



IMPLEMENTACIÓ INFORMÀTICA

CCUPB/BIBLIOTECA

N.º REGISTRO: 193

FECHA: 27.06.01

20/5/02

8 1 ALGORISME DE GENERACIO

8.1.1. ESTRUCTURA GENERAL.

L'algorisme de generació s'ha programat en llenguatge FORTRAN, del qual n'existeixen 2 versions: l'una, preparada per a un ordinador FACOM 230-25, i l'altra per a un IBM 360/370. La diferència més significativa rau en les variables enteres (o INTEGER); el compilador del primer les pren, per defecte, com de 2 bytes, mentre que el segon, com de 4 bytes. Aquest detall implica que algunes sentències de declaració hagin de ser diferents en una i altra versions.

Consta de 16 mòduls, a més del programa principal i, per un dimensionament de 44 zones, ve a ocupar unes 86 K.; el programa tot sol però, n'ocupa cap a 50 K. El programa s'encarrega de la preparació de les dades per a fer-les directament processables per l'algorisme, així com de la sortida de la informació fonamental i complementària, tant al cap de cada iteració com al final del procés.

8.1.2. RELACIO D'ARXIUS.

El programa no utilitza arxius de treball durant el procés.

La càrrega de dades comprèn:

- paràmetres.
- matrius de demandes.
- " de cost (o distància) a peu.
- " de temps de recorregut o distància en bus.
- eventualment, xarxa donada.

Llur lectura es fa al començament del programa, seqüencialment; el suport físic de les matrius pot ser adés la cinta, adés el disc, adés la tarja perforada. El canvi d'opció es fa per programa, en el block data, canviant el número de les unitats lògiques d'entrada.

La sortida és per impressora, per bé que és fàcil de fer gravar la xarxa resultant damunt de tarja o suport magnètic, així com d'altres arxius, com ara la matriu final de costos mitjans.

Es veu doncs, que és un programa amb estructura típicament científica:

- . ocupació important de memòria central.
- . poques entrades i sortides.
- . possibilitat de reduir les entrades a tarja i les sortides a impressora.

8.1.3. SORTIDES O INFORMACIO FORNIDA PEL PROGRAMA.

8.1.3.1. Paràmetres del procés.

Es un llistat que s'imprimeix al començament del programa, i que informa sobre dades definitòries d'aquell procés en concret. Es distingeixen els següents grups de paràmetres:

- físics.
- de comportament de l'usuari.
- socio-econòmics.
- de limitació geomètrica.

El quadre de la figura 8.1.1., en dona totes les característiques. El llistat adjunt n'és un exemple, corresponent a la xarxa X6.

NOM	NOM O EQUIVALENT EN L'ALGORISME	VARIABLE EN EL PROGRAMA	UNITAT EN QUE S'EXPRESSA
Nombre de zones	n	NN	-
Nombre de busos	A	AUT	-
Velocitat a peu	-	V	m./min.
Velocitat en bus	-	YBUS	m./min.
Penalització de l'espera	P	PE	-
Equivalent en temps de la tarifa	E	E	min.
Sensibilitat per a la diferència de costos	β	BETA	min. ⁻¹
Diferència entre M i el cost mínim	h	HAC	min.
Població activa total	$\sum_i \sum_j d_{ij}$	W	
Població potencial absorbible amb hipòtesi de busos infinits	-	WP	
Cost per mitjans alternatius	$\frac{\sum_i \sum_j d_{ij} p_{ij}}{\sum_i \sum_j d_{ij}}$	C1	min.
Cost unitari mínim, sota hipòtesi anterior	-	C2	min.
Nombre màxim inicial de nusos/línia	-	LINUS	-
Nombre màxim d'iteracions	-	ITMAX	-
Coefficient d'inserció	γ_i	GEØ	-
Coefficient d'inserció a l'extrem	$\sqrt{1+\gamma_i^2}$	GEX	-
Coefficient de clausura	γ_c	COCLA	-
Coefficient de ruptura	γ_r	COBURP	-

FIG. 8.1.1

PARAMETRES DEL PROCES

NOMBRE DE ZONES 6
NOMBRE DE BUSOS 50

VELOCITAT A PEU 66.667 M./MIN.
VELOCITAT DEL BUS 200.000 M./MIN.
PENALITZACIO DE L'ESPERA 1
EQUIVALENT DE LA TARIFA 3 MIN.
SENSIBILITAT A LA DIF. DE COSTOS 2.00
DIFERENCIA AMB EL COST MINIM 0.50 MIN.

POBLACIO ACTIVA TOTAL 12750
POBLACIO POTENCIAL ABSORBIBLE 7459
COST INICIAL 8.78431 MIN.
COST MINIM 5.22709 MIN.

NOMBRE MAXIM INICIAL DE BUSOS/LINIA 4
MAXIM D'ITERACIONS 10

PARAMETRES GEOMETRICS

INSERCIO (EL·LIPTIC) 5.00
INSERCIO A L'EXTREM (HIPERBOLIC), 0.20
CLAUSURA 2.00
RUMTURA 0.50

8.1.3.2. Informació sobre les línies. Llistat reduït.

Després de generar o tractar cada línia, s'imprimeixen una sèrie de dades relatives a ella, així com d'altres que informen sobre l'estat de la xarxa en aquell moment, o sigui, de com la variació i els canvis que han tingut lloc a la línia han incidit en els paràmetres generals de la xarxa.

Concretament, el llistat reduït indica per a cada línia:

- el seu número identificatiu.
- la càrrega total absorbida.
- la longitud de la línia en Km.
- el nombre total de nusos que la constitueixen.
- el nombre de nusos inserits en la darrera iteració.
- el nombre de nusos suprimits en la darrera iteració.
- una informació eventual sobre si la línia és circular o ha estat eliminada.

Pel que fa a la xarxa calcula:

- la demanda total absorbida.
- la longitud total de la xarxa en Km.
- el cost mitjà en aquell instant.

El llistat adjunt també correspon a la xarxa X6.

ITERACIO 1, LINIES.

INFORMACIO SOBRE LA LINIA

INFORMACIO SOBRE LA XARXA

LI. LA	DEMANDA	LONGITUD (KM.)	NUSOS ACTUALS	NUSOS INSERTATS	NUSOS SUPRIMITS	DEMANDA	LONGITUD (KM.)	COST (MIN.)	LINIA
1	1143	2.132	4	0	2	6392	16.130	6.16544	1
2	1257	1.200	3	1	3	6972	14.798	6.01829	2
3	214	1.200	2	0	3	6755	12.798	5.95924	3
4			0		6				ELIMINADA
4	6633	4.266	6	0	0	6815	8.532	5.85309	4

Handwritten marks and scribbles at the top of the page.

8.1.3.3. Informació sobre les línies (l·listat complet).

Es alternatiu de l'anterior, segons el valor que pren un paràmetre de entrada. Forneix totes les informacions del l·listat reduït i, a més, les següents:

- la l·lista dels nusos que la constitueixen.
- la càrrega per tram (entre nus i nus).
- el temps de recorregut entre nusos.
- la l·lista dels nusos suprimits i inscrits en la darrera iteració.

Cal aclarir, a més, que en cas que hom demani el l·listat reduït, al final en fa un de complet; aquesta opció, doncs, té interès quan només cal conèixer el resultat final, sense que convinguin gai re els intermedis.

El l·listat adjunt correspon a una prova de la xarxa X44.

LINIA * 7*

DEM.LIN.	20640	ORD.	1	2	3	4	5	6	7	8	9	10	11	12
DEM.XRX.	497301	TEMPS	9.25	5.11	10.40	7.51	5.10	10.57	7.40	9.04	9.96	9.69	7.85	
LNG.LIN.	18.440	CAR.	1603	2945	3392	6258	5274	5684	5138	5547	5827	5754	3076	
LNG.XRX.	387.686	NUS.	38	34	31	28	21	18	17	20	25	33	32	36
COST	26.99876													

LINIA * 8*

DEM.LIN.	19582	ORD.	1	2	3	4	5	6	7	8	9	10
DEM.XRX.	497301	TEMPS	9.91	6.14	9.14	7.58	8.10	5.10	10.07	10.46	9.59	
LNG.LIN.	15.217	CAR.	3062	5391	8941	5563	5305	4363	3653	3127	2456	
LNG.XRX.	387.686	NUS.	24	19	20	26	30	31	34	35	37	36
COST	26.99876											

LINIA * 9*

DEM.LIN.	15453	ORD.	1	2	3	4	5	6	7	8	9	10
DEM.XRX.	496875	TEMPS	11.00	8.96	5.46	9.18	6.14	5.88	8.01	8.22	5.10	
LNG.LIN.	15.589	CAR.	2817	4489	4452	5146	5793	5310	5147	3750	2531	
LNG.XRX.	387.024	NUS.	1	2	7	12	13	17	22	27	31	34
COST	26.98772											

SUPRIMITS 23

LINIA *10*

DEM.LIN.	20136	ORD.	1	2	3	4	5	6	7	8	9	10	11
DEM.XRX.	496675	TEMPS	7.52	5.38	5.10	8.21	5.10	9.55	10.93	7.65	6.41	6.09	
LNG.LIN.	14.609	CAR.	2695	3440	7028	8858	8981	7937	6678	5821	4827	2611	
LNG.XRX.	387.024	NUS.	21	28	23	27	31	34	38	41	43	42	44
COST	26.98772												

LINIA *11*

DEM.LIN.	15154	ORD.	1	2	3	4	5	6	7	8	9
DEM.XRX.	496734	TEMPS	8.31	5.88	8.56	7.75	9.90	5.76	8.43	7.85	
LNG.LIN.	12.429	CAR.	3765	4087	6559	5265	4734	5380	5103	2896	
LNG.XRX.	385.381	NUS.	27	22	17	16	19	24	29	32	36
COST	26.98618										

SUPRIMITS 31

LINIA *12*

DEM.LIN.	22271	ORD.	1	2	3	4	5	6	7	8	9	10	11	12	13
DEM.XRX.	497704	TEMPS	7.24	10.79	8.50	9.87	7.57	5.27	7.04	12.62	8.58	6.35	9.72	9.06	
LNG.LIN.	20.526	CAR.	2936	4610	6293	7302	6656	6345	4399	2526	2860	3670	4647	4076	
LNG.XRX.	385.381	NUS.	40	39	37	33	30	26	27	28	18	14	12	8	4
COST	26.97070														

INSERTATS 8

LINIA *13*

DEM.LIN.	28806	ORD.	1	2	3	4	5	6	7	8	9	10	11	12	13
DEM.XRX.	497581	TEMPS	12.05	5.95	5.46	6.35	8.59	8.17	5.70	8.21	8.10	9.87	9.69	7.85	
LNG.LIN.	19.201	CAR.	1270	4872	5814	5906	6421	8853	6689	6237	7303	7314	6416	2651	
LNG.XRX.	387.790	NUS.	5	6	7	12	14	18	23	27	31	30	33	32	36
COST	26.96417														

INSERTATS 5

LINIA *14*

DEM.LIN.	20317	ORD.	1	2	3	4	5	6	7	8	9	10			
DEM.XRX.	497581	TEMPS	6.56	5.70	5.27	6.85	9.96	8.51	10.79	7.24	8.42				
LNG.LIN.	13.860	CAR.	3794	8284	9633	9995	7611	6125	4807	3834	1893				
LNG.XRX.	387.790	NUS.	21	23	27	26	25	33	37	39	40	43			
COST	26.96417														

INSERTATS 4

LINIA *15*

DEM.LIN.	20324	ORD.	1	2	3	4	5	6	7	8	9	10	11	12	13
DEM.XRX.	497594	TEMPS	7.09	9.68	10.97	6.89	6.14	5.88	11.49	10.40	5.10	10.07	11.73	9.69	
LNG.LIN.	21.029	CAR.	1356	2199	2879	6163	5731	5484	4484	2290	2661	4237	4414	3405	
LNG.XRX.	389.207	NUS.	4	11	15	14	13	17	22	28	31	34	35	33	32
COST	26.96332														

INSERTATS 4

LINIA *16*

DEM.LIN.	21595	ORD.	1	2	3	4	5	6	7	8	9	10	11	12	13
DEM.XRX.	498170	TEMPS	9.55	5.11	8.10	9.87	9.91	5.76	9.90	7.75	5.37	8.58	10.79	5.10	
LNG.LIN.	19.162	CAR.	1337	4347	5767	5325	6117	4554	3497	4260	4596	5078	4406	1810	
LNG.XRX.	389.872	NUS.	38	34	31	30	33	29	24	19	16	9	13	18	21
COST	26.95656														

INSERTATS 31

LINIA *22*

DEM.LIN.	19472	ORD.	1	2	3	4	5	6	7	8	9	10	11	12	13
DEM.XRX.	498902	TEMPS	11.00	7.03	14.78	5.37	7.75	6.14	6.88	5.61	6.56	10.76	9.68	7.08	
LNG.LIN.	19.732	CAR.	2593	3585	3883	5082	5779	6104	8765	8290	5597	2061	1552	1365	
LNG.XRX.	395.958	NUS.	1	2	6	9	16	19	20	22	23	21	15	11	4
COST	26,93555														

INSERTATS 19

LINIA *23*

DEM.LIN.	22523	ORD.	1	2	3	4	5	6	7	8	9	10	11	12
DEM.XRX.	500267	TEMPS	9.07	8.03	5.95	6.27	11.66	6.59	6.14	5.88	5.93	7.57	9.88	
LNG.LIN.	16.993	CAR.	5636	6377	4609	2901	2312	3658	4853	5598	5936	5370	3540	
LNG.XRX.	398.092	NUS.	4	8	7	6	10	9	13	17	22	26	30	34
COST	26,93132													

INSERTATS 9

LINIA *24*

DEM.LIN.	21072	ORD.	1	2	3	4	5	6	7	8	9	10	11	12	13
DEM.XRX.	500338	TEMPS	5.47	6.36	6.90	6.14	7.40	9.04	9.96	8.50	10.79	7.24	9.47	6.09	
LNG.LIN.	18.672	CAR.	1700	3331	6065	5939	5832	3868	3317	3684	3406	4311	4815	2503	
LNG.XRX.	399.185	NUS.	7	12	14	13	17	20	25	33	37	39	40	42	44
COST	26,92856														

INSERTATS 7

LINIA *25*

DEM.LIN.	10276	ORD.	1	2	3	4	5	6
DEM.XRX.	500338	TEMPS	6.56	5.70	5.27	7.57	12.20	
LNG.LIN.	7.461	CAR.	3030	5788	6392	4875	2783	
LNG.XRX.	399.185	NUS.	21	23	27	26	30	35
COST	26,92856							

LINIA *26*

DEM.LIN.	3221	ORD.	1	2	3	4
DEM.XRX.	500338	TEMPS	11.00	9.69	12.94	
LNG.LIN.	6.726	CAR.	871	1993	2264	
LNG.XRX.	399.185	NUS.	1	2	3	4
COST	26,92856					

8.1.3.4. Informació sobre la xarxa.

Es un quadre que s'imprimeix al final de cada iteració, el qual informa de l'estat de la xarxa en aquell instant. Consisteix en una taula de doble entrada n^2 itineraris/ n^2 línies, on cada casella conté el nombre de fluxos o parells de nusos, servits pel nombre de línies que indica la fila, les quals s'agrupen en el nombre d'itineraris que indica la columna.

Al mateix temps dóna les sumes marginals, per files (línies) i per columnes (itineraris) de l'esmentada taula, i la demanda absorbida per la xarxa, la no absorbida i la total que correspon a cada un d'aquests grups de fluxos. Una altra informació lligada al nombre de línies (i suposant que pertanyen a un mateix itinerari) és el temps d'espera corresponent, calculat segons la fórmula (2.2.3), vista a la hipòtesi H14).

