

**Mètodes eficients per a la
resolució de problemes
de fluxos multiarticle**

Jordi Castro Pérez

Tesi Doctoral

**Mètodes eficients
per a la resolució
de problemes de
fluxos multiarticle**

**Autor: Jordi Castro
Director: Narcís Nabona**

Dept. d'Estadística i Investigació Operativa
Secció Informàtica
Universitat Politècnica de Catalunya

*A les persones que més estimo:
els meus pares, Ana i Segundo,
la Rosi,
i la Susanna.*

Agraïments

Moltes han estat les persones que durant aquests darrers anys m'han ajudat, directa o indirectament, en l'elaboració d'aquest treball. Pretendre enumerar-les a totes seria inviable, i, a més, ho trobaria injust, donat que de ben segur hi hauria noms que ometria per oblit. Només esmentaré a dues persones, que són alhora les que de més a prop han seguit l'evolució d'aquest treball, i les que més m'han hagut de "suportar": en Narcís Nabona, director del projecte de tesi, i en Javier Heredia, company durant aquests anys. M'agradaria que l'agraïment a ells dos es fes extensiu a la resta de persones que aquí no esmento.

Barcelona, juny de 1995.

Índex

1	Introducció	1
1.1.	Objectius	2
1.2.	Aportacions	3
1.3.	Esquema general de la memòria	4
2	Formulació	7
2.1.	Formulació del problema	7
2.2.	Aplicacions dels fluxos multiarticle	10
3	Estat de l'art	13
3.1.	Fluxos lineals multiarticle	13
3.1.1.	Mètodes clàssics	14
3.1.1.1.	Descomposició dictada per preus	14
3.1.1.2.	Descomposició dictada per recursos	15
3.1.1.3.	Particionament primal	16
3.1.2.	Mètodes de punt interior	16
3.1.3.	Altres mètodes	17
3.2.	Fluxos no lineals multiarticle	18
3.3.	Fluxos amb constriccions a banda	18
4	Fluxos lineals multiarticle per particionament primal	19
4.1.	Mètode del particionament primal	19
4.1.1.	Estructura del problema	19
4.1.2.	Motivació de la matriu de treball	22
4.1.2.1.	Càlcul de $Bx = b$	22
4.1.2.2.	Càlcul de $u^t B = c^t$	23
4.1.2.3.	Invertibilitat de Q	23
4.2.	Càlcul eficient de la matriu de treball	24
4.2.1.	Càlcul de Q_{cm}	26
4.2.2.	Càlcul de Q_{cb}	27
4.3.	Actualització eficient de la matriu de treball	28
4.3.1.	Tipus de pivotacions	29
4.3.2.	Mètode general	30
4.3.2.1.	Actualitzacions i recàlculs	31
4.3.2.2.	Matriu estesa	31
4.3.2.3.	Actualització de la matriu estesa	32
4.3.3.	Mètode general aplicat a la matriu Q en particular	35
4.3.3.1.	Conceptes previs	36
4.3.3.2.	Surt de la base una folga o arc complementari	38
4.3.3.3.	Surt de la base un arc del i -èssim arbre d'expansió	43
4.4.	Implementació del particionament primal	48

4.4.1.	Fases de l'algorisme desenvolupat	48
4.4.1.1.	Fase 0	48
4.4.1.2.	Fase 1	49
4.4.1.3.	Fase 2	50
4.4.2.	Procediment de taxació	50
4.4.3.	Tractament de la matriu de treball	52
4.4.4.	Estructures de dades	53
5	Fluxos no lineals multiarticle per particionament primal	57
5.1.	Optimització no lineal a gran escala amb constriccions lineals	57
5.1.1.	Conceptes previs	57
5.1.2.	Optimització al subespai actual	58
5.1.3.	Determinació de la matriu de constriccions actives òptima	60
5.1.4.	Algorisme del conjunt de constriccions actives	61
5.2.	Particionament primal per a problemes no lineals	62
5.3.	Implementació de l'algorisme	64
5.3.1.	Fases de l'algorisme desenvolupat	64
5.3.2.	Càlcul de la direcció de descens	65
5.3.2.1.	Mètode de Newton-truncat	65
5.3.2.2.	Mètode quasi-Newton	67
5.3.3.	Test d'optimalitat al subespai actual	70
5.3.4.	Elecció de la variable no bàsica que esdevindrà superbàsica	71
5.3.5.	Variables superbàsiques quasi-actives	73
5.3.6.	Estructures de dades	74
6	Experiència computacional amb el particionament primal	77
6.1.	Breu descripció del paquet PPRN	77
6.2.	Problemes test usats	80
6.2.1.	Problemes test lineals	80
6.2.1.1.	Problemes lineals obtinguts de generadors automàtics	81
6.2.1.2.	Problemes lineals de coordinació hidrotèrmica	88
6.2.2.	Problemes test no lineals	92
6.2.2.1.	Problemes artificials	92
6.2.2.2.	Problemes no lineals de coordinació hidrotèrmica	93
6.2.2.3.	Problema d'assignació de trànsit	94
6.3.	Resultats computacionals	95
6.3.1.	Resultats computacionals amb els problemes test lineals	95
6.3.1.1.	Anàlisi dels resultats obtinguts	100
6.3.2.	Resultats computacionals amb els problemes test no lineals	103
6.3.2.1.	Anàlisi dels resultats obtinguts	107
7	Algorisme primal-dual de punt interior	109
7.1.	Preliminars	109

7.2.	L'algorisme primal-dual considerant fites superiors a algunes variables	110
7.2.1.	Formulació dels problemes primal i dual	110
7.2.2.	Obtenció dels Lagrangians a través una barrera logarísmica	111
7.2.3.	Condicions d'optimalitat de Karush-Kuhn-Tucker de primer ordre	112
7.2.4.	Solució del sistema no lineal	114
7.2.5.	Actualització del nou punt i del paràmetre μ de penalització	117
7.3.	Implementació de l'algorisme	118
7.3.1.	Obtenció de dy a cada iteració	118
7.3.2.	Elecció del punt inicial	121
7.3.3.	Càlcul de les passes α_p i α_d	121
7.3.4.	Actualització del paràmetre μ	122
7.3.5.	Condicions d'aturada	122
7.4.	Resultats computacionals	123
8	Especialització d'un mètode primal-dual per a fluxos multiarticle	129
8.1.	Formulació del problema considerat	129
8.1.1.	Tractament dels arcs	130
8.1.2.	Tractament de les folgues de les constriccions de capacitat mútua	131
8.1.3.	Tractament de les folgues de les constriccions a banda	132
8.1.4.	Formulació del problema	133
8.2.	Solució de (FMLCB) a través de l'algorisme primal-dual	133
8.2.1.	Aplicació de la descomposició de dominis a la factorització de $\tilde{A}S\tilde{A}^t$	137
8.2.2.	Gradient conjugat preconditionat	139
8.2.3.	Precondicionament per sèries truncades de potències	141
8.3.	Implementació de l'algorisme primal-dual per a problemes multiarticle	145
9	Experiència computacional amb el mètode primal-dual multiarticle	149
9.1.	Aspectes del codi desenvolupat	149
9.2.	Problemes test usats	150
9.3.	Resultats computacionals	150
9.4.	Comparació de IPM amb codis alternatius	154
10	Conclusions i àrees de futura recerca	157
10.1.	Conclusions	157
10.2.	Ampliacions i futura recerca	159
10.3.	Publicacions conseqüència del treball de tesi	160
	Bibliografia	163
	Glossari de símbols	169
	Índex de matèries	173

Índex d'algorismes

Algorisme 4.1 : Càlcul de Q_{cm}	27
Algorisme 5.1 : Mètode del conjunt de constriccions actives.	61
Algorisme 5.2 : Mètode de Newton-truncat per solucionar $Z^t H Z p_S = -g_z$	65
Algorisme 5.3 : Elecció de la variable no bàsica que esdevindrà superbàsica.	72
Algorisme 6.1 : Conversió de problemes DIMACS uniarticle a multiarticle.	81
Algorisme 7.1 : Càlcul de $\tilde{A} S \tilde{A}^t$	120
Algorisme 8.1 : Solució de $\hat{D} dy_2 = \hat{\beta}_2$ pel mètode del gradient conjugat preconditionat.	140
Algorisme 8.2 : Producte de $\hat{D} = D - C^t G^{-1} C$ per un vector $\hat{x} \in \mathbb{R}^{n+t}$ ($\hat{y} := \hat{D} \hat{x}$).	141
Algorisme 8.3 : Solució de $z := M^{-1} r$	144

Índex de figures

Figura 2.1 : Xarxa del problema de l'emmagatzemament estacional.	12
Figura 6.1 : Exemple d'un sistema amb quatre embassaments.	88
Figura 6.2 : Xarxa replicada multiarticle amb els fluxos d'aigua.	89
Figura 6.3 : Representació de les funcions ISR i DSR.	91
Figura 6.4 : Representació de la xarxa de les variables de generació tèrmica.	91
Figura 7.1 : Temps de CPU per a IP, LoQo i MINOS segons la mida del problema.	127
Figura 7.2 : Temps per iteració per a IP i LoQo segons la mida del problema.	127
Figura 9.1 : Nombre d'iteracions del GCP per iteració primal-dual del problema L_3^4	153
Figura 9.2 : Nombre d'iteracions del GCP per iteració primal-dual del problema L_8^1	154

Índex de taules

Taula 6.I : Problemes generats amb Rmfgn.	83
Taula 6.II : Problemes generats amb Grid-on-torus.	84
Taula 6.III : Problemes generats amb Gridgraph.	85
Taula 6.IV : Problemes generats amb Gridgen.	86
Taula 6.V : Paràmetres usats en la generació de problemes amb Mnetgen.	87
Taula 6.VI : Problemes generats amb Mnetgen.	87
Taula 6.VII : Problemes lineals de coordinació hidrotèrmica a llarg termini.	90
Taula 6.VIII : Problemes lineals de coordinació hidrotèrmica a curt termini.	92
Taula 6.IX : Problemes no lineals artificials i de coordinació hidrotèrmica a llarg termini.	93
Taula 6.X : Problemes no lineals de coordinació hidrotèrmica a curt termini.	94
Taula 6.XI : Problema d'assignació de trànsit.	94
Taula 6.XII : Resultats obtinguts amb els problemes Rmfgn.	96
Taula 6.XIII : Resultats obtinguts amb els problemes Grid-on-torus.	97
Taula 6.XIV : Resultats obtinguts amb els problemes Gridgraph.	98
Taula 6.XV : Resultats obtinguts amb els problemes Gridgen.	98
Taula 6.XVI : Resultats obtinguts amb els problemes Mnetgen.	99
Taula 6.XVII : Resultats obtinguts amb els problemes de coordinació hidrotèrmica.	100
Taula 6.XVIII : Comparació de temps entre PPRN i MINOS, MCNF85 i LoQo.	101
Taula 6.XIX : Resultats obtinguts amb la funció artificial $f_1(x)$	104
Taula 6.XX : Resultats obtinguts amb la funció artificial $f_2(x)$	104
Taula 6.XXI : Resultats obtinguts amb la funció artificial $f_3(x)$	105
Taula 6.XXII : Resultats amb els problemes de coord. hidrot. a llarg termini.	105
Taula 6.XXIII : Resultats amb els problemes de coord. hidrot. a curt termini.	106
Taula 6.XXIV : Resultats obtinguts amb el problema d'assignació de trànsit.	106
Taula 6.XXV : Comparació de temps entre PPRN i MINOS.	107
Taula 7.I : Característiques dels problemes test.	124
Taula 7.II : Resultats obtinguts amb els problemes test.	125
Taula 9.I : Resultats obtinguts amb els problemes Rmfgn (usant IPM).	151
Taula 9.II : Resultats obtinguts amb els problemes Grid-on-torus (usant IPM).	151
Taula 9.III : Resultats obtinguts amb els problemes Gridgraph (usant IPM).	152
Taula 9.IV : Resultats obtinguts amb els problemes Gridgen (usant IPM).	152
Taula 9.V : Comparació entre IPM i LoQo.	155
Taula 9.VI : Comparació entre IPM i PPRN.	156

Capítol 1

Introducció.

Tal i com indica el seu títol, la present memòria de tesi té com objecte d'estudi el desenvolupament d'algorismes i implementacions eficients per resoldre el problema conegut dins del món de l'Optimització i Investigació Operativa com a problema de *fluxos multiarticle en xarxa*. Com es desprèn del seu nom, és un problema d'optimització en xarxa on, a diferència del problema clàssic uniarticle, diversos productes (els articles) comparteixen el mateix canal físic (la xarxa) sense poder ser combinats entre ells. Això provoca que sigui necessària una replicació de les variables —fluxos en cada arc— de la xarxa original, tantes vegades com articles hi ha, el qual incrementa considerablement el nombre de variables i restriccions a ser tractades.

L'optimització de fluxos en xarxes multiarticle és un problema ben conegut, i durant anys diverses aplicacions reals han estat modelitzades mitjançant aquesta tècnica. Alhora, és un problema molt costós des d'un punt de vista computacional, donat el gran nombre de variables i equacions que intervenen com abans s'ha esmentat. Aquest doble fet (el seu interès per solucionar problemes reals i el seu elevat cost) ha motivat l'estudi i desenvolupament de tècniques per tractar-lo de forma específica. Sovint, però, aquest esforç ha conduït a la formulació de diversos algorismes però rarament ha donat lloc a implementacions de caràcter pràctic.

Amb això que acabem de dir queda clar que suposarem que hi ha una clara distinció entre el que és algorisme i el que és implementació, tot i que sovint és difícil decidir on hi ha la frontera entre un i altre concepte. En general, per algorisme entendrem la seqüència de processos que s'han de seguir per tal d'aconseguir un cert objectiu, mentre que la implementació fa referència a com, en última instància, s'han dut a la pràctica. Des d'aquest punt de vista es podria dir que un algorisme té moltes implementacions, però que una implementació només correspon a un únic algorisme.

S'ha cregut convenient fer una breu discussió sobre els conceptes d'algorisme i implementació, donat que al treball aquí presentat ambdós tenen un pes específic. Al treball desenvolupat no s'ha pretès només detallar com modificar, ampliar o obtenir algorismes per solucionar el problema de fluxos multiarticle, sinó que a més s'ha plantejat com objectiu prioritari l'obtenció d'implementacions el més eficients i robustes possibles. Vista així, la tasca desenvolupada es troba a cavall entre dos camps com ara la Investigació Operativa i la Informàtica. També cal tenir present que en aquesta memòria es presentarà detalladament tot el que faci referència a la part algorísmica, però no s'aprofundirà tant a nivell d'implementació. Això implicaria haver de descriure una gran quantitat d'estructures de dades i una gran quantitat de rutines desenvolupades i altres usades de llibreries numèriques estàndard. Tanmateix, això no ha de fer oblidar que la major part de l'esforç necessari ha

estat dedicat a l'obtenció i disseny d'aquestes rutines i estructures de dades. I el fet de que tot aquest treball quedi plasmat en només l'obtenció d'una sèrie de taules amb valors numèrics no ha de fer oblidar la gran quantitat de feina no descrita que hi ha darrera d'aquests resultats.

Un cop s'ha definit el marc on es troba el treball realitzat, procedirem a definir amb més detall els objectius que s'han perseguit i les aportacions que representa respecte el que fins ara s'havia fet. Finalitzarem la introducció amb una breu descripció sobre com ha estat estructurada aquesta memòria.

1.1 Objectius.

Ja s'ha comentat anteriorment que l'objecte d'estudi del treball realitzat són les tècniques de fluxos multiarticle, i que el problema ha estat abordat des d'una òptica força aplicada. Cal ara, però, detallar de forma més precisa quins són els objectius que s'han perseguit. L'ordre en que es presenten correspon a la seqüència temporal de tasques que han estat realitzades.

- Tots els algorismes especialitzats per a problemes de fluxos multiarticle descrits a la literatura consideren el problema en la seva forma més simple, on es disposa de funció objectiu lineal, constriccions de xarxa i límits simples a les variables. Tanmateix per moltes aplicacions reals cal considerar variants d'aquesta formulació inicial. Una d'aquestes és l'addició de constriccions lineals, a banda de les que ja hi ha de xarxa. La primera tasca a fer, doncs, consisteix en l'obtenció d'un algorisme per a fluxos multiarticle que suporti constriccions a banda lineals. Cal dir que altres autors ja han desenvolupat algorismes específics per constriccions a banda, però sempre considerant només problemes de xarxes amb un únic article. Des d'aquest punt de vista, l'objectiu pretès consistirà, en certa manera, en combinar en un únic algorisme dues tècniques, una especialitzada per a fluxos multiarticle i una per a problemes uniarticle amb constriccions a banda.
- Una altra de les mancances dels algorismes especialitzats de fluxos multiarticle descrits fins el moment, és el fet de només tractar problemes amb funció objectiu lineal. L'algorisme prèviament desenvolupat per tractar problemes multiarticle amb constriccions a banda pot ser adaptat per tal de suplir aquesta absència. Per dur a terme aquesta tasca, es considerarà un mètode capdavanter d'optimització no lineal per a problemes de gran escala, i serà introduït dins l'algorisme multiarticle amb constriccions a banda previ. Els dos objectius fins aquí esmentats es podrien considerar com variants de mètodes "clàssics" de programació lineal i no lineal.
- En els darrers anys els mètodes anomenats de punt interior s'han consolidat com a tècniques molt efectives per solucionar problemes de programació lineal (fins i tot quadràtica). Per tant, es pretén especialitzar un mètode de punt interior per tal de

tractar problemes de fluxos multiarticle amb constriccions a banda. Aquesta especialització, però, només es farà considerant funcions objectiu lineals.

- A part del desenvolupament dels algorismes detallats als tres punts previs, la implementació concreta de cada un d'ells esdevé un objectiu primordial. A més, es pretén obtenir implementacions el més robustes i eficients possibles, el qual implica l'ús de llenguatges de tercera generació en comptes de paquets matemàtics (numèrics i simbòlics) d'alt nivell. Aquestes implementacions han de ser capaces de tractar problemes de gran escala.
- Un cop desenvolupats els codis, resulta possible comparar el seu rendiment respecte d'altres paquets d'optimització. Així mateix, el fet d'haver desenvolupat implementacions multiarticle usant tècniques clàssiques i de punt interior, permet comparar l'eficiència relativa d'unes respecte les altres.

1.2 Aportacions.

Donada la vessant aplicada del treball de tesi realitzat, la majoria d'innovacions recauen dins del camp de la implementació de software numèric. En alguns casos les implementacions realitzades constitueixen una millora respecte codis previs desenvolupats per altres autors, mentre que en altres representen el primer intent d'abordar numèricament i de forma específica certs mètodes. Naturalment, les implementacions fetes no són ni molt menys definitives, i, probablement, són millorables considerant d'altres enfoc més escaients. Tanmateix, el treball realitzat constitueix un primer pas vers la consecució d'implementacions el més eficients possibles.

Des del punt de vista algorísmic, cal fer una separació clara entre el treball fet relacionat amb tècniques clàssiques i les de punt interior. Pel que fa a les primeres, la tasca realitzada ha estat una ampliació i combinació de mètodes especialitzats ja existents de programació lineal i no lineal. Bàsicament, l'aportació més novedosa està lligada a l'àlgebra del càlcul i actualització d'una matriu de treball usada al llarg de tot l'algorisme. Pel que fa a l'aplicació de tècniques de punt interior, el mètode multiarticle desenvolupat ha estat dissenyat per complet, tot i que presenta algunes similituds amb enfoc proposats per altres autors. Aquests, però, en cap moment proporcionaven una forma raonable per solventar el problema de forma numèrica. Per la seva banda, l'algorisme aquí presentat solventa aquesta dificultat, i, de fet, la implementació realitzada s'ha mostrat força eficient per segons quins tipus de problemes.

Concretant més, podem desglossar les aportacions d'aquest treball en els següents punts:

- Ampliació de l'algorisme anomenat del *particionament primal* (especialització del simplex per resoldre problemes de fluxos multiarticle) per tal de considerar constriccions a banda lineals i funcions no lineals.

- Implementació de l'algorisme de particionament primal per a fluxos multiarticle amb constriccions a banda lineals per a funcions lineals i no lineals. Resolució de problemes de gran escala —provinents d'aplicacions reals i obtinguts per generadors automàtics— tant per al cas lineal com per al no lineal.
- Desenvolupament d'un algorisme de punt interior per resoldre problemes de fluxos multiarticle amb constriccions a banda lineals.
- Implementació de l'algorisme de punt interior especialitzat per a problemes de fluxos multiarticle, i resolució de problemes de gran escala.
- Comparació de l'actuació dels codis especialitzats desenvolupats amb altres codis existents d'ús general i especialitzats.
- Estudi comparatiu dels mètodes especialitzats de punt interior respecte els mètodes especialitzats basats en el símplex en la resolució de problemes multiarticle.

Cal dir que en parlar de resolució de problemes de gran escala s'està fent referència a problemes que en el cas lineal poden tenir fins a 150000 variables i 45000 constriccions, i fins a 17000 variables i 4800 constriccions per a funcions no lineals. La considerable mida d'aquests problemes obliga a la realització d'implementacions força acurades i eficients, per tal de poder obtenir resultats satisfactoris en un temps raonable.

1.3 Esquema general de la memòria.

S'ha pretès que la present memòria fos el més autocontinguda possible, dins d'uns marges raonables. Per tant, s'intentarà fer un breu esment d'aquelles aportacions prèvies d'altres autors que siguin rellevants. I pel que fa a aquells aspectes que no es considerin clau, tot i ser usats dins d'aquest treball, només seran referenciats i s'adreçarà el lector cap a textos més especialitzats.

Podem considerar dividida la present memòria en quatre grans blocs. Aquests són:

- Bloc 1. (Capítols 2 i 3) Aquest primer bloc es dedica a la presentació del problema de fluxos multiarticle, detallant la seva formulació, les àrees d'aplicacions d'aquests i les contribucions prèvies d'altres autors (estat de l'art).
- Bloc 2. (Capítols 4, 5 i 6) Aquest segon bloc detalla la tasca realitzada en el desenvolupament i implementació d'un mètode per resoldre el problema de fluxos multiarticle considerant constriccions a banda lineals i funció objectiu no lineal. El mètode presentat és una extensió d'un ja existent i conegut com a *mètode del particionament primal*, una especialització del símplex per a fluxos multiarticle. Es presenten els resultats computacionals obtinguts amb el codi desenvolupat, i la comparació del seu rendiment respecte codis alternatius.

- Bloc 3. (Capítols 7, 8 i 9) En aquest tercer bloc es presenta un algorisme de punt interior que considera problemes amb fites superiors a les variables, i la seva especialització per tal de resoldre problemes multiarticle. S'adjunten els resultats obtinguts amb el codi especialitzat desenvolupat, tot comparant-lo amb implementacions de punt interior de caràcter general. També es fa una comparació de les dos especialitzacions multiarticle realitzades dins d'aquest treball (la basada en el mètode de particionament primal, i aquesta de punt interior).
- Bloc 4. (Capítol 10) El darrer capítol, a mode de conclusió, es dedica a destacar les aportacions més significatives del treball presentat i a exposar les possibles extensions i línies de futura recerca.

Durant tot el document, les lletres majúscules s'usaran per indicar matrius, mentre que les minúscules denotaran escalars o vectors (en tot moment es deixarà clar quan fem referència a uns o altres). Els vectors es consideraran vectors columna, i un vector fila es referirà com a vector transposat. Els conjunts (d'índexs, de constriccions, etc.) es denotaran per lletres majúscules calligràfiques (p.e., \mathcal{A}). La matriu identitat de dimensió n es representarà per $\mathbb{1}_n$. Anàlogament, la matriu zero (amb tots els elements nuls) de dimensió n es representarà per $\mathbf{0}_n$.

Capítol 2

Formulació.

Els problemes de fluxos multiarticle apareixen en aplicacions on diversos articles (com ara vehicles o missatges) comparteixen la capacitat dels arcs d'una única xarxa per anar des d'uns nusos origen a uns altres nusos destí. Als nusos de la xarxa s'ha de satisfer un balanç d'entrades i sortides específic per a cada article. Un clar exemple apareix en el camp de les telecomunicacions, on es pot considerar com un article el conjunt de trucades que es fan entre dos punts qualsevols, els quals pertanyen a una determinada xarxa telefònica. Si els fluxos d'un determinat article no influeixen en els fluxos dels demés articles el problema pot ser formulat com k problemes independents de fluxos uniarticle (suposem que k és el nombre d'articles total). Però si, degut a una limitació de capacitat de l'arc per al conjunt dels articles, dita capacitat mútua, els fluxos en els arcs d'un article limiten la capacitat per a la resta dels articles, llavors ens trobem davant d'un problema on hi ha un acoblament entre els k problemes uniarticle anteriors. El problema així plantejat correspon a la formulació clàssica dels fluxos multiarticle. Tanmateix, aquesta formulació inicial pot ser ampliada per tal de tractar un ventall més ampli de problemes. En aquest capítol detallarem la formulació específica amb que es treballarà durant la resta de la memòria. Així mateix, es farà un breu esment de possibles camps d'aplicació dels fluxos multiarticle.

2.1 Formulació del problema.

Considerem un graf dirigit $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, on \mathcal{V} és un conjunt de m nodes i \mathcal{E} és un conjunt de n arcs. Suposem, a més, que disposem d'un conjunt \mathcal{K} d'articles, de cardinalitat $k = |\mathcal{K}|$. El problema de fluxos multiarticle amb funció objectiu lineal —que anomenarem **(FML)**— pot ser llavors formulat com:

$$\min_{x^{(1)}, x^{(2)}, \dots, x^{(k)}} \sum_{i=1}^k c^{(i)t} x^{(i)} \quad (2.1)$$

$$\text{(FML)} \quad \text{subj. a} \quad \left. \begin{aligned} Ax^{(i)} &= b^{(i)} \\ \underline{0} \leq x^{(i)} &\leq \bar{x}^{(i)} \end{aligned} \right\} i = 1, \dots, k \quad (2.2)$$

$$\sum_{i=1}^k x^{(i)} \leq b_{cm} \quad (2.4)$$

on:

- $x^{(i)} \in \mathbb{R}^n$ és el vector de fluxos de l'article i .
- $c^{(i)} \in \mathbb{R}^n$ és el vector de costos lineals per als fluxos de l'article i .

- $A \in \mathbb{R}^{m \times n}$ és la matriu d'incidències entre arcs i nodes, que representa la topologia del graf \mathcal{G} . Sense pèrdua de generalitat, podem considerar que el darrer arc és un arc arrel afegit per garantir que $\text{rang}(A) = m$.
- $b^{(i)} \in \mathbb{R}^m$ és el vector amb les injeccions per a l'article i a cada node de la xarxa. Si $b_j^{(i)} > 0$, el node j és un punt de subministrament per a l'article i . Si $b_j^{(i)} < 0$, el node j és un punt de demanda per a l'article i . Si $b_j^{(i)} = 0$, el node j és un punt de traspàs per a l'article i (tot el que entra de l'article i n'ha de sortir).
- $b_{cm} \in \mathbb{R}^n$ és el vector amb les capacitats mútues (capacitat total per a tots els articles considerats conjuntament) de cada arc.
- $\bar{x}^{(i)} \in \mathbb{R}^n$ és el vector amb les capacitats de cada arc per a l'article i . Generalment s'ha de verificar que $\bar{x}_j^{(i)} \leq b_{cm_j}$. En molts casos, però, es té que $\bar{x}_j^{(i)} = b_{cm_j}$.
- $\underline{0} \in \mathbb{R}^n$ és el vector amb els límits inferiors dels fluxos per a tots els articles (es considera, doncs, que aquest flux mínim és sempre 0).

S'observa com el problema **(FML)** té, a més dels límits simples de les variables (2.3), dos grups d'equacions. Per una banda hi ha les equacions de xarxa (2.2), i en segon lloc les constriccions de capacitat mútua (2.4). Tenint en compte l'estructura de la matriu A la qual representa la topologia de la xarxa, les equacions (2.2) poden detallar-se com:

$$\sum_{(v,w) \in \mathcal{E}} x_{(v,w)}^{(i)} - \sum_{(w,v) \in \mathcal{E}} x_{(w,v)}^{(i)} = b_v^{(i)} \quad \forall v \in \mathcal{V} \quad i = 1, \dots, k \quad (2.5)$$

El problema **(FML)** podria ser considerat com la formulació clàssica dels problemes de fluxos multiarticle, i és la que més sovint apareix en la literatura [2,47,49]. Aquesta formulació també sovint és anomenada *formulació arc-node*, en contraposició a altres formulacions alternatives on les variables no són arcs de la xarxa sinó camins entre els nodes (*formulació arc-camí*). D'altres autors [42,43] consideren formulacions on el nombre d'articles k va associat a parells de nodes (s_i, t_i) , $i = 1, \dots, k$, de forma que a s_i es subministren $b_{s_i}^{(i)}$ unitats de l'article i , les quals han d'arribar al node t_i (és a dir, el vector $b^{(i)}$ només tindria dos components diferents de 0, una associada al node s_i , amb valor $b_{s_i}^{(i)}$, i l'altra al node t_i , amb valor $-b_{s_i}^{(i)}$). El problema **(FML)** abans formulat és més general, donat que el nombre de nodes que produeixen o als quals arriba flux de l'article i no està limitat a un de sol. De fet és possible transformar el problema **(FML)** a una forma alternativa on cada article es trobi associat a un parell de nodes (s_i, t_i) , només afegint arcs amb capacitat. No és possible, però, fer la transformació de l'altra formulació a **(FML)**.

El problema **(FML)** ha estat ampliat en aquest treball de dos formes diferents. La primera variant consisteix en considerar problemes on la funció objectiu sigui de la forma

$$f(x^{(1)}, x^{(2)}, \dots, x^{(k)}) : \mathbb{R}^{k \times n} \rightarrow \mathbb{R} \quad f \in C^2 \quad (2.6)$$

El problema de fluxos multiarticle amb funció objectiu no lineal **(FMN)** queda llavors formulat com:

$$\begin{aligned}
 \text{(FMN)} \quad & \min_{x^{(1)}, x^{(2)}, \dots, x^{(k)}} \quad (2.6) \\
 & \text{subj. a} \quad (2.2) - (2.4)
 \end{aligned}$$

La segona ampliació que s'ha fet a la formulació original ha estat afegir constriccions a banda lineals definides per:

$$\underline{b}_{cb} \leq \sum_{i=1}^k T^{(i)} x^{(i)} \leq \bar{b}_{cb} \quad (2.7)$$

on $T^{(i)} \in \mathbb{R}^{t \times n}$, i $\underline{b}_{cb}, \bar{b}_{cb} \in \mathbb{R}^t$, essent t el nombre de constriccions a banda. Afegint (2.7) a **(FML)** i **(FMN)** obtenim respectivament el problema de fluxos multiarticle lineals amb constriccions a banda lineals **(FMLCB)**, i el problema de fluxos multiarticle no lineals amb constriccions a banda lineals **(FMNCB)**. Considerant les folgues $s_{cm} \in \mathbb{R}^n$ per a les constriccions de capacitat mútua (2.4), i $s_{cb} \in \mathbb{R}^t$ per a les constriccions a banda (2.7), podem formular **(FMLCB)** i **(FMNCB)** en la forma final en que han estat considerats al llarg de tot aquest treball:

$$\begin{aligned}
 & \min_{x^{(1)}, \dots, x^{(k)}, s_{cm}, s_{cb}} \quad (2.1)/(2.6) \\
 & \text{subj. a} \quad (2.2), (2.3) \\
 \text{(FMLCB)/(FMNCB)} \quad & \sum_{i=1}^k x^{(i)} + s_{cm} = b_{cm} \quad (2.8)
 \end{aligned}$$

$$\sum_{i=1}^k T^{(i)} x^{(i)} + s_{cb} = \bar{b}_{cb} \quad (2.9)$$

$$\underline{0} \leq s_{cm} \quad \underline{0} \leq s_{cb} \leq \bar{b}_{cb} - \underline{b}_{cb} \quad (2.10)$$

El problema resultant d'optimització té un total de $\tilde{n} = kn + n + t$ variables i $\tilde{m} = km + n + t$ constriccions. La matriu \tilde{A} de constriccions del nostre problema, exceptuant els límits simples, queda formada per les equacions (2.2), (2.8) i (2.9) i té la següent estructura:

$$\tilde{A} = \begin{array}{|c|c|c|c|c|c|}
 \hline
 A & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 \hline
 \mathbf{0} & A & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
 \hline
 \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 \hline
 \mathbf{0} & \mathbf{0} & \dots & A & \mathbf{0} & \mathbf{0} \\
 \hline
 \mathbf{1} & \mathbf{1} & \dots & \mathbf{1} & \mathbf{1} & \mathbf{0} \\
 \hline
 T^{(1)} & T^{(2)} & \dots & T^{(k)} & \mathbf{0} & \mathbf{1} \\
 \hline
 \end{array} \quad (2.11)$$

El sistema d'equacions definit per \tilde{A} és llavors:

$$\begin{array}{|c|c|c|c|c|c|} \hline A & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & A & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \hline \mathbf{0} & \mathbf{0} & \dots & A & \mathbf{0} & \mathbf{0} \\ \hline \mathbb{1} & \mathbb{1} & \dots & \mathbb{1} & \mathbb{1} & \mathbf{0} \\ \hline T^{(1)} & T^{(2)} & \dots & T^{(k)} & \mathbf{0} & \mathbb{1} \\ \hline \end{array}
 \begin{array}{c} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(k)} \\ s_{cm} \\ s_{cb} \end{array}
 =
 \begin{array}{c} b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(k)} \\ b_{cm} \\ \bar{b}_{cb} \end{array}
 \quad (2.12)$$

2.2 Aplicacions dels fluxos multiarticle.

Els problemes de fluxos en xarxes multiarticle apareixen dins una gran varietat de camps d'aplicació, tant per a problemes lineals [2,47] com no lineals [22,62]. A continuació es fa un breu esment d'algunes aplicacions. Per a un major detall adrecem al lector als textos referenciats.

- 1.- *Xarxes de comunicacions* [82]. En una xarxa de comunicacions els nodes representen estacions emissores i receptors de missatges, mentre que els arcs corresponen a les línies de transmissió. Al conjunt de missatges entre cada parell de nodes emissor/receptor se li associa un article. Tots els articles comparteixen les línies de transmissió, les quals tenen una capacitat fixada, i per tant cal imposar constriccions de capacitat mútua. Associant un cost al pas d'un missatge per cada línia de transmissió, el problema consisteix en trobar la distribució de missatges per la xarxa que minimitza el cost de transmissió total. Aquest tipus de problemes apareixen en entorns tan diversos com poden ser xarxes telefòniques i xarxes d'ordinadors. Sovint el nombre d'articles acostuma a ser gran (un per cada parell emissor/receptor), mentre que la mida de la xarxa pot ser moderada.
- 2.- *Transport de mercaderies per una flota de vehicles* [20,31]. En aquest problema els nodes representen punts de producció i consum de diverses mercaderies (els articles), mentre que els arcs corresponen als vehicles que les comuniquen. El cost associat al transport de cada mercaderia és diferent per a cada una. Els vehicles que han de fer el transport tenen una capacitat màxima de càrrega (constriccions de capacitat mútua). El problema consisteix en decidir la càrrega de cada article en cada vehicle per tal de satisfer el subministrament dels productes anteriors entre parells de nodes en un període de temps determinat. Aquest tipus de problemes apareixen tan a un nivell local (transport regional de certes mercaderies) com a un internacional (per exemple, es pot considerar el problema d'exportació i importació de cereals entre diversos països).

- 3.- *Assignació de trànsit estàtic* [7]. Donat un conjunt de destins i orígens dins una xarxa de comunicació urbana, es considera com a articles el flux de trànsit comunicant parells origen-destí. Els arcs de la xarxa (els carrers) tenen un cost (no lineal) en funció del trànsit que suporten: com més carregat va l'arc, major és el seu cost. El problema consisteix en trobar la distribució de trànsit que minimitza el cost total, satisfent l'entrada/sortida de vehicles als nodes origen/destí. La capacitat dels carrers ens proporciona les constriccions de capacitat mútua. En aquest tipus de problemes tant el nombre d'articles com la dimensió de la xarxa acostuma a ser gran. Per tal de reduir la dimensionalitat del problema resultant poden agrupar-se els articles, bé per nodes orígens, bé per nodes destins. Cal remarcar que en aquest tipus de problemes la funció objectiu és no lineal.
- 4.- *Optimització de la generació d'electricitat hidràulica* [62]. En aquest cas es considera una xarxa hidràulica formada per diversos embassaments, replicada al llarg d'un període de temps tantes vegades com intervals es vulguin considerar. Els nodes representen embassaments i els arcs poden significar, o bé volums d'embassaments, o bé vessaments i descàrregues d'aigua, produint aquestes darreres una determinada generació d'electricitat. Les injeccions de flux als nodes representen aportacions estocàstiques d'aigua, les quals es modelitzen aproximant la distribució real d'aportacions mitjançant una distribució per blocs, obtenint aigües de probabilitat diferent que seran els articles del nostre problema. Els volums màxims dels embassament i el màxim permès que es pot descarregar ens proporciona les constriccions de capacitat mútua. El problema consisteix en trobar la distribució òptima de recursos hidràulics per tal de garantir una determinada generació d'electricitat amb un cost mínim. En aquest cas la funció objectiu torna a ser (força) no lineal. Aquest tipus de problemes acostuma a tenir xarxes molt grans (degut a que hi ha una rèplica de la xarxa original tantes vegades com intervals considerem), però, per altra banda, el nombre d'articles acostuma a ser petit (quatre, generalment).
- 5.- *Planificació del recorregut de camions cisterna* [8]. En aquest cas es disposa de diferents tipus de camions cisterna (cada tipus serà un article), els quals han de fer un cert repartiment de productes (els repartiments es troben associats a arcs en aquest model) en unes dates concretes. El problema consisteix a coneixer quin tipus de camions s'usaran, i el seu número, per tal de poder fer tot el repartiment de productes de forma òptima (per exemple, usant el mínim nombre possible de camions cisterna, o bé, si s'associa un cost a cada camió en funció del seu tipus, minimitzant la funció de cost resultant). Per més detalls sobre la xarxa obtinguda adrecem al lector interessat a l'article abans referenciat. D'aquest problema existeix una versió uniarticle també, la qual es soluciona aplicant tècniques de *max-flow*. Cal tenir en compte que aquest model requereix integritat de les variables (ja que volem conèixer un nombre necessari de camions), requisit que les tècniques de fluxos multiarticle no garanteixen. Tanmateix, la solució fraccionària trobada pot usar-se com a fita inferior de l'òptim enter, i, fins i tot, per aproximar bones solucions enteres.

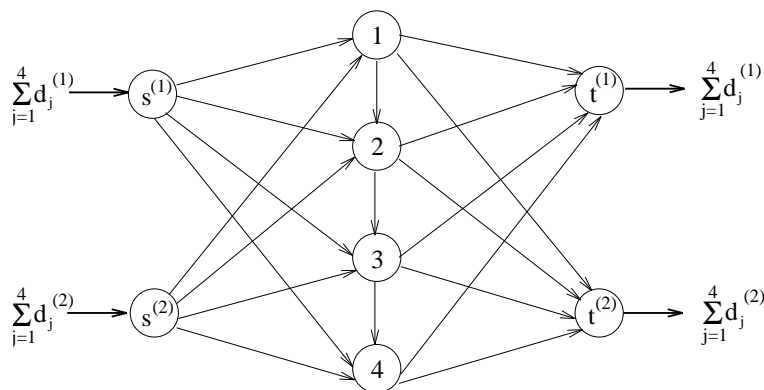


Figura 2.1. Xarxa del problema de l'emmagatzemament estacional.

- 6.- *Emmagatzemament de productes estacionals* [41]. El problema consisteix en decidir quan cal produir i quan emmagatzemar una sèrie de productes (els articles) per tal de satisfer una demanda estacional d'ells (estacional vol dir cada setmana, cada mes, cada trimestre ...). Per entendre millor el problema considerarem un exemple on disposem d'un únic magatzem de capacitat M , i de dos productes, la demanda trimestral dels quals al llarg d'un any és de $d_j^{(i)}$, $i = 1, 2$, $j = 1, \dots, 4$. La capacitat de producció del producte i al trimestre j és de $u_j^{(i)}$, el cost de produir una unitat és de $c_j^{(i)}$, i el cost d'emmagatzemar una unitat de producte del trimestre j al $j+1$ és de $h_j^{(i)}$. La Fig. 2.1 mostra la xarxa associada al problema que estem tractant. Els dos nodes $s^{(i)}$ són els centres productors, mentre que $t^{(i)}$ són els centres de consum. Els nodes j , $j = 1, \dots, 4$ representen el magatzem al llarg dels quatre trimestres del nostre període d'estudi (un any en aquest cas). Els arcs $(s^{(1)}, j)$ només poden transportar unitats de l'article 1, i llavors es té que la capacitat de les variables $(s^{(1)}, j)^{(1)}$ serà $u_j^{(1)}$ i el seu cost $c_j^{(1)}$ (imposem que no es pugui transportar res de l'article 2 fent que la capacitat de les variables $(s^{(1)}, j)^{(2)}$ sigui 0). De forma anàloga, els arcs $(s^{(2)}, j)$ només poden transportar unitats de l'article 2. Per satisfer la demanda tenim els arcs $(j, s^{(i)})$, que també només poden subministrar unitats de l'article i , amb cost 0 i capacitat $d_j^{(i)}$. Finalment, els arcs $(j, j+1)$ sí que comparteixen els dos articles, i representen el total dels dos productes que s'emmagatzema d'un trimestre per a l'altre. El cost d'aquest arcs és $h_j^{(i)}$, i la seva capacitat és M . El problema consisteix en trobar la distribució òptima dels productes, sabent quan produir i quan emmagatzemar per tal de satisfer la demanda, tenint en compte els costos de producció i d'emmagatzemament. Aquest problema pot ser clarament ampliat, considerant un més gran nombre d'articles i intervals de temps, així com considerant d'altres costos, com per exemple, costos de transport entre els centres de producció i magatzem, i magatzem i centres de consum.

Capítol 3

Estat de l'art.

Des de l'aparició de l'algorisme del símplex, desenvolupat el 1947 per George Dantzig, aquest va ser objecte de diferents especialitzacions per resoldre problemes de fluxos en xarxes. La literatura sobre els mètodes de fluxos en xarxes és molt àmplia, i en aquest capítol només ens centrarem en aquelles tècniques específiques dels fluxos multiarticle. El lector interessat en tècniques de caràcter general pot consultar els textos [2,49], que representen una molt bona introducció als problemes de fluxos en xarxes.

Per tal de realitzar una exposició sobre els antecedents que hi ha dins d'aquest camp, es realitzarà una clara distinció entre els problemes de fluxos multiarticle lineals i els no lineals. Aquesta divisió va lligada al fet de que les tècniques emprades per resoldre un tipus i l'altre de problemes difereixen substancialment, i, fins i tot, en alguns casos només han estat aplicades amb èxit al cas lineal. De fet és en aquest tipus de problemes on l'activitat de recerca ha estat més intensa per part de la comunitat científica, i això queda reflectit en la quantitat de tècniques existents per a fluxos multiarticle lineals, mentre que per al cas no lineal escassament es troben intents d'abordar el problema. Aquest fet té una doble explicació. En primer lloc, el guany que s'obté en l'especialització per a fluxos multiarticle lineals és superior a l'obtingut per a problemes no lineals, per la pròpia naturalesa dels mètodes numèrics emprats. En segon lloc, el cost d'abordar problemes no lineals és superior al dels problemes lineals. Tots dos fets fan decantar la balança de forma favorable als fluxos lineals multiarticle, i això pot ser fàcilment comprovat a l'exposició sobre els antecedents en aquest camp que es farà més endavant.

Cal també destacar que no es coneix cap intent previ de combinar tècniques de fluxos multiarticle i tècniques per a fluxos amb constriccions a banda. En aquest capítol, per tant, solament es farà referència a les tècniques que s'han desenvolupat per tractar els dos problemes de forma separada, amb especial èmfasi per a les que tracten problemes multiarticle, i de forma breu per a les aplicades a constriccions a banda.

3.1 Fluxos lineals multiarticle.

Els primers intents de solucionar problemes de fluxos multiarticle (lineals) apareixen ja documentats el 1958 [27]. A partir d'aquest moment diverses tècniques i metodologies han estat proposades per solucionar aquest problema o variants del mateix. Aquestes poden ser agrupades en dos grans blocs: les que podrien ser considerades tècniques derivades de mètodes clàssics, i les derivades de mètodes de punt interior. Per tècniques clàssiques entendrem totes aquelles associades al mètode del símplex (amb les seves variants i especialitzacions per a programació lineal). Les tècniques de punt interior queda clar que són

aquelles directa o indirectament derivades de l'algorisme de Karmarkar [44] aparegut el 1984. Des d'aquest punt de vista, es podria dir que considerarem com tècniques clàssiques totes aquelles que no segueixen la metodologia dels mètodes de punt interior. Aquesta divisió pot considerar-se, certament, arbitrària. Tanmateix, ha estat considerada oportuna, donat que correspon als dos enfoc que han estat considerats en aquest treball. A banda d'aquestes dues clares divisions, s'afegeix un darrer grup de metodologies, de caràcter més recent, de les quals es tenen alguns resultats de caràcter teòric, tot i que encara està per veure la seva eficiència en implementacions de caràcter pràctic.

3.1.1 Mètodes clàssics.

Tres són els mètodes desenvolupats per solucionar problemes de fluxos multiarticle que podrien qualificar-se com clàssics: descomposició dictada per preus, descomposició dictada per recursos i particionament primal.

3.1.1.1 Descomposició dictada per preus.

El primer dels tres mètodes consisteix en l'aplicació de l'anomenada descomposició de Dantzig-Wolfe [21,53] al problema de fluxos lineals multiarticle. En aquest mètode el problema original es divideix en un problema mestre i k problemes (un per article) de cost mínim en xarxes uniarticle. Definint els conjunts convexos

$$\Lambda^{(i)} = \{x^{(i)} : Ax^{(i)} = b^{(i)}, \underline{0} \leq x^{(i)} \leq \bar{x}^{(i)}\} \quad i = 1, \dots, k \quad (3.1)$$

qualsevol flux $x^{(i)}$ solució de (2.2) i (2.3) pot ser escrit com

$$x^{(i)} = \sum_j x_{(j)}^{(i)} \lambda_{(j)}^{(i)} \quad (3.2)$$

on $x_{(j)}^{(i)}$ representen els punts extrems de $\Lambda^{(i)}$, i $\lambda_{(j)}^{(i)}$ els coeficients convexos associats (on $\lambda_{(j)}^{(i)} \geq 0$ i $\sum_j \lambda_{(j)}^{(i)} = 1$). D'aquesta forma el problema (FML) formulat a (2.1–2.4) pot ser escrit en funció de les variables $\lambda_{(j)}^{(i)}$ de forma:

$$\begin{aligned} \min_{\lambda_{(j)}^{(1)}, \dots, \lambda_{(j)}^{(k)}} \quad & \sum_{i,j} c^{(i)} x_{(j)}^{(i)} \lambda_{(j)}^{(i)} \\ \text{subj.a} \quad & \sum_{i,j} x_{(j)}^{(i)} \lambda_{(j)}^{(i)} \leq b_{em} \\ & \left. \begin{aligned} \sum_j \lambda_{(j)}^{(i)} &= 1 \\ \sum_j \lambda_{(j)}^{(i)} &\geq 0 \end{aligned} \right\} i = 1, \dots, k \end{aligned} \quad (3.3)$$

El problema mestre correspon a (3.3), i els costos reduïts d'aquest problema són usats per definir la funció objectiu dels k problemes de cost mínim en xarxes uniarticle definits per

(3.1). Per a una descripcions més acurada sobre aquesta metodologia [2,47,49,61] poden ser consultats.

Ford i Fulkerson [27] van ser els primers que van descriure i aplicar la metodologia anterior als problemes de fluxos multiarticle. En aquest article presenten una tècnica de generació de columnes a una formulació arc-camí del problema del flux màxim multiarticle. Aquest treball va ser, de fet, l'inspirador del posterior mètode de Dantzig-Wolfe [21]. Tomlin [75] va estendre la idea de Ford i Fulkerson per a problemes multiarticle de cost mínim, i la va aplicar a una formulació arc-node, mostrant que aquesta era equivalent a la formulació arc-camí. El mateix Tomlin va realitzar la primera implementació que usava aquesta tècnica, però relaxava els k subproblemes eliminant els límits de capacitat dels arcs. Després d'aquesta van aparèixer d'altres implementacions. Una de les darreres va ser desenvolupada per Ali, Kennington i Helgason, i la seva eficiència és estudiada a [3].

3.1.1.2 Descomposició dictada per recursos.

Al mètode de la descomposició dictada per preus es descomposava el problema original en k subproblemes atenent als preus (o costos) de les constriccions de capacitat mútua del problema original. A la descomposició dictada per recursos, com indica el seu nom, també s'aconsegueix descomposar el problema original, però ara distribuint la capacitat mútua de cada arc pels k articles individuals, de forma que solucionant els k subproblemes s'obté un flux solució pel problema acoblat. En aquest cas el problema consisteix en trobar la distribució òptima de la capacitat mútua (és a dir, la distribució òptima de recursos) de forma que la solució dels k subproblemes ens proporcioni l'òptim del problema multiarticle. Aquesta aproximació va ser suggerida inicialment per Robacker [70].

La formulació original del problema (**FML**) pot ser llavors reescrita com:

$$\begin{aligned}
 & \min_{\substack{x^{(1)}, \dots, x^{(k)} \\ r^{(1)}, \dots, r^{(k)}}} \sum_{i=1}^k c^{(i)} x^{(i)} \\
 & \text{subj. a } \left. \begin{aligned} Ax^{(i)} &= b^{(i)} \\ \underline{0} &\leq x^{(i)} \leq r^{(i)} \end{aligned} \right\} i = 1, \dots, k \\
 & \sum_{i=1}^k r^{(i)} \leq b_{cm}
 \end{aligned} \tag{3.4}$$

essent $r^{(i)} \in \mathbb{R}^n$ el vector de recursos permesos als fluxos dels arcs de l'article i . El problema (3.4) és equivalent al problema (**FML**), però el conjunt de variables ha estat ampliat i ara consta de (x, r) , on x denota els fluxos i r els recursos. La solució del problema s'obté tractant de forma seqüencial els dos grups de variables, en comptes de solucionar-lo conjuntament per a x i r . Primer fixem els recursos r , i després trobem els fluxos òptims

associats (solucionant k subproblemes de cost mínim en xarxes uniarticle amb capacitat). Això equival a solucionar:

$$\begin{aligned} \min_{r^{(1)}, \dots, r^{(k)}} \quad & \sum_{i=1}^k z_i(r^{(i)}) = \min_{x^{(1)}, \dots, x^{(k)}} \sum_{i=1}^k c^{(i)} x^{(i)} \\ \text{subj. a} \quad & \left. \begin{aligned} Ax^{(i)} &= b^{(i)} \\ \underline{0} \leq x^{(i)} &\leq r^{(i)} \end{aligned} \right\} i = 1, \dots, k \end{aligned} \quad (3.5)$$

$$\begin{aligned} \text{subj. a} \quad & \sum_{i=1}^k r^{(i)} \leq b_{cm} \\ & \underline{0} \leq r^{(i)} \leq \bar{x}^{(i)} \quad i = 1, \dots, k \end{aligned}$$

La funció objectiu de (3.5) és clarament no diferenciable, i per tal de solucionar el problema s'han proposat diverses tècniques. Entre aquestes cal destacar el mètode de l'aproximació tangencial (implementat per Swoveland a [73]) i l'aplicació de tècniques d'optimització subgradient (proposades per Held, Wolfe i Crowder [40]). El mètode de la descomposició dictada per recursos no s'ha mostrat com una alternativa molt eficient per a la resolució de problemes multiarticle, i això ha estat confirmat a [3].

3.1.1.3 Particionament primal.

El mètode del particionament primal consisteix en una especialització del mètode del símplex on la matriu bàsica és particionada per tal d'explotar l'estructura del problema multiarticle. La motivació és, doncs, retenir el màxim possible l'estructura de xarxa subjacent als problemes multiarticle. Aquesta tècnica va ser emprada ja el 1972 per Grigoriadis i White [35] per solucionar el problema dual de **(FML)**, i el 1975 per Kennington [46] per solucionar el problema primal. El mateix Kennington va fer posteriorment una implementació eficient d'aquesta tècnica [48]. Descripcions detallades sobre aquest mètode poden ser trobades a [2,47,49].

El mètode del particionament primal ha estat escollit en aquest treball per ser ampliat amb constriccions a banda i funcions no lineals, donat que és un dels que ha mostrat un millor comportament en la solució de problemes multiarticle. Això ja va ser observat per Ali, Helgason i Kennington a [3], on es van comparar tres implementacions basades en la descomposició dictada per preus, la descomposició dictada per recursos i un particionament primal. Es va poder observar que els mètodes de la descomposició dictada per preus i el particionament primal van tenir un millor comportament.

3.1.2 Mètodes de punt interior.

Des de l'aparició el 1984 d'un algorisme basat en transformacions projectives a [44] per resoldre problemes de programació lineal (anomenat també algorisme de Karmarkar), es va iniciar una forta activitat de recerca dins del camp dels anomenats mètodes de

punt interior. Aquests, a diferència del fins llavors majoritàriament emprat algorisme del símplex, arriben al punt òptim a través de l'interior de la zona factible delimitada pel conjunt de constriccions, en comptes de bellugar-se per la frontera com ho feien els iterats obtinguts amb el símplex. L'algorisme de les transformacions projectives, però, va ser precedit per l'algorisme de l'elipsoid [51], també de complexitat polinòmica, encara que mai va ser un competidor del símplex des del punt de vista pràctic. Des d'aquell article inicial d'en Karmarkar nous mètodes van aparèixer, entre ells els anomenats d'escalat afí, on la transformació projectiva de l'algorisme de Karmarkar era substituïda per una transformació afí [6,55,76,81]. De fet aquests mètodes ja daten del 1967, any en que Dikin va presentar un mètode de punt interior per solucionar problemes lineals i quadràtics [24]. Posteriorment van aparèixer els anomenats mètodes primal-dual [56,57], basats en la substitució de les constriccions de no-negativitat de les variables per una barrera logarísmica [83]. Els mètodes de punt interior han guanyat un ampli reconeixement com a procediment d'optimització per a problemes lineals en els darrers anys, i s'han mostrat força més eficients que l'algorisme del símplex [1,45,79]. Per tenir una visió general sobre els diversos mètodes de punt interior existents poden consultar-se [5], com a text introductor, i [80], on es fa una comparació del rendiment de diversos algorismes.

En vista dels resultats exitosos obtinguts amb els mètodes de punt interior per a problemes de programació lineal general, aquests no van trigar en ser adaptats per solucionar problemes de fluxos en xarxes de grans dimensions. Els primers treballs daten del 1993 i van ser desenvolupats per Resende i Veiga [68,69]. En aquests s'usava l'anomenat mètode de l'escalat afí dual. Per tal d'agilitzar la convergència de l'algorisme s'usava un criteri d'acabament per detectar el punt òptim un cop ens trobàvem en un entorn d'ell [67]. Amb posterioritat es va assajar amb èxit l'aplicació d'un mètode primal-dual a problemes de fluxos [65]. Una descripció detallada sobre aquests mètodes pot ser trobada a [66].

