HERMES PROJECT

HERMES I USER'S MANUAL

PREPARED FOR THE

NATIONAL TELECOMMUNICATIONS BRANCH

DEPARTMENT OF COMMUNICATIONS

BY

LABORATOIRE D'ECONOMETRIE

UNIVERSITE LAVAL

AND

SORÈS INC., MONTREAL

P
91
C6541
H47
1972

sorēs
Queen
P
91
C6541
H47
1972

HERMES PROJECT

(1)

HERMES I

USER'S MANUAL

Industry Canada
Library Queen
JUL 16 1998
Industrie Canada
Bibliothèque Queen

prepared for the

NATIONAL TELECOMMUNICATIONS BRANCH

DEPARTMENT OF COMMUNICATIONS

by

SORES INC.
Montreal

COMMUNICATIONS CANADA
NOV 5 1984
LIBRARY — BIBLIOTHÈQUE

March 1972

P
91
C654
H47
1972

DD4926475
DL 4926481

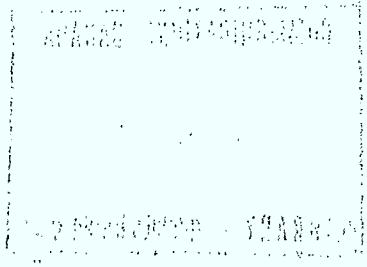


TABLE OF CONTENTS

		Page
1.	GENERAL DESCRIPTION	1
2.	INPUT DATA FORMAT	3
2.1	First Card	3
2.2	Second Group of Cards	4
2.3	Third Group of Cards	4
2.4	Fourth Group of Cards	5
2.5	Last data Card	6
APPENDIX A: SAMPLE PROBLEM		
1.	THE PROBLEM	A-1
2.	INPUT SPECIFICATION	A-1
3.	OPTIMAL INTEGER SOLUTION	A-4
4.	HERMES I OUTPUT	A-5
4.1	General description	A-5
4.2	Tables of output	A-8
APPENDIX B: HERMES I INTERMEDIATE INDEPTH FLOWCHART & LISTING		

LIST OF FIGURES

		Page
1.	HERMES I model - General Flow Chart	2
2.	Order of Input Data Cards	7
3.	Order of Cards for HERMES I	8

APPENDIX A

4.	Sample Problem	A-1
5.	Optimal Solution	A-4

APPENDIX B

6.	HERMES I - Intermediate in-depth flow chart	B-1

1. GENERAL DESCRIPTION

Given an interregional telecommunications network, integer values are assigned to capacity expansion activities, constituting the minimum cost capacity expansion program.

Although HERMES I consists of two steps (see Figure 1), no manual intervention is required.

Step 1. CADUCEE I

A bounded mixed integer programming problem has a finite number of feasible solutions. The problem one encounters is that this finite number can be too large despite today's computer capabilities. The purpose of CADUCEE I is to reduce the number of admissible chains which need to be examined by the mixed integer linear programming package (step 2 - TRANCHE).

Step 2. TRANCHE

TRANCHE is a mixed integer linear programming package utilizing the branch and bound algorithm.

Storage Requirements

CORE (272 K), Disc (temporary).

Equipment Specifications

Card reader
Disc drive.

Language

Fortran IV, Level G.

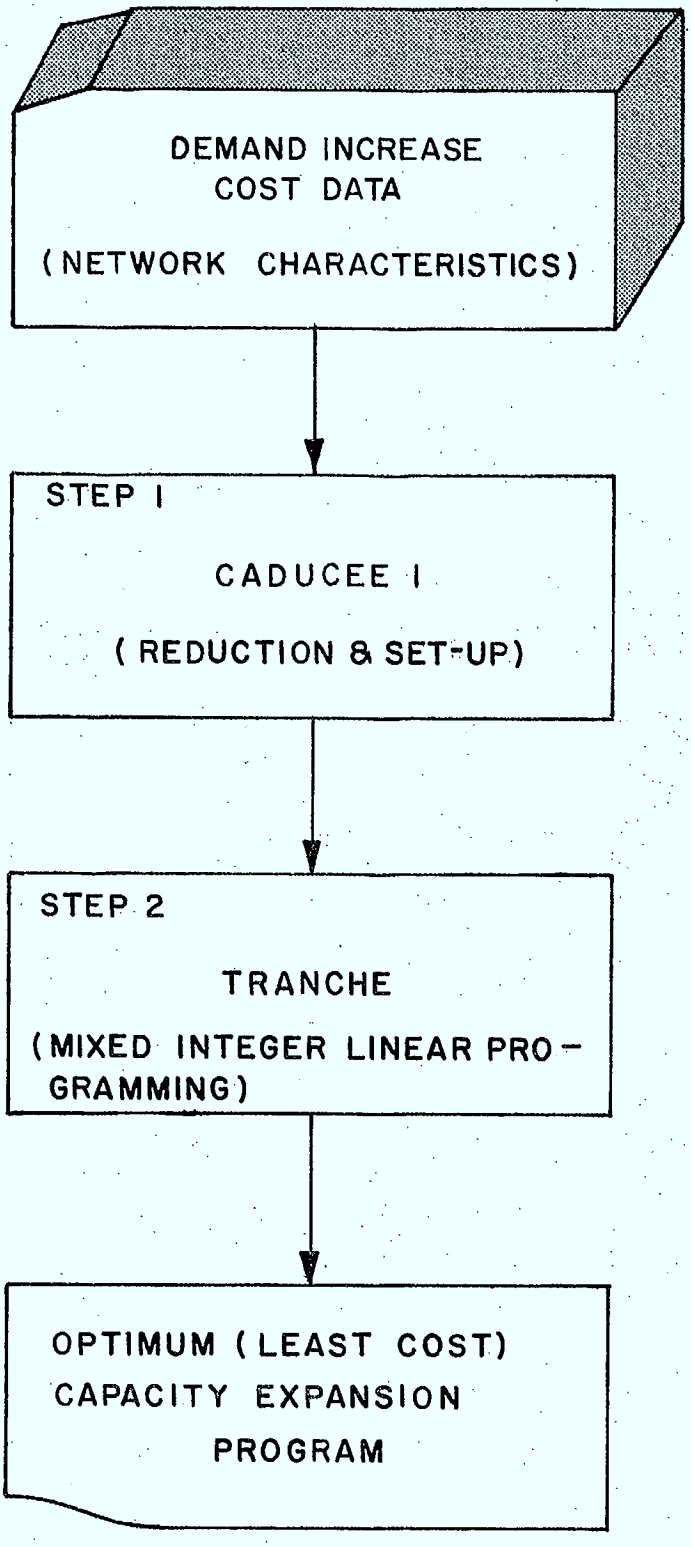
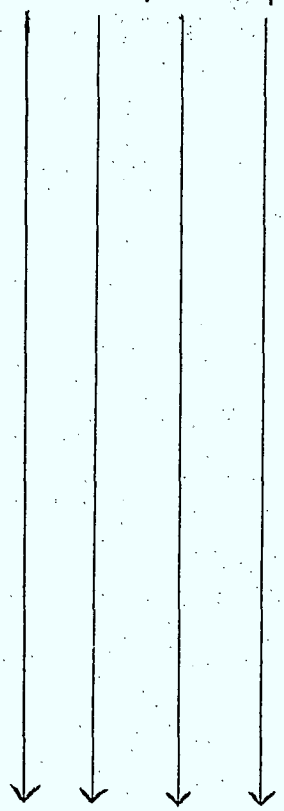


Figure 1 - HERMES1 - General Flow Chart

2. INPUT DATA FORMAT:

2.1 First card

Field	Columns	Contents
1	1-5	NCØUP = Number of pairs of points of demand
2	6-10	IPA = 0 default, > 0 implement
3	11-15	IBORNE
4	16-20	IBA
5	21-25	IMPR
6	26-30	INDOM
7	31-35	IADCH
8	36-40	ISTAT
9	41-45	IAMCL
10	46-50	ISTET
11	51-55	ICTGT
12	56-60	ILKDTA
13	61-65	IBCD
14	66-70	IPICT = 0 default, > 0 implement
15	71-80	Not used



Fields 2 to 14 inclusive are print-out options. See Appendix A for full descriptions.

2.2 Second Group of Cards: There are as many cards in this group as there are pairs of points of demand.

Field	Columns	Contents
1	1-5	NØRG = Node number of origin
2	6-10	NDEST = Node number of destination
3	11-15	NDEM = Service channel demand for this pair of demand points
4	16-80	Not used

2.3 Third Group of Cards: Incremental costs describing the cost function of each of the four cost categories. There are four cards in this group.

Field	Columns	Contents
1	1-5	X_1 = initial investment cost for a new route
2	6-10	ΔX = cost of installing one additional channel on existing route Cost/mile/channel.
3	11-15	Y = initial investment cost for a new system on existing route
4	16-20	ΔY = cost of installing one additional channel on new system Cost/mile/channel.
5	21-80	Not used.

2.4 Fourth Group of Cards : This group of cards is composed of as many cards as there are links.

Field	Columns	Contents
1	1-3	A = number identifying link
2	4-6	Ø = number identifying first node of link
3	7-9	D = number identifying 2nd node of link
4	10-12	MI = length of link in miles
5	13-14	LD(1) = how many X_1's exist at present
6	15-16	LD(2) = how many ΔX_1's are presently installed
7	17-18	LD(3) = total number of channels that can be installed on this link before encountering the next step.
8	19-20	LD(4) = how many X_2's exist
9	21-22	LD(5) = how many ΔX_2's are presently installed
10	23-24	LD(6) = total number of channels that can be installed on this link before encountering the next step

Field	Columns	Contents	
11	25-26	LD(7)	How many Y_1's exist at present
12	27-28	LD(8)	How many ΔY_1's are presently installed
13	29-30	LD(9)	Total number of channels that can be installed on this link before encountering the next step
14	31-32	LD(10)	How many Y_2's exist
15	33-34	LD(11)	How many ΔY_2's are presently installed
16	35-36	LD(12)	Total number of channels that can be installed before reaching full capacity
17	37	EX =	cost category of link
18	38-43	$X1$ =	costs which take priority over costs described in 4 cost categories
19	44-49	$\Delta X1$ =	similar to above
20	50-55	$Y1$ =	similar to above
21	56-61	$\Delta Y1$ =	similar to above
22	62-80	Not used	

2.5 Last Data Card

Field	Columns	Contents
1	1-6	999999 - marker signalling the end of link data
2	7-80	Not used

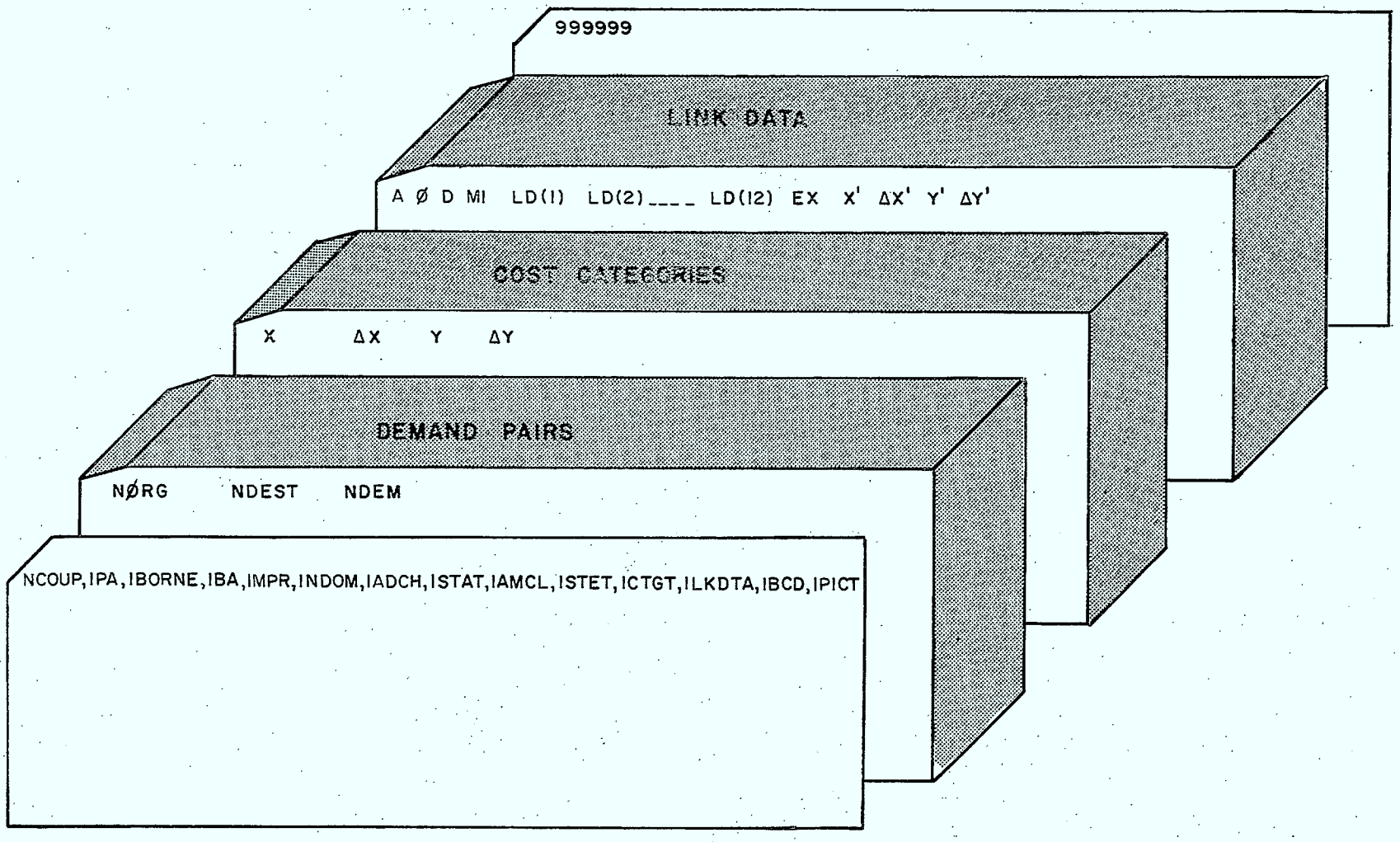


Figure 2 : Order of input data cards

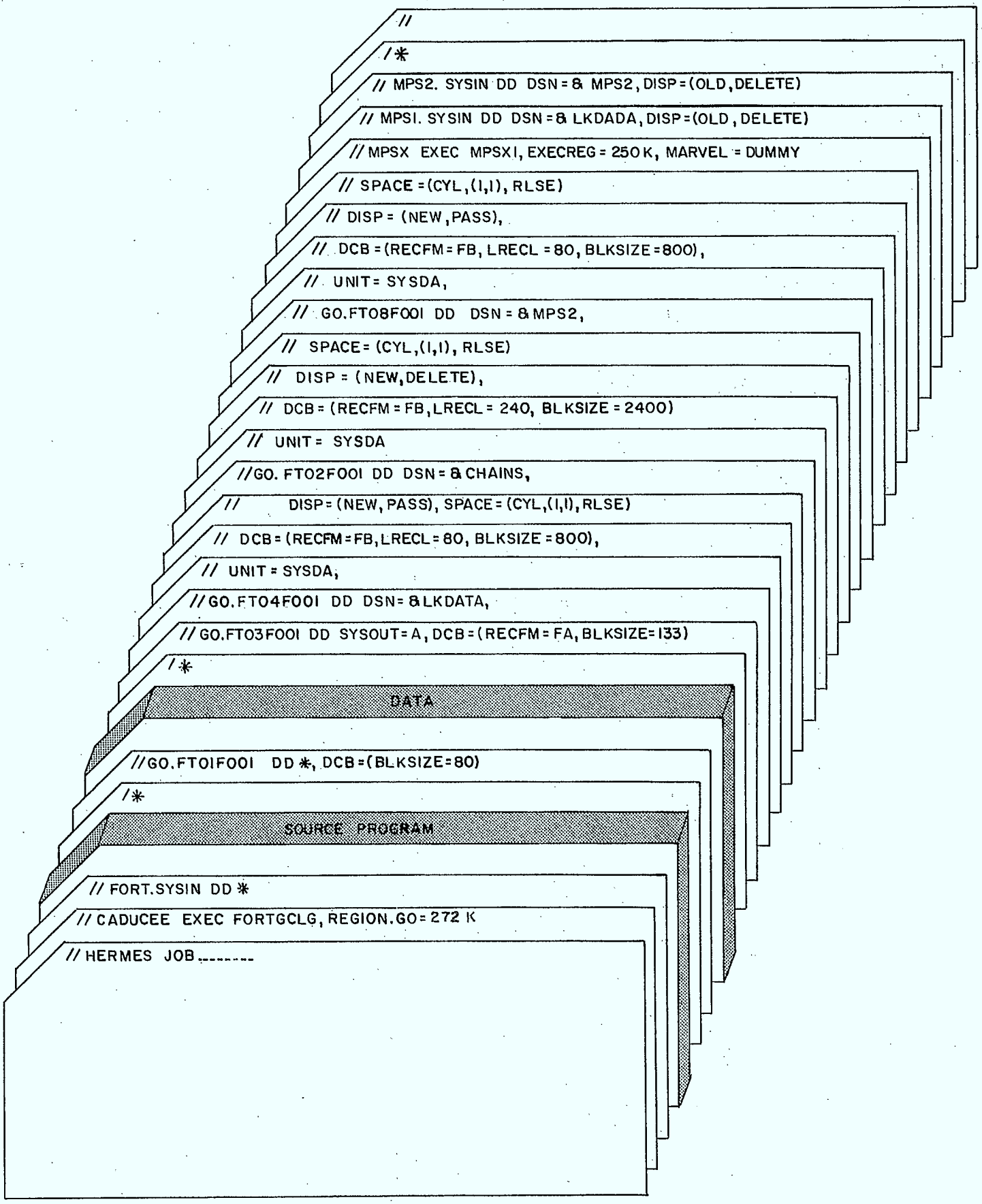


Figure 3: Order of HERMES I Cards

APPENDIX A : SAMPLE PROBLEM

1. THE PROBLEM

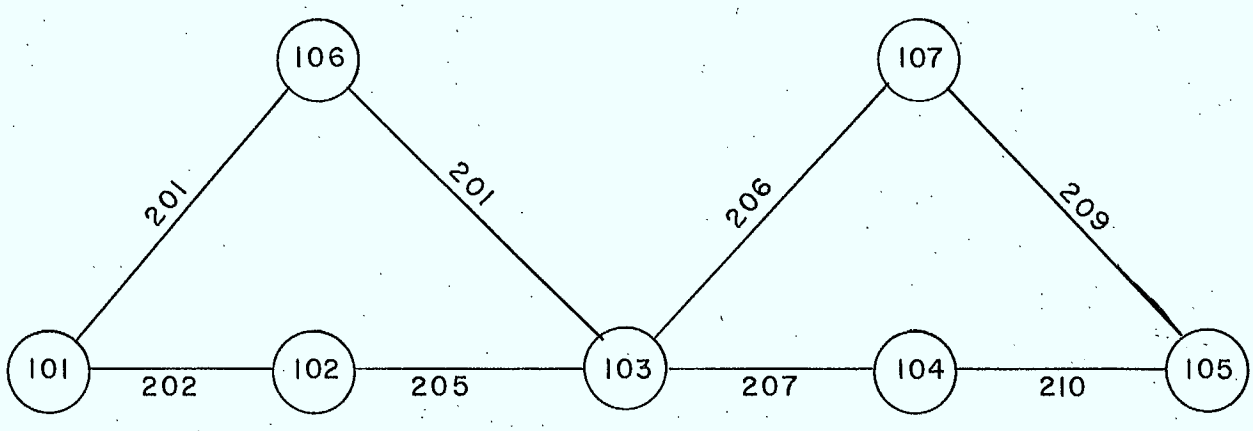


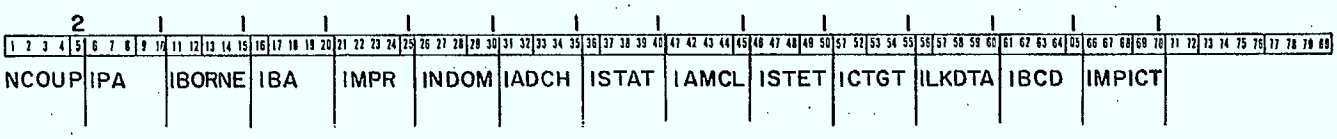
Figure 4 : Sample Problem

Referring to figure 4, the problem is that we wish to introduce an additional 2 service channels between stations 101 and 107 and an additional 3 service channels between stations 101 and 105. Although the network shown and the problem posed is trivial, its intent is to aid the user in setting up the data required and interpreting the output generated.

Note : links are numbered sequentially from 101 upward.

2. INPUT SPECIFICATION

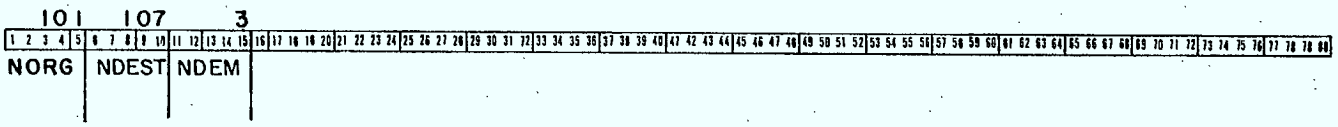
First data card: NCØUP = 2
All print options implemented.



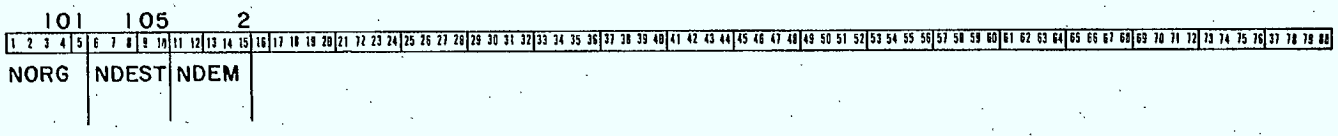
Second group of cards

Since the problem specifies two pairs of points of demand, there are therefore two data cards in this group.

1 st data card.



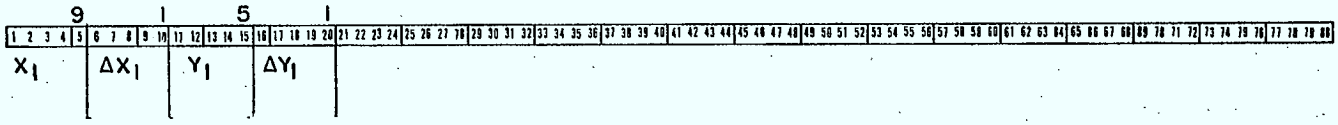
2 nd data card.



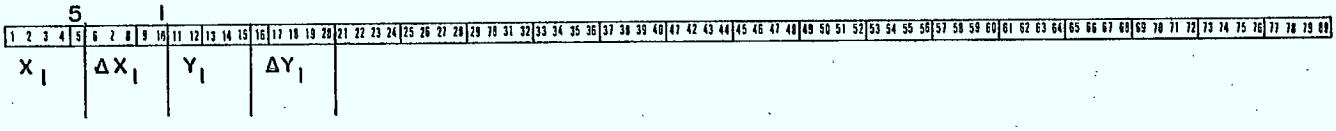
Third Group of Cards:

For this problem there are four cost categories. Costs are in $1,000.00 units.

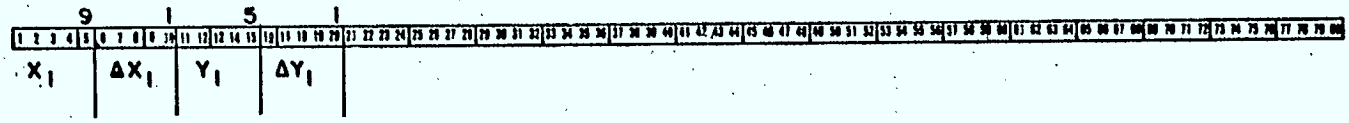
1 st cost categ.



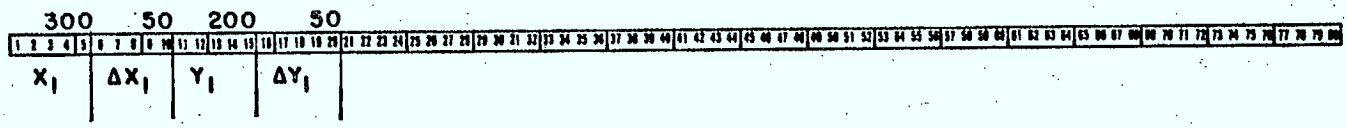
2 nd cost catg.



3 rd cost catg.



4 th cost catg.



Fourth group of data cards

LAST DATA CARD

999999

MARKER SIGNALLING THE END OF LINK DATA

LINK #	NODE #1	NODE #2	MILES	LD(1)	LD(2)	LD(3)	LD(4)	LD(5)	LD(6)	LD(7)	LD(8)	LD(9)	LD(10)	LD(11)	LD(12)	EX	XI	ΔXI	YI	ΔYI
210	104	105	385	1		5			3							1				
209	105	107	60			5			5							2				
208	104	107	295	1	3	5			5							1				
207	103	104	210	1	2	5			3							1				
206	103	107	275	1	2	5			5			3				3			960	
205	102	103	385	1	3	5			5							1				
204	103	106	120	1	1	5			5							1				
203	102	106	130	1	3	5			5							1				
202	101	102	65	2		10			5			1				3			195	
201	101	106	60			5			5							1				

3. OPTIMAL INTEGER SOLUTION

From Table 53. (Node 10).

Functional = $3,965,00 (in units of $1,000.00).