Finalment, calcula i imprimeix una sèrie de dades globals, relatives a la xarxa:

- cost mitjà (en min.).
- guany percentual (en %).
- guany acumulat (en%).
- interval (en min.).
- longitud de la xarxa (en Km).
- nombre total de línies.
- nombre total de nusos inserits durant la iteració.
- nombre total de nusos suprimits durant la iteració.

Algun d'aquests conceptes requereix aclariment; així si:

- C : cost mitjà fins al moment.
- C_1 : cost per mitjans alternatius (màxim).
- C_2 : cost mínim.
- C_0 : cost mitjà a l'anterior iteració,

aleshores

$$\text{guany percentual} = 100 \frac{C_0 - C}{C_1 - C_2} \quad (8.1.1.)$$

$$\text{guany acumulat} = 100 \frac{C_1 - C}{C_1 - C_2} \quad (8.1.2.)$$

Per index de redundància s'entén el següent:

$$\text{index redund} = \frac{\sum n^2 \text{ fluxos} \times n^2 \text{ línies que els uneixen}}{\sum n^2 \text{ fluxos}} \quad (8.1.3.)$$

Es un index, doncs, de la riquesa de la xarxa.

El llistat correspon a la mateixa passada i mateixa iteració que l'anterior, relatiu a les línies.

ITERACIO 5. XARXA.

NOMBRE DE PARELLS DE NUSOS SEGONS .LINIES
.ITINERARIS QUE LES SATISFAN,

NO. LINIES	NO. ITINERARIS							TOTAL PARELLS NUSOS	NO ABSORB.	DEMANDA ABSORB.	TOTAL	TEMPS ESPERA (MIN.)
	0	1	2	3	4	5	6					
0	253						253	143526			143526	
1		356					356	15012	121183	136195		3,99
2		55	149				204	16681	136864	153546		2,66
3		29	31	58			118	14772	137140	151912		2,00
4		19	5	10	6		40	14424	59649	74073		1,60
5		7	0	0	0	4	11	11655	19582	31238		1,33
6		4	1	0	0	0	5	6021	20293	26315		1,14
7		3	0	0	0	0	3	5403	5633	11036		1,00
TOTAL PARELLS NUSOS	253	473	186	68	6	4	990	227496	500348	727844		

NO ABSORB. 143526 72571 9186 2024 92 93 0 227494
 ABSORB. 313653 131268 49678 4857 887 0 500345

TOTAL 143526 386224 140455 51702 4949 981 0 727839

COST MITJA 26,92856 MIN.
 GUANY PERCENTUAL 0,62118
 GUANY ACUMULAT 79,19475
 INTERVAL 7,98 MIN.
 LONGITUD DE XARXA 399,185 KM.

NOMBRE DE LINIES 26
 NUSOS INSERTATS 11
 NUSOS SUPRIMITS 7
 INDEX REDUNDANCIA 1,463

8.1.3.5. Informació resum de l'algorisme.

Només s'imprimeix al final de l'execució, i no aporta res de nou, sinó que, vol ser un recull d'alguns valors obtinguts al final de cada iteració i que, agrupats, poden donar una idea del desenvolupament del procés.

Les dades relatives a cada iteració que ara es repeteixen són:

- cost mitjà (en min.).
- guany percentual (en %).
- guany acumulat (en%).
- longitud de la xarxa (enKm).
- demanda absorbida.
- nombre de línies.
- nombre total de nusos inserits a la iteració.
- nombre total de nusos suprimits a la iteració.
- index de redondància.

RESUM DE L'ALGORISME

ITERACIO	COST MITJA (M/N.)	GUANY PERCENTUAL	ACUMULAT	LONG.XRX. (KM.)	DEM.ABSORB	NOMBRE LINES INSERTATS	NUSSOS SUPRIMITS	REDUNDANCIA	ITERACIO
	40.68018								
0	26.78232	68.51904	68.51904	363.822	442452	26	204	0	0.739
1	27.67488	5.22589	73.74495	375.034	465430	26	32	9	0.942
2	27.49896	2.16488	75.90982	396.578	483924	26	23	2	1.143
3	27.22215	1.59413	77.50395	395.742	492488	26	22	9	1.283
4	27.03642	1.06961	78.57356	393.784	497663	26	14	6	1.388
5	26.92856	0.62118	79.19475	399.185	500348	26	11	7	1.463
6	26.89717	0.18076	79.37550	409.108	502768	26	8	0	1.568
7	26.87099	0.15079	79.52629	404.539	502917	26	3	5	1.575

8.1.4. FITXES DE DADDES. DESCRIPCIÓ.

Només hi ha 4 formats, corresponents a cada un dels tipus de dades que es llegeixen.

Aquests formats són:

I - de variable entera (I3, 77x)

R - " " real (F10.0, 70x)

M.- de matriu, per fitxa, si és llegeix (13F6.0)

L.- línia (5x,25I3)

A continuació es detalla la taula de la figura 8.1.2, amb les dades a entrar, llur significat i el format de lectura que necessiten.

ORDRE DE FITXA O GRUP FITXES	NOM DE LA VARIABLE	NOM EN EL PROGRAMA	FORMAT	CONDICIONS
1	Nombre de zones	NN	I	≤ 44 altrament s'atura
2	Nombre de busos	AUT	R	
3	Velocitat a peu	V	R	
4	Relació vel. bus/vel. peu	Z	R	
5	Penalització temps espera	PE	R	
6	Equivalent en temps tarifa	E	R	
7	Sensibilitat a la dif. de costos en min ⁻¹	BETA	R	
8	Error relatiu màxim entre costos per deturar el programa	TØ	R	
9	Nombre inicial màxim de nusos/línia.	LINUS	I	
10	Límit geomètric inserció	GEØ	R	
11	Límit geomètric clausura	CØCLA	R	
12	Límit geomètric ruptura	CØRUP	R	
13	Opció de llistat -reduït = 0 -complet = 1	ILLIS	I	
14	Nombre màxim d'iteracions	ITMAE	I	
15	Opció de la xarxa -generada = 0 -donada = 1	IX	I	
16	Matriu de demandes	D(I,K)	M/-	Es llegeixen de cop, en una sola instrucció, si l'entrada és per fitxa, o fila a fila si és per cinta o disc.
17	Matriu de $\left\{ \begin{array}{l} \text{temps} \\ \text{distància} \end{array} \right\}$ en autobús.	A(I,K)	M/-	
18	Matriu de $\left\{ \begin{array}{l} \text{temps} \\ \text{distància} \end{array} \right\}$ a peu mit. alternat.	C(I,K)	M/-	
19	Nombre de línies	MRE	I	Només si IX = 1
20	Descripció de cada línia.	NL(J,NLIN)	L	Altrament, res.

FIG. 8.1.2

8.1.5. ESTRUCTURA INTERNA DEL PROGRAMA.

Aquest paràgraf pretén de descriure la jerarquitització del programa principal i de les 16 subrutines que conté, així com d'explicar l'objectiu de cada una. Les primeres responen exactament a cada un dels 4 mòduls de l'algorisme, o bé són subrutines d'impressió de resultats. Les de nivell inferior, són clarament de servei de les altres. L'estructura s'exposa a la figura 8.1.3.

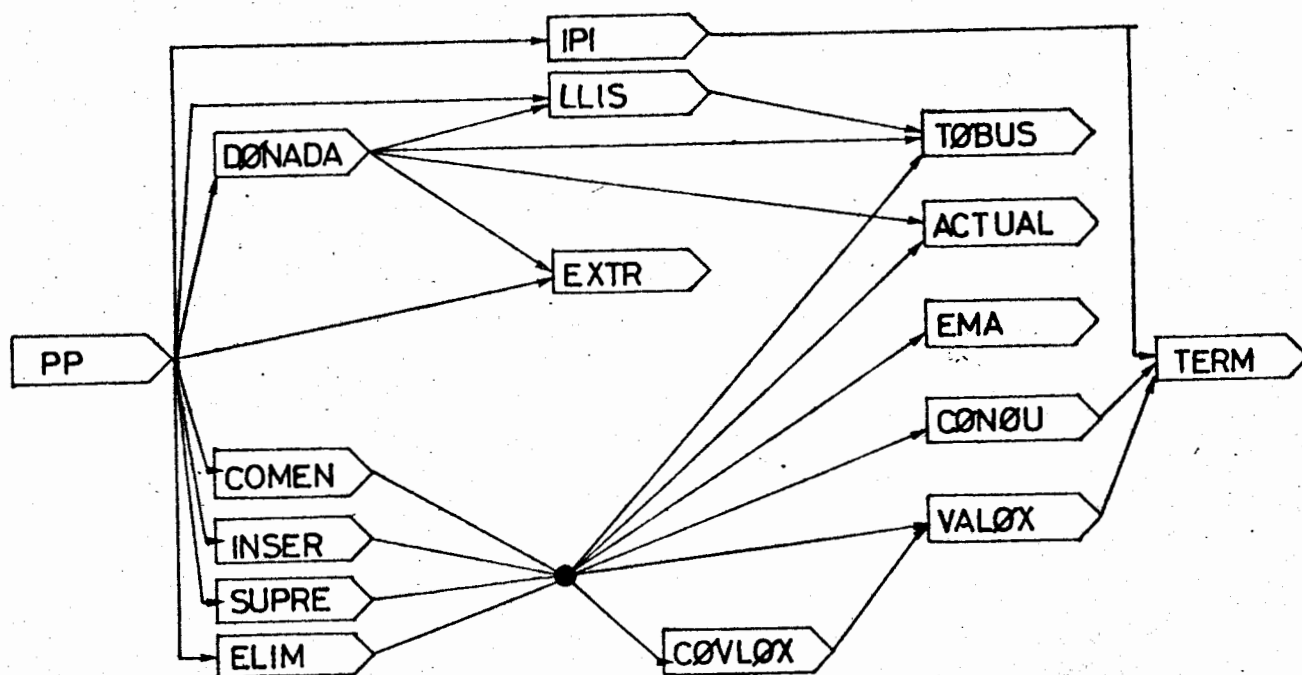


FIG. 8.1.3

8.1.5.1. Block data.

Assigna dos tipus de valor:

- a) d'inicialització, pròpiament dits, tals com:
 - matriu lògica de pertanyença.
 - matriu de nombre d'itineraris.
 - vectors suplementaris per al 2on. i següents itineraris.
 - de temps total de la xarxa.

- b) propis de l'ordenador en què s'executa, i de les dimensions, com ara:
 - nº de les unitats lògiques de la lectura de targetes i de la impressora.
 - nombre màxim permès de zones, de línies, de nusos/línia i d'iteracions.

FACOM BOS2 FORTRAN -740601- V03 L08

ISN * I SOURCE STATEMENT / ERROR MESSAGE *ID, SEQ.

C *****

C ALGORISME PER A LA GENERACIO I/O MILLORA D'UNA XARXA URBANA D'AUTOBUS

C *****

C PARAMETRES DEL PROGRAMA I INICIALIZACIONS.

C *****

1 BLOCK DATA

2 LOGICAL CIRC, CIR, NUS

3 COMMON /ETI O/ NL(25,40), T(25,40), NNI(40), CIRC(40), LINUS,

1 /ETI 1/ D(4,4,4), TA, AUT, NN, NOO, IENT, ISOR

2 /ETI 2/ LI(25), BU(25), CIR, NF, NO, NUS(4,4,40)

5 /ETI 5/ C(4,4,4), NIL(4,4,4)

9 /ETI 9/ NIT, R(9), NLI(9), III(9), YY(9),

A IS(44), KS(900), RS(900), NS(900), NTS

C UNITATS D'ENTRADA/SORTIDA

C DATA IENT, ISOR /5, 6/

C LONGITUD DE XARXA

C DATA TA /0./

C DIMENSIO DE LES Matrius

C DATA NNUM, MREM, MFMAX, /44, 40, 25/15/

C INICIALIZACIO DE LES Matrius DE PERTANYENCA, D'ITINERARIS I DELS POINTERS

C DATA NIL, NUS, IS, KS, NTS / 1936*0, 1760*, FALSE, 44*1, 900*0, 0/

8 END

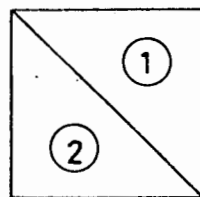
8.1.5.2. Programa principal.

S'ocupa progressivament de tres tasques:

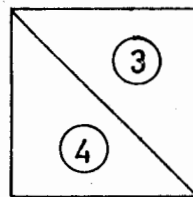
- lectura i preparació de dades.
- control del procés.
- final del programa.

A continuació hom s'ocupa de descriure cada una d'aquestes tasques:

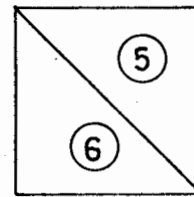
- a) Lectura dels paràmetres descrits anteriorment i preparació d'algun que no es llegeix, sinó que és derivació dels altres com ara YBUS, HAC i GEX.
- b) Lectura i preparació de les matrius D i C i de NIL. Les matrius estan triangularitzades, amb el següent contingut:



D



C



NIL

- ① conté $d_{ik} + d_{ki}$, o demanda de transport.
- ② " q_{ik} , o cost dels camins mínims.
- ③ " p_{ik} , o cost a peu (o per mitjans alternatius).
- ④ " r_{ik}^1 , o temps de recorregut del ler. itinerari.
- ⑤ " s_{ik} , o nombre de itineraris.
- ⑥ " g_{ik}^1 , o nombre de línies del ler. itinerari.

Naturalment, només són dades ① ② i ③ ; al començament, doncs, ④ està inicialitzat a ∞ i ⑤ i ⑥ a zero.

- c) Impressió dels paràmetres del procés, tal com s'ha descrit a 8.1.3.1.
- d) Bucle de generació o 1ª fase de l'algorisme. Es la implementació del seu equivalent, ja explicat en descriure l'estructura de l'algorisme.
Després de generar cada línia crida la rutina d'impressió de línies, i al final de la generació, o sigui, d'aquesta fase, crida la de càlcul de paràmetres relatius a la xarxa.

e) Bucle de revisió, o 2^a fase de l'algorisme. Pròpiament, és un doble bucle. L'interior, correspon a cada línia, amb les corresponents crides a les rutines de supressió i inserció i, eventualment, d'eliminació. L'exterior, és el bucle d'iteració, que conté, a més del que s'acaba d'esmentar, una crida a la rutina d'impressió, una còpia de la 1^a fase, per intentar generar noves línies, i el test de final, funció de les millores obtingudes. Per tant, hi ha tantes iteracions com vegades es recorre el bucle exterior; i hi ha tantes línies com vegades s'ha recorregut el bucle interior en la darrera iteració. Aquest punt és, sens dubte, la clau de tot l'algorisme.

f) El final del programa principal té previst de fer un llistat complet de la xarxa, suposant que fins aleshores hagi estat reduït.

A més s'ocupa del llistat final resum de l'algorisme.

Té previstos els acabaments anormals, en què:

- o bé no s'ha generat cap línia,
- o bé s'arriba al límit d'iteracions.
- o bé s'ha detectat un error en les dades.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C *****

C PROGRAMA PRINCIPAL. CONTROL DEL PROCES.