Pel que fa als fluxos multiarticle, diversos autors han presentat mètodes teòricament eficients, com ara Kapoor i Vaidya [43], que presenten resultats sobre la complexitat d'un algorisme per solucionar problemes multiarticle. Amb posterioritat Choi i Goldfarb adapten un mètode primal-dual per tal d'aprofitar l'estructura de certs tipus de problemes [19], i, en particular, l'apliquen a la solució de problemes multiarticle [18]. En aquests dos casos, però, no es presenta cap tipus de resultat computacional. Aquests sí que són, però, presentats a [42], on s'especialitza un mètode d'escalat afí dual a la solució de problemes multiarticle. Més recentment a [17] s'aplica amb èxit un mètode primal-dual a la solució de problemes multiarticle.

3.1.3 Altres mètodes.

Tal i com ja es va dir anteriorment, a part dels mètodes clàssics i els de punt interior, hi ha un seguit de mètodes addicionals que han estat aplicats a la resolució de problemes multiarticle. Entre aquests destacarem únicament dos. El primer consisteix en l'aplicació de tècniques de relaxació lagrangiana a problemes de fluxos multiarticle [2]. En aquest cas, les constriccions de capacitat mútua són passades a la funció objectiu afegint els

corresponents multiplicadors. D'aquesta forma s'aconsegueix desacoblar el problema pel que fa a les constriccions. El segon mètode és presentat a [52], on es soluciona de forma aproximada el problema del flux concurrent, la solució del qual proporciona punts factibles del problema (**FML**). Cal indicar que tant en aquest mètode com en el de la relaxació lagrangiana no s'obtenen solucions exactes, sinó aproximades.

3.2 Fluxos no lineals multiarticle.

A diferència del cas lineal, on han estat suggerides diverses tècniques i algorismes, per solucionar problemes multiarticle no lineals hi ha una manca prou palpable tant de mètodes com d'implementacions. Aquest fet, però, no és exclusiu dels fluxos multiarticle, i es pot fer extensiu als problemes de fluxos generals. Entre els articles que tracten problemes de fluxos no lineals es pot destacar el treball de Toint i Tuyttens [74], on es combinen tècniques de fluxos amb el mètode de Murtagh i Saunders [58] per a problemes de gran escala.

Pel que fa als fluxos multiarticle no lineals, es va intentar aplicar amb poc èxit una especialització no lineal del mètode de la descomposició dictada per preus [61,63]. En aquest cas, de nou, s'usà el mètode de Murtagh i Saunders. Amb posterioritat es va desenvolupar una versió no lineal del mètode del particionament primal [13,15,16]. En aquest cas els resultats obtinguts van ser satisfactoris. Val a dir que, tot i que en aquests treballs es va emprar el mètode de Murtagh i Saunders, es podria haver emprat qualsevol altre mètode per a programació no lineal [30].

3.3 Fluxos amb constriccions a banda.

En aquest treball, com ja s'ha esmentat, s'ha aplicat la tècnica del particionament primal a la resolució de problemes multiarticle amb constriccions a banda. La tècnica del particionament primal, però, ja ha estat utilitzada per separat al cas multiarticle i al de constriccions a banda, tal i com es presenta a [49]. Aplicant aquesta tècnica a la solució de problemes amb constriccions a banda s'han obtingut diverses implementacions, com les presentades a [37,50]. Tot això pel que fa al cas lineal. Pel cas no lineal, a [38] es presenta un algorisme on s'aplica el mètode de Murtagh i Saunders a la solució de problemes de fluxos amb constriccions a banda.

Capítol 4

Fluxos lineals multiarticle per particionament primal.

Al capítol 2 es va realitzar la formulació dels problemes **(FMLCB)** i **(FMNCB)**. En aquest es presentarà la metodologia a seguir per solucionar el problema amb la funció objectiu lineal. El tractament del cas no lineal serà l'objecte d'estudi del següent capítol. De fet, tots dos capítols podrien haver estat agrupats en un de sol on es fes la descripció global del mètode per al cas lineal i no lineal. Tanmateix, s'ha preferit fer aquesta separació per clarificar l'explicació. El material presentat en ambdós capítols ha estat descrit parcialment en altres treballs de l'autor [10,13,14,15].

De totes les aproximacions presentades al capítol 3 per solucionar el problema **(FML)**, la tècnica del particionament primal es va destacar com una de les més eficients. Aquest fet, junt amb el de permetre un aprofitament del mateix particionament en el cas de disposar de constriccions addicionals del tipus (2.7), fa que aquesta metodologia hagi estat escollida per desenvolupar un algorisme per solucionar problemes multiarticle amb constriccions a banda. Passem tot seguit a detallar-lo.

4.1 Mètode del particionament primal.

El mètode del particionament primal va ser originalment formulat per tractar problemes sense les constriccions a banda (2.7). En aquest apartat es detallarà l'adaptació del mètode per tal de solucionar el problema ampliat **(FMLCB)**, el qual representa una extensió del mètode original desenvolupat a [49]. En primer lloc es mostrarà l'estructura del problema, tot detallant el particionament de la matriu bàsica, per passar a continuació a justificar l'ús de l'anomenada matriu de treball.

4.1.1 Estructura del problema.

Tenint en compte l'estructura matricial de les constriccions del problema **(FMLCB)**, presentada a (2.12), pot observar-se que cada flux $j^{(i)}$ (és a dir, el flux de l'arc j de l'article i , exceptuant l'arc arrel d'aquesta consideració) té una columna associada $\tilde{A}^{j^{(i)}}$ dins la matriu \tilde{A} definida a (2.11) amb les següents components no nul·les:

$$(\tilde{A}^{j^{(i)}})^t = \left(\underbrace{\dots 1 \dots -1 \dots}_{\text{Xarxa}} \mid \underbrace{\dots 1 \dots}_{\text{Capacitat Mútua}} \mid \underbrace{a_{j_1} \dots a_{j_t}}_{\text{Constriccions a Banda}} \right)^t \quad (4.1)$$

on v i w identifiquen el node origen i node destí de l'arc j . Pot observar-se com cada arc intervé en tres tipus de constriccions clarament diferenciades: constriccions de xarxa, constriccions de capacitat mútua, i constriccions a banda. Les constriccions de xarxa són d'igualtat, i per tant sempre estaran actives. Per la seva, les constriccions de capacitat mútua (n'hi ha una per cada arc de la xarxa) són de desigualtat, i per a alguns arcs estaran actives (esdevenint aleshores arcs saturats) i per a altres no. Les constriccions a banda poden ser d'igualtat i de desigualtat, i de nou poden estar actives o no. Cal esmentar que els dos primers tipus de constriccions (de xarxa i capacitat mútua) són independents respecte l'article i (només sabent que es tracta de l'arc j podem determinar la seva estructura), mentre que les constriccions a banda sí depenen de l'arc i de l'article en qüestió. Per tal d'identificar els conjunts de constriccions actives, introduïrem els següents conceptes:

- $|\mathcal{C}|$: cardinalitat del conjunt qualsevol \mathcal{C} .
- \mathcal{A}_{cm} : conjunt de constriccions de capacitat mútua actives al punt actual. Es verifica sempre que $0 \leq |\mathcal{A}_{cm}| \leq n$.
- \mathcal{A}_{cb} : conjunt de constriccions a banda actives al punt actual. Es verifica sempre que $0 \leq |\mathcal{A}_{cb}| \leq t$.
- \mathcal{A} : conjunt de constriccions de capacitat mútua i a banda actives ($\mathcal{A} = \mathcal{A}_{cm} \cup \mathcal{A}_{cb}$), on $0 \leq |\mathcal{A}| \leq n + t$.

Si usem el mètode del símplex com a procediment de resolució del nostre problema d'optimització, haurem de descomposar la nostra matriu de constriccions \tilde{A} de forma $\tilde{A} = [B|N]$, essent $B \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$ la matriu amb les variables bàsiques, i $N \in \mathbb{R}^{\tilde{m} \times (\tilde{n} - \tilde{m})}$ la matriu de les variables no bàsiques. La següent proposició ens mostra la forma en que pot ser escrita tota matriu bàsica B al mètode del particionament primal.

Proposició 4.1.

Tota base B de \tilde{A} pot ser escrita de la forma següent

$$B = \begin{array}{ccccccc}
 \boxed{B^{(1)}} & & & \boxed{R^{(1)}} & & & \\
 & \ddots & & & \ddots & & \\
 & & \boxed{B^{(k)}} & & & \boxed{R^{(k)}} & \\
 \boxed{P_B^{(1)}} & \dots & \boxed{P_B^{(k)}} & \boxed{P_R^{(1)}} & \dots & \boxed{P_R^{(k)}} & \\
 \boxed{S_B^{(1)}} & \dots & \boxed{S_B^{(k)}} & \boxed{S_R^{(1)}} & \dots & \boxed{S_R^{(k)}} & \mathbb{1}
 \end{array} \begin{array}{l}
 \} m \\
 \vdots \\
 \} m \\
 \} |\mathcal{A}| \\
 \} n+t-|\mathcal{A}|
 \end{array} \tag{4.2}$$

$\underbrace{\hspace{1.5cm}}_m \quad \dots \quad \underbrace{\hspace{1.5cm}}_m \quad \underbrace{\hspace{3.5cm}}_{n+t}$

on $B^{(i)} \in \mathbb{R}^{m \times m}$ són bases de la matriu de xarxa A (és a dir, són arbres d'expansió mínima), i les matrius $P_B^{(i)}$, $P_R^{(i)}$, $S_B^{(i)}$ i $S_R^{(i)}$ són una partició de les constriccions de capacitat mútua i a banda (les matriu P estan associades a constriccions actives i les S a constriccions inactives).

Demostració.

Si B és una base per a \tilde{A} aleshores serà una matriu quadrada de dimensió $km+n+t$, i tindrà una estructura com ara:

$$B = \begin{array}{cccc} V^{(1)} & & & \\ & \ddots & & \\ & & V^{(k)} & \\ X^{(1)} & \dots & X^{(k)} & W \end{array} \begin{array}{l} \}m \\ \vdots \\ \}m \\ \}n+t \end{array}$$

El rang de cada $V^{(i)}$ no pot ser menor que m , ja que si ho fos hi hauria files de $V^{(i)}$ que serien combinació lineal unes d'altres, amb el qual B no podria ser base. Per tant tenim que podem escriure cada $V^{(i)}$ com $V^{(i)} = [B^{(i)}|R^{(i)}]$, essent $B^{(i)}$ quadrada i de rang m (per tant, una base per a la matriu de xarxa A), i on la part $R^{(i)}$ pot ser buida. Aplicant aquesta reordenació de columnes a les matrius $X^{(i)}$, i considerant una altra reordenació de files en funció de tenir constriccions actives o no, directament obtenim que B pot ser escrita tal i com es mostra a (4.2). ■

Els arcs que apareixen dins les matrius $B^{(i)}$ els anomenarem arcs bàsics, mentre que els que formen part de les matrius $R^{(i)}$ seran referits com a arcs complementaris. Si ara denotem les submatrius de (4.2) de la forma següent:

$$\begin{array}{l} L_1 = \begin{array}{|c|c|c|} \hline B^{(1)} & & \\ \hline & \ddots & \\ \hline & & B^{(k)} \\ \hline \end{array} \\ L_2 = \begin{array}{|c|c|c|} \hline P_B^{(1)} & \dots & P_B^{(k)} \\ \hline \end{array} \\ L_3 = \begin{array}{|c|c|c|} \hline S_B^{(1)} & \dots & S_B^{(k)} \\ \hline \end{array} \end{array} \quad \begin{array}{l} R_1 = \begin{array}{|c|c|c|} \hline R^{(1)} & & \\ \hline & \ddots & \\ \hline & & R^{(k)} \\ \hline \end{array} \\ R_2 = \begin{array}{|c|c|c|} \hline P_R^{(1)} & \dots & P_R^{(k)} \\ \hline \end{array} \\ R_3 = \begin{array}{|c|c|c|} \hline S_R^{(1)} & \dots & S_R^{(k)} \\ \hline \end{array} \end{array} \quad (4.3)$$

podem escriure B com segueix:

$$B = \begin{array}{|c|c|c|} \hline L_1 & R_1 & \mathbf{0} \\ \hline L_2 & R_2 & \mathbf{0} \\ \hline L_3 & R_3 & \mathbf{1} \\ \hline \end{array} \begin{array}{l} \}n_1 \\ \}n_2 \\ \}n_3 \end{array} \quad (4.4)$$

$\underbrace{\hspace{1.5cm}}_{n_1} \quad \underbrace{\hspace{1.5cm}}_{n_2} \quad \underbrace{\hspace{1.5cm}}_{n_3}$

essent $n_1 = km$, $n_2 = |\mathcal{A}|$, i $n_3 = n + t - |\mathcal{A}|$. De forma breu podem descriure cada submatriu tal com segueix:

- L_1 està associada a les constriccions de xarxa i arcs dels k arbres d'expansió mínima. Cal fer notar que L_1 és una matriu quadrada ($L_1 \in \mathbb{R}^{km \times km}$).
- R_1 està associada a les constriccions de xarxa i arcs complementaris dels k articles.

- L_2 està associada a les constriccions de capacitat mútua i a banda actives, per als arcs dels arbres d'expansió mínima.
- R_2 està associada a les constriccions de capacitat mútua i a banda actives, per als arcs complementaris. Cal fer notar que R_2 és una matriu quadrada ($R_2 \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$).
- L_3 està associada a les constriccions de capacitat mútua i a banda inactives, per als arcs dels arbres d'expansió mínima.
- R_3 està associada a les constriccions de capacitat mútua i a banda inactives, per als arcs complementaris.
- $\mathbb{1}$ és la matriu identitat de dimensió $n + t - |\mathcal{A}|$, i està associada a les folgues de les constriccions de capacitat mútua i a banda inactives (cal indicar que les constriccions quines folgues apareixen a la matriu $\mathbb{1}$ són tractades com inactives, encara que els valors de les folgues siguin zero).

Cal observar que això és una extensió al cas de presència de constriccions a banda de la formulació del particionament primal de Kennington i Helgason [49].

4.1.2 Motivació de la matriu de treball.

Dins del procés d'optimització del símplex a cada iteració cal resoldre sistemes del tipus $Bx=b$ i $u^t B=c^t$. Tenint en compte la partició de la base presentada a (4.4) i considerant una partició similar per als vector x , b , u i c , observem, tal com proposa [49], quines operacions són necessàries per obtenir la solució dels dos sistemes:

4.1.2.1 Càlcul de $Bx = b$.

Aquest sistema pot ser escrit com:

$$\begin{array}{|c|c|c|} \hline L_1 & R_1 & 0 \\ \hline L_2 & R_2 & 0 \\ \hline L_3 & R_3 & \mathbb{1} \\ \hline \end{array} \begin{array}{|c|} \hline x_1 \\ \hline x_2 \\ \hline x_3 \\ \hline \end{array} = \begin{array}{|c|} \hline b_1 \\ \hline b_2 \\ \hline b_3 \\ \hline \end{array}$$

Multiplicant per blocs obtenim:

$$L_1 x_1 + R_1 x_2 = b_1 \tag{4.5}$$

$$L_2 x_1 + R_2 x_2 = b_2 \tag{4.6}$$

$$L_3 x_1 + R_3 x_2 + x_3 = b_3 \tag{4.7}$$

Aïllant x_1 de (4.5) obtenim $x_1 = L_1^{-1} b_1 - L_1^{-1} R_1 x_2$. Substituint aquest valor a (4.6) obtenim $L_2 L_1^{-1} b_1 - L_2 L_1^{-1} R_1 x_2 + R_2 x_2 = b_2$. Aïllant el valor de x_2 trobem la seva expressió:

$$x_2 = (R_2 - L_2 L_1^{-1} R_1)^{-1} (b_2 - L_2 L_1^{-1} b_1) \tag{4.8}$$

Coneixent x_2 el càlcul de x_1 és directament:

$$x_1 = L_1^{-1} b_1 - L_1^{-1} R_1 x_2 \tag{4.9}$$

Pel que fa a x_3 , un cop coneguts x_1 i x_2 , directament fem:

$$x_3 = b_3 - L_3x_1 - R_3x_2 \quad (4.10)$$

Calculant consecutivament (4.8), (4.9) i (4.10)) obtenim la solució del sistema.

4.1.2.2 Càlcul de $u^t B = c^t$.

El sistema pot ser escrit com:

$$\begin{array}{|c|c|c|} \hline u_1 & u_2 & u_3 \\ \hline \end{array} \begin{array}{|c|c|c|} \hline L_1 & R_1 & 0 \\ \hline L_2 & R_2 & 0 \\ \hline L_3 & R_3 & \mathbb{1} \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline c_1 & c_2 & c_3 \\ \hline \end{array}$$

Multiplicant per blocs obtenim:

$$u_1^t L_1 + u_2^t L_2 + u_3^t L_3 = c_1^t \quad (4.11)$$

$$u_1^t R_1 + u_2^t R_2 + u_3^t R_3 = c_2^t \quad (4.12)$$

$$u_3^t = c_3^t \quad (4.13)$$

El valor u_3 es troba directament a partir de (4.10). Aïllant u_1 de (4.11) obtenim

$$u_1^t = (c_1^t - u_3^t L_3 - u_2^t L_2) L_1^{-1} \quad (4.14)$$

Usant (4.14) i (4.13) a (4.12) obtenim:

$$\begin{aligned} (c_1^t - c_3^t L_3 - u_2^t L_2) L_1^{-1} R_1 + u_2^t R_2 + c_3^t R_3 &= c_2^t \\ (c_1^t - c_3^t L_3) L_1^{-1} R_1 + u_2^t (R_2 - L_2 L_1^{-1} R_1) &= c_2^t - c_3^t R_3 \\ ((c_2^t - c_3^t R_3) - (c_1^t - c_3^t L_3) L_1^{-1} R_1) (R_2 - L_2 L_1^{-1} R_1)^{-1} &= u_2^t \end{aligned} \quad (4.15)$$

Resolent consecutivament (4.13), (4.15) i (4.14) obtenim la solució del sistema.

4.1.2.3 Invertibilitat de Q .

Com es pot observar als dos subapartats anteriors, per tal de trobar la solució als sistemes plantejats només cal invertir la submatriu L_1 i una matriu l'expressió de la qual és $R_2 - L_2 L_1^{-1} R_1$. Aquesta matriu serà anomenada *la matriu de treball* i la denotarem per Q :

$$Q = R_2 - L_2 L_1^{-1} R_1 \quad (4.16)$$

Cal indicar que el fet d'haver d'invertir L_1 no suposa cap problema donat que, tal i com va ser definida a (4.3), aquesta és una matriu diagonal per blocs, on cada bloc representa un arbre d'expansió mínima. Com és ben sabut les solucions a sistemes d'aquest tipus poden ser trobades de forma immediata tenint en compte només l'estructura de l'arbre [9,49]. Pel que fa a Q , però, cal poder garantir la seva invertibilitat. Aquest fet ens ve assegurat per la següent proposició.

Proposició 4.2.

Considerem una matriu B tal i com es defineix a (4.4). Si $\det(L_1) \neq 0$ i $\det(B) \neq 0$ aleshores $\det(R_2 - L_2L_1^{-1}R_1) \neq 0$. Observem que les dues hipòtesis es verifiquen al nostre cas: $\det(B) \neq 0$ donat que és una base del nostre problema, i $\det(L_1) \neq 0$ donat que té k bases (situades en blocs diagonals) per a la matriu A de xarxa.

Demostració.

Considerem les matrius T_1 i T_2 definides de forma:

$$T_1 = \begin{array}{|c|c|c|} \hline \mathbb{1} & -L_1^{-1}R_1 & \\ \hline & \mathbb{1} & \\ \hline & & \mathbb{1} \\ \hline \end{array} \quad \text{i} \quad T_2 = \begin{array}{|c|c|c|} \hline \mathbb{1} & & \\ \hline & \mathbb{1} & \\ \hline & L_3L_1^{-1}R_1 - R_3 & \mathbb{1} \\ \hline \end{array}$$

Clarament $\det(T_1) = \det(T_2) = 1$. Si ara postmultipliquem B per T_1 i T_2 tenim que:

$$BT_1T_2 = \begin{array}{|c|c|c|} \hline L_1 & & \\ \hline L_2 & R_2 - L_2L_1^{-1}R_1 & \\ \hline L_3 & & \mathbb{1} \\ \hline \end{array}$$

Llavors $\det(BT_1T_2) = \det(B) \det(T_1) \det(T_2) = \det(B) \neq 0$. Però $\det(BT_1T_2) \neq 0$ implica que B té rang complet, el qual a la seva vegada implica que $R_2 - L_2L_1^{-1}R_1$ ha de tenir rang complet i per tant $\det(R_2 - L_2L_1^{-1}R_1) \neq 0$. ■

Per tant, un cop hem vist que Q és invertible, només resta tenir un tractament eficient d'aquesta matriu per tal de tenir solventat el càlcul dels dos sistemes anteriors, que és la part més costosa de tot el procés d'optimització al cas lineal, i força rellevant a la no lineal. En aquest sentit dos són els punts bàsics necessaris: tenir un procediment eficient de calcular la matriu Q i tenir una forma —igualment eficient— d'actualitzar-la per no haver d'invertir-la a cada iteració. Donat que la dimensió de la matriu Q és, en general, força menor que la de tota la base B , el fet de treballar amb la matriu de treball ens millora considerablement l'eficiència del mètode.

Les seccions següents tractaran els dos punts clau en treballar amb la matriu Q : com calcular-la (de forma eficient) i com actualitzar-la (també de forma eficient).

4.2 Càlcul eficient de la matriu de treball.

Tenint en compte l'expressió de Q presentada a (4.16), i les dimensions de les submatrius que componen B a (4.4), si denotem per \dim la dimensió d'una matriu, directament s'obté:

$$\begin{aligned} \dim(Q) &= \dim(R_2 - L_2L_1^{-1}R_1) \\ \dim(Q) &= (|\mathcal{A}| \times |\mathcal{A}|) - (|\mathcal{A}| \times Km)(Km \times Km)(Km \times |\mathcal{A}|) \end{aligned}$$

$$\begin{aligned}\dim(Q) &= (|\mathcal{A}| \times |\mathcal{A}|) - (|\mathcal{A}| \times |\mathcal{A}|) \\ \dim(Q) &= (|\mathcal{A}| \times |\mathcal{A}|)\end{aligned}$$

Anàlogament a com ocorre amb R_2 , les files de Q es troben associades a les constriccions (a banda i capacitat mútua) actives (\mathcal{A}), mentre que les seves columnes es troben associades amb els arcs complementaris. Agrupant les files associades amb constriccions a banda actives per una banda i les files associades amb constriccions de capacitat mútua actives per una altra, podem considerar la matriu Q dividida en dos submatrius, tal com ara $Q = \begin{bmatrix} Q_{cm} \\ Q_{cb} \end{bmatrix}$, essent Q_{cm} la submatriu les files de la qual estan associades a constriccions $\in \mathcal{A}_{cm}$ i Q_{cb} la submatriu les files de la qual estan associades a constriccions $\in \mathcal{A}_{cb}$ (on $\dim(Q_{cm}) = |\mathcal{A}_{cm}| \times |A|$ i $\dim(Q_{cb}) = |\mathcal{A}_{cb}| \times |A|$). Cal tenir en compte que aquesta divisió que s'ha fet a la matriu Q pot ser aplicada a la matriu R_2 i la matriu L_2 (ja que les seves files estan associades amb constriccions actives). Per tant, bé podríem escriure el càlcul de Q com

$$Q = \begin{bmatrix} Q_{cm} \\ Q_{cb} \end{bmatrix} = \begin{bmatrix} R_{2_{cm}} \\ R_{2_{cb}} \end{bmatrix} - \begin{bmatrix} L_{2_{cm}} \\ L_{2_{cb}} \end{bmatrix} L_1^{-1} R_1 \quad (4.17)$$

Cal observar que el terme $L_1^{-1} R_1$ no depèn per a res del tipus de constricció activa associat amb les files de Q . Cada columna R_{1_j} , $j = 1, \dots, |\mathcal{A}|$ ens indica la connexió de l'arc complementari j dins de l'arbre d'expansió mínima que li pertoca (en funció de l'article i associat a l'arc complementari j); és a dir la columna R_{1_j} només té un $+1$ i un -1 a les posicions v i w , essent v el node origen de l'arc j i w el node destí. Donada l'estructura particular de R_1 i L_1 , solucionar $L_1^{-1} R_{1_j}$ per a l'arc j concret que pertany a l'article i és equivalent a solucionar $B^{(i)-1} R_{1_j}$ essent $B^{(i)}$ l'arbre d'expansió mínima de l'article i . La solució és $B^{(i)-1} R_{1_j} = V_j$, on V_{j_l} , $l = 1, \dots, m$ (essent m el nombre de nodes) s'obté directament com:

$$V_{j_l} = \begin{cases} +1 : & \text{si l'arc de l'arbre que hi ha per sobre del node } l \\ & \text{apareix en el cicle de l'arc } j \text{ dins l'arbre, i apunta} \\ & \text{cap al node destí de l'arc } j. \\ -1 : & \text{si l'arc de l'arbre que hi ha per sobre del node } l \\ & \text{apareix en el cicle de l'arc } j \text{ dins l'arbre, i apunta} \\ & \text{cap al node origen de l'arc } j. \\ 0 : & \text{en cas contrari.} \end{cases} \quad (4.18)$$

Per tant, solucionar $L_1^{-1} R_1$ és equivalent a tenir els arcs que formen els cicles (que denotarem per \mathcal{P}_j , $j = 1, \dots, |\mathcal{A}|$) de cada arc complementari j dins l'arbre d'expansió mínima que li pertoca. Donat un arc $a \in \mathcal{P}_j$ direm que a té *orientació normal* si apunta al node origen de l'arc complementari j ; en cas contrari, direm que a té *orientació inversa* (dit d'una altra forma, l'arc $a \in \mathcal{P}_j$ té orientació normal si $V_{j_l} = -1$, essent l el node de la xarxa que té per sobre seu l'arc a de l'arbre; si $V_{j_l} = +1$ l'arc a tindrà orientació inversa).

Un cop ja tenim calculat el terme $L_1^{-1} R_1$ cal veure com calcular les submatrius Q_{cm} i Q_{cb} . Això es veurà als dos punts que segueixen.

4.2.1 Càlcul de Q_{cm} .

La següent proposició presenta un procediment de càlcul de la submatriu Q_{cm} .

Proposició 4.3.

Si denotem per:

- a_j l'arc associat amb la j -èssima columna de Q , $j = 1, \dots, |\mathcal{A}|$.
- cm_i la constricció de capacitat mútua associada amb la i -èssima fila de Q , $i = 1, \dots, |\mathcal{A}_{cm}|$ (donat que hi ha una constricció de capacitat mútua per cada arc de la xarxa podem tractar indistintament el valor cm_i com referent a un arc o com referent a la constricció de capacitat mútua associada a aquest arc).

llavors podem calcular la matriu Q_{cm} directament fent:

$$Q_{ij} \begin{matrix} i=1, \dots, |\mathcal{A}_{cm}| \\ j=1, \dots, |\mathcal{A}| \end{matrix} := \begin{cases} +1, & \text{si } a_j = cm_i \\ +1, & \text{si } cm_i \in \mathcal{P}_j \text{ amb orientació normal} \\ -1, & \text{si } cm_i \in \mathcal{P}_j \text{ amb orientació inversa} \\ 0, & \text{altrament} \end{cases}$$

Demostració.

Recordem que l'expressió de la matriu Q_{cm} és

$$Q_{cm} = R_{2_{cm}} - L_{2_{cm}} L_1^{-1} R_1$$

Així mateix cal tenir present l'aspecte de la fila i -èssima de $L_{2_{cm}}$ (que anomenarem $L_{2_{cm}}^i$): és un vector que només té k 1's a les columnes associades amb els arcs cm_i de cada article del conjunt \mathcal{K} d'articles, i a la resta 0's. Considerem tots els casos possibles (obviem afegir el subíndex cm a les matrius per simplificar la notació):

- $a_j = cm_i$. En aquest cas l'element $R_{2_{ij}} = +1$. Per altra banda és clar que $a_j \notin \mathcal{P}_j \Rightarrow cm_i \notin \mathcal{P}_j$, i llavors $(L_2 L_1^{-1} R_1)_{ij} = L_2^i V_j = 0$. Per tant $Q_{ij} = R_{2_{ij}} = +1$.
- $a_j \neq cm_i$. En aquest cas l'element $R_{2_{ij}} = 0$. Cal llavors observar el valor del terme $-(L_2 L_1^{-1} R_1)_{ij} = -(L_2^i V_j)$. Podem considerar dos subcasos:
 - $cm_i \notin \mathcal{P}_j$. Clarament llavors el producte $L_2^i V_j = 0$, donat que l'arc cm_i no apareix en el cicle de l'arc a_j .
 - $cm_i \in \mathcal{P}_j$. En aquest cas l'arc cm_i sí que apareix en el cicle de l'arc a_j , per tant $L_{2_j}^i = 1$. Si cm_i té orientació normal a \mathcal{P}_j llavors l'element $V_{j_i} = -1$, i per tant $Q_{ij} = -(L_{2_j}^i V_{j_i}) = -(1 \times -1) = 1$. Si per contra cm_i té orientació inversa a \mathcal{P}_j llavors l'element $V_{j_i} = 1$, i per tant $Q_{ij} = -(L_{2_j}^i V_{j_i}) = -(1 \times 1) = -1$, amb el qual tots els casos queden tractats. ■

Algorímicament es pot descriure el procediment per calcular Q_{cm} com segueix:

Algorisme A4.1. Càlcul de Q_{cm} .

```

{ Inicialment  $Q_{ij} = 0, \forall i, j$  }
per j = 1 fins  $|\mathcal{A}|$ 
     $a_j := \text{arc\_complementari\_j}$ 
    si  $a_j \in \mathcal{A}_{cm}$  llavors
         $i := \text{índex}(a_j, \mathcal{A}_{cm})$ 
         $Q_{ij} := +1$ 
    si_no
         $\mathcal{P}_j := \text{calcular\_cicle}(a_j)$ 
        per_tot  $a \in \mathcal{P}_j$  fer
            si  $a \in \mathcal{A}_{cm}$  llavors
                 $i := \text{índex}(a, \mathcal{A}_{cm})$ 
                si té_orientació_normal( $a, \mathcal{P}_j$ ) llavors
                     $Q_{ij} := +1$ 
                si_no /* orientació inversa */
                     $Q_{ij} := -1$ 
            fi_si
        fi_per_tot
    fi_si
fi_per
    
```

4.2.2 Càlcul de Q_{cb} .

La següent proposició presenta un procediment de càlcul de la submatriu Q_{cb} .

Proposició 4.4.

Si denotem per:

- a_j l'arc associat amb la j -èssima columna de Q , $j = 1, \dots, |\mathcal{A}|$.
- cb_i la constricció a banda associada amb la i -èssima fila de Q , $i = |\mathcal{A}_{cm}| + 1, \dots, |\mathcal{A}|$.
- $B(a, n)$ una funció lògica que retorna *cert* si l'arc a apareix a la constricció a banda n , i *fals* altrament.

- $c_{a,n}$ el coeficient de l'arc a dins la constricció a banda n . Llavors podem calcular la matriu Q_{cb} fent:

$$Q_{ij} := \begin{cases} \text{Fer els següents 4 passos:} \\ 1) \text{ Inicialitzar } Q_{ij} = 0 \\ 2) \text{ si } B(a_j, cb_i) \text{ llavors } Q_{ij} = c_{a_j, cb_i} \\ \text{per tot } a \in \mathcal{P}_j, \text{ fer els següents 2 passos} \\ 3) \text{ si } B(a, cb_i) \text{ i } a \text{ té orientació normal llavors} \\ \quad Q_{ij} = Q_{ij} + c_{a, cb_i} \\ 4) \text{ si } B(a, cb_i) \text{ i } a \text{ té orientació inversa llavors} \\ \quad Q_{ij} = Q_{ij} - c_{a, cb_i} \end{cases}$$

Demostració.

Es pot comprovar (més aviat que demostrar) que la metodologia anterior és correcta només observant l'expressió del càlcul de Q_{cb} :

$$Q_{cb} = R_{2_{cb}} - L_{2_{cb}} L_1^{-1} R_1$$

Llavors el terme $Q_{cb_{ij}}$ és directament:

$$\begin{aligned} Q_{cb_{ij}} &= R_{2_{cb_{ij}}} - L_{2_{cb}}^i L_1^{-1} R_{1_j} \\ Q_{cb_{ij}} &= R_{2_{cb_{ij}}} - L_{2_{cb}}^i V_j \end{aligned}$$

(On $L_{2_{cb}}^i$ representa la fila i -èssima de $L_{2_{cb}}$, i R_{1_j} la columna j -èssima de R_1). El terme $R_{2_{cb_{ij}}}$ serà 0 si $B(a_j, cb_i) = fals$, i en cas contrari $R_{2_{cb_{ij}}} = c_{a_j, cb_i}$, amb el qual queden justificats els passos 1) i 2). Els passos 3) i 4) aplicats repetidament per a tot $a \in \mathcal{P}_j$ no reflecteixen més que el producte escalar $-L_{2_{cb}}^i V_j$, tenint en compte la definició de V_j a (4.18), el concepte d'orientació normal i inversa —prèviament exposat— i que

$$L_{2_{cb}}^i = \begin{cases} c_{a_l, cb_i} & \text{si } B(a_l, cb_i) = cert \\ 0 & \text{si } B(a_l, cb_i) = fals \end{cases}$$

La proposició anterior ja presenta de forma algorísmica el procés de construcció de la matriu Q_{cb} , pel qual no ens estendrem més sobre com implementar aquest càlcul.

Fins aquí hem presentat la forma de calcular la matriu Q . L'altre objectiu, l'actualització de la matriu de treball, serà l'objecte d'estudi de la següent secció.

4.3 Actualització eficient de la matriu de treball.

Un cop es té formada la matriu Q cal poder resoldre els sistemes del tipus $Qx = b$ i $x^t Q = b^t$. Cal tenir en compte que el nombre d'elements no-zero a Q en general serà petit (s'ha pogut observar fent diversos assaigs que acostuma a ser $< 10\%$). Per tant és recomanable usar rutines que tractin amb matrius esparses. A més, els sistemes abans

esmentats han de ser resolts a cada iteració del procés d'optimització. Per tant sembla lògic disposar d'un procediment per evitar calcular i fer la reinversió de Q a cada iteració. Un procés similar al que aquí descriurem ja va ser presentat a [49]. Tanmateix la tècnica introduïda en aquest treball considera dos principals modificacions respecte la metodologia original [49]:

- incorpora constriccions a banda en la formulació del problema.
- realitza una actualització de Q , en comptes de Q^{-1} tal i com es presentava a [49]. El fet de que Q pugui modificar la seva dimensió d'una iteració a una altra (com es veurà tot seguit), fa que actualitzar Q^{-1} sigui una operació costosa si s'emmagatzema com una matriu esparsa, donat que s'han d'afegir o eliminar files i/o columnes. Per altra banda, no semba gaire apropiat emmagatzemar-la com una matriu densa, donada la gran esparsitat d'aquesta matriu.

Abans de començar, però, descriurem les particularitats que ha de tenir el mètode d'actualització de Q . Per tal de fer això descriurem els diferents tipus de pivotació que poden donar-se a la base B del mètode del particionament primal durant el procés d'optimització.

4.3.1 Tipus de pivotacions.

Tant si estem minimitzant (o maximitzant) una funció lineal o una de no lineal, els procediments algorísmics utilitzats impliquen pivotacions a la base (una variable bàsica és reemplaçada per una no-bàsica, en el cas de funció lineal, o per una superbàsica en el cas de funció no lineal, com es veurà a seccions posteriors). Això implica que el conjunt \mathcal{A} de constriccions a banda i de capacitat mútua actives pot variar a cada iteració (donat que les variables de tipus folga associades a aquestes constriccions poden sortir de o entrar a la base), i pot fer-ho tant pel que fa a quines constriccions hi ha actives com al nombre d'elles. Aquest darrer fet (que el nombre de constriccions actives pugui variar d'una iteració a una altra) implica que la dimensió de la nostra matriu Q no és fixa durant tot el procés (recordem que $\dim(Q) = |\mathcal{A}| \times |\mathcal{A}|$). Llavors com a conseqüència immediata d'això tenim que el nostre mecanisme d'actualització de Q ha de ser capaç de tractar aquesta dimensionalitat variable de la matriu de treball.

Segons el tipus de variable que entra i surt de la base tenim sis pivotacions diferents. A continuació descriurem breument com afecta cada tipus de pivotació particular a la matriu Q , segons els tipus de variable que entra o surt de la base.

1) *Surt una folga o un arc complementari.*

- 1.a) Entra: folga – Surt: folga. La fila de Q associada amb la folga que entra a la base és eliminada i substituïda per una nova fila associada a la folga que surt de la base. Es modifica el conjunt \mathcal{A} de constriccions actives, però no la seva cardinalitat $|\mathcal{A}|$ (per tant $\dim(Q)$ no es modifica).

- 1.b) Entra: folga – Surt: arc complementari. La fila i columna de Q associades amb la folga que entra i l'arc complementari que surt de la base, respectivament, són eliminades. El conjunt \mathcal{A} de constriccions actives es redueix en un element (degut a la folga que entra a la base). Per tant hem d'actualitzar $\dim(Q) := \dim(Q) - 1$.
- 1.c) Entra: arc – Surt: folga. Una nova fila associada amb la folga que surt de la base és afegida a Q . Per tal de mantenir la no-singularitat de Q una nova columna per a l'arc que entra —el qual esdevindrà arc complementari— és també afegida a Q . El conjunt \mathcal{A} de constriccions actives és ampliat (amb la constricció associada a la folga que entra a la base) i per tant s'ha d'actualitzar $\dim(Q) := \dim(Q) + 1$.
- 1.d) Entra: arc – Surt: arc complementari. La columna de Q associada a l'arc complementari que surt de la base és eliminada, i serà reemplaçada per la columna associada al nou arc que entra a la base (el qual esdevindrà arc complementari). El conjunt \mathcal{A} no es modifica, i, per tant, $\dim(Q)$ tampoc.
- 2) *Surt un arc del i -èssim arbre d'expansió.*
- 2.a) Entra: folga – Surt: arc del i -èssim arbre. Un arc complementari del i -èssim article, per ex. el j -èssim arc complementari, tenint l'arc que deixa la base en el seu cicle \mathcal{P}_j , ha de ser buscat. Aquest j -èssim arc complementari sempre existirà (altrament la matriu Q esdevindria singular) i llavors passarà a formar part del i -èssim arbre d'expansió de la base. La fila i columna de Q associades a la folga que entra i el j -èssim arc que deixa de ser complementari són eliminades. El conjunt \mathcal{A} de constriccions actives es redueix en un element (degut a la folga que entra a la base). Per tant hem d'actualitzar $\dim(Q) := \dim(Q) - 1$.
- 2.b) Entra: arc – Surt: arc del i -èssim arbre. Es busca un arc complementari del i -èssim article, per ex. el j -èssim arc complementari, que tingui l'arc que deixa la base en el seu cicle \mathcal{P}_j . Si aquest arc és trobat llavors reemplaçarà l'arc que surt de la base en el i -èssim arbre d'expansió, i l'arc que entra a la base esdevindrà complementari. Si cap arc complementari és trobat verificant l'abans dit, llavors l'arc que entra a la base substituirà l'arc que surt de la base dins del i -èssim arbre. Un dels dos casos anteriors ha de ser sempre possible per tal de preservar la no-singularitat de la matriu Q . El conjunt \mathcal{A} no es modifica, i, per tant, $\dim(Q)$ tampoc.

Es pot observar a la casuística anterior com dels sis casos possibles de pivotació, en tres d'ells $\dim(Q)$ no es modifica, en dos $\dim(Q)$ disminueix en un, i en un cas la dimensió de Q es veu incrementada en un.

4.3.2 Mètode general.

En aquest subapartat descriurem la metodologia a seguir considerant una matriu general M que pot incrementar o disminuir la seva dimensió en una fila o columna a cada iteració. Més endavant centrarem l'atenció en el cas particular de tractar-se de la matriu

Q . Per això introduïrem el concepte de matriu estesa, i posteriorment veurem com fer la seva actualització.

4.3.2.1 Actualitzacions i recàlculs.

El concepte d'actualització i recàlcul de la factorització s'associa a processos iteratius on cal solucionar a cada iteració j un sistema d'equacions amb una matriu M_j , de forma que la matriu M_j és funció de l'anterior:

$$M_j = \psi(M_{j-1})$$

(per ex., M_j pot diferir només de M_{j-1} en una fila i/o columna). En aquest cas, en comptes de factoritzar la nova matriu M_j , resulta convenient realitzar (sempre que sigui possible i rentable) una actualització ψ_F de la factorització de M_{j-1} de forma que:

$$\text{Fact}(M_j) = \text{Fact}(\psi(M_{j-1})) = \psi_F(\text{Fact}(M_{j-1}))$$

Aplicant repetidament aquesta idea, podem escriure la factorització de M_j com

$$\text{Fact}(M_j) = \psi_F(\psi_F(\dots(\psi_F(M_0))\dots)) = \psi_F^j(\text{Fact}(M_0))$$

on ψ_F^j significa l'aplicació de j vegades la funció d'actualització ψ_F , i M_0 és la matriu a la iteració zero.

Clarament, al procediment anterior teòricament només caldria realitzar una única factorització a l'inici del procediment iteratiu. Això, però, podria provocar problemes de dos tipus. En primer lloc, caldria emmagatzemar tota la informació necessària per realitzar les j actualitzacions ψ_F , i si j es fa molt gran es requeriria una gran quantitat d'espai de memòria. En segon lloc, l'aplicació d'un nombre excessiu de vegades de ψ_F podria provocar errors de precisió numèrica, amb el qual la factorització que s'obtingria seria errònia. Per evitar aquest problema el que s'acostuma a fer és recalculer la factorització de M cada cert nombre d'iteracions (per ex., cada ν iteracions). Naturalment cal prendre un valor de ν suficientment "bo": és a dir ni massa petit (sinó estaríem recalculant i factoritzant la matriu M massa vegades) ni massa gran (per evitar possibles errors de precisió i un emmagatzemament excessiu d'informació per a les ψ_F).

4.3.2.2 Matriu estesa.

Considerem que a la iteració p del nostre procés d'optimització recalculerem la nostra matriu general M_p , que té dimensió $\dim(M_p) = n_p$, i que no serà recalculada de nou fins passades ν iteracions (això és fins la iteració $p+\nu$), on tindrà dimensió $\dim(M_{p+\nu}) = n_{p+\nu}$. Si considerem, a més, que la dimensió de M pot incrementar-se a cada iteració com a molt en una fila i una columna (com és el cas de la matriu Q), llavors directament tenim que $n_j \leq n_p + \nu$, $\forall j$ $p \leq j \leq p + \nu$. És a dir, la dimensió màxima de la matriu M_j entre les

iteracions p i $p + \nu$ és $n_p + \nu$. Llavors el mètode proposat consisteix en treballar amb la matriu estesa \overline{M}_j a les iteracions j , $p \leq j \leq p + \nu$, on \overline{M}_j es defineix com

$$\overline{M}_j = \begin{matrix} & n_j & l_j \\ n_j & \left(\begin{matrix} M_j & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{matrix} \right) & \\ l_j & & \end{matrix} \quad (4.19)$$

Les dimensions n_j i l_j de les matriu M_j i la identitat $\mathbf{1}$ satisfan $n_j + l_j = n_p + \nu$, això és, la matriu estesa \overline{M}_j té en tot moment la màxima dimensió que pot assolir la matriu M_j entre les iteracions p i $p + \nu$.

Llavors l'estructura que s'anirà mantenint actualitzada serà de fet la de les matrius esteses \overline{M}_j , tot i que ens interressi resoldre els sistemes $M_j x_j = b_j$ i $x_j^t M_j = b_j^t$. De fet aquest càlcul és directe a partir de \overline{M}_j i treballant amb \overline{x}_j i \overline{b}_j , extensions de x_j i b_j de forma que

$$\overline{x}_j = \begin{matrix} n_j \\ l_j \end{matrix} \begin{pmatrix} x_j \\ \alpha_j \end{pmatrix} \quad \overline{b}_j = \begin{matrix} n_j \\ l_j \end{matrix} \begin{pmatrix} b_j \\ 0 \end{pmatrix}$$

Llavors:

$$\overline{M}_j \overline{x}_j = \overline{b}_j \iff \begin{pmatrix} M_j & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} x_j \\ \alpha_j \end{pmatrix} = \begin{pmatrix} b_j \\ 0 \end{pmatrix} \iff \begin{cases} \alpha_j = 0 \\ M_j x_j = b_j \end{cases}$$

Obviant les darreres l_j components de \overline{x}_j (és a dir, el terme α_j) tenim directament x_j que era el valor desitjat. Així mateix, el cost pel que fa a temps de càlcul no és pas més elevat, i pel que fa a l'espai de memòria només cal emmagatzemar les l_j darreres components dels vectors \overline{x}_j i \overline{b}_j . De forma anàloga s'opera en resoldre $x_j^t M_j = b_j^t$.

4.3.2.3 Actualització de la matriu estesa.

Fins ara hem vist que, pel que fa a la resolució dels sistemes desitjats, és indiferent treballar amb la matriu M_j que amb la matriu estesa \overline{M}_j , amb l'avantatge de que la matriu estesa té dimensió constant. A continuació mostrarem com fer l'actualització de M_{j+1} en funció de M_j . Abans, però, introduïrem el concepte de *matriu eta*. Una matriu eta és una matriu que difereix de la matriu identitat en només una columna (matriu eta columna), o en una fila (matriu eta fila). Per a més detalls sobre matrius eta pot consultar-se [53]. Aleshores pot comprovar-se que l'actualització de M_{j+1} a partir de M_j pot fer-se en tots els casos pre i post multiplicant la matriu M_j amb matrius etes i de permutació, sempre i quan mantinguem l'estructura abans detallada de la matriu estesa com una "condició invariant" que cal preservar a cada iteració. Considerem els casos següents:

a) Canvia una fila de M_j en passar a M_{j+1} .

En aquest cas $\dim(M_{j+1}) = \dim(M_j) \iff n_j = n_{j+1}$, $l_j = l_{j+1}$. Llavors simplement cal premultiplicar per una matriu eta fila escaient, per ex. E_{f_j} , i llavors $M_{j+1} = E_{f_j} M_j$.

Donat que el que ens interessa és actualitzar la matriu estesa \overline{M}_{j+1} en realitat el que es fa és

$$\overline{M}_{j+1} = \overline{E}_{f_j} \overline{M}_j \quad (4.20)$$

(on \overline{E}_{f_j} és la matriu estesa de E_{f_j} de forma anàloga a com s'ha definit \overline{M}_j a (4.19)).

b) Canvia una columna de M_j en passar a M_{j+1} .

Com al cas anterior $\dim(M_{j+1}) = \dim(M_j)$, i només cal afegir una matriu eta columna escaient, per ex. E_{c_j} , per postmultiplicar \overline{M}_j , de forma que $M_{j+1} = M_j E_{c_j}$. Directament llavors tenim que

$$\overline{M}_{j+1} = \overline{M}_j \overline{E}_{c_j} \quad (4.21)$$

c) Eliminem una fila i una columna de M_j en passar a M_{j+1} .

En aquest cas clarament $\dim(M_{j+1}) = \dim(M_j) - 1 \Leftrightarrow n_{j+1} = n_j - 1$, $l_{j+1} = l_j + 1$. Considerem que volem eliminar la fila r i columna s de M_j . El primer que es farà és intercanviar la fila r amb la darrera fila de M_j (és a dir amb la fila n_j) i la columna s amb la darrera columna de M_j (que és la columna n_j). Per tal d'intercanviar les files premultiplicarem per una matriu de permutació escaient que anomenarem P_{f_j} i per intercanviar les columnes postmultiplicarem per una altra matriu de permutació, que denotarem per P_{c_j} . Llavors tindrem la matriu intermèdia $\hat{M}_j = P_{f_j} M_j P_{c_j}$. Cal observar que la submatriu formada per les $n_j - 1$ primeres files i columnes de \hat{M}_j és la matriu M_{j+1} desitjada (bé, realment és “gairebé” la matriu M_{j+1} desitjada, donat que les matrius de permutació P_{f_j} i P_{c_j} ens han situat la fila n_j de M_j a la posició r i la columna n_j a la posició s , quan el que haurien d'haver fet és avançar una posició totes les files compreses entre la fila r i la fila n_j i el mateix amb les columnes entre la s i la n_j . El que succeeix és que és més eficient en temps de càlcul fer l'intercanvi aquí proposat que el desplaçament de files i columnes. Per tant considerarem com M_{j+1} la submatriu quadrada de dimensió $n_j - 1$ de \hat{M}_j abans esmentada).

Si consideréssim ara la matriu estesa $\overline{\hat{M}}_j$, la columna i fila n_j de \hat{M}_j (les originals fila r i columna s de M_j) formarien part de la zona on hauria d'aparèixer la matriu identitat, donat que aquesta és la condició invariant que hem de preservar. Llavors de forma anàloga a com hem fet als casos a) i b) hem de pre i postmultiplicar \hat{M}_j per matrius eta fila (E_{f_j}) i eta columna (E_{c_j}) escaients per convertir les originals fila r i columna s en una fila i columna de la matriu identitat. Si considerem tots els càlculs ja amb matrius esteses tenim finalment que

$$\overline{M}_{j+1} = \overline{E}_{f_j} \overline{P}_{f_j} \overline{M}_j \overline{P}_{c_j} \overline{E}_{c_j} \quad (4.22)$$

d) Afegim una fila i una columna de M_j en passar a M_{j+1} .

En aquest cas $n_{j+1} = n_j + 1$. Sabent l'aspecte que té la matriu estesa \overline{M}_j a la fila i columna $n_j + 1$ (són una fila i columna de la matriu identitat) directament podem trobar unes matrius eta fila E_{f_j} i eta columna E_{c_j} escaients per tal de transformar aquestes fila

i columna de la matriu identitat en la fila i columna noves a afegir a M_{j+1} . Llavors tenim que

$$\overline{M}_{j+1} = \overline{E}_{f_j} \overline{M}_j \overline{E}_{c_j} \quad (4.23)$$

Pot observar-se com és d'important mantenir la condició de que la matriu subdiagonal dreta inferior de la matriu estesa que actualitzem hagi de ser igual a alguna cosa coneguda i invariant (en aquest cas la matriu identitat), ja que si no seria imposible trobar \overline{E}_{f_j} i \overline{E}_{c_j} tals que multiplicades per “alguna cosa” —aquesta cosa és per a nosaltres coneguda perquè sabem que tenim una matriu identitat a la part subdiagonal de la matriu estesa— donessin la fila i columnes desitjades.

De les equacions (4.20–4.23) es pot concloure que en tot moment podem actualitzar \overline{M}_{j+1} en funció de \overline{M}_j de forma

$$\overline{M}_{j+1} = G_j \overline{M}_j F_j \quad (4.24)$$

estant formades G_j i F_j per matrius etes als casos a), b) i d) previs, i per matrius etes i de permutació al cas c). De forma inductiva podem escriure $\overline{M}_{j+1} = G_j G_{j-1} \overline{M}_{j-1} F_{j-1} F_j$, i així podríem continuar fins arribar al punt en que vam recalculer M_p a la iteració p . Llavors podem escriure de forma general

$$\forall j, \quad p \leq j < p + \nu \quad \overline{M}_j = G \overline{M}_p F \quad (4.25)$$

on

$$\begin{aligned} G &= \prod_{l=1}^{j-p} G_{j-l} \\ F &= \prod_{l=1}^{j-p} F_{p+l-1} \end{aligned} \quad (4.26)$$

Llavors per tal de resoldre els sistema $\overline{M}_j \overline{x}_j = \overline{b}_j$ cal fer:

$$\begin{aligned} \overline{M}_j \overline{x}_j &= \overline{b}_j \\ G \overline{M}_p F \overline{x}_j &= \overline{b}_j \\ \overline{M}_p F \overline{x}_j &= G^{-1} \overline{b}_j \\ \overline{M}_p z_j &= G^{-1} \overline{b}_j, \quad \text{on } z_j = F \overline{x}_j \\ z_j &= \overline{M}_p^{-1} G^{-1} \overline{b}_j \end{aligned} \quad (4.27)$$

I finalment $\overline{x}_j = F^{-1} z_j$

El sistema $\overline{x}_j^t \overline{M}_j = \overline{b}_j^t$ es resol de forma similar.

És a dir, per tal de resoldre els sistemes desitjats cal invertir els productes de matrius G i F i tenir alguna forma de calcular sistemes amb la matriu \overline{M}_p (que és el mateix que calcular sistemes amb la matriu M_p) calculada a la iteració p . Pel que fa a les inverses

de G i F , aquestes són directament calculables, donat que són productes de matrius etes i de permutació que tenen inverses conegudes. De fet, des del punt de vista d'eficiència computacional, és preferible no emmagatzemar G i F com producte de matrius etes i de permutació, sinó emmagatzemar directament G^{-1} i F^{-1} com producte de les inverses de les matrius de permutació i matrius etes (que continuen essent matrius de permutació i matrius etes). Per la seva banda, per tal de calcular sistemes amb la matriu M_p només cal reinvertir una vegada a la iteració p aquesta matriu, i després s'usarà la reinversió a cada iteració j , $p \leq j < p + \nu$.

Es veu clarament com el fet de treballar amb la matriu estesa (de dimensió $n_p + \nu$) elimina els problemes d'una actualització amb dimensió variable. Naturalment cal prendre un valor de ν —és a dir, cada quantes iteracions recalculem la matriu M — el suficientment “bo”: és a dir ni massa petit (sinó estariem recalculant i invertint la matriu M massa vegades) ni massa gran (ja que llavors la dimensió de la matriu estesa seria gran i, a més, augmentariem els possibles errors de precisió creats en afegir matrius etes i de permutació a cada iteració). Un punt important és el fet de mantenir a la matriu estesa el bloc diagonal de la part dreta inferior igual a la matriu identitat. Aquesta condició s'ha de considerar com una “condició invariant”, que s'ha de preservar en tot moment, i gràcies a ella sabrem com passar de M_j a M_{j+1} . Es podria haver escollit una altra condició invariant (per exemple, mantenir no la matriu identitat sinó una matriu diagonal de dosos) i tenint en compte això tot funcionaria d'igual forma. Tanmateix s'ha escollit la matriu identitat perquè simplifica el procés d'actualització.

4.3.3 Mètode general aplicat a la matriu Q en particular.

En aquesta secció es particularitzarà la metodologia descrita a l'apartat anterior en el cas concret de que la matriu a actualitzar sigui la matriu Q del mètode del particionament primal. Es tractarà l'actualització de Q per als sis casos possibles de pivotació que s'han descrit a §4.3.1. Es farà una divisió clara en la forma d'actualitzar segons abandonin la base o bé una variable folga o arc complementari, o bé una variable pertanyent a un arbre bàsic. Els dos tipus d'actualització es tractaran en dos apartats diferents. Prèviament, però, introduïrem uns conceptes previs comuns a ambdós tipus d'actualitzacions (una descripció alternativa d'aquests conceptes previs pot ser trobada a [49]).