VARIABLE	INTEGER	ACTIVITY
X1201	1	1
DX1201	4	4
		$\frac{4}{5}$ add five channels on link 201
DX1204	3	3
X2204	1	1
DX2204	1	1
		$\frac{1}{5}$ add five channels on link 204
DX1206	2	2
X2206	1	1
DX2206	2	2
		$\frac{2}{5}$ add five channels on link 206
X1209	1	1
DX1209	2	2
		$\frac{2}{3}$ add 3 channels on link 209

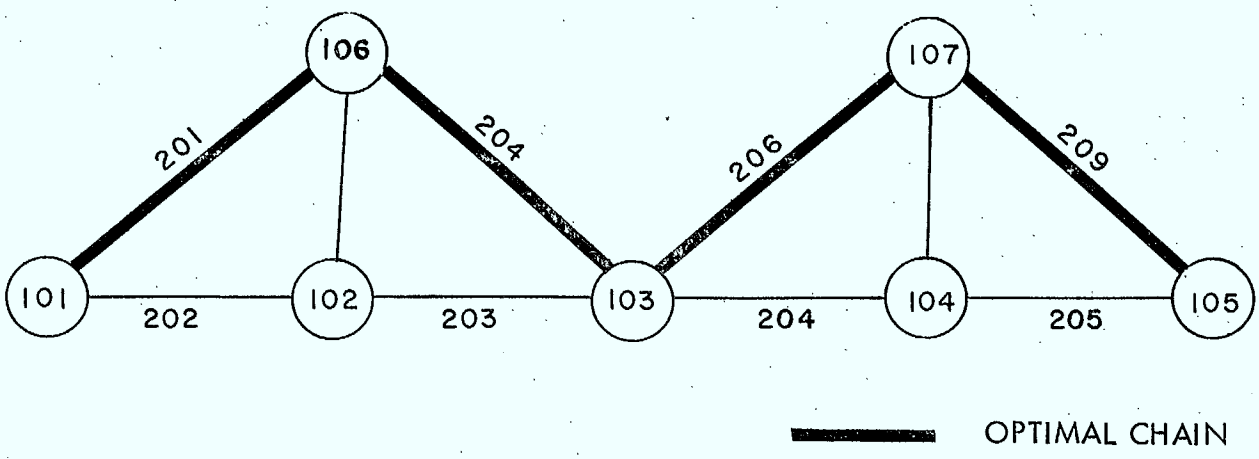


Figure 5 : Optimal Solution

4. HERMES I OUTPUT

4.1 General Description

CADUCEE I

OPTION	TABLE NUMBER	CONTENTS
IPA	1	Problem definition
IBORNE	2	Conversion of Node numbers to new identifiers
IBORNE	3	BORNE Table : Incremental upper and lower bound costs
IBA	4 & 16	Upper bound costs from NØRG to every other node in the network
IBA	5 & 17	Minimum upper bound cost chain from NØRG to NDEST
IBA	6 & 18	Upper bound cost from NDEST to every other node in the network
IBA	7, 19	Minimum upper bound cost chain from NDEST to NØRG
IBA	8, 20	Lower bound cost from NØRG to every other node in the network
IBA	9, 21	Minimum lower bound cost chain from NØRG to NDEST
IBA	10, 22	Lower bound cost from NDEST to every other node in the network
IBA	11, 23	Minimum lower bound cost chain from NDEST to NØRG
IMPR	12, 24	DOMINO Tables
INDOM	13, 25	List of non-dominated nodes
IADCH	14 & 26	List of admissible chains (composed of node numbers)

CADUCEE I (cont'd)

OPTION	TABLE NUMBER	CONTENTS
ISTAT	15 & 27	Admissible chain statistics
IAMCL	28	Admissible chains (composed of links)
ISTET	29	TRANCHE input statistics
ICTGT	30	Listing of cost categories
ILKDTA	31	Admissible links and link data

TRANCHE

OPTION	TABLE NUMBER	CONTENTS
	32	Control program listing for TRANCHE
	33	TRANCHE set-up statistics
IBCD	34	TRANCHE input data
	35	Internal files and storage allocations for TRANCHE
IPICT	36	Pictorial representation of TRANCHE matrix
IPICT	37	Summary of matrix statistics
	38	Crash-searches for a starting basis
	39	Primal-searches for optimal continuous solution
	40	Optimum continuous solution
	41	Continuous solution (rows section)
	42	Continuous solution (columns section)

TRANCHE (cont'd)

OPTION	TABLE NUMBER	CONTENTS
	43	Search for first integer solution
	44	First integer solution
	45	First integer solution (rows section)
	46	First integer solution (columns section)
	47	Search for second integer solution
	48	Second integer solution
	49	Second integer solution (rows section)
	50	Second integer solution (columns section)
	51	Search for third integer solution
	52	Optimal integer solution found
	53	Summary of two integer solutions obtained
	54	Statistical survey of continuous optimum, integer and optimal integer solutions
	55	Assignment of integer values to capacity expansion activities

All the above tables are reproduced in the following section, with reference to the sample problem.

4.2 Tables of Output

NOMBRE DE PAIRES DE SOMMETS = 2

PAIRES DE SOMMETS :

101	105	3
101	107	2

Table 1 : Listing of the problem with regard to the number of pairs of demand points, identification of the demand points and their respective demand increases.

NODE NUMBER	NEW NODE NUMBER
101	1
106	2
102	3
103	4
107	5
104	6
105	7

Table 2 : Listing of the node numbers with their associated new identifiers.

ITERATION NO = 1

NO D ARETE	SOMMETS CONCERNES		COUT B.S.	COUT B.I.	M.R.D.
201	1	2	540	60	5
202	1	3	65	65	5
203	3	2	390	130	5
204	4	2	360	120	5
205	3	4	1155	385	5
206	4	5	825	275	5
207	4	6	630	210	5
208	6	5	885	295	5
209	7	5	300	60	5
210	6	7	1155	385	5

Table 3 : BORNE TABLE : The upper and lower bound incremental costs associated with the links of the network

SORĒS inc.

VALEUR MINIMUM A PARTIR DU SOMMET 1

DEST.	VALEUR BS
1	0
2	455
3	65
4	815
5	1640
6	1445
7	1940

Table 4 : The upper bound costs from NØRG to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BS
1	3	65
3	2	455
2	4	815
4	5	1640
5	7	1940

VALEUR TOTALE 1940

Table 5 : The minimum upper bound cost chain from NØRG to NDEST

VALEUR MINIMUM A PARTIR DU SOMMET 7

DEST.	VALEUR BS
1	1940
2	1485
3	1875
4	1125
5	300
6	1155
7	0

Table 6 : Upper bound cost from NDEST to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BS
7	5	300
5	4	1125
4	2	1485
2	3	1875
3	1	1940

VALEUR TOTALE 1940

Table 7 : Minimum upper bound cost chain from NDEST to NØRG

VALEUR MINIMUM A PARTIR DU SOMMET 1

DEST.	VALEUR BI
1	0
2	60
3	65
4	180
5	455
6	390
7	515

Table 8 : Lower bound costs from NØRG to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BI
1	2	60
2	4	180
4	5	455
5	7	515

VALEUR TOTALE 515

Table 9 : Minimum lower bound cost chain from NØRG to NDEST

VALEUR MINIMUM A PARTIR DU SOMMET 7

DEST.	VALEUR BI
1	515
2	455
3	580
4	335
5	60
6	355
7	0

Table 10 : Lower bound cost from NDEST to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BI
7	5	60
5	4	335
4	2	455
2	1	515

VALEUR TOTALE 515

Table 11 : Minimum lower bound cost chain from NDEST to NØRG

COUTS MINIMUM

SOMMET NUMERO	BORNE SUPERIEURE DE NORG	BORNE SUPERIEURE DE NDEST	BORNE INFERIEURE DE NORG	BORNE INFERIEURE DE NDEST
1	0	1940	0	515
2	455	1485	60	455
3	65	1875	65	580
4	815	1125	180	335
5	1640	300	455	60
6	1445	1155	390	355
7	1940	0	515	0

Table 12 : DOMINO TABLE : The lower bound and upper bound costs (from NØRG to every other node in the network and from NDEST to every other node in the network).

sorēs inc.

SOMMETS NON DOMINES

1 2 3 4 5 6 7

Table 13 : Non-dominated nodes

CHAINES ADMISSIBLES

3 101 107	1 101 106 103 107
3 101 107	2 101 102 103 107
4 101 107	3 101 106 103 104 107
4 101 107	4 101 102 106 103 107
4 101 107	5 101 102 103 104 107
5 101 107	6 101 106 103 104 105 107
5 101 107	7 101 102 106 103 104 107
5 101 107	8 101 102 103 104 105 107
6 101 107	9 101 102 106 103 104 105 107

Table 14 : List of admissible chains (chains composed of nodes)

STATISTIQUES SUR LES CHAINES

****

NOMBRE TOTAL DE CHAINES = 9

NST(1) = 0

NST(2) = 0

NST(3) = 2

NST(4) = 3

NST(5) = 3

NST(6) = 1

Table 15 : Chain Statistics

VALEUR MINIMUM A PARTIR DU SOMMET 1

DEST.	VALEUR BS
1	0
2	455
3	65
4	815
5	1640
6	1445
7	1940

Table 16 : The upper bound costs from NØRG to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BS
1	3	65
3	2	455
2	4	815
4	5	1640

VALEUR TOTALE 1640

Table 17 : The minimum upper bound cost chain from NØRG to NDEST

VALEUR MINIMUM A PARTIR DU SOMMET 5

DEST.	VALEUR BS
1	1640
2	1185
3	1575
4	825
5	0
6	885
7	300

Table 18 : Upper bound cost from NDEST to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BS
5	4	825
4	2	1185
2	3	1575
3	1	1640

VALEUR TOTALE 1640

Table 19 : Minimum upper bound cost chain from NDEST to NØRG

VALEUR MINIMUM A PARTIR DU SOMMET 1

DEST.	VALEUR BI
1	0
2	60
3	65
4	180
5	455
6	390
7	515

Table 20 : Lower bound costs from NØRG to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BI
1	2	60
2	4	180
4	5	455

VALEUR TOTALE 455

Table 21 : Minimum lower bound cost chain from NØRG to NDEST

VALEUR MINIMUM A PARTIR DU SOMMET 5

DEST.	VALEUR BI
1	455
2	395
3	520
4	275
5	0
6	295
7	60

Table 22 : Lower bound cost from NDEST to every other node in the network

CHAINE OPTIMALE (PAS NECESSAIREMENT UNIQUE)

ORIG.	DEST.	VALEUR BI
5	4	275
4	2	395
2	1	455

VALEUR TOTALE 455

Table 23 : Minimum lower bound cost chain from NDEST to NØRG

COUTS MINIMUM

SOMMET NUMERO	BORNE SUPERIEURE DE NORG	BORNE SUPERIEURE DE NDEST	BORNE INFERIEURE DE NORG	BORNE INFERIEURE DE NDEST
1	0	1640	0	455
2	455	1185	60	395
3	65	1575	65	520
4	815	825	180	275
5	1640	0	455	0
6	1445	885	390	295
7	1940	300	515	60

Table 24 : DOMINO TABLE : The lower bound and upper bound cost (from NØRG to every other node in the network and from NDEST to every other node in the network)

sorès inc.

SOMMETS NON DOMINES

1 2 3 4 5 6 7

Table 25 : Non-dominated nodes

CHAINES ADMISSIBLES

4	101	105	1	101	106	103	107	105	
4	101	105	2	101	106	103	104	105	
4	101	105	3	101	102	103	107	105	
4	101	105	4	101	102	103	104	105	
5	101	105	5	101	106	103	107	104	105
5	101	105	6	101	106	103	104	107	105
5	101	105	7	101	102	106	103	107	105
5	101	105	8	101	102	106	103	104	105
5	101	105	9	101	102	103	107	104	105
5	101	105	10	101	102	103	104	107	105
6	101	105	11	101	102	106	103	107	104	105
6	101	105	12	101	102	106	103	104	107	105

Table 26 : List of admissible chains (chains composed of nodes)

STATISTIQUES SUR LES CHAINES

****

NOMBRE TOTAL DE CHAINES = 12

NST(1) = 0

NST(2) = 0

NST(3) = 0

NST(4) = 4

NST(5) = 6

NST(6) = 2

Table 27 : Chain statistics

ADMISSABLE CHAINS

NUMBER OF LINKS	NORG	NDEST	CHAIN NUMBER	LINKS IN CHAIN
4	101	105	1	201 204 206 209
4	101	105	2	201 204 207 210
4	101	105	3	202 205 206 209
4	101	105	4	202 205 207 210
5	101	105	5	201 204 206 208 210
5	101	105	6	201 204 207 208 209
5	101	105	7	202 203 204 206 209
5	101	105	8	202 203 204 207 210
5	101	105	9	202 205 206 208 210
5	101	105	10	202 205 207 208 209
6	101	105	11	202 203 204 206 208 210
6	101	105	12	202 203 204 207 208 209
3	101	107	1	201 204 206
3	101	107	2	202 205 206
4	101	107	3	201 204 207 208
4	101	107	4	202 203 204 206
4	101	107	5	202 205 207 208
5	101	107	6	201 204 207 210 209
5	101	107	7	202 203 204 207 208
5	101	107	8	202 205 207 210 209
6	101	107	9	202 203 204 207 210 209

Table 28 : Admissible chains (chains composed of links).

THERE ARE 21 ADMISSABLE CHAINS

THERE ARE 10 ADMISSABLE LINKS

Table 29 : Statistics for TRANCHE input.

13. ICTGT 0

COST CATEGORIES

CATG.	X1	DX1	Y1	DY1
1	9	1	5	1
2	5	1	0	0
3	9	1	5	1
4	300	50	200	50

Table 30 : Cost categories

ADMISSABLE LINK DATA

201	101	106	60	0	0	5	0	0	5	0	0	0	0	0	0	1	0	0	0	0
204	103	106	120	1	1	5	0	0	5	0	0	0	0	0	0	1	0	0	0	0
206	103	107	275	1	2	5	0	0	5	0	0	3	0	0	0	3	0	0	960	0
209	105	107	60	0	0	5	0	0	5	0	0	0	0	0	0	2	0	0	0	0
207	103	104	210	1	2	5	0	0	3	0	0	0	0	0	0	1	0	0	0	0
210	104	105	385	1	0	5	0	0	3	0	0	0	0	0	0	1	0	0	0	0
202	101	102	65	2	0	10	0	0	5	0	0	1	0	0	0	3	0	0	195	0
205	102	103	385	1	3	5	0	0	5	0	0	0	0	0	0	1	0	0	0	0
208	104	107	295	1	3	5	0	0	5	0	0	0	0	0	0	1	0	0	0	0
203	102	106	130	1	3	5	0	0	5	0	0	0	0	0	0	1	0	0	0	0

Table 31 : Admissible link data

sorès inc.

```
CONTROL PROGRAM COMPILER.  MPSX  RELEASE 1

0001                    PROGRAM
0002                    INITIALZ
0090                    XPROC=9000
0091                    MOVE(XDATA, 'TRANCHE')
0092                    MOVE(XPBNAME,'PROB')
0093                    CONVERT
0094                    BCDOUT
0095                    SETUP('BOUND','BND1')
0096                    PICTURE
0097                    MOVE(XOBJ,'COST1')
0098                    MOVE(XRHS,'RHS1')
0099                    OPTIMIZE
0397                    SOLUTION
0398                    INIMIX
0423                    OPTIMIX('COST',9500.,10,10,1)
0502                    EXIT
0503                    PEND
```

Table 32 : Control program listing for mixed integer linear programming package (TRANCHE)

```
                    EXECUTOR.  MPSX  RELEASE 1

CONVERT TRANCHE  TO PROB

TIME =    0.00

1- ROWS SECTION.

   0 MINOR ERROR(S) -    0 MAJOR ERROR(S).

2- COLUMNS SECTION.

   0 MINOR ERROR(S) -    0 MAJOR ERROR(S).

3- RHS'S SECTION.

  RHS1

   0 MINOR ERROR(S) -    0 MAJOR ERROR(S).

5- BOUNDS SECTION.

 THE COLUMN Y2206     IS NOT USED.
 THE COLUMN Y2206     IS NOT USED.
 THE COLUMN DY2206    IS NOT USED.
 THE COLUMN Y2202     IS NOT USED.
 THE COLUMN Y2202     IS NOT USED.
 THE COLUMN DY2202    IS NOT USED.
  BND1

   0 MINOR ERROR(S) -    0 MAJOR ERROR(S).

 THE FOLLOWING ROWS HAVE NO STRUCTURAL ELEMENTS

 Y2DY2206  Y2DY2202

 PROBLEM STATISTICS -     61 ROWS,      160 VARIABLES,      401 ELEMENTS, DENSITY =    4.10
                          44 INTEGER VARIABLES

 THESE STATISTICS INCLUDE ONE SLACK VARIABLE FOR EACH ROW.

     0 MINOR ERRORS,      0 MAJOR ERRORS.
```

Table 33 : TRANCHE Statistics

SORĒS Inc.

EXECUTOR. MPSX RELEASE 1

```
NAME          TRANCHE
ROWS
 G   D101105
 G   D101107
 G   D101106
 G   D103106
 G   D103107
 G   D105107
 G   D103104
 G   D104105
 G   D101102
 G   D102103
 G   D104107
 G   D102106
 L   C201
 G   X1DX1201
 G   X2DX2201
 G   X1X2201
 L   C204
 G   X1DX1204
 G   X2DX2204
 G   X1X2204
 L   C206
 G   X1DX1206
 G   X2DX2206
 G   X1X2206
 G   X2Y1206
 G   Y1DY1206
 G   Y2DY2206
 G   Y1Y2206
 L   C209
 G   X1DX1209
 G   X2DX2209
 G   X1X2209
 L   C207
 G   X1DX1207
 G   X2DX2207
 G   X1X2207
 L   C210
 G   X1DX1210
 G   X2DX2210
 G   X1X2210
 L   C202
 G   X1DX1202
 G   X2DX2202
 G   X1X2202
 G   X2Y1202
 G   Y1DY1202
 G   Y2DY2202
 G   Y1Y2202
 L   C205
 G   X1DX1205
 G   X2DX2205
 G   X1X2205
```

Table 34 : TRANCHE input data

EXECUTOR. MPSX RELEASE 1

```
 L   C208
 G   X1DX1208
 G   X2DX2208
 G   X1X2208
 L   C203
 G   X1DX1203
 G   X2DX2203
 G   X1X2203
 N   COST1
COLUMNS
    R1011001  D101105        1.00000   C201          1.00000
    R1011001  C204           1.00000   C206          1.00000
    R1011001  C209           1.00000
    R1021002  D101105        1.00000   C201          1.00000
    R1021002  C204           1.00000   C207          1.00000
    R1021002  C210           1.00000
    R1031003  D101105        1.00000   C202          1.00000
    R1031003  C205           1.00000   C206          1.00000
    R1031003  C209           1.00000
    R1041004  D101105        1.00000   C202          1.00000
    R1041004  C205           1.00000   C207          1.00000
    R1041004  C210           1.00000
    R1051005  D101105        1.00000   C201          1.00000
    R1051005  C204           1.00000   C206          1.00000
    R1051005  C208           1.00000   C210          1.00000
    R1061006  D101105        1.00000   C201          1.00000
    R1061006  C204           1.00000   C207          1.00000
    R1061006  C208           1.00000   C209          1.00000
    R1071007  D101105        1.00000   C202          1.00000
    R1071007  C203           1.00000   C204          1.00000
    R1071007  C206           1.00000   C209          1.00000
    R1081008  D101105        1.00000   C202          1.00000
    R1081008  C203           1.00000   C204          1.00000
    R1081008  C207           1.00000   C210          1.00000
    R1091009  D101105        1.00000   C202          1.00000
    R1091009  C205           1.00000   C206          1.00000
    R1091009  C208           1.00000   C210          1.00000
    R1101010  D101105        1.00000   C202          1.00000
    R1101010  C205           1.00000   C207          1.00000
    R1101010  C208           1.00000   C209          1.00000
    R1111011  D101105        1.00000   C202          1.00000
    R1111011  C203           1.00000   C204          1.00000
    R1111011  C206           1.00000   C208          1.00000
    R1111011  C210           1.00000
    R1121012  D101105        1.00000   C202          1.00000
    R1121012  C203           1.00000   C204          1.00000
    R1121012  C207           1.00000   C208          1.00000
    R1121012  C209           1.00000
    R1131001  D101107        1.00000   C201          1.00000
    R1131001  C204           1.00000   C206          1.00000
    R1141002  D101107        1.00000   C202          1.00000
    R1141002  C205           1.00000   C206          1.00000
    R1151003  D101107        1.00000   C201          1.00000
    R1151003  C204           1.00000   C207          1.00000
```

Table 34 : TRANCHE input data (cont'd)

EXECUTOR. MPSX RELEASE 1

R1151003	C208		1.00000		
R1161004	D101107		1.00000	C202	1.00000
R1161004	C203		1.00000	C204	1.00000
R1161004	C206		1.00000		
R1171005	D101107		1.00000	C202	1.00000
R1171005	C205		1.00000	C207	1.00000
R1171005	C208		1.00000		
R1181006	D101107		1.00000	C201	1.00000
R1181006	C204		1.00000	C207	1.00000
R1181006	C210		1.00000	C209	1.00000
R1191007	D101107		1.00000	C202	1.00000
R1191007	C203		1.00000	C204	1.00000
R1191007	C207		1.00000	C208	1.00000
R1201008	D101107		1.00000	C202	1.00000
R1201008	C205		1.00000	C207	1.00000
R1201008	C210		1.00000	C209	1.00000
R1211009	D101107		1.00000	C202	1.00000
R1211009	C203		1.00000	C204	1.00000
R1211009	C207		1.00000	C210	1.00000
R1211009	C209		1.00000		
R1221000	D101106		1.00000	C201	1.00000
R1231000	D103106		1.00000	C204	1.00000
R1241000	D103107		1.00000	C206	1.00000
R1251000	D105107		1.00000	C209	1.00000
R1261000	D103104		1.00000	C207	1.00000
R1271000	D104105		1.00000	C210	1.00000
R1281000	D101102		1.00000	C202	1.00000
R1291000	D102103		1.00000	C205	1.00000
R1301000	D104107		1.00000	C208	1.00000
R1311000	D102106		1.00000	C203	1.00000
IDX1201	C201	-	1.00000	X1X2201	1.00000
IDX2201	C201	-	1.00000		
IDX1204	C204	-	1.00000	X1X2204	1.00000
IDX2204	C204	-	1.00000		
IDX1206	C206	-	1.00000	X1X2206	1.00000
IDX2206	C206	-	1.00000	X2Y1206	1.00000
IDY1206	C206	-	1.00000	Y1Y2206	1.00000
IDY2206	C206	-	1.00000		
IDX1209	C209	-	1.00000	X1X2209	1.00000
IDX2209	C209	-	1.00000		
IDX1207	C207	-	1.00000	X1X2207	1.00000
IDX2207	C207	-	1.00000		
IDX1210	C210	-	1.00000	X1X2210	1.00000
IDX2210	C210	-	1.00000		
IDX1202	C202	-	1.00000	X1X2202	1.00000
IDX2202	C202	-	1.00000	X2Y1202	1.00000
IDY1202	C202	-	1.00000	Y1Y2202	1.00000
IDY2202	C202	-	1.00000		
IDX1205	C205	-	1.00000	X1X2205	1.00000
IDX2205	C205	-	1.00000		
IDX1208	C208	-	1.00000	X1X2208	1.00000
IDX2208	C208	-	1.00000		
IDX1203	C203	-	1.00000	X1X2203	1.00000
IDX2203	C203	-	1.00000		

Table 34 ; TRANCHE input data (cont'd)

EXECUTOR. MPSX RELEASE 1

MARKER1	'MARKER'			'INTORG'		
X1201	C201	-	1.00000	X1DX1201		9.00000
X1201	X1X2201		1.00000	COST1		540.00000
DX1201	C201	-	1.00000	X1DX1201	-	1.00000
DX1201	X1X2201		1.00000	COST1		60.00000
X2201	C201	-	1.00000	X2DX2201		9.00000
X2201	X1X2201	-	5.00000	COST1		180.00000
DX2201	C201	-	1.00000	X2DX2201	-	1.00000
DX2201	COST1		60.00000			
X1204	C204	-	1.00000	X1DX1204		9.00000
X1204	X1X2204		1.00000			
DX1204	C204	-	1.00000	X1DX1204	-	1.00000
DX1204	X1X2204		1.00000	COST1		120.00000
X2204	C204	-	1.00000	X2DX2204		9.00000
X2204	X1X2204	-	5.00000	COST1		360.00000
DX2204	C204	-	1.00000	X2DX2204	-	1.00000
DX2204	COST1		120.00000			
X1206	C206	-	1.00000	X1DX1206		9.00000
X1206	X1X2206		1.00000			
DX1206	C206	-	1.00000	X1DX1206	-	1.00000
DX1206	X1X2206		1.00000	COST1		275.00000
X2206	C206	-	1.00000	X2DX2206		9.00000
X2206	X1X2206	-	5.00000	X2Y1206		1.00000
X2206	COST1		825.00000			
DX2206	C206	-	1.00000	X2DX2206	-	1.00000
DX2206	X2Y1206		1.00000	COST1		275.00000
Y1206	C206	-	1.00000	X2Y1206	-	5.00000
Y1206	Y1DY1206		9.00000	Y1Y2206		1.00000
Y1206	COST1		960.00000			
DY1206	C206	-	1.00000	Y1DY1206	-	1.00000
DY1206	Y1Y2206		1.00000	COST1		275.00000
X1209	C209	-	1.00000	X1DX1209		9.00000
X1209	X1X2209		1.00000	COST1		300.00000
DX1209	C209	-	1.00000	X1DX1209	-	1.00000
DX1209	X1X2209		1.00000	COST1		60.00000
X2209	C209	-	1.00000	X2DX2209		9.00000
X2209	X1X2209	-	5.00000	COST1		180.00000
DX2209	C209	-	1.00000	X2DX2209	-	1.00000
DX2209	COST1		60.00000			
X1207	C207	-	1.00000	X1DX1207		9.00000
X1207	X1X2207		1.00000			
DX1207	C207	-	1.00000	X1DX1207	-	1.00000
DX1207	X1X2207		1.00000	COST1		210.00000
X2207	C207	-	1.00000	X2DX2207		9.00000
X2207	X1X2207	-	5.00000	COST1		630.00000
DX2207	C207	-	1.00000	X2DX2207	-	1.00000
DX2207	COST1		210.00000			
X1210	C210	-	1.00000	X1DX1210		9.00000
X1210	X1X2210		1.00000			
DX1210	C210	-	1.00000	X1DX1210	-	1.00000
DX1210	X1X2210		1.00000	COST1		385.00000
X2210	C210	-	1.00000	X2DX2210		9.00000
X2210	X1X2210	-	5.00000	COST1		1155.00000
DX2210	C210	-	1.00000	X2DX2210	-	1.00000

