PROGRAM

TOPOLOGICAL LAYOUT OF LINKS FOR OPTIMIZING TERMINAL RELIABILITY

NB: NUMBER OF BRANCHES: NPI: NUMBER OF PATHS

B: BRANCH COST, BPI: BRANCH RELIABILITY

SYSCOS: SYSTEM COST, CSi: CONSTRAINT ON COST, PP PATH RELIABILITY MATRIX

PREFI:SYSTEM RELIABILITY(CURRENT) :PCOST(i): ITH PATH COST

PREFLY(I): ITH PATH RELIABILITY

P(I,J) RELIABILITY TO COST PATH OF JTH PATH

P(I,J) PATH MATRIX WITH ZFRO. ONF ENTERIES

THK: NUMBER OF MISSING BRANCHES IN PATH K

INTEGER P1,PX,TPX,PORP,RCOMB,PBAR
COMMON/A/ WP,VR
COMMON/C/SYRCOS
COMMON/C/SAEL,BR
COMMON/E/P,PREP,TPR,
COMMON/E/TPX,TPX,IPX
COMMON/F/PORP,RPO
COMMON/G/PI,RCOMB
COMMON/H/MB,MB

DIMENSION RCOST(IO), PRELY(IO), BRI(IO), D(I0), MBT(2,10), TPX(10,10)

1, BCI(O), DFLATD(10), PRL(10), RCHEM(10), P0(10), P1(10), P0X(10,10)

DATA (IP1(i,1),IJ=1,10),IJ=1,8)/1.O,0.1,0.0,1,0.0,0.0,1,0.1,0.1,0.1,0.1,0.1

1.0,1.1,1.0,1.1,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0

DATA (RCOL1),IJ=1,8)/2.5,3.6,4.3,4.3,3/

DATA (RRI(I),IJ=1,8)/4,7,8,6,9,8,7,8/

TYPE 15

15 FORMAT(1X,'INPUT CS')

ACCPT8 CS

TYPE 17 CS

PRINT 17 CS

17 FORMAT(1X,'CS=IPR,7/)

DATA P0,TPX,RCOMB/210,20/

GENERATION OFPCOST(I), I=1,NP AND

PREFLY(I), I=1, NP,PCOST(I)=COST OF PATH IN P(I,J), D(I)=PREFLY(I)/PCOST(I) FOR ALL I

DO 10 I=1, NP

PCOST(I)=0.

PREFLY(I)=1.

D(I)=0

20 CONTINUE

DO 30 I=1, NP

IF (P(I,I),EQ,0) GOTO 20

PCOST(I)=PCOST(I)+BC(I)

PREFLY(I)=PREFLY(I)+RR(I)

30 CONTINUE

THIS SEGMENT DETERMINES MAXIMUM VALUE D(K) FROM D(I) FOR ALL I

DMAX=0.0

DO 40 I=1, NP

IF (D(I),LE,DMAX) GO TO 30

40 CONTINUE

K = I

44 CONTINUE
CONTINUE
IF(K.EQ.0).GO TO 888
PRINT SYSTEM CANNOT BE DESIGNED
SYSREL = PRELTY(K)
TF(CS=SYSCS) 11,22,33
D(K)=0.

THE ALLOCATED COST OF SYSTEM IS LESS THAN THE COST REQUIRED FOR P1
THUS CHOOSE NEXT BEST PATH
GO TO 899

THE COST ALLOCATED IS EQUAL TO THE COST OF KTH PATH ; STOP
GO TO 99

THE COST OF KTH PATH THEREFORE INCLUDE BRANCHES TO GENERATE MORE
PATHS THEREFORE INCREASING RELIABILITY
CONTINUE
DO 100 J=1,NB
BCOMB(J)=PI(K,J)
100 NOPG(1,J)=I(K,J)
NB=NB+1
THIS SEGMENT GENERATES MBT HAVING TWO ROWS WHERE MBT(1,J) IS
EQUAL TO SERIAL NUMBERS OF MISSING BRANCHES AND MBT(2,J) EQUALS
THE COST OF THAT BRANCH MBT IS 2X IMA MATRIX
I=0
DO 40 J=1,NB
TF(PI(K,J).NE.O).GOTO 40
I=I+1
MBT(1) = J
MBT(2) = RC(J)
40 CONTINUE
MBR(I) = 1

CALL APANGE
IF(TM.EQ.0).GO TO 777.