4.3.3.1 Conceptes previs.

Recordem que al mètode del particionament primal la base a la iteració qualsevol j pot ser escrita com:

$$B_j = \begin{array}{|c|c|c|} \hline L_1 & R_1 & \mathbf{0} \\ \hline L_2 & R_2 & \mathbf{0} \\ \hline L_3 & R_3 & \mathbf{1} \\ \hline \end{array}$$

Considerem ara la matriu L_j que definirem com:

$$L_j = \begin{array}{|c|c|c|} \hline \mathbf{1} & -L_1^{-1}R_1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \end{array}$$

Llavors el producte $B_j^* = B_j L_j$ és:

$$B_j^* = B_j L_j = \begin{array}{|c|c|c|} \hline L_1 & R_1 & \mathbf{0} \\ \hline L_2 & R_2 & \mathbf{0} \\ \hline L_3 & R_3 & \mathbf{1} \\ \hline \end{array} \begin{array}{|c|c|c|} \hline \mathbf{1} & -L_1^{-1}R_1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline L_1 & \mathbf{0} & \mathbf{0} \\ \hline L_2 & \mathbf{Y}_j & \mathbf{0} \\ \hline L_3 & \mathbf{Y}_j & \mathbf{1} \\ \hline \end{array}$$

On Y_j es defineix com:

$$Y_j = \begin{array}{|c|c|} \hline R_2 - L_2 L_1^{-1} R_1 & \mathbf{0} \\ \hline R_3 - L_3 L_1^{-1} R_1 & \mathbf{1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline Q_j & \mathbf{0} \\ \hline D_j & \mathbf{1} \\ \hline \end{array} \quad (4.28)$$

Cal observar que la submatriu diagonal superior de Y_j correspon exactament a la matriu de treball Q del mètode del particionament primal. Llavors actualitzar $Q_{j+1} = \psi(Q_j)$ es redueix a actualitzar $Y_{j+1} = \psi(Y_j)$.

Cada pivotació a la base B_j implica un canvi de columna a la mateixa. Aquesta operació pot ser representada matricialment mitjançant un producte amb una matriu eta

(o matriu de permutació en el cas de que en comptes d'introduir una nova variable el que fem sigui permutar dues columnes de la base actual; de totes formes tant en un cas com a l'altre representarem sempre amb E la matriu usada per fer l'actualització, on E tant pot ser una matriu eta com una de permutació) de forma $B_{j+1} = B_j E$ —eliminem el subíndex j de la matriu eta per simplificar la notació—. Tenint en compte la definició de B_j^* i l'expressió d'actualització de B_j a base de matrius E , obtenim:

$$\begin{aligned} B_{j+1}^* &= B_{j+1} L_{j+1} \\ B_{j+1}^* &= B_j E L_{j+1} \\ B_{j+1}^* &= B_j^* L_j^{-1} E L_{j+1} \end{aligned} \quad (4.29)$$

Directament de l'expressió (4.29) obtenim (fent la inversa de la part dreta i esquerra):

$$B_{j+1}^{*-1} = L_{j+1}^{-1} E^{-1} L_j B_j^{*-1} \quad (4.30)$$

De fet per als nostres propòsits serà més útil l'expressió amb la inversa (4.30) que l'equació (4.29). Per tal d'expressar l'equació (4.30) de forma matricial simplificada, considerarem les següents particions de les matrius B_j^* i L_j :

$$B_j^* = \begin{array}{|c|c|c|} \hline L_1 & \mathbf{0} & \mathbf{0} \\ \hline L_2 & \mathbf{Y}_j & \\ \hline L_3 & & \\ \hline \end{array} = \begin{array}{|c|c|} \hline C_j & \mathbf{0} \\ \hline F_j & Y_j \\ \hline \end{array} \quad \text{on} \quad C_j = L_1 \quad , \quad F_j = \begin{array}{|c|} \hline L_2 \\ \hline L_3 \\ \hline \end{array}$$

$$L_j = \begin{array}{|c|c|c|} \hline \mathbb{1} & -L_1^{-1} R_1 & \mathbf{0} \\ \hline \mathbf{0} & \mathbb{1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbb{1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \mathbb{1} & V_j \\ \hline \mathbf{0} & \mathbb{1} \\ \hline \end{array} \quad \text{on} \quad V_j = \begin{array}{|c|c|} \hline -L_1^{-1} R_1 & \mathbf{0} \\ \hline \end{array}$$

A l'expressió (4.30), però, són necessàries les inverses de les matrius L_{j+1} i B_j^* . A partir del particionament anterior pot veure's que les inverses anteriors poden expressar-se de forma:

$$L_{j+1}^{-1} = \begin{array}{|c|c|} \hline \mathbb{1} & -V_{j+1} \\ \hline \mathbf{0} & \mathbb{1} \\ \hline \end{array} \quad B_j^{*-1} = \begin{array}{|c|c|} \hline \hat{C}_j & \mathbf{0} \\ \hline \hat{F}_j & Y_j^{-1} \\ \hline \end{array} \quad \text{on} \quad \hat{C}_j = C_j^{-1} \quad \text{i} \quad \hat{F}_j = -F_j C_j^{-1} Y_j^{-1}$$

Per tant l'equació (4.30) en forma matricial pot ser escrita com:

$$\begin{array}{|c|c|} \hline \hat{C}_{j+1} & \mathbf{0} \\ \hline \hat{F}_{j+1} & Y_{j+1}^{-1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \mathbf{1} & -V_{j+1} \\ \hline \mathbf{0} & \mathbf{1} \\ \hline \end{array} E^{-1} \begin{array}{|c|c|} \hline \mathbf{1} & V_j \\ \hline \mathbf{0} & \mathbf{1} \\ \hline \end{array} \begin{array}{|c|c|} \hline \hat{C}_j & \mathbf{0} \\ \hline \hat{F}_j & Y_j^{-1} \\ \hline \end{array} \quad (4.31)$$

A partir de l'expressió (4.31) es pot obtenir la formulació per tal d'actualitzar Q . En fer l'actualització distingirem dos casos segons el tipus de variable que surt de la base, el qual modifica substancialment l'aspecte de la matriu E^{-1} (encara no descrita amb molt detall) que apareix a l'equació (4.31). Presentem a continuació els dos casos esmentats en funció de la variable que deixa la base. Caldrà tenir present en tot moment, però, que la matriu que actualitzarem realment serà: $\bar{Q}_j = \begin{pmatrix} Q_j & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}$.

4.3.3.2 Surt de la base una folga o arc complementari.

Aquest és el primer dels casos descrit a §4.3.1, que ahora es subdivideix en quatre casos més. En sortir una folga o un arc complementari de la base les submatrius L_1 , L_2 i L_3 resten intactes, i la matriu E de l'equació (4.31) per reflectir el canvi a la base i la seva inversa serà de la forma:

$$E = \begin{array}{|c|c|} \hline \mathbf{1} & E_2 \\ \hline \mathbf{0} & E_4 \\ \hline \end{array} \quad E^{-1} = \begin{array}{|c|c|} \hline \mathbf{1} & \hat{E}_2 \\ \hline \mathbf{0} & E_4^{-1} \\ \hline \end{array} \quad \text{on } \hat{E}_2 = -E_2 E_4^{-1}$$

Tenint en compte que la inversa d'una matriu eta també és una matriu eta, llavors la matriu E^{-1} ha de ser matriu eta, i per tant també ho seran les matrius \hat{E}_2 i E_4^{-1} en particular. Substituint el valor de E^{-1} anterior a l'equació (4.31) obtenim:

$$\begin{array}{|c|c|} \hline \hat{C}_{j+1} & \mathbf{0} \\ \hline \hat{F}_{j+1} & Y_{j+1}^{-1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \mathbf{1} & -V_{j+1} \\ \hline \mathbf{0} & \mathbf{1} \\ \hline \end{array} \begin{array}{|c|c|} \hline \mathbf{1} & \hat{E}_2 \\ \hline \mathbf{0} & E_4^{-1} \\ \hline \end{array} \begin{array}{|c|c|} \hline \mathbf{1} & V_j \\ \hline \mathbf{0} & \mathbf{1} \\ \hline \end{array} \begin{array}{|c|c|} \hline \hat{C}_j & \mathbf{0} \\ \hline \hat{F}_j & Y_j^{-1} \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline \hat{C}_{j+1} & \mathbf{0} \\ \hline \hat{F}_{j+1} & Y_{j+1}^{-1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \mathbf{1} & \hat{E}_2 - V_{j+1} E_4^{-1} \\ \hline \mathbf{0} & E_4^{-1} \\ \hline \end{array} \begin{array}{|c|c|} \hline \hat{C}_j + V_j \hat{F}_j & V_j Y_j^{-1} \\ \hline \hat{F}_j & Y_j^{-1} \\ \hline \end{array}$$

$$\begin{array}{|c|c|} \hline \hat{C}_{j+1} & \mathbf{0} \\ \hline \hat{F}_{j+1} & Y_{j+1}^{-1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline * & * \\ \hline * & E_4^{-1}Y_j^{-1} \\ \hline \end{array}$$

(A l'expressió anterior els termes * serien els resultats obtinguts en fer el producte per blocs, obviats aquí per no complicar la notació i per ser del tot irrelevants per als nostres propòsits). Igualant per blocs directament s'obté:

$$Y_{j+1}^{-1} = E_4^{-1}Y_j^{-1} \iff Y_{j+1} = Y_j E_4 \tag{4.32}$$

Cal recordar que la columna (o vector) eta de E_4 (l'única que és diferent de $\mathbf{0}$ i necessària per fer l'actualització) és un valor ja calculat prèviament durant el procés d'optimització. Aquesta columna eta és de fet el vector P trobat en resoldre $BP = N$, essent B la base en el moment actual i N una columna associada amb una variable no bàsica (en el cas lineal) o superbàsica (en el cas no lineal, com més tard es veurà). A continuació, i basant-nos en l'equació (4.32) per tal de fer l'actualització, descriurem cadascun dels quatre subcasos possibles que poden donar-se.

1.a) Entra: folga - Surt: folga.

La folga que entra a la base es trobava associada a una constricció (a banda o de capacitat mútua) saturada, i per tant, li corresponia una fila dins la matriu Q_j . Aquesta fila serà substituïda per una fila amb la constricció (a banda o de capacitat mútua) associada a la folga que surt de la base —és a dir, la constricció associada a aquesta folga que surt de la base es transforma en constricció activa. Per tant l'operació que fem és substituir una fila de Q_j —sigui la fila r , Q_j^r — per una fila de D_j —sigui la fila s , D_j^s —, estant definida D_j a l'expressió (4.28). La dimensió de la matriu Q no varia en aquest cas.

Recordant el dit al mètode general per qualsevol matriu, en aquest cas només cal premultiplicar la matriu Q_j per una matriu eta fila escaient. Per trobar aquesta matriu eta fila, resollem en primer lloc el sistema

$$\gamma^t Q_j = D_j^s$$

Primer veiem com trobar E_c . Si recordem la fórmula (4.32) per tal d'actualitzar la matriu Y_{j+1} i l'expressem en forma de matrius per blocs tenim:

$$Y_{j+1} = \begin{array}{|c|c|} \hline Q_j & \mathbf{0} \\ \hline D_j & \mathbf{1} \\ \hline \end{array} \quad E_4 = \begin{array}{|c|c|} \hline Q_j & \mathbf{0} \\ \hline D_j & \mathbf{1} \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline E_5 & \mathbf{0} \\ \hline E_6 & \mathbf{1} \\ \hline \end{array}$$

on hem particionat E_4 de forma escaient. Observem per tant com directament la submatriu E_5 el que fa es transformar la columna original s de Y_j que sortia de la base en una columna identitat. Llavors directament tenim que la matriu E_c desitjada és $E_c = E_5$ (i recordem que E_4 és ja una matriu coneguda, cosa que ens estalvia temps de càlcul per trobar E_c).

Per tal de trobar E_f operarem com al subcàs (1.a). Llavors buscarem un vector γ tal que

$$\gamma(\hat{Q}_{j+1}E_c) = e_{n_j}$$

essent e_{n_j} l' n_j -èssim vector de la matriu identitat. Amb aquest vector γ formem llavors directament la matriu E_f , amb γ situat a la fila n_j -èssima.

Finalment l'actualització en aquest subcàs queda com:

$$\begin{aligned} Q_{j+1} &= E_f \hat{Q}_{j+1} E_c \\ Q_{j+1} &= E_f P_f Q_j P_c E_c \end{aligned}$$

Donat que hem d'actualitzar la matriu estesa, directament escrivim:

$$\overline{Q}_{j+1} = \overline{E}_f \overline{P}_f \overline{Q}_j \overline{P}_c \overline{E}_c$$

Observem com l'únic càlcul necessari en aquest cas torna a ser trobar el vector γ de la matriu E_f .

1.c) Entra: arc - Surt: folga.

En aquest cas la matriu Q_{j+1} s'amplia amb una nova fila (associada a la constricció a banda o de capacitat mútua de la folga que surt a la base) i amb una nova columna (associada a l'arc que entra a la base i que esdevindrà arc complementari). Per tant en aquest cas la dimensió n_{j+1} de Q_{j+1} serà $n_{j+1} = n_j + 1$, i conseqüentment la submatriu identitat diagonal inferior de \overline{Q}_{j+1} disminuirà la seva dimensió respecte la iteració anterior, és a dir $l_{j+1} = l_j - 1$.

La nova fila a afegir a la nova Q_{j+1} serà una fila pertanyent a la submatriu D_j de (4.28). Considerem que sigui la fila s , D_j^s . Llavors ens interessa poder arribar a una

expressió del tipus:

$$Q_{j+1} = \begin{array}{|c|c|} \hline Q_j & 0 \\ \hline \hline \overline{D}_j^s & \mathbb{1} \\ \hline \hline \end{array} \quad \begin{array}{|c|c|} \hline 1 & \\ \hline \cdot & \\ \hline \cdot & \\ \hline \cdot & \\ \hline \cdot & \\ \hline \cdot & \\ \hline \cdot & \\ \hline \cdot & \\ \hline 1 & \gamma \\ \hline \hline \end{array}$$

essent γ la columna eta de la matriu E_c . Recordant de nou la fórmula (4.32) d'actualització de Y_{j+1} per blocs:

$$Y_{j+1} = \begin{array}{|c|c|} \hline Q_j & \mathbf{0} \\ \hline D_j & \mathbb{1} \\ \hline \hline \end{array} \quad \begin{array}{|c|c|} \hline E_5 & \mathbf{0} \\ \hline E_6 & \mathbb{1} \\ \hline \hline \end{array}$$

observem que la submatriu E_5 ens afegeix la columna associada amb l'arc que entra a la base. Per tant directament tenim que $E_c = E_5$. Podem observar que, llavors, l'actualització pot ser escrita com:

$$Q_{j+1} = \begin{array}{|c|c|} \hline Q_j & 0 \\ \hline \hline \overline{D}_j^s & \mathbb{1} \\ \hline \hline \end{array} \quad \begin{array}{|c|} \hline E_5 \\ \hline \hline \end{array}$$

$$= \begin{array}{|c|c|} \hline Q_j & 0 \\ \hline \hline 0 & \mathbb{1} \\ \hline \hline \end{array} \quad \begin{array}{|c|c|} \hline \mathbb{1} & 0 \\ \hline \hline \overline{D}_j^s & \mathbb{1} \\ \hline \hline \end{array} \quad \begin{array}{|c|} \hline E_5 \\ \hline \hline \end{array}$$

Com s'observa, per tal de poder fer l'actualització de Q_{j+1} d'aquesta forma és necessari tenir una fila i columna de la matriu identitat a la posició n_{j+1} (posició de les noves fila i columna afegides a la matriu de treball). Però, donat que el que fem és actualitzar la matriu estesa de \overline{Q}_{j+1} aquesta condició ja es verifica, ja que la matriu diagonal inferior de la matriu estesa és precisament la matriu identitat. Per tant podem escriure l'actualització de la matriu estesa d'aquest subcàs com:

$$\overline{Q}_{j+1} = \overline{Q}_j \overline{E}_f \overline{E}_c$$

essent \overline{E}_f la matriu eta fila amb el vector eta D_j^s a la posició n_{j+1} . Pel que fa al cost computacional s'observa que només és necessari calcular $D_j^s = (R_3 - L_3 L_1^{-1} R_1)^s$ per tenir E_f ; a diferència dels casos anteriors calcular la matriu E_f no suposa resoldre cap sistema d'equacions amb Q_j .

1.d) Entra: arc - Surt: arc complementari. En aquest cas la dimensió de la matriu de treball no varia. L'única operació a fer és substituir la columna associada a l'arc complementari que deixa la base per una altra associada a l'arc que entra a la base, el qual esdevindrà complementari. Directament a partir de la fórmula (4.32) considerant les matrius particionades per blocs, tal i com s'ha fet als subcasos anteriors, obtenim l'actualització per aquest subcàs (el qual pot ser formulat a partir de matrius esteses):

$$\overline{Q}_{j+1} = \overline{Q}_j \overline{E}_c, \quad \text{on } \overline{E}_c = \overline{E}_5$$

Com s'observa aquest cas no necessita de cap càlcul suplementari.

4.3.3.3 Surt de la base un arc del i -èssim arbre d'expansió.

Tal i com vam detallar en apartats anteriors, en el cas de sortir un arc pertanyent a l'arbre d'expansió mínima de l'article i mirarem si existeix algun arc complementari del mateix article que el pugui substituir (en el cas que la variable que entri a la base sigui una folga aquest arc existirà sempre, mentre que en el cas de que entri un arc pot o no existir).

Considerem que deixa la base l'arc s de l'arbre per a l'article i , $B^{(i)}$. Els cicles dels arcs complementaris de l'article i venen donats per $V^{(i)} = B^{(i)-1} R_1^{(i)}$, essent $R_1^{(i)}$ la submatriu de R_1 que conté els arcs complementaris de l'article i . Llavors la condició de "veure si algun arc complementari pot substituir a l'arc s que deixa la base" és equivalent a "buscar algun arc complementari tal que en el seu cicle aparegui l'arc s ". A partir de $V^{(i)}$ (que ens dona tots els cicles dels arcs complementaris de l'article i) aquesta darrera condició es redueix a veure si $\gamma = V^{(i)s}$ és o no un vector igual a 0 (on $V^{(i)s}$ representa la fila s de $V^{(i)}$). Llavors segons $\gamma = 0$ o $\gamma \neq 0$ tindrem els dos subcasos que a continuació descriurem.

2.a) $\gamma = 0$.

En aquest cas cap arc complementari pot ser intercanviat per l'arc que surt de l'arbre d'expansió mínima. Llavors el nou arc entrant substituirà l'arc que surt dins l'arbre. Implícitament assumim, aleshores, que quan $\gamma = 0$ la variable que entra a la base ha de ser un arc i no una folga (en cas de que fos una folga no podríem mantenir l'arbre d'expansió mínima i la base esdevindria singular). A partir de la definició de Q , l'expressió de la nova Q_{j+1} serà ara:

$$Q_{j+1} = R_{2_{j+1}} - L_{2_{j+1}} (L_{1_{j+1}}^{-1} R_{1_{j+1}})$$

on es verifica que $R_{2_{j+1}} = R_{2_j}$ i que L_{2_j} i $L_{2_{j+1}}$ només difereixen en la columna associada amb els arcs que han estat intercanviats (arc que ha sortit i arc que ha entrat). Pel que fa al terme $L_{1_{j+1}}^{-1} R_{1_{j+1}}$ amb els cicles dels complementaris dins els arbres, hem de tenir en compte que $R_{1_{j+1}} = R_{1_j}$ (és a dir, el conjunt d'arcs complementaris roman invariable), mentre que $L_{1_{j+1}}$ i L_{1_j} difereixen (igual a com passava amb L_2) en la columna de l'arc sortint/entrant. Donat que l'únic canvi que hi hagut en els arbres bàsics ha estat el reemplaçament de l'arc que ha sortit de $B^{(i)}$ (que anomenarem a_s) pel que ha entrat (representat per a_e), llavors

l'única possible diferència en els cicles pot ser que l'arc a_e aparegui al cicle d'algun arc complementari de l'article i . La següent proposició demostra que això no és possible.

Proposició 4.5.

Signi a_s l'arc que surt de $B^{(i)}$, a_e l'arc que el reemplaçarà a $B^{(i)}$, $\mathcal{R}_1^{(i)}$ el conjunt d'arcs complementaris de l'article i (associats a les columnes de $R_1^{(i)}$), i \mathcal{P}_a el conjunt d'arcs de $B^{(i)}$ que formen el cicle de l'arc a .

Si $\forall a \in \mathcal{R}_1^{(i)}$, $a_s \notin \mathcal{P}_a$ llavors també es verificarà després de fer la pivotació que $a_e \notin \mathcal{P}_a$

Demostració.

Suposem que després de fer la pivotació $\exists a \in \mathcal{R}_1^{(i)}$ tal que $a_e \in \mathcal{P}_a$. Això implica que hem hagut de trencar en pivotar el cicle previ de a per a que pogués entrar a_e . Però per trencar el cicle previ de a l'única possibilitat és que l'arc que surt a_s estigui contingut en ell. Això, però, contradiu la hipòtesi inicial de que $\forall a \in \mathcal{R}_1^{(i)}$, $a_s \notin \mathcal{P}_a$. Per tant no pot haver cap arc a tal que $a_e \in \mathcal{P}_a$. ■

Llavors tenim que els cicles dels arcs complementaris no queden modificats tampoc en fer la pivotació. A més a l'expressió $Q_{j+1} = R_{2_{j+1}} - L_{2_{j+1}}(L_{1_{j+1}}^{-1} R_{1_{j+1}})$, donat que l'arc a_e no apareix dins cap cicle dels complementaris —tot i que $L_{2_{j+1}}$ s'hagi modificat just en la columna de a_e — el terme $L_{2_{j+1}}(L_{1_{j+1}}^{-1} R_{1_{j+1}})$ no varia respecte la iteració anterior. Llavors tenim directament que

$$Q_{j+1} = Q_j \iff \bar{Q}_{j+1} = \bar{Q}_j$$

En aquest cas, doncs, la matriu de treball roman invariant en fer la pivotació.

2.b) $\gamma \neq 0$.

En aquest cas l'actualització es fa en dues etapes:

- i) En primer lloc prenem una posició r de γ tal que $\gamma_r \neq 0$. Llavors intercanviarem les columnes associades a l'arc s que surt de la base i l'arc complementari r . És a dir, el que fem realment és una permutació de dues columnes que ens preserven l'estructura general de la nostra base (aquesta és: tenir arbres d'expansió mínima per una banda i arcs complementaris per una altra).
- ii) Ara l'arc r que havíem de treure de la base es troba com arc complementari. Llavors en aquest segon pas el que hem de fer és aplicar la metodologia proposada als subcasos ja presentats segons la variable a entrar sigui una folga (amb el qual aplicarem el subcàs 1.b) o un arc (aplicant llavors el subcàs 1.d). Ara, però, en aplicar els subcasos 1.b) o 1.d) no disposarem directament del valor de la matriu $E_c = E_5$ com abans, i caldrà calcular-la.

Centrem-nos, però, en la primera de les etapes abans esmentades (on permutem l'arc s que deixa un arbre generador, i l'arc r complementari). Recordant l'expressió matricial (4.31) per fer l'actualització de B_{j+1}^* , veiem que en aquest cas la matriu E^{-1} no és realment

una matriu eta, sinó una matriu de permutació (entre les columnes dels arcs r i s). En comptes de parlar de matriu E ho farem de matriu P de permutació de dues columnes. Cal tenir en compte que, com amb tota matriu de permutació, es verifica que $PP = \mathbb{1}$, amb el qual es té que $P = P^{-1}$. Llavors avaluant l'expressió (4.31) obtenim:

$$\begin{array}{c}
 \begin{array}{|c|c|} \hline \hat{C}_{j+1} & \mathbf{0} \\ \hline \hat{F}_{j+1} & Y_{j+1}^{-1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline \mathbb{1} & -V_{j+1} \\ \hline \mathbf{0} & \mathbb{1} \\ \hline \end{array} \begin{array}{|c|c|} \hline P_1 & P_2 \\ \hline P_3 & P_4 \\ \hline \end{array} \begin{array}{|c|c|} \hline \mathbb{1} & V_j \\ \hline \mathbf{0} & \mathbb{1} \\ \hline \end{array} \begin{array}{|c|c|} \hline \hat{C}_j & \mathbf{0} \\ \hline \hat{F}_j & Y_j^{-1} \\ \hline \end{array} \\
 \\
 \begin{array}{|c|c|} \hline \hat{C}_{j+1} & \mathbf{0} \\ \hline \hat{F}_{j+1} & Y_{j+1}^{-1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline P_1 - V_{j+1}P_3 & P_2 - V_{j+1}P_4 \\ \hline P_3 & P_4 \\ \hline \end{array} \begin{array}{|c|c|} \hline \hat{C}_j + V_j\hat{F}_j & V_jY_j^{-1} \\ \hline \hat{F}_j & Y_j^{-1} \\ \hline \end{array} \\
 \\
 \begin{array}{|c|c|} \hline \hat{C}_{j+1} & \mathbf{0} \\ \hline \hat{F}_{j+1} & Y_{j+1}^{-1} \\ \hline \end{array} = \begin{array}{|c|c|} \hline * & * \\ \hline * & (P_3V_j + P_4)Y_j^{-1} \\ \hline \end{array}
 \end{array}$$

(A l'expressió anterior els termes $*$ serien els resultats obtinguts en fer el producte per blocs, obviats aquí per a no complicar la notació i ser del tot irrelevants per als nostres propòsits). Igualant per blocs tenim que

$$Y_{j+1}^{-1} = (P_3V_j + P_4)Y_j^{-1} \quad (4.33)$$

Donat que la matriu Q_j és una submatriu de Y_j (així com Q_{j+1} ho és de Y_{j+1}) tenim una forma d'actualitzar la matriu de treball en el cas de permutar els arcs r i s .

Per tal de poder desenvolupar l'expressió (4.33) detallarem amb més precisió la matriu P per tal de poder conèixer l'aspecte de P_3 i P_4 . Sabent que m és el nombre de nodes de la xarxa, n és el nombre d'arcs de la xarxa, k el nombre total d'articles, t el nombre de constriccions a banda, i l'article de l'arc s que deixa l'arbre $B^{(i)}$, r l'arc complementari que substituirà s dins $B^{(i)}$ i $r^* = \sum_{l=1}^i |\mathcal{R}_1^{(l)}|$ el nombre d'arcs complementaris fins l'article

i (on $\mathcal{R}_1^{(l)}$ és el conjunt d'arcs complementaris de l'article l), podem escriure P com:

$$P = \begin{matrix} & & (i-1)m+s & & km+r^*+r & & \\ & & \downarrow & & \downarrow & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ (i-1)m+s \rightarrow & \left(\begin{array}{cccccc} 1 & & & & & \\ & \ddots & & & & \\ & & 0 & & 1 & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \\ km+r^*+r \rightarrow & & 1 & & 0 & & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{array} \right) & & & & & & \end{matrix}$$

Sabent que $\dim(P) = (km + n + t) \times (km + n + t)$, llavors les submatrius P_3 i P_4 ocupen les $n + t$ darreres files de P ; d'aquestes, P_3 ocupa les km primeres columnes i P_4 les $n + t$ darreres (és a dir $\dim(P_3) = (n+t) \times km$, i $\dim(P_4) = (n+t) \times (n+t)$). L'aspecte d'aquestes matrius és:

$$P_3 = \begin{matrix} & & (i-1)m+s & & \\ & & \downarrow & & \\ r^*+r \rightarrow & \left(\begin{array}{cc} & \\ & 1 \end{array} \right) & ; P_4 = r^*+r \rightarrow & \left(\begin{array}{cccc} & & r^*+r & \\ & & \downarrow & \\ 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & 0 \\ & & & & 1 \\ & & & & & \ddots \\ & & & & & & 1 \end{array} \right) \end{matrix}$$

Donat que $V_j = -L_1^{-1}R_1$, llavors pot expressar-se com una matriu diagonal per blocs de forma:

$$V_j = \begin{matrix} & \underbrace{|\mathcal{R}_1^{(1)}|} & & \underbrace{|\mathcal{R}_1^{(k)}|} & \\ m \{ & \left(\begin{array}{ccc} -B^{(1)-1}R_1^{(1)} & & \\ & \ddots & \\ & & -B^{(k)-1}R_1^{(k)} \end{array} \right) & & \\ m \{ & & & & \end{matrix}$$

Llavors, si calculem l'expressió $P_4 + P_3V_j$ (que anomenarem P_5) de l'equació (4.33) de forma matricial, tenim:

$$P_5 = P_4 + P_3V_j$$

$$\begin{aligned}
 &= P_4 +_{r^*+r} \rightarrow \left(\begin{array}{c} (i-1)m+s \\ \downarrow \\ 1 \end{array} \right) m \left\{ \begin{array}{c} \underbrace{|\mathcal{R}_1^{(1)}|} \\ -B^{(1)-1}R_1^{(1)} \\ \dots \\ -B^{(i)-1}R_1^{(k)} \end{array} \right\} \\
 &=_{r^*+r} \rightarrow \left(\begin{array}{c} r^*+r \\ \downarrow \\ 1 \\ \dots \\ 1 \\ 0 \\ 1 \\ \dots \\ 1 \end{array} \right) +_{r^*+r} \rightarrow \left(\begin{array}{c} \underbrace{|\mathcal{R}_1^{(i)}|} \\ (B^{(i)-1}R_1^{(i)})^s \end{array} \right) \\
 &=_{r^*+r} \rightarrow \left(\begin{array}{c} \underbrace{|\mathcal{R}_1^{(i)}|} \\ 1 \\ \dots \\ 1 \\ (B^{(i)-1}R_1^{(i)})^s \\ 1 \\ \dots \\ 1 \end{array} \right)
 \end{aligned}$$

On $(B^{(i)-1}R_1^{(i)})^s$ no és més que l'afectació de l'arc s als cicles dels arcs complementaris de l'article i , cosa que coincideix exactament amb el vector γ que prèviament ja hem calculat. Per tant $(B^{(i)-1}R_1^{(i)})^s = \gamma$, i la matriu $P_5 = P_4 + P_3 V_j$ és directament calculable. A més P_5 és una matriu eta fila (cosa que fa que la seva inversa sigui immediata). Llavors l'expressió (4.33) d'actualització pot ser escrita com $Y_{j+1}^{-1} = P_5 Y_j^{-1}$. Donat que només ens interessa actualitzar Q_{j+1} (la submatriu diagonal superior de Y_{j+1}) realitzem un particionament escaient de P_5 i prenem la submatriu diagonal superior (amb les dimensions correctes per tal de poder premultiplicar Q_j^{-1}) que anomenarem P_6 . Llavors tenim que $Q_{j+1}^{-1} = P_6 Q_j^{-1}$. Llavors la fórmula d'actualització de Q_{j+1} serà directament $Q_{j+1} = Q_j P_6^{-1}$, on P_6^{-1} és immediata ja que P_6 continua essent una matriu eta fila. Finalment es pot escriure, considerant la matriu estesa:

$$\bar{Q}_{j+1} = \bar{Q}_j \bar{P}_6^{-1}$$

Clarament el cost per dur a terme aquesta operació és nul, ja que el vector γ (l'única cosa necessària) ja estava calculat prèviament.

Amb aquest darrer cas l'actualització de la matriu de treball queda completada.

4.4 Implementació del particionament primal.

A les seccions precedents s'ha descrit l'algorisme del particionament primal usat per solucionar el problema **(FMLCB)** introduït al capítol 2. A continuació es presentaran de forma breu certs aspectes sobre com s'ha dut a la pràctica l'algorisme prèviament detallat. Aquests aspectes tractaran sobre les fases de l'algorisme desenvolupat, el procediment de taxació de variables, el tractament de la matriu de treball, i la breu descripció d'algunes de les estructures de dades emprades.

4.4.1 Fases de l'algorisme desenvolupat.

La implementació realitzada del mètode del particionament primal consisteix en tres fases, anomenades fase 0, fase 1, i fase 2, a diferència de les dues fases clàssiques del mètode del símplex. Les fases 0 i 1 intenten obtenir un punt inicial factible, mentre que la fase 2 busca el punt òptim. Tanmateix, tot i que les fases 0 i 1 són usades de forma seqüencial per a un mateix propòsit, cercar un punt inicial factible, cal dir que la fase 1 és més propera a la fase 2 que no pas a la fase 0. Això es deu a que el mètode del particionament primal és usat únicament a les fases 1 i 2. A continuació es descriuran cada una d'aquestes fases, amb el qual es clarificaran aquests conceptes.

4.4.1.1 Fase 0.

A la fase 0 l'algorisme considera només les constriccions de xarxa (2.2) i els límits simples de les variables (2.3), sense tenir en compte les constriccions que acoblen els fluxos de diferents articles. Llavors s'intenta obtenir per a cada article $i, i = 1, \dots, k$ un punt inicial factible pel problema de fluxos lineals

$$\begin{aligned} \text{(F0)} \quad & \min_{x^{(i)}} c^{(i)t} x^{(i)} \\ & \text{subj. a } Ax^{(i)} = b^{(i)} \\ & \underline{0} \leq x^{(i)} \leq \bar{x}^{(i)} \end{aligned}$$

La solució a aquest problema s'obté mitjançant una especialització del símplex per a problemes de fluxos en xarxes. La implementació desenvolupada segueix bàsicament la metodologia descrita a [34] pel que fa a les operacions de pivotació dins els arbres d'expansió mínima. Cal observar que a la fase 0 la tècnica del particionament primal no és usada.

Al codi desenvolupat es permet escollir entre dues possibilitats: (i) obtenir simplement un punt factible pel problema **(F0)**; o bé (ii) obtenir el seu valor òptim (l'opció per defecte és obtenir un punt factible). Quan a la solució òptima del problema **(FMLCB)** el nombre

de constriccions de capacitat mútua i a banda actives $|\mathcal{A}|$ és petit, això pot indicar que aquest punt no es troba massa lluny del punt que s'obté unint les solucions dels k problemes **(F0)**. Per tant sembla convenient, en aquest cas, obtenir el punt òptim per a cada article del problema **(F0)**, i d'aquesta forma al final de la fase 0 ens trobarem a prop de la solució òptima del nostre problema multiarticle. En cas contrari, n'hi haurà prou amb obtenir un punt factible per **(F0)**.

També durant la fase 0 es pot activar un mecanisme per evitar pivotacions degenerades, mantenint una llista de variables que provocarien degeneració. Tanmateix, s'ha observat que, en molts casos, mantenir aquesta llista és menys eficient que permetre passes degenerades, i per tant l'opció per defecte no controla la degeneració de les pivotacions.

4.4.1.2 Fase 1.

Els k punts obtinguts a la fase 0 no satisfaran, en general, les constriccions de capacitat mútua i a banda, donant lloc, doncs, a un punt que anomenarem *pseudofactible*. Això implica que algunes folgues s_{cm} , per a les constriccions de capacitat mútua, i/o s_{cb} , per a les constriccions a banda, es trobaran fora de fites. Si denotem per $\hat{x}^{(i)}, i = 1, \dots, k$ el nostre punt pseudofactible, llavors podem definir els següents conjunts d'índexs:

- $\mathcal{J}_{cm}^- = \{j : \left(\sum_{i=1}^k \hat{x}^{(i)}\right)_j > b_{cm_j} \Leftrightarrow s_{cm_j} < 0\}$.
- $\mathcal{J}_{cb}^- = \{j : \left(\sum_{i=1}^k T^{(i)} \hat{x}^{(i)}\right)_j > \bar{b}_{cb_j} \Leftrightarrow s_{cb_j} < 0\}$.
- $\mathcal{J}_{cb}^+ = \{j : \left(\sum_{i=1}^k T^{(i)} \hat{x}^{(i)}\right)_j < \underline{b}_{cb_j} \Leftrightarrow s_{cb_j} > \bar{b}_{cb_j} - \underline{b}_{cb_j}\}$.

Introduint unes noves variables artificials $f_{cm} \in \mathbb{R}^n$ i $f_{cb} \in \mathbb{R}^t$, i fixant valors inicials per s_{cm} i s_{cb} tal que:

- $\left(\sum_{i=1}^k \hat{x}^{(i)}\right)_j + s_{cm_j} - f_{cm_j} = b_{cm_j} \quad ; \quad s_{cm_j} = 0 \quad ; \quad \forall j \in \mathcal{J}_{cm}^-$
- $\left(\sum_{i=1}^k T^{(i)} \hat{x}^{(i)}\right)_j + s_{cb_j} - f_{cb_j} = \bar{b}_{cb_j} \quad ; \quad s_{cb_j} = 0 \quad ; \quad \forall j \in \mathcal{J}_{cb}^-$
- $\left(\sum_{i=1}^k T^{(i)} \hat{x}^{(i)}\right)_j + s_{cb_j} + f_{cb_j} = \bar{b}_{cb_j} \quad ; \quad s_{cb_j} = \bar{b}_{cb_j} - \underline{b}_{cb_j} \quad ; \quad \forall j \in \mathcal{J}_{cb}^+$

Llavors el problema solucionat a la fase 1 pot ser formulat com segueix:

$$\min_{x^{(1)}, \dots, x^{(k)}, s_{cm}, s_{cb}, f_{cm}, f_{cb}} \sum_{j \in \mathcal{J}_{cm}^-} f_{cm_j} + \sum_{j \in \mathcal{J}_{cb}^-} f_{cb_j} + \sum_{j \in \mathcal{J}_{cb}^+} f_{cb_j} \quad (4.34)$$

subj. a (2.2), (2.3)

(F1)

$$\sum_{i=1}^k x^{(i)} + s_{cm} + \mathbb{1}^{cm} f_{cm} = b_{cm} \quad (4.35)$$

$$\sum_{i=1}^k T^{(i)} x^{(i)} + s_{cb} + \mathbb{1}^{cb} f_{cb} = \bar{b}_{cb} \quad (4.36)$$

(2.10)

$$0 \leq f_{cm} \quad 0 \leq f_{cb} \quad (4.37)$$

On les matrius $\mathbb{1}^{cm} \in \mathbb{R}^{n \times n}$ i $\mathbb{1}^{cb} \in \mathbb{R}^{t \times t}$ són diagonals i definides com:

$$(\mathbb{1}^{cm})_{jj} = \begin{cases} -1 & \text{si } j \in \mathcal{J}_{cm}^- \\ 0 & \text{altrament} \end{cases} \quad (\mathbb{1}^{cb})_{jj} = \begin{cases} -1 & \text{si } j \in \mathcal{J}_{cb}^- \\ +1 & \text{si } j \in \mathcal{J}_{cb}^+ \\ 0 & \text{altrament} \end{cases}$$

Cal observar que la funció objectiu (4.34) a la fase 1 no és més que la suma d'infactibilitats de les constriccions de capacitat mútua i constriccions a banda. Per tant el problema **(FMLCB)** serà factible si, a la fase 1, el valor de (4.34) al punt òptim és 0.

El fet de dividir el procés de buscar un punt factible inicial en dues etapes (fase 0 i fase 1) s'ha mostrat molt eficient pel que fa al nombre d'iteracions respecte mètodes que consideren la suma d'infactibilitats per totes les constriccions. A la implementació realitzada, al principi de la fase 1 se sap que no cal tenir en compte les infactibilitats respecte les constriccions de xarxa, ja que a la fase 0 s'han obtingut k arbres d'expansió mínima, el qual no és computacionalment gaire costós.

4.4.1.3 Fase 2.

Un cop s'ha obtingut un punt factible a les fases 0 i 1, la fase 2 intentarà trobar el punt òptim sense abandonar la regió factible. El problema a solucionar ara és el problema original **(FMLCB)**. Bàsicament, la única diferència entre les fases 1 i 2 rau en la funció objectiu que es minimitza, i en la consideració de les variables artificials f_{cm} i f_{cb} , però en tots dos casos la tècnica del particionament primal és aplicada.

4.4.2 Procediment de taxació.

En el cas de fluxos lineals que estem tractant en aquest capítol, el procediment de taxació determina quina variable no bàsica entrarà a la base, substituint a alguna variable bàsica. La implementació desenvolupada realitza una taxació per blocs de les variables

no bàsiques (en comptes de taxar-les totes a cada iteració). El nombre de variables no bàsiques a taxar a cada iteració ($\#nmb$) es calcula de forma

$$\#nmb = \max \left\{ 20, \frac{k(n - m + 1)}{\#\text{nblocs}} \right\} \quad (4.38)$$

essent $\#\text{nblocs}$ el nombre de blocs que volem considerar (per defecte es considera $\#\text{nblocs}=10$). Si després de taxar el primer bloc de $\#nmb$ variables no bàsiques no hi ha cap candidata a entrar a la base, es continuarà taxant el següent bloc, i així fins trobar-ne una, anotant-nos aleshores el proper bloc per on començarem la taxació a la següent iteració. Si no se'n troba cap, de variable candidata a entrar a la base, considerarem que estem a l'òptim del nostre problema.

Naturalment, als efectes de taxació de les variable no bàsiques intervé de forma decisiva l'ordre que es consideri de les variables. A la implementació realitzada es poden considerar quatre tipus d'ordenacions diferents:

- Ordenació 1. Les folgues són taxades sempre en primer lloc. Per als arcs, es consideren k llistes circulars, una per article.
- Ordenació 2. Les folgues són taxades sempre en primer lloc. Per als arcs, es manté una única gran llista circular que considera tots els arcs conjuntament.
- Ordenació 3. Les folgues són taxades d'igual forma que els arcs. Per als arcs, es consideren k llistes circulars, una per article.
- Ordenació 3. Les folgues són taxades d'igual forma que els arcs. Per als arcs, es manté una única gran llista circular que considera tots els arcs conjuntament.

En els primers dos tipus d'ordenació (ordenacions 1 i 2) el codi sempre intentarà taxar les folgues no bàsiques en primer lloc. D'aquesta forma es pot evitar fer créixer la dimensió de la matriu de treball Q , ja que, tal i com es va detallar a §4.3.1, fer entrar una folga a la base mai incrementa el nombre de constriccions actives. Si no s'ha trobat cap folga candidata, aleshores es passarà a taxar per blocs les variables associades a fluxos de la xarxa. Tanmateix, a les ordenacions 3 i 4 les folgues són taxades d'igual forma a com es fa amb els arcs, passant a formar part d'algun dels blocs de taxació.

La diferència que hi ha entre les ordenacions 1 i 2 per una banda, i 3 i 4 per una altra, és la mateixa, i consisteix en que a les primeres (ordenacions 1 i 3) es consideren k llistes circulars, una per article (i k punters-arc, un per llista, al següent arc que serà taxat per a cada article, i un punter-article addicional al següent article que cal taxar), mentre que a les segones (ordenacions 2 i 4) només es manté una única gran llista circular que considera tots els arcs conjuntament (i un únic punter al següent arc que cal taxar). Per tant, quan es realitzi la taxació considerant les ordenacions 1 o 3, es començarà amb l'article apuntat pel punter-article, i l'arc associat al corresponent punter-arc dels k possibles. I la taxació continuarà amb aquest article per blocs de dimensió $\#nmb$ fins arribar de nou a l'arc de sortida inicial (aquell apuntat pel punter-arc de l'article en qüestió). Llavors continuarem pel següent article, i així successivament. Al segon tipus d'ordenacions (2 i 4), però, si comencem amb un cert arc d'un determinat article (sigui l'arc j de l'article i), passarem a

taxar arcs de l'article $i+1$ quan arribem al darrer arc de l'article actual, sense recomençar per l'arc 1 de l'article i i esperar a donar una volta sencera fins arribar a l'arc j inicial.

A la majoria de tests realitzats, els tipus 1 i 2 s'han mostrat més eficients que les ordenacions 3 i 4. Per la seva banda, no s'ha apreciat una diferència substancial entre el rendiment del codi entre els tipus 1 i 2 (i 3 i 4). Tanmateix, a la implementació realitzada es pot escollir entre un dels 4 tipus, tant a nivell de fase 1 com de fase 2.

4.4.3 Tractament de la matriu de treball.

A §4.3.2.3 es va detallar el procediment per actualitzar una matriu estesa qualsevol, i posteriorment es va particularitzar pel cas de la matriu Q a §4.3.3. A l'equació (4.27) es van presentar les passes que calien fer per solucionar un sistema qualsevol amb l'actualització de la matriu de treball. En una d'aquestes passes, però, era necessari disposar d'una forma de solucionar sistemes d'equacions amb una factorització de la matriu Q .

La implementació concreta que s'ha fet realitza una descomposició LU esparsa, a base de matrius eta i amb pivotació parcial, d'una reordenació prèvia de la matriu Q , per tal d'evitar la creació de nous elements que poguessin degradar l'esparsitat original de la matriu. És a dir, s'han obtingut unes matrius R_f i R_c de reordenació de files i columnes, i unes matrius $E_j, P_j, j = 1, \dots, |\mathcal{A}|$ (on E_j són matrius eta columna triangulars inferior, P_j són matrius de permutació de dues files, i $|\mathcal{A}|$ ens dóna la dimensió de Q) de triangularització i pivotació parcial, de forma que:

$$E_{|\mathcal{A}|}P_{|\mathcal{A}|} \dots E_2P_2E_1P_1R_fQR_c = U \quad (4.39)$$

essent U una matriu triangular superior. La solució de sistemes del tipus $Qx = b$ seria ara de la següent forma:

$$\begin{aligned} Qx &= b \\ \left(\prod_{j=|\mathcal{A}|}^1 E_jP_j \right)^\dagger R_f Qx &= \tilde{b} \quad \text{on } \tilde{b} = \left(\prod_{j=|\mathcal{A}|}^1 E_jP_j \right) R_f b \\ \left(\prod_{j=|\mathcal{A}|}^1 E_jP_j \right) R_f QR_c y &= \tilde{b} \quad \text{on } x = R_c y \\ Uy &= \tilde{b} \quad \text{aplicant (4.39)} \end{aligned} \quad (4.40)$$

I finalment, un cop trobat y , $x = R_c y$

Aplicant una estratègia similar es resoldrien els sistemes $x^t Q = b^t$.

S'han emprat dos mètodes per obtenir matrius de reordenació R_f i R_c per evitar la degradació de l'esparsitat original de Q : l'algorisme P3 de Hellerman i Rarick, i una

[†]S'ha d'entendre que en aquest cas el productori itera de $j = |\mathcal{A}|$ fins a 1 decrementant el valor de j

variant del mateix per tal de deixar les columnes amb elements sobrediagonals (punxes) al final de la matriu reordenada $R_f Q R_c$ [26,36]. Les dues tècniques han donat resultats similars i satisfactoris. Cal afegir que les rutines desenvolupades per fer la descomposició LU amb matrius etes han estat comparades amb la rutina estàndard F01BRF de la llibreria NAG (derivació de la rutina MA28 [26] de la llibreria Harwell) havent-se obtinguts temps d'execució similars (lleugerament millors en el cas d'usar la factorització amb etes).

4.4.4 Estructures de dades.

En aquest apartat es comentarà molt breument com són les estructures de dades usades per emmagatzemar part de la informació necessària. No es pretén, però, exposar totes les estructures usades a la implementació feta (seria inviable), ni fer una descripció exhaustiva d'aquelles que a continuació detallarem.

- 1.- *Estructura de la xarxa.* La matriu de xarxa A s'emmagatzema usant només dos vectors $\mathbf{mkn}(1:n)$ i $\mathbf{mnl}(1:n)$, (essent n el nombre d'arcs de la xarxa), que ens donen per a cada arc el seu node origen, i el seu node destí (és a dir $\mathbf{mkn}(j)=v_j$ i $\mathbf{mnl}(j)=w_j$, essent v_j el node origen de l'arc j , i w_j el node destí del mateix arc).
- 2.- *Arbres bàsics.* Seguint algunes de les idees exposades a [9,34], tot arbre bàsic s'emmagatzema en cinc vectors anomenats: $\mathbf{itv}(1:m+1)$, $\mathbf{iti}(1:m+1)$, $\mathbf{ipf}(1:m)$, $\mathbf{ipr}(1:m)$, i $\mathbf{iab}(1:m)$, essent m el nombre de nodes de l'arbre. Cal dir que alguns d'aquests vectors no caldrien ser emmagatzemats i la informació que contenen podria ser obtinguda a partir de la resta de vectors. Tanmateix, el fet de mantenir-los es fa per motius d'eficiència computacional, en detriment de l'espai a memòria que ocupen. Passem tot seguit a fer una breu descripció d'aquests vectors.
 - $\mathbf{itv}(1:m+1)$: Aquest vector (anomenat vector travesser) emmagatzema en forma de llista encadenada l'ordenació en preordre dels nodes de l'arbre. Llavors $\mathbf{itv}(i)=j$ indica que, en el recorregut en preordre de l'arbre, després del node i ve el node j . Per saber quin és el node que segueix al j , consultariem la posició $\mathbf{itv}(j)$.
 - $\mathbf{iti}(1:m+1)$: Aquest vector (anomenat vector travesser invers) emmagatzema en forma de llista encadenada l'ordenació inversa de l'emmagatzemada al vector \mathbf{itv} . És a dir, es verifica que $\forall i, \mathbf{iti}(\mathbf{itv}(i))=i$.
 - $\mathbf{ipf}(1:m)$: Aquest vector (vector de profunditat) ens indica la profunditat o nivell en que es troba cada node de l'arbre. S'usa per determinar el node de juntura quan s'ascendeix per dos branques diferents de l'arbre.
 - $\mathbf{ipr}(1:m)$: Aquest vector (vector de predecessors) ens indica per a cada node i quin node j té per sobre dins l'arbre. A més, s'usa un criteri de signes per denotar el sentit de l'arc que uneix i i j . Llavors si $\mathbf{ipr}(i)=+j$, el sentit de l'arc serà $j \rightarrow i$. Si $\mathbf{ipr}(i)=-j$, aleshores el sentit serà el contrari $j \leftarrow i$.
 - $\mathbf{iab}(1:m)$: Aquest vector (vector d'índexs d'arcs de l'arbre) ens complementa la informació del vector \mathbf{ipr} . Així com aquell ens donava el node j que hi ha per sobre del node i dins l'arbre, aquest ens indica per tot node i quin és l'arc de

la xarxa que l'uneix amb el seu predecessor j . Aquesta informació podria ser obtinguda a partir del vector `ipr`, però caldria fer una cerca pels vectors `mnk` i `mn1` que ens defineixen la xarxa, el qual seria molt ineficient. El fet de mantenir el vector `iab` és, doncs, crucial pel que fa al rendiment del programa.

- 3.- *Arcs complementaris.* La informació que cal guardar sobre els arcs complementaris és doble: cal saber quins són, i, a més, tenir emmagatzemats els seus cicles dins l'arbre bàsic associat (això per agilitzar el procés de càlcul on apareguin sistemes del tipus $B^{(i)-1}R_1^{(i)}$). Per saber quins són els arcs complementaris a cada iteració, disposarem de dos vectors `jple(1:|A|)` i `kple(1:|A|)`, de forma que `jple(l)` ens donarà l'arc $j, 1 \leq j \leq n$ i `kple(l)` l'article $i, 1 \leq i \leq k$ que corresponen al l -èssim arc complementari. Per la seva banda, per emmagatzemar els cicles disposarem d'un vector d'una determinada dimensió M (més aviat gran) que podríem anomenar `cicles(M)`, i un punter a la primera posició lliure d'aquest vector, que denotarem per `freecycle`. Per cada arc complementari tindrem dos vectors addicionals `inicycle(1:|A|)` i `longcycle(1:|A|)`, de forma que els arcs del cicle del vector complementari l -èssim es trobaran entre les posicions `inicycle(l)` i `inicycle(l)+longcycle(l)` dins el vector `cicles`.

Amb aquesta estructura de dades cada nou cicle que s'hagi de calcular per un determinat arc complementari s'afegirà dins el vector `cicles` a partir de la posició `freecycle`, i s'apuntarà l'inici del cicle i la seva longitud dins els vectors `inicycle` i `longcycle`. Posteriorment s'actualitzarà el punter `freecycle` a la nova posició lliure. Si un arc complementari és eliminat, només caldrà netejar la component corresponent de `inicycle` i `longcycle`, però no es farà cap modificació dins el vector `cicles`, és a dir, no es recuperen les posicions que quedaran lliures. Això es fa per evitar fer compactacions dins aquest vector, el qual alentiria molt el procés. Amb aquesta estructura també és molt simple intercanviar arcs complementaris: només cal intercanviar les components corresponents dels vectors `jple`, `kple`, `inicycle` i `longcycle`. Cal observar que si no disposéssim del vector `longcycle` (determinant aleshores el final del cicle d'un arc complementari a partir de l'inici del cicle de l'arc següent) no es podrien efectuar les operacions d'intercanvi i eliminació d'arcs complementaris tan fàcilment, donat que estariem obligats a mantenir la contigüïtat dels arcs complementaris per saber l'inici i final de cicles.

Quan en afegir un nou cicle es supera la dimensió M del vector `cicles`, aleshores cal fer una compactació del mateix. Com ja s'ha esmentat anteriorment, aquesta compactació es podria haver fet cada cop que s'eliminés un arc complementari (i aleshores no superaríem la dimensió M). Tanmateix, és molt més eficient fer una compactació de tot el vector cada moltes iteracions (de fet s'ha observat que al llarg d'una execució el nombre de vegades que es supera la dimensió M és petita), que fer compactacions no tan grans a gairebé cada iteració.

- 4.- *Folgues i constriccions actives.* La informació que cal conèixer de les folgues és saber quines són variables bàsiques i quines no bàsiques. Alhora aquesta dada ens serveix per determinar quines constriccions de capacitat mútua i a banda són actives i quines

no: les folgues bàsiques es troben associades a constriccions inactives ($\notin \mathcal{A}$), i les no bàsiques a constriccions actives ($\in \mathcal{A}$). Per emmagatzemar la informació de folgues disposarem de dos vectors anomenats **folguescm**(1: n) i **folguescb**(1: t). El vector **folguescm** es troba associat a les folgues s_{cm} de les constriccions de capacitat mútua, mentre que **folguescb** manté la informació de les folgues s_{cb} de les constriccions a banda. Si **folguescm**(j) = 0, aleshores la folga s_{cm_j} serà bàsica, mentre que si **folguescm**(j) = l ($1 \leq l \leq |\mathcal{A}|$) serà no bàsica, i la corresponent constricció de capacitat mútua serà la l -èssima constricció activa (o el que és el mateix, determinarà la l -èssima fila de la matriu de treball Q). Anàlogament, tenim que si **folguescb**(j) = 0, llavors la folga s_{cb_j} serà bàsica, mentre que si **folguescb**(j) = l ($1 \leq l \leq |\mathcal{A}|$) s_{cb_j} serà no bàsica, i la corresponent constricció a banda serà la l -èssima constricció activa.