Table 34 : TRANCHE input data (cont'd)

EXECUTOR. MPSX RELEASE 1

```
    DX2210    COST1          385.00000
    X1202     C202      -      1.00000   X1DX1202        9.00000
    X1202     X1X2202          1.00000
    DX1202    C202      -      1.00000   X1DX1202  -     1.00000
    DX1202    X1X2202          1.00000   COST1          65.00000
    X2202     C202      -      1.00000   X2DX2202        9.00000
    X2202     X1X2202   -     10.00000   X2Y1202         1.00000
    X2202     COST1          195.00000
    DX2202    C202      -      1.00000   X2DX2202  -     1.00000
    DX2202    X2Y1202          1.00000   COST1          65.00000
    Y1202     C202      -      1.00000   X2Y1202   -     5.00000
    Y1202     Y1DY1202         9.00000   Y1Y2202         1.00000
    Y1202     COST1          195.00000
    DY1202    C202      -      1.00000   Y1DY1202  -     1.00000
    DY1202    Y1Y2202          1.00000   COST1          65.00000
    X1205     C205      -      1.00000   X1DX1205        9.00000
    X1205     X1X2205          1.00000
    DX1205    C205      -      1.00000   X1DX1205  -     1.00000
    DX1205    X1X2205          1.00000   COST1         385.00000
    X2205     C205      -      1.00000   X2DX2205        9.00000
    X2205     X1X2205   -      5.00000   COST1        1155.00000
    DX2205    C205      -      1.00000   X2DX2205  -     1.00000
    DX2205    COST1          385.00000
    X1208     C208      -      1.00000   X1DX1208        9.00000
    X1208     X1X2208          1.00000
    DX1208    C208      -      1.00000   X1DX1208  -     1.00000
    DX1208    X1X2208          1.00000   COST1          295.00000
    X2208     C208      -      1.00000   X2DX2208        9.00000
    X2208     X1X2208   -      5.00000   COST1          885.00000
    DX2208    C208      -      1.00000   X2DX2208  -     1.00000
    DX2208    COST1          295.00000
    X1203     C203      -      1.00000   X1DX1203        9.00000
    X1203     X1X2203          1.00000
    DX1203    C203      -      1.00000   X1DX1203  -     1.00000
    DX1203    X1X2203          1.00000   COST1          130.00000
    X2203     C203      -      1.00000   X2DX2203        9.00000
    X2203     X1X2203   -      5.00000   COST1          390.00000
    DX2203    C203      -      1.00000   X2DX2203  -     1.00000
    DX2203    COST1          130.00000
    MARKER2   'MARKER'                   'INTEND'
RHS
    RHS1      D101105          3.00000   D101107         2.00000
    RHS1      D103106          2.00000   D103107         3.00000
    RHS1      D103104          3.00000   D104105         1.00000
    RHS1      D101102          2.00000   D102103         4.00000
    RHS1      D104107          4.00000   D102106         4.00000
BOUNDS
 FX BND1      IDX1201          .
 FX BND1      IDX2201          .
 FX BND1      IDX1204          1.00000
 FX BND1      IDX2204          .
 FX BND1      IDX1206          2.00000
 FX BND1      IDX2206          .
 FX BND1      IDY1206          .
```

Table 34 : TRANCHE input data (cont'd)

EXECUTOR. MPSX RELEASE 1

FX BND1	IDY2206	.
FX BND1	IDX1209	.
FX BND1	IDX2209	.
FX BND1	IDX1207	2.00000
FX BND1	IDX2207	.
FX BND1	IDX1210	.
FX BND1	IDX2210	.
FX BND1	IDX1202	.
FX BND1	IDX2202	.
FX BND1	IDY1202	.
FX BND1	IDY2202	.
FX BND1	IDX1205	3.00000
FX BND1	IDX2205	.
FX BND1	IDX1208	3.00000
FX BND1	IDX2208	.
FX BND1	IDX1203	3.00000
FX BND1	IDX2203	.
LO BND1	X1201	.
UP BND1	X1201	1.00000
LO BND1	DX1201	.
UP BND1	DX1201	4.00000
LO BND1	X2201	.
UP BND1	X2201	1.00000
LO BND1	DX2201	.
UP BND1	DX2201	4.00000
FX BND1	X1204	1.00000
LO BND1	DX1204	.
UP BND1	DX1204	3.00000
LO BND1	X2204	.
UP BND1	X2204	1.00000
LO BND1	DX2204	.
UP BND1	DX2204	4.00000
FX BND1	X1206	1.00000
LO BND1	DX1206	.
UP BND1	DX1206	2.00000
LO BND1	X2206	.
UP BND1	X2206	1.00000
LO BND1	DX2206	.
UP BND1	DX2206	4.00000
LO BND1	Y1206	.
UP BND1	Y1206	1.00000
LO BND1	DY1206	.
UP BND1	DY1206	2.00000
LO BND1	X1209	.
UP BND1	X1209	1.00000
LO BND1	DX1209	.
UP BND1	DX1209	4.00000
LO BND1	X2209	.
UP BND1	X2209	1.00000
LO BND1	DX2209	.
UP BND1	DX2209	4.00000
FX BND1	X1207	1.00000
LO BND1	DX1207	.
UP BND1	DX1207	2.00000

Table 34 : TRANCHE input data (cont'd)

```
                EXECUTOR.  MPSX  RELEASE 1

 LO BND1       X2207              .
 UP BND1       X2207          1.00000
 LO BND1       DX2207             .
 UP BND1       DX2207         2.00000
 FX BND1       X1210          1.00000
 LO BND1       DX1210             .
 UP BND1       DX1210         4.00000
 LO BND1       X2210              .
 UP BND1       X2210          1.00000
 LO BND1       DX2210             .
 UP BND1       DX2210         2.00000
 FX BND1       X1202          2.00000
 LO BND1       DX1202             .
 UP BND1       DX1202         8.00000
 LO BND1       X2202              .
 UP BND1       X2202          1.00000
 LO BND1       DX2202             .
 UP BND1       DX2202         4.00000
 LO BND1       Y1202              .
 UP BND1       Y1202          1.00000
 FX BND1       DY1202             .
 FX BND1       X1205          1.00000
 LO BND1       DX1205             .
 UP BND1       DX1205         1.00000
 LO BND1       X2205              .
 UP BND1       X2205          1.00000
 LO BND1       DX2205             .
 UP BND1       DX2205         4.00000
 FX BND1       X1208          1.00000
 LO BND1       DX1208             .
 UP BND1       DX1208         1.00000
 LO BND1       X2208              .
 UP BND1       X2208          1.00000
 LO BND1       DX2208             .
 UP BND1       DX2208         4.00000
 FX BND1       X1203          1.00000
 LO BND1       DX1203             .
 UP BND1       DX1203         1.00000
 LO BND1       X2203              .
 UP BND1       X2203          1.00000
 LO BND1       DX2203             .
 UP BND1       DX2203         4.00000
ENDATA
```

Table 34 : TRANCHE input data (cont'd)

```
                EXECUTOR.  MPSX  RELEASE 1

SETUP  PROB

TIME =    0.02

    BOUND     = BND1
    SCALE


 MATRIX1  ASSIGNED TO MATRIX1
 MATRIX2  ASSIGNED TO MATRIX2

 ETA1     ASSIGNED TO ETA1
 ETA2     ASSIGNED TO ETA2

 SCRATCH1 ASSIGNED TO SCRATCH1
 SCRATCH2 ASSIGNED TO SCRATCH2
 MPSCRAT  ASSIGNED TO MPSCRAT

 MIXWORK  ASSIGNED TO MIXWORK

 MAXIMUM PRICING NOT REQUIRED - MAXIMUM POSSIBLE    7

 NO CYCLING

     POOLS          NUMBER    SIZE     CORE
 H.REG-BITS MAP                          277
 INTEGER TABLES                         4480
 BOUND  VECTOR                           512
 WORK   REGIONS       9          512    4608
 MATRIX BUFFERS       4         2288    9152
 ETA    BUFFERS       7         3504   24528

                          TOTAL  NORMAL  .FREE.   FIXED BOUNDED
 ROWS     (LOG.VAR.)         61      60       1       0       0
 COLUMNS (STR.VAR.)          99      31       0      33      35

      35 COLUMNS ARE INTEGER

      401 ELEMENTS - DENSITY =    4.10 -      4 MATRIX RECORDS (WITHOUT RHS'S)


PICTURE - USING PROB

TIME =    0.03
```

Table 35 : Listing of internal files and storage allocations for TRANCHE

Table 36 : Pictorial representation of problem matrix

		R1011001	R1021002	R1031003	R1041004	R1051005	R1061006	R1071007	R1081008	R1091009	R1101010	R1111011	R1121012	R1131001	R1141002	R1151003	R1161004	R1171005	R1181006	R1191007	R1201008	R1211009	R1221000	R1231000	R1241000	R1251000	R1261000	R1271000	R1281000	R1291000	R1301000	R1311001
Y1DY1202	G																															
Y2DY2202	G																															
Y1Y2202	G																															
C205	L			1	1					1	1				1			1			1									1		
X1DX1205	G																															
X2DX2205	G																															
X1X2205	G																															
C208	L					1	1			1	1	1	1			1		1		1											1	
X1DX1208	G																															
X2DX2208	G																															
X1X2208	G																															
C203	L							1	1			1	1				1			1		1										1
X1DX1203	G																															
X2DX2203	G																															
X1X2203	G																															
COST1	N																															

		IDX1201	IDX2201	IDX1204	IDX2204	IDX1206	IDX2206	IDY1206	IDY2206	IDX1209	IDX2209	IDX1207	IDX2207	IDX1210	IDX2210	IDX1202	IDX2202	IDY1202	IDY2202	IDX1205	IDX2205	IDX1208	IDX2208	IDX1203	IDX2203	
LOWER BOUND		0	0	1	0	A	0	0	0	0	0	A	0	0	0	0	0	0	0	A	0	A	0	A	0	2, 1
UPPER BOUND		0	0	1	0	A	0	0	0	0	0	A	0	0	0	0	0	0	0	A	0	A	0	A	0	
Y1DY1202	G																									
Y2DY2202	G																									
Y1Y2202	G																	1								
C205	L																			-1	-1					
X1DX1205	G																									
X2DX2205	G																									
X1X2205	G																			1						
C208	L																					-1	-1			
X1DX1208	G																									
X2DX2208	G																									
X1X2208	G																					1				
C203	L																							-1	-1	
X1DX1203	G																									
X2DX2203	G																									
X1X2203	G																							1		
COST1	N																									

Table 36 : Pictorial representation of problem matrix (cont'd)

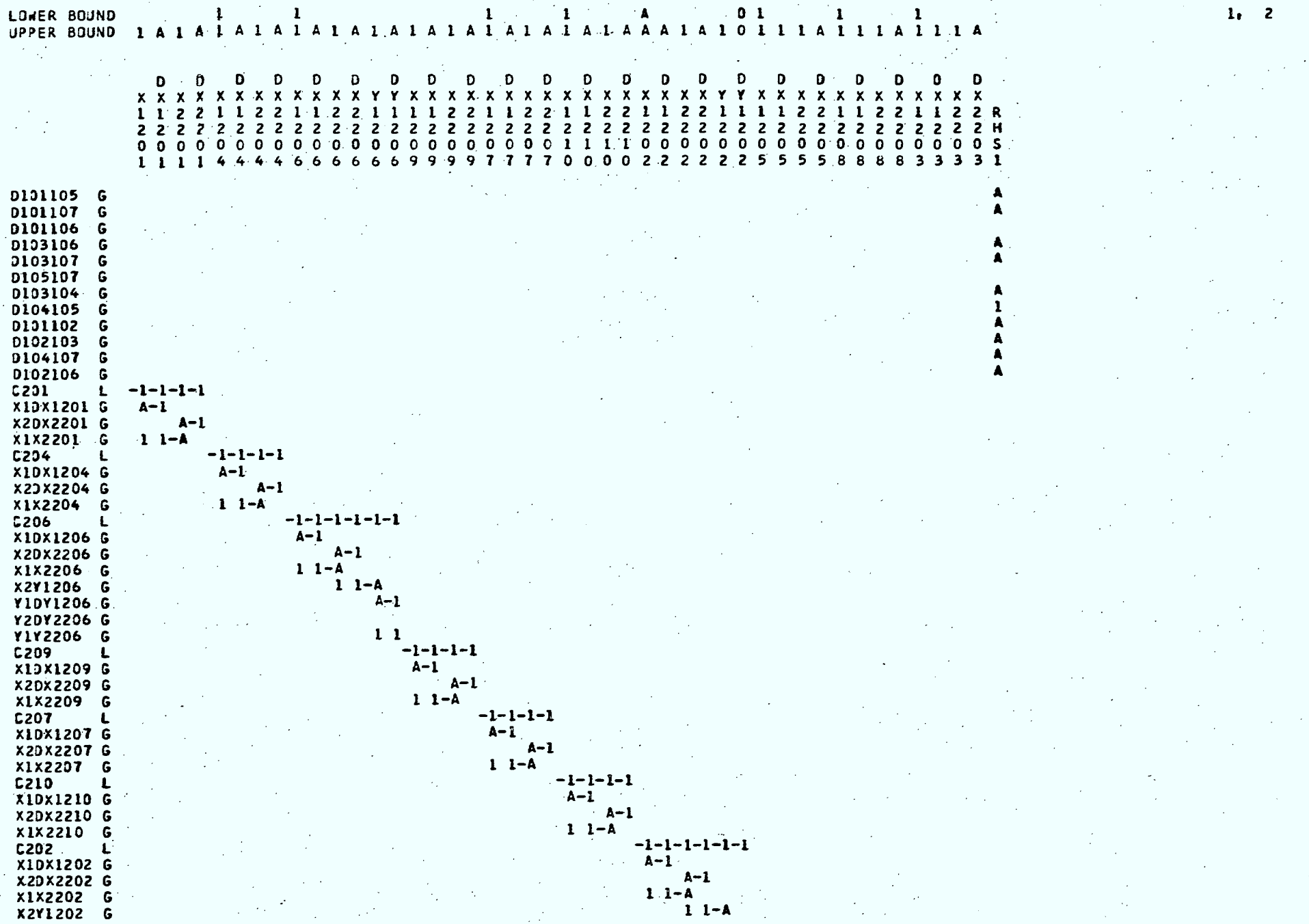


Table 36 : Pictorial representation of problem matrix (cont'd)

2, 2

		X1201	OX1201	X2201	DX2201	X1204	DX1204	X2204	DX2204	X1206	DX1206	X2206	DX2206	Y1206	DY1206	X1209	DX1209	X2209	DX2209	X1207	OX1207	X2207	DX2207	X1210	DX1210	X2210	DX2210	X1202	OX1202	X2202	DX2202	Y1202	OY1202	X1205	DX1205	X2205	DX2205	X1208	OX1208	X2208	OX2208	X1203	OX1203	X2203	OX2203	RHS1
LOWER BOUND						1				1										1				1				A					0	1				1				1				
UPPER BOUND		1	A	1	A	1	A	1	A	1	A	1	A	1	A	1	A	1	A	1	A	1	A	1	A	1	A	A	A	1	A	1	0	1	1	1	A	1	1	1	A	1	1	1	A	
Y1DY1202	G																															A	-1													
Y2OY2202	G																																													
Y1Y2202	G																															1	1													
C205	L																																	-1	-1	-1	-1									
X1DX1205	G																																	A	-1											
X2DX2205	G																																			A	-1									
X1X2205	G																																	1	1	-A										
C208	L																																					-1	-1	-1	-1					
X1DX1208	G																																					A	-1							
X2OX2208	G																																							A	-1					
X1X2208	G																																					1	1	-A						
C203	L																																									-1	-1	-1	-1	
X1DX1203	G																																									A	-1			
X2DX2203	G																																											A	-1	
X1X2203	G																																									1	1	-A		
COST1	N	C	B	C	B		C	C	C		C	C	C	C	C	C	B	C	B		C	C	C		C	D	C		B	C	B	C	B		C	O	C		C	C	C		C	C	C	

Table 36 : Pictorial representation of problem matrix (cont'd)

EXECUTOR. MPSX RELEASE 1

SUMMARY OF MATRIX

SYMBOL	RANGE			COUNT (INCL.RHS)
Z	LESS THAN		.000001	
Y	.000001	THRU	.000009	
X	.000010		.000099	
W	.000100		.000999	
V	.001000		.009999	
U	.010000		.099999	
T	.100000		.999999	
1	1.000000		1.000000	271
A	1.000001		10.000000	43
B	10.000001		100.000000	7
C	100.000001		1,000.000000	27
D	1,000.000001		10,000.000000	2
E	10,000.000001		100,000.000000	
F	100,000.000001		1,000,000.000000	
G	GREATER THAN		1,000,000.000000	

MINIMUM = .100000E+01 MAXIMUM = .115500E+04

Table 37 : Summary of matrix statistics

```
            EXECUTOR.  MPSX  RELEASE 1

WRITE

TIME =    0.04

    OPTIMIZE SYSTEM MACRO CALLED


CRASH          TIME    0.04 MINS.
        INFEASIBILITIES     10  AT START
                             4  AFTER PASS A


 INVERT CALLED     TIME    0.04   CURRENT INVERSE ---- ETA-VECTORS ....50   ELEMENTS ....50   RECORDS .....1   ITERATION .....38
  BASIS ---- NO.OF ROWS ....61   LOGICALS .......33   STRUCTURALS ....28   ELEMENTS ...139
  INVERSE -- NUCLEUS ........0   TRANSFORMED .....0   ETA-VECTORS ....54   ELEMENTS ...132   RECORDS .....1   TIME TAKEN     0.000

                             0  AFTER PASS B
                      FEASIBLE
```

Table 38 : Crash

```
VEGDJ

TIME =     0.05  MINS.

PRIMAL        OBJ = COST1        RHS = RHS1

TIME =     0.05  MINS.        PRICING     7
 SCALE =    1.00000

   ITER    NUMBER  VECTOR  VECTOR  REDUCED    FUNCTION
  NUMBER  NONOPT    OUT      IN      COST       VALUE
M    48        6     62     128    .89165    10028.3
     49          25     49    .84814-   9589.44
     50          14    117    .59505    9216.11
     51           3    120    .42081    8842.78
     52         122    124    .56374    8752.11
 INVERT DEMANDED AFTER  10 MAJOR/  14 MINOR ITERATIONS - CLOCK CONTROL

 INVERT CALLED     TIME    0.05   CURRENT INVERSE ---- ETA-VECTORS ....67   ELEMENTS ...216   RECORDS .....1   ITERATION .....52
  BASIS ---- NO.OF ROWS ....61   LOGICALS .......26   STRUCTURALS ....35   ELEMENTS ...156
  INVERSE -- NUCLEUS .......10   TRANSFORMED .....4   ETA-VECTORS ....67   ELEMENTS ...164   RECORDS .....1   TIME TAKEN     0.001

VEGDJ

TIME =     0.05  MINS.

PRIMAL        OBJ = COST1        RHS = RHS1

TIME =     0.05  MINS.        PRICING     7
 SCALE =    1.00000

   ITER    NUMBER  VECTOR  VECTOR  REDUCED    FUNCTION
  NUMBER  NONOPT    OUT      IN      COST       VALUE
M    53        6     17     20   1.19380-   8496.11
```

Table 39 : Primal

EXECUTOR. MPSX RELEASE 1

	ITER NUMBER	NUMBER NONOPT	VECTOR OUT	VECTOR IN	REDUCED COST	FUNCTION VALUE
	54		1	21	.15807-	8460.00
	55		79	7	.23405-	8441.89
M	56	6	49	78	.23405-	8188.33
	57		123	122	.05969-	8175.00
	58		76	80	.02728-	8166.55
	59		117	73	.02728-	8164.44
M	60	6	133	134	.24046	7844.44
	61		6	132	.28054	7527.11
	62		21	130	1.37396	7376.44
	63		7	142	2.47405	7299.63
	64		43	146	.37542-	7299.63
M	65	6	57	160	.43109	6692.96
	66		78	152	.44375	6382.96
	67		140	156	1.84267	6226.00
M	68	6	142	40	5.01477-	4762.00
	69		130	25	.03381-	4737.00
	70		70	63	1.01663-	4737.00
	71		67	62	.26708-	4737.00
M	72	6	126	126	.13375-	4627.00
	73		122	117	12.0126-	4375.78
M	74	6	150	150	.16533-	4298.78
	75		154	154	.13853-	4239.78
	76		73	138	1.14129	4148.00
	77		32	133	.12023-	4124.00
M	78	6	158	158	.06466-	4098.00
	79		156	74	.59303-	3685.00
	80		160	78	.19672-	3685.00
M	81	4	158	158	.27607	3574.00
	82		152	123	2.47879-	3275.00
	83		138	154	.50950	3058.00
M	84	1	150	150	.47668	2836.00

OPTIMAL SOLUTION

Table 39 : Primal (cont'd)

EXECUTOR. MPSX RELEASE 1

SOLUTION (OPTIMAL)

TIME = 0.07 MINS. ITERATION NUMBER = 84

...NAME...	...ACTIVITY...	DEFINED AS
FUNCTIONAL	2836.00000	COST1
RESTRAINTS		RHS1
BOUNDS....		BND1

Table 40 : Optimal continuous solution

EXECUTOR.. MPSX RELEASE 1

SECTION 1 - ROWS

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
1	D101105	LL	3.00000	.	3.00000	NONE	634.00000-
2	D101107	LL	2.00000	.	2.00000	NONE	558.00000-
3	D101106	LL	.	.	.	NONE	84.00000-
4	D103106	LL	2.00000	.	2.00000	NONE	144.00000-
5	D103107	LL	3.00000	.	3.00000	NONE	330.00000-
6	D105107	LL	.	.	.	NONE	76.00000-
7	D103104	LL	3.00000	.	3.00000	NONE	35.00000-
8	D104105	LL	1.00000	.	1.00000	NONE	371.00000-
9	D101102	LL	2.00000	.	2.00000	NONE	65.00000-
10	D102103	LL	4.00000	.	4.00000	NONE	163.00000-
11	D104107	LL	4.00000	.	4.00000	NONE	295.00000-
12	D102106	LL	4.00000	.	4.00000	NONE	19.00000-
13	C201	UL	.	.	NONE	.	84.00000
14	X1DX1201	LL	.	.	.	NONE	48.00000-
15	X2DX2201	LL	.	.	.	NONE	24.00000-
16	X1X2201	LL	.	.	.	NONE	24.00000-
17	C204	UL	.	.	NONE	.	144.00000
18	X1DX1204	BS	6.00000	6.00000-	.	NONE	.
19	X2DX2204	LL	.	.	.	NONE	24.00000-
20	X1X2204	BS	4.00000	4.00000-	.	NONE	.
21	C206	UL	.	.	NONE	.	330.00000
22	X1DX1206	BS	7.00000	7.00000-	.	NONE	.
23	X2DX2206	LL	.	.	.	NONE	55.00000-
24	X1X2206	BS	3.50000	3.50000-	.	NONE	.
25	X2Y1206	BS	3.00000	3.00000-	.	NONE	.
26	Y1DY1206	LL	.	.	.	NONE	70.00000-
27	Y2DY2206	BS	.	.	.	NONE	.
28	Y1Y2206	BS	.	.	.	NONE	.
29	C209	UL	.	.	NONE	.	76.00000
30	X1DX1209	LL	.	.	.	NONE	24.00000-
31	X2DX2209	LL	.	.	.	NONE	16.00000-
32	X1X2209	LL	.	.	.	NONE	8.00000-
33	C207	UL	.	.	NONE	.	35.00000
34	X1DX1207	BS	9.00000	9.00000-	.	NONE	.
35	X2DX2207	LL	.	.	.	NONE	66.11111-
36	X1X2207	BS	3.00000	3.00000-	.	NONE	.
37	C210	UL	.	.	NONE	.	371.00000
38	X1DX1210	BS	9.00000	9.00000-	.	NONE	.
39	X2DX2210	LL	.	.	.	NONE	87.11111-
40	X1X2210	BS	1.00000	1.00000-	.	NONE	.
41	C202	UL	.	.	NONE	.	65.00000
42	X1DX1202	BS	18.00000	18.00000-	.	NONE	.
43	X2DX2202	LL	.	.	.	NONE	13.00000-
44	X1X2202	BS	2.00000	2.00000-	.	NONE	.
45	X2Y1202	LL	.	.	.	NONE	13.00000-
46	Y1DY1202	BS	.	.	.	NONE	.
47	Y2DY2202	BS	.	.	.	NONE	.
48	Y1Y2202	LL	.	.	.	NONE	195.00000-
49	C205	UL	.	.	NONE		163.00000