C *****

```

1 LOGICAL NUS, CIRC, CIR
2 INTEGER*4 IDII
3 COMMON /ZETI 0/ NL(25,40), T(25,40), NNU(40), CIRC(50), LINUS,
   & RNUM, MREM, MFMX, ITMAX
4
5 /ZETI 1/ D(44,44), TA, AUT, NN, NOO, IENT, ISOR
6 /ZETI 2/ LI(25), BU(25), CIR, NF, NA, NP, NUS(44,40)
7 /ZETI 3/ NINS(25), NSUP(25), NNINS, MNSUP
8 /ZETI 4/ MRE, NSINS, NSSUP, YBUS, W, CANT, CINI, CDIF, CM
9 /ZETI 5/ C(44,44), NIL(44,44)
10 /ZETI 6/ DRMAX
11 /ZETI 7/ GEO, GEX, COCLA, CORUP
12 /ZETI 8/ E, PE, HAC
13 /ZETI 10/ IX, ITER, CNN, CMII(15), GPII(15), GALL(15),
14 SALL(15), IDII(15), MHII(15), NINSI(15), NSUPI(15),
15 REDUI(15)

```

C ENTRADA DE PARAMETRES I D'OPCIONES

```

4 READ(IENT,52) NN
5 READ(IENT,54) AUT, V, Z, PE, E, BETA, TO
6 IF (NN.GT.NNUM) GO TO 49
7 READ(IENT,52) LINUS
8 READ(IENT,54) GEO, COCLA, CORUP
9 READ(IENT,52) ILLIS, ITMAE, IX
10 NOO = NN - 1
11 CNN = AN * (NN-1) / 2
12 YBUS = V * Z
13 HAC = 1. / BETA
14 GEX = 1. / SUPT(1. + GEO*GEO)
15 ITMAX = MING(ITMAE,ITMAX)

```

C FORMATS D'ENTRADA

```

16 52 FORMAT(I3, ' 77X)
17 54 FORMAT(F10.0, 70X)
18 56 FORMAT(13F6.0)

```

C PREPARACIO DE LA MATRIU 'D', DEMANDA I TEMPS D'AUTOBUS.

```

19 READ(IENT,56) (DC(I,K),K=1,NN), I=1,NN)
20 READ(IENT,56) (CC(I,K),K=1,NN), I=1,NN)
21 DO 2 I=1,NOO
22 NO = I + 1
23 DO 2 K=NO,NN
24 D(I,K) = DC(I,K) + D(K,I)
25 2 D(K,I) = C(I,K) / YBUS

```

C PREPARACIO DE LA MATRIU 'C'. TEMPS A PEU I TEMPS DEL PRIMER ITINERARI.

```

26 READ(IENT,56) (CCI(K),K=1,NN), I=1,NN)
27 DO 4 I=1,NOO
28 NO = I + 1
29 DO 4 K=NO,NN
30 CCI(K) = CCI(K) / V

```

Z

Z

```

31 * C(K,I) = 1.E+20
C
C CALCUL DELS COSTOS I DE LES DEMANDES.
C
W = 0.
C1 = 0.
C2 = 0.
WP = 0.
DUMAX = 0.
DO 6 I=1,NCO
NO = 1 + I
DO 5 K=NO,NN
DD = D(I,K)
P = C(I,K)
B = D(K,I)
IF(DD.GT.0. .AND. B.GT.DBMAX) DBMAX = B
B = B * E
X = ANLN(P*B) - HAC
S = 1. / (1.7*(B-X) + 1.7*(P-X))
W = W + DD
C1 = C1 + DD * P
C2 = C2 + DD * (B/(B-X) + P/(P-X)) * S
6 WP = WP + DD * S/(R-X)
DO 8 I=1,NN
8 W = W + D(I,I)
C1 = C1 / W
C2 = C2 / W
CINT = C1
CANT = C1
CDIF = C1 - C2
C
C IMPRESSIO DE PARAMETRES
C
28 * WRITE(ISOR,60) NN, AUT, V, YBUS, PE, E, BETA, HAC, W, WP, C1, C2,
1 LINUS, ITHAX, GEO, GEX, COCLA, CORUP
C
C ENTRADA DE LA XARXA, EN EL CAS DE MILLORA
C
59 NSINS = 0
60 NSSUP = 0
61 NNSUP = 0
62 ITER = 0
63 IF(IXIEG.0) GO TO 10
64 * WRITE(ISOR,65)
65 CALL DONADA(LLEIS)
66 * WRITE(ISOR,65)
67 CALL IFT
68 CINT = CM
69 GO TO 18
C
C GENERACIO DE LA XARXA
C
70 *RITE(ISOR,62) ITER
71 DO J= NLIN,1,PREM
72 CIR = .FALSE.
73 CALL CONEN(CIR,NLIN)
74 IF(CIR.EQ.0) GO TO 16
75 CALL INSEK(LIN,NLIN)
76 IF(INDINE.0 .AND. NP.LT.LINUS) GO TO 12
77 NSINS = NSINS + NrlINS
78 CALL LLIS(LIN,NLIN)
  
```


ISN * 1 SOURCE STATEMENT / ERROR MESSAGE *ID. SEQ.

79 14 CALL EXTR(1,NLIN)
 80 16 MRE = NLIN - 1
 81 IF(MRE.LE.0) GO TO 50
 82 WRITE(ISOR,64) ITER
 83 CALL IPI
 84 18 CANT = CM

C REVISIO DE LA XARXA
 C DO 42 ITER=1,ITMAX

C PROVA DE COVLOX

COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX
 COVLOX

86 *WRITE(9,1500)
 87 1500 FORMAT(1,'10X',10X,'PROVA DE COVLOX' / ' ',10X,15(' '), //
 10X,1 DB PENDING AUGMENT, //)

88 XX = -32.
 89 DO 19 ICG=1,61
 90 XX = XX + 2.
 91 IF(CX.EQ.0.) GO TO 19
 92 YY = COVLOX(0,XX)
 93 ZZ = XX * YY
 94 *WRITE(9,1501) XX, YY, ZZ
 95 1501 FORMAT(10X, F10.0, F11.5, F11.2)

19 CONTINUE
 96 WRITE(ISOR,62) ITER
 97 LIMIT = MINO(LINUS*ITER, MFMX)
 98 NINS = 0
 99 NSSUP = 0
 100 DO 32 NLIN=MRE
 101 IF(PREV.GE.MRE) GO TO 40
 102 MINS = 0
 103 NSSUP = 0
 104 CALL EXTR(2,NLIN)

22 IF(MP.EQ.2) GO TO 24
 CALL SUPRE(IND,NLIN)
 107 IF(LIND) 23, 28, 22
 24 CALL ELIM(IND,NLIN)
 109 IF(IND.EQ.0) GO TO 28
 110

C ELIMINACIO D'UNA LINIA

111 CALL LLIS(ILLIS,NLIN)
 112 NSSUP = NSSUP + NSSUP
 113 NNSUP = C
 114 MRE = MRE - 1
 115 IF(CALIN.EQ.MRE+1) GO TO 34
 116 CALL EXTR(2,MRE+1)
 117 DO 26 J=1,NF
 118 LIJ = LI(J)

26 NUS(LIJ,NLIN) = .TRUE,
 GO TO 22
 28 CALL INSER(IND,NLIN)
 121 IF(IND.EQ.0 .AND. NF.LI.LIMIT) GO TO 28
 122 NINS = NINS + NINS
 123 NSSUP = NSSUP + NSSUP
 124 CALL LLIS(ILLIS,NLIN)
 125 32 CALL EXTR(1,NLIN)
 126

C POSSIBLE CREACIO DE NOVES LINES

Z

FACOM 90S2 FORTRAN -740601- V03 L08 FTMAIN 75.09.25 PAGE 4
*ID. SEQ. SOURCE STATEMENT / ERROR MESSAGE

```

127 34 IF(PE.EE.MREH) GO TO 40
128 MNSUP = 0
129 NDU = MRE + 1
130 DO 38 NLINE=NDU,MREM
131 CTR = .FALSE.
132 CALL COMER(IND,NLIN)
133 IF(IND.EW.O) GO TO 40
134 MRE = MRE + 1
135 CALL INSER(IND,NLIN)
136 IF(IND.NE.O .AND. NF.LT.LIMIT) GO TO 36
137 NSINS = NSINS + NNINS
138 IF(NHINS.NE.O .OR. NNSUP.NE.O) NREV = 0
139 IF(NHINS.EG.O .AND. NNSUP.EG.O) NREV = NREV + 1
140 CALL LLIS(ILLIS,NLIN)
141 CALL EXTR(1,NLIN)
142 40 WRITE(ISOR,64) ITER
143 CALL IPI
144 IF(ABS(CM-CPT) .LE. TO*CM) GO TO 44
145 42 CANT = CM
146 WRITE(ISOR,65)
C POSSIBLE LLISTAT COMPLET
C
147 44 IF(LLIS.EW.O) GO TO 47
148 WRITE(ISOR,62) ITER
149 DO 46 NLINE=1,MRE
150 CALL EXTR(2,NLINE)
151 46 CALL LLIS(1,NLINE)
C RESUM DE L'ALGORISME
C
152 47 ITMAX = ITER + 1
153 WRITE(ISOR,66)
154 IF(IX.EW.O) WRITE(ISOR,70) CINI
155 DO 49 I=1,ITMAX
156 ITER = I - 1
157 48 WRITE(ISOR,72) ITER, GMTI(1), OPTI(1), GATI(1), SXI(1), IDIT(1),
158 MRLI(1), NINSI(1), NSUPI(1), REDUCI(1), ITER
159 STOP
160 C FINALS ANORMALS DE L'ALGORISME
161 49 WRITE(ISOR,74) NN, NNUM
162 STOP
C FORMATS DE SORTIDA
C
163 60 FORMAT(1, '///' PARAMETRES DEL PROCES' / , 21('*)
164 1 31A, 'NOMBRE DE ZONES' / , 17
165 2 31A, 'NOMBRE DE BUSES' / , 8.0
166 3 31A, 'VELOCITAT A PEU' / , 511.3, 3X, 'M./MIN.' /
167 4 31A, 'VELOCITAT DEL BUS' / , 511.3, 3X, 'M./MIN.' /
168 5 31A, 'OPERALITZACIO DE L'ESPERA' / , 8.0
169 6 31A, 'EQUIVALENT DE LA TARIFA' / , 8.0, 6X, 'MIN.' /
170 7 31A, 'SENSIBILITAT A LA DIF. DE COSTS' / , 510.2
171 8 31A, 'DIFERENCIA AMB EL COST MINIM' / , 510.2, 4X, 'MIN.' /
172 9 31A, 'POBLACIO ACTIVA TOTAL' / , 8.0
173 A 31A, 'POBLACIO POTENCIAL ABSORBIBLE' / , 8.0
174 B 31A, 'COST INICIAL' / , 513.5, 1X, 'MIN.' /

```

Z

```

159 * I SOURCE STATEMENT / ERROR MESSAGE *ID. SEQ.
C 31X, 'COST MINIM' //
D 31X, 'NOMBRE MAXIM INICIAL DE NUSOS/LINIA', 'F13.5, 1X, 'MIN.' //
E 22X, 'MAXIM D'ITERACIONS' //
F 31X, 'INSECCIO A L'EXTREM (HIPERBOLIC)', 'F10.2 //
G 31X, 'CLAUSSURA' //
H 31X, 'RUPTURA' //
I 31X, 'ITERACIO', '13, 'LINES', ' / 1X, 20('*)' //
62 FORMAT(1, ' / ' // 'ITERACIO', '13, 'LINES', ' / 1X, 20('*)' //
64 FORMAT(1, ' / ' // 'ITERACIO', '13, 'XARXA DONADA', ' / 1X, 20('*)' //
65 FORMAT(1, ' / ' // 'XARXA DONADA', ' / 1X, 13('*)' //
66 FORMAT(1, ' / ' // 'RESUM DE L'ALGORISME', ' / 1X, 20('*)' //
68 FORMAT(1, ' / ' // 'RESUM DE L'ALGORISME', ' / 1X, 20('*)' //
1 38X, 'QUANY', '37X, 'NOMBRE', '8X, 'NUSOS', '14X, 'INDEX' //
2 8X, 'ITERACIO COST MITJA PERCENTUAL ACUMULAT LONG.XRX. //
3 22X, 'LINES INSERTATS SUPRIMITS REDUNDANCIA ITERACIO / //
4 22X, '(MIN.)', '31X, '(CM)' //
70 FORMAT(1, '16X, F12.5)
72 FORMAT(1, '6X, 110, 3, 12.5, F12.3, 112, 3, 110, F12.3, 110)
74 FORMAT(1, ' / ' // 'NOMBRE DE NUSOS EXCESSIU.(', '13, ')'.EL LIMIT ES',
1 13, 'EL PROGRAMA S'ATURADA.')
76 FORMAT(1, 'OFI DE L'ALGORISME.NO S'HA POGUT GENERAR CAP LINIA.')
END

```

8.1.5.3. Subrutina DONADA.

Només es crida quan l'algorisme treballa en l'opció de revisió d'una xarxa donada. Aleshores, en lloc d'executar-se la 1ª fase, s'activa la rutina DONADA.

Té una estructura de bucle, on cada un dels passos que ara es descriuen es fa per a cada línia.

Té com a missió, per tant:

- a) Llegir la xarxa donada, tot fent algunes comprovacions respecte dels nusos, o sigui $[X_{ih}]$
- b) Construir la fila corresponent de la matriu $[t_{ih}^t]$ o matriu de temps de recorregut per una línia, a partir d' $[x_{ih}^x]$ i de $[t_{ih}^t]$
Igualment, pel que fa a la matriu de pertanyença $[q_{ij}]$.
- c) Actualitzar els costos i identificar l'itinerari al qual pertany la línia, i adés fer-hi les modificacions corresponents, adés crear-ne un de nou.
- d) Llistar la xarxa, línia per línia, segons el nivell d'informació prefixat, un cop ja s'ha incorporat tota la xarxa.

Z

Z

ISN * I SOURCE STATEMENT / ERROR MESSAGE *ID, SEQ. PAGE 1

FACOM BUS2 FORTRAN -740601- V03 L08

```

C *****
C ENTRADA D'UNA XARXA INICIAL
C *****
C
1 SUBROUTINE DONADA (ILLIS)
2 LOGICAL NUS, CIRC, CIR
3 COMMON /ETI 0/ NL(25,40), T(25,40), NNU(40), CIRC(40), LINUS,
4 NNUM, MREH, MEMAX, ITMAX
5 /ETI 1/ DL(4,44), TA, AUT, NN, NOO, IENT, ISOR
6 /ETI 2/ LI(25), BU(25), CIR, NF, NP, NUS(4,40)
7 /ETI 3/ NINS(25), NSUP(25), NNINS, NNSUP
8 /ETI 4/ MRE, NSINS, NSSUP, YBUS, W, CANT, CINI, CDIF, CM
C
C ENTRADA DE LINIES I COMPROVACIONS INICIALS
C
4 READ(1,2) MRE
5 IF(MRE.GT.MREH) GO TO 80
6 DO 16 NLI=1,MRE
7 READ(1,3) (LI(J), J=1,MFMAX)
8 CIR = .FALSE.
9 DO 6 L=1,MFMAX
10 I = LI(L)
11 IF(I.GT.NN) GO TO 82
12 IF(I.LE.0) GO TO 2
13 NF = L - 1
14 GO TO 8
15 2 IF(L.EQ.1) GO TO 6
16 MO = L - 1
17 DO 4 N=1,MO
18 IF(L.NE.LI(N)) GO TO 4
19 IF(LI(L+1).NE.0 .AND. L.NE.MNUM) GO TO 84
20 CIR = .TRUE.
21 NF = L
22 GO TO 8
23 4 CONTINUE
24 6 CONTINUE
25 NF = MNUM
26 N0 = NF - 1
27 NP = NF - 2
28 IF(NF.GT.LINUS) LINUS = NF
C
C VECTOR DE TEMPS (BU) I MATRIU DE PERTANYENCA (NUS)
C
29 BU(1) = 0.
30 DO 10 L=1,NG
31 IO = LI(L)
32 KO = LI(L+1)
33 I = MINO(IO,KO)
34 K = MAXO(IO,KO)
35 BU(L+1) = BU(L) + DCK(I)
36 10 NUS(IO,NLIH) = .TRUE.
37 IO = LI(NF)
38 NUS(IO,NLIN) = .TRUE.
39 TA = TA + BU(NF)
C
C ACTUALITZACIO DE COSTOS
C
40 MF = NF

```

*ID. SEQ.