Queda clar que amb la informació anterior poden recuperar-se quines són les constriccions actives del conjunt \mathcal{A} . Tanmateix, per conèixer-les hauríem de recórrer els vectors **folguescm** i **folguescb**. Per evitar això, es manté un altre vector **actives**(1: $|\mathcal{A}|$) de forma que si **actives**(l) = $+j$ (j positiu) llavors la l -èssima constricció activa correspon a la constricció de capacitat mútua de l'arc j , mentre que si **actives**(l) = $-j$ la j -èssima constricció a banda serà la constricció activa número l . Es pot observar com, en el cas de que **folguescm**(j) = $l \neq 0$, es verifica que **actives**(**folguescm**(j)) = j ; per la seva banda, si **folguescb**(j) = $l \neq 0$, sempre tindrem que **actives**(**folguescb**(j)) = $-j$.

Capítol 5

Fluxos no lineals multiarticle per particionament primal.

Al capítol precedent s'ha presentat el mètode de resolució usant el particionament primal per a problemes lineals. En aquest capítol es descriurà la metodologia emprada en el cas de tractar problemes de fluxos multiarticle no lineals. Abans, però, d'entrar en l'especialització del particionament primal per a problemes no lineals i els detalls de l'implementació, es farà una breu descripció de l'algorisme general de programació no lineal emprat.

5.1 Optimització no lineal a gran escala amb restriccions lineals.

El mètode de programació no lineal general usat per solucionar problemes de fluxos no lineals ha estat desenvolupat per Murtagh i Saunders, i és descrit a [58]. La metodologia seguida forma part d'una família de mètodes anomenats del conjunt de restriccions actives [30]. De forma molt breu, presentarem les principals característiques del mètode.

5.1.1 Conceptes previs.

Considerarem el problema general de programació lineal següent:

$$\begin{array}{llll} \min & f(x) & (5.1) & f: \mathbb{R}^{\tilde{n}} \rightarrow \mathbb{R}, f \in C^2 \\ \text{(NCL)} \quad \text{subj. a :} & \tilde{A}x = b & (5.2) & ; \quad \tilde{A} \in \mathbb{R}^{\tilde{m} \times \tilde{n}}, \text{rang}(\tilde{A}) = \tilde{m} \\ & 0 \leq x \leq u & (5.3) & x, u \in \mathbb{R}^{\tilde{n}}, b \in \mathbb{R}^{\tilde{m}} \end{array}$$

El vector gradient de $f(x)$ es denotarà per $g(x) \in \mathbb{R}^{\tilde{n}}$, mentre que la matriu hessiana serà referida com $H(x) \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$. En aquesta secció, la funció $f()$ farà referència a una funció qualsevol (al capítol 2 es referia a una funció on les variables eren els fluxos multiarticle de la xarxa). D'igual forma, \tilde{A} farà referència a una matriu de restriccions qualsevol, de dimensió $\tilde{m} \times \tilde{n}$ (a diferència de la resta del document on representa la matriu de restriccions del problema multiarticle).

El conjunt de variables x es divideix en variables bàsiques $x_B \in \mathbb{R}^{\tilde{m}}$, superbàsiques $x_S \in \mathbb{R}^s$, i no bàsiques $x_N \in \mathbb{R}^{\tilde{n}-\tilde{m}-s}$. Anàlogament, la matriu de constriccions \tilde{A} es divideix de forma:

$$\tilde{A} = \begin{array}{|c|c|c|} \hline & \tilde{m} & s & \tilde{n} - \tilde{m} - s \\ \hline & B & S & N \\ \hline \end{array} \quad (5.4)$$

Les variables bàsiques x_B associades a B verifiquen $0 \leq x_{B_i} \leq u_i$, i determinen una base de l'espai de dimensió \tilde{m} (és a dir, B és no singular). Les variables no bàsiques x_N (associades a N) es troben sempre a fita inferior o superior. La resta de variables que no es troben dins la base B i que verifiquen $0 < x_i < u_i$ seran les variables superbàsiques.

Al nostre problema, donat un punt factible $x^t = [x_B^t | x_S^t | x_N^t]$ la matriu de constriccions actives al punt x es defineix com:

$$\hat{A} = \begin{array}{|c|c|c|} \hline & \tilde{m} & s & \tilde{n} - \tilde{m} - s \\ \hline & B & S & N \\ \hline & \mathbf{0} & & \mathbf{1} \\ \hline \end{array} \begin{array}{l} \tilde{m} \\ \tilde{n} - \tilde{m} - s \end{array} \quad (5.5)$$

L'expressió de les constriccions actives en aquest punt és:

$$\hat{A}x = \hat{b} \quad ; \quad \hat{b} = \begin{bmatrix} b \\ r_N \end{bmatrix} \quad (5.6)$$

essent r_N el vector de fites actives (inferiors, 0, o superiors, u_{N_i}) de les variables no bàsiques x_N .

5.1.2 Optimització al subespai actual.

Si coneguéssim el conjunt de variables x_N que són no bàsiques al punt òptim, podríem determinar la matriu de constriccions actives òptima. Aquesta matriu, però, te s graus de llibertat, associats a les variables superbàsiques, i caldria fer una optimització amb la matriu de constriccions òptima sobre el subespai de les variables superbàsiques. Si ens

trobem al punt factible $x^t = [x_B^t | x_S^t | x_N^t]$ (per tant $\hat{A}x = \hat{b}$) haurem de trobar aleshores direccions $p \in \mathbb{R}^{\tilde{n}}$, $p^t = [p_B^t | p_S^t | p_N^t]$ factibles i de descens, de forma que

$$\hat{A}(x+p) = \hat{b} \Rightarrow \hat{A} \begin{bmatrix} p_B \\ p_S \\ p_N \end{bmatrix} = 0 \Rightarrow \begin{cases} Bp_B + Sp_S + Np_N = 0 \\ p_N = 0 \end{cases} \Rightarrow \begin{cases} p_S \in \mathbb{R}^s \text{ lliure} \\ p_B = -B^{-1}Sp_S \\ p_N = 0 \end{cases}$$

Això equival a dir que totes les direccions factibles p seran de la forma $p = Zp_S$, on $Z \in \mathbb{R}^{\tilde{n} \times s}$ expandeix el subespai nul de la matriu de constriccions actives (5.5), i es defineix com:

$$Z = \begin{matrix} & \tilde{m} \{ & \overbrace{\hspace{1.5cm}}^s & \\ & -B^{-1}S & & \\ & \hline & \mathbf{1} & & \\ & \hline \tilde{n} - \tilde{m} - s \{ & \mathbf{0} & & \end{matrix} \quad (5.7)$$

Pot comprovar-se fàcilment que $\hat{A}Z = \mathbf{0}$.

Un cop assegurada la factibilitat, només resta obtenir direccions de descens. Ara el nostre problema d'optimització al subespai de les variables superbàsiques pot ser formulat com

$$(\mathbf{PZ}\hat{A}) \quad \min_{p_S \in \mathbb{R}^s} f(x + Zp_S)$$

Prenent una aproximació quadràtica de $f(x + Zp_S)$, i imposant que la seva primera derivada s'anulli, s'arriba a que es poden obtenir direccions de descens de $(\mathbf{PZ}\hat{A})$ resolent el sistema

$$H_z(x)p_S = -g_z(x) \quad (5.8)$$

on $g_z = Z^t g(x)$ ($g_z \in \mathbb{R}^s$) i $H_z = Z^t H(x) Z^t$ ($H_z \in \mathbb{R}^{s \times s}$) són respectivament el gradient reduït (o gradient projectat) i l'hessiana reduïda (o hessiana projectada) de f al punt x .

Un cop calculada la direcció de descens p_S s'ha de fer exploració lineal al llarg de la direcció $p = Zp_S$, amb longitud de pas limitada pel valor $\bar{\alpha} = \min\{\bar{\alpha}_B, \bar{\alpha}_S\}$, imposat per les fites de les variables bàsiques i superbàsiques:

$$\bar{\alpha}_B = \min_{1 \leq i \leq \tilde{m}} \{ \{-x_{B_i}/p_{B_i} : p_{B_i} < 0\}, \{(u_{B_i} - x_{B_i})/p_{B_i} : p_{B_i} > 0\} \} \quad (5.9)$$

$$\bar{\alpha}_S = \min_{1 \leq i \leq s} \{ \{-x_{S_i}/p_{S_i} : p_{S_i} < 0\}, \{(u_{S_i} - x_{S_i})/p_{S_i} : p_{S_i} > 0\} \} \quad (5.10)$$

Si el resultat de l'exploració lineal és $\alpha^* = \bar{\alpha}$ aleshores alguna variable superbàsica o bàsica arriba a la seva fita inferior o superior, i passarà a ser variable no bàsica. Aleshores caldrà modificar la nostra partició de variables, considerar una nova matriu de constriccions actives i una nova definició del problema $(\mathbf{PZ}\hat{A})$.

5.1.3 Determinació de la matriu de restriccions actives òptima.

Un cop solucionat el problema $(\mathbf{PZ}\hat{A})$ s'haurà determinat un punt $x_{\hat{A}}^*$ tal que $g_z(x_{\hat{A}}^*) = Z^t g(x_{\hat{A}}^*) = 0$ (és a dir, $x_{\hat{A}}^*$ és un punt estacionari). Si $H_z(x_{\hat{A}}^*)$ és definida positiva $x_{\hat{A}}^*$ serà un mínim local de $(\mathbf{PZ}\hat{A})$. A partir d'aquest punt, però, no poden obtenir-se noves direccions factibles i de descens de (\mathbf{NCL}) , a no ser que es modifiqui la matriu de restriccions actives (5.5). És a dir, caldrà determinar una nova partició de variables, modificar la matriu de restriccions actives de forma escaient, i solucionar de nou el problema $(\mathbf{PZ}\hat{A})$ amb la nova matriu \hat{A} . La forma de modificar la matriu \hat{A} , però, no pot ser qualsevol, i es farà atenent a les condicions d'optimalitat del problema (\mathbf{NCL}) .

Donada una solució factible x de (\mathbf{NCL}) , les condicions necessàries de primer ordre de (\mathbf{NCL}) (o condicions de Karush-Kuhn-Tucker), estableixen que si x és un mínim local aleshores existeixen $\pi \in \mathbb{R}^{\tilde{m}}$ i $\lambda \in \mathbb{R}^{\tilde{n}-\tilde{m}-s}$, anomenats multiplicadors de Lagrange, tals que:

$$i) \quad \hat{A}^t \begin{bmatrix} \pi \\ \lambda \end{bmatrix} = g(x) \quad \Leftrightarrow \quad \begin{bmatrix} B^t & \mathbf{0} \\ S^t & \mathbf{0} \\ N^t & \mathbf{1} \end{bmatrix} \begin{bmatrix} \pi \\ \lambda \end{bmatrix} = \begin{bmatrix} g_B \\ g_S \\ g_N \end{bmatrix} \quad (5.11)$$

$$ii) \quad \lambda_i \geq 0, \quad \forall i : x_{N_i} = 0 \quad (5.12)$$

$$\lambda_i \leq 0, \quad \forall i : x_{N_i} = u_{N_i} \quad (5.13)$$

on g_B , g_S i g_N determinen una partició del vector gradient segons les seves components es trobin associades a variables bàsiques, superbàsiques o no bàsiques. A partir de (5.11) s'obté:

$$B^t \pi = g_B \quad ; \quad \pi^t = g_B^t B^{-1} \quad (5.14)$$

$$N^t \pi + \lambda = g_N \quad ; \quad \lambda^t = g_N^t - \pi^t N \quad (5.15)$$

$$S^t \pi = g_S \quad ; \quad g_S - S^t \pi = 0 \quad (5.16)$$

Si x és un punt estacionari ($g_z(x) = 0$) aleshores es verifica que

$$Z^t g(x) = 0 \quad \Leftrightarrow \quad \left[-S^t B^{t-1} \mid \mathbf{1} \mid \mathbf{0} \right] \begin{bmatrix} g_B \\ g_S \\ g_N \end{bmatrix} \Rightarrow -S^t B^{t-1} g_B + g_S = 0$$

amb el qual queda satisfeta la condició (5.16). Per tant, si ens trobem al punt estacionari $x_{\hat{A}}^*$ solució de $(\mathbf{PZ}\hat{A})$, els multiplicadors π i λ poden ser determinats a partir de (5.14) i (5.15), i si, a més, ja verifiquen les condicions (5.12) i (5.13), $x_{\hat{A}}^*$ serà solució del problema (\mathbf{NCL}) . Si, per contra, es viola alguna de les condicions (5.12) i/o (5.13), aleshores es considerarà l'índex q d'alguna de les component λ_q on s'ha produït la violació, i la variable no bàsica associada x_{N_q} serà considerada variable superbàsica, eliminant la columna N_q de N , i afegint-la a S . En aquest cas el nombre de variables superbàsiques s'incrementaria en

un ($s:=s+1$), es definiria una nova matriu de constriccions actives (5.5), i de nou passariem a solucionar $(\mathbf{PZ}\hat{A})$ amb la nova \hat{A} . Cal observar que, en augmentar el nombre de variables superbàsiques, la nova matriu Z definida a (5.7) incorporarà una nova columna associada a la nova variable superbàsica, i el mateix succeirà amb el nou gradient reduït $g_z = Z^t g(x)$. Pot comprovar-se que la component de g_z associada a la nova variable superbàsica és directament el multiplicador λ_q prèviament calculat (només cal veure que $\lambda_q = g_{N_q} - N_q^t \pi = g_{N_q} - N_q^t B^{t-1} g_B$, i que $g_z = Z^t g(x) = g_S - S^t B^{t-1} g_B$).

5.1.4 Algorisme del conjunt de constriccions actives.

Un cop s'han presentat els principals trets del mètode usat, es presenta de forma algorísmica les etapes que cal seguir. Considerarem que a l'inici de la iteració i (el subíndex i serà omès a la majoria de casos per simplificar la notació) disposem de la següent informació:

- $x_i, f(x_i)$: el punt actual i el valor de funció objectiu en aquest punt.
- B, S, N : una partició de la matriu \tilde{A} tal i com es detalla a (5.4).
- $g(x_i)$: on $g(x_i) = \nabla f(x_i)$ dividida en $g(x_i) = [g_B | g_S | g_N]$ per a les variables bàsiques, superbàsiques i no bàsiques.
- Z : una representació del subespai nul de \hat{A} , tal i com es presenta a (5.7).
- g_z, ϵ_{g_z} : el gradient reduït i una tolerància per estimar quan la seva norma és considerada el suficientment petita.
- H_z : l'hessiana reduïda.
- π : vector de multiplicadors que verifiquen l'equació (5.14).

Aleshores l'algorisme del conjunt de constriccions actives pot ser detallat tal i com segueix:

Algorisme A5.1. Mètode del conjunt de constriccions actives.

PAS (1): Test d'optimalitat al subespai actual.

- i) Si $\|g_z\| \geq \epsilon_{g_z}$ anar al PAS (3).

PAS (2): Taxació de variables no bàsiques.

- i) Càlcul dels multiplicadors de Lagrange $\lambda = g_N - N^t \pi$.
- ii) Escollir una λ_q que violi (5.12) o (5.13), considerant la seva columna associada N_q . Si no es pot escollir cap multiplicador anar al PAS (8).
- iii) Actualitzar estructures de dades: eliminar N_q de N i afegir-la a S ; afegir λ_q com la nova component de g_z .

PAS (3): Buscar una direcció de descens $p = [p_B^t | p_S^t | \mathbf{0}]^t$ per a les variables bàsiques i superbàsiques.

- i) Solucionar $H_z p_S = -g_z$.
- ii) Calcular $B p_B = -S p_S$.

PAS (4): Buscar la longitud de pas màxima.

- i) Buscar $\bar{\alpha} \geq 0$ tal que $x_i + \bar{\alpha}p$ és factible
 - ii) si $\bar{\alpha} = 0$ anar al PAS (7).
- PAS (5): Exploració lineal.
- i) Buscar α^* tal que $\alpha^* = \arg \min_{0 \leq \alpha \leq \bar{\alpha}} f(x_i + \alpha p)$.
 - ii) Actualitzem el nou punt $x_{i+1} = x_i + \alpha^*p$ i calculem $f(x_{i+1})$ i $g(x_{i+1})$.
- PAS (6): Calcular el gradient reduït.
- i) Solucionar $\pi^t B = g_B^t$.
 - ii) Fer $g_z = g_S - S^t \pi$.
 - iii) Si $\alpha^* < \bar{\alpha}$ anar al PAS (1).
- PAS (7): Una variable bàsica o superbàsica esdevé no bàsica. ($\alpha^* = \bar{\alpha}$).
- i) Si una variable superbàsica x_{S_p} (associada a la columna S_p de S) assoleix la seva fita aleshores:
 - a) Eliminar la component de g_z associada amb la columna S_p de S .
 - b) Eliminar S_p de S i afegir-la a N .
 - ii) Si una variable bàsica x_{B_p} (associada a la columna B_p de B) assoleix la seva fita aleshores:
 - a) Buscar una columna superbàsica S_q de S que substituirà B_p dins B preservant la invertibilitat de la base.
 - b) Pivotació: eliminar B_p de B i afegir-la a N ; eliminar S_q de S i afegir-la a B .
 - c) Actualitzar π .
 - d) Calcular $g_z = g_S - S^t \pi$.
 - iii) Anar al PAS (1).
- PAS (8): Fi de l'algorisme: Solució òptima trobada.

5.2 Particionament primal per a problemes no lineals.

El mètode del conjunt de restriccions actives presentat a la secció anterior ha estat especialitzat per tal de solucionar el problema de fluxos multiarticle no lineals amb restriccions a banda (**FMNCB**). Això s'ha fet combinant la tècnica del particionament primal introduïda al capítol 4 amb l'algorisme del conjunt de restriccions actives per a problemes no lineals. D'aquesta forma, igual com succeïa al cas lineal, és possible explotar l'estructura de la matriu de restriccions, agilitzant el càlcul de certes passes de l'algorisme A5.1. Concretament, aquestes passes són les següents:

- 1.- Càlcul dels multiplicadors π (Pas (6) i)). El càlcul a realitzar era $\pi^t B = g_B^t$. Donat que B és una base de la matriu de restriccions \tilde{A} del problema (**FMNCB**), es va veure al capítol 4 que pot ser descomposada tal i com es va presentar a (4.4). Aleshores,

realitzant unes particions escaients $\pi^t = [\pi_1^t \mid \pi_2^t \mid \pi_3^t]$ i $g_B = [g_{B_1}^t \mid g_{B_2}^t \mid g_{B_3}^t]$, el sistema $\pi^t B = g_B^t$ pot ser resolt mitjançant el procés presentat a §4.1.2.2. En aquest cas, doncs, també pot ser usada la matriu de treball Q , la qual agilitza els càlculs.

- 2.- *Càlcul de la direcció de descens bàsica p_B* (Pas (3) ii)). El càlcul consistia en solucionar $Bp_B = -Sp_S$. Tenint en compte l'estructura de la base B , els vectors p_B i $\gamma = -Sp_S$ poden particionar-se de forma $p_B^t = [p_{B_1}^t \mid p_{B_2}^t \mid p_{B_3}^t]$ i $\gamma^t = [\gamma_1^t \mid \gamma_2^t \mid \gamma_3^t]$, i aleshores podem aplicar el procés de resolució descrit a §4.1.2.1. Novament la matriu Q pot ser usada per solucionar aquest sistema de forma eficient.

En aquest cas, a més, també podem explotar l'estructura del problema multiarticle a l'hora de calcular el vector $\gamma = -Sp_S = \sum_{i=1}^s -S_i p_{S_i}$ (essent S_i i p_{S_i} la columna de S i la component de p_S associades amb la i -èssima variable superbàsica). La columna S_i pot ser una arc, tenint aleshores l'estructura presentada a (4.1), o una folga, amb el qual no és més que una columna de la matriu identitat. Aquest fet pot ser tingut en compte a l'hora de calcular eficientment $\gamma = \sum_{i=1}^s -S_i p_{S_i}$.

- 3.- *Càlcul del gradient reduït g_z* (Passos (6) ii), i (7) ii) d)). Donat que $g_z = Z^t g(x)$, si es té en compte l'expressió de Z a (5.7), i es particiona el gradient de forma $g^t(x) = [g_B \mid g_S \mid g_N]$, s'obté que $g_z = g_S - S^t B^{t-1} g_B$. Donat que $B^t \pi = g_B$, l'expressió anterior pot ser escrita com $g_z = g_S - S^t \pi$, on cada component serà calculada de forma $g_{z_i} = g_{S_i} - S_i^t \pi$. En aquest cas, per realitzar els productes escalars $S_i^t \pi$ de nou pot ser explotada l'estructura de les columnes de S (arcs o folgues), tal i com s'ha esmentat al punt anterior.

- 4.- *Càlcul dels multiplicadors λ* (Pas (2) i)). El càlcul a fer és $\lambda = g_N - N^t \pi$, i és equivalent al presentat anteriorment per trobar g_z . Per tant, procedint de forma anàloga a com vam operar per trobar g_z , podem escriure que $\lambda_i = g_{N_i} - N_i^t \pi$, i, de nou, a l'hora d'obtenir els productes escalars $N_i^t \pi$ aprofitarem l'estructura de les columnes no bàsiques (arcs o folgues).

- 5.- *Determinació de la columna superbàsica que substituirà una columna bàsica* (Pas (7) ii) a)). Considerem que deixa la base la variable bàsica x_{B_p} associada a la columna B_p de B . L'operació de buscar una variable superbàsica x_{S_q} que la pugui substituir consisteix en determinar una q tal que $y_{qp} \neq 0$, on $By_q = S_q$. D'aquesta forma, però, s'haurien de solucionar s sistemes d'equacions del tipus $By_i = S_i, i = 1, \dots, s$. Donat que només ens interessa la component p de cada un dels vectors y_i , aleshores el càlcul pot ser fet de forma més eficient calculant la p -èssima fila de la inversa de B (B^{p-1}), i posteriorment trobant $\gamma^t = S^t B^{p-1}$, on ara $\gamma_i = y_{ip}, i = 1, \dots, s$. Ara només cal escollir una posició q tal que $\gamma_q \neq 0$, i la variable superbàsica associada serà la candidata a entrar a la base.

Al procediment anterior, però, cal trobar la fila p de la inversa de B , B^{p-1} . Això passa per solucionar $\omega^t B = e_p^t$, on e_p és el vector p -èssim de la matriu identitat, i la solució ω ens proporcionarà la desitjada fila B^{p-1} . Aquest càlcul, però, és equivalent

al dels multiplicadors π , i, per tant, pot ser aplicat el procediment descrit a §4.1.2.2, emprant de nou la matriu de treball Q .

- 6.- *Pivotació* (Pas (7) ii) b)). Tret del fet de que en aquest cas es pivota una variable superbàsica (i no una variable no bàsica) amb una bàsica, la pivotació és exactament igual al cas de tenir una funció objectiu lineal. Per tant, els diferents tipus de pivotació tornen a ser els descrits a §4.3.1, i de nou caldrà fer una actualització de la matriu de treball Q tal i com es va descriure al capítol 4.

Queda clar, doncs, que la tècnica del particionament primal pot ser introduïda dins l'algorisme A5.1, treient profit de l'ús de la matriu Q a l'hora de solucionar sistemes amb la matriu bàsica B . Cal fer notar, però, que, a diferència del cas lineal, on bàsicament les úniques operacions costoses que calien realitzar involucraven la resolució de sistemes amb B i B^t , al cas no lineal apareixen d'altres càlculs addicionals a fer (per ex., avaluacions de la funció objectiu i del gradient, exploració lineal, càlcul de la direcció de descens superbàsica usant l'hessiana reduïda, ...). Això provoca que el guany que s'obté pel fet de treballar amb la matriu Q sigui menor al cas no lineal que al lineal, precisament perquè el nombre d'etapes on es treu profit del particionament de la base representa un tant per cent menor sobre el total de la tasca a fer.

5.3 Implementació de l'algorisme.

A les seccions precedents s'ha presentat l'algorisme de resolució de problemes no lineals emprat, i s'ha vist que pot ser usat conjuntament amb el mètode del particionament primal per tal d'agilitzar el procés de càlcul. En aquest apartat descriurem breument certs aspectes sobre com s'ha dut a la pràctica el mètode anterior. Concretament, tractarem les fases de l'algorisme desenvolupat, detallarem algunes etapes de l'algorisme A5.1 (càlcul de la direcció de descens, test d'optimalitat al subespai actual, elecció de la variable no bàsica, ...) i farem una molt breu descripció d'algunes de les estructures de dades usades.

5.3.1 Fases de l'algorisme desenvolupat.

De forma anàloga a com es va descriure a §4.4.1 per al cas de fluxos lineals, l'algorisme per a fluxos no lineals consisteix en tres fases. Les dues primeres, fase 0 i fase 1, solucionen els mateixos problemes **(F0)** i **(F1)** que al cas lineal. Això és així ja que aquestes fases cerquen un punt inicial factible, fet que és independent del tipus de funció objectiu de que es disposa. Cal fer notar que al problema **(F0)**, per al cas lineal, el costos $c^{(i)}$ de la funció objectiu eren els mateixos que els costos originals del problema **(FMLCB)**. Al cas no lineal, però, per la naturalesa de la funció objectiu, aquests costos no existeixen. Aleshores pot usar-se una linealització de la funció objectiu en algun punt (gradient) que ens porti certa informació sobre el model en solucionar **(F0)**, o simplement considerar uns $c^{(i)}$ nuls (el qual, per als nostres propòsits de trobar un punt factible, és suficient).

Un cop les dues primeres fases han trobat un punt inicial factible, la fase 2 intentarà trobar el punt òptim sense abandonar la regió factible, solucionant el problema (FMNCB). A diferència del cas lineal (on s'usava el mètode del particionament primal com a especialització del símplex), ara s'usarà l'algorisme del conjunt de constriccions actives conjuntament amb el mètode del particionament primal.

5.3.2 Càlcul de la direcció de descens.

La implementació que s'ha fet de l'algorisme permet solucionar el sistema $Z^t H(x) Z p_S = -g_z$ del pas (3) i) de dues formes: a través del mètode de Newton-truncat, o usant una aproximació quasi-Newton de l'hessiana reduïda $Z^t H(x) Z$. En cap dels dos casos és necessari disposar d'una expressió analítica de l'hessiana $H(x)$ de la funció objectiu, i només es requereix avaluar el gradient $g(x)$. Les següents subseccions descriuran breument com s'han usat les dues metodologies. Per simplificar la notació, la matriu hessiana serà referida com a H , en comptes de $H(x)$.

5.3.2.1 Mètode de Newton-truncat.

L'algorisme de Newton-truncat usat per solucionar $Z^t H Z p_S = -g_z$ segueix bàsicament la descripció feta a [23]. Es basa en el mètode del gradient conjugat per a la solució de sistemes $Mx = b$, essent M simètrica i definida positiva. En aquest cas la matriu del sistema M és directament l'hessiana reduïda $Z^t H Z$.

El següent algorisme mostra les etapes que cal seguir per solucionar $Z^t H Z p_S = -g_z$ amb el mètode de Newton-truncat.

Algorisme A5.2. Mètode de Newton-truncat per solucionar $Z^t H Z p_S = -g_z$.

Pas 0) Inicialitzar $j = 0$, $p_0 = 0$, $r_0 = g_z$, $d_0 = -r_0$, $\delta_0 = d_0^t d_0$, $\eta = \theta^2 (g_z^t g_z)$.

Pas 1) $q_j = Z^t H Z d_j$

si $d_j^t q_j \leq \sqrt{\varepsilon_M} \delta_j$ llavors acabar: $p_S = \begin{cases} -g_z & \text{si } j=0 \\ p_j & \text{altrament} \end{cases}$

si no $\alpha_j = r_j^t r_j / d_j^t q_j$

Pas 2) $p_{j+1} = p_j + \alpha_j d_j$

$r_{j+1} = r_j + \alpha_j q_j$

si $r_{j+1}^t r_{j+1} \leq \eta$ llavors acabar: $p_S = p_{j+1}$

Pas 3) $\beta_j = r_{j+1}^t r_{j+1} / r_j^t r_j$

$d_{j+1} = -r_{j+1} + \beta_j d_j$

$\delta_{j+1} = r_{j+1}^t r_{j+1} + \beta^2 \delta_j$

$j = j + 1$

anar al Pas 1

Els tres punts crítics d'aquest algorisme són el càlcul de q_j i el control de les dues condicions d'acabament. A continuació veurem com són tractats per la implementació realitzada.

i) Donat que l'hessiana actual H és desconeguda, cal disposar d'un procediment per aproximar-la. La seqüència seguida per calcular $q_j = Z^t H Z d_j$ és:

- $u = Z d_j$
- $v = (g(x + \varepsilon u) - g(x)) / \varepsilon$.
- $q_j = Z^t v$

El pas segon es justifica fent l'expansió en sèrie de Taylor de $g(x + \varepsilon u)$:

$$\begin{aligned} g(x + \varepsilon u) &\approx g(x) + \varepsilon \nabla g(x) u \\ \nabla g(x) u &\approx (g(x + \varepsilon u) - g(x)) / \varepsilon = v \end{aligned}$$

Llavors, donat que $\nabla g(x) u = H u = H Z d_j$, v pot ser considerat una bona aproximació de $H Z d_j$. L'interval de diferenciació ε usat realment és $\varepsilon = \sqrt{\varepsilon_M} / \sqrt{\delta_j}$, on ε_M és la precisió de la màquina, i $\delta_j = d_j^t d_j$ és actualitzada a cada iteració (pas 3) de l'algorisme A5.2. De fet, el valor correcte que hauria de ser usat és $\varepsilon = \sqrt{\varepsilon_M} / \sqrt{u^t u}$ on $u = Z d_j$. L'efecte d'escollir aquest valor ε seria equivalent a realitzar diferències finites usant el vector unitat $u / \|u\|_2$ (on $\|\cdot\|_p$ denota la norma p d'un vector) i un interval de diferenciació igual a $\sqrt{\varepsilon_M}$, situació que és molt més desitjable. Tanmateix, això implicaria calcular la norma 2 de u a cada iteració, mentre que el valor $\sqrt{\delta_j} = \|d_j\|_2$ ja havia estat calculat prèviament i, aleshores, serà emprat per agilitzar el procés (tot i que si $\|u\|_2 \gg \|d_j\|_2$ l'aproximació per diferències finites pot produir resultats erronis).

ii) La primera condició d'acabament al pas 1) de A5.2 intenta controlar si l'hessiana H és o no definida positiva. Si el valor $d_j^t Z^t H Z d_j$ és negatiu es pot concloure que $Z^t H Z$ no és definida positiva (i tampoc ho serà H). Tanmateix, donat que només tenim una aproximació de $Z^t H Z d_j$, tindrem que comparar $d_j^t Z^t H Z d_j$ amb un valor major que 0. La implementació realitzada usa el valor $\sqrt{\varepsilon_M} \delta_j$, on $\delta_j = d_j^t d_j$. Usant aquest valor, el test de comparació pot ser vist com:

$$\begin{aligned} d_j^t Z^t H Z d_j &\leq \sqrt{\varepsilon_M} \delta_j \\ d_j^t Z^t H Z d_j &\leq \sqrt{\varepsilon_M} \|d_j\|_2^2 \\ \frac{d_j^t}{\|d_j\|_2} Z^t H Z \frac{d_j}{\|d_j\|_2} &\leq \sqrt{\varepsilon_M} \end{aligned}$$

el qual significa que s'estan usant vectors normalitzats, disminuint els possibles errors de precisió.

iii) La segona condició d'acabament al pas 2) de A5.2 serà activa quan s'hagi assolit una suficient reducció en la norma del vector r_j . Inicialment $r_0 = g_z$, i (assumint precisió completa) a cada iteració es verifica que $\|r_j\|_2 \leq \|r_{j-1}\|_2$, $j = 1, \dots, s$, essent s el nombre

de variables superbàsiques (que és la dimensió del sistema). Després de s iteracions el vector r_s serà igual a 0, implicant que p_S és la solució del sistema. Tanmateix, per tal d'evitar un excés de càlcul, el procés serà aturat quan s'hagi obtingut una suficient disminució en la norma 2 de r_j . De fet, la condició d'acabament s'hauria d'escriure com:

$$\begin{aligned} r_{j+1}^t r_{j+1} &\leq \eta \\ r_{j+1}^t r_{j+1} &\leq \theta^2 (g_z^t g_z) \\ \frac{r_{j+1}^t r_{j+1}}{g_z^t g_z} &\leq \theta^2 \\ \frac{\|r_{j+1}\|_2}{\|g_z\|_2} &\leq \theta = \max\{\varepsilon_1, \min\{\varepsilon_2, \|g_z\|_2\}\} \end{aligned}$$

on ε_1 és la precisió màxima requerida (a la implementació feta $\varepsilon_1 = \sqrt{\varepsilon_M}$), ε_2 és la precisió requerida quan l'iterat actual x és troba lluny de l'òptim del subespai actual de les variables superbàsiques ($\varepsilon_2 \in [0, 1]$ i pot ser escollit per l'usuari), i $\|g_z\|_2$ és usat com la precisió requerida quan estem a prop de l'òptim al subespai actual.

Un clar desavantatge d'aquesta metodologia és que a cada iteració cal fer una avaluació de la funció objectiu per calcular q_j per diferències finites, el qual augmenta considerablement el cost de l'algorisme. Per altra banda, no cal emmagatzemar cap tipus d'estructura de dades per mantenir $Z^t H Z$ (com ho requereixen els mètodes quasi-Newton que es veuran a continuació), i només s'ha de disposar de quatre vectors de dimensió s per p_j, r_j, d_j i q_j . Si el nombre de superbàsiques és molt gran això pot implicar un gran estalvi de memòria, i pot accelerar considerablement el rendiment del programa.

5.3.2.2 Mètode quasi-Newton.

La segona via per solucionar $Z^t H Z p_S = -g_z$ consisteix en aproximar la matriu $Z^t H Z$ per una factorització $R^t R$ ($R \in \mathbb{R}^{s \times s}$ triangular superior), seguint el procediment descrit a [58]. Aquesta és l'opció per defecte a la implementació realitzada. Per tant, la direcció superbàsica p_S és calculada al pas (3) i) de A5.1 solucionant $R^t R p_S = -g_z$ per substitució directa i inversa. Aquest factorització ha de ser actualitzada si una variable passa a ser superbàsica (pas (2) iii) de A5.1), una variable deixa de ser superbàsica (pas (7) i) b)), hi ha un canvi de base (pas (7) ii) b)), o ens belluguem de x_i a x_{i+1} (pas (5) ii)). Descriurem breument com és realitzada cada una d'aquestes operacions:

i) Una variable passa a ser superbàsica.

En aquest cas s'afegeix una nova columna a la matriu S (submatriu de \tilde{A} associada a les variables superbàsiques) per la nova variable que passa a ser superbàsica. Aleshores la dimensió de la matriu R serà $(s+1) \times (s+1)$, és a dir, s'augmenta en una fila i columna. La implementació realitzada simplement afegeix e_{s+1} (el vector unitari $s+1$) com a nova columna de R . Això implica que, en solucionar $R^t R p_S = -g_z$, a la nova iteració es tindrà que $p_{S_{s+1}} = -g_{z_{s+1}}$ (la direcció del gradient serà usada per a la component $s+1$ de p_S).

Tanmateix, en successives iterations, nova informació de segon ordre serà afegida a aquesta component a través de l'actualització quasi-Newton.

ii) Una variable deixa de ser superbàsica.

En aquest cas desapareix una columna de S (sigui la columna S_q), el qual provoca que la columna associada de R (R_q) també haurà de ser eliminada, obtenint aleshores una matriu de Hessenberg superior. Per tornar R a la seva forma triangular original es pre-multiplica per matrius de Givens escaients $G_{l,l+1}$, $l = q, \dots, s-1$, tal i com es descriu a [58].

iii) Un canvi de base és realitzat.

Considerem que la p -èssima variable bàsica és substituïda per la q -èssima variable superbàsica. Aleshores, la matriu del nou subespai nul Z_n pot ser escrita com $Z_n = Z(\mathbb{1} + e_q u^t)$, on e_q és el q -èssim vector unitari, i $u = -1/y_q(y + \mathbb{1}_q)$, essent $y = B^{p-1}S$. Donat que $R^t R \approx Z^t H Z$, la nova factorització R_n pot ser obtinguda tal i com segueix:

$$\begin{aligned} R_n^t R_n &\approx Z_n^t H Z_n \\ &\approx (\mathbb{1} + u e_q^t) Z^t H Z (\mathbb{1} + e_q u^t) \\ &\approx (\mathbb{1} + u e_q^t) R^t R (\mathbb{1} + e_q u^t) \\ &\approx (R^t + u R_q^t) (R + R_q u^t) \\ &\Downarrow \\ R_n &= (R + R_q u^t) \end{aligned}$$

essent R_q la q -èssima columna de R . Una descripció més detallada d'aquest procés pot ser trobada a [58].

iv) Actualització quasi-Newton.

Quan hi ha un moviment de x_i a x_{i+1} l'aproximació $R^t R$ de $Z^t H(x_i) Z$ ha de ser modificada per tenir en compte la nova hessiana reduïda $Z^t H(x_{i+1}) Z$. A la implementació desenvolupada s'ha usat la ben coneguda formula d'actualització BFGS (Broyden-Fletcher-Goldfarb-Shanno) de la matriu hessiana [30], la qual en aquest cas pot ser escrita com:

$$R_n^t R_n = R^t R + \frac{1}{\alpha^* y^t p_S} y y^t + \frac{1}{g_z^t p_S} g_z g_z^t \quad (5.17)$$

on α^* és la passa òptima obtinguda en fer l'exploració lineal (pas (5) i) de A5.1), g_z és el gradient reduït $Z^t g(x_i)$ al punt x_i , p_S és la direcció superbàsica (trobada al pas (3) i) de A5.1), i y es defineix com $y = g(x_{i+1}) - g(x_i)$. La següent proposició mostra com s'ha calculat la nova matriu de factorització R_n a partir de l'anterior R a la implementació desenvolupada.

Proposició 5.1.

Donada la fórmula BFGS d'actualització

$$R_n^t R_n = R^t R + \frac{1}{\alpha^* y^t p_S} y y^t + \frac{1}{g_z^t p_S} g_z g_z^t$$

i definint $\delta_1 = \frac{1}{\sqrt{-g_z^t p_S}}$, $\delta_2 = \frac{1}{\sqrt{\alpha^* y^t p_S}}$, v tal que $R^t v = g_z$, $\bar{v} = -\delta_1 v$, i $p = \delta_2 y + \delta_1 g_z$ ($\delta_1, \delta_2 \in \mathbb{R}$; $v, \bar{v}, p \in \mathbb{R}^s$, essent s la dimensió de R),

aleshores R_n pot ser calculada com $R_n = R + \bar{v} p^t$. (Així és com està implementat al paquet MINOS [59]).

Demostració.

Es veurà que, definint $R_n = R + \bar{v} p^t$, llavors $R_n^t R_n = R^t R + \frac{1}{\alpha^* y^t p_S} y y^t + \frac{1}{g_z^t p_S} g_z g_z^t$, essent R_n , per tant, la nova matriu de factorització. Efectivament, pot comprovar-se que:

$$\begin{aligned} R_n^t R_n &= (R^t + p \bar{v}^t)(R + \bar{v} p^t) \\ &= R^t R + R^t \bar{v} p^t + p \bar{v}^t R + p(\bar{v}^t \bar{v}) p^t \\ &\quad (\text{usant que } \bar{v} = -\delta_1 v) \\ &= R^t R + (-\delta_1) R^t v p^t + (-\delta_1) p v^t R + \delta_1^2 (v^t v) p p^t \\ &\quad (\text{usant que } R^t v = g_z) \\ &= R^t R + (-\delta_1) g_z p^t + (-\delta_1) p g_z^t + \delta_1^2 (v^t v) p p^t \\ &\quad (\text{usant que } p = \delta_2 y + \delta_1 g_z) \\ &= R^t R - \delta_1 \delta_2 g_z y^t - \delta_1^2 g_z g_z^t - \delta_1 \delta_2 y g_z^t - \delta_1^2 g_z g_z^t + \delta_1^2 (v^t v) p p^t \\ &= R^t R - \delta_1 \delta_2 (g_z y^t + y g_z^t) - 2\delta_1^2 g_z g_z^t + \delta_1^2 (v^t v) p p^t \end{aligned} \quad (5.18)$$

Donat que $p = \delta_2 y + \delta_1 g_z$ llavors $p p^t$ pot ser expressat directament com:

$$\begin{aligned} p p^t &= (\delta_2 y + \delta_1 g_z)(\delta_2 y^t + \delta_1 g_z^t) \\ &= \delta_2^2 y y^t + \delta_2 \delta_1 y g_z^t + \delta_1 \delta_2 g_z y^t + \delta_1^2 g_z g_z^t \end{aligned} \quad (5.19)$$

Ara es mostrarà que el coeficient $\delta_1^2 (v^t v)$ és igual a 1. La direcció de descens superbàsica p_S és calculada com $R^t R p_S = -g_z$. Donat que $R^t v = g_z$ es té que $v = -R p_S$. Per tant $v^t v = p_S^t R^t R p_S = -p_S^t g_z$. Com que $\delta_1 = \frac{1}{\sqrt{-g_z^t p_S}}$, directament es té

$$\delta_1^2 (v^t v) = \frac{1}{-g_z^t p_S} (-p_S^t g_z) = 1 \quad (5.20)$$

Substituint (5.19) i (5.20) a (5.18) tenim:

$$\begin{aligned} R_n^t R_n &= R^t R - \delta_1 \delta_2 (g_z y^t + y g_z^t) - 2\delta_1^2 g_z g_z^t + \delta_2^2 y y^t + \delta_2 \delta_1 y g_z^t + \delta_1 \delta_2 g_z y^t + \delta_1^2 g_z g_z^t \\ &= R^t R - \delta_1^2 g_z g_z^t + \delta_2^2 y y^t \end{aligned}$$

$$= R^t R + \frac{1}{g_z^t p_S} g_z g_z^t + \frac{1}{\alpha^* y^t p_S} y y^t$$

obtenint el resultat desitjat. ■

El principal avantatge del mètode quasi-Newton respecte el mètode de Newton-truncat és que no requereix avaluacions addicionals de la funció objectiu. Per a funcions objectius que tinguin una avaluació costosa això pot suposar un important estalvi de temps. Per altra banda, la matriu R ha de ser emmagatzemada de forma densa (s'emmagatzemen $s(s+1)/2$ components) per poder realitzar eficientment les operacions d'eliminació i addició d'una columna, i actualització BFGS. Tanmateix, en el cas de tenir un nombre de variables superbàsiques molt elevat, la gestió de R pot esdevenir prohibitiva. L'opció per defecte de la implementació realitzada usa el mètode quasi-Newton fins que s'assoleix un determinat nombre de variables superbàsiques (aquest valor pot ser modificat per l'usuari), passant a usar aleshores el mètode de Newton-truncat.

5.3.3 Test d'optimalitat al subespai actual.

A cada iteració s'ha de comprovar si el punt òptim del subespai actual ha estat assolit (pas (1) i) de A5.1). Tanmateix, el test realitzat és molt més exhaustiu que la simple verificació de si $\|g_z\| \leq \epsilon_{g_z}$. La implementació feta discerneix entre dues situacions: quan encara no s'ha assolit el conjunt òptim de constriccions actives (per tant, s'està lluny del punt òptim), i quan ja estem al conjunt òptim de constriccions actives i només cal fer una darrera i més acurada optimització al subespai actual. La variable sa ens dirà quina és la situació actual, i pot prendre els valors “lluny” o “a prop” depenent de si som al primer o al segon cas. S'usen sis variables lògiques (T_i) per decidir si el punt òptim del subespai actual ha estat assolit. Cada T_i és definida com segueix:

$$\begin{aligned} T_1 &:= (\alpha^* \|p_S\|_1 \leq (\epsilon_x^{sa} + \sqrt{\epsilon_M})(1 + \|x_S\|_1)) \\ T_2 &:= (|\Delta f| \leq (\epsilon_f^{sa} + \epsilon_M)(1 + |f|)) \\ T_3 &:= (\|g_z\|_\infty \leq T_{g_z}) \\ T_4 &:= (\|g_z\|_\infty \leq \max\{T_{g_z}/10, \epsilon_g \epsilon(\|\pi\|_1)\}) \\ T_5 &:= ((T_5_és_activa) \text{ i } (sa = \text{“lluny”}) \text{ i } (nmateix \leq MAXmateix)) \\ T_6 &:= (nactual \leq MAXactual) \end{aligned}$$

- T_1 : El primer test, T_1 , controla si la norma 1 del moviment actual en les variables superbàsiques $\alpha^* \|p_S\|_1$ és significatiu respecte la norma 1 de les components superbàsiques a l'iterat actual $\|x_S\|_1$, usant a la comparació la precisió de la màquina ϵ_M i el valor ϵ_x^{sa} que depèn de la variable sa (si $sa = \text{“a prop”}$ aquest valor serà menor que quan $sa = \text{“lluny”}$, exigint un menor moviment per satisfer el test).
- T_2 : El segon test, T_2 , serà cert quan la variació en la funció objectiu $|\Delta f|$ no és significativa respecte el valor absolut de f a l'iterat actual ($|f|$). El valor ϵ_f^{sa} usat a la comparació depèn de la variable sa , i, com al cas previ, $\epsilon_f^{\text{“a prop”}} \ll \epsilon_f^{\text{“lluny”}}$.

- T_3 : En realitzar el test T_3 , la tolerància T_{g_z} ja ha estat prèviament calculada com $T_{g_z} = \eta_{g_z}^{sa} \|g_z^0\|_\infty$, on g_z^0 era el gradient reduït al primer punt del subespai actual, i $\eta_{g_z}^{sa} \in [0, 1]$ és un valor que pot ser escollit per l'usuari. Per tant, aquest test intenta controlar quan s'ha assolit una suficient reducció del gradient reduït des que s'inicià l'optimització al subespai actual. Quan $sa = \text{"a prop"}$ és desitjable imposar una major reducció, i llavors, com als casos previs, $\eta_{g_z}^{\text{"a prop"}} \ll \eta_{g_z}^{\text{"lluny"}}$.
- T_4 : El següent test, T_4 , serà cert si el gradient reduït és prou petit com per considerar el punt actual com a òptim del subespai actual. En aquest cas el valor ε_g no depèn de com de lluny estem del conjunt òptim de constriccions actives (variable sa), i $\epsilon(\|\pi\|_1)$ és una funció que depèn del vector de multiplicadors π calculat al pas (6) i) de A5.1. A la implementació actual $\epsilon(\|\pi\|_1)$ s'ha definit com:

$$\epsilon(\|\pi\|_1) = \max\left\{1, \frac{\|\pi\|_1}{\sqrt{mk + n + t}}\right\} \quad (5.21)$$

on m era el nombre de nodes, k el nombre d'articles, n el nombre d'arcs i t el nombre de constriccions a banda. El coeficient $\sqrt{mk + n + t}$ representa un intent d'escalar les toleràncies d'optimalitat segons la mida del problema.

- T_5, T_6 : Els dos darrers tests s'han inclòs per poder tractar funcions molt poc "suaus" (entenen per "suaus" funcions que no tenen canvis de pendent molt sobtats), en les quals l'aplicació única dels quatre tests previs podria provocar una convergència molt lenta. El primer d'aquests tests (T_5) per defecte es troba inactiu (l'usuari pot decidir activar-lo si ho desitja) i només s'aplica quan $sa = \text{"lluny"}$. T_5 esdevindrà cert si els tres primers tests, T_1, T_2 i T_3 , donen el mateix resultat (comptatge portat per la variable $nmateix$) durant $MAXmateix$ iteracions consecutives al subespai actual (on el valor $MAXmateix$ pot ser escollit per l'usuari). El segon test, (T_6) controla si el nombre d'iteracions al subespai actual ($nactual$) és major que un màxim nombre permès d'iteracions $MAXactual$.

Un cop els sis tests lògics han estat fets, es considerarà que l'òptim del subespai actual s'ha assolit si la variable lògica T és certa, on T es defineix com:

$$T := (T_1 \text{ i } T_2 \text{ i } T_3) \text{ o } T_4 \text{ o } T_5 \text{ o } T_6$$

Per tant, al pas (1) i) de A5.1, la condició que realment és verificada per veure si s'ha d'anar al pas (3) és "si T és certa" en comptes de "si $\|g_z\| \geq \epsilon_{g_z}$ ".

5.3.4 Elecció de la variable no bàsica que esdevindrà superbàsica.

Al pas (2) ii) de A5.1, el procés d'elecció d'una variable no bàsica va ser reduït a *Escollir una λ_q que violi (5.12) o (5.13), amb la seva columna associada N_q* . De fet la implementació realitzada contempla un algorisme força més sofisticat per escollir la variable que esdevindrà superbàsica, punt crucial del procediment ja que una mala elecció

o unes toleràncies poc acurades poden significar una molt lenta convergència en el procés de cerca del conjunt òptim de constriccions actives.

El següent algorisme mostra la seqüència exacta de passes en que el punt (2) ii) de A5.1 és expandit a la implementació realitzada. L'algorisme usa una tolerància T_{λ_q} per escollir un bon multiplicador λ_q . Al principi de la fase 2 aquesta tolerància és inicialitzada amb un valor arbitrari gran $T_{\lambda_q}^0$. La variable *sa* per saber com de lluny estem del conjunt òptim de constriccions actives és també consultat i actualitzat en aquest algorisme. La funció $\epsilon(\|\pi\|_1)$ va ser definida a (5.21), i la tolerància T_{g_z} —per detectar una reducció suficient en la norma del gradient reduït $\|g_z\|_\infty$ — ja ha estat introduïda a la secció anterior. El valor ϵ_{opt} , escollit per l'usuari, és la precisió d'optimalitat requerida al punt òptim (per defecte $\epsilon_{opt} = 10^{-6}$). Cal remarcar que al pas 1 sempre s'escull el major dels valors $1.12 \cdot \|g_z\|_\infty$ i T_{λ_q} per procedir a buscar un bon multiplicador λ_q , seguint la recomanació de [58]. Això es fa perquè el valor λ_q escollit serà afegit com a component del nou gradient reduït, el qual implica que així garantim que aquest valor serà significatiu respecte la resta de components de l'actual gradient reduït g_z .

Algorisme A5.3. Elecció de la variable no bàsica que esdevindrà superbàsica.

- 0) En començar la fase 2 inicialitzem $T_{\lambda_q} := T_{\lambda_q}^0$ i *sa* := “lluny”.
- 1) $T_{\lambda_q} := \max\{T_{\lambda_q}, 1.12 \cdot \|g_z\|_\infty\}$.
- 2) Buscar el primer λ_q tal que $|\lambda_q| \geq T_{\lambda_q}$ o bé, si no n'hi ha cap, el major $|\lambda_q|$.
- 3) Si ($|\lambda_q| \geq T_{\lambda_q}$) anar a 7).
- 4) Cap multiplicador satisfà la tolerància actual T_{λ_q} .
Si ($|\lambda_q| \geq \epsilon_{opt}\epsilon(\|\pi\|_1)$) llavors
 - i) $T_{\lambda_q} := \max\{|\lambda_q|/10, \epsilon_{opt}\epsilon(\|\pi\|_1)\}$.
 - ii) Anar a 7).
- 5) Cap multiplicador és major que la tolerància d'optimalitat $\epsilon_{opt}\epsilon(\|\pi\|_1)$. Estem a prop de l'òptim.
Si (*sa* = “a prop”) o ($\|g_z\|_\infty \leq \epsilon_{opt}\epsilon(\|\pi\|_1)$) llavors anar a 8).
- 6) Cap multiplicador és major que la tolerància d'optimalitat i el punt no satisfà les condicions d'optimalitat. Ajustem les toleràncies per una optimització més acurada —probablement la darrera— al subespai actual.
 - i) $T_{g_z} := \frac{\min\{T_{g_z}, \|g_z\|_\infty\}}{10}$.
 - ii) Si ($T_{g_z} < \epsilon_{opt}\epsilon(\|\pi\|_1)$) llavors
 - *sa* := “a prop”.
 - $T_{g_z} := \epsilon_{opt}\epsilon(\|\pi\|_1)$.
 - iii) Continuar amb el pas (3) de l'algorisme A5.1.
- 7) Una λ_q escaient ha estat trobada:
 - i) Actualitzar $\|g_z\|_\infty := \max\{|\lambda_q|, \|g_z\|_\infty\}$.

- ii) Actualitzar $T_{g_z} := \eta_{g_z}^{sa} \|g_z\|_\infty$.
- iii) Si ($sa = \text{“a prop”}$) llavors $sa := \text{“lluny”}$.
- iv) Continuar amb el pas (2) iii) de A5.1.

8) El punt actual satisfà les condicions d’optimalitat. Anar al pas (8) de A5.1 (la solució òptima ha estat trobada).

La idea d’aquest algorisme consisteix en usar inicialment unes toleràncies T_{λ_q} arbitràriament grans i en reduir-les quan no s’ha pogut trobar cap multiplicador amb un valor major. Aquest valor és reduït fins que s’arriba a la tolerància d’optimalitat $\epsilon_{opt} \epsilon(\|\pi\|_1)$. Quan això ocorre, es pot considerar que el conjunt òptim de constriccions actives ha estat assolit, i la tolerància T_{g_z} és ajustada per una darrera i més acurada optimització al subespai actual.

5.3.5 Variables superbàsiques quasi-actives.

En solucionar $Z^t H Z p_S = -g_z$, la implementació feta permet considerar un conjunt de variables superbàsiques que seran referides com a *variables superbàsiques quasi-actives*. El conjunt de variables quasi-actives es defineix com:

$$\mathcal{I}^+ = \{i : |x_{S_i}| < \epsilon_{qact} \text{ i } g_{z_i} < 0\} \cup \{i : |x_{S_i} - \bar{x}_{S_i}| < \epsilon_{qact} \text{ i } g_{z_i} > 0\} \quad (5.22)$$

essent x_S el vector de variables superbàsiques, \bar{x}_S les seves fites superiors, g_z el vector gradient reduït, i ϵ_{qact} un valor proper a 0 usat per determinar quan una variable superbàsica és propera a la seva fita i cal considerar-la quasi-activa. La resta de variables superbàsiques es consideraran pertanyents al conjunt \mathcal{I}^- . Quan l’opció de considerar aquest tipus de variables superbàsiques és activa, $Z^t H Z p_S = -g_z$ serà solucionat de la següent forma:

$$\begin{array}{|c|c|} \hline Z^t H Z_{\mathcal{I}^-} & 0 \\ \hline 0 & \mathbb{1}_{\mathcal{I}^+} \\ \hline \end{array} \begin{array}{|c|} \hline p_{S_{\mathcal{I}^-}} \\ \hline p_{S_{\mathcal{I}^+}} \\ \hline \end{array} = \begin{array}{|c|} \hline -g_{z_{\mathcal{I}^-}} \\ \hline -g_{z_{\mathcal{I}^+}} \\ \hline \end{array} \quad (5.23)$$

on $M_{\mathcal{I}^*}$ (* pot ser + o -) és el vector (matriu) compost per les i -èsimes components (files i columnes) de M tal que $i \in \mathcal{I}^*$. Per tant la solució de (5.23) és directament:

$$Z^t H Z_{\mathcal{I}^-} p_{S_{\mathcal{I}^-}} = -g_{z_{\mathcal{I}^-}} \quad (5.24)$$

$$p_{S_{\mathcal{I}^+}} = -g_{z_{\mathcal{I}^+}} \quad (5.25)$$

Per la definició de (5.22), queda clar que usant la direcció de descens $p_{S_{\mathcal{I}^+}}$ calculada a (5.25) obligarem a les variables quasi-actives a prendre un valor allunyat de la seva fita (tot mantenint la factibilitat) en fer l’actualització del nou punt al pas (5) ii) de A5.1. El motiu per usar aquest moviment p_S és, doncs, evitar trobar possibles direccions de descens mitjançant $Z^t H Z p_S = -g_z$ que obliguessin a prendre el valor $\bar{\alpha} = 0$ al pas (4) i) de A5.1 (una passa degenerada). El fet d’evitar aquest tipus de passes en aquest cas elimina la possibilitat de ciclatge en el procés d’introducció i eliminació de variables superbàsiques. L’opció per defecte és no usar les variables quasi-actives. Tanmateix, quan

una passa degenerada és assolida, provocada per una variable superbàsica quasi-activa, l'opció s'activa automàticament —només a la següent iteració— per tal d'evitar l'inici d'un possible ciclatge.