Table 41 : Continuous solution, rows section

EXECUTOR. MPSX RELEASE 1

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
50	X1DX1205	BS	9.00000	9.00000-	.	NONE	.
51	X2DX2205	LL	.	.	.	NONE	110.22222-
52	X1X2205	BS	4.00000	4.00000-	.	NONE	.
53	C208	UL	.	.	NONE	.	295.00000
54	X1DX1208	BS	9.00000	9.00000-	.	NONE	.
55	X2DX2208	LL	.	.	.	NONE	65.55556-
56	X1X2208	BS	4.00000	4.00000-	.	NONE	.
57	C203	UL	.	.	NONE	.	19.00000
58	X1DX1203	BS	9.00000	9.00000-	.	NONE	.
59	X2DX2203	LL	.	.	.	NONE	41.22222-
60	X1X2203	BS	4.00000	4.00000-	.	NONE	.
61	COST1	BS	2836.00000	2836.00000-	NONE	NONE	1.00000

Table 41 : Continuous solution, rows section (cont'd)

EXECUTOR. MPSX RELEASE 1

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	62	R1011001	BS	3.00000	.	.	NONE	.
	63	R1021002	BS	.	.	.	NONE	.
	64	R1031003	BS	.	.	.	NONE	.
A	65	R1041004	LL	.	.	.	NONE	.
	66	R1051005	LL	.	.	.	NONE	590.00000
A	67	R1061006	LL	.	.	.	NONE	.
A	68	R1071007	LL	.	.	.	NONE	.
A	69	R1081008	LL	.	.	.	NONE	.
	70	R1091009	LL	.	.	.	NONE	590.00000
A	71	R1101010	LL	.	.	.	NONE	.
	72	R1111011	LL	.	.	.	NONE	590.00000
A	73	R1121012	LL	.	.	.	NONE	.
	74	R1131001	BS	2.00000	.	.	NONE	.
A	75	R1141002	LL	.	.	.	NONE	.
A	76	R1151003	LL	.	.	.	NONE	.
A	77	R1161004	LL	.	.	.	NONE	.
	78	R1171005	BS	.	.	.	NONE	.
	79	R1181006	LL	.	.	.	NONE	152.00000
	80	R1191007	BS	.	.	.	NONE	.
	81	R1201008	LL	.	.	.	NONE	152.00000
	82	R1211009	LL	.	.	.	NONE	152.00000
	83	R1221000	BS	.	.	.	NONE	.
	84	R1231000	BS	2.00000	.	.	NONE	.
	85	R1241000	BS	3.00000	.	.	NONE	.
	86	R1251000	BS	.	.	.	NONE	.
	87	R1261000	BS	3.00000	.	.	NONE	.
	88	R1271000	BS	1.00000	.	.	NONE	.
	89	R1281000	BS	2.00000	.	.	NONE	.
	90	R1291000	BS	4.00000	.	.	NONE	.
	91	R1301000	BS	4.00000	.	.	NONE	.
	92	R1311000	BS	4.00000	.	.	NONE	.
	93	IDX1201	EQ	.	.	.	.	108.00000-
	94	IDX2201	EQ	.	.	.	.	84.00000-
	95	IDX1204	EQ	1.00000	.	1.00000	1.00000	144.00000-
	96	IDX2204	EQ	.	.	.	.	144.00000-
	97	IDX1206	EQ	2.00000	.	2.00000	2.00000	330.00000-
	98	IDX2206	EQ	.	.	.	.	330.00000-
	99	IDY1206	EQ	.	.	.	.	330.00000-
	100	IDY2206	EQ	.	.	.	.	330.00000-
	101	IDX1209	EQ	.	.	.	.	84.00000-
	102	IDX2209	EQ	.	.	.	.	76.00000-
	103	IDX1207	EQ	2.00000	.	2.00000	2.00000	35.00000-
	104	IDX2207	EQ	.	.	.	.	35.00000-
	105	IDX1210	EQ	.	.	.	.	371.00000-
	106	IDX2210	EQ	.	.	.	.	371.00000-
	107	IDX1202	EQ	.	.	.	.	65.00000-
	108	IDX2202	EQ	.	.	.	.	78.00000-
	109	IDY1202	EQ	.	.	.	.	260.00000-
	110	IDY2202	EQ	.	.	.	.	65.00000-

Table 42 : Continuous solution columns section

EXECUTOR. MPSX RELEASE 1

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
111	IDX1205	EQ	3.00000	.	3.00000	3.00000	163.00000-
112	IDX2205	EQ	.	.	.	.	163.00000-
113	IOX1208	EQ	3.00000	.	3.00000	3.00000	295.00000-
114	IOX2208	EQ	.	.	.	.	295.00000-
115	IOX1203	EQ	3.00000	.	3.00000	3.00000	19.00000-
116	IDX2203	EQ	.	.	.	.	19.00000-
117	X1201	BS	.16667	540.00000	.	1.00000	.
118	0X1201	BS	1.50000	60.00000	.	4.00000	.
119	X2201	BS	.33333	180.00000	.	1.00000	.
120	0X2201	BS	3.00000	60.00000	.	4.00000	.
121	X1204	EQ	1.00000	.	1.00000	1.00000	144.00000-
122	0X1204	UL	3.00000	120.00000	.	3.00000	24.00000-
123	X2204	BS	.20000	360.00000	.	1.00000	.
124	0X2204	BS	1.80000	120.00000	.	4.00000	.
125	X1206	EQ	1.00000	.	1.00000	1.00000	330.00000-
126	0X1206	UL	2.00000	275.00000	.	2.00000	55.00000-
127	X2206	BS	.30000	825.00000	.	1.00000	.
128	0X2206	BS	2.70000	275.00000	.	4.00000	.
129	Y1206	BS	.	960.00000	.	1.00000	.
130	0Y1206	LL	.	275.00000	.	2.00000	15.00000
131	X1209	BS	.10000	300.00000	.	1.00000	.
132	0X1209	BS	.90000	60.00000	.	4.00000	.
133	X2209	BS	.20000	180.00000	.	1.00000	.
134	0X2209	BS	1.80000	60.00000	.	4.00000	.
135	X1207	EQ	1.00000	.	1.00000	1.00000	35.00000-
136	0X1207	LL	.	210.00000	.	2.00000	175.00000
137	X2207	BS	.	630.00000	.	1.00000	.
138	0X2207	LL	.	210.00000	.	2.00000	241.11111
139	X1210	EQ	1.00000	.	1.00000	1.00000	371.00000-
140	0X1210	LL	.	385.00000	.	4.00000	14.00000
141	X2210	BS	.	1155.00000	.	1.00000	.
142	0X2210	LL	.	385.00000	.	2.00000	101.11111
143	X1202	EQ	2.00000	.	2.00000	2.00000	65.00000-
144	0X1202	BS	.	65.00000	.	8.00000	.
145	X2202	BS	.	195.00000	.	1.00000	.
146	0X2202	BS	.	65.00000	.	4.00000	.
147	Y1202	BS	.	195.00000	.	1.00000	.
148	0Y1202	EQ	.	65.00000	.	.	195.00000-
149	X1205	EQ	1.00000	.	1.00000	1.00000	163.00000-
150	0X1205	LL	.	385.00000	.	1.00000	222.00000
151	X2205	BS	.	1155.00000	.	1.00000	.
152	0X2205	LL	.	385.00000	.	4.00000	332.22222
153	X1208	EQ	1.00000	.	1.00000	1.00000	295.00000-
154	0X1208	BS	.	295.00000	.	1.00000	.
155	X2208	BS	.	885.00000	.	1.00000	.
156	0X2208	LL	.	295.00000	.	4.00000	65.55556
157	X1203	EQ	1.00000	.	1.00000	1.00000	19.00000-
158	0X1203	LL	.	130.00000	.	1.00000	111.00000
159	X2203	BS	.	390.00000	.	1.00000	.
160	0X2203	LL	.	130.00000	.	4.00000	152.22222

Table 42 : Continuous solution columns section (cont'd)

```
                EXECUTOR.  MPSX  RELEASE 1

MIXSTART - TIME =    0.07

    COST

MIXFLOW - TIME =    0.08

-------------------------------------------------------------------------------
I NODES POSTPONED IF ESTIMATION BEYOND .100000E+76                            I
I NODES POSTPONED IF FUNCTIONAL BEYOND 9500.00                                I
I NODES DROPPED   IF FUNCTIONAL BEYOND 9500.00                                I
I REACH   NODE     1 - FUNCTIONAL =     2836.0000  ESTIMATION = 2836.00      ITER ......84   I
I FIRST  BRANCH - VARIABLE   127 =    .3000    FORCED TOWARDS UPPER ACTIVITIES  I
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
I REACH   NODE     2 - FUNCTIONAL =     3221.0000  ESTIMATION = 3221.00      ITER ......87   I
I RESTORE NODE     1 - FUNCTIONAL =     2836.0000  ESTIMATION = 2836.00                      I
I SECOND BRANCH - VARIABLE   127 =    .3000    FORCED TOWARDS LOWER ACTIVITIES  I
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
I REACH   NODE     3 - FUNCTIONAL =     3431.0000  ESTIMATION = 3431.00      ITER ......92   I
I BEST POSSIBLE SOLUTION IS NOW       3221.00                                 I
I RESTORE NODE     2 - FUNCTIONAL =     3221.0000  ESTIMATION = 3221.00                      I
I FIRST  BRANCH - VARIABLE   117 =    .1667    FORCED TOWARDS UPPER ACTIVITIES  I
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
I REACH   NODE     4 - FUNCTIONAL =     3581.0000  ESTIMATION = 3581.00      ITER ......94   I
I RESTORE NODE     2 - FUNCTIONAL =     3221.0000  ESTIMATION = 3221.00                      I
I SECOND BRANCH - VARIABLE   117 =    .1667    FORCED TOWARDS LOWER ACTIVITIES  I
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
I REACH   NODE     5 - FUNCTIONAL =     3880.0000  ESTIMATION = 3880.00      ITER .....100   I
I BEST POSSIBLE SOLUTION IS NOW       3431.00                                 I
I RESTORE NODE     4 - FUNCTIONAL =     3581.0000  ESTIMATION = 3581.00                      I
I FIRST  BRANCH - VARIABLE   123 =    .2000    FORCED TOWARDS UPPER ACTIVITIES  I
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
I REACH   NODE     6 - FUNCTIONAL =     3773.0000  ESTIMATION = 3773.00      ITER .....101   I
I RESTORE NODE     4 - FUNCTIONAL =     3581.0000  ESTIMATION = 3581.00                      I
I SECOND BRANCH - VARIABLE   123 =    .2000    FORCED TOWARDS LOWER ACTIVITIES  I
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
I REACH   NODE     7 - FUNCTIONAL =     4150.0000  ESTIMATION = 4150.00      ITER .....106   I
I RESTORE NODE     6 - FUNCTIONAL =     3773.0000  ESTIMATION = 3773.00                      I
I FIRST  BRANCH - VARIABLE   133 =    .2000    FORCED TOWARDS UPPER ACTIVITIES  I
-------------------------------------------------------------------------------
```

Table 43 : Search for first integer solution

```
EXECUTOR.  MPSX  RELEASE 1

-------------------------------------------------------------------------------
I REACH   NODE     8 - FUNCTIONAL =     4265.0000  ESTIMATION = 4265.00      ITER .....109   I
I  - - - - - - -  INTEGER SOLUTION                                                             I
-------------------------------------------------------------------------------
```

Table 43 : Search for first integer solution (cont'd)

```
                EXECUTOR.  MPSX  RELEASE 1

SOLUTION

TIME =    0.10 MINS. ITERATION NUMBER =    109

          ...NAME...     ...ACTIVITY...        DEFINED AS

          FUNCTIONAL        4265.00000         COST1
          RESTRAINTS                           RHS1
          BOUNDS....                           BND1
```

Table 44 : First integer solution

EXECUTOR. MPSX RELEASE 1

SECTION 1 - ROWS

	NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	D101105	LL	3.00000	.	3.00000	NONE	467.00000-
	2	D101107	LL	2.00000	.	2.00000	NONE	467.00000-
	3	D101106	LL	.	.	.	NONE	72.00000-
	4	D103106	LL	2.00000	.	2.00000	NONE	120.00000-
	5	D103107	LL	3.00000	.	3.00000	NONE	275.00000-
A	6	D105107	LL	.	.	.	NONE	.
A	7	D103104	LL	3.00000	.	3.00000	NONE	.
	8	D104105	LL	1.00000	.	1.00000	NONE	275.00000-
	9	D101102	LL	2.00000	.	2.00000	NONE	65.00000-
	10	D102103	LL	4.00000	.	4.00000	NONE	127.00000-
	11	D104107	LL	4.00000	.	4.00000	NONE	295.00000-
	12	D102106	LL	4.00000	.	4.00000	NONE	7.00000-
	13	C201	UL	.	.	NONE	.	72.00000
	14	X1DX1201	BS	5.00000	5.00000-	.	NONE	.
	15	X2DX2201	LL	.	.	.	NONE	12.00000-
	16	X1X2201	BS	5.00000	5.00000-	.	NONE	.
	17	C204	UL	.	.	NONE	.	120.00000
	18	X1DX1204	BS	6.00000	6.00000-	.	NONE	.
	19	X2DX2204	BS	8.00000	8.00000-	.	NONE	.
	20	X1X2204	BS	.	.	.	NONE	.
	21	C206	UL	.	.	NONE	.	275.00000
	22	X1DX1206	BS	7.00000	7.00000-	.	NONE	.
	23	X2DX2206	BS	7.00000	7.00000-	.	NONE	.
	24	X1X2206	BS	.	.	.	NONE	.
	25	X2Y1206	BS	3.00000	3.00000-	.	NONE	.
	26	Y1DY1206	LL	.	.	.	NONE	76.11111-
	27	Y2DY2206	BS	.	.	.	NONE	.
	28	Y1Y2206	BS	.	.	.	NONE	.
	29	C209	BS	3.00000-	3.00000	NONE	.	.
	30	X1DX1209	BS	5.00000	5.00000-	.	NONE	.
	31	X2DX2209	BS	9.00000	9.00000-	.	NONE	.
	32	X1X2209	LL	.	.	.	NONE	300.00000-
	33	C207	BS	.	.	NONE	.	.
	34	X1DX1207	BS	9.00000	9.00000-	.	NONE	.
	35	X2DX2207	LL	.	.	.	NONE	70.00000-
	36	X1X2207	BS	3.00000	3.00000-	.	NONE	.
	37	C210	UL	.	.	NONE	.	275.00000
	38	X1DX1210	BS	9.00000	9.00000-	.	NONE	.
	39	X2DX2210	LL	.	.	.	NONE	97.77778-
	40	X1X2210	BS	1.00000	1.00000-	.	NONE	.
	41	C202	UL	.	.	NONE	.	65.00000
	42	X1DX1202	BS	18.00000	18.00000-	.	NONE	.
	43	X2DX2202	LL	.	.	.	NONE	13.00000-
	44	X1X2202	BS	2.00000	2.00000-	.	NONE	.
	45	X2Y1202	LL	.	.	.	NONE	13.00000-
	46	Y1DY1202	BS	.	.	.	NONE	.
	47	Y2DY2202	BS	.	.	.	NONE	.
	48	Y1Y2202	LL	.	.	.	NONE	195.00000-
	49	C205	UL	.	.	NONE	.	127.00000

Table 45 : First integer solution (row's section)

EXECUTOR. MPSX RELEASE 1

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
50	X10X1205	BS	9.00000	9.00000-	.	NONE	.
51	X20X2205	LL	.	.	.	NONE	114.22222-
52	X1X2205	BS	4.00000	4.00000-	.	NONE	.
53	C208	UL	.	.	NONE	.	295.00000
54	X1DX1208	BS	9.00000	9.00000-	.	NONE	.
55	X2DX2208	LL	.	.	.	NONE	65.55556-
56	X1X2208	BS	4.00000	4.00000-	.	NONE	.
57	C203	UL	.	.	NONE	.	7.00000
58	X10X1203	BS	9.00000	9.00000-	.	NONE	.
59	X2DX2203	LL	.	.	.	NONE	42.55556-
60	X1X2203	BS	4.00000	4.00000-	.	NONE	.
61	COST1	BS	4265.00000	4265.00000-	NONE	NONE	1.00000

Table 45 : First integer solution (row's section)

EXECUTOR. MPSX RELEASE 1

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	62	R1011001	BS	3.00000	.	.	NONE	.
	63	R1021002	BS	.	.	.	NONE	.
	64	R1031003	BS	.	.	.	NONE	.
A	65	R1041004	LL	.	.	.	NONE	.
	66	R1051005	LL	.	.	.	NONE	570.00000
	67	R1061006	LL	.	.	.	NONE	20.00000
	68	R1071007	BS	.	.	.	NONE	.
A	69	R1081008	LL	.	.	.	NONE	.
	70	R1091009	LL	.	.	.	NONE	570.00000
	71	R1101010	LL	.	.	.	NONE	20.00000
	72	R1111011	LL	.	.	.	NONE	570.00000
	73	R1121012	LL	.	.	.	NONE	20.00000
	74	R1131001	BS	2.00000	.	.	NONE	.
A	75	R1141002	LL	.	.	.	NONE	.
	76	R1151003	LL	.	.	.	NONE	20.00000
A	77	R1161004	LL	.	.	.	NONE	.
	78	R1171005	LL	.	.	.	NONE	20.00000
A	79	R1181006	LL	.	.	.	NONE	.
	80	R1191007	LL	.	.	.	NONE	20.00000
A	81	R1201008	LL	.	.	.	NONE	.
A	82	R1211009	LL	.	.	.	NONE	.
	83	R1221000	BS	.	.	.	NONE	.
	84	R1231000	BS	2.00000	.	.	NONE	.
	85	R1241000	BS	3.00000	.	.	NONE	.
	86	R1251000	BS	.	.	.	NONE	.
	87	R1261000	BS	3.00000	.	.	NONE	.
	88	R1271000	BS	1.00000	.	.	NONE	.
	89	R1281000	BS	2.00000	.	.	NONE	.
	90	R1291000	BS	4.00000	.	.	NONE	.
	91	R1301000	BS	4.00000	.	.	NONE	.
	92	R1311000	BS	4.00000	.	.	NONE	.
	93	I0X1201	EQ	.	.	.	.	72.00000-
	94	I0X2201	EQ	.	.	.	.	72.00000-
	95	I0X1204	EQ	1.00000	.	1.00000	1.00000	120.00000-
	96	I0X2204	EQ	.	.	.	.	120.00000-
	97	I0X1206	EQ	2.00000	.	2.00000	2.00000	275.00000-
	98	I0X2206	EQ	.	.	.	.	275.00000-
	99	I0Y1206	EQ	.	.	.	.	275.00000-
	100	I0Y2206	EQ	.	.	.	.	275.00000-
	101	I0X1209	EQ	.	.	.	.	300.00000-
A	102	I0X2209	EQ	.	.	.	.	.
A	103	I0X1207	EQ	2.00000	.	2.00000	2.00000	.
A	104	I0X2207	EQ	.	.	.	.	.
	105	I0X1210	EQ	.	.	.	.	275.00000-
	106	I0X2210	EQ	.	.	.	.	275.00000-
	107	I0X1202	EQ	.	.	.	.	65.00000-
	108	I0X2202	EQ	.	.	.	.	78.00000-
	109	I0Y1202	EQ	.	.	.	.	260.00000-
	110	I0Y2202	EQ	.	.	.	.	65.00000-

Table 46 : First integer solution (columns section)

EXECUTOR. MPSX RELEASE 1

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	111	IDX1205	EQ	3.00000	.	3.00000	3.00000	127.00000-
	112	IDX2205	EQ	.	.	.	.	127.00000-
	113	IDX1208	EQ	3.00000	.	3.00000	3.00000	295.00000-
	114	IDX2208	EQ	.	.	.	.	295.00000-
	115	IDX1203	EQ	3.00000	.	3.00000	3.00000	7.00000-
	116	IDX2203	EQ	.	.	.	.	7.00000-
	117	X1201	IV	1.00000	540.00000	.	1.00000	468.00000
	118	DX1201	IV	4.00000	60.00000	.	4.00000	12.00000-
	119	X2201	IV	.	180.00000	.	1.00000	.
	120	DX2201	IV	.	60.00000	.	4.00000	.
	121	X1204	EQ	1.00000	.	1.00000	1.00000	120.00000-
A	122	DX1204	IV	3.00000	120.00000	.	3.00000	.
	123	X2204	IV	1.00000	360.00000	.	1.00000	240.00000
	124	DX2204	IV	1.00000	120.00000	.	4.00000	.
	125	X1206	EQ	1.00000	.	1.00000	1.00000	275.00000-
A	126	DX1206	IV	2.00000	275.00000	.	2.00000	.
	127	X2206	IV	1.00000	825.00000	.	1.00000	550.00000
	128	DX2206	IV	2.00000	275.00000	.	4.00000	.
	129	Y1206	IV	.	960.00000	.	1.00000	.
	130	DY1206	IV	.	275.00000	.	2.00000	76.11111
	131	X1209	IV	1.00000	300.00000	.	1.00000	.
	132	DX1209	IV	4.00000	60.00000	.	4.00000	240.00000-
	133	X2209	IV	1.00000	180.00000	.	1.00000	1680.00000
	134	DX2209	IV	.	60.00000	.	4.00000	60.00000
A	135	X1207	EQ	1.00000	.	1.00000	1.00000	.
	136	DX1207	IV	.	210.00000	.	2.00000	210.00000
	137	X2207	IV	.	630.00000	.	1.00000	.
	138	DX2207	IV	.	210.00000	.	2.00000	280.00000
	139	X1210	EQ	1.00000	.	1.00000	1.00000	275.00000-
	140	DX1210	IV	.	385.00000	.	4.00000	110.00000
	141	X2210	IV	.	1155.00000	.	1.00000	.
	142	DX2210	IV	.	385.00000	.	2.00000	207.77778
	143	X1202	EQ	2.00000	.	2.00000	2.00000	65.00000-
	144	DX1202	IV	.	65.00000	.	8.00000	.
	145	X2202	IV	.	195.00000	.	1.00000	.
	146	DX2202	IV	.	65.00000	.	4.00000	.
	147	Y1202	IV	.	195.00000	.	1.00000	.
	148	DY1202	EQ	.	65.00000	.	.	195.00000-
	149	X1205	EQ	1.00000	.	1.00000	1.00000	127.00000-
	150	DX1205	IV	.	385.00000	.	1.00000	258.00000
	151	X2205	IV	.	1155.00000	.	1.00000	.
	152	DX2205	IV	.	385.00000	.	4.00000	372.22222
	153	X1208	EQ	1.00000	.	1.00000	1.00000	295.00000-
	154	DX1208	IV	.	295.00000	.	1.00000	.
	155	X2208	IV	.	885.00000	.	1.00000	.
	156	DX2208	IV	.	295.00000	.	4.00000	65.55556
	157	X1203	EQ	1.00000	.	1.00000	1.00000	7.00000-
	158	DX1203	IV	.	130.00000	.	1.00000	123.00000
	159	X2203	IV	.	390.00000	.	1.00000	.
	160	DX2203	IV	.	130.00000	.	4.00000	165.55556

Table 46 : First integer solution (columns section)

```
                EXECUTOR.  MPSX  RELEASE 1

MIXFLOW - TIME =    0.11

 ------------------------------------------------------------------------------------------
 I NODES POSTPONED IF FUNCTIONAL BEYOND 4265.00                                           I
 I NODES DROPPED   IF FUNCTIONAL BEYOND 4265.00                                           I
 I RESTORE NODE     6 - FUNCTIONAL =      3773.0000  ESTIMATION = 3773.00                 I
 I SECOND BRANCH - VARIABLE   133 =    .2000    FORCED TOWARDS LOWER ACTIVITIES           I
 ------------------------------------------------------------------------------------------

 ------------------------------------------------------------------------------------------
 I REACH   NODE     9 - FUNCTIONAL =      3797.0000  ESTIMATION = 3797.00      ITER .....110  I
 I FIRST  BRANCH - VARIABLE   131 =    .3000    FORCED TOWARDS UPPER ACTIVITIES           I
 ------------------------------------------------------------------------------------------

 ------------------------------------------------------------------------------------------
 I REACH   NODE    10 - FUNCTIONAL =      3965.0000  ESTIMATION = 3965.00      ITER .....111  I
 I  - - - - - - -  INTEGER SOLUTION                                                       I
 ------------------------------------------------------------------------------------------
```