SOURCE STATEMENT / ERROR MESSAGE

ISN

```

41 M0 = N0
42 IF(.NOT.CIR) GO TO 12
43 MF = NF - 1
44 M0 = N0 - 1
45 DO 14 L=1,M0
46 I = L(L)
47 MS = L + 1
48 DO 14 M=MS,MF
49 B = TOBUS(L,M)
50 14 CALL ACTUAL(1,I,L1(M),1,E+20,B)

```

C EMMAGATZEMAMENT

```

51 16 CALL EXTR(1,NLIN)

```

C INFORMACIO SOBRE LES LINES

```

52 DO 18 NLIN=1,MRE
53 CALL EXTR(2,NLIN)
54 18 CALL LLIS(LLIS,NLIN)
55 RETURN

```

C FORMATS

```

56 52 FORMAT(13, 77X)
57 54 FORMAT(5X, 2513)

```

C MISSATGES D'ERROR

```

58 80 WRITE(ISOR,90) MRE, MREM
59 STOP
60 82 WRITE(ISOR,92) L, NLIN, I, NN
61 STOP
62 84 WRITE(ISOR,94) M, L, NLIN, I
63 STOP

```

90 FORMAT('1' // 'ONOMBRE DE LINES EXCESSIU,(', I3, ')', EL LIMIT ES',

1 I3, ', EL PROGRAMA S'ATURA,')

92 FORMAT('1' // 'OEL NUS', I3, '-E DE LA LINIA', I3, ' (', I4, ') ES

1 MES GRAN QUE EL NOMBRE TOTAL DE NUSOS. (', I2, ')', EL PROGRAMA S'AT

ZURA,')

94 FORMAT('1' // 'OELS NUSOS', I3, '-E I', I3, '-E DE LA LINIA', I3,

1', (', I4, ') SON REPETITS SENSE SER EXTREMS,')

2 'OLA LINIA, DONCS, ES CREUA A SI MATEIXA,')

3 'OEL PROGRAMA S'ATURA,')

67 END

8.1.5.4. Subrutina COMEN.

Implementa exactament el mòdul que duu el seu mateix nom.

Per tant, els 2 graus de llibertat corresponents al parell de nusos en què aleshores s'ha parlat, corresponen a un niu doble de bucles.

- a) La 1^a part, doncs, és aquest bucle doble, on per a cada parell es fa el còmput del guany que representa la creació d'una nova línia, i es compara amb els altres. Si no hi ha cap parell amb guany positiu, el control es retorna al programa principal; altrament, es va a la 2^a part.
- b) Fase d'actualització: Com que la xarxa té una línia nova, modifica els valors de la llista següent, que han canviat:
 - T temps total de recorregut de la xarxa.
 - X ja que a la xarxa se li ha d'afegir una línia.
 - [t_{ih}] temps de recorregut de la nova línia.
 - [a_{ij}] matriu de pertanyença.

FACOM BOS2 FORTRAN -740601- V03 L08

*ID. SEQ.

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE

C *****

C CONENCAMENT D'UNA LINIA, AMB 2 NUSOS INICIALS

C *****

C SUBROUTINE COMEN(INO, NLIN)

LOGICAL CIR, NUS

COMMON /ETI 1/ D(AA,AA), TA, AUT, NN, NOO, IENT, ISOR

2 /ETI 2/ LI(25), BU(25), CIR, NF, NG, NP, NUS(AA,40)

3 /ETI 3/ NINS(25), NSUP(25), NNINS, NNSUP

6 /ETI 6/ DBMAX

DATA EXT /0.5/

IND = 0

RMAX = 0.

C COEFFICIENT COST/LONGITUD PER A LA XARXA

C CO = COVLOX(CO, EXT*DBMAX)

C BALANC

C *****

DO 102 I=1,NOO

NO = I + 1

DO 102 K=NO,NN

P = D(I,K)

IF(P.LE.0.) GO TO 102

B = D(I,I)

R = P * (CONOU(I,I,K,B,I-E+20,B) - VALDX(I,K,B)) + CO * B

IF(R.LT.RMAX) GO TO 102

RMAX = R

BOPT = B

IOPT = I

KOPT = K

102 CONTINUE

C COMPARACIO

C IF(RMAX.LE.0.) RETURN

C ACTUALITZACIO

C *****

TA = TA + BOPT

CALL ACTUAL(I, IOPT, KOPT, O., BOPT)

LI(1) = IOPT

LI(2) = KOPT

BU(1) = 0.

BU(2) = BOPT

NUS(IOPT,NLIN) = .TRUE.

NUS(KOPT,NLIN) = .TRUE.

NNINS = 2

NINS(1) = IOPT

NINS(2) = KOPT

NF = 2

NG = 1

NP = 0

IND = 1

RETURN

37

*ID. SEQ.

SOURCE STATEMENT / ERROR MESSAGE

ISN

38

END

8.1.5.5. Subrutina INSER.

Es correspon amb el mòdul INSER i, en essència, té les mateixes 2 parts que el seu precedent més una intermèdia, relativa a la clausura.

Prèviament a les 3 parts, calcula el coeficient cost/temps per a la resta de la xarxa, valor que es farà servir en cada prova de nus.

- a) La 1ª part, de recerca del conjunt buit-nus òptims, s'estructura a base de 2 bucles. L'exterior, fa referència al buit on inserir el nus, i l'interior al nus inserit. Els càlculs que successivament fa són:
- càlcul del màxim augment de recorregut permès, en funció del buit.
 - test de limitació geomètrica per al nus en qüestió.
En cas que no caigui dintre de l'el·lipse o de la hipèrbole, segons el cas, es passa al nus següent.
 - càlcul del guany per al nus inserit, o sigui φ_2 .
 - pèrdua per a la resta de la línia, φ_1 .

Els dos darrers punts estan subdividits segons si es tracta d'una línia dreta o circular, ja que el mètode de càlcul no és idèntic en un cas que en l'altre.

- pèrdua per a la resta de la xarxa φ_x , si té sentit, és a dir, si hi ha un augment efectiu del temps de recorregut.

- b) En cas de línia dreta i que, a més, acompleixi el test de limitació adient, estudia el guany en cas de clausura. Si és superior al màxim obtingut en la inserció normal, la línia es clou; altrament, s'insereix el nus escollit.
Dintre d'aquesta part, hi ha compresa l'actualització en cas de clausura; a part de les varibles ja esmentades en parlar de la rutina COMEN, en aquest cas, s'afegeix un nus al final, idèntic al inicial, que és l'identificador de la línia circular.
- c) La darrera part s'ocupa de l'actualització en el cas general. Al seu torn està subdividida en dues parts, segons si la inserció s'ha fet en una línia dreta o circular; en aquest darrer cas, poden canviar els temps de recorregut per a qualsevol flux que empri la línia, mentre que si la línia és dreta, el canvi només pot esdevenir-se si el flux travessa l'indret on s'ha inserit el nus.
En canvi l'actualització de vectors inherents a la línia és comuna a ambdues possibilitats.

*ID, SEQ.

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE

```

C *****
C INSERCIO D'UN NUS EN UNA LINIA
C *****
C
1 SUBROUTINE INSER(IND, NLIN)
2 LOGICAL CIR, NUS
3 COMMON /ETI 1/ D(44,44), TA, AUT, NN, NOO, IENT, ISOR
4 /ETI 2/ LI(25), BU(25), CIR, NF, NO, NP, NUS(44,40)
5 /ETI 3/ NINS(25), NSUP(25), NNINS, NNSUP
6 /ETI 7/ GEO, GEX, COCLA, CORUP
7 DATA EXT /0,2/
8 IND = 0
9 RMAX = 0.
10 BF = BUCNF)
11 NO = NF + 1
12 IF(.NOT. CIR) MF = NF
13 IF( CIR) MF = MF - 1
14
15 C MAXIM AUGMENT PERMES
16 C
17 DBMAX = 0.
18 DO 202 L=1,NO
19 DB = TOBUS(L,1)
20 IF(DB.GT.DBMAX) DBMAX = DB
21 202 CONTINUE
22
23 C COEFFICIENT COST/LONGITUD PER A LA XARXA
24 C
25 CO = COVLOX(NLIN, EXT*GEO*DBMAX)
26
27 C MAXIM AUGMENT PERMES, SEGONS, EL BUIT ON S'INSERTA
28 DO 251 N=1,NO
29 NA = N - 1
30 IF(N.GT.1) BA = BUCNA)
31 BN = 8 (N)
32 IF(N.EQ.1) OR, N.EQ.NO) GO TO 204
33 TB = BN - BA
34 QLIM = GEO * TB
35 GO TO 210
36
37 204 IF(CIR) GO TO 251
38 IF(N.EQ.1) TB = TOBUS(1,2)
39 IF(N.EQ.NO) TB = TOBUS(NO,NF)
40 QLIM = GEX * TB
41
42 C TEST DE LIMITACIO GEOMETRICA I DE NO PERTANYENCA A LA LINIA, PER AL NOU NUS
43 C
44 210 DO 250 J=1,NN
45 DO 212 L=1,NF
46 IF(J.EQ.LI(L)) GO TO 250
47 212 CONTINUE
48 Q1 = 0.
49 IF(N.NE.1) Q1 = EMA(2,J,LI(NA))
50 Q2 = 0.
51 IF(N.NE.NO) Q2 = EMA(2,J,LI(N))
52 Q = Q1 + Q2
53 IF(N.NE.1 .AND. N.NE.NO) GO TO 218
54 IF(N.EQ.1) Q5 = EMA(2,J,LI(2))
55 IF(N.EQ.NO) Q5 = EMA(2,J,LI(NO))
56 IF((Q5-Q).LT.QLIM) GO TO 250

```

*ID. SEQ.

SOURCE STATEMENT / ERROR MESSAGE

ISN

42	DB = 0
43	GO TO 220
44	218 IF(0.GT.0LIM) GO TO 250
45	DB = 0 - TB
46	220 R = 0.
	C GUANY PER AL NUS INSERTAT EN UNA LÍNIA DRETA
47	IF(CIR) GO TO 224
48	DO 222 L=1,NF
49	I = LI(L)
50	P = EMA(1,J,I)
51	IF(P.LE.0.) GO TO 222
52	IF(L.L.NA) B = TOBUS(L,NA) + 01
53	IF(L.E.NA) B = 01
54	IF(L.EG.N) B = 02
55	IF(L.GT.N) B = TOBUS(N,L) + 02
56	R = R + P * CONOU(1,J,I,DB,1,E+20,B)
57	222 CONTINUE
58	GO TO 228
	C GUANY PER AL NUS INSERTAT EN UNA LÍNIA CIRCULAR.
59	224 BA1 = BA + 01
60	BN2 = BF - BN + 02
61	DO 226 L=1,N0
62	I = LI(L)
63	P = EMA(1,J,I)
64	IF(P.LE.0.) GO TO 226
65	IF(L.L.NA) B = AMINI(BA1-BU(L) , BN2+BU(L))
66	IF(L.EG.NA) B = AMINI(01 , BN2+BA)
67	IF(L.EW.N) B = AMINI(BF-BN+BA+01 , 02)
68	IF(L.G.N) B = AMINI(BF+BA1-B(L) , 02-BN+BU(L))
69	R = R + P * CONOU(1,J,I,D5,1,E+20,E)
70	226 CONTINUE
	C PERDUA PER A LA RESTA DE LA LÍNIA. (LÍNIA DRETA).
71	228 IF(DB.LE.0.) GO TO 248
72	IF(CIR) GO TO 238
73	IF(N.E3.1 .OR. N.E9.NO) GO TO 234
74	IL = 0
75	DO 232 L=1,N0
76	IF(L.GE.N) IL = 1
77	I = LI(L)
78	MS = L + 1
79	IM = 0
80	DO 232 M=MS,NF
81	IF(M.GE.N) IM = 1
82	K = LI(M)
83	P = EMA(1,I,K)
84	IF(P.LE.0.) GO TO 232
85	B = TO.US(L,M)
86	IF(IL.EE.IM) R = R + P * CONOU(2,I,K,DB,9,B+DB)
87	IF(IL.EE.IM) R = R + P * VALOX(1,K,DB)
88	232 CONTINUE
89	GO TO 244
	C PERDUA PER A LA RESTA DE LA LÍNIA. QUAN S'INSERTA UN NUS EXTREH.
90	234 DO 236 L=1,N0

*ID, SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

```

91 I = LI(L)
92 MS = L + 1
93 DO 236 M=MS,NF
94 K = LI(M)
95 P = EMAC(I,K)
96 IF(P.GT.O.) R = R + P * VALOX(I,K,DB)
97 236 CONTINUE
98 GO TO 244

C
C PERDUA PER A LA RESTA DE LA LINIA. (LINIA CIRCULAR).
C
238 IL = 0
DO 242 L=1,MP
IF(L.GE.N) IL = 1
I = LI(L)
BL = BU(L)
MS = L + 1
IM = 0
DO 242 M=MS,NF
IF(M.GE.N) IM = 1
K = LI(M)
P = EMAC(I,K)
IF(P.LE.O.) GO TO 242
B = BU(M) - BL
S = BF - B
IF(IL.EQ.IM.AND.B.LE.S.OR. IL.NE.IM.AND.B.GE.S) GO TO 240
IF(IL.EQ.IM) U = AMINI(B,S,DB)
IF(IL.NE.IM) U = AMINI(B,DB,S)
R = R + P * CONOU(2,I,K,DB,AMINI(B,S),U)
GO TO 242
240 R = R + P * VALOX(I,K,DB)
242 CONTINUE

C
C PERDUA PER A LA RESTA DE LA XARXA
C
244 R = R + CO * DB
DO 246 L=1,MF
I = LI(L)
246 R = R - EMAC(I,J,I) * VALOX(J,I,DB)

C COMPANACIO AMB ANTERIORS
C
248 IF(R.LT.RMAX) GO TO 250
RMAX = R
DROPT = DB
NOPT = N
JOPT = J
@1OPT = @1
@2OPT = @2
250 CONTINUE
251 CONTINUE

C CLAUDURA D'UNA LINIA DRETA
C *****
C TEST PREVI
C
IF(CIR .OR. NF.LE.2) GO TO 258
L1 = LI(1)
L2 = LI(2)
L3 = LI(3)
L4 = LI(4)

```

*ID. SEQ.

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE

```

137 LF = L(NF)
138 B = EMA(2,L1,LF)
139 IF( B.GT.COCLA*DBMAX
1 .AND. EMA(2,L2,LF)=B .LT. GEX*TOBUS(1,2)
2 .AND. EMA(2,L0,L1)=B .LT. GEX*TOBUS(N0,NF)) GO TO 256
140 BLIM = 0.5 * (BF+B)
141 R = 0.