5.3.6 Estructures de dades.

A §4.4.4 es van comentar breument algunes de les estructures de dades usades en la implementació del particionament primal per a problemes lineals. Donat que aquell era un cas particular del mètode descrit en aquest capítol per a problemes no lineals, totes les estructures que allà van ser comentades continuen essent usades en la implementació no lineal. A més, al cas no lineal apareixen noves estructures, associades amb el fet de disposar de variables superbàsiques. Comentarem molt breument tres d'aquestes estructures.

- 1.- *Estat dels arcs.* A diferència del cas lineal, on un arc només podia ser bàsic o no bàsic, al cas no lineal s'afegeix la nova categoria d'arc superbàsic. En tot moment cal poder saber, donat un arc qualsevol j , a quin dels tres tipus pertany. Això s'aconsegueix mantenint un vector de doble entrada `statarc(1:n,1:k)`, on cada component pot prendre els valors 1, 2 o 3 (on 1 significa “bàsic”, 2 “no bàsic”, i 3 “superbàsic”). Aleshores, per conèixer l'estat de l'arc j de l'article i , només caldrà consultar el valor de `statarc(j,i)`.
- 2.- *Folgues.* Ja va ser comentat a §4.4.4 com s'emmagatzemava la informació relativa a les folgues. Concretament, si es volia conèixer quines folgues eren no bàsiques, donat que aquestes estaven associades a les constriccions actives del conjunt $|\mathcal{A}|$, només calia recórrer el vector `actives`, de forma que `actives(l)=±j` ens indicava que la l -èssima constricció activa corresponia a la j -èssima constricció de capacitat mútua (cas $+j$) o a banda (cas $-j$). D'aquesta forma es determinava que la folga associada amb aquesta constricció de capacitat mútua o a banda era no bàsica, ja que no apareixia a la base B . Al cas no lineal, però, el fet de que no aparegui a B no significa que sigui no bàsica, ja que bé podria ser una folga superbàsica. Per solucionar això es pot disposar de dos vectors per conèixer l'estat de cada folga, anomenats `statslcm(1:n)` —per a les folgues de les constriccions de capacitat mútua— i `statslcb(1:t)` —per a les folgues de les constriccions a banda—, de forma que si `statslcm(l)=0` (o `statslcb(l)=0`) es considerarà la folga no bàsica, mentre que si `statslcm(l)=1` (o `statslcb(l)=1`) es considerarà superbàsica. Donat que només cal fer aquesta distinció a la fase 2 de l'algorisme (ja que és la única on poden aparèixer folgues superbàsiques), aquests dos vectors són usats com espai de treball durant la fase 1 per emmagatzemar informació relativa a les variables f_{cm} i f_{cb} de les equacions (4.34), (4.35), i (4.36), les quals emmagatzemen les infactibilitats del problema.
- 3.- *Variables superbàsiques.* Per tal de saber quines variables són superbàsiques, es disposa de dos vectors de dimensió s (nombre de superbàsiques) anomenats `ksup(1:s)` i `jsup(1:s)`. La forma de procedir per determinar quina és la variable superbàsica j -èssima contempla dos casos:

- $\mathbf{k\text{sup}}(l) = 0$. La l -èssima variable superbàsica serà una folga, i s'haurà de consultar el vector $\mathbf{j\text{sup}}$ per saber a quina constricció es troba associada. Així, si $\mathbf{j\text{sup}}(l) = +j$ ($j > 0$) la variable superbàsica serà la folga s_{cm_j} (associada amb la constricció j de capacitat mútua), mentre que si $\mathbf{j\text{sup}}(l) = -j$ es tractarà de la folga s_{cb_j} de la j -èssima constricció a banda.
- $\mathbf{k\text{sup}}(l) = i$. En aquest cas la l -èssima variable és un arc de l'article i . Per saber l'arc de que es tracta haurem de consultar el valor de $\mathbf{j\text{sup}}(l)$ (és a dir, si $\mathbf{k\text{sup}}(l) = i$ i $\mathbf{j\text{sup}}(l) = j$, l'arc j de l'article i és la l -èssima variable superbàsica).

Capítol 6

Experiència computacional amb el particionament primal.

Al llarg dels capítols 4 i 5 s'han descrit les bases algorísmiques del mètode del particionament primal per problemes lineals i no lineals, així com certs aspectes de la implementació desenvolupada, la qual ha donat lloc al paquet PPRN [11] (de Particionament PRimal No lineal). En aquest capítol es farà una molt breu descripció d'alguns aspectes del codi, i es presentaran els resultats computacionals que amb ell s'han obtingut executant una bateria de tests, tot comparant-lo amb d'altres codis especialitzats.

6.1 Breu descripció del paquet PPRN.

PPRN és un codi especialitzat per solucionar els problemes (**FMLCB**) i (**FMNCB**), basat en el mètode del particionament primal per a problemes lineals i no lineals descrit als dos capítols anteriors. Ha estat escrit bàsicament en Fortran-77, i només s'han codificat en ANSI-C les rutines que serveixen com d'interfície entre el paquet i l'aplicació de l'usuari. En total, el paquet PPRN consta de 18 fitxers Fortran-77 (que suposen un total de 16968 línies de codi) i 2 fitxers ANSI-C (amb 1224 línies de codi). A continuació es farà una breu descripció d'alguns aspectes rellevants del paquet. Per una descripció més acurada adresem al lector interessat al manual d'usuari del paquet, descrit a [11].

Les rutines escrites en ANSI-C, a més de servir com d'interfície entre l'usuari i el paquet, són les encarregades de gestionar de forma dinàmica la memòria requerida en executar un model determinat. El fet d'usar assignació dinàmica de memòria evita innecessàries recompilacions de tot el paquet per tal d'ajustar les dimensions dels vector a la mida del problema, facilitant-ne el seu ús.

PPRN pot ser usat de dos formes diferents: com a programa independent, o bé com a subrutina dins d'una aplicació més gran. Quan és usat com a programa independent, se li ha de subministrar el problema a solucionar a través d'un fitxer, mentre que quan s'usa com a rutina totes les dades li han de ser transmeses per la línia de paràmetres. En ambdós casos, però, la informació a subministrar és la mateixa, i consisteix en les dimensions del problema (nombre de nodes, arcs, articles i constriccions a banda), costos lineals per arc i article, capacitats dels arcs per article, injeccions als nodes per article, capacitat mútua dels arcs, topologia de la xarxa, límits inferiors i superiors de les constriccions a banda, i estructura de les constriccions a banda.

Quan s'executa un problema lineal, amb la informació prèviament detallada n'hi ha prou per obtenir la solució òptima. Tanmateix, per al cas no lineal s'ha de subministrar

una rutina per tal d'avaluar la funció objectiu $f(x)$ i el gradient $g(x)$, donat un punt factible x . Aquesta rutina, de nom qualsevol, és passada com un paràmetre més al paquet PPRN.

L'usuari, a més, pot modificar diversos paràmetres, els quals controlen l'execució del paquet, a través d'un fitxer d'especificacions. Si aquest fitxer no és trobat per PPRN, s'usaran els valors per defecte. En aquest fitxer s'han d'introduir comandes amb la sintaxi: "Paràmetre_a_especificar valor_del_paràmetre". Hi ha un total de 30 paràmetres que poden ser controlats per l'usuari, la descripció exhaustiva dels quals pot ser trobada a [11]. A continuació es presenta de forma resumida algunes de les opcions algorísmiques més significatives implementades a PPRN, les quals poden ser modificades per l'usuari.

1.- *Fase 0.* Tal i com es va comentar a §4.4.1.1, PPRN soluciona a la fase 0 un problema de fluxos lineals uniaarticle per a cada un dels k articles, usant una versió ampliada i millorada de les rutines especialitzades LEXA descrites a [60]. L'usuari pot decidir entre trobar el punt òptim per aquests problemes, o simplement un punt factible. L'opció per defecte és trobar un punt factible.

Així mateix, durant la fase 0 pot ser activat un control per evitar passes degenerades, mantenint una llista de variables no bàsiques que produïrien una passa nul·la. Per defecte aquest control es troba inactivat, i en la majoria de casos aquesta és l'opció més eficient.

Un altre dels aspectes que poden ser controlats durant la fase 0 és la connectivitat de la xarxa. En principi, PPRN requereix que tots els nodes de la xarxa han d'estar connectats (és a dir, no es permet tenir grafs disjunts). Per defecte no es verifica la connectivitat de la xarxa, i s'assumeix. Això, però, pot provocar errors durant l'optimització. Per evitar aquest fet, el control de connectivitat pot ser activat.

2.- *Freqüència d'inversió de la matriu de treball.* Aquesta freqüència, introduïda i denotada per ν a §4.3.2.1, permet controlar cada quantes iteracions es calcularà la matriu Q (aplicant els resultats de les proposicions P4.3 i P4.4) i serà posteriorment factoritzada. El valor per defecte és $\nu= 50$ iteracions.

3.- *Taxació de les variables no bàsiques.* Com ja va ser comentat a §4.4.2, es poden considerar quatre tipus d'ordenacions de les variables no bàsiques. PPRN permet escollir entre una d'aquestes quatre, de forma independent pel que fa a la fase 1 o la fase 2. Per defecte l'ordenació número 1 és usada en ambdues fases.

Així mateix, l'usuari pot controlar els paràmetres $\#nmb$ i $\#nblocs$, introduïts a §4.4.2, que ens donen la dimensió dels blocs de taxació i el nombre de blocs a taxar respectivament. Per defecte es té que $\#nblocs= 10$, i que $\#nmb$ es calcula com va ser definit a (4.38) (si l'usuari, però, fixa un valor determinat per $\#nmb$, no s'aplicarà el càlcul (4.38), i el valor de $\#nblocs$ deixarà de ser rellevant).

4.- *Optimització a partir d'un punt prèviament salvat.* PPRN pot iniciar l'optimització a partir d'un punt qualsevol. Aquest punt es troba emmagatzemat dins un fitxer que s'ha generat en una execució prèvia del paquet. Per defecte no es llegeix cap fitxer

amb un punt inicial, i l'usuari haurà d'explicitar el nom d'aquest fitxer si no vol que l'execució s'iniciï en un punt per defecte. Així mateix, PPRN salva l'estat actual de l'execució en funció d'un paràmetre que també pot ser modificat per l'usuari. Per defecte, l'estat actual és salvat cada 500 iteracions.

- 5.- *Càlcul de la direcció de descens.* El càlcul de la direcció de descens per a les variables superbàsiques p_S pot ser escollida entre tres opcions. La primera consisteix en usar la direcció de $-g_z$, de càlcul fàcil, però de convergència lenta. Les altres dues ja van ser comentades a §5.3.2, i consisteixen en usar el mètode de Newton-truncat o un mètode quasi-Newton. L'opció per defecte consisteix en usar el mètode quasi-Newton, en general més eficient, a no ser que el nombre de variables superbàsiques es faci massa gran.

Quan s'usa el mètode de Newton-truncat per calcular la direcció de descens, es pot afegir una condició d'acabament addicional, a banda de les dues que ja incorpora l'algorisme A5.2. Aquesta consisteix en fixar un nombre màxim d'iteracions permès, reduint el cost del procés (recordem que a cada iteració es requereix una avaluació addicional de la funció objectiu) a costa d'obtenir pitjors direccions. Per defecte aquest límit màxim està en 20 iteracions, tot i que pot ser modificat per l'usuari.

També usant l'algorisme de Newton-truncat, l'usuari pot determinar el paràmetre $\varepsilon_2 \in [0, 1]$, el qual va ser usat a §5.3.2.1 per controlar la bondat de la solució obtinguda quan ens trobàvem lluny de l'òptim. Com més a prop estigui ε_2 de 0, més acurat serà el punt trobat, i més costós el seu càlcul. El valor per defecte és 0.5.

- 6.- *Control del nombre de variables superbàsiques que determinen el canvi de mètode per calcular la direcció de descens.* Aquest paràmetre es troba directament associat amb el punt anterior, i amb ell l'usuari pot controlar a partir de quin nombre de variables superbàsiques es passarà a usar el mètode de Newton-truncat en comptes de l'actual mètode quasi-Newton (naturalment aquest control només té sentit si es comença l'execució usant el mètode quasi-Newton). Per defecte el valor límit considerat és de 500 variables superbàsiques.

- 7.- *Exploració lineal.* Per realitzar l'exploració lineal s'ha usat la rutina GETPTC del paquet MINOS [59] d'optimització. Aquesta rutina ve controlada per un paràmetre $\alpha_l \in [0, 1]$. Com més a prop α_l estigui de 0, més acurada serà l'exploració lineal. El valor per defecte és $\alpha_l = 0.1$.

- 8.- *Precisió de l'optimització al subespai actual.* El mètode de les constriccions actives, descrit a l'algorisme A5.1, es basava en successives optimitzacions al subespai actual de les variables superbàsiques, per tal de reduir la norma del gradient reduït g_z en el primer iterat d'aquest subespai. Aquesta reducció venia controlada pel paràmetre $\eta_{g_z}^{sa} \in [0, 1]$, usat a §5.3.3 i §5.3.4. L'usuari pot modificar aquest valor quan $sa =$ "lluny" (el valor per defecte és 0.5). Quan $sa =$ "a prop" el valor es fixe i val $\eta_{g_z}^{\text{a prop}} = 0.1$.

- 9.- *Precisió d'optimalitat.* A §5.3.4 es va usar el paràmetre ϵ_{opt} com a tolerància d'optimalitat a imposar al punt que es consideri solució del problema. Per defecte es té que $\epsilon_{opt} =$

10^{-6} . Valors més petits d'aquest valor donen lloc a òptims més acurats (tot i que no és recomanable disminuir-lo massa per evitar problemes de precisió i de convergència, i, en general, el guany obtingut no serà significatiu).

Cal fer notar que PPRN, tot i haver estat dissenyat per solucionar problemes multiarticle (tant lineals com no lineals), pot ser usat també com a paquet especialitzat per resoldre:

- problemes de fluxos uniarticle (lineals i no lineals)
- problemes de fluxos uniarticle amb constriccions a banda (lineals i no lineals).

Per solucionar problemes uniarticle lineals només caldrà executar la fase 0 amb l'opció d'obtenir un punt òptim. Si el problema és no lineal, després d'executar la fase 0 es passarà directament a la fase 2 (la fase 1 no intervé donat que no hi ha constriccions a banda) per tractar les no-linealitats.

Si el problema definit només té un únic article ($k=1$) i constriccions a banda, aleshores PPRN es comportarà com un codi especialitzat per solucionar problemes de fluxos amb constriccions a banda. A diferència d'altres codis, però, com ara els presentats a [37,38,50] (tots ells basats també en un particionament de la base per a problemes amb constriccions a banda [49]), PPRN considera una dimensió variable de la matriu de treball, com va ser descrit al capítol 4, la dimensió de la qual està associada al nombre de constriccions actives. Per la seva banda, les altres implementacions consideren sempre una matriu de treball de dimensió fixa i igual al total de constriccions del problema. Naturalment, si totes les constriccions del problema són d'igualtat, al punt òptim la dimensió de la base serà la mateixa per les dues estratègies (dimensió fixa i variable), ja que totes són actives. Per contra, si el nombre de constriccions de desigualtat és significatiu (i pot esperar-se que a l'òptim només una part d'elles siguin actives), gestionar matrius amb dimensió variable pot suposar un estalvi de temps important respecte haver de tractar una matriu de dimensió fixa i major.

6.2 Problemes test usats.

En aquesta secció es descriuran els problemes test usats per comparar el rendiment del paquet PPRN respecte altres codis especialitzats. Es farà una clara diferenciació entre els tests lineals i els no lineals.

6.2.1 Problemes test lineals.

S'han usat dos tipus de problemes lineals. El primer tipus ha estat obtingut de diversos generadors automàtics de xarxes, mentre que el segon consisteix en problemes reals que apareixen en el camp de la coordinació hidrotèrmica. Passem a descriure, tot seguit les característiques de cada problema en concret.

6.2.1.1 Problemes lineals obtinguts de generadors automàtics.

S'han usat dos tipus de generadors automàtics. El primer grup consisteix en quatre generadors de problemes de xarxes uniarticle (Rmfgen [32], Grid-on-torus, Gridgraph i Gridgen), obtinguts del conjunt de generadors distribuïts pel First DIMACS International Algorithm Implementation [25]. El segon grup, per la seva banda, està format només per un únic generador multiarticle, anomenat Mnetgen [4]. Tots aquests generadors han estat obtinguts via ftp anònim de *dimacs.rutgers.edu* i es troben al directori */pub/netflow*.

1. Generadors DIMACS.

Els generadors del primer grup generen una sortida en format DIMACS la qual conté un problema uniarticle. Aquesta sortida proporciona la següent informació:

- El nombre de nodes i arcs de la xarxa.
- Les injeccions (positives i negatives) als nodes de producció i demanda.
- Per a cada arc, els seus nodes origen i destí, les capacitats mínima i màxima, i el coeficient de cost lineal.

Tot i que PPRN pot tractar constriccions a banda, els problemes obtinguts amb aquest generador no en tenen cap. A més, com ja ha estat esmentat prèviament, generen un problema uniarticle, el qual ha de ser convertit a un problema multiarticle. Per a aquest propòsit s'ha desenvolupat un conversor de problemes uniarticle (llegits en format DIMACS) a problemes multiarticle (directament en format que pugui ser llegit per PPRN). Aquest conversor es basa en el següent algorisme (on m és el nombre de nodes, n el nombre d'arcs, k el nombre d'article desitjat, $b^{(1)}$ són les injeccions del problema uniarticle, $\bar{x}^{(1)}$ són les capacitats del problema uniarticle, i $c^{(1)}$ és el vector de costos uniarticle):

Algorisme A6.1. Conversió de problemes DIMACS uniarticle a multiarticle.

```

per i = 1 fins k
    fri := random( $\frac{1}{2}$ , 1)
fi_per

per j = 1 fins m
    r := bj(1)
    per i = 1 fins k
        bj(i) := fri · r
    fi_per
fi_per

per j = 1 fins n
    r :=  $\bar{x}_j^{(1)}$ 
    per i = 1 fins k
         $\bar{x}_j^{(i)}$  := fri · r
    fi_per
fi_per

```

```
per j = 1 fins n
  r := cj(1)
  per i = 1 fins k
    cj(i) := random(0 , r)
  fi_per
fi_per
```

On “random(a,b)” és una funció que retorna un valor aleatori x , $a \leq x \leq b$ (derivada de [71]). Cal observar que les capacitats dels arcs i les injeccions dels nodes del problema multiarticle s’obtenen com a fracció fr_i de les capacitats i injeccions originals de la xarxa uniarticle, on fr_i sempre es troba entre 1/2 i 1. La raó per no permetre valors de fr_i menors que 1/2 és evitar tenir xarxes multiarticle on, o bé les injeccions serien tan petites que no apareixeria cap constricció de capacitat mútua activa a l’òptim, o bé les capacitats dels arcs serien tan properes a 0 que podria no existir una solució factible.

Per cada un dels quatre generadors DIMACS s’han creat vuit instàncies, referenciades com L_j^i , $i=1,\dots,4$, $j=1,\dots,8$. D’aquestes, les quatre primeres corresponen a problemes amb una xarxa de tamany mig i pocs articles, mentre que les quatre darreres són problemes amb xarxes petites i molts articles. En generar els problemes de pocs articles s’han considerat per tots els generadors xarxes amb 1, 4, 8 i 16 articles. Per al cas de molts articles s’ha treballat amb 50, 100, 150 i 200 articles.

A continuació passem a descriure les principals característiques de cada un dels generadors. Tots ells necessiten d’una llavor inicial per generar nombres aleatoris, i aquesta ha estat en tots els casos el valor enter 3141592.

1.a) El generador Rmfgen.

El generador Rmfgen, desenvolupat per Goldfarb i Grigoriadis [32], a partir dels paràmetres d’entrada a , b , c_1 i c_2 crea una xarxa formada per b subxarxes quadrades de mida $a \times a$ (per tant el nombre de nodes total serà $a \times a \times b$). A cada subxarxa els nodes es troben connectats amb els seus veïns (endavant i endarrere). A més, els nodes de cada subxarxa estan connectats a un dels nodes de la següent subxarxa usant una permutació aleatòria de nodes. Hi ha una injecció positiva de flux al node superior esquerre de la primera subxarxa, i el node pou és l’inferior dret de la b -èssima subxarxa. Les capacitats són enters escollits aleatòriament dins el rang (c_1, c_2) , per als arcs que interconnecten dues subxarxes, i $c_2 \times a \times a$, per als arcs de dins d’una subxarxa.

En aquest treball el generador original ha estat ampliat amb dos paràmetres addicionals $cost$ i dem , per tal de produir un cost per cada arc (escollit aleatòriament dins el rang $(0, cost)$) i un flux (escollit dins el rang $(0, dem)$) que serà injectat al node de producció i haurà de sortir pel pou.

La taula 6.1 mostra les característiques dels problemes generats amb Rmfgen. Els primers quatre problemes (L_j^1 $j=1,\dots,4$, amb xarxes de tamany mig i pocs articles) s’han generat usant els paràmetres: $a=16$, $b=8$, $c_1=0$, $c_2=10000$, $cost=1000$, $dem=1000$. Per

la seva banda, per als quatre problemes amb molts articles (L_j^1 $j=5, \dots, 8$) s’han emprat $a=4$, $b=8$, $c_1=0$, $c_2=4000$ (per L_1^1 i L_2^1), $c_2=6000$ (per L_3^1), $c_2=8000$ (per L_3^1), $cost=1000$, $dem=1000$. La primera columna de la taula, “test”, és el nom del problema concret. La columna k denota el nombre d’articles del test. Les columnes “nodes” i “arcs” donen el nombre d’arcs i nodes de la xarxa A , mentre que “files \tilde{A} ” i “columnes \tilde{A} ” representen el nombre total de constriccions i variables de la matriu del problema multiarticle \tilde{A} definida a (2.11). Finalment, la columna “Valor Òptim” indica el valor de la funció de cost lineal al punt òptim.

Taula 6.I. Problemes generats amb Rmfgen.

test	k	nodes	arcs	files \tilde{A}	columnes \tilde{A}	Valor Òptim
L_1^1	1	2048	9472	2048	9472	375675.1
L_2^1	4	2048	9472	17664	47360	2027285.0
L_3^1	8	2048	9472	25856	85248	4506263.3
L_4^1	16	2048	9472	42240	161024	9870432.7
L_5^1	50	128	496	6896	25296	11839382.1
L_6^1	100	128	496	13296	50096	27150952.5
L_7^1	150	128	496	19696	74896	39835825.1
L_8^1	200	128	496	26096	99696	54343948.3

1.b) *El generador Grid-on-torus.*

El generador Grid-on-torus va ser desenvolupat per A.V. Goldberg (1991, Dept. of Computer Science, Stanford University). Genera un problema de transport en una xarxa que recobreix un tor. Requereix cinc paràmetres enters:

- M: el nombre de nodes.
- N: el nombre d’arcs.
- MAXCAP: la capacitat màxima dels arcs (la mínima és sempre 0).
- MAXCOST: el cost lineal màxim per cada arc.
- LLAVOR: valor per generar nombres aleatoris.

La taula 6.II mostra les instàncies generades amb Grid-on-torus. Els problemes amb pocs articles (L_j^2 , $j=1, \dots, 4$) han estat generats usant: $M=1500$, $N=9000$, MAXCAP=10000, MAXCOST=1000, LLAVOR=3141592. La injecció de flux a la xarxa ha estat posteriorment dividida per 10 per tal d’obtenir problemes multiarticle factibles. Per al cas de problemes amb molts articles (L_j^2 , $j=5, \dots, 8$) els paràmetres usats han estat: $M=100$, $N=600$, MAXCAP=10000, MAXCOST=1000, LLAVOR=3141592. Com al cas

Taula 6.II. Problemes generats amb Grid-on-torus.

test	k	nodes	arcs	files \tilde{A}	columnes \tilde{A}	Valor Òptim
$L_1^{(2)}$	1	1500	9000	1500	9000	36896.8
$L_2^{(2)}$	4	1500	9000	15000	45000	187962.0
$L_3^{(2)}$	8	1500	9000	21000	81000	1197048.7
$L_4^{(2)}$	16	1500	9000	33000	153000	5876840.3
$L_5^{(2)}$	50	100	600	5600	30600	5207622.6
$L_6^{(2)}$	100	100	600	10600	60600	12922703.9
$L_7^{(2)}$	150	100	600	15600	90600	22663204.5
$L_8^{(2)}$	200	100	600	20600	120600	36829147.5

previ, la injecció de flux ha estat dividida, en aquest cas per 100. El significat de les columnes de 6.II és el mateix que el de la taula 6.I.

1.c) El generador Gridgraph.

El generador Gridgraph va ser desenvolupat per M.G.C. Resende (1991, AT&T Bell Laboratories). Requereix cinc paràmetres (h , w , MAXCAP, MAXCOST i LLAVOR), i a partir d'aquests genera un graf amb $h \times w + 2$ nodes: un node de producció s , un pou t , i una xarxa rectangular de $h \times w$ nodes. Els arcs van:

- De s als h nodes de la primera columna de la xarxa.
- Dels h nodes de la darrera columna de la xarxa a t .
- Del node (i, j) als nodes $(i + 1, j)$ i $(j + 1, i)$, excepte per a la darrera fila, la qual va de (h, j) a $(h, j + 1)$, i la darrera columna, la qual va de (i, w) a t .

Les capacitats dels arcs és distribueixen uniformement entre 0 i MAXCAP. Els arcs connectats al node de producció s o al pou t no tenen capacitat màxima. Els costos dels arcs es distribueixen uniformement entre 0 i MAXCOST. Els arcs que uneixen s i t amb la xarxa tenen cost zero. Es genera una injecció de flux a s , el qual ha de ser transportat per la xarxa fins arribar a t . Aquest flux per defecte generat per Gridgraph ha hagut de ser reduït per a cada test en particular en convertir el problema en un de multiarticle, per tal de mantenir la factibilitat.

La taula 6.III mostra les instàncies obtingudes amb Gridgraph. Els problemes amb pocs articles ($L_j^{(3)}$, $j=1, \dots, 4$) s'han generat amb els paràmetres: $h=50$, $w=50$, MAXCAP=10000, MAXCOST=1000, LLAVOR=3141592. El flux transportat de s a t ha estat de 15000 unitats per $L_1^{(3)}$ i $L_2^{(3)}$, i 3000 unitats per $L_3^{(3)}$ i $L_4^{(3)}$. Els problemes amb molts articles ($L_j^{(3)}$, $j=5, \dots, 8$), per la seva banda, s'han generat usant: $h=15$, $w=15$, MAXCAP=10000, MAXCOST=1000, LLAVOR=3141592. El flux transportat és en aquest cas de 400 unitats

Taula 6.III. Problemes generats amb Gridgraph.

test	k	nodes	arcs	files \tilde{A}	columnes \tilde{A}	Valor Òptim
$L_1^{(3)}$	1	2502	5000	2502	5000	94212753.2
$L_2^{(3)}$	4	2502	5000	15008	25000	355884986.5
$L_3^{(3)}$	8	2502	5000	25016	45000	128743093.6
$L_4^{(3)}$	16	2502	5000	45032	85000	253615755.8
$L_5^{(3)}$	50	227	450	11800	22950	27853327.9
$L_6^{(3)}$	100	227	450	23150	45450	65144564.0
$L_7^{(3)}$	150	227	450	34500	67950	27066715.2
$L_8^{(3)}$	200	227	450	45850	90450	37964963.7

per $L_5^{(3)}$ i $L_6^{(3)}$, i 125 unitats per $L_7^{(3)}$ i $L_8^{(3)}$. El significat de les columnes de la taula és el mateix que a les taules anteriors.

1.d) *El generador Gridgen.*

El generador Gridgen va ser desenvolupat per Y. Lee and J. Orlin. Aquest crea una xarxa amb un supernode addicional. A més dels arcs que connecten els nodes de la xarxa, hi ha un arc per a cada node de producció al supernode, i del supernode a cada centre de consum per garantir la factibilitat. Aquests arcs tenen capacitats molt grans.

El funcionament del generador és tal com segueix. En primer lloc una xarxa de $n1 \times n2$ nodes és generada. Els nodes es numeren de 1 a $n1 \times n2$, i el supernode serà el node $n1 \times n2 + 1$. Aleshores es generen arcs entre nodes adjacents. Per aquests arcs, l'usuari pot escollir entre obtenir dos arcs (un en cada sentit), o un únic. En aquest darrer cas els arcs es generen en sentits alternatius. Llavors els arcs entre el supernode i els nodes de producció/consum són incorporats com abans s'ha esmentat. Si el nombre d'arcs requerit no s'ha assolit, aleshores s'afegeixen arcs addicionals entre parells de nodes escollits aleatòriament. No es comprova que hi hagi arcs múltiples. Tanmateix, segur que no hi ha arcs que apunten cap al seu node d'origen. Els nodes de producció i consum són escollits de forma aleatòria dins la xarxa, igual que es fa amb les injeccions.

Els paràmetres d'entrada del generador són:

- Arcs de doble sentit: 1 si es vol tenir dos arcs, un en cada sentit, i 0 altrament.
- Llabor: enter positiu per generació aleatòria de nombres.
- Nombre de nodes: el nombre de nodes generats pot ser lleugerament diferent per aconseguir una xarxa escaient.
- Amplada de la xarxa.
- Nombre de nodes de producció i pous.
- Grau promig: per comptar els arcs que van i venen del supernode.
- Flux total.

- Distribució dels costos dels arcs: 1 per uniforme, 2 per exponencial. Si s'escull 1, aleshores s'han de subministrar dos paràmetres addicionals: el cost mínim i el màxim. Si s'escull la distribució exponencial llavors s'ha de donar el valor lambda de la distribució.

- Distribució de les capacitats dels arcs: 1 per uniforme, 2 per exponencial. Si s'escull 1, aleshores s'han subministrar la capacitat mínima i la màxima. Si s'escull la distribució exponencial de nou s'ha de donar el valor lambda de la distribució.

La taula 6.IV mostra les instàncies generades amb Gridgen. Els quatre primers problemes ($L_j^{(4)}$ $j=1, \dots, 4$), de pocs articles, han estat generats amb els següent paràmetres: arcs de doble sentit=0, llavor=3141592, #nodes=1000, amplada de la xarxa=75, #nodes de producció=50, #pous=30, grau promig=8, flux total=50000, distribució dels costos dels arcs=1, cost mínim=0, cost màxim=1000, distribució de les capacitats dels arcs=1, capacitat mínima=0, capacitat màxima=10000. Els paràmetres per als problemes amb molts articles ($L_j^{(4)}$ $j=5, \dots, 8$) han estat: arcs de doble sentit=0, llavor=3141592, #nodes=100, amplada de la xarxa=20, #nodes de producció=10, #pous=10, grau promig=6, flux total=200, distribució dels costos dels arcs=1, cost mínim=0, cost màxim=1000, distribució de les capacitats dels arcs=1, capacitat mínima=0, capacitat màxima=20000. El significat de les columnes de la taula és el mateix que en taules anteriors.

Taula 6.IV. Problemes generats amb Gridgen.

test	k	nodes	arcs	files \tilde{A}	columnes \tilde{A}	Valor Òptim
$L_1^{(4)}$	1	976	7808	976	7808	5541980.3
$L_2^{(4)}$	4	976	7808	11712	39040	23223474.9
$L_3^{(4)}$	8	976	7808	15616	70272	61792270.7
$L_4^{(4)}$	16	976	7808	23424	132736	165808232.3
$L_5^{(4)}$	50	101	606	5656	30906	1409470.3
$L_6^{(4)}$	100	101	606	10706	61206	2940217.3
$L_7^{(4)}$	150	101	606	15756	91506	4614971.4
$L_8^{(4)}$	200	101	606	20806	121806	6440385.5

2. *Generador Mnetgen.*

El generador Mnetgen [4], a diferència dels anteriors, produeix directament un problema multiarticle (també sense constriccions a banda). Aquest generador requereix els següents paràmetres per tal de generar un model:

- LLAVOR: per generació de nombre aleatoris.
- NODES: nombre de nodes de la xarxa.
- ARCS: nombre aproximat d'arcs multiarticle (és a dir tenint en compte les k repeticions per a cada article).
- NSORC: nombre de nodes de producció (purs —només emanen arcs d'ells— i de transport —arriben i surten arcs).

Taula 6.v. Paràmetres usats en la generació de problemes amb Mnetgen.

test	NODES	ARCS	NSORC	NSINK	NCOMM	NTSORC	NTSINK	LCHAIN
$L_1^{(6)}$	50	120	20	20	4	08	08	06
$L_2^{(6)}$	300	5360	100	100	8	60	60	64
$L_3^{(6)}$	300	10720	100	100	16	60	60	64
$L_4^{(6)}$	400	12800	100	100	16	60	60	64
$L_5^{(6)}$	300	20770	100	100	31	60	60	64
$L_6^{(6)}$	300	32160	100	100	48	60	60	64

Taula 6.vi. Problemes generats amb Mnetgen.

test	k	nodes	arcs	files \tilde{A}	columnes \tilde{A}	Valor Òptim
$L_1^{(5)}$	4	50	104	304	520	378009.3
$L_2^{(5)}$	8	300	671	3071	6039	9890447.9
$L_3^{(5)}$	16	300	681	5481	11577	18700864.0
$L_4^{(5)}$	16	400	821	7221	13957	23697345.2
$L_5^{(5)}$	31	300	673	9973	21536	36907046.0
$L_6^{(5)}$	48	300	683	15083	33467	54636652.8

- NSINK: nombre de pous (purs i de transport).
- NCOMM: nombre d'articles.
- NTSORC: nombre de nodes de producció de transport.
- NTSINK: nombre de pous de transport.
- LCHAIN: longitud de la cadena més llarga dins l'esquelet de la xarxa.
- MINCST: fita inferior dels costos dels arcs.
- MAXCST: fita superior dels costos dels arcs.
- MINSUP: fita inferior de la injecció total de flux.
- MAXSUP: fita superior de la injecció total de flux.
- MINBND: fita inferior de les capacitats de cada arc.
- MAXBND: fita superior de les capacitats de cada arc.
- MINCAP: fita inferior de la capacitat mútua.
- MAXCAP: fita superior de la capacitat mútua.
- PHICST: percentatge d'arcs amb cost elevat.
- PBND: percentatge dels arcs amb capacitat.
- PCAP: percentatge dels arcs amb capacitat mútua.

S'han generat un total de sis problemes amb Mnetgen, denotats per $L_j^{(6)}$ $j=1, \dots, 6$. Hi ha uns paràmetres que han estat comuns a tots els problemes generats. Aquests són: LLAVOR=3141592, MINCST=1, MAXCST=100, MINSUP=100, MAXSUP=1000, MINBND=0, MAXBND=50, MINCAP=20, MAXCAP=200, PHICST=10, PBND=75,

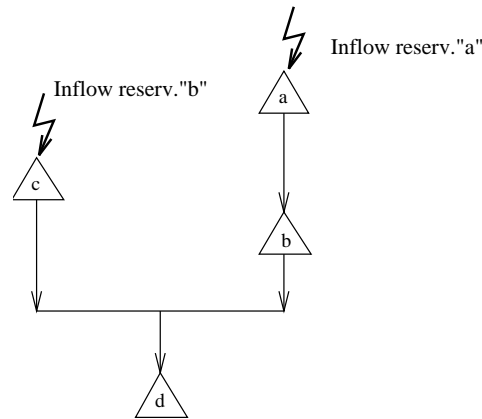


Figura 6.1. Exemple d'un sistema amb quatre embassaments.

PCAP=60. D'altres paràmetres han estat variables, i són presentats a la taula 6.v. Les característiques dels sis problemes generats es mostren a la taula 6.vi. El significat de les columnes és el mateix que en taules anteriors.

6.2.1.2 Problemes lineals de coordinació hidrotèrmica.

El segon tipus de problemes lineals provenen del camp de la coordinació hidrotèrmica a llarg i a curt termini, segons els models proposats a [62] i [39] respectivament. A continuació es farà una molt breu descripció del tipus de problema solucionat. Per a més detalls, el lector interessat pot consultar els articles prèviament referenciats.

1. Problemes de coordinació hidrotèrmica a llarg termini.

La solució de l'optimització de la generació hidrotèrmica a llarg termini en un companyia elèctrica, indica com distribuir al llarg d'un període de temps la generació hidràulica a cada embassament del sistema d'embassaments per tal de minimitzar el cost esperat de producció tèrmica. A la generació a llarg termini, la disponibilitat de les unitats tèrmiques, la demanda d'electricitat i les aportacions d'aigua als embassaments no són deterministes i només es disposa de les seves distribucions de probabilitat. La demanda i la disponibilitat de les plantes pot ser modelitzada a través de funcions de cost probabilista de producció respecte de la generació hidràulica. El problema a solucionar és, doncs, minimitzar la generació a cada interval del nostre període d'estudi, tenint en compte que les aportacions d'aigua als nostres embassaments són estocàstiques.

És important realitzar l'optimització tenint en compte el conjunt sencer d'aportacions probabilistes d'aigua. En aquest sentit, es pot usar un model que consideri diversos tipus d'aigua en funció de la seva probabilitat. Això pot ser fet aproximant les funcions de densitat de probabilitat de les aportacions, de cada embassament i a cada interval, per una funció de densitat de probabilitat de $k-1$ blocs d'àrees p_1, \dots, p_k , $\sum_{i=1}^k p_i = 1$. Els articles del problema són, doncs, les quantitat d'aigua associades a cada probabilitat. El primer article és l'aigua determinista (va de l'origen de la funció de densitat de probabilitat fins la posició de l'eix on comença el primer bloc rectangular), el segon article és l'aigua associada amb

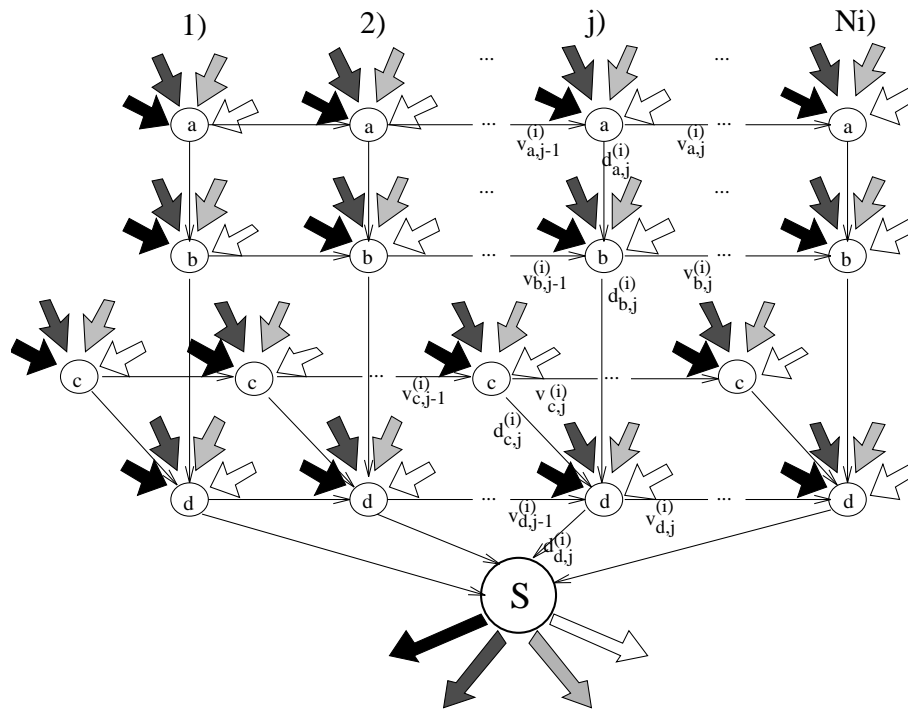


Figura 6.2. Xarxa replicada multiarticle amb els fluxos d'aigua.

el primer bloc, i així fins arribar al k -èssim article associat al bloc d'àrea p_{k-1} . Es pot derivar una expressió del cost esperat de producció tèrmica a cada interval en funció de les quantitats dels diferents tipus d'aigua dedicats a emmagatzemament, generació, bombeig o vessament per cada embassament, i la funció objectiu a minimitzar serà la suma del cost esperat de producció tèrmica per tots els articles.

Donat un sistema d'embassaments com el de la Fig. 6.1, una xarxa replicada d'ella, com mostra la Fig. 6.2, és usada per representar les constriccions de balanç d'aigua per tots els intervals. Els volums màxims dels embassaments i les descàrregues màximes s'imposen com a constriccions de capacitat mútua dels fluxos multiarticle dels arcs de volums (horitzontals) i arcs de descàrrega (verticals i inclinats).

La funció objectiu d'aquest problema (la suma del cost esperat de producció tèrmica per tots els intervals) és no lineal, i en aquest cas s'ha considerat una versió linealitzada, la qual penalitza —amb costos positius alts— els arcs de vessament, i incentiva —amb costos negatius— les descàrregues, ponderant cada article amb la seva disponibilitat en funció de les probabilitats p_1, \dots, p_k . Els volums no intervenen en la funció objectiu lineal definida. Constriccions a banda addicionals poden ser imposades per tenir en compte limitacions en la generació i irrigació.

La taula 6.VII mostra les característiques dels cinc problemes de coordinació hidrotèrmica a llarg termini considerats, els qual seran denotats per $L_j^{(6)}$, $j=1, \dots, 5$. Pot comprovar-se com el nombre d'articles en tots els casos és quatre, donat que amb aquest

nombre de blocs la funció de densitat per blocs ja aproxima suficientment bé la funció de densitat original. Hi ha una columna afegida respecte taules anteriors, denotada amb “#c.b.”, la qual indica el nombre de constriccions a banda del problema.

Taula 6.VII. Problemes lineals de coordinació hidrotèrmica a llarg termini.

test	k	#c.b.	nodes	arcs	files \tilde{A}	columnes \tilde{A}	Valor Òptim
$L_1^{(6)}$	4	2	37	117	267	587	-1282480.0
$L_2^{(6)}$	4	12	37	153	313	777	-571304.7
$L_3^{(6)}$	4	18	25	98	216	508	-400688.8
$L_4^{(6)}$	4	3	99	315	714	1578	-995067.1
$L_5^{(6)}$	4	3	685	2141	4884	10708	71115082.6

2. Problemes lineals de coordinació hidrotèrmica a curt termini.

La solució de la coordinació hidrotèrmica a curt termini determina com distribuir la generació hidràulica (sense cost) a cada embassament d'un sistema d'embassaments, i com fixar la generació de les unitats tèrmiques que han d'operar durant un període de temps curt, de forma que el consum de combustible durant aquest període és minimitzat.

El model de xarxa usualment emprat a la coordinació hidrotèrmica de curt termini ha estat ampliat per afegir les unitat tèrmiques de forma acoblada [39], imposant constriccions de càrrega i de reserva rodant sobre la generació hidràulica i tèrmica de cada interval, i minimitzant directament els costos de producció tèrmica, sense desacoblar el problema en subproblemes hidràulics i tèrmics. Quan s'afegeixen constriccions de forma que la generació hidràulica més la tèrmica han de satisfer la demanda i verificar els requeriments de reserva rodant a cada interval, les tècniques de xarxes no poden ser aplicades perquè la generació hidràulica és una funció no lineal. Tanmateix, si aquestes constriccions són linealitzades, poden ser usats algorismes especialitzats per optimitzar problemes de fluxos en xarxes amb constriccions a banda. La generació hidràulica ha estat, doncs, linealitzada en funció de les variables de la xarxa (volums inicials i finals, i descàrregues, a cada embassament) per tal de disposar de constriccions a banda lineals.

La base del model usat és la següent. Sigui P_j la potència de sortida de de la j -èssima unitat tèrmica, i siguin \bar{P}_j i \underline{P}_j els seus límits operatius superior i inferior: $\underline{P}_j \leq P_j \leq \bar{P}_j$. La reserva rodant incremental (ISR) r_{Ij} de la unitat “ j ” és el total de potència amb la qual la generació actual P_j pot ser ampliada en un interval de temps donat. \bar{r}_{Ij} és la ISR màxima possible. De forma anàloga, la reserva rodant decremental (DSR) r_{Dj} és el total de potència amb la qual es pot disminuir la potència actual P_j . El seu valor màxim serà \bar{r}_{Dj} . La ISR r_{Ij} i la DSR r_{Dj} de la j -èssima unitat verifiquen que $r_{Ij} = \min\{\bar{r}_{Ij}, \bar{P}_j - P_j\}$

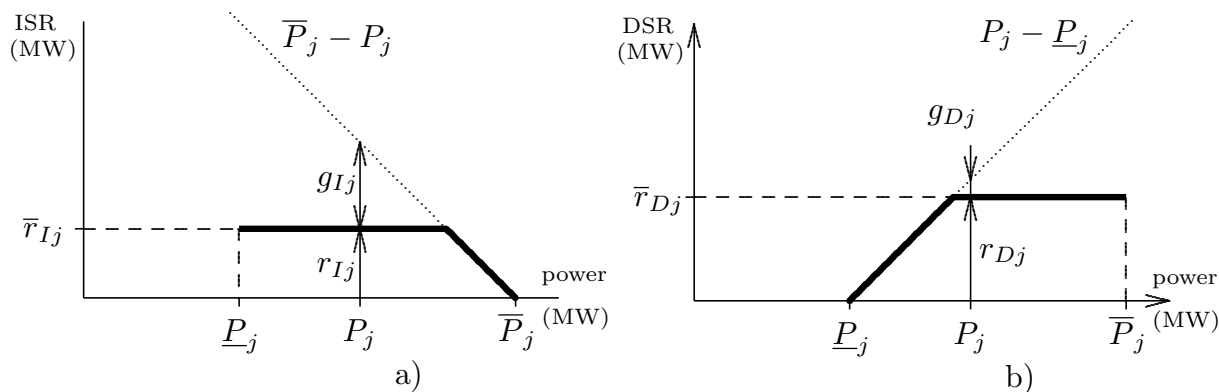


Figura 6.3. Representació de les funcions ISR i DSR.
 a) Funció reserva rodant incremental (ISR) per a la j -èssima unitat tèrmica.
 b) Funció reserva rodant decremental (DSR) per a la j -èssima unitat tèrmica.

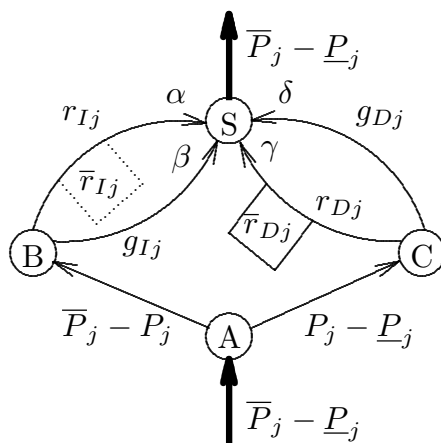


Figura 6.4. Representació de la xarxa de les variables de generació tèrmica.

i $r_{Dj} = \min\{\bar{r}_{Dj}, P_j - \underline{P}_j\}$, i són representades per les línies gruixudes de Fig. 6.3a) i Fig. 6.3b).

A la potència P_j es té una ISR r_{Ij} i una DSR r_{Dj} , i hi ha un marge de potència $g_{Ij} \geq 0$ des de la ISR r_{Ij} fins $\bar{P}_j - P_j$ de forma que $r_{Ij} + g_{Ij} = \bar{P}_j - P_j$, i, de forma anàloga, hi ha un marge de potència $g_{Dj} \geq 0$ entre la DSR r_{Dj} i $P_j - \underline{P}_j$ tal que $r_{Dj} + g_{Dj} = P_j - \underline{P}_j$. La generació d'una unitat tèrmica, les seves ISR i DSR, els marges de potència associats, i els seus límits operatius poden ser modelitzats en forma de xarxa, tal i com mostra la Fig. 6.4. S'ha d'imposar una fita superior \bar{r}_{Ij} a l'arc α per prevenir que la reserva augmenti per sobre del seu límit. Per tal d'assegurar que els fluxos dels arcs α i β són com les variables de la Fig. 6.3a), pot afegir-se un cost $w_{\beta\delta}$ positiu i petit al flux de l'arc β , mentre que el cost de l'arc α serà zero. De forma anàloga, a γ se li associarà un cost zero, mentre que δ

tindrà de nou un cost $w_{\beta\delta}$ petit i positiu, com ocorria amb β . El flux $P_j - \underline{P}_j$ que va del node A al node C s'associa amb el cost de la generació que cal ser minimitzat.

Aquest model descrit per un generador pot ser ampliat per tal de considerar totes les unitats tèrmiques assignades a cada interval “i”. Una única xarxa representarà la generació, ISR, DSR i marges de potència de totes les unitats assignades. Les xarxes de les unitats poden compartir el node pou S. La xarxa descrita correspondria a l'interval “i”. Les constriccions a banda addicionals vindrien donades per una constricció de càrrega i un requeriment sobre les ISR i DSR, per cada interval.

Taula 6.VIII. Problemes lineals de coordinació hidrotèrmica a curt termini.

test	k	#c.b.	nodes	arcs	files \tilde{A}	columnes \tilde{A}	Valor Òptim
$L_1^{(7)}$	1	36	85	348	121	384	-16.9344
$L_2^{(7)}$	1	504	1177	4872	1681	5376	-54.43203

El problema resultant és un problema de fluxos uniarticle amb constriccions a banda. La funció a minimitzar és no lineal (pot trobar-se una completa descripció d'ella a [39]), i en aquest cas s'ha considerat una linealització de la mateixa.

La taula 6.VIII mostra les característiques dels dos problemes test usats amb. El significat de les columnes és el mateix que en taules anteriors. Als dos casos s'obté un problema uniarticle.

6.2.2 Problemes test no lineals.

Pel que fa al cas no lineal, s'han usat tres tipus de problemes. El primer tipus són problemes amb funcions objectiu artificials, el segon correspon a problemes de coordinació hidrotèrmica (a llarg i curt termini), i el darrer cas és un problema d'assignació de trànsit. Passem tot seguit a detallar-los.

6.2.2.1 Problemes artificials.

Aquest primer conjunt de problemes no lineals considera tres tipus de funcions objectiu. Les dos primeres són simples funcions convexes definides com:

$$f_1(x_1, x_2, \dots, x_k) = \sum_{i=1}^k \sum_{j=1}^n x_{ij}^2 \quad (6.1)$$

$$f_2(x_1, x_2, \dots, x_k) = \sum_{i=1}^k \sum_{j=1}^n x_{ij}^4 \quad (6.2)$$

essent x_{ij} el flux del j -èssim arc de l'article i . La tercera funció objectiu ha estat derivada de la presentada a [74], i es defineix com:

$$f_3(x_1, x_2, \dots, x_k) = \sum_{i=1}^k F_i(x_i) \tag{6.3}$$

on

$$F_i(x_i) = \frac{1}{c_1} \sum_{j=1}^n x_{ij}^2 + \frac{1}{c_2} \left(\sum_{j=1}^{n-1} \sqrt{1 + x_{ij}^2 + (x_{ij} - x_{i,j+1})^2} + \frac{1}{c_3} \left(10 + \sum_{j=1}^n (-1)^j x_{ij} \right)^4 \right) \tag{6.4}$$

on $c_1, c_2, c_3 \in \mathbb{R}$ (als tests realitzats $c_1 = 1000$, $c_2 = 1000$ i $c_3 = 1200$). Malgrat la seva simplicitat, aquestes tres funcions objectiu tenen solucions amb un nombre molt elevat de variables superbàsiques, el qual incrementa considerablement el cost de l'execució. Aquestes funcions objectiu han estat usades amb les xarxes obtingudes per a problemes no lineals de generació hidrotèrmica a llarg termini, les dimensions de les quals són mostrades a la taula 6.IX. Els tres problemes artificials seran referits com a N_j^i , $i=1, \dots, 3$, on el supraíndex i fa referència al tipus de funció objectiu usada ((6.1), (6.2) o (6.3) respectivament).

6.2.2.2 Problemes no lineals de coordinació hidrotèrmica.

S'han considerat els models de coordinació hidrotèrmica a llarg i curt termini exposats per al cas lineal a §6.2.1.2. En aquest cas, però, s'han usat les funcions objectiu no lineals reals, en comptes de linealitzacions de les mateixes. La taula 6.IX presenta les característiques dels problemes a llarg termini (que són les mateixes que les dels problemes artificials, com abans s'ha exposat). Per la seva banda, la taula 6.X mostra les dimensions del problemes a curt termini. El significat de les columnes és el mateix que a les taules prèvies per a problemes lineals. No es presenta el valor òptim de la funció objectiu ja que, per als problemes no lineals, aquest serà mostrat a les taules on es presentin els resultats de les execucions realitzades.

Taula 6.IX. Problemes no lineals artificials i de coordinació hidrotèrmica a llarg termini.

test	k	#c.b.	nodes	arcs	files \tilde{A}	columnes \tilde{A}
N_1^j ^(a)	4	12	37	153	313	777
N_2^j ^(a)	4	3	99	315	714	1578
N_3^j ^(a)	4	3	685	2141	4884	10708

^(a) $j = 1, \dots, 4$

Taula 6.x. Problemes no lineals de coordinació hidrotèrmica a curt termini.

test	k	#c.b.	nodes	arcs	files \tilde{A}	columnes \tilde{A}
$N_1^{(5)}$	1	528	1345	4416	1873	4944
$N_2^{(5)}$	1	840	1975	6048	2815	6888
$N_3^{(5)}$	1	840	2479	8064	3319	8904
$N_4^{(5)}$	1	1848	4741	15600	2289	17448

6.2.2.3 Problema d'assignació de trànsit.

El problema d'assignació de trànsit estàtic en equilibri d'usuari amb demanda inelàstica i funcions de cost separables va ser originalment formulat per Beckman et al. [7]. Donats \mathcal{I} orígens i \mathcal{J} destins, el problema d'assignació estàtica en equilibri consisteix en trobar una distribució de fluxos entre els camins p_{ij} que uneixen l'origen $i \in \mathcal{I}$ amb la destinació $j \in \mathcal{J}$ de forma que tots els camins amb flux positiu tinguin un cost igual a u_{ij}^* , i els camins sense flux tinguin un cost $(u_{ij})_p \geq u_{ij}^*$. Aquest problema pot ser formulat com a problema multiarticle, considerant que cada article k_{ij} és el flux de trànsit que surt de $i \in \mathcal{I}$ i arriba a $j \in \mathcal{J}$. Tanmateix, això significaria tenir un gran nombre d'articles ($k = |\mathcal{I}| \times |\mathcal{J}|$). Una altra possibilitat consistiria en agrupar els fluxos de trànsit per orígens o per destins, tenint aleshores $k = |\mathcal{I}|$ o $k = |\mathcal{J}|$ segons l'elecció, amb una gran reducció en el nombre d'articles i variables. A l'execució realitzada l'agrupament va ser fet per orígens.