Table 47 : Search for second integer solution

EXECUTOR. MPSX RELEASE 1

SOLUTION

TIME = 0.12 MINS. ITERATION NUMBER = 111

...NAME...	...ACTIVITY...	DEFINED AS
FUNCTIONAL	3965.00000	COST1
RESTRAINTS		RHS1
BOUNDS....		BND1

Table 48 : Second integer solution

EXECUTOR. MPSX RELEASE 1

SECTION 1 - ROWS

	NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
	1	D101105	LL	3.00000	.	3.00000	NONE	527.00000-
	2	D101107	LL	2.00000	.	2.00000	NONE	467.00000-
	3	D101106	LL	.	.	.	NONE	72.00000-
	4	D103106	LL	2.00000	.	2.00000	NONE	120.00000-
	5	D103107	LL	3.00000	.	3.00000	NONE	275.00000-
	6	D105107	LL	.	.	.	NONE	60.00000-
A	7	D103104	LL	3.00000	.	3.00000	NONE	.
	8	D104105	LL	1.00000	.	1.00000	NONE	335.00000-
	9	D101102	LL	2.00000	.	2.00000	NONE	65.00000-
	10	D102103	LL	4.00000	.	4.00000	NONE	127.00000-
	11	D104107	LL	4.00000	.	4.00000	NONE	295.00000-
	12	D102106	LL	4.00000	.	4.00000	NONE	7.00000-
	13	C201	UL	.	.	NONE	.	72.00000
	14	X1DX1201	BS	5.00000	5.00000-	.	NONE	.
	15	X2DX2201	LL	.	.	.	NONE	12.00000-
	16	X1X2201	BS	5.00000	5.00000-	.	NONE	.
	17	C204	UL	.	.	NONE	.	120.00000
	18	X1DX1204	BS	6.00000	6.00000-	.	NONE	.
	19	X2DX2204	BS	8.00000	8.00000-	.	NONE	.
	20	X1X2204	BS	.	.	.	NONE	.
	21	C206	UL	.	.	NONE	.	275.00000
	22	X1DX1206	BS	7.00000	7.00000-	.	NONE	.
	23	X2DX2206	BS	7.00000	7.00000-	.	NONE	.
	24	X1X2206	BS	.	.	.	NONE	.
	25	X2Y1206	BS	3.00000	3.00000-	.	NONE	.
	26	Y1DY1206	LL	.	.	.	NONE	76.11111-
	27	Y2DY2206	BS	.	.	.	NONE	.
	28	Y1Y2206	BS	.	.	.	NONE	.
	29	C209	UL	.	.	NONE	.	60.00000
	30	X1DX1209	BS	7.00000	7.00000-	.	NONE	.
A	31	X2DX2209	LL	.	.	.	NONE	.
	32	X1X2209	BS	3.00000	3.00000-	.	NONE	.
	33	C207	BS	.	.	NONE	.	.
	34	X1DX1207	BS	9.00000	9.00000-	.	NONE	.
	35	X2DX2207	LL	.	.	.	NONE	70.00000-
	36	X1X2207	BS	3.00000	3.00000-	.	NONE	.
	37	C210	UL	.	.	NONE	.	335.00000
	38	X1DX1210	BS	9.00000	9.00000-	.	NONE	.
	39	X2DX2210	LL	.	.	.	NONE	91.11111-
	40	X1X2210	BS	1.00000	1.00000-	.	NONE	.
	41	C202	UL	.	.	NONE	.	65.00000
	42	X1DX1202	BS	18.00000	18.00000-	.	NONE	.
	43	X2DX2202	LL	.	.	.	NONE	13.00000-
	44	X1X2202	BS	2.00000	2.00000-	.	NONE	.
	45	X2Y1202	LL	.	.	.	NONE	13.00000-
	46	Y1DY1202	BS	.	.	.	NONE	.
	47	Y2DY2202	BS	.	.	.	NONE	.
	48	Y1Y2202	LL	.	.	.	NONE	195.00000-
	49	C205	UL	.	.	NONE	.	127.00000

Table 49 : Second integer solution (rows section)

EXECUTOR. MPSX RELEASE 1

NUMBER	...ROW..	AT	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY
50	X1DX1205	BS	9.00000	9.00000-	.	NONE	.
51	X2DX2205	LL	.	.	.	NONE	114.22222-
52	X1X2205	BS	4.00000	4.00000-	.	NONE	.
53	C208	UL	.	.	NONE	.	295.00000
54	X1DX1208	BS	9.00000	9.00000-	.	NONE	.
55	X2DX2208	LL	.	.	.	NONE	65.55556-
56	X1X2208	BS	4.00000	4.00000-	.	NONE	.
57	C203	UL	.	.	NONE	.	7.00000
58	X1DX1203	BS	9.00000	9.00000-	.	NONE	.
59	X2DX2203	LL	.	.	.	NONE	42.55556-
60	X1X2203	BS	4.00000	4.00000-	.	NONE	.
61	COST1	BS	3965.00000	3965.00000-	NONE	NONE	1.00000

Table 49 : Second integer solution (rows section)

EXECUTOR. MPSX RELEASE 1

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	62	R1011001	BS	3.00000	.	.	NONE	.
	63	R1021002	BS	.	.	.	NONE	.
	64	R1031003	BS	.	.	.	NONE	.
A	65	R1041004	LL	.	.	.	NONE	.
	66	R1051005	LL	.	.	.	NONE	570.00000
	67	R1061006	LL	.	.	.	NONE	20.00000
	68	R1071007	BS	.	.	.	NONE	.
A	69	R1081008	LL	.	.	.	NONE	.
	70	R1091009	LL	.	.	.	NONE	570.00000
	71	R1101010	LL	.	.	.	NONE	20.00000
	72	R1111011	LL	.	.	.	NONE	570.00000
	73	R1121012	LL	.	.	.	NONE	20.00000
	74	R1131001	BS	2.00000	.	.	NONE	.
A	75	R1141002	LL	.	.	.	NONE	.
	76	R1151003	LL	.	.	.	NONE	20.00000
A	77	R1161004	LL	.	.	.	NONE	.
	78	R1171005	LL	.	.	.	NONE	20.00000
	79	R1181006	LL	.	.	.	NONE	120.00000
	80	R1191007	LL	.	.	.	NONE	20.00000
	81	R1201008	LL	.	.	.	NONE	120.00000
	82	R1211009	LL	.	.	.	NONE	120.00000
	83	R1221000	BS	.	.	.	NONE	.
	84	R1231000	BS	2.00000	.	.	NONE	.
	85	R1241000	BS	3.00000	.	.	NONE	.
	86	R1251000	BS	.	.	.	NONE	.
	87	R1261000	BS	3.00000	.	.	NONE	.
	88	R1271000	BS	1.00000	.	.	NONE	.
	89	R1281000	BS	2.00000	.	.	NONE	.
	90	R1291000	BS	4.00000	.	.	NONE	.
	91	R1301000	BS	4.00000	.	.	NONE	.
	92	R1311000	BS	4.00000	.	.	NONE	.
	93	IDX1201	EQ	.	.	.	.	72.00000-
	94	IDX2201	EQ	.	.	.	.	72.00000-
	95	IDX1204	EQ	1.00000	.	1.00000	1.00000	120.00000-
	96	IDX2204	EQ	.	.	.	.	120.00000-
	97	IDX1206	EQ	2.00000	.	2.00000	2.00000	275.00000-
	98	IDX2206	EQ	.	.	.	.	275.00000-
	99	IDY1206	EQ	.	.	.	.	275.00000-
	100	IDY2206	EQ	.	.	.	.	275.00000-
	101	IDX1209	EQ	.	.	.	.	60.00000-
	102	IDX2209	EQ	.	.	.	.	60.00000-
A	103	IDX1207	EQ	2.00000	.	2.00000	2.00000	.
A	104	IDX2207	EQ	.	.	.	.	.
	105	IDX1210	EQ	.	.	.	.	335.00000-
	106	IDX2210	EQ	.	.	.	.	335.00000-
	107	IDX1202	EQ	.	.	.	.	65.00000-
	108	IDX2202	EQ	.	.	.	.	78.00000-
	109	IDY1202	EQ	.	.	.	.	260.00000-
	110	IDY2202	EQ	.	.	.	.	65.00000-

Table 50 : Second integer solution (columns section)

EXECUTOR. MPSX RELEASE 1

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT.	.REDUCED COST.
	111	IDX1205	EQ	3.00000	.	3.00000	3.00000	127.00000-
	112	IDX2205	EQ	.	.	.	.	127.00000-
	113	IDX1208	EQ	3.00000	.	3.00000	3.00000	295.00000-
	114	IDX2208	EQ	.	.	.	.	295.00000-
	115	IDX1203	EQ	3.00000	.	3.00000	3.00000	7.00000-
	116	IDX2203	EQ	.	.	.	.	7.00000-
	117	X1201	IV	1.00000	540.00000	.	1.00000	468.00000
	118	DX1201	IV	4.00000	60.00000	.	4.00000	12.00000-
	119	X2201	IV	.	180.00000	.	1.00000	.
	120	DX2201	IV	.	60.00000	.	4.00000	.
	121	X1204	EQ	1.00000	.	1.00000	1.00000	120.00000-
A	122	DX1204	IV	3.00000	120.00000	.	3.00000	.
	123	X2204	IV	1.00000	360.00000	.	1.00000	240.00000
	124	DX2204	IV	1.00000	120.00000	.	4.00000	.
	125	X1206	EQ	1.00000	.	1.00000	1.00000	275.00000-
A	126	DX1206	IV	2.00000	275.00000	.	2.00000	.
	127	X2206	IV	1.00000	825.00000	.	1.00000	550.00000
	128	DX2206	IV	2.00000	275.00000	.	4.00000	.
	129	Y1206	IV	.	960.00000	.	1.00000	.
	130	DY1206	IV	.	275.00000	.	2.00000	76.11111
	131	X1209	IV	1.00000	300.00000	.	1.00000	240.00000
	132	DX1209	IV	2.00000	60.00000	.	4.00000	.
	133	X2209	IV	.	180.00000	.	1.00000	120.00000
	134	DX2209	IV	.	60.00000	.	4.00000	.
A	135	X1207	EQ	1.00000	.	1.00000	1.00000	.
	136	DX1207	IV	.	210.00000	.	2.00000	210.00000
	137	X2207	IV	.	630.00000	.	1.00000	.
	138	DX2207	IV	.	210.00000	.	2.00000	280.00000
	139	X1210	EQ	1.00000	.	1.00000	1.00000	335.00000-
	140	DX1210	IV	.	385.00000	.	4.00000	50.00000
	141	X2210	IV	.	1155.00000	.	1.00000	.
	142	DX2210	IV	.	385.00000	.	2.00000	141.11111
	143	X1202	EQ	2.00000	.	2.00000	2.00000	65.00000-
	144	DX1202	IV	.	65.00000	.	8.00000	.
	145	X2202	IV	.	195.00000	.	1.00000	.
	146	DX2202	IV	.	65.00000	.	4.00000	.
	147	Y1202	IV	.	195.00000	.	1.00000	.
	148	DY1202	EQ	.	65.00000	.	.	195.00000-
	149	X1205	EQ	1.00000	.	1.00000	1.00000	127.00000-
	150	DX1205	IV	.	385.00000	.	1.00000	258.00000
	151	X2205	IV	.	1155.00000	.	1.00000	.
	152	DX2205	IV	.	385.00000	.	4.00000	372.22222
	153	X1208	EQ	1.00000	.	1.00000	1.00000	295.00000-
	154	DX1208	IV	.	295.00000	.	1.00000	.
	155	X2208	IV	.	885.00000	.	1.00000	.
	156	DX2208	IV	.	295.00000	.	4.00000	65.55556
	157	X1203	EQ	1.00000	.	1.00000	1.00000	7.00000-
	158	DX1203	IV	.	130.00000	.	1.00000	123.00000
	159	X2203	IV	.	390.00000	.	1.00000	.
	160	DX2203	IV	.	130.00000	.	4.00000	165.55556

Table 50 : Second integer solution (columns section) (cont'd)

```
                EXECUTOR.  MPSX  RELEASE 1

MIXFLOW - TIME =    0.12

 ------------------------------------------------------------------------------------------
 I NODES POSTPONED IF FUNCTIONAL BEYOND 3965.00                                           I
 I NODES DROPPED   IF FUNCTIONAL BEYOND 3965.00                                           I
 I RESTORE NODE     9 - FUNCTIONAL =      3797.0000  ESTIMATION = 3797.00                 I
 I SECOND BRANCH - VARIABLE   131 =    .3000    FORCED TOWARDS LOWER ACTIVITIES           I
 ------------------------------------------------------------------------------------------
 BRANCH DROPPED

 ------------------------------------------------------------------------------------------
 I ------- NODE     3 - FUNCTIONAL =      3431.0000  ESTIMATION = 4069.65                 I
 I ------- NODE     5 - FUNCTIONAL =      3880.0000  ESTIMATION = 4311.99                 I
 I RESTORE NODE     3 - FUNCTIONAL =      3431.0000  ESTIMATION = 4069.65                 I
 I FIRST  BRANCH - VARIABLE   117 =    .1667    FORCED TOWARDS LOWER ACTIVITIES           I
 ------------------------------------------------------------------------------------------

 INVERT CALLED     TIME   0.13   CURRENT INVERSE ---- ETA-VECTORS ...102   ELEMENTS ...509   RECORDS .....2   ITERATION ....113
  BASIS ---- NO.OF ROWS ....61   LOGICALS ........21   STRUCTURALS ....40   ELEMENTS ...167
  INVERSE -- NUCLEUS .......14   TRANSFORMED .....7   ETA-VECTORS ....74   ELEMENTS ...182   RECORDS .....1   TIME TAKEN     0.001

MIXFLOW - TIME =    0.13
 BRANCH DROPPED

 ------------------------------------------------------------------------------------------
 I RESTORE NODE     3 - FUNCTIONAL =      3431.0000  ESTIMATION = 4069.65                 I
 I SECOND BRANCH - VARIABLE   117 =    .1667    FORCED TOWARDS UPPER ACTIVITIES           I
 ------------------------------------------------------------------------------------------

 ------------------------------------------------------------------------------------------
 I REACH   NODE    11 - FUNCTIONAL =      3791.0000  ESTIMATION = 4069.65      ITER .....116   I
 I BEST POSSIBLE SOLUTION IS NOW        3791.00                                           I
 I FIRST  BRANCH - VARIABLE   123 =    .2000    FORCED TOWARDS LOWER ACTIVITIES           I
 ------------------------------------------------------------------------------------------
 BRANCH DROPPED

 ------------------------------------------------------------------------------------------
 I RESTORE NODE    11 - FUNCTIONAL =      3791.0000  ESTIMATION = 4069.65                 I
 I SECOND BRANCH - VARIABLE   123 =    .2000    FORCED TOWARDS UPPER ACTIVITIES           I
 ------------------------------------------------------------------------------------------
 BRANCH DROPPED

 ------------------------------------------------------------------------------------------
 I BEST POSSIBLE SOLUTION IS NOW        3880.00                                           I
 I RESTORE NODE     5 - FUNCTIONAL =      3880.0000  ESTIMATION = 4311.99                 I
 I FIRST  BRANCH - VARIABLE   159 =    .4000    FORCED TOWARDS UPPER ACTIVITIES           I
 ------------------------------------------------------------------------------------------

 INVERT CALLED     TIME   0.14   CURRENT INVERSE ---- ETA-VECTORS ....77   ELEMENTS ...196   RECORDS .....1   ITERATION ....118
  BASIS ---- NO.OF ROWS ....61   LOGICALS ........24   STRUCTURALS ....37   ELEMENTS ...156
  INVERSE -- NUCLEUS .......10   TRANSFORMED .....5   ETA-VECTORS ....69   ELEMENTS ...165   RECORDS .....1   TIME TAKEN     0.001

MIXFLOW - TIME =    0.14
 BRANCH DROPPED
```

Table 51 : Search for third integer solution

```
                EXECUTOR.  MPSX  RELEASE 1

-------------------------------------------------------------------------------
I RESTORE NODE     5 - FUNCTIONAL =     3880.0000  ESTIMATION = 4311.99          I
I SECOND BRANCH - VARIABLE   159 =    .4000     FORCED TOWARDS LOWER ACTIVITIES  I
-------------------------------------------------------------------------------
BRANCH DROPPED

-------------------------------------------------------------------------------
I BEST POSSIBLE SOLUTION IS NOW        3965.00                                 I
I RESTORE NODE    10 - FUNCTIONAL =     3965.0000  ESTIMATION = 3965.00        I
I  O P T I M A L  INTEGER SOLUTION                                             I
-------------------------------------------------------------------------------

MIXSAVE - TIME =   0.14

  NODE SAVED          FUNCTIONAL VALUE          ESTIMATION

         8                4265.0000              INTEGER
        10                3965.0000              INTEGER

MIXSTATS - TIME =   0.14
```

Table 52 : Optimal Integer solution found

EXECUTOR. MPSX RELEASE 1

INTEGER NODES

NODE	8	10
FUNCTIONAL	4265.0000	3965.0000
ESTIMATION	INTEGER	INTEGER
117= X1201	1.0000	1.0000
118= DX1201	4.0000	4.0000
119= X2201	.	.
120= DX2201	.	.
122= DX1204	3.0000	3.0000
123= X2204	1.0000	1.0000
124= DX2204	1.0000	1.0000
126= DX1206	2.0000	2.0000
127= X2206	1.0000	1.0000
128= DX2206	2.0000	2.0000
129= Y1206	.	.
130= DY1206	.	.
131= X1209	1.0000	1.0000
132= DX1209	4.0000	2.0000
133= X2209	1.0000	.
134= DX2209	.	.
136= DX1207	.	.
137= X2207	.	.
138= DX2207	.	.
140= DX1210	.	.
141= X2210	.	.
142= DX2210	.	.
144= DX1202	.	.
145= X2202	.	.
146= DX2202	.	.
147= Y1202	.	.
150= DX1205	.	.
151= X2205	.	.
152= DX2205	.	.
154= DX1208	.	.
155= X2208	.	.
156= DX2208	.	.
158= DX1203	.	.
159= X2203	.	.
160= DX2203	.	.

Table 53 : Summary of two integer solutions obtained

MPSX - MIXED INTEGER PROGRAM - RELEASE 1

PROBLEM NAME = PROB

.....PROBLEM STATISTICS.....		...COMPUTATIONAL ELEMENTS....		.........ENVIRONMENT..........		
ROWS	61	FUNCTIONAL (MIN)	COST1	C.P.U		
COLUMNS	99	RESTRAINTS	RHS1	CORE ALLOCATED		43560
VARIABLES	160	BOUNDS	BND1			
INTEGER VARIABLES	35			SCRATCH	ON ..... DISK	2314
ELEMENTS	401			MATRIX	ON ..... DISK	2314
DENSITY	4.10			ETA	ON ..... DISK	2314
				MIXWORK	ON ..... DISK	2314
				PROBFILE	ON ..... DISK	2314

	TIME SINCE MIXSTART	ITERATION NO. SINCE SETUP	NODE NO.	FUNCTIONAL VALUE	STRATEGY
CONTINUOUS OPTIMUM		84	1	2836.0000	
FIRST INTEGER SOLUTION	0.02	109	8	4265.0000	
OPTIMAL INTEGER SOLUTION	0.04	111	10	3965.0000	
OPTIMALITY PROVED	0.06	120	11		
TIME OF SEARCH	0.06	120	11		

NUMBER OF INTEGER VARIABLES NOT INTEGER AT CONTINUOUS OPTIMUM = 11

NUMBER OF INTEGER SOLUTIONS FOUND = 2

BRANCHES ABANDONED WHILE COMPUTING = 6

Table 54 : Statistical survey of continuous and integer solutions

EXECUTOR. MPSX RELEASE 1

MIXFIX - TIME = 0.15

INTEGER VARIABLES ARE FIXED AT

VAR.NO	ACTIVITY	VAR.NO	ACTIVITY	VAR.NO	ACTIVITY	VAR.NO	ACTIVITY	VAR.NO	ACTIVITY	VAR.NO	ACTIVITY
117	1	118	4	119	0	120	0	122	3	123	1
124	1	126	2	127	1	128	2	129	0	130	0
131	1	132	2	133	0	134	0	136	0	137	0
138	0	140	0	141	0	142	0	144	0	145	0
146	0	147	0	150	0	151	0	152	0	154	0
155	0	156	0	158	0	159	0	160	0		

EXIT - TIME = 0.15

Table 55 : Integer values for optimal solution

APPENDIX B

HERMES I INTERMEDIATE INDEPTH FLOWCHART

LISTING OF HERMES I

DEMAND PAIRS, DEMAND INCREASES, COST DATA & NETWORK CHARACTERISTICS

CALCULATE MAXIMUM CONTEMPLATED DEMAND INCREASE

OPTION
IPA

TEMPORARY DISC STORAGE

SUBROUTINE BORNE
GENERATE BOUNDS FOR GIVEN DEMAND INCREASE

OPTION
IBORNE

SUBROUTINE SET-UP
CONVERT ADMISSIBLE CHAINS (NODES TO LINKS)