```

C GUANY PER ALS NUSOS DE LA LINIA

```

142 DO 254 L=1,N0
143 I = LI(L)
144 BB = BU(L) * BF + B
145 MS = L + 1
146 DO 254 M=MS,NF
147 K = LI(M)
148 P = EM (1,I,K)
149 IF(P.LE.0.) GO TO 254
150 S = TO.US(L,M)
151 IF(S.GE.BLIM) R = R + P * CONOU(2,I,K,B*S,BB=BU(M))
152 IF(S.L.BLIM) R = R + P * VALOX(1,K,B)
153 254 CONTINUE

```

C PERDUA PER A LA RESTA DE LA XARXA

C R = R + CO * B

C COMPARACIO CLAUSURA=INSERCIO.

C IF(R.LT.RMAX) GO TO 256

C ACTUALITZACIO EN CAS DE CLAUSURA D'UNA LINIA DRETA

C *****

C TA = TA + B

C ACTUALITZACIO DE COSTOS

```

157 DO 256 L=1,N0
158 I = LI(L)
159 BB = BU(L) * BF + B
160 MS = L + 1
161 DO 256 M=MS,NF
162 S = TO.US(L,M)
163 U = BB - BU(M)
164 IF(U.LT.S) CALL ACTUAL(2,I,LI(M),S,U)
165 256 CONTINUE

```

C ACTUALITZACIO DE VECTORS

```

166 NF = NF + 1
167 N0 = N0 + 1
168 NP = NP + 1
169 CIR = .TRUE.
170 LI(NF) = LI(1)
171 BU(NF) = BU(N0) + B
172 NNINS = NNINS + 1
173 NINS(NNINS) = LI(1)
174 IND = 1
175 RETURN

```

ISN

* 1

SOURCE STATEMENT / ERROR MESSAGE

C ACTUALITZACIO EN CAS D'INSERCIO

C *****

176 258 IF(RMAX.LE.U.) RETURN

177 TA = TA + DBOPT

178 NA = NOPT - 1

C ACTUALITZACIO DE COSTOS PER AL NUS INSERTAT EN UNA LINIA DRETA.

179 IF(CIR) GO TO 264

180 DO 260 L=1,NF

181 IF(LT.NA) B = TOBUS(L,NA) + @1OPT

182 IF(L.E.NA) B = @1OPT

183 IF(L.EG.NOPT) B = @2OPT

184 IF(L.G.NOPT) B = TOBUS(NOPT,L) + @2OPT

185 260 CALL ACTUAL(1,JOPT,LI(L), 1.E+20*B)

C ACTUALITZACIO DE COSTOS PER A LA RESTA DE LA LINIA. (DRETA).

C IF(NOPT.EG.1.OR.NOPT.EG.NO.OR.DBOPT.LE.0.) GO TO 280

186 DO 262 L=1,NA

187 I = LI(L)

188 DO 262 M=NOPT,NF

189 B = TORUS(L,M)

190 262 CALL ACTUAL(2,I,LI(M), B,@DBOPT)

191 GO TO 280

C ACTUALITZACIO DE COSTOS PER AL NUS INSERTAT EN UNA LINIA CIRCULAR

193 264 BA = BU(NA)

194 BN = BU(NOPT)

195 BAI = BA + @1OPT

196 BN2 = BF - BN + @2OPT

197 DO 266 L=1,N0

198 IF(LT.NA) B = AMIN1(BAI-BU(L) , BN2+BU(L))

199 IF(L.E.NA) B = AMIN1(@1OPT , BN2+BA)

200 IF(L.EG.NOPT) B = AMIN1(BF-BN+BA+@1OPT, @2OPT

201 IF(L.G.NOPT) B = AMIN1(BF+BA-BU(L) , @2OPT-BN+BU(L))

202 266 CALL ACTUAL(1,JOPT,LI(L),1.E+20*B)

C ACTUALITZACIO DE COSTOS PER A LA RESTA DE LA LINIA. (CIRCULAR).

203 IF(DBOPT.LE.0.) GO TO 280

204 IL = 0

205 DO 278 L=1,NP

206 IF(L.GE.NOPT) IL = 1

207 BL = B/(L)

208 I = LI(L)

209 MS = L + 1

210 IM = 0

211 DO 278 M=MS,N0

212 IF(M.GE.NOPT) IM = 1

213 B = TOBUS(L,M)

214 U = BU(M) - BL

215 V = BF - U

216 IF(L.E.IM) U = U + DBOPT

217 IF(L.EG.IM) V = V + DBOPT

218 U = AMIN1(U,V)

219 IF(LT.U) CALL ACTUAL(2,I,LI(M), B,U)

220 278 CONTINUE

C

INSER

FACOM BOS2 FORTRAN -740601- V03 L08

*ID. SEQ.

SOURCE STATEMENT / ERROR MESSAGE

ISN * 1

C ACTUALIZACIO DE VECTORS

```

221 280 IF(NOPT,NE,NO) GO TO 282
222 BU(NO) = BU(NF) + DBOPT
223 GO TO 288
224 282 N = NO
225 284 LI(N) = LI(N-1)
226 BU(N) = BU(N-1) + DBOPT
227 IF(N.LE,NOPT+1) GO TO 286
228 N = N - 1
229 GO TO 284
230 286 IF(NOPT,NE,1) BU(NOPT) = BU(NOPT-1) + @IOPT
231 288 LI(NOPT) = JOPT
232 NF = NF + 1
233 NO = NO + 1
234 NP = NP + 1
235 NUS(JOPT,NLIN) = .TRUE.
236 NNINS = NNINS + 1
237 NINS(NNINS) = JOPT
238 IND = 1
239 RETURN
240 END

```

8

8.1.5.6. Subrutina SUPRE.

Com que es tracta d'un mòdul negatiu només té un grau de llibertat; traduït a termes informàtics, vol dir que hi ha un sol bucle, relatiu al nus a suprimir.

Té una estructura ben bé paral·lela a la d'INSER, amb tres parts diferenciades.

- a) La primera part estudia quin és el nus que obté un guany òptim, cas que n'hi hagi algun. Consta dels següents punts:
 - determinació de la disminució del temps de recorregut. Si és zero, es passa al nus següent, sense examinar res més.
 - pèrdua per al nus suprimit.
 - guany per a la resta de la línia, amb segregació de mètodes segons si la línia és dreta o circular; la distinció té la mateixa raó de ser que la vista en parlar d'INSER.
 - guany per a la resta de la xarxa.

- b) La 2ª part s'ocupa de la ruptura:
 - analitza, nus per nus, si el trencament per aquell indret està d'acord amb el test de limitació geomètrica. Només en cas afirmatiu continua el còmput per a aquell nus.
 - calcula la pèrdua per a la línia i el guany per a la resta de la xarxa.
 - si el guany total és més gran que l'obtingut en la primera fase, es consuma el trencament de la línia, és a dir, s'actualitza.
 - actualització en cas de ruptura; comprèn un nou ordre dels nusos, ja que, en trencar-se pel mig, els nusos constituents adopten una nova ordenació. Tots els altres vectors i matrius a retocar són els mateixos que fins ara.

- c) La 3ª part és l'actualització quan es duu a terme la supressió. Subsisteix la dicotomia entre línia dreta i circular, a l'hora de rectificar els costos de la resta de la línia.

Z

*ID. SEQ.

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE

```

C*****
C
C SUPRESSIO D'UN NUS EN UNA LINIA
C*****
C
1 SUBROUTINE SUPRE(IND, MLIN)
2 LOGICAL CIR, NUS
3 COMMON /ETI 1/ D(4,44), TA, AUT, NN, NOO, IENT, ISOR
4 /ETI 2/ LI(25), BU(25), CIR, NF, NO, NP, NUS(4,40)
5 /ETI 3/ NINS(25), NSUP(25), NNINS, NNSUP
6 /ETI 4/ GEO, GEX, COCLA, CORUP
7 /TREP / LIR(25), BUR(25)
8 DATA EXT /0.5/
9 IND = 0
10 RMAX = 0.
11 BF = BU(NF)
12
C
C MAXIMA DISMINUCIO POSSIBLE DE LONGITUD DE LINIA
C
8 DBMAX = 0.
9 DO 302 N=2,N0
10 DB = T(BUS(N-1, N*1) - EMA(2, LI(N-1), LI(N*1)))
11 IF (DB.GT.DBMAX) DBMAX = DB
12 302 CONTINUE
13 IF(CIR) GO TO 304
14 DB = A*AXI(TOBUS(1,2), TOBUS(N0,NF))
15 IF (DB.GT.DBMAX) DBMAX = DB
16 GO TO 308
17 304 DB = TCBUS(N0,NF) + TOBUS(1,2) - EMA(2, LI(N0), LI(2))
18 IF (DB.GT.DBMAX) DBMAX = DB
19 DO 306 N=1,N0
20 DB = TOBUS(N, N*1)
21 IF (DB.GT.DBMAX) DBMAX = DB
22 306 CONTINUE
C
C COEFFICIENT COST/LONGITUD PER A LA XARXA
C
23 308 CO = COVLOX(MLIN, -EXT*DBMAX)
24 IF (CIR .AND. NR.EQ.3) GO TO 356
C
C DETERMINACIO DE LA DISMINUCIO DE RECORREGUT
C*****
C
25 DO 354 N=1,NF
26 R = 0.
27 NA = N - 1
28 NB = N + 1
29 IF (N.EQ.1) GO TO 312
30 IF (CIR) GO TO 310
31 DB = TOBUS(1,2)
32 GO TO 316
33 310 DB = BF + BU(N0) + BU(2) - EMA(2, LI(2), LI(N0))
34 GO TO 316
35 312 IF (N.EQ.NF) GO TO 314
36 IF (CIR) GO TO 354
37 DB = TCBUS(N0,NF)
38 GO TO 316
39 314 DB = BU(NB) - BU(NA) - EMA(2, LI(NA), LI(NB))
40 316 IF (DB.LE.0.) GO TO 354

```

ISN

```

C
C BALANC PER AL NUS ESCOLLIT
C *****
C PERDUA EN LA LINIA. (DRETA O CIRCULAR).
C
41 I = LI(N)
42 DO 320 L=1,NF
43 IF(L.EQ.N.OR.(CIR.AND.L.EQ.NF)) GO TO 320
44 K = LI(L)
45 P = EMAC(I,K)
46 IF(P.LE.0.) GO TO 320
47 B = TO US(L,N)
48 R = R + P * CONOU(3,I,K,-DB,B,I,E*20)
49
320 CONTINUE
C
C GUANY PER A LA RESTA DE LA LINIA.(LINIA DRETA).
C
50 IF(CIR) GO TO 334
51 IL = 0
52 DO 333 L=1,N0
53 IF(L=N) 324, 333, 322
54 IL = L
55 I = LI(L)
56 BL = BU(L)
57 MS = L + 1
58 IM = 0
59 DO 332 M=MS,NF
60 K = LI(M)
61 P = EMAC(I,K)
62 IF(P.LE.0.) GO TO 332
63 IF(M=N) 328, 332, 326
64 326 IM = I
65 328 B = TOBUS(L,M)
66 IF(IL.NE.IM) R = R + P * CONOU(2,I,K,-DB,B,B-DB)
67 332 CONTINUE
68 IF(IL.EQ.IM) R = R + P * VALOX(L,K,-DB)
69 333 CONTINUE
70 GO TO 332
C
C GUANY PER A LA RESTA DE LA LINIA.(LINIA CIRCULAR).
C
71 334 IL = 0
72 DO 350 L=1,NP
73 IF(L=N) 338, 350, 336
74 336 IL = L
75 338 BL = BU(L)
76 I = LI(L)
77 MS = L + 1
78 IM = 0
79 DO 348 M=MS,N0
80 K = LI(M)
81 P = EMAC(I,K)
82 IF(P.LE.0.) GO TO 348
83 IF(M=N) 342, 348, 340
84 340 IM = I
85 342 B = TOBUS(L,M)
86 U = BU(M) - BL
87 V = BF - U
88 IF(IL.NE.IM) U = U - DB
89 IF(IL.EQ.IM) V = V - DB

```

75.09.22 PAGE 3

FACOM BOS2 FORTRAN -740601- V03 L08 SUPRE

*10. SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

```

90 U = AMINI(U*V)
91 IF(U.LT.B) R = R + P * CONOU(2,I,K,-DB*B*U)
92 IF(U.GE.B) R = R + P * VALOX(1,K,-DB)
93 348 CONTINUE
94 350 CONTINUE
C
C GUANY PER A LA RESTA DE LA XARXA
C
95 352 R = R + CO * (-DB)
C
C COMPARACIO AMB ELS ANTERIORS
C
96 IF(R.LT.RMAX) GO TO 354
97 RMAX = R
98 DBOPT = DB
99 NOPT = N
100 354 CONTINUE
C
C RUPTURA D'UNA LINIA CIRCULAR
C *****
101 IF(.NOT.CIR) GO TO 380
102 356 RUP = 0.
103 DO 366 N=1,N6
104 DB = BU(N+1) - BU(N)
105 IF(DB.LT.CORUP*DBMAX) GO TO 366
C
C PERDUA PER ALS NUSOS D'UNA LINIA
C
106 R = 0.
107 IL = 0
108 DO 364 L=1,NP
109 IF(L.GT.N) IL = 1
110 I = LI(L)
111 BL = BU(L)
112 MS = L + 1
113 IM = 0
114 DO 364 M=MS,N6
115 K = LI(M)
116 P = EMAC(I,K)
117 IF(P.LE.0.) GO TO 364
118 IF(M.GT.N) IM = 1
119 B = TOUS(L,M)
120 U = BU(M) - BL
121 IF(IL.NE.IM) U = BF - U
122 IF(U.GT.B) R = R + P * CONOU(2,I,K,-DB*B*U)
123 IF(U.LE.B) R = R + P * VALOX(1,K,-DB)
124 364 CONTINUE
C
C GUANY PER A LA RESTA DE LA XARXA
C
125 R = R + CO * (-DB)
C
C COMPARACIO AMB ELS ANTERIORS
C
126 IF(R.LT.RUP) GO TO 366
127 RUP = R
128 DBRUP = DB
129 NRUP = N
130 366 CONTINUE
C

```

ISN * 1 FACOM 802 FORTRAN -740601- V03 L08 SUPRE 75.09.22 PAGE * 1D. SEQ.

```

C COMPARACIO RUPTURA= SUPRESIO
C
131 C IF (RUP.LE.RMAX) GO TO 380
C
132 C ACTUALITZACIO EN CAS DE RUPTURA D'UNA L'INIA CIRCULAR
C *****
TA = TA - DBRUP
C
133 C ACTUALITZACIO DE COSTOS
C
133 IL = 0
134 DO 370 L=1,NP
135 IF (L.GT.NRUP) IL = 1
136 I = L / CL
137 BL = B / CL
138 MS = L + 1
139 IM = 0
140 DO 370 M=MS,NB
141 IF (M.GT.NRUP) IM = 1
142 B = TO / US (L,M)
143 IF (IL.E.IM) U = OF - BU(M) + BL
144 IF (IL.EO.IM) U = BU(M) - BL
145 IF (U.GT.B) CALL ACTUAL(2,I,LI(M),B,U)
146 370 CONTINUE
C
C ACTUALITZACIO DE VECTORS
C
147 IF (NRUP.EQ.NB) GO TO 378
148 DO 372 N=1,NF
149 LIR(N) = LI(N)
150 BUR(N) = BU(N)
151 BRUP = BU(NRUP+1)
152 MS = NF - NRUP
153 DO 374 N=1,MS
154 NNRUP = N + NRUP
155 LIR(N) = LIR(NNRUP)
156 BUR(N) = BUR(NNRUP) - BRUP
157 IF (NRUP.EQ.1) GO TO 378
158 BL = B (MS)
159 HT = NRUP - 1
160 DO 376 N=1,MT
161 NMS = N + MS
162 N1 = N + 1
163 LI(NMS) = LIR(N1)
164 BU(NMS) = BUR(N1) + BL
165 376 NF = NF - 1
166 378 N0 = N - 1
167 NP = NP - 1
168 CIR = .FALSE.
169 NNSUP = NNSUP + 1
170 NSUP (NNSUP) = 0
171 IND =
172 RETURN
C
C ACTUALITZACIO EN CAS DE SUPRESIO
C *****
173 380 IF (RMAX.LE.0.) RETURN
174 TA = TA - DBOPT
175 NA = NOPT - 1
176 NB = NOPT + 1

```

L

*ID. SEQ.