Taula 6.xi. Problema d'assignació de trànsit.

test	k	#c.b.	nodes	arcs	files \tilde{A}	columnes \tilde{A}
$N_1^{(6)}$	16	0	182	309	3221	5253

La funció objectiu a minimitar és la suma per a cada arc a de la xarxa d'una funció de cost definida en termes del volum de trànsit v_a . Aquest funció pot ser escrita com:

$$f_t(v_1, v_2, \dots, v_k) = \sum_{\forall a} \int_0^{v_a} C_a(v) dv \quad (6.5)$$

on $C_a(v)$ és el valor instantani de la funció de cost per un volum donat, i $v_a = \sum_{\forall i \in \mathcal{K}} v_{i_a}$ (essent \mathcal{K} el conjunt d'articles). Resolent la funció integral, (6.5) pot ser escrit com:

$$f_t(v_1, v_2, \dots, v_k) = \sum_{\forall a} c_{1_a} + c_{2_a} l_a \left(1 + \alpha_a \left(\frac{v_a + c_{3_a}}{c_{4_a}} \right)^{\beta_a} \right) \quad (6.6)$$

essent l_a la longitud de la línia, i c_{j_a} ($j=1, \dots, 4$), α_a i β_a alguns coeficients dependents de la línia. L'expressió (6.6) ha estat usada com a funció objectiu del problema d'assignació de trànsit.

Cal indicar que l'aproximació aquí considerada per solucionar aquest problema és lleugerament diferent de les usualment emprades i descrites a la literatura d'aquest camp [72]. A més d'algunes heurístiques, les tècniques de Frank-Wolfe i de la Descomposició Simplicial han estat usualment usades. El principal avantatge d'aquests algorismes és que permeten tractar xarxes molt grans, l'estructura de la qual és molt explotada. Tanmateix, presenten una convergència molt lenta i la cerca de solucions molt acurades esdevé prohibitiva. La formulació del problema i l'algorisme emprat solventen aquesta dificultat, encara que no poden tractar xarxes extremadament grans (donat que per cada arc s'han de considerar k volums, un per article, obtenint un nombre molt elevat de variables).

Només un problema ha estat usat amb aquesta funció objectiu. Aquest problema és un cas real corresponent a una subxarxa de la ciutat de Barcelona (zona de Ciutat Vella). Les dimensions d'aquest problema es mostren a la taula 6.XI.

6.3 Resultats computacionals.

Igual que es va fer en presentar els problemes test usats, es farà una clara distinció entre els problemes lineals i els no lineals a l'hora de presentar els resultats computacionals. Totes les execucions presentades s'han fet en una estació de treball SunSparc 10/41 sota UNIX (amb una única CPU), d'arquitectura RISC, amb 40MHz de rellotge, aproximadament uns 100 Mips i 20 Mflops de CPU, i 64 Mbytes de memòria (32 reals i 32 mapejats a disc). Passem tot seguit a presentar els resultats obtinguts.

6.3.1 Resultats computacionals amb els problemes test lineals.

Els problemes lineals L_j^i presentats a §6.2.1 han estat executats amb quatre paquets diferents. El primer d'ells és el codi PPRN presentat en aquest treball, el qual ha estat comparat amb la versió 5.3 de MINOS [59] (un paquet general d'optimització lineal i no lineal), MCNF85 [48] (un codi especialitzat per a problemes multiarticle sense constriccions a banda, basat també en un particionament primal, i desenvolupat per J. Kennington), i LoQo [77] (una implementació eficient d'un mètode primal-dual de punt interior). Per als quatre paquets s'han usat els valors per defecte de tots els paràmetres que podien ser ajustats (excepte pel paquet PPRN, on s'ha usat l'opció de trobar el punt òptim a la fase 0 als tests L_j^i $i=1, \dots, 4$, $j=1, \dots, 8$, en comptes de l'opció per defecte consistent en trobar un punt factible). El valor de la funció objectiu lineal al punt òptim assolit ha estat el mateix per tots els paquets (hi ha petites variacions, però són menyspreables) i coincideix amb el presentat prèviament a les taules 6.I–6.IV i 6.VI–6.VIII.

Les taules 6.XII–6.XVII mostren els resultats obtinguts amb els quatre paquets per a cada un dels grups de problemes lineals (problemes generats amb Rmfgn, problemes generats amb Grid-on-torus, problemes generats amb Gridgraph, problemes generats amb Gridgen, problemes generats amb Mnetgen, i problemes de coordinació hidrotèrmica a llarg i curt termini). Cada taula es troba dividida en dos: la primera part presenta els resultats

amb PPRN, i la segona amb la resta de paquets. La informació presentada per a PPRN és: nom del problema (columna “test”), nombre d’iteracions a les fases 0, 1 i 2 (columnes “It.0”, “It.1” i “It.2”), nombre de constriccions actives a l’òptim (columna “ $|\mathcal{A}|$ ”, valor que ens indica la dimensió de la matriu de treball Q al punt òptim), temps de CPU (en milisegons) per iteració (en promig), comptabilitzant només les iteracions de les fases 1 i 2 i el seu temps associat (columna “T.It”), i temps de CPU en segons requerit per assolir la solució òptima (columna “CPU sec.”). Per a MINOS es presenta la mateixa informació excloent la columna “It.0”. Per a MCNF85 i LoQo, a més del temps de CPU (“CPU sec.”), la columna “It.” dóna el nombre total d’iteracions requerit, i “T.It” presenta el quocient dels dos valors (és a dir, dóna el temps de CPU per iteració).

Cal indicar que a les execucions realitzades s’han produït certs resultats anòmals. Aquests es presenten codificats a les taules (amb una lletra entre parèntesi, per ex. ^(a)), i el significat de cadascun d’ells és el següent:

- ^(a): Problema amb massa constriccions per ser executat amb aquest paquet.
- ^(b): S’ha produït un error durant l’execució i el programa ha abortat.
- ^(c): Problema no executat amb aquest paquet, donat que el temps d’execució seria excessiu.
- ^(d): Els problemes uniarticle no han estat executats amb MCNF85.
- ^(e): Error de factibilitat: no s’ha pogut trobar un punt factible.
- ^(f): El paquet MCNF85 no pot solucionar problemes amb constriccions a banda.
- ^(g): No hi ha prou memòria per executar aquest model amb aquest paquet.
- ^(h): S’han hagut de relaxar les condicions d’aturada i detecció d’òptim per evitar problemes de convergència. El punt obtingut és gairebé òptim.

Així mateix, pot comprovar-se com a les execucions dels problemes amb només un article i cap constricció a banda només intervé la fase 0 del paquet PPRN (excepte al cas $L_1^{(2)}$, on es realitza una darrera i més acurada cerca de l’òptim a la fase 2). En aquests casos s’estarà comparant l’algorisme especialitzat de fluxos uniarticle de la fase 0, en comptes de la tècnica del particionament primal, amb la resta de paquets.

Taula 6.XII. Resultats obtinguts amb els problemes Rmfgn.

test	PPRN					
	It.0	It.1	It.2	$ \mathcal{A} $	T.It	CPU sec.
$L_1^{(1)}$	5689	0	0	0	—	6.5
$L_2^{(1)}$	12103	13	818	2	102.12	98.7
$L_3^{(1)}$	24486	389	4510	4	144.91	737.9
$L_4^{(1)}$	48928	6317	21024	6	248.05	6838.7
$L_5^{(1)}$	7962	2219	5861	25	33.99	275.1
$L_6^{(1)}$	15924	5856	32175	53	72.20	2747.0
$L_7^{(1)}$	23729	8879	60974	57	115.49	8069.0
$L_8^{(1)}$	32188	11219	88548	57	154.49	15415.3

Taula 6.XII.(cont.) Resultats obtinguts amb els problemes Rmfgn.

test	MINOS				MCNF85			LoQo		
	It.1	It.2	T.It	CPU sec.	It.	T.It	CPU sec.	It.	T.It	CPU sec.
L ₁ ¹⁾	0	4084	27.57	112.6	(d)			20	24982.5	499.6
L ₂ ¹⁾	5	19864	182.62	3628.5	7809	75.36	588.5	(g)		
L ₃ ¹⁾	9	47769	317.04	15147.7	16452	108.08	1778.2	(g)		
L ₄ ¹⁾	(a)				31178	181.25	5651.3	(g)		
L ₅ ¹⁾	6669	22077	91.81	2639.4	8719	45.71	398.6	36	94522.2	3402.8
L ₆ ¹⁾	20829	123619	205.37	29665.3	36717	109.90	4305.5	(g)		
L ₇ ¹⁾	(b)				55817	202.78	11319.0	(g)		
L ₈ ¹⁾	(c)				82306	321.72	26479.9	(g)		

Taula 6.XIII. Resultats obtinguts amb els problemes Grid-on-torus.

test	PPRN					
	It.0	It.1	It.2	$ \mathcal{A} $	T.It	CPU sec.
L ₁ ²⁾	6746	0	74	0	25.68	6.9
L ₂ ²⁾	15464	556	8320	25	59.05	537.9
L ₃ ²⁾	31241	3840	40630	58	110.96	4962.2
L ₄ ²⁾	60086	12080	170041	178	205.45	37470.5
L ₅ ²⁾	7348	1313	3660	46	34.02	169.2
L ₆ ²⁾	15053	5091	19170	96	66.99	1625.4
L ₇ ²⁾	22345	12590	59436	168	105.59	7605.5
L ₈ ²⁾	29930	31574	127657	230	139.53	22218.4

Taula 6.XIII.(cont.) Resultats obtinguts amb els problemes Grid-on-torus.

test	MINOS				MCNF85			LoQo		
	It.1	It.2	T.It	CPU sec.	It.	T.It	CPU sec.	It.	T.It	CPU sec.
L ₁ ²⁾	72	8007	19.84	160.3	(d)			25	16416.0	410.4
L ₂ ²⁾	312	68429	154.12	10594.5	10358	66.06	684.3	(g)		
L ₃ ²⁾	(b)				43226	111.80	4833.0	(g)		
L ₄ ²⁾	(c)				182343	188.56	34383.0	(g)		
L ₅ ²⁾	3163	12932	87.15	1402.8	9777	47.75	466.9	49	106348.9	5211.1
L ₆ ²⁾	38924	75696	157.78	18085.3	26345	119.02	3135.8	(g)		
L ₇ ²⁾	144062	253256	264.47	105082.5	74806	211.69	15836.2	(g)		
L ₈ ²⁾	(c)				246426	332.36	81903.5	(g)		

Taula 6.xiv. Resultats obtinguts amb els problemes Gridgraph.

test	PPRN					
	It.0	It.1	It.2	$ \mathcal{A} $	T.It	CPU sec.
$L_1^{(3)}$	4178	0	0	0	—	4.4
$L_2^{(3)}$	16590	2544	9429	235	66.0	807.7
$L_3^{(3)}$	26161	2500	8620	110	124.24	1409.2
$L_4^{(3)}$	51537	12221	49034	224	229.9	14139.8
$L_5^{(3)}$	8735	1645	5497	24	51.09	364.9
$L_6^{(3)}$	17425	6549	33218	61	107.12	4264.1
$L_7^{(3)}$	25693	6052	22811	27	155.21	4480.5
$L_8^{(3)}$	53524	10392	43132	40	219.15	11736.8

Taula 6.xiv.(cont.) Resultats obtinguts amb els problemes Gridgraph.

test	MINOS				MCNF85			LoQo		
	It.1	It.2	T.It	CPU sec.	It.	T.It	CPU sec.	It.	T.It	CPU sec.
$L_1^{(3)}$	820	2874	26.39	97.5	(d)			32	931.2	29.8
$L_2^{(3)}$	12561	44814	178.22	10225.6	22280	61.52	1370.7	37	23624.3	874.1
$L_3^{(3)}$	(b)				25426	83.93	2134.1	(g)		
$L_4^{(3)}$	(c)				91939	159.99	14709.9	(g)		
$L_5^{(3)}$	3869	16580	155.15	3172.8	12533	42.59	533.9	39	81548.7	3180.4
$L_6^{(3)}$	(b)				42553	126.45	5381.2	(g)		
$L_7^{(3)}$	(a)				37013	165.96	6142.8	(g)		
$L_8^{(3)}$	(a)				59798	325.39	19458.0	(g)		

Taula 6.xv. Resultats obtinguts amb els problemes Gridgen.

test	PPRN					
	It.0	It.1	It.2	$ \mathcal{A} $	T.It	CPU sec.
$L_1^{(4)}$	7576	0	0	0	—	5.7
$L_2^{(4)}$	33544	625	5845	41	47.79	334.5
$L_3^{(4)}$	69242	3011	35263	108	88.11	3424.7
$L_4^{(4)}$	137543	11892	248422	264	157.0	40974.1
$L_5^{(4)}$	15113	110	592	1	54.13	39.4
$L_6^{(4)}$	30825	498	2046	3	72.72	192.2
$L_7^{(4)}$	44646	831	2999	5	104.43	415.8
$L_8^{(4)}$	59639	1528	7504	9	139.50	1273.4

Taula 6.XV.(cont.) Resultats obtinguts amb els problemes Gridgen.

test	MINOS				MCNF85			LoQo		
	It.1	It.2	T.It	CPU sec.	It.	T.It	CPU sec.	It.	T.It	CPU sec.
L ₁ ⁽⁴⁾	193	4246	15.85	70.4	(d)			22	16459.0	362.1
L ₂ ⁽⁴⁾	7330	68026	119.66	9017.3	(e)			(g)		
L ₃ ⁽⁴⁾	57801	870882	221.64	205836.0	(e)			(g)		
L ₄ ⁽⁴⁾	(c)				(e)			(g)		
L ₅ ⁽⁴⁾	1617	9779	80.57	918.2	(e)			(g)		
L ₆ ⁽⁴⁾	5336	31486	166.95	6147.6	(e)			(g)		
L ₇ ⁽⁴⁾	8862	46991	272.79	15236.6	(e)			(g)		
L ₈ ⁽⁴⁾	(c)				(e)			(g)		

Taula 6.XVI. Resultats obtinguts amb els problemes Mnetgen.

test	PPRN					
	It.0	It.1	It.2	$ \mathcal{A} $	T.It	CPU sec.
L ₁ ⁽⁵⁾	310	19	21	4	3.75	0.3
L ₂ ⁽⁵⁾	4094	288	1190	18	13.53	21.7
L ₃ ⁽⁵⁾	11019	299	1514	17	24.26	46.5
L ₄ ⁽⁵⁾	14190	497	2386	17	32.25	95.0
L ₅ ⁽⁵⁾	21559	525	5413	19	48.66	293.2
L ₆ ⁽⁵⁾	34521	695	5930	13	78.79	526.0

Taula 6.XVI.(cont.) Resultats obtinguts amb els problemes Mnetgen.

test	MINOS				MCNF85			LoQo		
	It.1	It.2	T.It	CPU sec.	It.	T.It	CPU sec.	It.	T.It	CPU sec.
L ₁ ⁽⁵⁾	145	162	5.21	1.6	400	10.00	4.0	14	57.14	0.8
L ₂ ⁽⁵⁾	3125	4173	29.15	212.8	5450	20.42	111.3	20	5995.0	119.9
L ₃ ⁽⁵⁾	5471	6697	60.27	733.4	8743	30.44	266.2	19	40163.1	763.1
L ₄ ⁽⁵⁾	5473	7666	84.35	1108.3	10240	36.23	371.0	22	86795.4	1909.5
L ₅ ⁽⁵⁾	9570	13547	118.83	2747.2	17671	44.59	788.0	22	93572.7	2058.6
L ₆ ⁽⁵⁾	13961	18887	217.37	7140.4	23440	72.84	1707.6	(g)		

Taula 6.XVII. Resultats obtinguts amb els problemes de coordinació hidrotèrmica.

test	PPRN					
	It.0	It.1	It.2	$ \mathcal{A} $	T.It	CPU sec.
$L_1^{(6)}$	184	71	82	26	1.34	0.5
$L_2^{(6)}$	162	79	86	43	2.42	0.6
$L_3^{(6)}$	219	276	343	42	1.13	0.9
$L_4^{(6)}$	646	434	1508	112	3.34	6.8
$L_5^{(6)}$	5200	3665	10933	746	23.56	348.5
$L_1^{(7)}$	228	106	40	24	3.42	0.5
$L_2^{(7)}$	3750	1921	446	347	19.01	46.0

Taula 6.XVII.(cont.) Resultats obtinguts amb els problemes de coordinació hidrotèrmica.

test	MINOS				MCNF85			LoQo		
	It.1	It.2	T.It	CPU sec.	It.	T.It	CPU sec.	It.	T.It	CPU sec.
$L_1^{(6)}$	238	79	7.88	2.5	(f)			14	71.42	1.0
$L_2^{(6)}$	154	230	8.07	3.1	(f)			13	84.61	1.1 (h)
$L_3^{(6)}$	222	218	5.22	2.3	(f)			23	56.52	1.3
$L_4^{(6)}$	542	796	10.23	13.7	(f)			18	261.11	4.7
$L_5^{(6)}$	2372	8333	103.71	1110.3	(f)			24	4650.0	111.6
$L_1^{(7)}$	128	159	9.75	2.8	(f)			17	58.82	1.0 (h)
$L_2^{(7)}$	2777	1812	54.97	252.3	(f)			21	1460.9	30.68 (h)

Pot comprovar-se a les taules precedents com MINOS no va poder solucionar alguns problemes per tenir aquests massa constriccions. Això és degut a que MINOS emmagatzema el nombre de constriccions del problema en una variable entera (i amb signe) de dos bytes, el qual implica que no pot tractar més de 32767 constriccions. També pot ser observat que LoQo no va poder executar la majoria de tests donat que requeria una quantitat excessiva de memòria. Com es veurà al capítol 8, aquest és un dels inconvenients dels mètodes de punt interior quan són aplicats a problemes multiarticle. El mateix codi LoQo, en algunes execucions ($L_1^{(6)}$, $L_1^{(7)}$ i $L_2^{(7)}$) va tenir certs problemes de convergència i les condicions d'aturada van ser relaxades. En aquests casos la solució obtinguda no va ser l'òptima, i això ha de ser tingut en compte a l'hora de comparar la seva eficiència respecte la resta de paquets.

6.3.1.1 Anàlisi dels resultats obtinguts.

De les taules 6.XII–6.XVII poden ser extrets alguns resultats sobre el rendiment comparatiu dels programes. Aquests són mostrats a la taula 6.XVIII, on el significat de cada columna és:

- “T.It ratio”: quocient del temps de CPU per iteració de cada paquet (MINOS i MCNF85) pel temps de CPU per iteració de PPRN ($\frac{T.It (MINOS/MCNF85)}{T.It (PPRN)}$), el qual ens indica com és de ràpid PPRN respecte cada un dels paquets realitzant una única iteració.
- “CPU ratio”: quocient del temps de CPU total de cada paquet (MINOS, MCNF85 i LoQo) pel temps de CPU de l’execució amb PPRN ($\frac{CPU sec (MINOS/MCNF85/LoQo)}{CPU sec (PPRN)}$), el qual ens indica com és de ràpid PPRN respecte cada paquet.

Taula 6.xviii. Comparació de temps entre PPRN i MINOS, MCNF85 i LoQo.

test	PPRN vs MINOS		PPRN vs MCNF85		PPRN vs LoQo	test	PPRN vs MINOS		PPRN vs MCNF85		PPRN vs LoQo
	T.It ratio	CPU ratio	T.It ratio	CPU ratio	CPU ratio		T.It ratio	CPU ratio	T.It ratio	CPU ratio	CPU ratio
L ₁ ¹⁾	—	17.32	—	—	76.86	L ₈ ³⁾	—	—	1.48	1.65	—
L ₂ ¹⁾	1.78	36.83	0.73	5.96	—	L ₁ ⁴⁾	—	12.35	—	—	63.52
L ₃ ¹⁾	2.18	20.52	0.74	2.40	—	L ₂ ⁴⁾	2.50	26.99	—	—	—
L ₄ ¹⁾	—	—	0.73	0.82	—	L ₃ ⁴⁾	2.51	60.11	—	—	—
L ₅ ¹⁾	2.70	9.59	1.34	1.44	12.36	L ₄ ⁴⁾	—	—	—	—	—
L ₆ ¹⁾	2.84	10.79	1.52	1.56	—	L ₅ ⁴⁾	1.48	23.30	—	—	—
L ₇ ¹⁾	—	—	1.75	1.40	—	L ₆ ⁴⁾	2.29	32.01	—	—	—
L ₈ ¹⁾	—	—	2.08	1.71	—	L ₇ ⁴⁾	2.61	36.71	—	—	—
L ₁ ²⁾	0.77	23.23	—	—	59.47	L ₈ ⁴⁾	—	—	—	—	—
L ₂ ²⁾	2.61	19.69	1.11	1.27	—	L ₁ ⁵⁾	1.38	5.33	2.66	13.33	2.66
L ₃ ²⁾	—	—	1.00	0.97	—	L ₂ ⁵⁾	2.15	9.80	1.50	5.12	5.52
L ₄ ²⁾	—	—	0.91	0.91	—	L ₃ ⁵⁾	2.48	15.77	1.25	5.72	16.41
L ₅ ²⁾	2.56	8.29	1.40	2.75	30.79	L ₄ ⁵⁾	2.61	11.66	1.12	3.90	20.1
L ₆ ²⁾	2.35	11.12	1.77	1.92	—	L ₅ ⁵⁾	2.44	9.36	0.91	2.68	7.02
L ₇ ²⁾	2.50	13.81	2.00	2.08	—	L ₆ ⁵⁾	2.75	13.57	0.92	3.24	—
L ₈ ²⁾	—	—	2.38	3.68	—	L ₁ ⁶⁾	5.88	5.0	—	—	2.0
L ₁ ³⁾	—	22.15	—	—	6.77	L ₂ ⁶⁾	3.33	5.16	—	—	1.83
L ₂ ³⁾	2.70	12.66	0.93	1.69	1.08	L ₃ ⁶⁾	4.62	2.55	—	—	1.44
L ₃ ³⁾	—	—	0.67	1.51	—	L ₄ ⁶⁾	3.06	2.01	—	—	0.69
L ₄ ³⁾	—	—	0.69	1.04	—	L ₅ ⁶⁾	4.40	3.18	—	—	0.32
L ₅ ³⁾	3.03	8.69	0.83	1.46	8.71	L ₁ ⁷⁾	2.85	5.6	—	—	2.0
L ₆ ³⁾	—	—	1.18	1.26	—	L ₂ ⁷⁾	2.89	5.47	—	—	0.66
L ₇ ³⁾	—	—	1.06	1.37	—						

Cal observar que per a LoQo només es mostra el quocient de temps total, donat que el temps per iteració no és comparable amb els altres codis, donat que la filosofia dels mètodes usats és molt diferent. En aquest sentit, tant MINOS com MCNF85 realitzen iteracions basades en el mètode del símplex (com PPRN), mentre que LoQo usa un mètode de punt interior. Això pot ser fàcilment comprovat veient que el nombre d'iteracions requerit per LoQo per assolir el punt òptim és molt inferior al que necessiten la resta de paquets.

A la vista dels resultats presentats per la taula 6.XVIII poden fer-se els següent comentaris:

- PPRN ha pogut solucionar tots els problemes assajats (els valors no mostrats a la taula es deu a que no es disposa de resultats per algun dels altres paquets i no pot realitzar-se el quocient). A continuació, MINOS gairebé ha pogut solucionar-los tots (no es disposa d'alguns resultats amb MINOS donat que no s'han executat amb ell tots els problemes en previsió d'un temps excessiu de càlcul, i no perquè hi hagi hagut errors d'execució). Per la seva banda, MCNF85 ha tingut certs problemes en trobar punts factibles per algun conjunt de problemes (per ex., $L_i^{(4)}$), i a més no pot solucionar els casos amb constriccions a banda ($L_i^{(6)}$, $L_i^{(7)}$). Pel que fa a LoQo, la majoria d'execucions amb aquest paquet no ha estat possible realitzar-les per manca de memòria.
- PPRN i MCNF85 presenten un temps per iteració millor que MINOS a tots els casos, donat que usen una especialització del símplex per a problemes multiarticle. Pel que fa a ells dos, pot comprovar-se com als problemes amb pocs articles MCNF85 té un millor rendiment per iteració, mentre que per als problemes amb molts articles PPRN es comporta millor.
- A la majoria de casos el quocient de temps de CPU total és més favorable a PPRN que el quocient de temps per iteració. Això es deu a que PPRN, en general, realitza moltes menys iteracions que MINOS i MCNF85. Fins i tot en alguns casos (per ex., $L_3^{(1)}$ per a PPRN i MCNF85) tot i tenir PPRN un comportament pitjor a cada iteració, el temps total d'execució és menor que el del codi MCNF85. En aquest cas és fonamental, doncs, el fet de dividir el procés de càlcul en les fases 0, 1 i 2.
- Per als problemes de mida gran, LoQo ha tingut un rendiment més baix que PPRN i MCNF85 en tots els casos. Tanmateix, per als problemes de coordinació hidrotèrmica, que són de mida moderada, en alguns casos (per ex., $L_5^{(6)}$ i $L_2^{(7)}$) ha tingut un molt bon comportament, essent el paquet més eficient. Aquest fet podria deure's a que la topologia de les xarxes que defineixen aquests problemes afavoreixen l'ús d'un mètode de punt interior. Cal observar que, per al cas $L_5^{(6)}$, LoQo és tres vegades més ràpid que PPRN (és el cas on s'obté un resultat més desfavorable per a PPRN). Això es deu, principalment, a que per aquest problema PPRN ha finalitzat amb un total de 746 constriccions actives (com s'observa a la taula 6.XVII), i per tant la matriu de treball que ha de tractar és força gran, fet que li alenteix tot el procediment de càlcul.
- Per acabar, es pot concloure que PPRN és, clarament, més eficient que MINOS en totes les execucions. Comparat amb MCNF85, tot i que aquest ha tingut un millor rendiment global en alguns casos (per ex., $L_4^{(1)}$), en general PPRN ha estat més eficient

i, a més, s’ha mostrat més robust (ha pogut solucionar tots els casos, mentre que MCNF85 ha tingut certs problemes de factibilitat). Finalment, comparat amb LoQo, PPRN pot executar models més grans amb els mateixos recursos (per ex., la memòria), i per als problemes grans també s’ha mostrat més eficient que aquell.

6.3.2 Resultats computacionals amb els problemes test no lineals.

Els problemes no lineals N_j^i presentats a §6.2.2 han estat executats amb només dos paquets, PPRN i MINOS (versió 5.3), donat que no es té constància de l’existència de cap altre codi especialitzat per solucionar problemes de fluxos multiarticle no lineals. Les taules 6.XIX–6.XXIV mostren els resultats obtinguts amb els dos codis. Per clarificar la lectura de la taula, per files es mostren els diferents camps d’informació, mentre que les columnes s’associen a les execucions de cada model amb ambdós paquets. El significat de cada fila d’informació és el següent:

- “It.0”: nombre d’iteracions a la fase 0.
- “It.1”: nombre d’iteracions a la fase 1.
- “It.2”: nombre d’iteracions a la fase 2.
- “ $f(x^*)$ ”: valor de la funció objectiu al punt òptim.
- “ $|\mathcal{A}|$ ”: nombre de constriccions de capacitat mútua i a banda actives (igual a la dimensió de la matriu de treball).
- “ s ”: nombre de variables superbàsiques al punt òptim.
- “CPU sec.”: temps de CPU, en segons, requerit per assolir la solució òptima.
- “ ϵ_{opt}^* ”: precisió d’optimalitat del punt solució (aquesta precisió va ser introduïda a §5.3.4). En tots els casos aquest valor hauria de ser menor que 10^{-6} , que és el valor per defecte.
- “ $\#f(x)$ ”: nombre d’avaluacions de la funció objectiu.
- “T.It 1”: temps de CPU (en milisegons) requerit en promig per cada iteració de la fase 1.
- “T.It 2”: temps de CPU (en milisegons) requerit en promig per cada iteració de la fase 2.
- “T.It 2– f ”: temps (en milisegons) requerit per cada iteració de la fase 2, excloent el temps d’avaluació de la funció objectiu.

Per a MINOS es proporciona la mateixa informació excloent les files “It.0” i “ $|\mathcal{A}|$ ”. Els valors “T.It 1”, “T.It 2” i “T.It 2– f ” van ser calculats de forma exacta a les execucions amb PPRN, mentre que per a MINOS van ser aproximats (podria haver-hi, doncs, cert soroll, però seria poc significatiu).

Caldria fer alguns comentaris sobre les taules de resultats presentades. En un problema (N_3^4) PPRN i MINOS van obtenir diferents solucions. Això es deu a la alta no-convexitat de la funció objectiu de coordinació hidrotèrmica a llarg termini. En aquest cas pot veure’s com PPRN va assolir un punt millor (amb un valor de funció objectiu menor). En aquest

Taula 6.xix. Resultats obtinguts amb la funció artificial $f_1(x)$.

	$N_1^{(1)}$		$N_2^{(1)}$		$N_3^{(1)}$	
	PPRN	MINOS	PPRN	MINOS	PPRN	MINOS
It.0	107	—	460	—	4083	—
It.1	63	163	226	617	2896	16598
It.2	654	687	1459	1753	9458	12401
$f(x^*)$	285515.68	285515.68	867915.88	867915.87	210894646.82	210894647.06
$ \mathcal{A} $	34	—	87	—	810	—
s	272	272	544	544	2723	2727
CPU sec.	10.4	17.0	87.2	139.7	3086.7	23579.6
ϵ_{opt}^*	$7.7 \cdot 10^{-7}$	$9.9 \cdot 10^{-8}$	$9.2 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$	$5.3 \cdot 10^{-7}$	$1.6 \cdot 10^{-7}$
$\#f(x)$	1200	1271	3437	2798	59198	18541
T.It 1	1.10	3.41	2.31	11.40	21.31	24.28
T.It 2	15.02	23.93	58.69	86.70	315.47	1868.88
T.It 2- f	14.72	23.63	57.91	86.10	295.21	1864.04

Taula 6.xx. Resultats obtinguts amb la funció artificial $f_2(x)$.

	$N_1^{(2)}$		$N_2^{(2)}$		$N_3^{(2)}$	
	PPRN	MINOS	PPRN	MINOS	PPRN	MINOS
It.0	107	—	460	—	4083	—
It.1	63	163	226	689	2896	16598
It.2	845	1160	2127	2297	15463	17019
$f(x^*)$	$3.9718 \cdot 10^{10}$	$3.9718 \cdot 10^{10}$	$6.0465 \cdot 10^{10}$	$6.0465 \cdot 10^{10}$	$1.9431 \cdot 10^{14}$	$1.9431 \cdot 10^{14}$
$ \mathcal{A} $	34	—	95	—	788	—
s	211	233	449	449	1238	1900
CPU sec.	10.4	22.5	83.1	118.4	5823.3	16524.7
ϵ_{opt}^*	$4.8 \cdot 10^{-7}$	$2.3 \cdot 10^{-7}$	$1.8 \cdot 10^{-7}$	$2.5 \cdot 10^{-8}$	$5.0 \cdot 10^{-7}$	$7.7 \cdot 10^{-8}$
$\#f(x)$	1780	2299	4010	4320	152579	24174
T.It 1	1.10	3.48	2.31	10.39	20.65	73.56
T.It 2	11.18	18.90	37.46	48.42	367.96	899.21
T.It 2- f	10.55	18.31	36.43	47.40	330.70	893.84

exemple, per comparar el comportament d'ambdós codis caldria tenir en compte aquest fet.

A totes les execucions la tolerància d'optimalitat requerida va ser $\epsilon_{opt} = 10^{-6}$. Només al test $N_2^{(4)}$ aquesta no va poder ser assolida pels dos codis, degut a les no-linealitats de la funció objectiu. En aquest cas, PPRN va reduir ϵ_{opt}^* més que MINOS, assolint per tant un millor valor de funció objectiu (aquest és el motiu de que els codis tinguin un valor diferent de funció objectiu, més que no pas considerar que han assolit òptims locals diferents).

Taula 6.xxi. Resultats obtinguts amb la funció artificial $f_3(x)$.

	N ₁ ³⁾		N ₂ ³⁾		N ₃ ³⁾	
	PPRN	MINOS	PPRN	MINOS	PPRN	MINOS
It.0	175	—	460	—	4083	—
It.1	74	256	226	689	2896	16598
It.2	1640	1638	3539	3772	18289	28314
$f(x^*)$	971.69	971.69	891.69	891.69	212682.61	212682.61
$ \mathcal{A} $	27	—	81	—	713	—
s	266	266	546	546	1927	1929
CPU sec.	30.9	34.4	236.0	292.3	6831.2	19398.0
ϵ_{opt}^*	$1.5 \cdot 10^{-7}$	$2.8 \cdot 10^{-9}$	$4.8 \cdot 10^{-7}$	$4.1 \cdot 10^{-8}$	$6.1 \cdot 10^{-7}$	$7.4 \cdot 10^{-8}$
$\#f(x)$	5851	3666	14337	8212	139047	47686
T.It 1	1.08	6.78	2.16	6.37	17.92	73.45
T.It 2	17.75	19.94	65.37	76.32	367.16	642.04
T.It 2- f	14.32	17.79	54.99	70.74	235.49	612.87

Taula 6.xxii. Resultats obtinguts amb els problemes de coordinació hidrotèrmica a llarg termini.

	N ₁ ⁴⁾		N ₂ ⁴⁾		N ₃ ⁴⁾	
	PPRN	MINOS	PPRN	MINOS	PPRN	MINOS
It.0	175	—	460	—	4083	—
It.1	74	261	226	574	2813	16069
It.2	395	202	4457	5092	14186	16569
$f(x^*)$	$-3.7747 \cdot 10^{12}$	$-3.7747 \cdot 10^{12}$	$1.0792 \cdot 10^8$	$1.2228 \cdot 10^8$	$-7.9171 \cdot 10^{9\dagger}$	$-6.7860 \cdot 10^{9\dagger}$
$ \mathcal{A} $	51	—	99	—	642	—
s	2	2	260	69	112	118
CPU sec.	3.2	3.9	183.1	213.9	2504.1	4284.9
ϵ_{opt}^*	$4.7 \cdot 10^{-8}$	$8.0 \cdot 10^{-8}$	$3.1 \cdot 10^{-3\dagger}$	$1.7 \cdot 10^{-2 \dagger}$	$7.5 \cdot 10^{-8}$	$3.8 \cdot 10^{-10}$
$\#f(x)$	432	211	12379	11699	21068	20559
T.It 1	0.94	4.94	2.57	9.12	17.85	72.09
T.It 2	6.60	12.91	44.12	40.97	169.98	188.68
T.It 2- f	2.60	9.09	8.77	11.79	46.10	85.15

[†] La precisió d'optimalitat requerida $\epsilon_{opt} = 10^{-6}$ no va poder ser assolida.

[‡] Mínims locals diferents van ser assolits.

Com es va esmentar anteriorment, PPRN i MINOS van ser executats amb les opcions per defecte. Això significa que PPRN va calcular $Z^t H Z p_S = -g_z$ usant un mètode quasi-Newton mentre $s \leq 500$ (s denota el nombre de variables superbàsiques) i canvià al mètode de Newton-truncat en el moment en que $s > 500$. Per la seva banda, el paquet MINOS en tot moment usa el mètode quasi-Newton. Aquest fet és crucial pel que fa al rendiment

Taula 6.xxiii. Resultats obtinguts amb els problemes de coordinació hidrotèrmica a curt termini.

	N ₁ ⁵⁾		N ₂ ⁵⁾		N ₃ ⁵⁾		N ₄ ⁵⁾	
	PPRN	MINOS	PPRN	MINOS	PPRN	MINOS	PPRN	MINOS
It.0	3419	—	5209	—	7044	—	11650	—
It.1	1177	2653	915	2905	1216	2965	3341	12257
It.2	1692	2522	1608	2299	1850	2455	6768	7507
$f(x^*)$	0.4009	0.4009	0.8715	0.8715	0.3844	0.3844	1.0920	1.0917 [†]
$ \mathcal{A} $	470	—	530	—	510	—	1525	—
s	66	66	335	338	288	296	230	231
CPU sec.	110.2	226.5	156.7	357.8	217.3	485.5	2026.0	3316.9
ϵ_{opt}^*	$7.4 \cdot 10^{-13}$	$1.9 \cdot 10^{-11}$	$4.1 \cdot 10^{-7}$	$5.8 \cdot 10^{-8}$	$8.4 \cdot 10^{-11}$	$1.0 \cdot 10^{-17}$	$4.5 \cdot 10^{-7}$	$1.2 \cdot 10^{-8}$
$\#f(x)$	2036	6219	2583	6359	2714	6897	9821	19092
T.It 1	19.47	28.79	27.85	38.52	31.58	58.46	120.87	128.62
T.It 2	45.89	59.52	71.22	106.94	91.05	127.14	230.58	231.83
T.It 2- f	41.19	49.91	63.07	92.92	79.85	105.69	208.56	193.25

[†] En aquest execució el paràmetre “tolerància de la factibilitat” del paquet MINOS va ser incrementada considerablement per tal de poder obtenir una solució factible. El diferent valor $f(x^*)$ per a MINOS i PPRN pot ser degut a aquest fet.

Taula 6.xxiv. Resultats obtinguts amb el problema d’assignació de trànsit.

	N ₁ ⁶⁾	
	PPRN	MINOS
It.0	869	—
It.1	78	660
It.2	460	526
$f(x^*)$	288.9697	288.9697
$ \mathcal{A} $	1	—
s	9	9
CPU sec.	11.6	69.6
ϵ_{opt}^*	$1.3 \cdot 10^{-10}$	$2.7 \cdot 10^{-8}$
$\#f(x)$	258	657
T.It 1	6.66	50.88
T.It 2	19.60	68.47
T.It 2- f	16.93	62.52

dels problemes N₃¹⁾, N₃²⁾ i N₃³⁾, on el nombre de variables superbàsiques al punt òptim és molt gran. En aquests tres casos el temps requerit per PPRN és molt menor que el de MINOS, tot i que PPRN realitza moltes més avaluacions de la funció objectiu ($\#f(x)$) —donat que està usant l’algorisme de Newton-truncat. Per tant, es pot concloure que

el comportament tan diferent d'ambdós codis és degut, en aquest exemples, a la forma en que calculen la direcció superbàsica p_S quan el nombre de variables superbàsiques és molt gran, i, clarament, en aquest cas l'algorisme de Newton-truncat sembla ser molt més eficient que el mètode quasi-Newton.

6.3.2.1 Anàlisi dels resultats obtinguts.

De les taules 6.XIX–6.XXIV poden ser extrets alguns resultats sobre el rendiment comparatiu dels dos paquets. Aquests són mostrats a la taula 6.XXV, on el significat de cada columna és:

Taula 6.XXV. Comparació de temps entre PPRN i MINOS.

test	T.It 1 ratio	T.It 2 ratio	T.It 2– f ratio	CPU ratio	test	T.It 1 ratio	T.It 2 ratio	T.It 2– f ratio	CPU ratio
$N_1^{(1)}$	3.10	1.59	1.60	1.63	$N_1^{(4)}$	5.25	1.95	3.49	1.21
$N_2^{(1)}$	4.93	1.47	1.48	1.60	$N_2^{(4)}$	3.54	0.92	1.34	1.16
$N_3^{(1)}$	1.13	5.92	6.31	7.63	$N_3^{(4)}$	4.03	1.11	1.84	1.71
$N_1^{(2)}$	3.16	1.69	1.73	2.16	$N_1^{(5)}$	1.47	1.29	1.21	2.05
$N_2^{(2)}$	4.49	1.29	1.30	1.42	$N_2^{(5)}$	1.38	1.50	1.47	2.28
$N_3^{(2)}$	3.56	2.44	2.70	2.83	$N_3^{(5)}$	1.85	1.39	1.32	2.23
$N_1^{(3)}$	6.27	1.12	1.24	1.11	$N_4^{(5)}$	1.06	1.00	0.92	1.63
$N_2^{(3)}$	2.94	1.22	1.28	1.23	$N_1^{(6)}$	7.63	3.49	3.69	6.00
$N_3^{(3)}$	4.09	1.74	2.60	2.83					

- “T.It 1 ratio”: quocient del temps de CPU promig de cada iteració de la fase 1 entre MINOS i PPRN. $\left(\frac{\text{T.It 1 (MINOS)}}{\text{T.It 1 (PPRN)}}\right)$, és a dir, com de ràpid és PPRN respecte MINOS realitzant una iteració de la fase 1.
- “T.It 2 ratio”: quocient del temps de CPU promig de cada iteració de la fase 2 entre MINOS i PPRN. $\left(\frac{\text{T.It 2 (MINOS)}}{\text{T.It 2 (PPRN)}}\right)$, és a dir, com de ràpid és PPRN respecte MINOS realitzant una iteració de la fase 2.
- “T.It 2– f ratio”: quocient del temps de CPU promig de cada iteració de la fase 2 (excloent el temps de les avaluacions de la funció objectiu) entre MINOS i PPRN $\left(\frac{\text{T.It 2–}f \text{ (MINOS)}}{\text{T.It 2–}f \text{ (PPRN)}}\right)$, és a dir, com de ràpid és PPRN respecte MINOS realitzant una iteració de la fase 2 sense considerar les avaluacions de $f(x)$.
- “CPU ratio”: quocient del temps total d’execució entre MINOS i PPRN $\left(\frac{\text{CPU sec. (MINOS)}}{\text{CPU sec. (PPRN)}}\right)$, és a dir, quantes vegades PPRN és més ràpid que MINOS.

De la taula 6.XXV poden ser extretes les següents conclusions:

- Pot observar-se com PPRN és més eficient que MINOS en tots els casos (la columna “CPU ratio” té sempre un valor major que 1). Si es comparen aquests resultats amb els obtinguts per al cas lineal (taula 6.XVIII) pot comprovar-se que PPRN tenia un rendiment relatiu respecte MINOS molt millor en tractar funcions objectius lineals.

Això és lògic, donat que s'explota més la tècnica del particionament primal per a funcions lineals que per al cas no lineal.

- El mateix fenomen abans comentat (que s'explota més l'estructura del problema al cas lineal), pot comprovar-se observant que PPRN és més eficient realitzant una iteració de la fase 1 (columna "T.it 1 ratio") que de la fase 2 (columna "T.it 2 ratio"), donat que en aquell cas la funció objectiu a optimitzar és lineal.
- També, com al cas lineal, s'observa que el rendiment global de PPRN respecte MINOS (columna "CPU ratio") és major que el rendiment per iteració. Això també és degut a que PPRN realitza menys iteracions de fase 1 i fase 2 que MINOS a la majoria d'execucions.
- Pel que fa a les columnes "T.it 2 ratio" i "T.it 2- f ratio", s'observa que no hi ha cap tendència clara: en alguns casos el primer valor és superior al segon, i en altres ocorre la situació contrària. Això indica que PPRN realitza un nombre d'avaluacions de la funció objectiu major (amb un cost d'avaluació associat major també) en alguns casos, i menor en d'altres. No hi ha, doncs, una clara tendència, i depèn del problema concret que es realitzin més o menys avaluacions de $f(x)$ amb un paquet que amb l'altre.
- Hi ha tres problemes (N_3^1 , N_3^2 , N_3^3), un per cada una de les funcions objectius artificials usades, on al punt òptim assolit el nombre de variables superbàsiques és elevat (el primer d'ells, N_3^1 , és qui té un major valor de superbàsiques a l'òptim: 2723 per a PPRN i 2727 per a MINOS). En aquests tres casos PPRN automàticament passarà a usar el mètode de Newton-truncat quan el nombre de superbàsiques superi el valor de 500, mentre que MINOS continua usant un mètode quasi-Newton (amb certes modificacions, per no haver d'emmagatzemar una matriu densa massa gran). S'observa a la taula 6.XVIII com en aquests casos el guany en temps per iteració a la fase 2 és molt millor per a PPRN que per a MINOS (sobre tot per al problema N_3^1), i això repercuteix de forma molt favorable en el temps total de l'execució.

Capítol 7

Algorisme primal-dual de punt interior.

Als capítols precedents s'ha desenvolupat la metodologia seguida per tractar problemes multiarticle lineals i no lineals usant la tècnica del particionament primal, detallant tant aspectes teòrics com aportant resultats computacionals. L'objectiu que es perseguirà d'aquí al final de la memòria serà la solució de problemes multiarticle lineals per mètodes de punt interior (per aquests mètodes no s'abordaran els problemes amb funcions objectiu no lineals). Per tal d'assolir aquest objectiu, en aquest capítol es detallarà l'algorisme primal-dual de punt interior (per a problemes de programació lineal en general) que serveix de base per a la posterior especialització per tractar problemes multiarticle, presentada al capítol següent. El capítol 9 presentarà els resultats computacionals obtinguts amb l'especialització multiarticle. El material presentat en aquest capítol prové d'un altre treball de l'autor [12].

7.1 Preliminars.

L'algorisme presentat en aquest capítol es basa, com ja s'ha dit, en un mètode primal-dual de punt interior, i s'aplica a la solució de problemes de programació lineal en general. Malgrat ser, des d'un punt de vista formal, més complexos que els mètodes d'escalat afí, els mètodes primal-dual s'han mostrat computacionalment més eficients que aquells. L'algorisme desenvolupat tracta les fites superiors a les variables de forma explícita, sense afegir restriccions addicionals. Això fa que no incrementem la dimensió del sistema simètric i definit positiu a solucionar a cada iteració (donat que aquesta dimensió és exactament el nombre de restriccions del nostre problema). A més, el fet de que la topologia d'aquest sistema sigui sempre la mateixa a cada iteració, permet usar rutines de factorització simbòlica per tal de determinar a priori el nombre i la situació dels elements nous no-zero creats en fer la descomposició de Cholesky. Aquesta factorització simbòlica caldrà fer-la només una vegada al principi de l'execució, i serà aprofitada posteriorment a cada iteració. En aquest punt el codi difereix clarament respecte altres implementacions capdavanteres d'algorismes primal-dual (com ara [77,78]), que solucionen un sistema simètric i indefinit mitjançant una factorització de Bunch-Parlett [26,79] (implementant en aquest cas un mètode predictor-corrector [54]), que, en teoria, ha de ser lleugerament menys eficient que una factorització de Cholesky coneixent el patró d'esparsitat.

El present capítol s'estructurarà de la següent forma. En primer lloc es detallaran les bases de l'algorisme primal-dual amb fites superiors a algunes variables. A continuació es comentaran certs aspectes referents a detalls de la implementació feta. Finalment el

codi desenvolupat per tractar problemes lineals es compararà amb dos codis capdavanters d'optimització lineal, usant una bateria estàndard de 80 problemes de programació lineal. Cal fer notar que és important presentar amb un cert detall aquest algorisme i certs aspectes de la seva implementació, donat que aquests constitueixen la base i són aplicats d'igual manera a la posterior especialització multiaricle. Aquest fet també justifica la presentació de resultats computacionals obtinguts amb la implementació de l'algorisme per solucionar problemes lineals generals.

7.2 L'algorisme primal-dual considerant fites superiors a algunes variables.

En aquesta secció es farà un breu esment del mètode primal-dual usat, que no és més que una especialització del mètode general per a problemes amb fites superiors en algunes variables. No es pretén, però, aprofundir en els aspectes de convergència ni justificar el mètode. Per més detalls sobre els mètodes primal-dual hom pot consultar [5] i [56].

7.2.1 Formulació dels problemes primal i dual.

Considerem el següent problema de minimització amb fites superiors a algunes variables

$$\begin{aligned} \min \quad & c^t x \\ \text{subj. a} \quad & \tilde{A}x = b \\ & 0 \leq x_u \leq \bar{x}_u \\ & 0 \leq x_l \end{aligned} \tag{7.1}$$

on $x_u \in \mathbb{R}^{n_u}$, $x_l \in \mathbb{R}^{n_l}$, $x = (x_u^t, x_l^t)^t$, $x \in \mathbb{R}^{\tilde{n}}$ (llavors $\tilde{n} = n_u + n_l$), $c \in \mathbb{R}^{\tilde{n}}$, $b \in \mathbb{R}^{\tilde{m}}$ i $\tilde{A} \in \mathbb{R}^{\tilde{m} \times \tilde{n}}$. Considerant un particionament escaient de c i \tilde{A} , i afegint folgues $f \in \mathbb{R}^{n_u}$ per als límits superiors, el problema anterior pot ser escrit com:

$$\begin{aligned} \min \quad & c_u^t x_u + c_l^t x_l \\ \text{subj. a} \quad & A_u x_u + A_l x_l = b \\ & x_u + f = \bar{x}_u \\ & x \geq 0 \\ & f \geq 0 \end{aligned} \tag{7.2}$$

on $c_u \in \mathbb{R}^{n_u}$, $c_l \in \mathbb{R}^{n_l}$, $A_u \in \mathbb{R}^{\tilde{m} \times n_u}$ i $A_l \in \mathbb{R}^{\tilde{m} \times n_l}$.

Considerant les variables duals $y \in \mathbb{R}^{\tilde{m}}$, $u \in \mathbb{R}^{n_u}$, i les folgues $z = (z_u^t, z_l^t)^t$ ($z_u \in \mathbb{R}^{n_u}$, $z_l \in \mathbb{R}^{n_l}$, per tant $z \in \mathbb{R}^{\tilde{n}}$) i $w \in \mathbb{R}^{n_u}$, el problema dual associat a (7.2) pot ser escrit

com:

$$\begin{aligned}
 & \max \quad b^t y + \bar{x}_u^t u \\
 \text{subj. a} \quad & A_u^t y + u + z_u = c_u \\
 & A_l^t y + z_l = c_l \\
 & u + w = 0 \\
 & y \in \mathbb{R}^{\tilde{m}} \quad u \in \mathbb{R}^{n_u} \\
 & z_l \geq 0 \quad z_u \geq 0 \quad w \geq 0
 \end{aligned} \tag{7.3}$$

De la tercera constricció es té que $u = -w$ i, aleshores, eliminant u el problema dual pot ser formulat com:

$$\begin{aligned}
 & \max \quad b^t y - \bar{x}_u^t w \\
 \text{subj. a} \quad & A_u^t y + z_u - w = c_u \\
 & A_l^t y + z_l = c_l \\
 & y \in \mathbb{R}^{\tilde{m}} \\
 & z_l \geq 0 \quad z_u \geq 0 \quad w \geq 0
 \end{aligned} \tag{7.4}$$

7.2.2 Obtenció dels Lagrangians a través una barrera logarísmica.

Als problemes primal i dual prèviament definits les constriccions de no-negativitat de les variables poden ser substituïdes per una penalització de barrera logarísmica (tal i com descriu [83]). Llavors el problema primal (7.2) pot ser expressat com:

$$\begin{aligned}
 \min \quad & c_u^t x_u + c_l^t x_l - \mu \sum_{i=1}^{\tilde{n}} \ln x_i - \mu \sum_{i=1}^{n_u} \ln(\bar{x}_{u_i} - x_{u_i}) \\
 \text{subj. a} \quad & A_u x_u + A_l x_l = b \\
 & \mu \geq 0
 \end{aligned} \tag{7.5}$$

mentre que el dual (7.4) pot formular-se tal i com segueix:

$$\begin{aligned}
 \max \quad & b^t y - \bar{x}_u^t w + \mu \sum_{i=1}^{\tilde{n}} \ln z_i + \mu \sum_{i=1}^{n_u} \ln w_i \\
 \text{subj. a} \quad & A_u^t y + z_u - w = c_u \\
 & A_l^t y + z_l = c_l \\
 & y \in \mathbb{R}^{\tilde{m}} \quad \mu \geq 0
 \end{aligned} \tag{7.6}$$

A continuació, associant els multiplicadors de Lagrange $y \in \mathbb{R}^{\tilde{m}}$ a les constriccions d'igualtat de (7.5) construïm el Lagrangia L_p del problema primal:

$$L_p(x, y, \mu) = c_u^t x_u + c_l^t x_l - \mu \sum_{i=1}^{\tilde{n}} \ln x_i - \mu \sum_{i=1}^{n_u} \ln(\bar{x}_{u_i} - x_{u_i}) - y^t (A_u x_u + A_l x_l - b) \tag{7.7}$$

Anàlogament, associant els multiplicadors $x_u \in \mathbb{R}^{n_u}$ i $x_l \in \mathbb{R}^{n_l}$ a les constriccions d'igualtat de (7.6) obtenim L_d , el Lagrangià del dual:

$$L_d(x, y, z, w, \mu) = b^t y - \bar{x}_u^t w + \mu \sum_{i=1}^{\tilde{n}} \ln z_i + \mu \sum_{i=1}^{n_u} \ln w_i - x_u^t (A_u^t y + z_u - w - c_u) - x_l^t (A_l^t y + z_l - c_l) \quad (7.8)$$

7.2.3 Condicions d'optimalitat de Karush-Kuhn-Tucker de primer ordre.

Si denotem per e_l el vector l -dimensional de 1's i considerem $X_u, X_l, X, Z_u, Z_l, Z, W, F$ matrius diagonals, on:

$$\begin{aligned} e_l &= (1_1, \dots, 1_l)^t \\ X_u &= \text{diag}(x_{u_1}, \dots, x_{u_{n_u}}) \\ X_l &= \text{diag}(x_{l_1}, \dots, x_{l_{n_l}}) \\ X &= \begin{pmatrix} X_u & \mathbf{0} \\ \mathbf{0} & X_l \end{pmatrix} \\ Z_u &= \text{diag}(z_{u_1}, \dots, z_{u_{n_u}}) \\ Z_l &= \text{diag}(z_{l_1}, \dots, z_{l_{n_l}}) \\ Z &= \begin{pmatrix} Z_u & \mathbf{0} \\ \mathbf{0} & Z_l \end{pmatrix} \\ W &= \text{diag}(w_1, \dots, w_{n_u}) \\ F &= \text{diag}(\bar{x}_{u_1} - x_{u_1}, \dots, \bar{x}_{u_{n_u}} - x_{u_{n_u}}) \end{aligned} \quad (7.9)$$

llavors les condicions d'optimalitat necessàries de primer ordre de L_p (7.7) poden ser escrites com:

$$\frac{\partial L_p}{\partial x_u} = c_u - \mu X_u^{-1} e_{n_u} + \mu F^{-1} e_{n_u} - A_u^t y := 0 \quad (7.10)$$

$$\frac{\partial L_p}{\partial x_l} = c_l - \mu X_l^{-1} e_{n_l} - A_l^t y := 0 \quad (7.11)$$

$$\frac{\partial L_p}{\partial y} = -(A_u x_u + A_l x_l - b) := 0 \quad (7.12)$$

Per la seva banda, les condicions d'optimalitat de primer ordre del Lagrangià dual (7.8) són:

$$\frac{\partial L_d}{\partial x_u} = c_u - A_u^t y - z_u + w := 0 \quad (7.13)$$

$$\frac{\partial L_d}{\partial x_l} = c_l - A_l^t y - z_l := 0 \quad (7.14)$$

$$\frac{\partial L_d}{\partial y} = b - A_u x_u - A_l x_l := 0 \quad (7.15)$$

$$\frac{\partial L_d}{\partial z} = \mu Z^{-1} e_n - X e_n := 0 \quad (7.16)$$

$$\frac{\partial L_d}{\partial w} = \mu W^{-1} e_{n_u} - F e_{n_u} := 0 \quad (7.17)$$

Les condicions (7.12) i (7.15) són la mateixa, i imposen la *factibilitat primal* a l'òptim. Les condicions (7.13) i (7.14) imposen la *factibilitat dual* de la solució. Les condicions (7.16) i (7.17) (usant (7.9)) poden ser reescrites com:

$$\left. \begin{aligned} Z_u X_u e_{n_u} &= \mu e_{n_u} \\ Z_l X_l e_{n_l} &= \mu e_{n_l} \end{aligned} \right\} \quad (7.18)$$

$$F W e_{n_u} = \mu e_{n_u} \quad (7.19)$$

i les anomenarem condicions de *complementarietat* (això degut a que, quan $\mu \rightarrow 0$ tenim que (7.18) i (7.19) són exactament les condicions del teorema de la folga complementària, les quals han de ser garantides per la solució del problema primal i dual). Finalment, es pot veure fàcilment que, verificant-se la factibilitat primal, la factibilitat dual i la complementarietat, automàticament queden satisfetes les dues condicions restants (7.10) i (7.11):

- pel que fa a (7.10) veiem que:

$$\begin{aligned} 0 &\stackrel{?}{=} c_u - \mu X_u^{-1} e_{n_u} + \mu F^{-1} e_{n_u} - A_u^t y \\ &= A_u^t y + z_u - w - \mu X_u^{-1} e_{n_u} + \mu F^{-1} e_{n_u} - A_u^t y && \text{[usant (7.13)]} \\ &= Z_u e_{n_u} - W e_{n_u} - \mu X_u^{-1} e_{n_u} + \mu F^{-1} e_{n_u} && \text{[usant (7.9)]} \\ &= (Z_u e_{n_u} - \mu X_u^{-1} e_{n_u}) + (\mu F^{-1} e_{n_u} - W e_{n_u}) \\ &= 0 && \text{[usant (7.18) i (7.19)]} \end{aligned}$$

- per la seva banda, per (7.11) observem que:

$$\begin{aligned} 0 &\stackrel{?}{=} c_l - \mu X_l^{-1} e_{n_l} - A_l^t y \\ &= A_l^t y + z_l - \mu X_l^{-1} e_{n_l} - A_l^t y && \text{[usant (7.14)]} \\ &= Z_l e_{n_l} - \mu X_l^{-1} e_{n_l} && \text{[usant (7.9)]} \\ &= 0 && \text{[usant (7.18)]} \end{aligned}$$

Per tant, les condicions finals de Karush-Kuhn-Tucker de primer ordre que ha de verificar un punt per ser considerat òptim del nostre problema seran les sis següents:

$$X_u Z_u e_{n_u} = \mu e_{n_u} \quad (7.20)$$

$$X_l Z_l e_{n_l} = \mu e_{n_l} \quad (7.21)$$

$$A_u x_u + A_l x_l = b \quad (7.22)$$

$$A_l^t y + z_l = c_l \quad (7.23)$$

$$FWe_{n_u} = \mu e_{n_u} \quad (7.24)$$

$$A_u^t y + z_u - w = c_u \quad (7.25)$$

Clarament, quan $n_u = 0$ ($\tilde{n} = n_l$, és a dir, no hi ha cap variable amb fita superior) només cal considerar les equacions (7.21), (7.22) i (7.23), les quals són, efectivament, les condicions d'optimalitat del mètode primal-dual estàndard sense fites a les variables (com es pot veure a [5]).