OPTION
IAMCL

HAVE BOUNDS CHANGED

NO

ADMISSIBLE CHAINS

IDENTIFY ADMISSIBLE LINKS

OPTION
ILKDTA
ISTET
ICTGT

YES

CONVERT TO INPUT FORMAT OF TRANCHE

TEMPORARY DISC STORAGE

SUBROUTINE DOMINO
IDENTIFY MAXIMUM COST CHAINS AND COST TABLES

OPTION
IBA
IMPR

GENERATE TRANCHE CONTROL PROGRAM

TEMPORARY DISC STORAGE

IDENTIFY ADMISSIBLE NODES

OPTION
INDOM

IDENTIFY ADMISSIBLE CHAINS

OPTION
IADCH
ISTAT

TRANCHE
MIXEDINTEGER LINEAR PROGRAMMING PACKAGE

OPTION
IBCD
IPICT

YES

ANY MORE DEMAND PAIRS

NO

OPTIMUM (LEAST COST) CAPACITY EXPANSION PROGRAM

IDENTIFY COMMON ELEMENTS OF ADMISSIBLE CHAINS AND CALCULATE MAXIMUM RELEVANT DEMAND INCREASE

FIG. 6
HERMES 1 INTERMEDIATE IN DEPTH FLOW CHART

```
//HERMES   JOB (0752,838514,1660),SORES,MSGLEVEL=(1,1),CLASS=D            JOB 717
//CADUCEE  EXEC FORTGCLG,REGION.GO=272K
XX       PROC FORTREG=100K,LKEDREG=134K,GOREG=100K                      00002000
XXFORT   EXEC  PGM=IEYFORT,REGION=&FORTREG                              00004000
IEF653I SUBSTITUTION JCL - PGM=IEYFORT,REGION=100K
XXSYSPRINT  DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=120,BLKSIZE=3480)          00006000
XXSYSLIN    DD DSN=&&LOADSET,UNIT=SYSDA,DISP=(NEW,PASS),                X00008000
XX             SPACE=(800,(40,10),,,ROUND),DCB=(RECFM=FB,LRECL=80,      X00010000
XX             BLKSIZE=800)                                             00012000
//FORT.SYSIN DD *
IEF236I ALLOC. FOR HERMES   FORT     CADUCEE
IEF237I 41E    ALLOCATED TO SYSPRINT
IEF237I 231    ALLOCATED TO SYSLIN
IEF237I 451    ALLOCATED TO SYSIN
IEF285I   SYS72092.T111719.RV000.HERMES.R0000001       DELETED
IEF285I   VOL SER NOS=       .
IEF285I   SYS72092.T111719.RV000.HERMES.LOADSET        PASSED
IEF285I   VOL SER NOS= D14W03.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000002       DELETED
IEF285I   VOL SER NOS=       .
*
ACT100I JOB NAME-HERMES  , STEP NAME-FORT    , CLASS-D, PRIORITY- *,  DATE- 72.092, T.O.D.- 11.18.49
ACT101I STEP TOTALS --     .743 MIN CPU TIME,       425 DASD EXCPS,        0 OTHER EXCPS,     .849 MIN PROCESS TIME
ACT102I STEP RESOURCES --   100K REGION,   100K CORE USED,  0 DASD,  0 TAPES,  1 DASD DATASETS
ACT104I BASIC STEP PROCESSING COST-
*
XXLKED   EXEC  PGM=IEWL,PARM='MAP,LIST,LET',COND=(5,LT,FORT),           X00014000
XX             REGION=&LKEDREG                                          00016000
IEF653I SUBSTITUTION JCL - REGION=134K
XXSYSPRINT  DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=3509)          00018000
XXSYSUT1    DD DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(3072,(40,10),,,ROUND),    X00020000
XX             DCB=BLKSIZE=3072                                         00022000
XXSYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)                          00024000
XX          DD DDNAME=SYSIN                                             00026000
XXSYSLIB    DD DSN=SYS1.FORTLIB,DISP=SHR,UNIT=2314,VOL=SER=MVL620       00028000
XXSYSLMOD   DD DSNAME=&&GOSET(MAIN),UNIT=(SYSDA,SEP=(SYSUT1,SYSLIB)),  X00030000
XX             DISP=(MOD,PASS),SPACE=(3072,(40,10,1),,,ROUND),          X00032000
XX             DCB=BLKSIZE=3072                                         00034000
IEF236I ALLOC. FOR HERMES   LKED     CADUCEE
IEF237I 41B    ALLOCATED TO SYSPRINT
IEF237I 380    ALLOCATED TO SYSUT1
IEF237I 231    ALLOCATED TO SYSLIN
IEF237I 235    ALLOCATED TO SYSLIB
IEF237I 3B1    ALLOCATED TO SYSLMOD
IEF285I   SYS72092.T111719.RV000.HERMES.R0000003       DELETED
IEF285I   VOL SER NOS=       .
IEF285I   SYS72092.T111719.RV000.HERMES.SYSUT1         DELETED
IEF285I   VOL SER NOS= D14W04.
IEF285I   SYS72092.T111719.RV000.HERMES.LOADSET        DELETED
IEF285I   VOL SER NOS= D14W03.
IEF285I   SYS1.FORTLIB                                 KEPT
IEF285I   VOL SER NOS= MVL620.
IEF285I   SYS72092.T111719.RV000.HERMES.GOSET          PASSED
IEF285I   VOL SER NOS= D14W02.
*
ACT100I JOB NAME-HERMES  , STEP NAME-LKED    , CLASS-D, PRIORITY- *,  DATE- 72.092, T.O.D.- 11.19.38
ACT101I STEP TOTALS --     .040 MIN CPU TIME,       673 DASD EXCPS,        0 OTHER EXCPS,     .208 MIN PROCESS TIME
ACT102I STEP RESOURCES --   134K REGION,   132K CORE USED,  0 DASD,  0 TAPES,  4 DASD DATASETS
ACT104I BASIC STEP PROCESSING COST-
```

```
*
XXGO     EXEC  PGM=*.LKED.SYSLMOD,COND=((5,LT,FORT),(5,LT,LKED)),      X00036000
XX             REGION=&GOREG                                           00038000
IEF653I SUBSTITUTION JCL - REGION=100K
XXFT05F001  DD DDNAME=SYSIN                                            00040000
XXFT06F001  DD SYSOUT=A                                                00042000
XXFT07F001  DD DDNAME=SYSPUNCH                                         00044000
//GO.FT01F001 DD *,DCB=(BLKSIZE=80)
//GO.FT03F001 DD SYSOUT=A,DCB=(RECFM=FA,BLKSIZE=133)
//GO.FT04F001 DD DSN=&LKDATA,
//          UNIT=SYSDA,
//       DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//           DISP=(NEW,PASS),SPACE=(CYL,(1,1),RLSE)
//GO.FT02F001 DD DSN=&CHAINS,
//         UNIT=SYSDA,
//        DCB=(RECFM=FB,LRECL=240,BLKSIZE=2400),
//        DISP=(NEW,DELETE),
//         SPACE=(CYL,(1,1),RLSE)
//GO.FT08F001 DD DSN=&MPS2,
//        UNIT=SYSDA,
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
//          DISP=(NEW,PASS),
//          SPACE=(CYL,(1,1),RLSE)
IEF236I ALLOC. FOR HERMES   GO       CADUCEE
IEF237I 381    ALLOCATED TO PGM=*.DD
IEF237I 411    ALLOCATED TO FT06F001
IEF237I 451    ALLOCATED TO FT01F001
IEF237I 412    ALLOCATED TO FT03F001
IEF237I 230    ALLOCATED TO FT04F001
IEF237I 231    ALLOCATED TO FT02F001
IEF237I 384    ALLOCATED TO FT08F001
IEF285I   SYS72092.T111719.RV000.HERMES.GOSET          PASSED
IEF285I   VOL SER NOS= 014W02.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000004       DELETED
IEF285I   VOL SER NOS=       .
IEF285I   SYS72092.T111719.RV000.HERMES.R0000005       DELETED
IEF285I   VOL SER NOS=       .
IEF285I   SYS72092.T111719.RV000.HERMES.R0000006       DELETED
IEF285I   VOL SER NOS=       .
IEF285I   SYS72092.T111719.RV000.HERMES.LKDATA         PASSED
IEF285I   VOL SER NOS= 081003.
IEF285I   SYS72092.T111719.RV000.HERMES.CHAINS         DELETED
IEF285I   VOL SER NOS= 014W03.
IEF285I   SYS72092.T111719.RV000.HERMES.MPS2           PASSED
IEF285I   VOL SER NOS= 014W05.
*
ACT100I JOB NAME-HERMES  , STEP NAME-GO      , CLASS-D, PRIORITY- *,  DATE- 72.092, T.O.D.- 11.23.13
ACT101I STEP TOTALS --     .064 MIN CPU TIME,       168 DASD EXCPS,        0 OTHER EXCPS,     .106 MIN PROCESS TIME
ACT102I STEP RESOURCES --   272K REGION,   270K CORE USED,  0 DASD,  0 TAPES,  4 DASD DATASETS
ACT104I BASIC STEP PROCESSING COST-
*
//MPSX     EXEC MPSXI,EXECREG=250K,MARVEL=DUMMY
XX        PROC  INDEX=PP,VERSION=V1L0,COMP=1,EXEC=1,EXECREG=200K,      X00000010
XX              MARVEL='SPACE=(CYL,(2,1)),UNIT=SYSDA',                 X00000020
XX              MIP='SPACE=(CYL,(15),,CONTIG),UNIT=SYSDA'               00000030
*******  MATHEMATICAL PROGRAMMING SYSTEM EXTENDED - MPSX  (5734-XM4)  **00000040
XXSETCOND EXEC PGM=IEFBR14                                             00000050
*
ACT100I JOB NAME-HERMES  , STEP NAME-SETCOND , CLASS-D, PRIORITY- *,  DATE- 72.092, T.O.D.- 11.23.20
ACT101I STEP TOTALS --     .000 MIN CPU TIME,         2 DASD EXCPS,        0 OTHER EXCPS,     .001 MIN PROCESS TIME
```

```
ACT102I STEP RESOURCES --    60K REGION,     4K CORE USED,  0 DASD,  0 TAPES,  0 DASD DATASETS
ACT104I BASIC STEP PROCESSING COST-
*
XX*PS1   EXEC  PGM=DJLCOMP,REGION=100K,COND=((&COMP,EQ,SETCOND)                    00000060
IEF653I SUBSTITUTION JCL - PGM=DJLCOMP,REGION=100K,COND=((1,EQ,SETCOND)
XXSTEPLIB  DD  DSNAME=&INDEX..MPSX.LIBRARY.&VERSION,DISP=SHR                       00000070
IEF653I SUBSTITUTION JCL - DSNAME=PP.MPSX.LIBRARY.V1LO,DISP=SHR
XXSCRATCH1 DD  SPACE=(CYL,(1,1)),UNIT=SYSDA                                        00000080
XXSCRATCH2 DD  SPACE=(CYL,(1,1)),UNIT=SYSDA                                        00000090
XXSCRATCH3 DD  SPACE=(CYL,(1,1)),UNIT=SYSDA                                        00000100
XXSCRATCH4 DD  SPACE=(CYL,(1,1)),UNIT=SYSDA                                        00000110
XXSYSMLCP  DD  SPACE=(CYL,(1,1,2)),UNIT=SYSDA,DISP=(,PASS)                         00000120
XXSYSPRINT DD  SYSOUT=A                                                            00000130
XXSYSUDUMP DD  SYSOUT=A                                                            00000140
//*PS1.SYSIN DD DSN=&LKDATA,DISP=(OLD,DELETE)
IEF653I SUBSTITUTION JCL - DSN=&LKDATA,DISP=(OLD,DELETE)
IEF236I ALLOC. FOR HERMES   MPS1     MPSX
IEF237I 280   ALLOCATED TO STEPLIB
IEF237I 231   ALLOCATED TO SCRATCH1
IEF237I 380   ALLOCATED TO SCRATCH2
IEF237I 232   ALLOCATED TO SCRATCH3
IEF237I 381   ALLOCATED TO SCRATCH4
IEF237I 230   ALLOCATED TO SYSMLCP
IEF237I 413   ALLOCATED TO SYSPRINT
IEF237I 417   ALLOCATED TO SYSUDUMP
IEF237I 230   ALLOCATED TO SYSIN
IEF285I   PP.MPSX.LIBRARY.V1LO                          KEPT
IEF285I   VOL SER NOS= PPXSC1.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000007        DELETED
IEF285I   VOL SER NOS= D14W03.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000008        DELETED
IEF285I   VOL SER NOS= D14W04.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000009        DELETED
IEF285I   VOL SER NOS= PCSES1.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000010        DELETED
IEF285I   VOL SER NOS= D14W02.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000011        PASSED
IEF285I   VOL SER NOS= DB1003.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000012        DELETED
IEF285I   VOL SER NOS=      .
IEF285I   SYS72092.T111719.RV000.HERMES.LKDATA          DELETED
IEF285I   VOL SER NOS= DB1003.
*
ACT100I JOB NAME-HERMES  , STEP NAME-MPS1    , CLASS-D, PRIORITY- *,  DATE- 72.092, T.O.D.- 11.24.04
ACT101I STEP TOTALS --     .069 MIN CPU TIME,      526 DASD EXCPS,        0 OTHER EXCPS,     .200 MIN PROCESS TIME
ACT102I STEP RESOURCES --   100K REGION,    82K CORE USED,  0 DASD,  0 TAPES,  7 DASD DATASETS
ACT104I BASIC STEP PROCESSING COST-
*
XX*PS2   EXEC  PGM=DJLEXEC,PARM=TASK,REGION=&EXECREG,                             X00000150
IEF653I SUBSTITUTION JCL - PGM=DJLEXEC,PARM=TASK,REGION=250K,
XX             COND=((&EXEC,EQ,SETCOND),(0,NE,MPS1))                              00000160
IEF653I SUBSTITUTION JCL - COND=((1,EQ,SETCOND),(0,NE,MPS1))
XXSTEPLIB  DD  DSNAME=&INDEX..MPSX.LIBRARY.&VERSION,DISP=SHR                       00000170
IEF653I SUBSTITUTION JCL - DSNAME=PP.MPSX.LIBRARY.V1LO,DISP=SHR
XXETA1     DD  SPACE=(CYL,(5),,CONTIG),UNIT=SYSDA                                  00000180
XXETA2     DD  SPACE=(CYL,(5),,CONTIG),UNIT=(SYSDA,SEP=ETA1)                       00000190
XX*ATRIX1  DD  SPACE=(CYL,(5),,CONTIG),UNIT=SYSDA                                  00000200
XXMATRIX2  DD  SPACE=(CYL,(5),,CONTIG),UNIT=(SYSDA,SEP=MATRIX1)                    00000210
XX*IXWORK  DD  &MIP                                                                00000220
IEF653I SUBSTITUTION JCL - SPACE=(CYL,(15),,CONTIG),UNIT=SYSDA
```

```
XX*MPSCRAT  DD  SPACE=(CYL,(5),,CONTIG),UNIT=SYSDA                    00000230
XX*VELOBJ   DD  &MARVEL                                               00000240
IEF653I SUBSTITUTION JCL - DUMMY
XXPROBFILE DD  SPACE=(CYL,(5),,CONTIG),UNIT=SYSDA                     00000250
XXSCRATCH1 DD  SPACE=(CYL,(5),,CONTIG),UNIT=SYSDA                     00000260
XXSCRATCH2 DD  SPACE=(CYL,(5),,CONTIG),UNIT=(SYSDA,SEP=SCRATCH1)      00000270
XXSYSMLCP  DD  DSNAME=*.MPS1.SYSMLCP,DISP=(OLD,DELETE)                00000280
XXSYSPRINT DD  SYSOUT=A                                               00000290
XXSYSUDUMP DD  SYSOUT=A                                               00000300
//MPS2.SYSIN DD DSN=&MPS2,DISP=(OLD,DELETE)
//
IEF236I ALLOC. FOR HERMES   MPS2     MPSX
IEF237I 280   ALLOCATED TO STEPLIB
IEF237I 230   ALLOCATED TO ETA1
IEF237I 231   ALLOCATED TO ETA2
IEF237I 232   ALLOCATED TO MATRIX1
IEF237I 230   ALLOCATED TO MATRIX2
IEF237I 380   ALLOCATED TO MIXWORK
IEF237I 231   ALLOCATED TO MPSCRAT
IEF237I 380   ALLOCATED TO PROBFILE
IEF237I 232   ALLOCATED TO SCRATCH1
IEF237I 380   ALLOCATED TO SCRATCH2
IEF237I 230   ALLOCATED TO SYSMLCP
IEF237I 412   ALLOCATED TO SYSPRINT
IEF237I 413   ALLOCATED TO SYSUDUMP
IEF237I 384   ALLOCATED TO SYSIN
IEF285I   PP.MPSX.LIBRARY.V1L0                        KEPT
IEF285I   VOL SER NOS= PPXSC1.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000014      DELETED
IEF285I   VOL SER NOS= DB1003.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000015      DELETED
IEF285I   VOL SER NOS= D14W03.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000016      DELETED
IEF285I   VOL SER NOS= PCSES1.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000017      DELETED
IEF285I   VOL SER NOS= DB1003.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000018      DELETED
IEF285I   VOL SER NOS= D14W04.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000019      DELETED
IEF285I   VOL SER NOS= D14W03.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000020      DELETED
IEF285I   VOL SER NOS= D14W04.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000021      DELETED
IEF285I   VOL SER NOS= PCSES1.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000022      DELETED
IEF285I   VOL SER NOS= D14W04.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000011      DELETED
IEF285I   VOL SER NOS= DB1003.
IEF285I   SYS72092.T111719.RV000.HERMES.R0000023      DELETED
IEF285I   VOL SER NOS=      .
IEF285I   SYS72092.T111719.RV000.HERMES.MPS2          DELETED
IEF285I   VOL SER NOS= D14W05.
*
ACT100I JOB NAME-HERMES  , STEP NAME-MPS2    , CLASS-D, PRIORITY- *,  DATE- 72.092, T.O.D.- 11.30.01
ACT101I STEP TOTALS --     .157 MIN CPU TIME,       496 DASD EXCPS,         0 OTHER EXCPS,     .281 MIN PROCESS TIME
ACT102I STEP RESOURCES --   250K REGION,   250K CORE USED,  0 DASD,  0 TAPES, 12 DASD DATASETS
ACT104I BASIC STEP PROCESSING COST-
*
IEF285I   SYS72092.T111719.RV000.HERMES.GOSET          DELETED
IEF285I   VOL SER NOS= D14W02.
```

```
*
ACT100I JOB NAME-HERMES   , STEP NAME-JOB TERM, CLASS-D, PRIORITY- *,  DATE- 72.092, T.O.D.- 11.30.02
ACT101I JOB  TOTALS --     1.072 MIN CPU TIME,      2,290 DASD EXCPS,          0 OTHER EXCPS,     1.645 MIN PROCESS TIME
ACT103I JOB VOLUME MOUNTS --  0 DASD,  0 TAPES
ACT104I BASIC JOB  PROCESSING COST-
```

MAIN

```
      INTEGER*2 MCH1(1500,20),MCH2(1500,20)
      INTEGER*2 NOMINE(100),JAR(145,6),MCOST(100,100),MA(100,100)
      DIMENSION NOMIN(145,4),NPOINT(100),MDT(145),MRDA(145)
      DIMENSION           NST(100)
      INTEGER  CST1,CST2,CST3,CST4
      DIMENSION                  NAD(9,6,7),NSO(100,3)
      COMMON /BLB/ MA,MNUM(100)
      COMMON /DOM/ NOMINE,JAR,MCOST
      COMMON/POP/CST1(4),CST2(4),CST3(4),CST4(4)
      COMMON/DAD/NDEM(12),NARR(7),NDEP(7),NCOUP
      COMMON/MOM/MCH1,MCH2
C
C     LECTURE DES PARAMETRES
C
      READ(1,9)NCOUP,IPA,IBORNE,IBA,MPR,INDOM,IADCH,ISTAT,IAMCL,
     1ISTET,ICTGT,  ILKDTA,IBCD,IPICT
    9 FORMAT(16I5)
      IF(IPA.EQ.0)GO TO 715
      WRITE(3,11) NCOUP
   11 FORMAT(10X,'NOMBRE DE PAIRES DE SOMMETS = ',I3///)
C
C     LECTURE DES PAIRES DE SOMMETS
C
      WRITE(3,713)
  713 FORMAT(10X,'PAIRES DE SOMMETS :')
  715 NN=0
      DO 404 I=1,NCOUP
      READ(1,505) (NDEP(I),NARR(I),NDEM(I))
      IF(IPA.EQ.0) GO TO 716
      WRITE(3,505) (NDEP(I),NARR(I),NDEM(I))
  505 FORMAT(3I5)
  716 NN=NN+NDEM(I)
  404 CONTINUE
      DO 820 I=1,145
      DO 820 J=1,6
  820 JAR(I,J)=0
C
C     CALCUL DES COUTS (BORNES SUPERIEURES ET INFERIEURES)
C
      DO 664 IQ=1,9
      DO 664 IR=1,6
      DO 664 IS=1,7
  664 NAD(IQ,IR,IS)=99
      DO 620 I=1,100
      DO 620 J=1,100
  620 MA(I,J)=0
      DO 316 IZ=1,100
  316 MNUM(IZ)=0
```

MAIN

```
      NIT=0
      CALL BORNE(NIT,NN,NK,M,JKY)
      NIT=1
      NSOM=NK
      IF(IBORNE.EQ.0)GO TO 3234
      WRITE(3,717)
717   FORMAT('0',6X,'NODE NUMBER',3X,'NEW NODE NUMBER')
3234   CONTINUE
      DO 323 IY=1,NK
      N8=MNUM(IY)
      NSO(IY,1)=N8/100
      NSO(IY,2)=(N8-NSO(IY,1)*100)/10
      NSO(IY,3)=N8-NSO(IY,1)*100-NSO(IY,2)*10
      NAD(NSO(IY,1),NSO(IY,2),NSO(IY,3))=IY
      IF(IBORNE.EQ.0)GO TO 323
      WRITE(3,714)(NSO(IY,IH),IH=1,3),IY
714   FORMAT(10X,3I1,9X,I6)
  323 CONTINUE
      DO 683 IG=1,NSOM
      DO 683 IH=1,NSOM
  683 IF(MA(IG,IH).EQ.0) MA(IG,IH)=MA(IH,IG)
   21 FORMAT(////,10X,16HMATRICE ASSOCIEE,///)
  711 FORMAT(10X,8211/)
  697 NTO=0
C
C     CALCULER DIFFERENTES PAIRES DE SOMMETS
C
      DO 50 ITER=1,NCOUP
      REWIND 2
      NZZ=0
      JJJ=1
      LXZ=5
      IF(IBORNE.EQ.0) GO TO 8001
      GO TO 8007
8001  IF(IBA.EQ.0) GO TO 8002
      GO TO 8007
8002  IF(IBB.EQ.0) GO TO 8003
      GO TO 8007
8003  IF(IMPR.EQ.0) GO TO 8004
      GO TO 8007
8004  IF(INDOM.EQ.0) GO TO 8005
      GO TO 8007
8005  IF(IDACH.EQ.0) GO TO 8006
      GO TO 8007
8006  IF(ISTAT.EQ.0) GO TO 1853
8007  CONTINUE
  855 WRITE(3,851) ITER
851   FORMAT(1H1,/2X,'ITERATION NO =',I2)
```

MAIN

```
      IF(IBORNE.EQ.0) GO TO 1853
      WRITE(3,1849)
1849  FORMAT('0',2X,'NO D ARETE',10X,'SOMMETS CONCERNES',15X,'COUT B.S.'
     1,10X,'COUT B.I.',15X,'M.R.D.'//)
1853  CONTINUE
      DO 853 I=JJJ,M
      LXZ=LXZ+1
      IF (LXZ.GT.28) GO TO 854
      IF(IBORNE.EQ.0)GO TO 853
      WRITE(3,852) JAR(I,6),(JAR(I,J),J=1,5)
  852 FORMAT(6X,I3,16X,I3,4X,I3,20X,I6,12X,I6,18X,I3/)
  853 CONTINUE
      GO TO 857
  854 JJJ=I
      LXZ=5
      GO TO 855
  857 DO 856 III=1,M
      MRDA(III)=JAR(III,5)
  856 JAR(III,5)=0
      DO 30 NGT=1,NCOUP
      NZZ=NZZ+1
      NTO=0
      DO 23 I=1,NSOM
      NST(I)=0
   23 CONTINUE
      DO 821 I=1,145
  821 MDT(I)=0
      DO 630 IXY=1,NSOM
      IF(NDEP(NGT).NE.MNUM(IXY)) GO TO 630
      KDEP=IXY
      GO TO 631
  630 CONTINUE
  631 DO 635 IXY=1,NSOM
      IF(NARR(NGT).NE.MNUM(IXY)) GO TO 635
      KARR=IXY
      GO TO 636
  635 CONTINUE
C
C     CONSTRUIRE  LES 4 TABLEAUX DES COUTS (MINIMUM)
C
  636 IB=3
      NC=1
      DO 301 JX=1,4
  721 NOPT=MOD(JX,2)
      IF(NOPT.EQ.0) GO TO 497
      NORG=KDEP
      NDEST=KARR
      GO TO 496
```

MAIN

```
  497 NORG=KARR
      NDEST=KDEP
  496 IF(JX.GT.2) IB=4
C
C     LIRE LES COUTS DES ARETES (B. INFERIEURES)
C
C
C     APPELER LA SOUS-ROUTINE POUR CALCULER LES COUTS MINIMUM
C
499   CALL DOMINO(NORG,NDEST,M,NSOM,IB,NC,IBA)
C
C     STOCKER LES COUTS MINIMUM
C
      DO 306 IX=1,NSOM
  306 NOMIN(IX,JX )=NOMINE(IX)
  301 CONTINUE
      IB=3
      IF (NZZ.GT.1) GO TO 421
      CALL DOMINO(NORG,NDEST,M,NSOM,IB,NC,IBA)
C
C     IMPRIMER LES 4 TABLEAUX DES COUTS MINIMUM (OPTION)
C
421   IF(IMPR.EQ.0)GO TO 699
      JJJ=1
  423 LXZ=4
      WRITE(3,307)
  307 FORMAT(1H1,10X,'COUTS MINIMUM'//10X,'SOMMET',14X,'BORNE SUPERIEURE
     1', 3X,'BORNE SUPERIEURE',3X,'BORNE INFERIEURE',3X,'BORNE INFERIEUR
     2E',3X,/10X,'NUMERO',14X,'DE NORG ',12X,'DE NDEST ',11X,'DE NORG ',
     312X,'DE NDEST'//)
      DO 308 IX=JJJ,NSOM
      LXZ=LXZ+1
      IF (LXZ.GT.49) GO TO 422
      WRITE(3,309)IX,(NOMIN(IX,J),J=1,4)
  309 FORMAT(12X,I3,5X,4(10X,I10))
  308 CONTINUE
      GO TO 699
  422 JJJ=IX
      GO TO 423
C
C      ELIMINATION DES SOMMETS DOMINES
C
  699 NPT=0
      DO 311 I=1,NSOM
      IF((NOMIN(I,3)+NOMIN(I,4)).GT.NOMIN(KARR,1)) GO TO 311
      NPT=NPT+1
      NPOINT(NPT)=I
  311 CONTINUE
```

MAIN

```
C
C       IMPRIMER LES NUMEROS DES SOMMETS NON DOMINES
C
        IF(INDOM.EQ.0) GO TO 9748
        WRITE(3,799)
        WRITE(3,701)(NPOINT(I),I=1,NPT)
  701   FORMAT(20I4)
  799   FORMAT(1H1,10X,'SOMMETS NON DOMINES'//)
 9748   NS=NSOM-1
        KD=KDEP+1
        KLON=0
        IRES=1
        MCH1(1,1)=KDEP
        LXZ=4
        IF(IADCH.EQ.0) GO TO 110
        WRITE(3,1001)
 1001   FORMAT(1H1,10X,'CHAINES ADMISSIBLES',///)
  110   KLON=KLON+1
        L=0
C
C       CONSTRUIRE LES CHAINES
C
        K=0
        JK=KLON+1
        ICOMB=KLON-1
        DO 140 I=1,IRES
        N1=MCH1(I,KLON)
        DO 150 JS=1,NPT
        J=NPOINT(JS)
        IF(MA(N1,J).NE.1) GO TO 150
        DO 160 J1=1,KLON
        IF(J.EQ.MCH1(I,J1)) GO TO 150
        IF(J1.LT.KLON) GO TO 160
C
C       VERIFICATION POUR SAVOIR SI LA CHAINE EST ADMISSIBLE AU SENS
C       DU COUT MINIMUM
C
        MCH1(I,JK)=J
        NCAP=0
        DO 312 IX=2,JK
        DO 313 JX=1,M
        IF(JAR(JX,1).NE.MCH1(I,IX-1)) GO TO 313
        IF(JAR(JX,2).NE.MCH1(I,IX)) GO TO 313
        NCAP=NCAP+JAR(JX,4)
        GO TO 312
  313   CONTINUE
        DO 315 JX=1,M
        IF(JAR(JX,1).NE.MCH1(I,IX)) GO TO 315
```

```
      IF(JAR(JX,2).NE.MCH1(I,IX-1)) GO TO 315
      NCAP=NCAP+JAR(JX,4)
      GO TO 312
  315 CONTINUE
  312 CONTINUE
C
C        TEST
C
      IF(NCAP-NOMIN(J,1))599,599,150
C      REMISER LA CHAINE ADMISSIBLE
C
  599 K=K+1
      DO 153 K1=1,JK
      IF(K1.EQ.JK) GO TO 154
      MCH2(K,K1)=MCH1(I,K1)
      GO TO 153
  154 MCH2(K,K1)=J
  153 CONTINUE
  160 CONTINUE
  150 CONTINUE
  140 CONTINUE
      IF(K.EQ.0) GO TO 80
      DO 151 I=1,K
      DO 151 J=1,JK
      MCH1(I,J)=MCH2(I,J)
  151 CONTINUE
      L3=0
C
C     IMPRIMER LES CHAINES TERMINEES; ELIMINER LES CHAINES NON TERMINEES
C     DONT LE DERNIER SOMMET EST UN SOMMET PENDANT
C
      DO 180 I=1,K
      IF(MCH1(I,JK).NE.KARR) GO TO 190
      DO 800 J=2,JK
      DO 801 JX=1,M
      IF(JAR(JX,1).NE.MCH1(I,J-1)) GO TO 802
      IF(JAR(JX,2).NE.MCH1(I,J)) GO TO 802
      MDT(JX)=1
      GO TO 800
  802 IF(JAR(JX,1).NE.MCH1(I,J)) GO TO 801
      IF(JAR(JX,2).NE.MCH1(I,J-1)) GO TO 801
      MDT(JX)=1
      GO TO 800
  801 CONTINUE
  800 CONTINUE
      NTO=NTO+1
      WRITE(2,687)KLON,(NSO(KDEP,ID),ID=1,3),(NSO(MCH1(I,JK),ID),
     1 ID=1,3),NTO,((NSO(MCH1(I,I4),ID),ID=1,3),I4=1,JK)
```