FACOM 8052 FORTRAN -740601- V03 L08 SUPRE 75.09.22 PAGE 5

*ID. SEQ.

ISN

C ACTUALITZACIO DE COSTOS PER AL NUS SUPRINT

```

177 I = LI(NOPT)
178 DO 382 L=1,NF
179 IF(L.EQ.NOPT) GO TO 382
180 IF(CIR .AND. L.EQ.NF) GO TO 386
181 B = TOBUS(L,NOPT)
182 CALL ACTUAL(3,I,LI(L),B,I,E+20)
183 382 CONTINUE

```

C ACTUALITZACIO DE COSTOS PER A LA RESTA DE LA LINIA. (DRETA).

```

184 IF(NOPT.EQ.1 .OR. NOPT.EQ.NF) GO TO 396
185 DO 384 L=1,NA
186 I = LI(L)
187 DO 384 M=NB,NF
188 B = TOBUS(L,M)
189 CALL ACTUAL(2,I,LI(M),B,B-DBOPT)
190 GO TO 396

```

C ACTUALITZACIO DE COSTOS PER A LA RESTA DE LA LINIA. (CIRCULAR).

```

191 386 IL = 0
192 DO 395 L=1,NP
193 IF(L.NOPT) 390, 393, 388
194 388 IL = 1
195 390 I = LI(L)
196 BL = BJ(L)
197 MS = L + 1
198 IM = 0
199 DO 394 M=MS,NO
200 IF(M.NOPT) 393, 394, 392,
201 392 IM = 1
202 393 B = TOBUS(L,M)
203 U = BU(M) - BL
204 V = BF - U
205 IF(IL.NE.IM)BU = U - DBOPT
206 IF(IL.EQ.IM)IV = V - DBOPT
207 U = AMIN(U,V)
208 IF(CALL.B) CALL ACTUAL(2,I,LI(M),B,U)
209 394 CONTINUE
210 395 CONTINUE

```

C ACTUALITZACIO DE VECTORS

```

211 396 NNSUP = NNSUP + 1
212 I = LI(NOPT)
213 NSUP(NNSUP) = I
214 NUS(I,LI(N)) = .FALSE.
215 IF(NOPT.EQ.NF) GO TO 398
216 DO 397 N=NOPT,NO
217 LI(N) = LI(N+1)
218 BU(N) = BU(N+1) - DBOPT
219 IF(NOPT.NE.1 .OR. .NOT.CIR) GO TO 398
220 LI(NB) = LI(1)
221 BU(NB) = BU(NP) + EMA(2,LI(NP),LI(NB))
222 NF = NF - 1
223 NB = NB - 1
224 NP = NP - 1
225 IND = 1

```

*ID. SEQ.

SOURCE STATEMENT / ERROR MESSAGE

* I

ISN

RETURN
END

226
227

8.1.5.7. Subrutina ELIM.

Es la darrera i més senzilla de les rutines que hom podria anomenar d'alt nivell.

Quan hi ha una línia de 2 nusos, calcula si surt a compte d'eliminar-la. En cas afirmatiu, ho fa.

*ID. SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C *****

C ELIMINACIO D'UNA LINIA EN UNA XARXA

C *****

1 SUBROUTINE ELIM(IND, NLIN)

2 LOGICAL CIR, NUS

3 COMMON /ETI 1/ D(44,44), TA, AUT, MN, NOO, IENT, ISOR

4 /ETI 2/ LI(25), BU(25), CIR, NF, NO, NP, NUS(44,40)

5 /ETI 3/ NINS(25), NSUP(25), NNINS, NNSUP

6 IND = 0

7 B = TOBUS(1,2)

8 CO = COVLOX(NLIN, -B)

C BALANC I COMPARACIO

C

9 I = LI(1)

10 K = LI(2)

11 R = EM(1,I,K) * CONOU(3,I,K,B,1,E+20) + CO * (-B)

12 IF(R.LE.0.) RETURN

C ACTUALITZACIO DE COSTOS

C

13 TA = TA - B

14 CALL ACTUAL(3,I,K,B,1,E+20)

C ACTUALITZACIO DE LA LINIA

C

15 NF = 0

16 NUS(I,NLIN) = .FALSE.

17 NUS(K,NLIN) = .FALSE.

18 NNSUP = NNSUP + 2

19 NSUP(NNSUP-1) = I

20 NSUP(NNSUP) = K

21 IND = 1

22 RETURN

23 END

8.1.5.8. Funció CONOU.

Es cridada per les 4 rutines anteriors; i té com a objectiu de calcular la variació del cost mitjà associat a un flux, conseqüència d'un canvi en els itineraris.

El procés de càlcul és el següent:

- formació de la taula itineraris-línies.
- càlcul del cost inicial, per mitjà de la funció TERM, que ja es descriurà al seu torn.
- modificació en la taula anterior, segons una de les 3 opcions:
 - 1 vol dir addició d'una nova línia, que representa per al flux considerat un temps: S
 - 2 canvi de l'antic, que valia Q pel nou que val S.
 - 3 supressió de l'antic de valor Q.

Segons els casos, la funció hi

afegeix
suprimeix

 un itinerari o be

augmenta
disminueix

 el nombre de línies constituents d'un dels itineraris ja existents.

A més, per tal d'accelerar el càlcul, té previst 3 processos paral.lels, segons si el nombre d'itineraris és 0, 1 o ≥ 2 .

- càlcul del cost mitjà, després de la variació, i diferència amb l'antior. Aquesta diferència és el valor de sortida.

FACOM 8052 FORTRAN -740601- V03 L08
* I SOURCE STATEMENT / ERROR MESSAGE

ISN * I

*ID, SEQ.

```

C *****
C VARIACIO DE COST PER INCLUSIO, CANVI O SUPRESSIO D'UNA LINIA
C *****

```

```

1 FUNCTION CONQU(IND, TO, KO, DB, @, S)
2 COMMON /ETI 1/ D(44,44), TA, AUT, NN, NOO, IENT, ISOR
3 /ETI 5/ C(44,44), NIL(44,44)
4 /ETI 9/ NIT, R(9), NLIT(9), TIT(9), YY(9),
5 IS(44), KS(900), RS(900), NS(900), NTS
6 DATA /1,E-3/

```

```

C TEMPS INICIAL, ANTERIOR A LA VARIACIO
C *****

```

```

4 I = MINO(10,KO)
5 K = MAXO(10,KO)
6 T = TERM(1,I,K,0.)
7 NI = NIT + 1
8 GO TO (502, 504, 518, 518, 518, 518, 518, 518, 518), NI

```

```

C NO HI HAVIA CAP LINIA
C *****

```

```

9 502 R(1) = S
10 NLIT(1) = 1
11 NIT = 1
12 GO TO 536

```

```

C NOMES HI HAVIA UN ITINERARI
C *****

```

```

13 504 GO TO (506, 508, 514), IND

```

```

C INCLUSIO

```

```

14 506 IF(ABS(R(1)-S),GT.E) GO TO 512
15 NLIT(1) = NLIT(1) + 1
16 GO TO 536

```

```

C CANVI

```

```

17 508 IF(NIL(K,I),NE.1) GO TO 510
18 R(1) = S
19 GO TO 536
20 510 NLIT(1) = NLIT(1) - 1
21 512 NIT = 2
22 R(2) = S
23 NLIT(2) = 1
24 GO TO 536

```

```

C SUPRESSIO

```

```

25 514 IF(NIL(K,I),NE.1) GO TO 516
26 NIT =
27 GO TO 536
28 516 NLIT(1) = NLIT(1) - 1
29 GO TO 536

```

*ID. SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C HI HAVIA 2 O MES ITINERARIS

C *****

C ELIMINACIO DE LA LINIA ANTIGA

C

30 518 IF(IND.EQ.1) GO TO 530

31 DO 520 IT=1,NIT

32 IF(ABS(R(IT)-0).LE.E) GO TO 522

33 520 CONTINUE

34 522 IF(NLIT(IT).EQ.1) GO TO 524

35 NLIT(IT) = NLIT(IT) - 1

36 GO TO 528

37 524 IF(IT.EQ.NIT) GO TO 526

38 R(IT) = R(NIT)

39 NLIT(IT) = NLIT(NIT)

40 526 NIT = NIT - 1

C

C INCLUSIO DE LA LINIA NOVA

C

41 528 IF(IND.EQ.3) GO TO 536

42 530 DO 532 IT=1,NIT

43 IF(ABS(R(IT)-S).LE.E) GO TO 534

44 532 CONTINUE

45 NIT = NIT + 1

46 R(NIT) = S

47 NLIT(NIT) = 1

48 GO TO 536

49 534 NLIT(IT) = NLIT(IT) + 1

C

C DIFERENCIA DE COSTOS

C *****

C

50 536 CONOU = T - TERM(2,I,K,DB)

51 RETURN

52 END

8.1.5.9. Subrutina ACTUAL.

Un cop decidida una actualització per una rutina d'alt nivell, aquesta s'ocupa de dur-la a terme per a un parell de nusos determinat.

La lògica de funcionament és idèntica a la de la funció CONOU, amb 3 opcions d'entrada i 3 processos paral·lels; aquí, però no hi ha cap càlcul de cost.

En canvi es presenta una complicació addicional, deguda al fet que calgui revisar els itineraris 2on i següents; com que llurs valors estan emmagatzemats en vectors suplementaris, pot esdevenir-se que se'ls hi hagi de fer cabuda per als nous valors o, a l'inrevés, que per supressió d'un itinerari s'escurci la pila d'itineraris suplementaris. Aquests canvis comporten desplaçaments al llarg dels esmentats vectors.

Z

FACOM BOS2 FORTRAN -740601- V03 L08
* I SOURCE STATEMENT / ERROR MESSAGE *ID. SEQ.

```

1  C *****
2  C ACTUALITZACIO DELS TEMPS DE VIATGE DE CADA ITINERARI
3  C *****
4  C SUBROUTINE ACTUAL(IND, IO, KO, O, S)
5  LOGICAL KEJM
6  COMMON /ETI 1/ D(44,44), TA, AUT, NM, NOO, IENT, ISOR
7  /ETI 5/ C(44,44), NIL(44,44)
8  /ETI 9/ NIT, R(9), NLIT(9), TIT(9), YV(9),
9  /ETI 9/ IS(44), KS(900), RS(900), NS(900), NTS
10 DATA E /1.E-3/
11
12 C BIFURCACIO SEGONS NOMBRE D'ITINERARIS ANTERIOR
13 C *****
14 I = MINO(IO,KO)
15 K = MAXO(IO,KO)
16 NIT=NIL(I,K)
17 NI = NIT + 1
18 GO TO (602, 604, 616, 616, 616, 616, 616, 616, 616), NI
19
20 C NO HI HAVIA CAP ITINERARI
21 C *****
22
23 602 C(K,I) = S
24 NIL(I,K) = 1
25 NIL(K,I) = 1
26 RETURN
27
28 C HI HAVIA UN SOL ITINERARI
29 C *****
30
31 604 LIP = NIL(K,I)
32 GO TO (606, 608, 612), IND
33
34 C INCLUSIO
35 C *****
36 606 IF(ABS(C(K,I)-S).GT,E) GO TO 616
37 NIL(K,I) = NIL(K,I) + 1
38 RETURN
39
40 C CANVI
41 C *****
42 608 IF(FLIP.NE.S) GO TO 616
43 C(K,I) = S
44 RETURN
45
46 C SUPRESSIO
47 C *****
48 612 IF(FLIP.NE.S) GO TO 614
49 C(K,I) = 1.E+20
50 NIL(I,K) = 0
51 NIL(K,I) = 0
52 RETURN
53
54 614 NIL(K,I) = NIL(K,I) - 1
55 RETURN
56
57 C HI HAVIA 2 O MES ITINERARIS
58 C *****

```

B

*ID. SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C EXTRACCIO DEL SEGON I SEGUENTS ITINERARIS

```

29 616 R(1) = C(K,1)
30 NLIT(1) = NIL(K,1)
31 NANT = NIT
32 JIN = IS(1)
33 JSU = IS(1+1) - 1
34 IF(JIN.LE.JSU) GO TO 620
35 KIN = JIN
36 GO TO 632
37 620 KLIM = .FALSE.
38 IT = 1
39 DO 626 J=JIN,JSU
40 IF(KS(J)-K) 626, 622, 628
41 622 IF(KL(J) GO TO 624
42 KIN = J
43 KLIM = .TRUE.
44 IT = IT + 1
45 RC(IT) = RS(J)
46 NLIT(IT) = NS(J)
47 CONTINUE
48 IF(.NOT.KLIM) KIN = JSU + 1
49 IF(KLIM) KSU = JSU
50 GO TO 632
51 628 IF(.NOT.KLIM) KIN = J
52 IF(KLIM) KSU = J - 1

```

C ELIMINACIO DE LA LINEA ANTIGA

```

53 632 IF(IND.EQ.1) GO TO 644
54 DO 634 IT=1,NIT
55 IF(ABS(R(IT)-0).LE.E) GO TO 636
56 634 CONTINUE
57 636 IF(NLIT(IT).EQ.1) GO TO 638
58 NLIT(IT) = NLIT(IT) - 1
59 GO TO 642
60 638 IF(IND.EQ.NIT) GO TO 640
61 RC(IT) = RC(IT)
62 NLIT(IT) = NLIT(IT)
63 640 NIT = NIT - 1

```

C INCLUSIO DE LINEA NOVA

```

64 642 IF(IND.EQ.3) GO TO 638
65 644 DO 646 IT=2,NIT
66 IF(ABS(R(IT)-S).LE.E) GO TO 648
67 646 CONTINUE
68 NIT = NIT + 1
69 RC(NIT) = S
70 NLIT(NIT) = 1
71 GO TO 638
72 648 NLIT(IT) = NLIT(IT) + 1

```

C REINSERCCIO DELS ITINERARIS EN ELS VECTORS D'EMMAGATZAMENT

```

73 *****
73 IF(NIT-NANT) 668, 674, 660

```

C AUGMENT D'ITINERARIS

```

74 660 NIS = NIS + 1

```

*ID. SE0.