PICK ONE BRANCH AT A TIME FROM FINAL
MBT ADD TO PI(K,J) AND DETERMINE WHETHER VALID PATH EXISTS
COMPUTE DELTAT(1) = DELTAC(1) AND DELTAD(1)
DO 110 I=1,NB
TRFL(1) = 0; DELTAD(I) = 0; I1=MBT(1)1;
BCOMB(I1) = 1
CALL VLDPTW(K)
110 CONTINUE
IF(NPO.EQ.WPOLD) GO TO 101
CALL DTSMNG
CALL RELRTY
TRFL(1) = SRFL
SRFL IS VALUE OF RELIABILITY RETURNED BY SUBROUTINE RELRTY
DELTAD(1) = (SRFL-SYSREL)/MBT(2,1)
SYSREL IS RELIABILITY OF SYSTEM SO FAR DESIGNED
BCOMB(I1) = 0
101 CONTINUE
IM = 1
.DELMAX = 0.0
DO 120 I=1,NB
TF(DELTAD(I).LE.DELMAX) GO TO 120
IM = I
DELTAD(I) = DELTAD(I) + ITM = I
120 CONTINUE
IF(IM.EQ.0) IM=1
SUBROUTINE ARANGE

THIS SEGMENT OF MBT ARANGES MBT IN ASCENDING ORDER OF COST THAT IS MBT (2, J) INTEGER P, P, PX, TPX, POPG, BCOMB, PRAR COMMON /B/ CS, SYSCOS, SYSRFL, (BCOMB(I), I = 1, NB), PRMT, 45, CS, SYSCOS, SYSRFL, (BCOMB(I), I = 1, NB)
45 FORMAT (1X, 1CS='FR.5, 1', SYSCOS='FR.5, 2X, 1', SYSRFL='FR.7/1Y, 1') !COMB='G0T/1/>
123 FORMAT (1X, 'OPTIMAL SYSTEM DESIGNED HAVING BRANCHES THAT OF BCOMBI') STOP
888 TYPE 234 STOP
234 FORMAT (1X, 'COST OF EACH PATH EXCEEDS THE COST CONSTRAINT SYSTEM CANNOT BE DESIGNED') STOP TYPE 345
345 FORMAT (1X, 'SYSTEM IS OPTIMAL. IT CONTAINS ONLY ONE PATH') GO TO 555 STOP END

SURROUNTE ARANGE
SUBROUTINE VLDPTH(K)

THIS SUBROUTINE WILL TEST WHETHER VALID PATHS WILL BE MADE BY BRANCHES CONTAINED IN VECTOR BCOMB(J), J = 1, NB AND PUT THE VALID PATHS IN PORG(I,J), I = 1, NPO; J = 1, NB

INTEGER PI, PX, TPX, PORG, BCOMB, PBAR
COMMON /A/ NP, NB
COMMON /G/ P1, BCOMB
DIMENSION BCOMB(10), P1(10,10), PORG(10,10)

NP = 0
DO 10 J = 1, NP
   CONTINUE
   NP = NP + 1
DO 20 J = 1, NB
   CONTINUE
   RETURN
END

SUBROUTINE PELBTY

THIS SUBROUTINE DETERMINES RELIABILITY OF A SYSTEM WHOSE MUTUALLY DISJOINTED SUCCESS TERMS ARE IN PX(I,J)

INTEGER PI, PX, TPX, PORG, BCOMB, PBAR
COMMON /A/ NP, NB
COMMON /C/ SRFL, BR
COMMON /E/ PX, TPX, TPX, TTPX
DIMENSION PX(10,10), BR(10)

IF (PX(I,J) .EQ. 0) GO TO 20
IF (PX(I,J) .EQ. 1) REL = REL + REL(REL(J))
CONTINUE
SRFL = SRFL + REL.