MAIN

```
  688 LXZ=LXZ+1
      IF (LXZ.GT.54) GO TO 689
      IF(IADCH.EQ.0) GO TO 2548
      WRITE(3,687)KLON,(NSO(KDEP,ID),ID=1,3),(NSO(MCH1(I,JK),ID),
     1 ID=1,3),NTO,((NSO(MCH1(I,I4),ID),ID=1,3),I4=1,JK)
  687 FORMAT(I4,2(1X,3I1),I4,16(1X,3I1)/20(1X,3I1)/20(1X,3I1))
 2548 NST(KLON)=NST(KLON)+1
      GO TO 180
  689 WRITE(3,1001)
      LXZ=4
      GO TO 688
  190 NTES=MCH1(I,JK)
      CTR=0
      DO 184 J6=1,NSOM
      CTR=CTR+MA(NTES,J6)
  184 CONTINUE
      IF(CTR.EQ.1) GO TO 180
      L3=L3+1
      DO 183 J2=1,JK
      MCH2(L3,J2)=MCH1(I,J2)
  183 CONTINUE
  180 CONTINUE
C
C     NE GARDER DANS MCH1 QUE LES CHAINES NON TERMINEES
C
      IRES=L3
      DO 181 I=1,L3
      DO 181 J=1,JK
      MCH1(I,J)=MCH2(I,J)
  181 CONTINUE
      IF(KLON.EQ.NS) GO TO 80
      IF(L3.NE.0) GO TO 110
   80 DO 803 JX=1,M
      IF(MDT(JX).GT.0) JAR(JX,5)=JAR(JX,5)+NDEM(NGT)
  803 CONTINUE
C
C     IMPRIMER LES STATISTIQUES
      JJJ=1
  825 LXZ=9
      IF(ISTAT.EQ.0) GO TO 8241
      WRITE(3,200)
  200 FORMAT(1H1,/10X,28HSTATISTIQUES SUR LES CHAINES,//,10X,28H********
     1**********************,////)
      WRITE(3,210) NTO
  210 FORMAT(//,10X,26HNOMBRE TOTAL DE CHAINES = ,I5,//)
 8241 CONTINUE
      DO 822 I=JJJ,KLON
      LXZ=LXZ+1
```

MAIN

```
      IF (LXZ.GT.29) GO TO 823
      IF(ISTAT.EQ.0) GO TO 822
      WRITE(3,220) I,NST(I)
  220 FORMAT(/,10X,4HNST(,I2,4H) = ,I4)
  822 CONTINUE
   30 CONTINUE
      GO TO 824
  823 JJJ=I
      GO TO 825
  824 DO 805 I=1,M
      IF(MRDA(I).EQ.JAR(I,5)) GO TO 805
      GO TO 806
  805 CONTINUE
      GO TO 875
  806 CALL BORNE(NIT,NN,NK,M,JKY)
   50 CONTINUE
875   WRITE(2,1097)
1097  FORMAT('9999')
      ENDFILE2
      CALL SETUP(JKY,IAMCL,ISTET,ICTGT,ILKDTA,IBCD,IPICT)
      CONTINUE
      STOP
      END
```

BORNE

```
      SUBROUTINE BORNE(NIT,NN,NK,IX,JKY)
      INTEGER    LD(12),COST(4)
      INTEGER  CST1,CST2,CST3,CST4
      INTEGER*2 A,O,D,EX
      COMMON/POP/CST1(4),CST2(4),CST3(4),CST4(4)
      INTEGER*2 NOMINE(100),JAR(145,6)
      INTEGER*2 MA(100,100)
      DIMENSION CH(12),ICH(12),ICC(145,12)
      COMMON /BLB/ MA,MNUM(100)
      COMMON /DOM/ NOMINE,JAR
  100 FORMAT(4I5)
  101 FORMAT(   4I3,12I2,I1,4I6)
      IF (NIT.GT.0) GO TO 73
      READ(1,100) CST1
      READ(1,100) CST2
      READ(1,100) CST3
      READ(1,100) CST4
      NK=1
      IX=0
      REWIND 4
      JKY=0
    1 READ(1,101)A,O,D,MI,LD,EX,COST
      IF(O.EQ.999) GO TO 98
      WRITE(4,101)A,O,D,MI,LD,EX,COST
      JKY=JKY+1
      IX=IX+1
      JAR(IX,6)=A
      JAR(IX,5)=NN
      DO 318 IY=1,NK
      IF(MNUM(IY).EQ.O ) GO TO 317
  318 CONTINUE
      MNUM(NK)=O
      JAR(IX,1)=NK
      NK=NK+1
      GO TO 319
  317 JAR(IX,1)=IY
  319 DO 321 IY=1,NK
      IF(MNUM(IY).EQ.D ) GO TO 322
  321 CONTINUE
      MNUM(NK)=D
      JAR(IX,2)=NK
      NK=NK+1
      GO TO 326
  322 JAR(IX,2)=IY
  326 MA(JAR(IX,1),JAR(IX,2))=1
      IF(EX.EQ.4) GO TO 14
      IF(EX.EQ.1) GO TO 10
      IF(EX.EQ.2) GO TO 20
```

BORNE

```
      DO 30 I=1,4
      IF(COST(I).GT.0) GO TO 30
      COST(I) = CST3(I)* MI
   30 CONTINUE
      GO TO 50
   10 DO 11 I=1,4
      IF(COST(I).GT.0) GO TO 11
      COST(I) = CST1(I)* MI
   11 CONTINUE
      GO TO 50
   20 DO 21 I=1,4
      IF(COST(I).GT.0) GO TO 21
      COST(I) = CST2(I)*MI
   21 CONTINUE
      GO TO 50
   14 DO 22 I=1,4
      IF(COST(I).GT.0) GO TO 22
      COST(I)=CST4(I)*MI
   22 CONTINUE
   50 L1 = LD(3) -(LD(1)+LD(2))
      L2 = LD(6) -(LD(4)+LD(5))
      IC = 1
      FACT = 0.
      IF(LD(1).GT.0) GO TO 51
      NBC=LD(3)
      IN=MINO(NN,NBC)
      DO 205 I=1,IN
  205 CH(I)=COST(1)+(I-1)*COST(2)
      IC=IN+1
      IF(NN.GT.LD(3)) GO TO 53
      GO TO 70
   51 IF(L1.EQ.0)GO TO 53
      IN=MINO(NN,L1)
      DO 52 I=1,IN
      IC = I
      CH(I)=  COST(2)*I
   52  CONTINUE
      IF(L1.GE.NN) GO TO 70
      IC = IC + 1
   53  IF(LD(4).GT.0) GO TO 55
   54  IF(IC.GT.1) FACT = CH(IC-1)
      CH(IC)=(3*COST(2)+FACT)
      IC=IC+1
      IF(IC.GT.NN) GO TO 70
   55 L3=MINO((NN-IC+1),L2)
      IF (L3.EQ.0) GO TO 118
   56 DO 57 I = 1,L3
      IF(IC.GT.1) FACT = CH(IC-1)
```

BORNE

```
      CH(IC)=(COST(2)+ FACT)
      IC = IC + 1
   57  CONTINUE
  118 IC=IC-1
      IF(IC.GE.NN) GO TO 70
      FACT = CH(IC)
      IC = IC + 1
      CH(IC) =(COST(3)+FACT)
      L4=MINO((NN-IC),4)
      IF(L4.EQ.0) GO TO 70
      DO 58 I= 1,L4
      FACT = CH(IC)
      IC = IC + 1
      CH(IC) = (COST(4)+ FACT)
   58  CONTINUE
      IF(IC.GE.NN) GO TO 70
      IC=IC+1
      CH(IC)=(3*COST(4)+CH(IC-1))
      L4 = NN-IC
      IF(L4.EQ.0) GO TO 70
      DO 59 I=1,L4
      IC=IC+1
      CH(IC)=(COST(4)+CH(IC-1))
   59 CONTINUE
   70 NS=NN-1
      DO 75 I=1,NS
      CH(NN-I+1)=CH(NN-I+1)-CH(NN-I)
   75 CONTINUE
      DO 72 I= 1,NN
   72 ICH(I)= CH(I) + .9
      DO 200 I=1,NN
  200 ICC(IX,I)=ICH(I)
      GO TO 1
98    CONTINUE
      ENDFILE4
      NK=NK-1
      MRD=NN +1
   73 DO 96 IM=1,IX
      IF(NIT.GT.0) MRD=JAR(IM,5) +1
      GO TO (81,82,83,84,85,86,87,88,89,90,91,92,93),MRD
   81 IUB=32700
      ILB=32700
      GO TO 97
   82 IUB=ICC(IM,1)
      ILB=ICC(IM,1)
      GO TO 97
   83 IUB=MAXO(ICC(IM,1),ICC(IM,2))
      ILB=MINO(ICC(IM,1),ICC(IM,2))
```

```
      GO TO 97
   84 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3))
      GO TO 97
   85 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4))
      GO TO 97
   86 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5))
      GO TO 97
   87 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6))
      GO TO 97
   88 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7))
      GO TO 97
   89 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8))
      GO TO 97
   90 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9))
      GO TO 97
   91 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9),ICC(IM,10))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9),ICC(IM,10))
      GO TO 97
   92 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9),ICC(IM,10),ICC(IM,11))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9),ICC(IM,10),ICC(IM,11))
      GO TO 97
   93 IUB=MAXO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9),ICC(IM,10),ICC(IM,11),
     2ICC(IM,12))
      ILB=MINO(ICC(IM,1),ICC(IM,2),ICC(IM,3),ICC(IM,4),ICC(IM,5),
     1ICC(IM,6),ICC(IM,7),ICC(IM,8),ICC(IM,9),ICC(IM,10),ICC(IM,11),
     2ICC(IM,12))
   97 JAR(IM,3)=IUB
      JAR(IM,4)=ILB
```

```
96 CONTINUE
99 RETURN
   END
```

DOMINO

```
      SUBROUTINE DOMINO(NORG,NDEST,M,N,NTC,NC,IBA)
      INTEGER*2 IV(14),MCOST(100,100)
      INTEGER*2 JAR(145,6),NOME(100),NOMI(100),IT(145),IOPT(30,3)
      COMMON /DOM/ NOME,JAR,MCOST
C
C
  101 FORMAT(5I5)
  102 FORMAT(2I3,2I6)
  103 FORMAT(1H1,//,20X,'VALEUR  MINIMUM  A PARTIR DU SOMMET',4X,I6,/,
     1 80X,'(NOMBRE D''ITERATIONS:',6X,I6,')',/,21X,'DEST.',7X,'VALEUR B
     2I',/)
  104 FORMAT(20X,I6,2X,I10)
  105 FORMAT(1H1,//,20X,'CHAINE    OPTIMALE     (PAS NECESSAIREMENT UNIQ
     1UE)',//20X,'ORIG.  DEST.  VALEUR BI',/)
  106 FORMAT(18X,I6,1X,I6,1X,I8)
  107 FORMAT(//,18X,'VALEUR TOTALE',1X,I8)
  108 FORMAT(1H1,20X,'COUT DE LA CHAINE DE COUT MINIMUM ENTRE TOUS LES C
     1OUPLES DE SOMMETS (I,J).',/,41X,'LES COUTS SONT A BORNES SUPERIEUR
     2ES',//,5X,14('   (',I4,')'),//)
  109 FORMAT(1X,I4,14I9)
  110 FORMAT(1H1,20X,'COUT DE LA CHAINE DE COUT MINIMUM ENTRE TOUS LES C
     1OUPLES DE SOMMETS (I,J).',/,41X,'LES COUTS SONT A BORNES INFERIEUR
     2ES',//,5X,14('   (',I4,')'),//)
  111 FORMAT(1H1,//,20X,'VALEUR  MINIMUM  A PARTIR DU SOMMET',4X,I6,/,
     1 80X,'(NOMBRE D''ITERATIONS:',6X,I6,')',/,21X,'DEST.',7X,'VALEUR B
     2S',/)
  112 FORMAT(1H1,//,20X,'CHAINE    OPTIMALE     (PAS NECESSAIREMENT UNIQ
     1UE)',//20X,'ORIG.  DEST.  VALEUR BS',/)
C
C
C     NC : 1 OU 0, SELON QU'IL S'AGIT DE L'ETAPE CADUCEE 1 OU CADUCEE 2.
C
C
C     NTC :4 OU 3, SELON QU'IL S'AGIT DES COUTS A BORNES INFERIEURES,
C                                  OU DES COUTS A BORNES SUPERIEURES.
C
C
C
C
C
C
1111  CONTINUE
      IF (NC.EQ.1) GO TO 1
    3 NORG=0
    5 NORG=NORG+1
      IF (NORG.GE.N) GO TO 998
      IX=NORG-1
      IF (IX.EQ.0) GO TO 4
```

```
      DO 6 I=1,IX
      IF (NTC.EQ.3) GO TO 12
      NOME(I)=MCOST(NORG,I)
      NOMI(I)=MCOST(NORG,I)
      GO TO 6
   12 NOME(I)=MCOST(I,NORG)
      NOMI(I)=MCOST(I,NORG)
    6 CONTINUE
    4 NOME(NORG)=0
      L=NORG+1
      GO TO 9
    1 L=1
C
C        INITIALISATION DES VECTEURS
C
    9 DO 2 I=L,N
      NOME(I)=32700
      NOMI(I)=32700
    2 CONTINUE
C
C        CHAINES DE LONGUEUR 1
C
      NOMI(NORG)=0
      DO 13 I=1,M
      IF (JAR(I,1).NE.NORG) GO TO 14
      INOM=JAR(I,2)
      IF (NC.EQ.1) GO TO 7
      IF (INOM.LT.NORG) GO TO 13
    7 NOMI(INOM)=JAR(I,NTC)
      GO TO 13
   14 IF (JAR(I,2).NE.NORG) GO TO 13
      INOM=JAR(I,1)
      IF (NC.EQ.1) GO TO 8
      IF (INOM.LT.NORG) GO TO 13
    8 NOMI(INOM)=JAR(I,NTC)
   13 CONTINUE
      LON=1
C
C        CHAINES DE LONGUEUR >1.
C
   27 IF (NC.EQ.1) GO TO 18
      DO 16 I=NORG,N
      NOME(I)=NOMI(I)
   16 CONTINUE
      GO TO 17
   18 DO 15 I=1,N
      NOME(I)=NOMI(I)
   15 CONTINUE
```

```
   17 LON=LON+1
      IF (NC.EQ.1) L=1
      DO 25 I=L,N
      DO 21 KK=1,M
      IT(KK)=0
      IF (JAR(KK,1).EQ.I) IT(KK)=1
      IF (JAR(KK,2).EQ.I) IT(KK)=2
   21 CONTINUE
      DO 22 LL=1,M
      IF (IT(LL).EQ.0) GO TO 22
      IF (IT(LL).EQ.2) GO TO 23
      INODE=JAR(LL,2)
   28 ICOUT=JAR(LL,NTC)+NOME(INODE)
      IF (ICOUT.LT.NOMI(I)) NOMI(I)=ICOUT
      GO TO 22
   23 INODE=JAR(LL,1)
      GO TO 28
   22 CONTINUE
   25 CONTINUE
C
C        TEST DE CONVERGENCE
C
      IF (NC.EQ.1) L=1
C
      DO 26 MM=L,N
      IF (NOME(MM).NE.NOMI(MM)) GO TO 27
   26 CONTINUE
      IF (NC.EQ.1) GO TO 32
      DO 33 I=NORG,N
      IF (NTC.EQ.4) MCOST(I,NORG)=NOME(I)
      IF (NTC.EQ.3) MCOST(NORG,I)=NOME(I)
   33 CONTINUE
      GO TO 5
C
C        CHAINE OPTIMALE. (PAS NECESSAIREMENT UNIQUE)
C
   32 KK=0
      NSOM=NDEST
      GO TO 62
   65 IF (NSOM.EQ.NORG) GO TO 82
   62 DO 70 I=1,M
      IF  (JAR(I,1).EQ.NSOM) GO TO 64
      IF  (JAR(I,2).NE.NSOM) GO TO 70
      JJ=JAR(I,1)
   66 ITE=NOME(NSOM)-JAR(I,NTC)
      IF (ITE.NE.NOME(JJ)) GO TO 70
      KK=KK+1
      IOPT(KK,1)=NSOM
```

DOMINO

```
      IOPT(KK,2)=JJ
      IOPT(KK,3)=NOME(NSOM)
      GO TO 71
   64 JJ=JAR(I,2)
      GO TO 66
   70 CONTINUE
   71 NSOM=JJ
      GO TO 65
C
C        IMPRESSION
C
82    LON=LON-1
      JJ=1
30    IF(IBA.EQ.0) GO TO 1974
      IF (NTC.EQ.4) WRITE(3,103) NORG,LON
      IF (NTC.EQ.3) WRITE(3,111) NORG,LON
1974   CONTINUE
      IF (JJ.EQ.N) GO TO 999
      LIGNE=7
      DO 81 I=JJ,N
      LIGNE=LIGNE+1
      IF (LIGNE.GT.57) GO TO 29
      IF(IBA.EQ.0) GO TO 81
      WRITE(3,104)I,NOMI(I)
   81 CONTINUE
      GO TO 99
   29 JJ=I
      GO TO 30
99    L=KK+1
      IF(IBA.EQ.0) GO TO 1975
      IF (NTC.EQ.4) WRITE(3,105)
      IF (NTC.EQ.3) WRITE(3,112)
1975   CONTINUE
      LLL=7
      DO 69 I=1,KK
      K=L-I
      LLL=LLL+1
      IF (LLL.LE.54) GO TO 67
      IF(IBA.EQ.0) GO TO 1976
      IF (NTC.EQ.4) WRITE(3,105)
      IF (NTC.EQ.3) WRITE(3,112)
1976   CONTINUE
      LLL=7
67    IF(IBA.EQ.0) GO TO 69
      WRITE(3,106) IOPT(K,2),IOPT(K,1),IOPT(K,3)
   69 CONTINUE
      IF(IBA.EQ.0) GO TO 1977
      WRITE(3,107) IOPT(1,3)
```

DOMINO

```
1977   CONTINUE
       GO TO 1000
  998  J1=1
       N2=1
  201  N3=N2+13
       IF (N3.GT.N) N3=N
  202  J=J1+1
       IE=J +43
       IF (IE.GT.N) IE=N
       GO TO (1000,1000,204,203),NTC
  204  WRITE(3,108) (I,I=N2,N3)
       GO TO 205
  203  WRITE(3,110) (I,I=N2,N3)
  205  DO 210 KJ=J,IE
       N4=KJ-1
       IF (N4.GT.N3) N4=N3
       IF (NTC.EQ.4) WRITE(3,109)KJ,(MCOST(KJ,KX),KX=N2,N4)
       IF (NTC.EQ.3) WRITE(3,109)KJ,(MCOST(KX,KJ),KX=N2,N4)
  210  CONTINUE
       IF (IE.EQ.N) GO TO 211
       J1=IE
       GO TO 202
  211  IF  (N3.EQ.N) GO TO 999
       N2=N3+1
       J1=N3+1
       GO TO 201
  999  IF (NC.EQ.1) GO TO 1000
       IF (NTC.NE.3) GO TO 1000
       NTC=4
       GO TO 3
 1000  RETURN
       END
```

BLK DATA

```
BLOCK DATA
INTEGER*2 JAP(1000,60)
COMMON/MOM/JAP
DATA JAP/60000*0/
END
```

SETUP

```
      SUBROUTINE SETUP(JKY,IAMCL,ISTET,ICTGT,ILKDTA,IBCD,IPICT)
      INTEGER*2 JAP(1000,60)
      COMMON/MOM/JAP
      INTEGER*2 DEMAND(145,3)/435*0/,JIM(145)/145*0/,ICHAI(56)/56*0/
      INTEGER CATGRY(4,4),YRGTAC
      COMMON/POP/YRGTAC(4,4)
      COMMON/DAD/NDEM(12),NARR(7),NDEP(7),NCOUP
      INTEGER JOE(145,21)/3045*0/,LKDATA(145,21)/3045*0/,LINK(100)/
     1100*0/,PP,P,Z,W,COST,CAP
      K=0
      M=JKY
      REWIND 4
       DO 27 I=1,M
      READ(4,26) (JOE(I,J),J=1,21)
   26 FORMAT(4I3,12I2,I1,4I6)
   27 CONTINUE
      REWIND 2
      IF(IAMCL.EQ.0) GO TO 4443
      WRITE(6,1376)
1376  FORMAT('1',8X,'ADMISSABLE CHAINS'/'0',1X,'NUMBER',3X,'NORG',2X,
     1'NDEST',2X,'CHAIN',3X,'LINKS IN'/' ','OF LINKS',15X,'NUMBER',
     23X,'CHAIN')
4443  P=8
      MAN = 0
      PP = 0
      LP = 0
3001  READ(2,141)NLINK,NORG,NDEST,NCHAI,(ICHAI(I),I=1,56)
141   FORMAT(60I4)
      IF(NLINK.EQ.9999) GO TO 185
      LP=LP+1
      JAP(LP,1)=NLINK
      JAP(LP,2)=NORG
      JAP(LP,3)=NDEST
      JAP(LP,4)=NCHAI
      NO=4
      DO 170 IQ=1,NLINK
      DO 165 IS=1,M
      IF(((JOE(IS,2).EQ.ICHAI(IQ)).OR.(JOE(IS,3).EQ.ICHAI(IQ))).AND.
     -((JOE(IS,2).EQ.ICHAI(IQ+1))
     1.OR.(JOE(IS,3).EQ.ICHAI(IQ+1)))) GO TO 167
  165 CONTINUE
      GO TO 170
  167 NO=NO+1
      JAP(LP,NO)=JOE(IS,1)
      MAN=MAN+1
      DO 169 IZ=1,MAN
      IF(JAP(LP,NO).NE.JIM(IZ))GO TO 169
      MAN=MAN-1
```

SETUP

```
      GO TO 170
  169 CONTINUE
      PP = PP+1
      JIM(PP)=JAP(LP,NO)
      IF(NLINK.EQ.(NO-4)) GO TO 1665
  170 CONTINUE
1665  CONTINUE
      IF(IAMCL.EQ.0) GO TO 4442
      WRITE(6,191) (JAP(LP,JJ),JJ=1,NO)
191   FORMAT('0',1X,I4,4X,I4,3X,I4,3X,I4,6X,16I4/' ',33X,20I4/' ',33X,20
     1I4)
4442  CONTINUE
      GO TO 3001
 185  CONTINUE
      IF(ISTET.EQ.0) GO TO 4445
      WRITE(6,222)LP,PP
222   FORMAT('0','THERE ARE',1X,I3,1X,'ADMISSABLE CHAINS'/'0','THERE ARE
     1',1X,I3,1X,'ADMISSABLE LINKS')
4445  CONTINUE
      DO 3002 IP= 1,PP
      DO 3003 IS=1,M
      IF(JIM(IP).EQ.JOE(IS,1)) GO TO114
3003  CONTINUE
 114  K=K+1
      DO 161 J=1,21
      LKDATA(K,J)=JOE(IS,J)
161   CONTINUE
3002  CONTINUE
C INITIATE VARIABLES
      RHSNUM=1
      CSTNUM=1
      NUMCHS=LP
      NDEMLK=PP
      NUMLKS=PP
       NDEMPR=NCOUP
C  RHSNUM= CODE FOR RHS VECTOR
C CSTNUM= CODE FOR COST VECTOR
C NUMCHS= NUM OF CHAINS FOR ALL PAIRS OF NODES WITH NEW DEMAND
C NDEMLK (= NUM OF LINKS WITH BASE DEMAND BETWEEN THEIR NODES
C NUMLKS = TOTAL NUM OF LINKS
C NDEMPR = NUM OF PAIRS OF NODES WITH NEW DEMAND BETWEEN THEM
      W=RHSNUM
      Z=CSTNUM
      M= NUMCHS+NDEMLK
      LL=NUMLKS
      MM=NDEMPR+NDEMLK
      IF(ICTGT.EQ.0) GO TO 4446
      WRITE(6,6692)
```

SETUP

```
6692  FORMAT('0',10X,'COST CATEGORIES'/'0','CATG.',7X,'X1',7X,'DX1',7X,
     1'Y1',7X,'DY1')
4446  CONTINUE
      DO 1 I=1,4
      DO 1 J=1,4
      CATGRY(I,J)=YRGTAC(J,I)
    1 CONTINUE
      IF(ICTGT.EQ.0) GO TO 4447
      DO 140 K=1,4
      WRITE(6,600)K, (CATGRY(K,J),J=1,4)
600   FORMAT('0',1X,I2,5X,I5,5X,I5,4X,I5,5X,I5)
  140 CONTINUE
4447  CONTINUE
      IF(ILKDTA.EQ.0) GO TO 4448
      WRITE(6,9248)
9248  FORMAT('1',20X,'ADMISSABLE LINK DATA')
      DO 620 II=1,LL
      WRITE(6,631) (LKDATA(II,JJ),JJ=1,21)
631   FORMAT('0',4(1X,I3),12(1X,I2),2X,I1,4(1X,I6))
  620 CONTINUE
4448  CONTINUE
C   DEMAND WILL BE GIVEN IN A DEMAND VECTOR,
      DO 8819 I=1,NCOUP
      DEMAND(I,1)=NDEP(I)
      DEMAND(I,2)=NARR(I)
      DEMAND(I,3)=NDEM(I)
8819  CONTINUE
      KK=NCOUP
      DO 192 J=1,PP
      KK=KK+1
      DEMAND(KK,1)=LKDATA(J,2)
      DEMAND(KK,2)=LKDATA(J,3)
      DEMAND(KK,3)=LKDATA(J,5)+LKDATA(J,6)+LKDATA(J,8)+LKDATA(J,9) +
     1LKDATA(J,11)+LKDATA(J,12)+LKDATA(J,14)+LKDATA(J,15)
 192   CONTINUE
      REWIND 7
      WRITE(P,1004)
1004  FORMAT ('NAME',10X,'TRANCHE')
      WRITE(P,820)
 820  FORMAT('ROWS')
C SET UP DEMAND RONS
C ROWS FOR DEMAND
      DO 862 KK=1,MM
      WRITE(P,864) DEMAND(KK,1),DEMAND(KK,2)
 864  FORMAT (1X,'G',2X,'D',I3,I3)
 862  CONTINUE
      DO 72 II= 1,LL
      LNK=LKDATA(II,1)
```