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE

75 IF(NTS.EQ.1) GO TO 664

76 J = NTS

77 662 KS(J) = KS(J-1)

78 RS(J) = RS(J-1)

79 NS(J) = NS(J-1)

80 IF(J.LE.KIN+1) GO TO 664

81 J = J - 1

82 GO TO 662

83 664 MM = I + 1

84 DO 666 J=MM,NN

85 666 IS(J) = IS(J) + 1

86 KS(KIN) = K

87 GO TO 674

C DISMINUCIO D'ITINERARIS

C

88 668 NTS = NTS - 1

89 DO 670 J=KSU,NTS

90 KS(J) = KS(J+1)

91 RS(J) = RS(J+1)

92 670 NS(J) = NS(J+1)

93 MM = I + 1

94 DO 672 J=MM,NN

95 672 IS(J) = IS(J) - 1

C REINSERCO DELS ITINERARIS

C

96 674 C(K,I) = R(I)

97 NIL(K,K) = NIT

98 NIL(K,I) = NLIT(I)

99 IF(NIT.EQ.1) RETURN

100 DO 676 IT=2,NIT

101 KK = KIN + IT - 2

102 RS(KK) = R(IT)

103 676 NS(KK) = NLIT(IT)

104 RETURN

105 END

8.1.5.10 Subrutina LLIS.

Imprimeix les dades relatives a una línia, en acabar-ne les manipulacions de les rutines de nivell superior. Té 2 opcions d'impressió: el llistat reduït i el llistat complet, dels quals ja s'ha parlat. També s'ha enumerat el contingut de la informació fornida en cada cas als apartats 8.1.3.2. i 8.1.3.3., i ara ja no s'hi insisteix més. La rutina consta de 4 parts:

- a) En cas de llistat complet, inicialitza el vector de càrrega per tram; si, a més, és circular, calcula per a cada nus quin és el més llunyà, per tal de carregar una semilínia o l'altra.
- b) Càlcul de les demandes. A través de crides successives a la funció TERM, esbrina quin és el valor total de cada flux, i quina proporció n'absorbeix la línia en qüestió.
- c) Càlcul dels paràmetres relatius a la xarxa.
- d) En el llistat complet, la rutina preveu que l'encapçalament sigui correcte; per tant, prèviament a la impressió de la línia determina si el llistat ha de començar una pàgina nova o no.

Z

Z

*ID. SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

```

C *****
C INFORMACIO RELATIVA A LA LINIA
C *****
1 SURROUTINE LLIS(CIND, MLIN)
2 INTEGER LIM(25)
3 INTEGER*4 ICAR(25), IDEL, IDEX, INFO(3,3), IDII
4 LOGICAL CIR, NJS
5 COMMON /ETI 1/ D(44,44), TA, AUT, NN, NCO, TENT, ISOR
   /ETI 2/ LI(25), BU(25), CIR, NF, NP, NUS(44,40)
   /ETI 3/ MINS(25), NSUP(25), NNINS, NNSUP
   /ETI 4/ MRE, NSING, NSUP, YBUS, *, CANT, CINI, CDIF, CH
   /ETI 9/ NIT, R(9), NLI(9), TII(9), YI(9),
6 A ISC(44), KSC(900), NS(900), NS(900), NTS
   /ETI10/ IK, IITER, CNN, CMI(15), GPI(15), GAI(15),
   /ETI11/ SKII(15), IDII(15), MII(15), NINSI(15), NSUPI(15),
   /ETI12/ REDUT(15)
7 T /TREC / CAR(25), DIBU(25), DEL
   /EQUIVALENCIA (ICAR(1),ICAR(1)), (IDEL,DEI), (IDEX,DEX),
   /EQUIVALENCIA (LIM(1),DIBU(1))
8 MU = NR
9 MF = NF
10 IF(NF.NE.0) GO TO 702
11 IA = 3
12 GO TO 727
13 702 IA = 1
14 IF(CIR) IA = 2
C INICIALIZACIO DELS COMPTADORS
C
15 DEL = 0
16 IF (IND.EQ.0) GO TO 712
17 DO 704 J=1,MN
18 704 CAR(J) = 0.
C PUNTS DE CANVI DE CARGA PER A LINES CIRCULARS
C
19 IF(.NOT.CIR) GO TO 712
20 SEMI = 0.5 * BU(NF)
21 MF = NF - 1
22 MN = MN - 1
23 DO 710 L=1,M0
24 X = BU(L)
25 MO = L + 1
26 IF(SEMI.LT.BU(M)-X) GO TO 708
27 706 CONTINUE
28 LIM(L) = MF + 1
29 GO TO /10
30 708 LIM(L) = M
31 710 CONTINUE
32 C CALCUL DE LA DEMANDA TOTAL I PER TRAN

```

*ID. SEB.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C*****

712 UO 726 L=1,MO

I = LI(L)

MO = L + 1

DC 726 M=MO*MF

K = LI(M)

T = TERM(3,I,K,0.)

J = 1

IF(.NOT.EG.1) GO TO 716

B = 10BUS(L,M)

DO 714 J=1,NLT

IFCH(J).EG.8) GO TO 716

714 CONTINUE

716 P = EMAC(1,K) * YY(J) / FLOAT(NLT(J))

DEL = DEL + P

C CARREGA DE LA LINEA

C*****

IF(IND.EG.0) GO TO 726

IF(CIR .AND. M.GE.LI*(L)) GO TO 720

C CAS DE LINEA DRETA O CIRCULAR AMB CARREGA DIRECTA

MO = M - 1

DO 718 J=L,MO

718 CAR(J) = CAR(J) + P

GO TO 726

C CAS DE LINEA CIRCULAR AMB CARREGA INVERSA

720 UO 722 J=N,MF

722 CAR(J) = CAR(J) + P

IF(L.EG.1) GO TO 726

MO = L - 1

DO 724 J=L,MO

724 CAR(J) = CAR(J) + P

726 CONTINUE

C COST MITJA, DEMANDA DE LA XARXA I D'ALTRES PARAMETRES

C*****

727 IF(IX.EG.1 .AND. ITER.EG.0 .AND. MLIN.EG.1) GO TO 728

IF(NNIPS.EG.0 .AND. NNSUP.EG.0) GO TO 730

DEX = 0.

DO 729 I=1,NOO

MO = I + 1

UO 729 K=NO*NN

P = D(I,K)

IF(P.LE.0.) GO TO 729

CM = CM + P * TERM(3,I,K,0.)

DEX = DEX + P * (1. - YY(NI+1))

729 CONTINUE

CM = CM / W

IDEX = DEX

SX = YBUS * TA / 1000.

IF(IA.EG.3) GO TO 732

730 IDEL = DEL

SL = YBUS * BU(MF) / 1000.

732 IF(IND.EG.0) GO TO 744

```

79 IF (NLINE.EQ.3) LL = NTF - 3
80 LO = 8
81 IF (N0.GT.NTRAM) LO = 12
82 IF (IA.EQ.3) LO = 5
83 IF (NNSUP.GT.0) LO = LO + 2
84 IF (NMINI.GT.0) LO = LO + 2
85 IF (LL.GT.LO) GO TO 734
86 LL = NTF + 1 - LO
87 WRITE (ISOR,752) NLINE, (INFO(J,IA), J=1,3)
88 GO TO 736
89 734 LL = LL - LO
90 WRITE (ISOR,754) NLINE, (INFO(J,IA), J=1,3)
C
C LINEA ELIMINADA
C
91 736 IF (IA.NE.3) GO TO 738
92 WRITE (ISOR,759) IDEX
93 WRITE (ISOR,763) SX
94 WRITE (ISOR,764) CM
95 GO TO 742
C
C IMPRESIO DE LES FRANGES
C
96 738 DC 740 J=1,N0
97 ICAR(J) = CAR(J)
98 740 DIBU(J) = BU(J+1) - BU(J)
99 FF = MIN(N0,NTRAM)
100 KO = KF + 1
101 *RITE (ISOR,756) IDEL, (J, J=1,KO)
102 *RITE (ISOR,758) IDEX, (DIBU(J), J=1,NF)
103 *RITE (ISOR,760) SX, (ICAR(J), J=1,KF)
104 *RITE (ISOR,762) CM, (LI (J), J=1,KO)
105 IF (N0.LE.NTRAM) GO TO 742
106 *RITE (ISOR,744) CM
107 WRITE (ISOR,766) CJ
108 WRITE (ISOR,768) (DIBU(J), J=KO,N0)
109 WRITE (ISOR,770) (ICAR(J), J=KO,NF)
110 *RITE (ISOR,772) (LI (J), J=KO,NF)
111 742 IF (NNSUP.GT.0) WRITE (ISOR,774) (NSUP (J), J=1,NNSUP)
112 IF (NMINI.GT.0) WRITE (ISOR,776) (NINI (J), J=1,NMINI)
113 RETURN
C
C LLISTAT REDUIT
C *****
C
744 IF (NLINE.EQ.1) WRITE (ISOR,778)
114 IF (IA.NE.3) WRITE (ISOR,780) NLINE, IDEL, SX, NF, NMINI, NNSUP,
115 (INFO(J,IA), J=1,3), IDEX, SA, CM, NLINE
116 IF (IA.EQ.3) WRITE (ISOR,782) NLINE, NF, NNSUP,
117 (INFO(J,IA), J=1,3), IDEX, SA, CM, NLINE
118 RETURN
C
C FORMATS
C
752 FORMAT(' / / / ' , LINEA *, ' 12. ', *, ' 5X. 3A')

```

*ID. SEE.

SOURCE STATEMENT / ERROR MESSAGE

```

119 754 FORMAT('OLINIA *', I2, '(#', I5, '344)
120 756 FORMAT('O DEM, LIN.,', I7, ', ', ORD., ', I2, I5I7)
121 758 FORMAT('O DEM, XRX.,', I7, ', ', TEMPS., I5F7.2 )
122 759 FORMAT('O DEM, XRX.,', I7, ', '
123 760 FORMAT('O LNG, LIN.,', F7.3, ', CAR.,', I5I7 )
124 762 FORMAT('O LNG, XRX.,', F7.3, ', NUS.,', I2, I5I7)
125 763 FORMAT('O COST
126 764 FORMAT('O
127 766 FORMAT(1EX, ', ORD.,', I2, I5I7)
128 768 FORMAT(18X, ', TEMPS.,', I5F7.2 )
129 770 FORMAT(18X, ', CAR.,', I5I7 )
130 772 FORMAT(18X, ', NUS.,', I2, I5I7)
131 774 FORMAT('O', I9X, 'SUPRIMITS ', I2I13)
132 776 FORMAT('O', I9X, 'INSERTATS ', I2I13)
133 778 FORMAT(///
1 133, INFORMACIO SOBRE LA LINIA', T96, INFORMACIO SOBRE LA XARXA',
2 133, 25('*,'), T96, 25('*,') NUSOS'
3 11X, LINIA DEMANDA LONGITUD NUSOS
4 28X, DEMANDA LONGITUD CUST LINIA'
5 30X, '(KM,.) ACTUALS INSEKTATS SUPRIMITS', 41X, '(KM,.) (MIN,.)')
780 FORMAT(/10X, I5, I10, F10.3, I10, F10.3, I4, I10X, I10, F10.3, F10.5, I5)
782 FORMAT(/10X, I5, I20X, I10, I10X, I10, I10, I10, I10, F10.3, F10.5, I5)
END

```

8.1.5.11. Subrutina IPI.

S'activa al final de cada iteració, i s'encarrega de calcular i d'imprimir la informació relativa a l'estat de la xarxa descrita a l'apartat 8.1.3.4.

La primera part, que fa el càlcul, està inclosa en un bucle doble, referit als parells de nusos. Aleshores, flux per flux calcula el nombre d'itineraris i de línies que el serveixen, la demanda absorbida, no absorbida i total, i afegeix els valors a la matriu i vectors corresponents. Aquesta part es clou amb el càlcul de les sumes marginals.

La segona part, relativa a la impressió, determina prèviament fins a quin element és plena la taula de línies-itineraris, per tal de determinar els límits de la impressió.

ISN * I SOURCE STATEMENT / ERROR MESSAGE *ID. SEQ.

C *****

C INFORMACIO POST-ITERACIO

C *****

```

1 SUBROUTINE IPI
2 INTEGER NUM(9), PZ, PAR, PLI, PIT, PS
3 INTEGER*4 IDZ, IGLI(8,3), IDIT(6,3), ISDIT(3), ISDLI(3), IDII
4 COMMON /ETI 1/ O(4,4), TA, AUT, NN, NCO, IENT, ISOR
5 /ETI 4/ MRE, NSINS, NSUP, YBUS, W, CANT, CINI, CDIF, CM
6 /ETI 9/ NIT, R(9), NLIT(9), TIT(9), YY(9),
7 IS(4,4), KS(900), RS(900), NS(900), NTS
8 /ETI10/ IX, ITER, CMN, CMII(15), GPII(15), GAPI(15),
9 /ETI11/ SALLI(15), IDII(15), MRIT(15), MINSI(15), NSUPI(15),
10 REDUI(15)
11 /TRER / PZ, PAR(8,6), PLI(8), PII(6), PS,
12 DZ, DLI(8,3), DIT(6,3), SDIT(3), SDLI(3)
13 EQUIVALENCE (DZ, IDZ), (DLI(1,1), IDLI(1,1)), (DIT(1,1), IDIT(1,1)),
14 (SDIT(1,1), ISDIT(1,1)), (SDLI(1,1), ISDLI(1,1))
15 DATA NUM /0, 1, 2, 3, 4, 5, 6, 7, 8/

```

C INICIALIZACIO DELS COMPUTADORS

C

```

7 CM = 0.
8 PZ = 0.
9 DZ = 0.
10 PS = 0.
11 DO 802 M=1,6
12 DIT(L,1) = 0.
13 802 DIT(L,2) = 0.
14 DO 804 L=1,8
15 DLI(L,1) = 0.
16 DLI(L,2) = 0.
17 DO 804 M=1,6
18 804 PAR(L,M) = 0

```

C CALCUL DE LES DEMANDES I CREACIO DE LA TAULA

C

```

19 DO 810 I=1,NGO
20 NC = I + 1
21 DO 810 K=NO,ANN
22 F = D(I,K)
23 CM = CM + F * TERM(3,1,K*0.)
24 IF(NIT,NE,0) GO TO 806
25 PZ = PZ + 1
26 DZ = DZ + F
27 GO TO 810
28 806 L = 0
29 DO 808 J=1,NIT
30 L = L + NLIT(J)
31 808 IF(L .GT. 8) L = 8
32 IF(NIT,GT,6) NIT = 6
33 PAR(L,NIT) = PAR(L,NIT) + 1
34 CPEU = YY(NIT+1) * F
35 DBUS = F - DPLU
36 DLI(L,1) = DLI(L,1) + DPEU
37 DLI(L,2) = DLI(L,2) + DBUS
38 DIT(NIT,1) = DIT(NIT,1) + DPEU
39 DIT(NIT,2) = DIT(NIT,2) + DBUS

```

6

73.09.26 PAGE 2

FACOM BOS2 FORTRAN -740601- V03 L08 IPT
SOURCE STATEMENT / ERROR MESSAGE

*ID. SEQ.

ISN

810 CONTINUE
PZ = PZ + NN
DO 812 I=1,NN
812 DZ = DZ + D(I,1)

C DEMANDES MARGINALS, PER LINIA I PER ITINERARI

44 DO 816 L=1,8
45 DLI(L,3) = DLI(L,1) + DLI(L,2)
46 MS = 0
47 IM = MINC(L,6)
48 DO 814 M=1,IM
49 814 MS = MS + PAR(L,M)
50 816 PLI(L) = MS
51 DO 820 N=1,6
52 DIT(M,3) = DIT(M,1) + DIT(M,2)
53 PS = 0
54 DO 818 L=1,8
55 818 MS = MS + PAR(L,M)
56 820 PIT(M) = MS

C DEMANDES TOTALS, PER LINIA I PER ITINERARI

57 S1 = 0
58 S2 = 0
59 DO 822 L=1,8
60 S1 = S1 + DLI(L,1)
61 822 S2 = S2 + DLI(L,2)
62 SOLI(1) = S1 + DZ
63 SOLI(2) = S2
64 SOLI(3) = S1 + S2 + DZ
65 S1 = 0
66 S2 = 0
67 DO 824 M=1,6
68 S1 = S1 + DIT(M,1)
69 S2 = S2 + DIT(M,2)
70 PS = PS + PIT(M)
71 824 SDIT(1) = S1 + DZ
72 SDIT(2) = S2
73 SDIT(3) = S1 + S2 + DZ
74 PS = PS + PZ

C PAS A VARIABLE ENTERA DE LES DEMANDES

75 IDZ = DZ
76 DO 826 J=1,3
77 ISDIT(J) = SDIT(J)
78 DO 828 L=1,8
79 826 IDIT(L,J) = DIT(L,J)
80 DO 828 J=1,3
81 ISOLI(J) = SOLI(J)
82 DO 828 M=1,6
83 828 IDLI(M,J) = DLI(M,J)

C COST, GUANY I D'ALTRES PARAMETRES

84 CM = CM / W
85 K1 = (CANT - CM) / CDIF * 100.
86 K2 = (CINT - CM) / CDIF * 100.
87 SX = YBUS + TA / 1000.
88 S = 2. * TA / AUT

FACOM 8052 FORTAN -740601- V03 L08 IPI
SOURCE STATEMENT / ERROR MESSAGE *ID. SEQ.