SUBROUTINE DISINT

THIS SUBROUTINE IS FOR GENERATING MUTUALLY EXCLUSIVE TERMS (PX(I,J), I = 1, TPX, J = 1, NB) FROM THE GIVEN ORIGINAL PATHS PX ARE GENERATED FROM PORG BY ALGORITHM PX = P1*(P2+P2'+...+PN-1(PN)))

INNERMOST EXPRESSIONS IS SIMPLIFIED AT ONE TIME

INTEGER PI, PX, TPX, PORG, BCOMB, PBAR
COMMON /A/ NP, NB
COMMON /G/ P1, PBP, PBB, PRR
COMMON /E/ PX, TPX, TPX, TTPX
DIMENSION PX(10,10), PORG(10,10), PBP(10,10), PBB(10,10), TTPX(10,10)

DO 10 J = 1, NB
   CONTINUE
   RETURN
END
CALL BAR
THTS DETERMINES PBAR(I,J), GIVEN P(I,J)
CALL STMPF
THTS FORMS (P(I,J), P(X)) AS TPX
CALL MINMIZE
THTS CHANGES P+TPX PX
20 CONTINUE
RETURN; END

SUBROUTINE MINIMIZE THE PRODUCT TERMS AND STORES IN TPX(I,J)
I = 1, TPX; J=1,NB AND TRANSFERS THESE TERMS TO PX(I,J), T=1, TPX
J=1,NB; PX(I,J)=P(I,J) FOR ALL J.
COMMON /A/ NB, IP
COMMON /P, PBAR, IPB
COMMON /IPX, TPX, TPX, ITPX
INTEGER ZERO, PX, TPX, DIFF, PONF, P1, P, PORQ, PBAR, RCOME
DIMENSION PX(10,10), TPX(10,10), P(10)
10 PX(I,J)=P(I,J)
20 IPX=I
30 T=I, ITPX
40 PX(IPX,J)=PX(IPX,J)
50 CONTINUE.
60 CONTINUE.
70 PX(IPX,J)=TPX(IPX,J)
80 CONTINUE.
90 CONTINUE.
100 CONTINUE.
110 CONTINUE.
120 CONTINUE.
130 CONTINUE.
SUBROUTINE BAR

THIS DETERMINES P'(J) AND STORES THE SUM OF PRODUCT TERMS IN PBAR(I,J). SUM OF PRODUCT TERMS ARE FOUND BY RULE

(A B) = A' + AB. UNCOMPLEMENTED VARIABLE IS DENOTED BY 1 AND COMPLEMENTED VARIABLE BY -1 AND ABSENCE BY 0

INTEGER P, PX, TX, PORG, BCOMB, PBAR
COMMON /A/ NP, NB
COMMON /D/ P, PBAR, IPB
DIMENSION P(10), PBAR(10,10)

COUNT NUMBER OF BRANCHES IN P(J)

COUNT OF ROWS IN PBAR(I,J)

DO 20 I=1,NB
  IF(P(I), EQ.1) IPB = IPB+1
  CONTINUE
  DO 02 J=1,IPB
    PBAR(I,J) = 0
    CONTINUE
    RETURN
  CONTINUE

SUBROUTINE SIMPLE

THIS SUBROUTINE TAKES TERMS FROM PX(I,J) ANDS THEM WITH TERMS IN PBAR(I,J) SIMPLIFIES RESULTING TERMS AND RESTORES THESE BACK

IN PX(I,J)

INTEGER P, PX, TX, PORG, BCOMB, PBAR
COMMON /A/ NP, NR
COMMON /B/ P, PBAR, IPB
COMMON /D/ PX, TX, IPX
DIMENSION PBAR(10,10), PX(10,10), TX(10,10), P(10)

ITPX = 0
DO 10 TA = 1, IPB
  PX(TA) = -1
  CONTINUE
  ITTX = 0
  DO 30 J=1,NR
    IF(PBAR(TA, J)) = 11, 22, 33
      IF(PX(TA, J), EQ.1) GO TO 44
      TX(TA) = 1
      GO TO 30
    22 TX(TA, J) = PX(TA, J)
      CONTINUE
    33 IF(PX(TA, J), EQ.1) GO TO 44
      TX(TA) = 1
      CONTINUE
      ITTX = ITTX+1
      GO TO 30

RETURN

Data used for this listing is of Example 4.2.
This section provides the flow chart of the computer program presented in Appendix A-II.
Add the next branch to already chosen set of branches

Are additional paths generated?

Yes

Determine Sys. Rel and Sys. Cost

Calculate $\Delta R/\Delta C$

Are all branches considered?

No

Add the least cost branch

Yes

Is $\Delta R/\Delta C = 0$ for all missing branches?

No

Select the branch, which gives max. $\Delta R/\Delta C$, add it already selected branches

Yes
Calculate the Sys Cost, Sys Ref, for the augmented network

Are all branches considered?

Yes

Print branches considered, Sys Cost consumed, Sys Ref.

Stop

No Go To A