SETUP

```
      WRITE(P,54) LNK
   54 FORMAT (1X,'L',2X,'C',I3)
      WRITE(P,55) LNK
   55 FORMAT (1X,'G',2X,'X1DX1',I3)
      WRITE(P,56) LNK
   56 FORMAT (1X,'G',2X,'X2DX2',I3)
      WRITE(P,57) LNK
   57 FORMAT (1X,'G',2X,'X1X2',I3)
      IF (LKDATA(II,13) .EQ. 0) GO TO 72
      WRITE(P,58) LNK
   58 FORMAT (1X,'G',2X,'X2Y1',I3)
      WRITE(P,59) LNK
   59 FORMAT (1X,'G',2X,'Y1DY1',I3)
      WRITE(P,60) LNK
   60 FORMAT (1X,'G',2X,'Y2DY2',I3)
      WRITE(P,61) LNK
   61 FORMAT (1X,'G',2X,'Y1Y2',I3)
   72 CONTINUE
      WRITE(P,42) Z
   42 FORMAT (1X,'N',2X,'COST',I1)
      WRITE(P,821)
  821 FORMAT(    'COLUMNS')
C  SET UP DEMAND ROW   D******
      NPN=100
      DO 805 I=1,LP
      NPN=NPN+1
      NOM=1000+JAP(I,4)
      WRITE(P,8041)NPN,NOM,JAP(I,2),JAP(I,3)
 8041 FORMAT(4X,'R',I3,I4,2X,'D',2I3,7X,'1.0')
      NTN=JAP(I,1)+4
      DO 805 J=5,NTN
      WRITE(P,806)NPN,NOM,JAP(I,J)
  806 FORMAT(4X,'R',I3,I4,2X,'C',I3,10X,'1.0')
  805 CONTINUE
      NOM=1000
      DO 803 I=1,NDEMLK
      NPN=NPN+1
      WRITE(P,804)NPN,NOM,LKDATA(I,2),LKDATA(I,3),NPN,NOM,LKDATA(I,1)
  804 FORMAT(4X,'R',I3,I4,2X,'D',2I3,7X,'1.0'/4X,'R',I3,I4,2X,'C',I3,
     110X,'1.0')
  803  CONTINUE
C SET UP INSTALLED CAPACITY "IDX1***"
      DO 74 II=1,LL
C     CAPACITY ROWS   "C***"
C REMOVE THIS LATER  ***     N.B.    ***       ***   REMOVE THIS LATER
      LNK = LKDATA(II,1)
      WRITE(P,67) LNK,LNK
   67 FORMAT (4X,'IDX1',I3,3X,'C',I3,6X,'-',3X,'1.0')
```

SETUP

```
C      SEQUENCE ROW  "X1X2***"
       WRITE(P,69) LNK,LNK
   69 FORMAT (4X,'IDX1',I3,3X,'X1X2',I3,7X,'1.0')
C
C     CAPACITY ROW "C***"
       WRITE(P,81) LNK,LNK
   81 FORMAT (4X,'IDX2',I3,3X,'C',I3,6X,'-',3X,'1.0')
C     TEST IF Y IS TO BE  ADDED
       IF (LKDATA(II,13) .EQ. 0) GO TO 74
C     SEQUENCE ROW  "X2Y1***"
       WRITE(P,82) LNK,LNK
   82 FORMAT (4X,'IDX2',I3,3X,'X2Y1',I3,7X,'1.0')
C   ADDIDY14 IDY2  FOR CASE  OF Y SYSTEM
C   IDY1***
       WRITE(P,450) LNK,LNK
  450 FORMAT (4X,'IDY1',I3,3X,'C',I3,6X,'-',3X,'1.0')
       WRITE(P,451) LNK,LNK
  451 FORMAT (4X,'IDY1',I3,3X,'Y1Y2',I3,7X,'1.0')
C   IDY2***
       WRITE(P,452) LNK,LNK
  452 FORMAT (4X,'IDY2',I3,3X,'C',I3,6X,'-',3X,'1.0')
   74 CONTINUE
C
C     SET UP  LINK ADDITION VARIABLES "X1***", "DX1***" ETC.
C   TEST IF LINK HAS X SYSYEM  ( TEST ON "X1" SORRY  "LX1")
       IF (LKDATA(II,7) .EQ. 0) GO TO 503
C    503   U  YS START OF  Y1
C   VARIABLE     "X1***"
       WRITE(P,1245)
1245   FORMAT(4X,'MARKER1',3X,'''MARKER''',17X,'''INTORG''')
       DO 75 II =1,LL
       LNK = LKDATA(II,1)
C   TAKE OUT  LATER                          ***          "JJT"              ***
C    CAPACITY  "C***"
       WRITE(P,84) LNK,LNK
   84 FORMAT (4X,'X1',I3,5X,'C',I3,6X,'-',3X,'1.0')
C     SEQUENCE ROW  "X1DX1***"
       WRITE(P,85) LNK,LNK
   85 FORMAT (4X,'X1',I3,5X,'X1DX1',I3,6X,'9.0')
C
C     SEQUENCE ROW  "X1X2***"
       WRITE(P,86) LNK,LNK
   86 FORMAT (4X,'X1',I3,5X,'X1X2',I3,7X,'1.0')
C
C
C      COST  ROW
C      TEST IF X1 = 0  IF SO  ADD COST, IF NOT  COST=0   NB X1=LKDATA(II,5)
       IF (LKDATA(II,17) .EQ. 1) GO TO 77
```

```
      IF (LKDATA(II,17) .EQ. 2) GO TO 78
      IF (LKDATA(II,17) .EQ. 3) GO TO 79
      IF (LKDATA(II,17) .EQ. 4) GO TO 88
      IF (LKDATA(II,17) .EQ. 5) GO TO 710
      IF (LKDATA(II,17) .GT. 5) GO TO 711
  711 WRITE(6,712) LNK
  712 FORMAT(1H,'ERROR',I3)
      GO TO 75
  710 JCAT=5
      GO TO 504
   77 JCAT = 1
      GO TO 504
   78 JCAT = 2
      GO TO 504
   79 JCAT = 3
      GO TO 611
   88 JCAT = 4
      GO TO 504
C                        N.B.CAT IS LKDATA(II,17)
C     NOTE  9 IS  $9000.UNIT COST OF X1
  504 COST = (LKDATA(II,4))*(CATGRY(JCAT,1))
      GO TO 610
C  FOR CAT.3  SEE IF COST IS GIVEN IN LKDATA(II,18)
  611 IF (LKDATA(II,18) .EQ. 0) GO TO 605
      COST = LKDATA(II,18)
      GO TO 610
  605 COST = CATGRY(JCAT,1)*LKDATA(II,4)
      GO TO 610
C  ASSUME  THAT COST IS AS FOR CAT. 1
C  CHECK TO SEE IF X1 IS ALREADY INSYTALLED
  610 IF (LKDATA(II,5) .GT. 0) GO TO 501
C     X1 IS NOT INSTALLED   THUS ADD COST ,
      GO TO 89
  501 COST = 0
   89 WRITE(P,90) LNK,Z,COST
   90 FORMAT (4X,'X1',I3,5X,'COST',I1,5X,I5,'.')
C   VARIABLE "DX1***" ,(DELTA X)
C
C   CAPACITY  ROW  "C***"
      WRITE(P,91) LNK,LNK
   91 FORMAT (4X,'DX1',I3,4X,'C',I3,6X,'-',3X,'1.0')
C
C  SEQUENCE  "X1DX1***"
      WRITE(P,92) LNK,LNK
   92 FORMAT (4X,'DX1',I3,4X,'X1DX1',I3,2X,'-',3X,'1.0')
C
C  SEQUENCE  ROW "X1X2***"
      WRITE(P,93) LNK,LNK
```

SETUP

```
   93 FORMAT (4X,'DX1',I3,4X,'X1X2',I3,7X,'1.0')
C
C
C  COST :UNLESS CAT. 3, COMPUTE ON CAT VECTOR   NB:DX = CATGRY(JCAT,2)
      IF (JCAT .EQ. 3) GO TO 95
      COST =CATGRY(JCAT,2)*LKDATA(II,4)
      GO TO 97
C
C   FOR CAT 3  SEE IF COST IS GIVEN IN LKDATA(I,19)
   95 IF (LKDATA(II,19) .EQ. 0) GO TO 98
      COST = LKDATA(II,19)
      GO TO 97
   98 COST = CATGRY(JCAT,2)*LKDATA(II,4)
C   SET UP"COST1"
   97 WRITE(P,99) LNK,Z,COST
   99 FORMAT (4X,'DX1',I3,4X,'COST',I1,5X,I5,'.')
C   SEE IF THE X2 SYSTEM IS IN.  EST ON LKDATA(II,10)
      IF (LKDATA(II,10) .EQ. 0) GO TO 75
C  VARIABLE  "X2***"
C
C  CAPACITY ROW   "C***"
      WRITE(P,100) LNK,LNK
  100 FORMAT (4X,'X2',I3,5X,'C',I3,6X,'-',3X,'1.0')
C
C SEQUENCE  ROW  "X2DX2***"
      WRITE(P,101) LNK,LNK
  101 FORMAT (4X,'X2',I3,5X,'X2DX2',I3,6X,'9.0')
C  SEQUENCE  ROW  "X1X2***"   VALUE = CAPACITY LTMIT ON X1= LKDATA(II,7)
      WRITE(P,102) LNK,LNK,LKDATA(II,7)
  102 FORMAT (4X,'X2',I3,5X,'X1X2',I3,3X,'-',I4,'.')
C
C    TEST IF Y IS TO BE ADDED
      IF (LKDATA(II,13) .EQ. 0) GO TO 701
      WRITE(P,702) LNK,LNK
  702 FORMAT (4X,'X2',I3,5X,'X2Y1',I3,7X,'1.0')
C COST :USE   CAT VECTOR    X2 = CATGRY(JCAT,2)* 3
  701 IF (LKDATA(II,8) .EQ. 0) GO TO 400
      COST = 0
      GO TO 105
  400 COST = CATGRY(JCAT,2)*3*LKDATA(II,4)
      GO TO 105
  105 WRITE(P,107) LNK,Z,COST
  107 FORMAT (4X,'X2',I3,5X,'COST',I1,5X,I5,'.')
C
C  VARIABLE "DX2***"
C
C  CAPACITY  "C***"
      WRITE(P,108) LNK,LNK
```

SETUP

```
  108 FORMAT (4X,'DX2',I3,4X,'C',I3,6X,'-',3X,'1.0')
C  SEQUENCE ROW "X2DX2***"
      WRITE(P,109) LNK,LNK
  109 FORMAT (4X,'DX2',I3,4X,'X2DX2',I3,2X,'-',3X,'1.0')
C     TEST  IF Y  IS TO BE ADDED
      IF (LKDATA(II,13) .EQ. 0) GO TO 706
      WRITE(P,703) LNK,LNK
  703 FORMAT (4X,'DX2',I3,4X,'X2Y1',I3,7X,'1.0')
C  COST . SEE  IF  CAT.3. COMPUTE  FROM VECTOR  DX=CATGRY(JCAT,2)
  706 IF (JCAT .EQ. 3) GO TO 111
      COST = CATGRY(JCAT,2)*LKDATA(II,4)
      GO TO 112
C     FOR CAT 3,SEE IF COST GIVEN
  111 IF (LKDATA(II,19) .EQ. 0) GO TO 113
      COST = LKDATA(II,19)
      GO TO 112
  113 COST = CATGRY(JCAT,2) * LKDATA(II,4)
  112 WRITE(P,130) LNK,Z,COST
  130 FORMAT (4X,'DX2',I3,4X,'COST',I1,5X,I5,'.')
C  TEST TO  SEE  IF "Y" SYSTEM IS REQUIRED. TEST ON  LKDATA(II,13),LY1
      IF (LKDATA(II,13) .EQ. 0) GO TO 75
C
C 'VARIABLE "Y1***"
C
C    CAPACITY "C***"
  503 WRITE(P,115) LNK,LNK
  115 FORMAT (4X,'Y1',I3,5X,'C',I3,6X,'-',3X,'1.0')
C
C    SEQUENCE ROW "X2Y1***"   VALUE = CAP.LIMIT ON X2 = LKDATA(II,10)
      WRITE(P,116) LNK,LNK,LKDATA(II,10)
  116 FORMAT (4X,'Y1',I3,5X,'X2Y1',I3,3X,'-',I4,'.')
C
C SEQUENCE  "Y1DY1***"
      WRITE(P,117) LNK,LNK
  117 FORMAT (4X,'Y1',I3,5X,'Y1DY1',I3,6X,'9.0')
C
C SEQUENCE ROW "Y1Y2***"
      WRITE(P,118) LNK,LNK
  118 FORMAT (4X,'Y1',I3,5X,'Y1Y2',I3,7X,'1.0')
C
C   ASSUME  NO  CAT 3 FOR Y
C COST ROW   USE  CAT. VECTOR . COST OF,  = CATGRY(JCAT,3).
      IF (LKDATA(II,11) .EQ. 0) GO TO 997
      COST = 0
      GO TO 998
  997 IF (LKDATA(II,20) .EQ. 0) GO TO 999
      COST = LKDATA(II,20)
      GO TO 998
```

SETUP

```
  999 COST = CATGRY(JCAT,3)* LKDATA(II,4)
  998 WRITE(P,128) LNK,Z,COST
  128 FORMAT (4X,'Y1',I3,5X,'COST',I1,5X,I5,'.')
C VARIABLE "DY1***"
C   CAPACITY  ROW "C***"
      WRITE(P,120) LNK,LNK
  120 FORMAT (4X,'DY1',I3,4X,'C',I3,6X,'-',3X,'1.0')
C
C   SEQUENCE  ROW "Y1DY1***"
      WRITE(P,121) LNK,LNK
  121 FORMAT (4X,'DY1',I3,4X,'Y1DY1',I3,2X,'-',3X,'1.0')
C   SEQUENCE  ROW "Y1Y2***"
      WRITE(P,122) LNK,LNK
  122 FORMAT (4X,'DY1',I3,4X,'Y1Y2',I3,7X,'1.0')
C
C
C ASSUME  NO  CAT3. FOR Y.
C COST BY SAME  METHOD AS  FOR "Y1***"  BUT DY IS  CATGRY(JCAT,4)
      COST = (CATGRY(JCAT,4))*(LKDATA(II,4))
      WRITE(P,127) LNK,Z,COST
  127 FORMAT (4X,'DY1',I3,4X,'COST',I1,5X,I5,'.')
C SEE IF THE Y2 SYSTEM IS IN.  EST ON LKDATA(II,16)
      IF (LKDATA(II,16) .EQ. 0) GO TO 75
C  VARIABLE "Y2***"
C  CAPACITY  ROW "C***"
      WRITE(P,124) LNK,LNK
  124 FORMAT (4X,'Y2',I3,5X,'C',I3,6X,'-',3X,'1.0')
C
C   SEQUENCE ROW  "Y1Y2***"    VALUE =LKDATA(II,13)= LY1
      WRITE(P,125) LNK,LNK,LKDATA(II,13)
  125 FORMAT (4X,'Y2',I3,5X,'Y1Y2',I3,3X,'-',I4,'.')
C
C   VARIABLE "Y2***"
C    SEQUENCING ROW  "Y2DY2***"
      WRITE(P,136) LNK,LNK
  136 FORMAT (4X,'Y2',I3,5X,'Y2DY2',I3,6X,'9.0')
C
C   COST.  ASSUME NO CAT 3  TREAT AS BEFORE   COST=DY*3
      IF (LKDATA(II,14) .EQ. 0) GO TO 401
      COST = 0
      GO TO 402
  401 COST = CATGRY(JCAT,4)*3*LKDATA(II,4)
  402 WRITE(P,131) LNK,Z,COST
  131 FORMAT (4X,'Y2',I3,5X,'COST',I1,5X,I5,'.')
C   VARIABLE "DY2***"
C
C   CAPACITY ROW "C***"
      WRITE(P,132) LNK,LNK
```

SETUP

```
  132 FORMAT (4X,'DY2',I3,4X,'C',I3,6X,'-',3X,'1.0')
C
C  SEQUENCE ROW  "Y2DY2***"
      WRITE(P,133) LNK,LNK
  133 FORMAT (4X,'DY2',I3,4X,'Y2DY2',I3,2X,'-',3X,'1.0')
C
C
C   COST: AS  BEFORE      COST=DY1
      COST =  CATGRY(JCAT,4)*(LKDATA(II,4))
      WRITE(P,135) LNK,Z,COST
  135 FORMAT (4X,'DY2',I3,4X,'COST',I1,5X,I5,'.')
C
   75 CONTINUE
C  SET UP  RIGHT HAND SIDES
C  SET UP       DEMAND ,ROW "D******++"
      WRITE(P,1246)
1246  FORMAT(4X,'MARKER2',3X,'''MARKER''',17X,'''INTEND''')
      WRITE(P,822)
  822 FORMAT('RHS')
      DO 215 KK=1,MM
C
C   FORMAT  IS  4X,RHS1,6X,ROWNAME
  215 WRITE(P,216) W,DEMAND(KK,1),DEMAND(KK,2),DEMAND(KK,3)
  216 FORMAT (4X,'RHS',I1,6X,'D',I3,I3,4X,I4,'.')
C
C BOUNDS
C
C BOUNDS ON  INSTALLED CAPACITY"IDX1***" ETC.
      WRITE(P,250)
  250 FORMAT ('BOUNDS')
      DO 251 II=1,LL
      LNK = LKDATA(II,1)
C  BOUND FORMAT  "FX"
C
C  BOUND  FOR "IDX1***"  VALUE = IDX1 =LKDATA(II,6)
      WRITE(P,252) LNK,LKDATA(II,6)
  252 FORMAT (1X,'FX',1X,'BND1',6X,'IDX1',I3,4X,I4,'.')
C
C  BOUND  FOR "IDX2***"  VALUE = IDX2=LKDATA(II,9)
      WRITE(P,253) LNK,LKDATA(II,9)
  253 FORMAT (1X,'FX',1X,'BND1',6X,'IDX2',I3,4X,I4,'.')
C  TEST IF "Y" IS NEEDED
      IF (LKDATA(II,13) .EQ. 0) GO TO 251
C
C  BOUND  FOR "IDY1***"  VALUE = IDY1 = LKDATA(II,12)
      WRITE(P,254) LNK,LKDATA(II,12)
  254 FORMAT (1X,'FX',1X,'BND1',6X,'IDY1',I3,4X,I4,'.')
C
```

SETUP

```
C    BOUND FOR  "IDY2***" VALUE = IDY2 =LKDATA(II,15)
       WRITE(P,255) LNK,LKDATA(II,12)
  255 FORMAT (1X,'FX',1X,'BND1',6X,'IDY2',I3,4X,I4,'.')
  251 CONTINUE
       DO 256 II=1,LL
       LNK = LKDATA(II,1)
C
C  BOUNDS ON INSTALLED "X1***"      "X2***","Y1***","Y2***"
C  BOUND FORMAT  "LO" ,       =
C  "X1"
C  VALUE  OF X1 IS IN CKDATA(II;5)  IF =0,LO=0 UP=1.,IF NOT=0
C    1                                                  FX=LKDATA(II,5)
       IF (LKDATA(II,5) .EQ. 0) GO TO 961
       WRITE(P,962) LNK,LKDATA(II,5)
  962 FORMAT (1X,'FX',1X,'BND1',6X,'X1',I3,6X,I4,'.')
       GO TO 963
  961 WRITE(P,964) LNK
  964 FORMAT (1X,'LO',1X,'BND1',6X,'X1',I3,9X,'0.0')
       WRITE(P,965) LNK
  965 FORMAT (1X,'UP',1X,'BND1',6X,'X1',I3,9X,'1.0')
  963 IF (LKDATA(II,5) .EQ. 0) GO TO 975
       LDX = LKDATA(II,7)- LKDATA(II,5)-LKDATA(II,6)
       GO TO 980
  975 LDX = LKDATA(II,7)-LKDATA(II,6)-1
  980 IF (LDX .LE. 0) GO TO 43
       WRITE(P,912) LNK
  912 FORMAT (1X,'LO',1X,'BND1',6X,'DX1',I3,8X,'0.0')
       WRITE(P,640) LNK,LDX
  640 FORMAT (1X,'UP',1X,'BND1',6X,'DX1',I3,5X,I4,'.')
       GO TO 1000
   43 WRITE(P,44) LNK
   44 FORMAT (1X,'FX',1X,'BND1',6X,'DX1',I3,8X,'0.0')
C
C  FOR X2, VALUE IS :LKDATA(II,8)
 1000 IF (LKDATA(II,8) .EQ. 0) GO TO 966
       WRITE(P,967) LNK,LKDATA(II,8)
  967 FORMAT (1X,'FX',1X,'BND1',6X,'X2',I3,6X,I4,'.')
       GO TO 968
  966 WRITE(P,969) LNK
  969 FORMAT (1X,'LO',1X,'BND1',6X,'X2',I3,9X,'0.0')
       WRITE(P,970) LNK
  970 FORMAT (1X,'UP',1X,'BND1',6X,'X2',I3,9X,'1.0')
  968 IF (LKDATA(II,8) .EQ. 0) GO TO 976
       LDX =  LKDATA(II,10) - LKDATA(II,8)-LKDATA(II,9)
       GO TO 981
  976 LDX = LKDATA(II,10) -LKDATA(II,9)-1
  981 IF (LDX .LE. 0) GO TO 45
       WRITE(P,913) LNK
```

```
  913 FORMAT (1X,'LO',1X,'BND1',6X,'DX2',I3,8X,'0.0')
      WRITE(P,641) LNK,LDX
  641 FORMAT (1X,'UP',1X,'BND1',6X,'DX2',I3,5X,I4,'.')
      GO TO 1001
   45 WRITE(P,46) LNK
   46 FORMAT (1X,'FX',1X,'BND1',6X,'DX2',I3,8X,'0.0')
C     SEE IF  Y1 9Y2 =0
 1001 IF (LKDATA(II,13) .EQ. 0) GO TO 256
      IF (LKDATA(II,11) .EQ. 0) GO TO 902
      WRITE(P,903) LNK,LKDATA(II,11)
  903 FORMAT (1X,'FX',1X,'BND1',6X,'Y1',I3,6X,I4,'.')
      GO TO 906
  902 WRITE(P,904) LNK
  904 FORMAT (1X,'LO',1X,'BND1',6X,'Y1',I3,9X,'0.0')
      WRITE(P,905) LNK
  905 FORMAT (1X,'UP',1X,'BND1',6X,'Y1',I3,9X,'1.0')
  906 IF (LKDATA(II,11) .EQ. 0) GO TO 977
      LDX=LKDATA(II,13)-LKDATA(II,11)-LKDATA(II,12)
      GO TO 982
  977 LDX = LKDATA(II,13) - LKDATA(II,12) -1
  982 IF (LDX .LE. 0) GO TO 47
      WRITE(P,914) LNK
  914 FORMAT (1X,'LO',1X,'BND1',6X,'DY1',I3,8X,'0.0')
      WRITE(P,642) LNK,LDX
  642 FORMAT (1X,'UP',1X,'BND1',6X,'DY1',I3,5X,I4,'.')
      GO TO 1002
   47 WRITE(P,48) LNK
   48 FORMAT (1X,'FX',IX,'BND1',6X,'DY1',I3,8X,'0.0')
C  Y2***
 1002 IF (LKDATA(II,14) .EQ. 0) GO TO 907
      WRITE(P,908) LNK,LKDATA(II,14)
  908 FORMAT (1X,'FX',1X,'BND1',6X,'Y2',I3,6X,I4,'.')
      GO TO 916
  907 WRITE(P,910) LNK
  910 FORMAT (1X,'LO',1X,'BND1',6X,'Y2',I3,9X,'0.0')
      WRITE(P,911) LNK
  911 FORMAT (1X,'UP',1X,'BND1',6X,'Y2',I3,9X,'1.0')
  916 IF (LKDATA(II,14) .EQ. 0) GO TO 978
      LDX = LKDATA(II,16) - LKDATA(II,14)-LKDATA(II,15)
      GO TO 983
  978 LDX = LKDATA(II,16) -LKDATA(II,15) - 1
  983 IF (LDX .LE. 0) GO TO 49
      WRITE(P,915) LNK
  915 FORMAT (1X,'LO',1X,'BND1',6X,'DY2',I3,8X,'0.0')
      WRITE(P,643) LNK,LDX
  643 FORMAT (1X,'UP',1X,'BND1',6X,'DY2',I3,5X,I4,'.')
      GO TO 256
   49 WRITE(P,50) LNK
```

SETUP

```
   50 FORMAT (1X,'FX',1X,'BND1',6X,'DY2',I3,8X,'0.0')
  256 CONTINUE
      WRITE(P,1003)
 1003 FORMAT ('ENDATA')
      ENDFILE8
      REWIND 4
      WRITE(4,9241)
9241  FORMAT(9X,'PROGRAM'/9X,'INITIALZ'/9X,'XPROC=9000'/9X,'MOVE(XDATA,
     1''TRANCHE'')'/9X,'MOVE(XPBNAME,''PROB'')'/9X,'CONVERT')
      IF(IBCD.EQ.0) GO TO 7002
      WRITE(4,7001)
7001  FORMAT(9X,'BCDOUT')
7002  WRITE(4,7003)
7003  FORMAT(9X,'SETUP(''BOUND'',''BND1'')')
       IF(IPICT.EQ.0) GO TO 7004
      WRITE(4,7005)
7005  FORMAT(9X,'PICTURE')
7004  WRITE(4,7006)
7006  FORMAT(9X,'MOVE(XOBJ,''COST1'')'/9X,'MOVE(XRHS,''RHS1'')'/
     19X,'OPTIMIZE'/
     19X,'SOLUTION'/9X,'INIMIX'/9X,'OPTIMIX(''COST'',9500.,10,10,1)'/
     29X,'EXIT'/9X,'PEND')
      ENDFILE4
      RETURN
      END
```

CACC / CCAC
82958

HERMES 1 USER'S MANUAL

P
91
C6541
H47
1972

DATE DUE
DATE DE RETOUR

LOWE-MARTIN No. 1137