C IMPRESSIO DE LA TAULA DE LINES-ITINERARIS

```

89 LM = 8
90 830 IF(PLI(LH).GT.0) GO TO 832
91 LM = LM - 1
92 GO TO 830
93 832 IM = MINO(LM,6) + 1
94 WRITE(ISOR,850) (NUMCM,N=1,IM)
95 WRITE(ISOR,852) NUM(1), PZ, PZ, DZ, DZ
96 IREV = 0
97 DO 834 L=1,LI
98 IRED = IRED + L * PLI(L)
99 E = S / FLOAT(L+1)
100 IM = MINO(L,6)
101 834 WRITE(ISOR,854) PLI(L), (IDL(L,M),M=1,3), E, NUM(L+1),
(PAR(L,M), M=1,IM)
WRITE(ISOR,856) PS, (ISDL(M), M=1,3), PZ, (PI(L),L=1,IM)
WRITE(ISOR,858) ISDIT(1), IDZ, (IDIT(L,1), L=1,IM)
WRITE(ISOR,860) ISDIT(2), (IDIT(L,2), L=1,IM)
WRITE(ISOR,862) ISDIT(3), IDZ, (IDIT(L,3), L=1,IM)

```

C IMPRESSIO DELS PARAMETRES

```

106 RED = FLOAT(IRED) / CNN
107 WRITE(ISOR,864) CM, R1, R2, S, SX, MRE, NSINS, NSSUP, RED

```

C RESUM DE L'ALGORISME

```

108 I = ITEM + 1
109 CM(I) = CM
110 GP(I) = P1
111 G(I) = R2
112 SX(I) = SX
113 ID(I) = ISDL(2)
114 MP(I) = NMG
115 NSINS(I) = NSINS
116 NSSUP(I) = NSSUP
117 RED(I) = RED
118 RETURN

```

C FORMATS

```

119 850 FORMAT(/// , 35X, 'NOMBRE DE PARELLS DE NUSOS SEGONS .LINES',
'///',
2 T50,'NO. ITINERARIS', T69,'TOTAL',
3 T50,'NO.', T67,'PARELLS', T92,'DEMANDA', T116,'TIEMPS ESPERA',
4 T69,'NUSOS', T79,'NO.ABSORB.', ABSORB., TOTAL,
5 12Z,'(MIN.)', T7,'LINES', T17,'(717)')
120 852 FORMAT(// 12, 11, T69,15, T79,110, T99,110)
121 854 FORMAT(// T69,15, T79,310, T16,2, T1,12, 1X, 617)
122 856 FORMAT(// 78 , 'TOTAL')
123 858 FORMAT(16, 'NUSOS',
1 /// T66,18, T1, 'NO.ABSORB.', T11, 617)
124 860 FORMAT(// T66,18, T1, 'ABSORB.', T11, 617)
125 862 FORMAT(// T66,18, T1, 'TOTAL', T11, 617)
126 864 FORMAT(// 80X, 'COST MITJA', T80, 'MIN.',
1 80X, 'GUANY PERCENTUAL', T80,
2 80X, 'GUANY ACUMULAT', T80,
3 80X, 'INTERVAL', T80, 'MIN.',
4 )

```

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE *ID. SEQ.

3	80X, 'LONGITUD DE XARXA', F11.3, 3X, 'KM.,'	//
4	80X, 'NOMBRE DE LINIES ', 17	//
5	80X, 'NUSOS INSERTATS ', 17	//
6	80X, 'NUSOS SUPRIMITS ', 17	//
7	80X, 'INDEX REDUNDANCIA', F11.3)	

127 END

8.1.5.12 Subrutina EXTR.

La xarxa i els temps de recorregut per cada línia estan emmagatzemats en sengles matrius. Per tant, llur accés sempre és més lent en l'execució i més feixuc en la programació. A més, donada l'estructura de l'algorisme, quan hom treballa en una línia no cal cap dada de les restants. Ja es veu, doncs, que ha de ser molt més rendible de passar la línia i els temps de recorregut a uns vectos també, creats "ad hoc". Aquesta és la missió, doncs, de la subrutina EXTR. Passar les dades de les matrius als vectors en començar o, a l'inrevés en acabar, el tractament d'una línia.

*ID. SEQ.

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE

C *****

C EXTRACCIO O INCLUSIO D'UNA LINEA

C *****

C SUBROUTINE EXTR(INO,NLIN)

LOGICAL NUS, CIRC, CIR

COMMON /ET1 O/ NL(25,40), T(25,40), NNU(40), CIRC(40), LINUS,

B NNUM, MREM, MEMAX

2 /ET1 2/ LI(25), BU(25), CIR, NF, NO, NP, NUS(44,40)

C INCLUSIO

C GO TO (902,906), IND

902 CIRC(NLIN) = CIR

NNU(NLIN) = NF

DO 904 J=1,NF

NL(J,NLIN) = LI(J)

904 T (J,NLIN) = BU(J)

RETURN

10 C EXTRACCIO

C 906 CIR = CIRC(NLIN)

NF = NNU(NLIN)

NO = NF - 1

NP = NF - 2

DO 908 L=1,NF

LI(L) = NL(L,NLIN)

908 BU(L) = T (L,NLIN)

RETURN

19 END

8.1.5.13. Funció COVLOX.

Resol el problema del guany per a la resta de la xarxa, φ_x , exposat en descriure l'algorisme. De fet, forneix el coeficient K , que el mòdul d'alt nivell farà servir després per al càlcul dels guanys generals.

*JD, SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C *****

C VARIACIO DEL TEMPS MITJA, PER UNA VARIACIO DE LA LONGITUD DE LA XARXA

C *****

FUNCTION COVLOX(NLIN, DBMAX)

LOGICAL CIR, NUS

COMMON /ETI 1/ D(44,44), TA, AUT, NN, NOO, IENT, ISOR

2 /ETI 2/ LI(25), BU(25), CIR, NF, NO, NP, NUS(44,40)

3 /ETI 3/ C(44,44), NIL(44,44)

4 G = 0.

5 DO 1004 I=1,NOO

6 NO = I + 1

7 DO 1004 K=NO,NN

8 IF(D(I,K).LE.0..OR. NIL(I,K).EQ.0) GO TO 1004

9 IF(NLIN.EQ.0) GO TO 1002

10 IF(NUS(I,NLIN) .AND. NUS(K,NLIN)) GO TO 1004

11 1002 G = G + VALOX(I,K,DBMAX) * D(I,K)

12 1004 CONTINUE

13 COVLOX = G / DBMAX

14 RETURN

15 END

8.1.5.14. Funció VALOX.

Es a la COVLOX el que un flux és a tota la xarxa. Calcula i dona com a sortida el guany experimentat per un flux si el temps total de recorregut sofreix una variació.

Aquest valor el fa servir, tant COVLOX per a fornir el coeficient mitjà per a tota la xarxa, com les rutines positives d'alt nivell (COMEN i INSER) per a fer les correccions en el càlcul del guany.

FACOM JOS2 FORTRAN -740601- V03 L08

*ID, SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

```

C *****
C *****
C *****
C *****
C *****
C *****
C *****
C *****
C *****
C *****

```

C VARIACIO DEL TEMPS DE VIATGE PER VARIACIO DE LA LONGITUD DE LA XARXA

C FUNCION VALOX(I, K, DB)

T = TERM(1,I,K,0,)

U = TERM(2,I,K,DB)

VALOX = T - U

RETURN

END

8.1.5.15. Funció TERM.

Es la funció base del programa. En línies generals, dóna el cost mitjà entre dos nusos, en un estadi determinat del procés.

De fet té 3 opcions:

- 1 . és la suara explicada.
- 2 . recalcula el cost quan la funció CONOU introdueix alguna variació en línies, itineraris o temps de recorregut de la xarxa.
- 3 . És idèntica a la 1 però, a més, calcula la distribució de càrregues entre els diversos itineraris.

Donat l'ús freqüentíssim que es fa de la funció, en el seu disseny hom s'ha esmerçat per a minimitzar-ne el temps de càlcul. Està desglossada en 3 processos paral·lels, tal com ja es féu a llur torn amb CONOU i ACTUAL, segons si hi ha 0, 1 o ≥ 2 itineraris.

Evidentment, el darrer cas és el general i inclou els altres dos; ara bé, donades les particularitats del cas de cap i un itineraris, pot aconseguir-se un gran estalvi de temps de càlcul amb un increment petit de complicació en el programa. A més, la major part dels fluxos es tan servits per cap o un itinerari, la qual cosa referma la necessitat d'un tractament especial.

FACOM BOS2 FORTRAN -740601- V03 L08

ISN * 1 SOURCE STATEMENT / ERROR MESSAGE *ID. SEQ.

```

C *****
C TEMPS DE VIATGE ENTRE UN PARELL DE NUSOS
C *****
C
1 FUNCTION TERM(IND, IO, KO, DB)
2 LOGICAL CTR, NUS
3 COMMON /ETI 1/ D(44,44), TA, AUT, NN, NOO, IENT, ISOR
4 /ETI 2/ LI(25), BUC(25), CIR, NF, NO, NP, NUS(44,44)
5 /ETI 3/ C(44,44), NIL(44,44)
6 /ETI 4/ E, PE, HAC
7 /ETI 5/ NIT, R(9), NLIT(9), TIT(9), Y(9),
8 /ETI 6/ RS(900), NS(900), NTS
9 /ETI 7/ IS(44), KS(900)
A
C VALORS DE BASE
C
4 GO TO (1200, 1201, 1202), IND
5 1200 I = MI(10,KO)
6 K = MAX(10,KO)
7 P = C(1,K)
8 NIT = NIL(1,K)
9 NI = NIT + 1
10 GO TO (1202, 1203, 1204, 1208, 1208, 1208, 1208, 1208), NI
11 1201 NI = NIT + 1
12 GO TO (1202, 1204, 1216, 1216, 1216, 1216), NI
C
C HARXA A PEU
C *****
C
13 1202 TERM = P
14 IF(IND.NE.3) RETURN
15 YY(1) = 1.
16 RETURN
C
C 1 SOL ITINERARI
C *****
17 1203 RC(1) = C(K,1)
18 NLIT(1) = NIL(K,1)
19 T = RC(1) + PE * (E + 2.*TA/(AUT*FLOAT(NLIT(1)+1)))
20 GO TO 1206
C
C REVISIO
C
21 1204 T = RC(1) + PE * (E + 2.*(TA+DB)/(AUT*FLOAT(NLIT(1)+1)))
C
C TEMPS I REPARTICIO
C
22 1206 X = AMINI(P,T) - HAC
23 S = 1. / (1./(T-X) + 1./(P-X))
24 TERM = S * (T/(T-X) + P/(P-X))
25 IF(IND.NE.3) RETURN
26 YY(1) = S / (T-X)
27 YY(2) = S / (P-X)
28 RETURN
C
C 2 O MES ITINERARIS
C *****
C
C EXTRACCIO
C

```

L

*ID, SEQ.

```

ISN * I SOURCE STATEMENT / ERROR MESSAGE
29 1208 R(I) = C(K,I)
30 NLIT(I) = NIL(K,I)
31 JIN = IS(I)
32 JSU = IS(I+1) - 1
33 IT = 1
34 DO 1212 J=JIN,JSU
35 IF(KS(J)-K) 1212, 1210, 1213
36 1210 IT = IT + 1
37 R(IT) = RS(J)
38 NLIT(IT) = NS(J)
39 1212 CONTINUE
40 1213 DO 1214 IT=1,NIT
41 1214 YIT(IT) = R(IT) + PE * (E + 2.*TA/(AUT*FLOAT(NLIT(IT)+1)))
42 GO TO 1221
C
C REVISIO
C
43 1216 DO 1220 J=1,NIT
44 1220 YIT(J) = R(J) + PE * (E + 2.*(TA*DB)/(AUT*FLOAT(NLIT(J)+1)))
C
C TEMPS I REPARTICAO
C
45 1221 X = P
46 DO 1222 J=1,NIT
47 IF(YIT(J).LT.X) X = YIT(J)
48 1222 CONTINUE
49 X = X - HAC
50 S = 1. / (P-X)
51 TERM = P / (P-X)
52 DO 1226 J=1,NIT
53 S = S + 1. / (YIT(J)-X)
54 TERM = TERM + YIT(J) / (YIT(J)-X)
55 TERM = TERM / S
56 IF(IND,NE,3) RETURN
57 S = 1. / S
58 YY(NIT+1) = S / (P-X)
59 DO 1228 J=1,NIT
60 YY(J) = S / (YIT(J) - X)
61 RETURN
62 END

```

8.1.5.16. Funció TOBUS.

Calcula el temps de recorregut entre dos nusos, damunt la línia que s'està tractant. De fet, el càlcul no és tan senzill com d'antuvi podria semblar, ja que, en el cas de línies circulars cal escatir per quin cantó o en quin sentit es fa el retorn. Aquesta complicació és la que explica l'existència d'una funció amb aquesta finalitat que, altrament no estaria justificada.

*ID. SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C *****
 C
 C TEMPS DE RECORREGUT SOBRE LA LINIA
 C *****

1 FUNCTION TOBUS(L, M)
 2 LOGICAL CIR, NUS
 3 COMMON /ETI 2/ LI(25), BU(25), CIR, NF, NO, NP, NUS(4,40)

C LINIES DRETES

4 TOBUS = ABS(BU(M) - BU(L))
 5 IF (.NOT. CIR) RETURN

C LINIES CIRCULARS

6 TOBUS = AMINI(TOBUS, BU(NF)-TOBUS)
 7 RETURN
 8 END

8.1.5.17. Funció EMA.

Es una funció d'accés a la matriu D, la qual conté la demanda de transport i el cost del camí mínim.

L'accés no és immediat, ja que segons el valor d'un paràmetre d'entrada caldrà buscar a la meitat superior o inferior. Això fa que hom li hagi donat forma de FUNCTION, en comptes de fer els càlculs cada cop que cal un valor de D.

75.09.22 PAGE 1

FACOM BOS2 FORTRAN -740601- V03 L08

*ID, SEQ.

ISN * I SOURCE STATEMENT / ERROR MESSAGE

C *****

C 6 666ES A LA MAT.-69U DE DEMANDA I TEMPS PER LA XARXA

C *****

1 FUNCTION EMA(N, IO, KO)
2 COMMON /ETI 1/ D(44,44), TA, AUT, NN, NOO, IENT, ISOR

C DETERMINACIO DELS SUBINDEXS

3 GO TO (1402, 1404), N

4 1402 I = MINO(10,KO)

5 K = MAXO(10,KO)

6 GO TO 1406

7 1404 I = MAXO(10,KO)

8 K = MINO(10,KO)

C ACCES AL VALOR

C 1406 EMA = D(I,K).

9 RETURN

10 END

11