7.2.4 Solució del sistema no lineal.

El sistema no lineal d'equacions (7.20–7.25) resultant es resoldrà usant el mètode de Newton, on el sistema lineal d'equacions $J_i d_i = -f_i$ —essent J_i el jacobià del sistema, d_i la direcció de Newton i f_i l'avaluació del sistema al punt actual— a ser resolt a cada iteració i és:

Z_u			X_u			=	$\mu e_{n_u} - X_u Z_u e_{n_u}$	(7.26)
	Z_l			X_l			$\mu e_{n_l} - X_l Z_l e_{n_l}$	
A_u	A_l						$b - \tilde{A}x$	
		A_l^t		$\mathbb{1}_{n_l}$			$c_l - A_l^t y - z_l$	
$-W$					F		$\mu e_{n_u} - W F e_{n_u}$	
		A_u^t	$\mathbb{1}_{n_u}$		$-\mathbb{1}_{n_u}$		$c_u - A_u^t y - z_u + w$	
						dx_u		
						dx_l		
						dy		
						dz_u		
						dz_l		
						dw		

Definint els nous vectors (que representen els termes independents del sistema lineal anterior):

$$b_{1_l} = \mu e_{n_l} - X_l Z_l e_{n_l} \quad (7.27)$$

$$b_{1_u} = \mu e_{n_u} - X_u Z_u e_{n_u} \quad (7.28)$$

$$b_2 = \mu e_{n_u} - W F e_{n_u} \quad (7.29)$$

$$b_3 = b - \tilde{A}x \quad (7.30)$$

$$b_{4_l} = c_l - A_l^t y - z_l \quad (7.31)$$

$$b_{4_u} = c_u - A_u^t y - z_u + w \quad (7.32)$$

el sistema lineal d'equacions a resoldre a cada iteració de l'algorisme pot ser escrit com:

$$Z_u dx_u + X_u dz_u = b_{1_u} \quad (7.33)$$

$$Z_l dx_l + X_l dz_l = b_{1_l} \quad (7.34)$$

$$-W dx_u + F dw = b_2 \quad (7.35)$$

$$A_u dx_u + A_l dx_l = b_3 \quad (7.36)$$

$$A_u^t dy + dz_u - dw = b_{4_u} \quad (7.37)$$

$$A_l^t dy + dz_l = b_{4_l} \quad (7.38)$$

La següent proposició presenta com solucionar el sistema anterior.

Proposició 7.1.

La solució del sistema (7.33–7.38) ve donada per:

$$(\tilde{A}S\tilde{A}^t)dy = b_3 + \tilde{A}Sr \quad (7.39)$$

$$dx = S(\tilde{A}^t dy - r) \quad (7.40)$$

$$dw = F^{-1}(b_2 + Wdx_u) \quad (7.41)$$

$$dz_u = b_{4_u} + dw - A_u^t dy \quad (7.42)$$

$$dz_l = b_{4_l} - A_l^t dy \quad (7.43)$$

essent

$$\begin{aligned} r &= (r_u^t, r_l^t)^t \quad r \in \mathbb{R}^{\tilde{n}} \quad r_u \in \mathbb{R}^{n_u} \quad r_l \in \mathbb{R}^{n_l} \\ r_u &= F^{-1}b_2 + b_{4_u} - X_u^{-1}b_{1_u} \quad r_l = b_{4_l} - X_l^{-1}b_{1_l} \end{aligned} \quad (7.44)$$

i

$$\begin{aligned} S &= \begin{pmatrix} S_u & \mathbf{0} \\ \mathbf{0} & S_l \end{pmatrix} \quad S \in \mathbb{R}^{\tilde{n} \times \tilde{n}}, \quad S_u \in \mathbb{R}^{n_u \times n_u}, \quad S_l \in \mathbb{R}^{n_l \times n_l} \\ S_u &= FX_u(Z_uF + X_uW)^{-1} \quad S_l = Z_l^{-1}X_l \end{aligned} \quad (7.45)$$

on S_u i S_l poden ser calculades directament, donat que no són més que productes i sumes de matrius diagonals

Demostració.

Aïllant dz_u de (7.37), dz_l de (7.38), dw de (7.35), dx_u de (7.33) i dx_l de (7.34) obtenim:

$$dz_u = b_{4_u} - A_u^t dy + dw \quad (7.46)$$

$$dz_l = b_{4_l} - A_l^t dy \quad (7.47)$$

$$dw = F^{-1}(b_2 + Wdx_u) \quad (7.48)$$

$$dx_u = Z_u^{-1}(b_{1_u} - X_u dz_u) \quad (7.49)$$

$$dx_l = Z_l^{-1}(b_{1_l} - X_l dz_l) \quad (7.50)$$

Substituint (7.46) i (7.48) a (7.49) tenim que:

$$\begin{aligned} dx_u &= Z_u^{-1}(b_{1_u} - X_u[b_{4_u} - A_u^t dy + dw]) \\ &= Z_u^{-1}(b_{1_u} - X_u b_{4_u} - X_u dw + X_u A_u^t dy) \\ &= Z_u^{-1}(b_{1_u} - X_u b_{4_u} - X_u[F^{-1}(b_2 + Wdx_u)] + X_u A_u^t dy) \\ &= Z_u^{-1}(b_{1_u} - X_u b_{4_u} - X_u F^{-1}b_2 - X_u F^{-1}Wdx_u + X_u A_u^t dy) \\ &= Z_u^{-1}b_{1_u} - Z_u^{-1}X_u b_{4_u} - Z_u^{-1}X_u F^{-1}b_2 - Z_u^{-1}X_u F^{-1}Wdx_u + Z_u^{-1}X_u A_u^t dy \end{aligned}$$

Agrupant termes s'arriba a

$$(\mathbb{1} + Z_u^{-1}X_uF^{-1}W)dx_u = Z_u^{-1}b_{1_u} - Z_u^{-1}X_ub_{4_u} - Z_u^{-1}X_uF^{-1}b_2 + Z_u^{-1}X_uA_u^tdy$$

Definint T_u com

$$T_u = \mathbb{1} + Z_u^{-1}X_uF^{-1}W \quad (7.51)$$

la seva inversa ve donada directament com

$$T_u^{-1} = FZ_u(FZ_u + X_uW)^{-1} \quad (7.52)$$

Llavors l'expressió anterior de dx_u pot ser escrita com

$$dx_u = T_u^{-1}(Z_u^{-1}b_{1_u} - Z_u^{-1}X_ub_{4_u} - Z_u^{-1}X_uF^{-1}b_2 + Z_u^{-1}X_uA_u^tdy) \quad (7.53)$$

Anàlogament, substituint (7.47) a (7.50) s'obté directament que

$$\begin{aligned} dx_l &= Z_l^{-1}(b_{1_l} - X_l[b_{4_l} - A_l^tdy]) \\ dx_l &= Z_l^{-1}b_{1_l} - Z_l^{-1}X_lb_{4_l} + Z_l^{-1}X_lA_l^tdy \end{aligned} \quad (7.54)$$

Ara, usant (7.53) i (7.54), l'equació (7.36) queda com segueix:

$$\begin{aligned} A_u[T_u^{-1}(Z_u^{-1}b_{1_u} - Z_u^{-1}X_ub_{4_u} - Z_u^{-1}X_uF^{-1}b_2 + Z_u^{-1}X_uA_u^tdy)] + \\ + A_l[Z_l^{-1}b_{1_l} - Z_l^{-1}X_lb_{4_l} + Z_l^{-1}X_lA_l^tdy] = b_3 \\ A_uT_u^{-1}Z_u^{-1}(b_{1_u} - X_ub_{4_u} - X_uF^{-1}b_2) + A_uT_u^{-1}Z_u^{-1}X_uA_u^tdy + \\ + A_lZ_l^{-1}(b_{1_l} - X_lb_{4_l}) + A_lZ_l^{-1}X_lA_l^tdy = b_3 \end{aligned} \quad (7.55)$$

Definint S_u i S_l com

$$S_u = T_u^{-1}Z_u^{-1}X_u = FX_u(FZ_u + X_uW)^{-1} \quad (7.56)$$

$$S_l = Z_l^{-1}X_l \quad (7.57)$$

i agrupant termes, l'expressió (7.55) queda

$$\begin{aligned} (A_uS_uA_u^t + A_lS_lA_l^t)dy = b_3 + A_uT_u^{-1}Z_u^{-1}(X_uF^{-1}b_2 + X_ub_{4_u} - b_{1_u}) + A_lZ_l^{-1}(X_lb_{4_l} - b_{1_l}) \\ (A_uS_uA_u^t + A_lS_lA_l^t)dy = b_3 + A_uS_u(F^{-1}b_2 + b_{4_u} - X_u^{-1}b_{1_u}) + A_lS_l(b_{4_l} - X_l^{-1}b_{1_l}) \end{aligned}$$

L'expressió final del càlcul dy és:

$$(\tilde{A}S\tilde{A}^t)dy = b_3 + \tilde{A}Sr \quad (7.58)$$

on r i S són respectivament:

$$r = (r_u^t, r_l^t)^t \quad r_u = F^{-1}b_2 + b_{4_u} - X_u^{-1}b_{1_u} \quad r_l = b_{4_l} - X_l^{-1}b_{1_l} \quad (7.59)$$

$$S = \begin{pmatrix} S_u & \mathbf{0} \\ \mathbf{0} & S_l \end{pmatrix} \quad (7.60)$$

Un cop hem calculat dy , substituint el seu valor a (7.53) i (7.54) dóna:

$$\begin{aligned} dx_u &= T_u^{-1}(Z_u^{-1}b_{1_u} - Z_u^{-1}X_u b_{4_u} - Z_u^{-1}X_u F^{-1}b_2 + Z_u^{-1}X_u A_u^t dy) \\ &= T_u^{-1}Z_u^{-1}X_u(X_u^{-1}b_{1_u} - b_{4_u} - F^{-1}b_2 + A_u^t dy) \\ dx_u &= S_u(A_u^t dy - r_u) \end{aligned} \quad (7.61)$$

$$\begin{aligned} dx_l &= Z_l^{-1}b_{1_l} - Z_l^{-1}X_l b_{4_l} + Z_l^{-1}X_l A_l^t dy \\ &= Z_l^{-1}X_l(X_l^{-1}b_{1_l} - b_{4_l} + A_l^t dy) \\ dx_l &= S_l(A_l^t dy - r_l) \end{aligned} \quad (7.62)$$

Agrupant (7.61) i (7.62) tenim que l'expressió final del càlcul de dx és:

$$dx = S(\tilde{A}^t dy - r) \quad (7.63)$$

Un cop dy i dx són coneguts, dz i dw es calculen directament a través de les expressions inicials (7.46), (7.47) i (7.48). ■

L'algorisme presentat soluciona a cada iteració el sistema (7.26) a través de (7.39–7.43), a diferència d'altres —com [77]— on (7.26) es soluciona usant un mètode predictor-corrector [54] el qual requereix de tècniques per a matrius simètriques indefinides basades en la factorització de Bunch-Parlett [26,79,78].

També cal fer notar que en la solució del nostre sistema cal invertir les matrius diagonals X, Z, W, F definides a (7.9). Naturalment això només és possible si cap element diagonal és igual a 0. Per tant, per poder garantir l'obtenció de (7.39–7.43) cal que les variables primals i duals no arribin mai a les seves fites (inferiors o superiors), és a dir, hem d'obligar a tenir sempre punts estrictament interiors respecte les seves fites.

7.2.5 Actualització del nou punt i del paràmetre μ de penalització.

Un cop s'han calculat les direccions (7.39–7.43), cal actualitzar les noves variables primals i duals per a la següent iteració. És a dir, calcularem:

$$\begin{aligned} x_{i+1} &= x_i + \alpha_p dx \\ y_{i+1} &= y_i + \alpha_d dy \\ z_{i+1} &= z_i + \alpha_d dz \\ w_{i+1} &= w_i + \alpha_d dw \end{aligned} \quad (7.64)$$

on α_p i α_d són tals que preserven el fet de que cada iterat sigui un punt interior (és a dir, α_p manté que $0 < x_{l_{i+1}}$ i $0 < x_{u_{i+1}} < \bar{x}_u$, mentre que α_d garanteix que $0 < z$ i $0 < w$).

Només queda fer l'actualització del paràmetre μ de la barrera logarísmica abans de tornar a calcular les direccions (7.39–7.43) per al nou iterat calculat prèviament. Com és habitual als mètodes primal-dual, el paràmetre μ s'actualitza en funció de la distància que hi ha al punt actual entre la funció primal i la funció dual (que anomenarem *gap dual*). Al nostre cas concret, tenim que:

$$\begin{aligned}
\text{gap dual} &= c^t x - (b^t y - \bar{x}_u^t w) && [\text{usant (7.2) i (7.4)}] \\
&= c^t x - x^t \tilde{A}^t y + \bar{x}_u^t w && [\text{usant (7.2)}] \\
&= (c^t - y^t \tilde{A})x + \bar{x}_u^t w \\
&= (z_u^t - w^t \quad z_l^t) \begin{pmatrix} x_u \\ x_l \end{pmatrix} + \bar{x}_u^t w && [\text{usant (7.4)}] \\
&= z_u^t x_u + z_l^t x_l - x_u^t w + \bar{x}_u^t w \\
&= z^t x + (\bar{x}_u - x_u)^t w \\
&= z^t x + f^t w && [\text{usant (7.2)}] \\
&= \mu \tilde{n} + \mu n_u && [\text{usant (7.20), (7.21) i (7.24)}] \\
&= (\tilde{n} + n_u) \mu \\
&\Downarrow \\
\mu &= \frac{\text{gap dual}}{(\tilde{n} + n_u)}
\end{aligned}$$

Per tal de que a la següent iteració es millori el *gap dual* (garantint la convergència de l'algorisme), es prendrà com nou paràmetre μ_{i+1} una fracció de l'anterior. És a dir:

$$\mu_{i+1} = \sigma \frac{\text{gap dual}}{(\tilde{n} + n_u)} \quad \text{on} \quad 0 < \sigma < 1 \quad (7.65)$$

7.3 Implementació de l'algorisme.

En aquesta secció es detallaran alguns aspectes de la implementació de l'algorisme realitzada. Els cinc punts concrets a tractar seran la forma de solucionar el sistema (7.39) a cada iteració, com escollir el punt inicial (x_0, y_0, z_0, w_0) , com calcular les passes α_p i α_d de (7.64), com fer l'actualització del paràmetre μ de la barrera logarísmica i quines són les condicions d'aturada del procés iteratiu de minimització.

7.3.1 Obtenció de dy a cada iteració.

El pas més costós, des del punt de vista computacional, de tot l'algorisme és l'obtenció de dy a través del sistema (7.39). Donat que es garanteix que en tot moment el punt actual és interior respecte les fites simples de les variables, tenim que la matriu $\tilde{A}S\tilde{A}^t$ serà simètrica i definida positiva, sempre i quan \tilde{A} sigui de rang complet (és a dir, sempre i

quan $\text{rang}(\tilde{A}) = \tilde{m}$). D'igual forma, per a la majoria de problemes la matriu \tilde{A} acostuma a ser força esparsa, i sovint aquesta propietat es manté en construir $\tilde{A}\tilde{S}\tilde{A}^t$. Per tant sembla raonable aplicar tècniques de resolució de sistemes especialment dissenyades per a matrius simètriques definides positives i esparses (és a dir, s'emprarà una factorització de Cholesky per a matrius esparses i de gran dimensió). Concretament, la implementació realitzada utilitza la llibreria SPARSPAK [29] dissenyada per aquest tipus de sistemes.

En fer la factorització de Cholesky de $\tilde{A}\tilde{S}\tilde{A}^t$ cal tenir molt present l'ordenació de les seves files (o columnes), donat que una mala ordenació pot donar lloc a que es creïn un gran nombre d'elements no-zero, tot degradant l'esparsitat original de la matriu. Per tant, abans de fer la factorització, cal reordenar les files de $\tilde{A}\tilde{S}\tilde{A}^t$ de forma que garantim la creació del menor nombre possible d'elements. Hi ha dos possibles mètodes a aplicar per tal de trobar aquesta ordenació millor: l'ordenació del *mínim grau* (“minimum-degree ordering”), i la del *mínim ompliment* (“minimum-local-fill-in ordering”) [26]. Al codi desenvolupat s'ha usat la rutina GENQMD de la llibreria SPARSPAK, la qual implementa l'algorisme del mínim grau.

Cal també observar que la topologia de la matriu $\tilde{A}\tilde{S}\tilde{A}^t$ no varia durant tota l'execució del programa. Només hi ha una variació en els coeficients de la matriu diagonal S definida a (7.45). Aquest fet és clau, donat que implica que només cal fer una única reordenació de files mitjançant l'algorisme del mínim grau al principi, en comptes d'una vegada a cada iteració. D'igual forma també es pot explotar el fet de tenir una topologia constant fent una factorització simbòlica prèvia del sistema. Això ens proporciona el patró d'esparsitat del sistema un cop ja factoritzat, i agilitza clarament el posterior procés de càlcul.

Per tal de calcular de forma eficient $\tilde{A}\tilde{S}\tilde{A}^t$ (només la part sub i diagonal, però, ja que és una matriu simètrica) el codi usa una variació de la idea proposada a [55]. Considerem que la part subdiagonal de $\tilde{A}\tilde{S}\tilde{A}^t$ s'emmagatzema per columnes al vector `lnz(maxlnz)`, on `maxlnz` és el nombre d'elements no nuls de la part subdiagonal de $\tilde{A}\tilde{S}\tilde{A}^t$ un cop realitzada la factorització simbòlica, mentre que els elements diagonals es troben al vector `diag(m)`. El nombre d'elements que originalment es tenen a $\tilde{A}\tilde{S}\tilde{A}^t$ (abans de factoritzar) correspon al nombre de parells de files de \tilde{A} ($\tilde{A}^{i_1}, \tilde{A}^{i_2}$) tal que alguna variable x_j apareix amb un coeficient no nul en ambdues constriccions i_1 i i_2 (és a dir $\exists j : a_{i_1,j} \neq 0$ i $a_{i_2,j} \neq 0$). Considerem que el nombre de parells de files que donen lloc a un element no nul de la part sub i diagonal de $\tilde{A}\tilde{S}\tilde{A}^t$ és `np`. Llavors tindrem un vector `ilnz(np)` que ens dirà per cada parell de files amb elements coincidents quina posició ocupa dins els vectors `lnz(.)` o `diag(.)`, tenint en compte la informació de la factorització simbòlica prèviament feta. Per tal de que `ilnz(.)` mantingui una única numeració, els vectors `lnz(.)` i `diag(.)` s'emmagatzemen a memòria de forma consecutiva, dins un vector que podem anomenar `asa(maxlnz+m)`, i llavors aquelles posicions i tals que $1 \leq \text{ilnz}(i) \leq \text{maxlnz}$ correspondran a la part subdiagonal, mentre que aquelles on $\text{maxlnz} < \text{ilnz}(i) \leq \text{maxlnz} + \tilde{m}$ estaran associades a la part diagonal.

També cal un vector `ifillin(nfill)`, on `nfill` és el nombre d'elements no-zero creats a la part subdiagonal en factoritzar (és a dir, `nfill = maxlnz - (np - m)`). Aquest vector ens

dóna les posicions dins `lnz(.)` dels nous elements creats, les quals han de ser inicialitzades a zero a cada iteració. A més, disposarem de tres vectors més anomenats `ka(np+1)`, `la(.)` i `va(.)`. El vector `ka(.)` per cada parell de files (i_1, i_2) de \tilde{A} que donen lloc a un element no-zero té un punter als vectors `la(.)` i `va(.)`. El primer d'aquests, `la(.)`, ens diu quines són les columnes j amb elements no-zero a les dues files i_1 i i_2 que estem considerant, mentre que `va(.)` ens dóna directament el producte $a_{i_1,j} \cdot a_{i_2,j}$. El nombre total de columnes amb elements no-zero d'un parell de files es troba fent `ka(i+1)-ka(i)`, essent i l'entrada associada a aquesta parella de files dins el vector `ka(.)`. Per exemple, si considerem la matriu de constriccions \tilde{A} següent:

$$\tilde{A} = \begin{pmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ & & 3 & 2 & 1 & & \\ 3 & & & 5 & & 1 & \\ 2 & 3 & -1 & & & & 1 \end{pmatrix}$$

on $\tilde{m}=3$, $\text{np}=6$, $\text{maxlnz}=3$ i $\text{nfill}=0$, tenim que els vectors anteriorment definits són:

$$\begin{aligned} (i) &= 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ \text{ka}(i) &= 1 & 4 & 5 & 7 & 10 & 12 & 16 & & & & & & & & \\ \text{la}(i) &= 2 & 3 & 4 & 3 & 2 & 3 & 1 & 3 & 5 & 1 & 3 & 1 & 2 & 3 & 6 \\ \text{va}(i) &= 9 & 4 & 1 & 10 & 9 & -2 & 9 & 25 & 1 & 6 & -5 & 4 & 9 & 1 & 1 \\ \text{ilnz}(i) &= 4 & 1 & 2 & 5 & 3 & 6 & & & & & & & & & \end{aligned}$$

Amb aquesta estructura de dades, considerant que la matriu diagonal S es troba al vector $\mathbf{s}(\tilde{n})$ de dimensió nombre de columnes de \tilde{A} , la creació de la part sub i diagonal de $\tilde{A}S\tilde{A}^t$ es redueix simplement a fer:

Algorisme A7.1. Càlcul de $\tilde{A}S\tilde{A}^t$.

```

per i = 1 fins nfill fer
    asa(ifillin(i))= 0.0
fi_per
per i = 1 fins np fer
    acum:= 0.0
    per j = ka(i) fins ka(i+1)-1 fer
        acum:= acum + va(j) · s(la(j))
    fi_per
asa(ilnz(i))= acum
fi_per

```

Tot i que la matriu $\tilde{A}S\tilde{A}^t$ es suposa definida positiva, podria ser que, si \tilde{A} no és de rang complet, s'arribés a tenir pivots diagonals amb valor nul en fer la descomposició de Cholesky. Aquest fet podria ser evitat si s'usés pivotació parcial en escollir el pivot,

però això ens modificaria l'esparsitat de la descomposició desaprofitant la factorització simbòlica feta prèviament. El que realment s'ha fet, seguint una recomanació de [55], ha estat, en comptes de solucionar el sistema $(\tilde{A}S\tilde{A}^t)x = b$, solucionar el sistema escalat $(\tilde{A}(S/\gamma)\tilde{A}^t)x = b/\gamma$, on $\gamma = \max\{S_{ii}, i = 1, \dots, m\}$. Si en solucionar el nou sistema escalat trobem un pivot nul, automàticament li assignem un valor petit (p.e, 10^{-12}), i continuem la factorització. Això és equivalent a introduir una petita perturbació dins la matriu \tilde{A} de forma que eliminem el fet de tenir alguna fila combinació lineal d'altres. Els resultats obtinguts amb aquesta regla han estat força satisfactoris. Tanmateix, a més d'aquesta simple estratègia, s'han provat d'altres de més acurades, com ara la triangularització de Gill-Murray [30], sense haver obtingut un millor resultat.

7.3.2 Elecció del punt inicial.

Un dels punts clau dins de l'algorisme és el càlcul del punt inicial d'iteració. Aquest punt, que ha de ser estrictament interior, s'inicialitza atenent dos criteris principals. El primer és el de no fixar variables a un valor que estigui a prop de les seves fites, donat que llavors tindríem un punt "poc interior". El segon és que, donat que el punt òptim ha de satisfer les condicions (7.20–7.25), interessa que el punt inicial satisfaci ja d'entrada el màxim nombre de condicions d'optimalitat possible. Dit això, el criteri seguit a l'hora de trobar el punt inicial és el següent:

- a) Variables x : $x_l = 100$ i $x_u = \min\{100, \frac{\bar{x}_u}{2}\}$.
- b) Variables y : $y = 0$ (y són variables lliures i poden tenir qualsevol valor).
- c) Variables z i w : donat que hem fixat $y = 0$, llavors les condicions de factibilitat dual (7.23) i (7.25) queden directament com

$$\begin{aligned} z_l &= c_l \\ z_u - w &= c_u \end{aligned}$$

La primera condició es podrà satisfer si $\forall i c_{l_i} > 0$, ja que sinó tindríem una component de z_l no interior. A més, no només interessa que totes les components de c_l siguin positives, sinó que també cal que estiguin lluny de zero per evitar tenir un punt "poc interior". Per tal de garantir això, z_l s'inicialitza com $z_{l_i} = \max\{c_{l_i}, 100\}$. La segona condició sempre la satisfarem inicialitzant w i z_u com segueix:

$$w_i, z_{u_i} := \begin{cases} w_i = 100, z_{u_i} = c_{u_i} + 100 & \text{si } c_{u_i} \geq 0 \\ z_{u_i} = 100, w_i = 100 - c_{u_i} & \text{si } c_{u_i} < 0 \end{cases}$$

7.3.3 Càlcul de les passes α_p i α_d .

Les passes α_p i α_d usades a (7.64) es calculen de forma que en tot moment es mantinguin els valors de les variables entre fites. A més, sempre que sigui possible fer-ho sense violar les fites de les variables, s'intenta que prenguin el valor de 1, ja que així estariem

resolent el sistema lineal (7.26) usant la direcció exacta del mètode de Newton. Per tal de garantir ambdós objectius α_p i α_d són calculades com:

$$\begin{aligned}\alpha_p &= \min \left\{ \rho \cdot \min \left\{ \frac{-x_i}{dx_i} \forall i : dx_i < 0 \right\}, \rho \cdot \min \left\{ \frac{\bar{x}_u - x_{u_i}}{dx_{u_i}} \forall i : dx_{u_i} > 0 \right\}, 1 \right\} \\ \alpha_d &= \min \left\{ \rho \cdot \min \left\{ \frac{-z_i}{dz_i} \forall i : dz_i < 0 \right\}, \rho \cdot \min \left\{ \frac{-w_i}{dw_i} \forall i : dw_i < 0 \right\}, 1 \right\}\end{aligned}\tag{7.66}$$

on ρ és un valor <1 però proper a 1, usat per impedir que la variable que ens proporciona la passa màxima arribi a la seva fita. A la implementació feta $\rho=0.99$.

7.3.4 Actualització del paràmetre μ .

L'expressió de l'actualització del paràmetre μ a la nova iteració $i+1$ venia donada per l'equació (7.65) i era $\mu_{i+1} = \sigma \cdot \text{gap dual}/(\tilde{n} + n_u)$, on σ havia de ser un valor tal que $0 < \sigma < 1$. En general un valor de $\sigma=0.1$ acostuma a ser prou satisfactori. Tanmateix s'ha usat una actualització dinàmica de μ , tal i com es descriu a [78]. Sigui $\alpha_{pd} = \min\{\alpha_p, \alpha_d\}$, α_p i α_d definits a (7.66), llavors el paràmetre σ es troba a cada iteració com:

$$\sigma = \left(\frac{1 - \alpha_{pd}}{10\alpha_{pd} + 1} \right)^2\tag{7.67}$$

Això fa que, quan α_{pd} és proper a 1 (és a dir, no ha estat necessari escurçar o bé α_p o bé α_d a causa de les fites) tenim que σ tendeix a 0, amb el qual disminuïm μ_{i+1} . Per l'altra banda, quan α_{pd} s'apropa a 0 (alguna variable està a prop de la seva fita) augmentem μ_{i+1} . Donat que el paràmetre μ controla la barrera logarísmica, actualitzant σ d'aquesta forma estem fent que, quan les variables estan lluny de les fites, disminuïm la barrera logarísmica, mentre que quan alguna està a prop incrementem el seu pes dins el Lagrangià.

7.3.5 Condicions d'aturada.

Les condicions d'aturada implementades són les habituals dels mètodes primal-dual: un punt es considera òptim quan supera un test de factibilitat primal, un de factibilitat dual i un sobre el *gap dual* entre les funcions objectiu primal i dual. Concretament les tres condicions a complir són:

$$\frac{\|\tilde{A}x - b\|_2}{1 + \|b\|_2} < \epsilon_p \quad \text{Factibilitat Primal}\tag{7.68}$$

$$\frac{\left\| \begin{pmatrix} A_u^t y + z_u - w \\ A_l^t y + z_l \end{pmatrix} - c \right\|_2}{1 + \|c\|_2} < \epsilon_d \quad \text{Factibilitat Dual}\tag{7.69}$$

$$\frac{|c^t x - (b^t y - \bar{x}_u^t w)|}{1 + c^t x} < \epsilon_g \quad \text{Gap Dual}\tag{7.70}$$

A la implementació realitzada els valors usats per a ϵ_p , ϵ_d i ϵ_g són $\epsilon_p = \epsilon_d = 10^{-6}$ i $\epsilon_g = 10^{-8}$.

7.4 Resultats computacionals.

Presentem en aquesta secció els resultats computacionals obtinguts amb la implementació feta de l'algorisme presentat en apartats anteriors. S'han usat un total de 80 problemes de programació lineal pertanyents a la bateria de tests de la Netlib [28]. Tots aquests tests han estat agafats a través d'un ftp anònim al compte *netlib.att.com*, i es troben dins del directory */netlib/lp/data*. Només s'han usat problemes sense variables lliures a la seva formulació, donat que la implementació feta no contempla aquesta possibilitat. La taula 7.I mostra les principals característiques dels problemes test usats. La taula presenta per a cada problema el seu nom (columna "Nom"), el nombre de constriccions del problema (columna "Constr."), el nombre de variables (columna "Vars.") , i el nombre d'elements no-zero de la matriu de constriccions (columna "No-zeros \tilde{A} ").

La implementació feta (a la qual ens referirem com IP) ha esta codificada en ANSI-C i s'ha comparat amb dos codis capdavanters de programació lineal. El primer d'ells és el paquet MINOS 5.3 [59], el qual està basat en el mètode del símplex per a programació lineal. El segon és el paquet LoQo [77], una implementació eficient d'un mètode primal-dual per a programació lineal i quadràtica. Totes les execucions s'han realitzat sobre una SunSparc 10/41 sota UNIX, amb un únic processador RISC de 40Mhz i aproximadament 20 Mflops i 100 Mips, i amb 64 Mbytes de memòria (32 Mbytes reals i 32 Mbytes mapejats a disc). La taula 7.II mostra els resultats obtinguts amb cada paquet. Per a cada un dels tres codis (IP, LoQo i MINOS) es presenta el valor òptim assolit (columna "Valor òptim"), el nombre d'iteracions realitzat (columna "Iter.") i el temps de CPU en segons requerit (columna "CPU").

Es pot observar a la taula 7.II com els tres codis en tot moment van assolir els mateixos valors òptims (hi ha petites diferències, però són menyspreables). També es pot veure que IP i LoQo realitzen moltes menys iteracions que MINOS, degut a que estan basats en un mètode de punt interior en comptes de en el símplex. I comparant només aquests dos, en general IP realitza més iteracions que LoQo. El codi desenvolupat (IP) no va poder executar dos problemes ("fit2p" i "maros-r7") degut a problemes de memòria. LoQo no va tenir aquests problemes perquè no resol el sistema (7.26) a través de (7.39–7.43), sinó directament aplicant tècniques per a matrius simètriques i indefinides. Aquest fet explica també el comportament tan diferent que tenen ambdós codis en alguns exemples (per exemple, als problemes "fit1p" i "fit2d"). IP va tenir problemes de convergència al problema "greenbea", mentre que MINOS no va poder satisfer les constriccions de forma acurada al test "maros-r7". La darrera fila de la taula 7.II mostra el valor promig per a cada codi pel que fa al nombre d'iteracions i temps de CPU. Pot observar-se com els dos codis de punt interior es mostren més eficients que MINOS. També es veu que LoQo té un millor rendiment que IP. Tanmateix IP realitza moltes més iteracions que LoQo, el qual indica que efectivament IP té un cost per iteració menor que LoQo, com s'havia previst inicialment per la forma especial de solucionar el sistema (7.26).

Taula 7.1. Característiques dels problemes test.

Nom	Constr.	Vars.	No-zeros A
25fv47	822	1571	11127
80bau3b	2263	9799	29063
adlittle	57	97	465
afiro	28	32	88
agg	489	163	2541
agg2	517	302	4515
agg3	517	302	4531
bandm	306	472	2659
beaconfd	174	262	3476
blend	75	83	521
bnl1	644	1175	6129
bnl2	2325	3489	16124
boeing1	351	384	3865
boeing2	167	143	1339
bore3d	234	315	1525
brandy	221	249	2150
czprob	930	3523	14173
d2q06c	2172	5167	35674
d6cube	416	6184	43888
degen2	445	534	4449
degen3	1504	1818	26230
e226	224	282	2767
etamacro	401	688	2489
ffff800	525	854	6235
finnis	498	614	2714
fit1d	25	1026	14430
fit1p	628	1677	10894
fit2d	26	10500	138018
fit2p	3001	13525	60784
ganges	1310	1681	7021
gfrd-pnc	617	1092	3467
greenbea	2393	5405	31499
grow15	301	645	5665
grow22	441	946	8318
grow7	141	301	2633
israel	175	142	2358
kb2	44	41	291
lotfi	154	308	1086
maros	847	1443	10006
maros-r7	3137	9408	151120

Nom	Constr.	Vars.	No-zeros A
nesm	663	2923	13988
pilot	1442	3652	43220
pilot87	2031	4883	73804
pilotnov	976	2172	13129
recipe	92	180	752
sc105	106	103	281
sc205	206	203	552
sc50a	51	48	131
sc50b	51	48	119
scagr25	472	500	2029
scagr7	130	140	553
scfxm1	331	457	2612
scfxm2	661	914	5229
scfxm3	991	1371	7846
scorpion	389	358	1708
scrs8	491	1169	4029
scsd1	78	760	3148
scsd6	148	1350	5666
scsd8	398	2750	11334
sctap1	301	480	2052
sctap2	1091	1880	8124
sctap3	1481	2480	10734
seba	516	1028	4874
share1b	118	225	1182
share2b	97	79	730
shell	537	1775	4900
ship04l	403	2118	8450
ship04s	403	1458	5810
ship08l	779	4283	17085
ship08s	779	2387	9501
ship12l	1152	5427	21597
ship12s	1152	2763	10941
sierra	1228	2036	9252
standata	360	1075	3038
standgub	362	1184	3147
standmps	468	1075	3686
stocfor1	118	111	474
stocfor2	2158	2031	9492
wood1p	245	2594	70216
woodw	1099	8405	37478

Taula 7.II. Resultats obtinguts amb els problemes test.

Nom	IP			LoQo			MINOS		
	Valor òptim	Iter.	CPU	Valor òptim	Iter.	CPU	Valor òptim	Iter.	CPU
25fv47	5501.8	45	23.9	5501.8	29	19.8	5501.8	6477	75.8
80bau3b	987224.2	71	61.3	987224.2	44	52.4	987229.0	11158	239.2
adlittle	225495.0	20	0.1	225495.0	14	0.1	225495.0	108	0.3
afiro	-464.8	13	0.0	-464.8	13	0.0	-464.8	14	0.1
agg	-35991767.3	43	6.8	-35991767.3	26	1.9	-35991767.3	158	1.6
agg2	-20239252.3	37	9.5	-20239252.3	22	4.4	-20239252.4	223	2.2
agg3	10312115.9	39	9.8	10312115.9	22	4.4	10312115.9	255	2.2
bandm	-158.6	34	1.5	-158.6	20	1.6	-158.6	474	2.7
beaconfd	33592.5	18	1.3	33592.5	14	1.2	33592.5	95	1.1
blend	-30.8	20	0.1	-30.8	14	0.2	-30.8	115	0.4
bnl1	1977.6	51	7.5	1977.6	35	7.1	1977.6	1166	9.5
bnl2	1811.2	61	124.7	1811.2	40	122.4	1811.2	5441	125.0
boeing1	-335.2	44	4.0	-335.2	28	3.1	-335.2	579	3.0
boeing2	-315.0	32	0.8	-315.0	28	1.1	-315.0	159	0.7
bore3d	1373.1	29	0.9	1373.1	17	0.9	1373.1	118	1.1
brandy	1518.5	29	1.2	1518.5	22	1.5	1518.5	310	1.7
czprob	2185196.7	56	12.8	2185196.7	38	10.9	2185196.7	1506	17.8
d2q06c	122784.2	58	263.6	122784.2	36	247.0	122784.2	43934	1246.4
d6cube	315.5	51	80.8	315.5	23	75.2	315.5	96780	1049.1
degen2	-1435.2	26	6.7	-1435.2	14	4.7	-1435.2	974	6.5
degen3	-987.3	38	165.6	-987.3	17	73.4	-987.3	7109	158.4
e226	-18.8	39	1.7	-18.8	22	1.4	-18.8	455	1.9
etamacro	-755.7	49	5.6	-755.7	30	8.9	-755.7	761	4.1
ffff800	555679.6	78	17.8	555679.6	36	10.6	555679.6	272	3.3
finnis	172791.1	42	2.7	172791.1	26	2.2	172791.0	547	3.4
fit1d	-9146.4	56	4.8	-9146.4	21	5.3	-9146.4	2626	6.4
fit1p	9146.4	26	381.6	9146.4	26	4.8	9146.4	894	13.6
fit2d	-68464.3	48	44.7	-68464.3	24	195.0	-68464.3	32235	291.2
fit2p	^(a)			68464.3	24	41.1	68464.3	14240	1128.3
ganges	-109585.7	41	15.2	-109585.7	23	10.7	-109585.8	663	9.5
gfrd-pnc	6902236.0	55	1.9	6902236.0	19	1.6	6902236.0	686	5.7
greenbea	^(b)			-72555247.5	47	86.9	-72462405.9	24537	645.5
grow15	-106870941.1	24	2.0	-106870941.3	23	3.0	-106870941.3	426	4.3
grow22	-160834336.2	25	3.3	-160834336.5	28	5.2	-160834336.5	676	8.0
grow7	-47787811.8	23	0.8	-47787811.8	20	1.2	-47787811.8	175	1.4
israel	-896644.8	71	12.3	-896644.8	28	1.2	-896644.8	250	1.2
kb2	-1749.9	28	0.1	-1749.9	21	0.2	-1749.9	50	0.2
lotfi	-25.3	29	0.5	-25.3	25	0.8	-25.3	205	0.8
maros	-58063.7	65	22.2	-58063.7	28	14.0	-58063.7	1910	20.0
maros-r7	^(a)			1497185.2	22	3603.3	^(c)		

^(a) No hi ha suficient memòria per executar aquest problema. ^(b) Problemes de convergència.

^(c) Error numèric. No es poden satisfer les constriccions de forma acurada.

Taula 7.II.(cont.) Resultats obtinguts amb els problemes test.

Nom	IP			LoQo			MINOS		
	Valor òptim	Iter.	CPU	Valor òptim	Iter.	CPU	Valor òptim	Iter.	CPU
nesm	14076036.6	65	19.4	14076036.5	38	22.2	14076073.1	3061	26.4
pilot	-557.5	72	520.5	-557.5	44	463.3	-557.4	15996	738.5
pilot87	301.7	79	2312.7	301.6	46	1923.7	301.7	24284	3041.2
pilotnov	-4497.3	40	40.1	-4497.3	29	39.7	-4497.3	2525	34.9
recipe	-266.6	18	0.1	-266.6	13	0.3	-266.6	27	0.4
sc105	-52.2	15	0.1	-52.2	14	0.1	-52.2	44	0.3
sc205	-52.2	17	0.2	-52.2	17	0.4	-52.2	77	0.6
sc50a	-64.6	15	0.0	-64.6	14	0.1	-64.6	28	0.2
sc50b	-70.0	12	0.0	-70.0	13	0.1	-70.0	15	0.2
scagr25	-14753433.1	47	1.4	-14753433.1	18	1.0	-14753433.1	319	2.4
scagr7	-2331389.8	23	0.2	-2331389.8	15	0.2	-2331389.8	106	0.5
scfxm1	18416.8	38	1.7	18416.8	26	1.9	18416.8	361	2.1
scfxm2	36660.3	46	4.8	36660.3	28	4.2	36660.3	753	6.5
scfxm3	54901.3	46	8.4	54901.3	28	6.4	54901.3	1097	13.4
scorpion	1878.1	23	0.7	1878.1	16	0.7	1878.1	134	1.3
scrs8	904.3	37	2.5	904.3	23	2.9	904.3	682	5.3
scsd1	8.7	14	0.3	8.7	15	0.9	8.7	359	1.8
scsd6	50.5	17	0.8	50.5	17	1.6	50.5	1106	5.2
scsd8	905.0	17	1.8	905.0	17	3.4	905.0	3026	25.6
sctap1	1412.3	30	0.8	1412.3	18	0.9	1412.2	257	1.6
sctap2	1724.8	33	7.7	1724.8	16	4.9	1724.8	661	9.0
sctap3	1424.0	36	11.9	1424.0	16	6.2	1424.0	953	20.2
seba	15711.6	40	68.4	15711.6	19	2.1	15711.6	366	3.5
share1b	-76589.3	89	1.0	-76589.3	26	0.7	-76589.3	250	0.8
share2b	-415.7	20	0.2	-415.7	14	0.2	-415.7	111	0.4
shell	1208825347.3	32	1.6	1208825346.9	25	3.0	1208825346.0	321	4.5
ship04l	1793324.5	21	2.4	1793324.6	19	2.9	1793324.5	290	4.3
ship04s	1798714.7	24	1.6	1798714.7	19	2.1	1798714.7	157	2.9
ship08l	1909055.2	27	6.9	1909055.2	20	6.5	1909055.2	473	10.0
ship08s	1920098.2	24	3.3	1920098.2	20	3.7	1920098.2	256	5.3
ship12l	1470187.9	25	10.0	1470187.9	24	9.7	1470187.9	961	18.6
ship12s	1489236.1	25	4.8	1489236.1	22	4.8	1489236.1	437	9.0
sierra	15394362.2	25	6.6	15394362.2	21	5.9	15394362.2	864	12.2
standata	1257.7	29	1.4	1257.7	23	1.9	1257.7	131	2.3
standgub	1257.7	29	1.4	1257.7	23	2.0	1257.7	105	2.2
standmps	1406.0	42	3.7	1406.0	32	3.4	1406.0	383	3.0
stocfor1	-41132.0	22	0.2	-41132.0	17	0.2	-41132.0	92	0.4
stocfor2	-39024.4	45	19.6	-39024.4	31	11.7	-39024.4	3092	69.8
wood1p	1.4	38	57.1	1.4	28	53.8	1.4	870	21.1
woodw	1.3	51	49.1	1.3	27	41.1	1.3	3572	76.5
PROMIG		37	58.1		23	46.0		3747	97.5

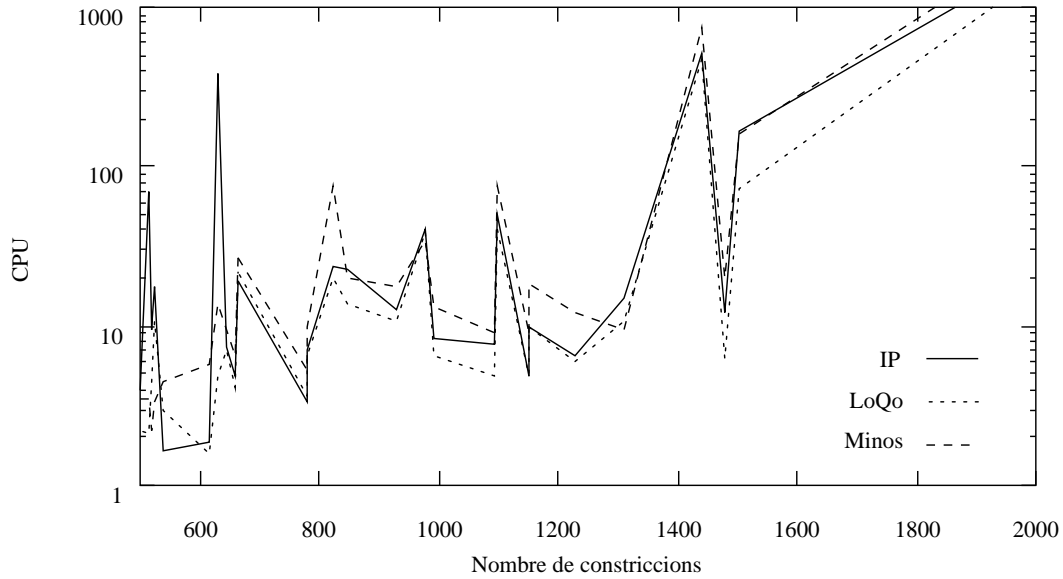


Figura 7.1. Temps de CPU per a IP, LoQo i MINOS segons la mida del problema.

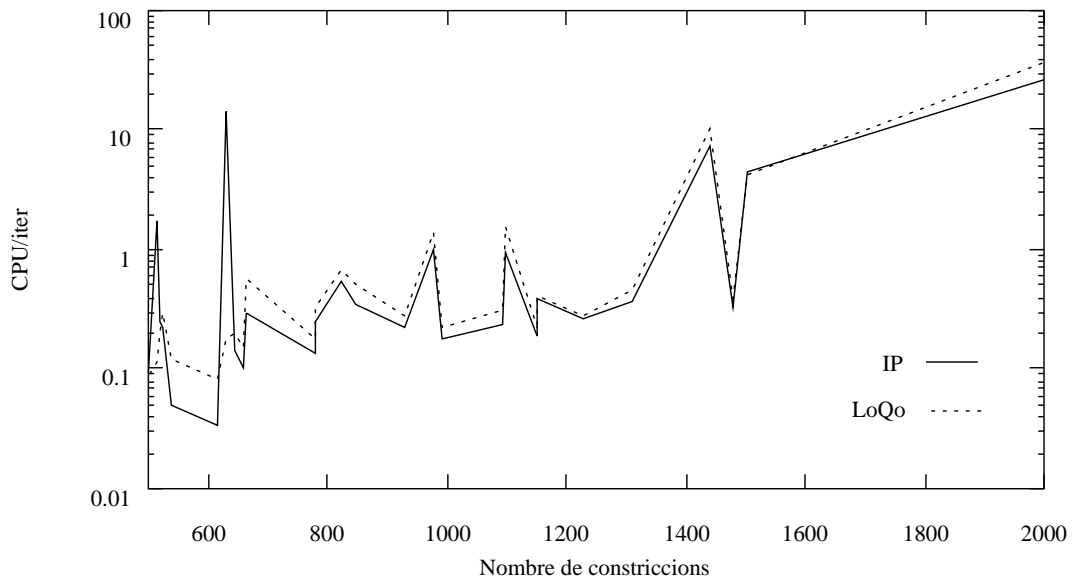


Figura 7.2. Temps per iteració per a IP i LoQo segons la mida del problema.

El dit anteriorment sobre el rendiment de cada codi pot observar-se clarament a les Fig. 7.1 i Fig. 7.2. La Fig. 7.1 mostra el temps de CPU (eix y) per als tres codis segons la mida del problema (eix x). Com a mida del problema s'ha agafat el nombre de restriccions de cada un (tot i que es podria haver seguit un altre criteri com el nombre de variables, d'elements no-zero de la matriu o una expressió funció d'aquests valors). Només es presenten els problemes amb una mida relativament gran (per ex., més de 500 restriccions), per

simplificar la gràfica. El temps de CPU s'ha escalat logarímicament. Pot observar-se que, en general, MINOS es troba per sobre dels dos codis de punt interior, mentre que LoQo dóna un millor rendiment. IP es troba en una situació intermèdia. Per la seva banda la Fig. 7.2 ens mostra, per a IP i LoQo, el temps per iteració (a l'eix y com abans) segons la mida del problema (eix x). Només es mostren, com a la figura anterior, els problemes amb més de 500 constriccions. De nou també el temps per iteració s'ha escalat logarímicament. Pot observar-se com IP té tendència a donar millors temps per iteració. Malgrat això, el fet de que requereixi força més iteracions que LoQo per convergir a l'òptim fa que no tingui un comportament global tan eficient com aquell. Aquests mateixos resultats han estat obtinguts per Vanderbei et al. a [80] i, tal i com s'esmenta en aquell document, el fet de que LoQo tingui una millor convergència i un cost per iteració major es deu al fet de que implementa un mètode predictor-corrector per solucionar el sistema d'equacions (7.26).

Capítol 8

Especialització d'un mètode primal-dual per a fluxos multiarticle.

En aquest capítol es detallarà un mètode primal-dual de punt interior per a la resolució de problemes de fluxos multiarticle lineals. Està basat en el mètode general descrit al capítol precedent, i la majoria de conceptes que a continuació es presentaran faran esment a d'altres d'introduïts prèviament.

El mètode es presentarà i desenvoluparà de forma que es considerin constriccions a banda en la formulació del problema. Tanmateix, a l'hora de descriure l'esquema de resolució numèrica, basat en un gradient conjugat amb preconditionament, es tindran en compte només les constriccions de capacitat mútua, donat que la cerca d'un preconditionador eficient per a les constriccions a banda (les quals poden tenir qualsevol estructura) és una tasca que ha quedat fora de l'abast d'aquest treball. Part del material presentat en aquest capítol prové d'un altre treball de l'autor [17].

8.1 Formulació del problema considerat.

Al capítol 2 es va fer la formulació del problema (**FMLCB**). Aquest va ser escrit en la seva forma estàndard (considerant només constriccions d'igualtat afegint folgues) com:

$$\min_{x^{(1)}, \dots, x^{(k)}, s_{cm}, s_{cb}} \sum_{i=1}^k c^{(i)t} x^{(i)} \quad (8.1)$$

subj. a

A	$\mathbf{0}$	\dots	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	=	$x^{(1)}$	$b^{(1)}$
$\mathbf{0}$	A	\dots	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$		$x^{(2)}$	$b^{(2)}$
\vdots	\vdots	\ddots	\vdots	\vdots	\vdots		\vdots	\vdots
$\mathbf{0}$	$\mathbf{0}$	\dots	A	$\mathbf{0}$	$\mathbf{0}$		$x^{(k)}$	$b^{(k)}$
$\mathbf{1}$	$\mathbf{1}$	\dots	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{0}$		s_{cm}	b_{cm}
$T^{(1)}$	$T^{(2)}$	\dots	$T^{(k)}$	$\mathbf{0}$	$\mathbf{1}$		s_{cb}	\bar{b}_{cb}

(8.2)

$$\underline{0} \leq x^{(i)} \leq \bar{x}^{(i)} \quad \forall i = 1, \dots, k \quad (8.3)$$

$$\underline{0} \leq s_{cm} \quad (8.4)$$

$$\underline{0} \leq s_{cb} \leq \bar{b}_{cb} - \underline{b}_{cb} \quad (8.5)$$

Per tal d'usar el mètode primal-dual presentat al capítol anterior per al cas de fluxos multiarticle, cal determinar quines variables correspondran a la part x_u (variables afitades inferior i superiorment) i quines a la part x_l (variables només afitades inferiorment), tal i com es va presentar a (7.1). Observant (8.3–8.5) sembla clar quines variables han de correspondre a cada partició. Tanmateix, i per tal d'explotar millor el mètode del capítol 7, cal fer una anàlisi més exhaustiva per determinar la partició correcta. En aquest sentit, es detallarà a continuació el tractament concret per a cada un dels tres tipus de variables: arcs, folgues de capacitat mútua, i folgues de constriccions a banda.

8.1.1 Tractament dels arcs.

Tal i com ja va ser dit al capítol 2 en formular el problema (**FMLCB**), es pot considerar sense pèrdua de generalitat que la matriu $A \in \mathbb{R}^{m \times n}$ conté un arc arrel (sigui per exemple, el n -èssim arc, associat amb la darrera columna de A) per garantir que $\text{rang}(A) = m$. Si es considera que l'arc arrel (el qual serà denotat, per a l'article i , per $x_a^{(i)}$) es troba associat al node 1 de la xarxa (lligat a la primera fila de A), aleshores la matriu A de xarxa pot ser escrita com

$$A = [\quad A_{\bar{a}} \quad | e_1] \quad (8.6)$$

essent $A_{\bar{a}} \in \mathbb{R}^{m \times (n-1)}$ la matriu associada a la xarxa del problema un cop eliminat l'arc arrel, i e_1 la primera columna de la matriu identitat, associada a l'arc arrel.

Considerant la partició anterior de A , i sabent que els arcs arrel tenen capacitat il·limitada (donat que són arcs ficticis), les constriccions de xarxa (2.2) i (2.3) poden ser escrites per a tot article com:

$$\left. \begin{array}{l} A_{\bar{a}} x_{\bar{a}}^{(i)} + e_1 x_a^{(i)} = b^{(i)} \\ \underline{0} \leq x_{\bar{a}}^{(i)} \leq \bar{x}_{\bar{a}}^{(i)} \\ 0 \leq x_a^{(i)} \end{array} \right\} i = 1, \dots, k \quad (8.7)$$

essent $x_{\bar{a}}^{(i)} \in \mathbb{R}^{n-1}$ el vector format per les $n-1$ primeres components de $x^{(i)}$ associades als arcs de la xarxa excloent l'arc arrel. Clarament, doncs, les constriccions de xarxa poden ser escrites de la forma presentada a (7.2), pertanyent aleshores $x_{\bar{a}}^{(i)}$ a la part x_u i $x_a^{(i)}$ a la part x_l . En el cas de disposar d'arcs sense capacitat màxima es podria generalitzar el raonament anterior, passant les seves components a formar part de x_l .

Cal indicar que, donat que se sap que al punt òptim el flux dels arcs arrel ha de ser 0, aquests podrien haver estat considerats com a variables afitades amb una fita petita. Això va ser intentat amb la implementació realitzada, però els resultats obtinguts no van ser satisfactoris (el fet d'afitar superiorment amb una fita petita els arcs arrel va empitjorar lleugerament la convergència de l'algorisme). Llavors es podria mirar d'imposar una fita gran. En aquest cas es pot comprovar com, fer tendir la fita cap a l'infinit és equivalent

a considerar la variable com a pertanyent a x_l . Això pot comprovar-se observant que el terme diagonal associat de la matriu S_u (definida a (7.45)) pren el valor:

$$\begin{aligned} \lim_{\bar{x}_i \rightarrow \infty} S_{u_i} &= \lim_{\bar{x}_i \rightarrow \infty} \frac{f_i x_{u_i}}{z_{u_i} f_i + x_{u_i} w_i} \\ &= \lim_{\bar{x}_i \rightarrow \infty} \frac{x_{u_i}}{z_{u_i} + \frac{x_{u_i} w_i}{f_i}} \\ &= \frac{x_{u_i}}{z_{u_i}} \quad \left[\text{donat que } \lim_{\bar{x}_i \rightarrow \infty} f_i = \lim_{\bar{x}_i \rightarrow \infty} \bar{x}_i - x_i = \infty \right] \end{aligned}$$

el qual equival a considerar la variable com a pertanyent a x_l i calcular la component associada de S_l (també definida a (7.45)) com $S_{l_i} = x_{l_i}/z_{l_i}$ (recordem que S_l i S_u intervenen en la solució del sistema (7.39–4.10) a resoldre a cada iteració de l'algorisme). Aquesta equivalència, però, és certa de forma teòrica. Tanmateix, s'ha pogut comprovar que emular el fet de tenir variables no afitades mitjançant la imposició de fites amb valors elevats provoca molts errors de precisió en calcular S_u , i, per tant, és millor considerar aquestes variables com del tipus x_l (en aquest punt s'observa com el mètode primal-dual emprat difereix clarament del símplex, ja que en les implementacions d'aquest és pràctica habitual considerar una fita gran per a les variables no afitades superiorment). Les raons exposades, doncs, justifiquen el fet de considerar els arcs arrel com variables només afitades inferiorment per 0.

8.1.2 Tractament de les folgues de les constriccions de capacitat mútua.

Recordant el que es va comentar a §7.3.2 sobre l'elecció del punt inicial, pot comprovar-se que es tindrà factibilitat dual si les folgues duals $z_l \in \mathbb{R}^{n_l}$, $z_u \in \mathbb{R}^{n_u}$ i $w \in \mathbb{R}^{n_u}$ poden ser inicialitzades a uns valors positius (i preferiblement allunyats de 0) de forma que satisfacin:

$$z_l = c_l \tag{8.8}$$

$$z_u - w = c_u \tag{8.9}$$

essent $c_l \in \mathbb{R}^{n_l}$ i $c_u \in \mathbb{R}^{n_u}$ els costos associats a cada grup de variables x_l i x_u . Clarament, com major sigui el nombre de variables afitades superiorment, menor serà el nombre d'infactibilitats duals (en el cas extrem, si totes les variables estan afitades superiorment el punt inicial escollit serà factible dual, donat que sempre es poden trobar uns z_u i w escaients que verifiquin (8.9)). Fins i tot, per al cas de variables x_l amb costos $c_l > 0$ es podrà continuar garantint la factibilitat dual, donat que es pot inicialitzar z_l amb c_l , essent un punt interior que verifica (8.8).

Tal i com mostra (8.4), les folgues de capacitat mútua pertanyen al grup de variables x_l (només tenen fita inferior igual a 0), i els seus costos lineals associats són 0 (no positius). Això fa que, segons el dit abans, es violi la factibilitat dual del punt inicial. A més, donat que el nombre de folgues de capacitat mútua és n (n'hi ha una per constricció de capacitat

mútua de cada arc de la xarxa), el nombre d'infactibilitats duals serà elevat. Per tal d'evitar aquest problema, pot deduir-se de l'expressió de les constriccions a banda, sabent que els fluxos als arcs són sempre positius, el següent resultat:

$$\left. \begin{array}{l} \sum_{i=1}^k x^{(i)} + s_{cm} = b_{cm} \\ \bar{0} \leq x^{(i)} \end{array} \right\} \Rightarrow \bar{0} \leq s_{cm} \leq b_{cm} \quad (8.10)$$

i d'aquesta forma poden considerar-se les folgues s_{cm} com variables afitades superiorment, eliminant les infactibilitats duals del punt inicial associades a les folgues de les constriccions de capacitat mútua. Cal indicar que, per a l'arc arrel (el n -èssim arc de la xarxa), es considera també una constricció de capacitat mútua associada. En la versió actual de la implementació això ha estat tractat considerant un límit fictici (ni molt gran ni gaire petit) de la component b_{cm_n} . S'observa que aquí s'ha operat de forma diferent al cas dels arcs arrel, donat que allà si que van ser tractats de forma especial, en comptes de fixar-los uns límits superiors. En aquest cas, però, això no ha provocat cap mena de problemes d'estabilitat numèrica, i s'ha preferit fer-ho per evitar tractar casos particulars, tot simplificant la implementació.

8.1.3 Tractament de les folgues de les constriccions a banda.

A la formulació presentada del problema, totes les folgues de les constriccions a banda es troben afitades superiorment a l'equació (8.5). Això, però, no té per què ser així: podem disposar de constriccions a banda només afitades superiorment (amb el qual la folga associada serà del tipus $s_{cb_j} \leq 0$), i d'altres d'igualtat (amb una folga $s_{cb_j} = 0$, amb el qual pot ser eliminada del conjunt de variables). Aleshores es consideraran tres tipus de constriccions a banda, amb les seves folgues associades, que podem escriure com:

$$\left. \begin{array}{l} \sum_{i=1}^k T_u^{(i)} + s_{cb_u} = \bar{b}_{cb_u} \\ 0 \leq s_{cb_u} \leq \bar{b}_{cb_u} - \underline{b}_{cb_u} \end{array} \right\} \quad (8.11)$$

$$\left. \begin{array}{l} \sum_{i=1}^k T_l^{(i)} + s_{cb_l} = \bar{b}_{cb_l} \\ 0 \leq s_{cb_l} \end{array} \right\} \quad (8.12)$$

$$\sum_{i=1}^k T_e^{(i)} = \bar{b}_{cb_e} \quad (8.13)$$

essent $T_u^{(i)} \in \mathbb{R}^{t_u \times n}$ la submatriu de $T^{(i)}$ associada a les t_u constriccions a banda afitades superior i inferiorment (amb folgues $s_{cb_u} \in \mathbb{R}^{t_u}$), $T_l^{(i)} \in \mathbb{R}^{t_l \times n}$ la submatriu de $T^{(i)}$ associada a les constriccions només afitades superiorment (amb folgues $s_{cb_l} \in \mathbb{R}^{t_l}$), i $T_e^{(i)} \in \mathbb{R}^{t_e \times n}$ la submatriu de $T^{(i)}$ relativa a les constriccions a banda d'igualtat (es verifica, doncs, que $t = t_u + t_l + t_e$, on t denotava el nombre total de constriccions a banda). Així mateix, el vector de fites superiors de les constriccions a banda \bar{b}_{cb} es descomposa segons

el tipus de constricció a banda de forma $\bar{b}_{cb}^t = [\bar{b}_{cb_u}^t | \bar{b}_{cb_l}^t | \bar{b}_{cb_e}^t]$, mentre que, pel que fa al vector de fites inferiors, només existeix el vector associat a les restriccions T_u , denotat per \underline{b}_{cb_u} .

8.1.4 Formulació del problema.

Un cop detallat el tractament específic que s'ha fet per a cada tipus de variable, el problema finalment solucionat pot ser escrit detallant clarament la part associada a variables afitades superior i inferiorment (x_u), i la part de variables només afitades inferiorment (x_l), tal i com segueix:

$$\min_{x^{(1)}, \dots, x^{(k)}, s_{cm}, s_{cb}} \sum_{i=1}^k c_{\bar{a}}^{(i)t} x_{\bar{a}}^{(i)} \tag{8.14}$$

subj. a

$A_{\bar{a}}$...	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	+	e_1				=	$x_{\bar{a}}^{(1)}$				$b^{(1)}$
\vdots	\ddots	\vdots	\vdots	\vdots		\vdots		\ddots				$x_{\bar{a}}^{(k)}$	$b^{(k)}$		
$\mathbf{0}$...	$A_{\bar{a}}$	$\mathbf{0}$	$\mathbf{0}$			e_1			$x_{\bar{a}}^{(k)}$	b_{cm}				\bar{b}_{cb_u}
$\mathbb{1}_{n-1}$...	$\mathbb{1}_{n-1}$	$\mathbb{1}_n$	$\mathbf{0}$					$\mathbf{0}$	s_{cm}					\bar{b}_{cb_l}
$T_u^{(1)}$...	$T_u^{(k)}$	$\mathbf{0}$	$\mathbb{1}_{t_u}$					$\mathbf{1}_{t_l}$	s_{cb_l}					\bar{b}_{cb_e}
$T_l^{(1)}$...	$T_l^{(k)}$	$\mathbf{0}$	$\mathbf{0}$					$\mathbf{0}$						
$T_e^{(1)}$...	$T_e^{(k)}$	$\mathbf{0}$	$\mathbf{0}$					$\mathbf{0}$						

$$\left. \begin{aligned} \underline{0} \leq x_{\bar{a}}^{(i)} \leq \bar{x}_{\bar{a}}^{(i)} \quad \forall i = 1, \dots, k \\ \underline{0} \leq s_{cm} \leq b_{cm} \\ \underline{0} \leq s_{cb_u} \leq \bar{b}_{cb_u} - \underline{b}_{cb_u} \end{aligned} \right\} x_u \tag{8.16}$$

$$\left. \begin{aligned} \underline{0} \leq x_{\bar{a}}^{(i)} \\ \underline{0} \leq s_{cb_l} \end{aligned} \right\} x_l \tag{8.17}$$

S'observa com clarament el problema així formulat s'ajusta perfectament al problema (7.2) solucionat al capítol anterior, quedant perfectament definides les matrius A_u i A_l , i els vectors x_u i x_l a (8.15). Aleshores el mètode primal-dual descrit al capítol anterior pot ser aplicat al problema (FMLCB), un cop formulat de la forma anterior. Les seccions següents descriuran l'adaptació de l'algorisme general a la solució de (8.14–8.17).

8.2 Solució de (FMLCB) a través de l'algorisme primal-dual.

Tal i com es va esmentar al capítol precedent, l'algorisme primal-dual presentat es basava en l'obtenció a cada iteració de direccions factibles i de descens mitjançant la resolució del sistema (7.39–7.43). La part més costosa consistia, precisament, en la solució

de (7.39), on s'obtenia la direcció dy a través de la factorització de la matriu $\tilde{A}S\tilde{A}^t$. En aplicar aquest mètode primal-dual al nostre problema multiarticle definit per (8.14–8.17), pot explotar-se l'estructura del mateix a qualsevol de les etapes de l'algorisme. Tanmateix, cal centrar-se de forma particular en la resolució de (7.39), donat que és el punt més costós de l'algorisme, i és on pot obtenir-se un major benefici de l'especialització que es faci. El càlcul de dy implica la solució d'un sistema d'equacions amb la matriu $\tilde{A}S\tilde{A}^t$, com ja s'ha dit anteriorment. Caldrà, doncs, observar l'estructura particular d'aquesta matriu per al problema de fluxos multiarticle. En primer lloc, cal fer notar que la matriu S definida a (7.45) pot ser particionada com:

$$S = \begin{array}{|c|} \hline S^{(1)} \\ \hline \vdots \\ \hline S^{(k)} \\ \hline S_{cm} \\ \hline S_{cb} \\ \hline \end{array} \quad (8.18)$$

on cada $S^{(i)} \in \mathbb{R}^{n \times n}$ es troba associada als arcs de l'article i , $S_{cm} \in \mathbb{R}^{n \times n}$ està associada a les folgues de les constriccions de capacitat mútua, i $S_{cb} \in \mathbb{R}^{(t_u+t_l) \times (t_u+t_l)}$ té relació amb les folgues de les constriccions a banda de desigualtat. La partició de S en S_u i S_l , tal i com es definirà a (7.45), ve donada per:

$$S_u = \begin{array}{|c|} \hline S_{\bar{a}}^{(1)} \\ \hline \vdots \\ \hline S_{\bar{a}}^{(k)} \\ \hline S_{cm} \\ \hline S_{cb_u} \\ \hline \end{array} \quad (8.19)$$

$$S_l = \begin{array}{|c|} \hline S_a^{(1)} \\ \hline \vdots \\ \hline S_a^{(k)} \\ \hline S_{cb_l} \\ \hline \end{array} \quad (8.20)$$

on cada $S_{\bar{a}}^{(i)} \in \mathbb{R}^{(n-1) \times (n-1)}$ es troba associada als arcs no-arrel, els $S_a^{(i)} \in \mathbb{R}$ són valors escalars associats a l'arc arrel de cada article, $S_{cb_u} \in \mathbb{R}^{t_u \times t_u}$ té relació amb les folgues de les constriccions a banda afitades superior i inferiorment, mentre que $S_{cb_l} \in \mathbb{R}^{t_l \times t_l}$ està associada a les folgues de les constriccions a banda afitades superiorment només. Tenint

en compte l'estructura de la matriu de constriccions \tilde{A} del problema multiarticle i de la matriu diagonal S definida a (8.18), la matriu $\tilde{A}S\tilde{A}^t$ pot ser particionada com segueix:

$$\tilde{A}S\tilde{A}^t = \begin{array}{c|c|c|c|c} AS^{(1)}A^t & \dots & \mathbf{0} & AS^{(1)} & AS^{(1)}T^{(1)t} \\ \hline \vdots & \ddots & \vdots & \vdots & \vdots \\ \hline \mathbf{0} & \dots & AS^{(k)}A^t & AS^{(k)} & AS^{(k)}T^{(k)t} \\ \hline S^{(1)}A^t & \dots & S^{(k)}A^t & S_{cm} + \sum_{i=1}^k S^{(i)} & \sum_{i=1}^k S^{(i)}T^{(i)t} \\ \hline T^{(1)}S^{(1)}A^t & \dots & T^{(k)}S^{(k)}A^t & \sum_{i=1}^k T^{(i)}S^{(i)} & \hat{S}_{cb} + \sum_{i=1}^k T^{(i)}S^{(i)}T^{(i)t} \end{array}$$

$$= \begin{array}{c|c|c|c|c} G^{(1)} & \dots & \mathbf{0} & C_1^{(1)} & C_2^{(1)} \\ \hline \vdots & \ddots & \vdots & \vdots & \vdots \\ \hline \mathbf{0} & \dots & G_k & C_1^{(k)} & C_2^{(k)} \\ \hline C_1^{(1)t} & \dots & C_1^{(k)t} & D_1 & D_2 \\ \hline C_2^{(1)t} & \dots & C_2^{(k)t} & D_3 & D_4 \end{array} = \begin{array}{c|c|c} G & C_1 & C_2 \\ \hline C_1^t & & \\ \hline C_2^t & & \end{array} = \begin{array}{c|c} G & C \\ \hline C^t & D \end{array} \quad (8.21)$$

on $\hat{S}_{cb} \in \mathbb{R}^{t \times t}$ es defineix com:

$$\hat{S}_{cb} = \begin{array}{c|c} & \\ \hline S_{cb} & \\ \hline & \mathbf{0}_{t_e} \end{array} \quad (8.22)$$

A l'expressió (8.21) pot comprovar-se que la matriu $D_1 \in \mathbb{R}^{n \times n}$ és diagonal (donat que no és més que un sumatori de matrius diagonals). Així mateix, el càlcul de la resta de submatrius pot ser fet directament (exceptuant aquelles on apareixen les matrius $T^{(i)}$, on cal tenir en compte l'estructura particular de les constriccions a banda). En el cas de $C_1^{(i)} = AS^{(i)}$ (i la seva transposada $C_1^{(i)t} = S^{(i)}A^t$) el càlcul es redueix a un simple escalat de les columnes (associades a arcs) de la matriu A . Pel que fa a $G^{(i)} = AS^{(i)}A^t \in \mathbb{R}^{m \times m}$, $\forall i = 1, \dots, k$, cada un dels k blocs té el mateix patró d'esparsitat (o mateixa topologia), i els elements no-zero només estan associats a parells de nodes comunicats per un o més arcs. Denotant per \mathcal{E} el conjunt d'arcs de la xarxa, i per \mathcal{I}_v el conjunt d'arcs incidents al node v , aleshores $AS^{(i)}A^t$ pot ser calculat com segueix:

$$AS^{(i)}A^t = \left(G_{vw}^{(i)} \right)_{\substack{v=1, \dots, m \\ w=1, \dots, m}} = \begin{cases} \sum_{\forall a} -S_a^{(i)} & \text{si } a \equiv (v, w) \in \mathcal{E} \text{ i } (w, v) \notin \mathcal{E} \\ \sum_{\forall a, b} (-S_a^{(i)} - S_b^{(i)}) & \text{si } a \equiv (v, w) \in \mathcal{E} \text{ i } b \equiv (w, v) \in \mathcal{E} \\ \sum_{\forall a \in \mathcal{I}_v} S_a^{(i)} & \text{si } (v = w) \\ 0 & \text{altrament} \end{cases} \quad (8.23)$$

on $S_a^{(i)}$ denota el valor diagonal de $S^{(i)}$ associat a l'arc a .

La matriu $\tilde{A}S\tilde{A}^t$ és, clarament, simètrica i, a més, definida positiva, donat que els elements diagonals de S verifiquen que $S_{ii} > 0$ (això ja va ser justificat al capítol 7 basant-nos en el fet de que en tot moment els punts iterats són interiors respecte les seves fites). Aleshores és possible solucionar el sistema (7.39) a través d'una factorització de Cholesky de $\tilde{A}S\tilde{A}^t$. Tanmateix, en realitzar aquesta factorització s'obté que la matriu D_1 esdevé completament densa. Aquest resultat ve donat per la següent proposició.

Proposició 8.1.

Als efectes de demostrar que la part de $\tilde{A}S\tilde{A}^t$ associada a la matriu D_1 esdevé densa en factoritzar $\tilde{A}S\tilde{A}^t$, basta amb demostrar-ho per a la següent submatriu corresponent a només un qualsevol dels articles:

$$\begin{array}{|c|c|} \hline AS^{(i)}A^t & C_1^{(i)} \\ \hline C_1^{(i)t} & D_1 \\ \hline \end{array} \tag{8.24}$$

Essent $L_iL_i^t = AS^{(i)}A^t$ la descomposició de Cholesky de $AS^{(i)}A^t$, l'eliminació gaussiana dels elements subdiagonals de $\begin{bmatrix} AS^{(i)}A^t \\ C_1^{(i)t} \end{bmatrix}$ pot ser realitzada premultiplicant (8.24) per la matriu de triangularització, de forma que:

$$\begin{array}{|c|c|} \hline L_i^{-1} & \mathbf{0} \\ \hline C_1^{(i)t}(L_iL_i^t)^{-1} & \mathbf{1} \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline AS^{(i)}A^t & C_1^{(i)} \\ \hline C_1^{(i)t} & D_1 \\ \hline \end{array} = \begin{array}{|c|c|} \hline L_i^t & L_i^{-1}C_1^{(i)} \\ \hline \mathbf{0} & D_1 - C_1^{(i)t}(L_iL_i^t)^{-1}C_1^{(i)} \\ \hline \end{array}$$

Aleshores la submatriu resultant $D_1 - C_1^{(i)t}(L_iL_i^t)^{-1}C_1^{(i)}$ és totalment densa.

Demostració.

Sigui \mathcal{N}_l el conjunt de nodes adjacents al node l qualsevol, \mathcal{I}_l el conjunt d'arcs incidents al mateix node, i \mathcal{C}_a el conjut de nodes connectats a l'arc a (inicialment la cardinalitat de \mathcal{C}_a és sempre 2). En realitzar la descomposició de Cholesky de $\tilde{A}S\tilde{A}^t$ cal anul·lar els termes subdiagonals de $AS^{(i)}A^t$ a partir dels valors diagonals. Suposem que ens trobem tractant la primera fila de $AS^{(i)}A^t$ (associada al node 1). Llavors s'hauran d'anul·lar els termes de les files associades amb els nodes $\in \mathcal{N}_1$, creant-se nous elements no-zero dins $AS^{(i)}A^t$ associats amb les parelles de nodes de \mathcal{N}_1 que prèviament no estaven connectades (una verificació exhaustiva d'aquest resultat pot ser trobat a [29]). D'igual forma, nous elements no-zero són creats dins la matriu $C_1^{(i)} = AS^{(i)}$. Inicialment, a la fila l -èsima de $AS^{(i)}$ els únics elements no-zero apareixien a les columnes associades als arcs incidents \mathcal{I}_l . Després de tractar la primera fila de $AS^{(i)}A^t$ es verificarà que $\forall j \in \mathcal{N}_1 \quad \mathcal{I}_j := \mathcal{I}_j \cup \mathcal{I}_1$, havent-se creat nous elements associats als arcs de \mathcal{I}_1 . El mateix ocorre a la matriu $C_1^{(i)t} = S^{(i)}A^t$, però aquí els nous elements creats es troben associats a arcs, verificant-se que $\forall i \in \mathcal{I}_1 \quad \mathcal{C}_i := \mathcal{C}_i \cup \mathcal{N}_1$ (és a dir, es creen nous elements associats als nodes de \mathcal{N}_1).

Pot observar-se que aquesta creació d'elements no zero a $AS^{(i)}$ i $S^{(i)}A^t$ preserva que una matriu és transposada de l'altra.

El procediment anteriorment descrit només s'ha referit a la primera fila de $AS^{(i)}A^t$. Aplicant repetidament el procés anterior per a cada fila de $AS^{(i)}A^t$ (en total m vegades, una per cada node de la xarxa), resulta que, en el moment de tractar la fila m de $AS^{(i)}A^t$, es verificarà que la darrera fila de $AS^{(i)}$ és densa, donat que es tindrà que $\mathcal{I}_m := \bigcup_{i=1}^m \mathcal{I}_i$. Anàlogament, la darrera columna de $S^{(i)}A^t$ també esdevindrà densa. Directament es té, aleshores, que en anullar els elements de la darrera columna de $S^{(i)}A^t$, es crearan a cada fila de D_1 tants elements no-zero (exceptuant la diagonal, que ja existeix) com elements hi hagi dins la darrera fila de $AS^{(i)}$. Donat que aquesta darrera fila és densa, llavors es té que $D_1 - C_1^{(i)t}(L_iL_i^t)^{-1}C_1^{(i)}$ esdevindrà totalment densa. ■

El fet de que D_1 esdevingui una matriu densa si es realitza la factorització de Cholesky de $\tilde{A}S\tilde{A}^t$ implica que caldria emmagatzemar i tractar un total de $n(n+1)/2$ valors (només la meitat dels elements per la simetria de la descomposició), on n denota el nombre d'arcs de la xarxa en qüestió. Per a xarxes grans aquesta quantitat de memòria esdevé prohibitiva, i provocaria un alentiment considerable de l'algorisme. Aquest fet és també destacat per Choi i Goldfarb a [18], on es presenta un mètode de punt interior equivalent per solucionar problemes multiarticle, la solució del qual també requereix el tractament d'una matriu densa de dimensió nombre d'arcs. En aquell article, però, no es presenta cap procediment per solventar aquest problema, i només s'esmenta que caldrien ser usades rutines de descomposició amb processament vectorial i paral·lel. Tanmateix, pot aprofitar-se l'estructura de $\tilde{A}S\tilde{A}^t$ per tal d'agilitzar la resolució del sistema d'equacions plantejat, tal i com es detalla a les subseccions següents.

8.2.1 Aplicació de la descomposició de dominis a la factorització de $\tilde{A}S\tilde{A}^t$.

Tenint en compte la partició de $\tilde{A}S\tilde{A}^t$ presentada a (8.21), el sistema d'equacions a solucionar per tal de calcular dy pot ser escrit de forma genèrica com:

$$\begin{array}{|c|c|} \hline G & C \\ \hline C^t & D \\ \hline \end{array} \begin{array}{|c|} \hline dy_1 \\ \hline dy_2 \\ \hline \end{array} = \begin{array}{|c|} \hline \beta_1 \\ \hline \beta_2 \\ \hline \end{array} \tag{8.25}$$

on $dy_1 \in \mathbb{R}^{kn}$, $dy_2 \in \mathbb{R}^{n+t}$, $\beta_1 \in \mathbb{R}^{kn}$, i $\beta_2 \in \mathbb{R}^{n+t}$ (recordem que $\beta^t = (\beta_1^t \ \beta_2^t)$ en aquest cas és el terme independent de l'equació (7.39)). Multiplicant per blocs s'obté:

$$Gdy_1 + Cdy_2 = \beta_1 \tag{8.26}$$

$$C^tdy_1 + Ddy_2 = \beta_2 \tag{8.27}$$

Tenint en compte que G és una matriu simètrica i definida positiva (donat que és una matriu diagonal per blocs, on cada bloc $AS^{(i)}A^t$ és clarament simètric, i definit positiu ja

que els elements diagonals de $S^{(i)}$ són positius), i per tant, invertible, dy_1 pot ser aïllada de (8.26), de forma que

$$dy_1 = G^{-1}(\beta_1 - Cdy_2) \quad (8.28)$$

Substituint (8.28) a (8.27) s'obté que:

$$(D - C^t G^{-1} C)dy_2 = \beta_2 - C^t G^{-1} \beta_1 \quad (8.29)$$

Solucionant consecutivament (8.29) i (8.28) obtenim la solució del nostre sistema d'equacions. Aquesta tècnica de descomposició de sistemes d'equacions, anomenada *descomposició de dominis*, s'usa comunment en la solució de diversos problemes d'equacions diferencials, tal i com es detalla a [33,64]. Així mateix, la matriu $D - C^t G^{-1} C$ s'anomena el *complement de Schur*, i, si G i $\tilde{A}S\tilde{A}^t$ són no singulars, és sempre una matriu invertible. Pot comprovar-se la similitud que hi ha entre aquest procediment per obtenir la solució d'aquest sistema (que ens proporciona la direcció dy) i la forma de solucionar sistemes amb la base del mètode del particionament primal descrit al capítol 4 (en aquell cas es va anomenar matriu de treball al complement de Schur). S'observa, doncs, com s'arriba a usar la mateixa tècnica en solucionar problemes de fluxos multiarticle per dos vies completament diferents (aquest fet es deu a que l'estructura dels problemes multiarticle és escaient per a l'ús d'aquests tipus de descomposicions).

Cal observar que per solucionar el sistema (8.25) només cal solucionar un sistema amb la matriu $D - C^t G^{-1} C$, i cal resoldre sistemes amb la matriu G . Aquests darrers, però, no presenten una dificultat excessiva, i poden ser resolts explotant l'estructura diagonal per blocs de G , de forma que:

$$G\gamma = \theta \Leftrightarrow G^{(i)}\gamma^{(i)} = \theta^{(i)} \quad \forall i = 1, \dots, k \Leftrightarrow (AS^{(i)}A^t)\gamma^{(i)} = \theta^{(i)} \quad \forall i = 1, \dots, k \quad (8.30)$$

Per tant, cal fer la factorització de Cholesky de les k matrius $AS^{(i)}A^t$, calculades prèviament segons (8.23). Per tal d'evitar la creació de nous elements no-zero en fer la factorització cal usar una reordenació simètrica de les files i columnes de $AS^{(i)}A^t$, usant, per ex., l'algorisme del mínim grau [26] (tal i com es va indicar al capítol 7). En aquest sentit, s'ha usat la rutina GENQMD de la llibreria SPARSPAK, la qual implementa l'esmentat algorisme. A més, cal observar que el patró d'esparsitat de les k matrius $AS^{(i)}A^t$ és el mateix, amb el qual només caldrà calcular un sol cop la reordenació de files i columnes (només es cridarà una vegada a la rutina GENQMD), amb el consegüent estalvi de temps.

Queda clar, doncs, que el càlcul de $G^{-1}(\beta_1 - Cdy_2)$ de (8.28) i de $G^{-1}\beta_1$ de (8.29) no suposa un cost excessiu. Tanmateix, en calcular dy_2 per (8.29), caldria formar la matriu $D - C^t G^{-1} C$, el qual implica solucionar $n + t$ sistemes de equacions per obtenir $G^{-1}C$ i fer una posterior descomposició de Cholesky de $D - C^t G^{-1} C$. Aquest procediment, però, té un cost molt elevat, i cal trobar una alternativa més eficient. Aquesta ve donada per l'ús del mètode del gradient conjugat amb preconditionament, tal i com es mostra al següent apartat.

8.2.2 Gradient conjugat preconditionat.

El mètode del gradient conjugat (amb o sense preconditionament) s'usa en la solució de sistemes del tipus $\hat{D}\gamma = \theta$, essent \hat{D} simètrica i definida positiva. Per tal d'aplicar aquesta tècnica a la nostra matriu $D - C^t G^{-1} C$ per a la resolució de (8.29), caldrà verificar que és simètrica i definida positiva. Això ens ho garanteix la següent proposició.

Proposició 8.2.

Sigui la matriu qualsevol

$$\hat{A} = \begin{array}{|c|c|} \hline G & C \\ \hline C^t & D \\ \hline \end{array}$$

Lavors es verifica que si \hat{A} és simètrica i definida positiva i G és definida positiva aleshores $D - C^t G^{-1} C$ és simètrica i definida positiva.

Demostració.

i) $D - C^t G^{-1} C$ és simètrica.

Donat que $\hat{A} = \hat{A}^t$, llavors es té que $G = G^t$ i que $D = D^t$. Aleshores es verifica que $(D - C^t G^{-1} C)^t = D^t - (C^t G^{-1} C)^t = D - C^t (G^{-1})^t C = D - C^t (G^t)^{-1} C = D - C^t G^{-1} C$, i, per tant, $D - C^t G^{-1} C$ és simètrica.

ii) $D - C^t G^{-1} C$ és definida positiva.

Considerem un vector $x_2 \neq 0$ qualsevol i un altre vector x_1 de forma que $x_1 = -G^{-1} C x_2$. Donat que \hat{A} és definida positiva llavors es verifica que

$$\begin{aligned} 0 &< (x_1^t x_2^t) \hat{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &\Downarrow \\ 0 &< x_1^t G x_1 + 2x_1^t C x_2 + x_2^t D x_2 = \\ &= x_2^t C^t G^{-1} G G^{-1} C x_2 - 2x_2^t C^t G^{-1} C x_2 + x_2^t D x_2 \quad [\text{aplicant que } x_1 = -G^{-1} C x_2] \\ &= x_2^t C^t G^{-1} C x_2 - 2x_2^t C^t G^{-1} C x_2 + x_2^t D x_2 \\ &= x_2^t (D - C^t G^{-1} C) x_2 \end{aligned}$$

Per tant, s'obté que $\forall x_2 \neq 0, x_2^t (D - C^t G^{-1} C) x_2 > 0$, i, per tant, $D - C^t G^{-1} C$ és definida positiva. ■

Donat que al nostre cas el sistema a solucionar és $\tilde{A} \tilde{S} \tilde{A}^t dy = \beta$, on $\tilde{A} \tilde{S} \tilde{A}^t$ es troba particionada tal i com es mostra a (8.21), es verifiquen les hipòtesis de P8.2 ($\tilde{A} \tilde{S} \tilde{A}^t = \hat{A}$ és simètrica i definida positiva, i G és definida positiva), amb el qual es té que $D - C^t G^{-1} C$ és simètrica i definida positiva.

En funció del dit anteriorment pot concloure's que el sistema (8.29):

$$(D - C^t G^{-1} C) dy_2 = \beta_2 - C^t G^{-1} \beta_1$$

pot ser solucionat mitjançant l'algorisme del gradient conjugat preconditionat. Si escrivim el sistema anterior com

$$\begin{aligned} \hat{D}dy_2 &= \hat{\beta}_2 \\ \text{on } \hat{D} &= D - C^t G^{-1} C \quad (\hat{D} \in \mathbb{R}^{n+t}) \\ \text{i } \hat{\beta}_2 &= \beta_2 - C^t G^{-1} \beta_1 \quad (\hat{\beta}_2 \in \mathbb{R}^{n+t}) \end{aligned} \quad (8.31)$$

aleshores el mètode del gradient conjugat preconditionat es basa en la solució de:

$$M^{-1} \hat{D}dy_2 = M^{-1} \hat{\beta}_2 \quad (8.32)$$

on M és una matriu definida positiva, de forma que la matriu del sistema $M^{-1} \hat{D}$ estigui més ben condicionada i s'acceleri la convergència (nombre d'iteracions) que es tindria usant simplement el mètode del gradient conjugat sense preconditionament (aquest es dona quan $M = \mathbb{1}$). El següent algorisme presenta en forma de pseudo-codi el mètode del gradient conjugat amb preconditionament:

Algorisme A8.1. Solució de $\hat{D}dy_2 = \hat{\beta}_2$ pel mètode del gradient conjugat preconditionat.

Inicialitzacions:

$$\begin{aligned} dy_{2_0} &:= 0 \\ r_0 &:= \hat{\beta}_2 \\ z_0 &:= M^{-1} r_0 \\ p_0 &:= z_0 \\ i &:= 0 \end{aligned}$$

mentre no_hi_ha_convergència fer

$$\begin{aligned} q_i &:= \hat{D}p_i \\ \alpha_i &:= z_i^t r_i / p_i^t q_i \\ dy_{2_{i+1}} &:= dy_{2_i} + \alpha_i p_i \\ r_{i+1} &:= r_i - \alpha_i q_i \\ z_{i+1} &:= M^{-1} r_{i+1} \\ \beta_i &:= z_{i+1}^t r_{i+1} / z_i^t r_i \\ p_{i+1} &:= z_{i+1} + \beta_i p_i \\ i &:= i + 1 \end{aligned}$$

fi_mentre

$$dy_2 := dy_{2_i}$$

S'observa a l'algorisme previ com els passos més costosos consisteixen en i) el producte de \hat{D} amb un vector ($q_i := \hat{D}p_i$), i ii) la solució, a la inicialització de variables i posteriorment a cada iteració, d'un sistema amb la matriu M ($z_i := M^{-1}r_i$, $i \geq 0$). Passem tot seguit a comentar breument ambdós casos.

El primer dels dos càlculs anteriors pot fer-se seguint la seqüència detallada al següent algorisme:

Algorisme A8.2. Producte de $\hat{D}=D-C^tG^{-1}C$ per un vector $\hat{x} \in \mathbb{R}^{n+t}$ ($\hat{y}:=\hat{D}\hat{x}$).

1. $v_1 := D\hat{x}$ $v_1 \in \mathbb{R}^{n+t}$
2. $u_1 := C\hat{x}$ $u_1 \in \mathbb{R}^{km}$
3. $u_2 := G^{-1}u_1$ $u_2 \in \mathbb{R}^{km}$, calculat tal i com segueix:

$$\text{per } i:=1 \text{ fins } k \text{ fer}$$

$$u_2^{(i)} := G^{(i)-1}u_1^{(i)} = (AS^{(i)}A^t)^{-1}u_1^{(i)} \quad u_1^{(i)}, u_2^{(i)} \in \mathbb{R}^m$$

$$\text{fi_per}$$
4. $v_2 := C^t u_2$ $v_2 \in \mathbb{R}^{n+t}$
5. $\hat{y} := v_1 - v_2$

Cal observar que per multiplicar \hat{D} per un vector cal solucionar k sistemes amb les matrius $AS^{(i)}A^t$. Tanmateix, tal i com va ser dit a §8.2.1, la factorització de les matrius $AS^{(i)}A^t$ ja ha estat feta prèviament, i pot ser utilitzada en aquest cas.

El segon càlcul implica la solució d'un sistema d'equacions amb M a cada iteració de l'algorisme A8.1. Per tal d'agilitzar el procés cal que el sistema amb el preconditionador M pugui ser fet de forma eficient. Un dels preconditionadors més usats consisteix en una matriu diagonal d'escalat, i la solució del sistema $z_i := M^{-1}r_i$ és, aleshores, directament calculable amb $O(n)$ operacions aritmètiques. Al nostre cas, però, aquest preconditionador va ser assajat i, tot i que a les primeres iteracions del mètode primal-dual el nombre d'iteracions de l'algorisme A8.1 era petit (per tant es comportava com un bon preconditionador), ben aviat es degradava la seva eficiència, esdevenint cada iteració de l'algorisme primal-dual considerablement més costosa (es requerien moltes iteracions de A8.1 per obtenir la direcció dy_2). Aquest mateix comportament del preconditionador diagonal ha estat observat per Resende et al. a [66,68]. Cal, aleshores, trobar un bon preconditionador per tal de poder solucionar eficientment (8.29). El següent apartat presenta un preconditionador amb el qual s'han obtingut uns bons resultats.

8.2.3 Precondicionament per sèries truncades de potències.

Tot i que el desenvolupament del mètode primal-dual per a problemes multiarticle detallat fins aquest moment ha tingut en compte constriccions a banda en la seva formulació, el preconditionador presentat en aquest apartat només considera el cas de problemes multiarticle amb constriccions de capacitat mútua, i només ha estat comprovat computacionalment usant aquest tipus de problemes. El preconditionament del sistema (8.29), considerant constriccions a banda, hauria de tenir en compte l'estructura particular d'aquestes, la qual pot ser diferent per a cada problema a tractar. Trobar un bon preconditionador esdevé, en aquest cas, una tasca d'una complexitat molt superior a només preconditionar

el sistema considerant únicament constriccions de capacitat mútua, i per aquest motiu ha quedat fora de l'abast d'aquest treball.

En funció del dit anteriorment, queda clar que el sistema a solucionar en aquest cas serà (en comptes de l'original (8.29)):

$$(D_1 - C_1^t G^{-1} C_1) dy_2 = \beta_2 - C_1^t G^{-1} \beta_1 \quad (8.33)$$

on D_1 , C_1 i G han estat definits a (8.21), i les dimensions dels vectors dy_2 i β_2 passen de $n + t$ a n ($dy_2, \beta_2 \in \mathbb{R}^n$), donat que $t = 0$ (no es considera el cas amb constriccions a banda). Tal i com es va fer a (8.31), podem escriure (8.33) com

$$\begin{aligned} \hat{D}_1 dy_2 &= \hat{\beta}_2 \\ \text{on } \hat{D}_1 &= D_1 - C_1^t G^{-1} C_1 \quad (\hat{D}_1 \in \mathbb{R}^n) \\ \text{i } \hat{\beta}_2 &= \beta_2 - C_1^t G^{-1} \beta_1 \quad (\hat{\beta}_2 \in \mathbb{R}^n) \end{aligned} \quad (8.34)$$

i la matriu a preconditionar serà \hat{D}_1 .

Cal recordar que, tal i com es va dir a §8.2.2, l'objectiu del preconditionador era obtenir matrius “més ben condicionades”. En aquest cas “més ben condicionada” significa que els valors propis de $M^{-1} \hat{D}_1$ estiguin propers entre ells. En aquest sentit pot veure's que \hat{D}_1^{-1} és un bon preconditionador, tot i que, òbviament, seria absurd usar-lo (si el tinguéssim solucionaríem (8.33) directament). En comptes d'això, però, la idea a seguir es basarà en l'ús d'una aproximació de \hat{D}_1^{-1} , obtinguda mitjançant una sèria truncada de potències. L'obtenció d'aquest preconditionador, que a continuació serà descrit, es recolza fortament en els dos teoremes següents (les demostracions dels quals poden ser trobades a [64]):

Teorema 8.1 : Sigui R una matriu quadrada no singular, i $R = P - Q$ un desdoblament d'aquesta, de forma que P sigui no singular i que el radi espectral de $P^{-1}Q$ (és a dir, el major valor propi en mòdul) verifiqui que $\rho(P^{-1}Q) < 1$. Aleshores, anomenant $H = P^{-1}Q$, tenim que:

$$R^{-1} = \left(\sum_{i=0}^{\infty} H^i \right) P^{-1} \quad (8.35)$$

Teorema 8.2 : Sigui $R = P - Q$, de forma que R i P són simètriques i definides positives. Aleshores $\rho(P^{-1}Q) < 1$ si i només si $P + Q$ és definida positiva.

Queda clar que el que pretenem és usar l'expressió (8.35) per tal de tenir una forma d'aproximar \hat{D}_1^{-1} . El desdoblament de \hat{D}_1 que es considerarà (tenint en compte (8.34)) serà:

$$\hat{D}_1 = P - Q, \quad \text{on} \quad \begin{cases} P = D_1 \\ Q = C_1^t G^{-1} C_1 \end{cases} \quad (8.36)$$

Aleshores, per tal de poder aplicar el teorema T8.1, caldrà que el desdoblament anterior de \hat{D}_1 verifiqui les seves hipòtesis. En primer lloc, cal que \hat{D}_1 sigui matriu quadrada i no singular. Que és quadrada és evident, i per la proposició P8.2 podem garantir que \hat{D}_1 és no singular. La següent hipòtesi, que P sigui no singular, és clarament verificada per $P = D_1$, donat que D_1 és una matriu diagonal amb tots els seus elements majors que zero. La darrera hipòtesi, que el radi espectral de $P^{-1}Q$ sigui menor que 1, ens ve donada per la següent proposició.

Proposició 8.3.

La partició de la matriu \hat{D}_1 donada per $\hat{D}_1 = P - Q$, P i Q definits tal i com es mostra a (8.36), verifica que $\rho(P^{-1}Q) < 1$.

Demostració.

La demostració es basarà en el resultat del teorema T8.2. Per poder-lo aplicar, però, cal que \hat{D}_1 i $P = D_1$ siguin simètriques i definides positives. Per la proposició P8.2 podem garantir que \hat{D}_1 és simètrica i definida positiva, i el mateix es pot dir de $P = D_1$, per ser D_1 una matriu diagonal amb tots els seus elements majors que 0. Aleshores, pel teorema T8.2, demostrar que $\rho(P^{-1}Q) < 1$ es redueix a veure que $P + Q$ es definida positiva, fet que es garantirà si $\forall x \in \mathbb{R}^n, x \neq 0, x^t(P + Q)x = x^tPx + x^tQx > 0$. Donat que P és definida positiva, es té que $x^tPx > 0$. Per la seva banda, per la definició de Q a (8.36), i definint $v = C_1x$, tenim que $x^tQx = x^tC_1^tG^{-1}C_1x = v^tG^{-1}v$. Donat que G (definida a 8.21) és una matriu diagonal per blocs, on cada bloc és de la forma $AS^{(i)}A^t$, amb S_i matriu diagonal amb termes diagonals majors que 0, directament es té que G és simètrica i definida positiva, i, aleshores, G^{-1} serà definida positiva. Per tant, si G^{-1} és definida positiva, es té que $x^tQx = v^tG^{-1}v > 0$, i, aleshores, $x^t(P + Q)x = x^tPx + x^tQx > 0$, quedant demostrat que $P + Q$ és definida positiva, i, per consegüent, que $\rho(P^{-1}Q) < 1$. ■

Un cop garantides totes les hipòtesis de T8.1, aplicant directament (8.35) s'obté que

$$\hat{D}_1^{-1} = \left(\sum_{i=0}^{\infty} (P^{-1}Q)^i \right) P^{-1} \tag{8.37}$$

i ara podem truncar aquesta sèrie en un cert terme (per exemple, al terme ϕ), el qual anomenarem grau del preconditionador, i aleshores la nostra matriu de preconditionament que aproximarà la inversa de \hat{D}_1 serà:

$$M^{-1} = (\mathbb{1} + (P^{-1}Q) + (P^{-1}Q)^2 + \dots + (P^{-1}Q)^{\phi-1}) P^{-1} \tag{8.38}$$

El càlcul que calia solucionar a cada iteració de l'algorisme del gradient conjugat preconditionat (A8.1) era (s'omet el subíndex i de z i r per simplificar la notació) $z := M^{-1}r$, i, considerant (8.38), aquest podrà ser obtingut aplicant el següent algorisme:

Algorisme A8.3. Solució de $z := M^{-1}r$

```

v := P-1r
z1 := v
per j=1 fins φ - 1 fer
    zj+1 := P-1Qzj + v
fi_per
z := zφ

```

Com major sigui el grau ϕ , millor serà l'aproximació que prenem de \hat{D}_1^{-1} i millor serà el nostre preconditionador (caldrà fer menys iteracions de l'algorisme del gradient conjugat amb preconditionament). Tanmateix, també caldrà fer més iteracions de A8.3 per calcular z . I cal recordar que a cada iteració cal fer el producte Qz_j , el qual implica, per la definició de Q a (8.36), solucionar un sistema amb G^{-1} , el qual, tot i que ja es tenen factoritzats cada un dels k blocs diagonals $AS^{(i)}A^t$ que la formen, pot alentir considerablement tot el procés. Cal observar, a més, que la partició escollida $\hat{D}_1 = P - Q$ ens permet solucionar els càlculs del tipus $P^{-1}r$ de forma molt eficient, donat que $P = D_1$ és una matriu diagonal.

Hi ha un aspecte força important que cal destacar del preconditionador presentat. Per tal d'obtenir-lo, en hem basat en el fet de que certes matrius ($P, P + Q, \dots$) eren definides positives. Això, tal i com s'ha comentat prèviament, quedava garantit gràcies a que les diferents submatrius que componen la nostra matriu S definida a (8.18) eren en tot moment definides positives. Tanmateix, a mida que, iterant amb l'algorisme primal-dual, ens atensem al punt òptim, certes components del vector de fluxos es trobaran més a prop d'alguna de les seves fites (indicant que ens estem apropant al vèrtex òptim). Llavors, tenint en compte la definició de S presentada a (7.45), pot observar-se que cada cop serà "menys definida positiva" (certs elements diagonals seran propers a 0), i, aleshores, la nostra matriu de preconditionament definida a (8.38) no serà una bona aproximació de la inversa de \hat{D}_1 , degradant-se la convergència del mètode del gradient conjugat preconditionat. Aquest fet ha estat comprovat experimentalment a les execucions realitzades, els resultats de les quals són presentades al capítol 9.

Fins aquí s'ha presentat com especialitzar el mètode-primal del capítol 7 per a la solució de problemes multiarticle. S'ha pogut observar com tots els esforços s'han centrat en calcular eficientment la direcció dy . Resumint tot el dit fins ara, es podria desglossar el càlcul anterior en la següent seqüència d'etapes:

- 1.) Calcular (tenint en compte una permutació de files (nodes) de A calculada a l'inici de l'algorisme primal-dual, i considerant (8.23)) les matrius $AS^{(i)}A^t$, $i = 1, \dots, k$.
- 2.) Realitzar la factorització de Cholesky de les matrius anteriorment calculades $L_i L_i^t = AS^{(i)}A^t$, $i = 1, \dots, k$.
- 3.) Càlcul de dy_2 , aplicant (8.29), usant l'algorisme del gradient preconditionat detallat a A8.1. Caldrà fer ús, aleshores, de la descomposició prèvia de les matrius $AS^{(i)}A^t$, de

l'algorisme A8.2 per fer productes de la matriu del sistema amb un vector qualsevol, i del preconditionament per sèries truncades de potències presentat a (8.38) (usant en aquest cas l'algorisme A8.3).

- 4.) Un cop calculat dy_2 , s'usarà aquest valor per obtenir dy_1 a través de (8.28).
- 5.) Finalment el nostre vector dy desitjat serà directament $dy = (dy_1^t, dy_2^t)^t$.

8.3 Implementació de l'algorisme primal-dual per a problemes multiarticle.

La implementació que s'ha realitzat de l'algorisme detallat a la secció anterior segueix bàsicament les idees de la implementació general presentada a §7.3, mirant en tot moment d'explotar l'estructura particular de la matriu de constriccions del problema (**FML**) (la implementació realitzada no contempla el cas de constriccions a banda, i per això no ens hem referit al problema (**FMLCB**)). L'elecció del punt inicial, el càlcul de les passes màximes α_p i α_d , i l'actualització del paràmetre μ de la barrera logarísmica, s'han fet tal i com es descriu a §7.3.

Pel que fa a les condicions d'aturada, però, aquestes han estat relaxades per evitar problemes de convergència a prop de l'òptim (donat que, en usar un gradient conjugat preconditionat, s'obtenen solucions dy aproximades, mentre que al capítol 7 es solucionava el sistema que proporciona dy de forma exacta). Els valors de ϵ_p , ϵ_d i ϵ_g usats per determinar la factibilitat primal, la factibilitat dual, i el *gap* dual (conceptes introduïts a les equacions (7.68–7.69)), són en aquest cas $\epsilon_p = 10^{-6}$, $\epsilon_d = 10^{-6}$ i $\epsilon_g = 10^{-5}$. Així mateix, a les execucions realitzades s'ha comprovat que la factibilitat dual sempre era assolida (la forma d'escollir el punt inicial ens facilita aquest fet), mentre que hi havia problemes per aconseguir la factibilitat primal (tot i que el *gap* dual ja s'hagués reduït per sota de ϵ_g). Això es així per la pròpia naturalesa del mètode primal-dual i la forma de solucionar dy a través d'un gradient conjugat preconditionat. Portugal et al. constaten el mateix fet a [65] en solucionar problemes de fluxos en xarxes (amb un sol article) usant una metodologia similar, anomenant-la aleshores mètode infactible-primal factible dual. És per això que, a la implementació aquí presentada, l'algorisme finalitzarà quan es satisfaci la factibilitat dual i el test del *gap* dual, permetent una petita violació de la factibilitat primal. Naturalment, degut a les relaxacions que s'han introduït a les condicions d'aturada, queda clar que el punt solució obtingut no serà exactament el veritable punt òptim, però sí que es trobarà molt a prop d'aquest. En aquest sentit, caldria introduir un procés final per, a partir del punt solució, identificar algun punt (o faceta) proper a ell que verifiqui les condicions d'optimalitat del problema multiarticle original, de forma anàloga a com es descriu a [67] per al cas de fluxos simples (amb un sol article).

Pel que fa al càlcul de dy , aquest ja ha estat extensament comentat a la secció anterior. Tanmateix, es farà un breu comentari sobre tres punts concrets:

- 1.- *Estructures de dades.* A §7.3.1 es van comentar breument les estructures de dades per emmagatzemar, calcular i factoritzar eficientment $\tilde{A}S\tilde{A}^t$. Al cas multiarticle, però, cal emmagatzemar i calcular només les submatrius $AS^{(i)}A^t$, $i = 1, \dots, k$. Com al cas general, s'usen rutines de factorització simbòlica per evitar degradar l'esparsitat de $AS^{(i)}A^t$ en fer la factorització de Cholesky, i de nou la rutina GENQMD (que implementa la reordenació del mínim grau) del paquet SPARSPAK ha estat usada. Aquesta, a més, només ha de ser usada un cop, donat que la topologia de les $AS^{(i)}A^t$ és la mateixa. D'igual forma, només caldrà fer la factorització simbòlica de $AS^{(i)}A^t$ una sola vegada, la qual és vàlida per als k articles. La factorització de Cholesky real, però, sí que caldrà tenir-la per a cada una de les k matrius $AS^{(i)}A^t$.
- 2.- *Grau del preconditionador.* El grau del preconditionador (anomenat ϕ a §8.2.3) usat a la implementació és $\phi = 1$. Això significa que en aquest cas, i segons (8.38), el nostre preconditionador consisteix només en la matriu diagonal $M^{-1} = P^{-1}$. Tal i com es va comentar a §8.2.3, d'aquesta forma calen més iteracions del gradient conjugat (el grau del preconditionador és baix), però cada una d'elles és molt més ràpida. Es podria considerar com una alternativa vàlida el fet d'incrementar el grau del preconditionador de forma dinàmica a mida que ens atansem al punt solució, contrarestant els efectes provocats per la progressiva pèrdua de la condició de ser un "bon preconditionador" (ja es va comentar a §8.2.3 que això es devia a que, per a punts propers al punt solució, la matriu S definida a (8.18) esdevenia cada cop "menys" definida positiva). Tanmateix, es va observar que, usant un valor de $\phi = 2$ per a punts propers al punt solució, tot i que el nombre d'iteracions disminuïa, el temps total d'execució augmentava de forma considerable. Per aquest motiu, el valor de $\phi = 1$ és mantingut al llarg de tota l'execució.
- 3.- *Criteri d'aturada del gradient conjugat preconditionat.* A l'algorisme A8.1 no es va detallar el criteri d'aturada del mètode del gradient conjugat preconditionat. A continuació farem un breu esment del criteri usat (seguint la idea descrita a [68]). El sistema a solucionar era $\hat{D}dy_2 = \hat{\beta}_2$. Si considerem que θ és l'angle format pels vectors $\hat{D}dy_2$ i $\hat{\beta}_2$, aleshores considerarem que l'iterat actual dy_{2_i} és satisfactori si $1 - \cos \theta < \epsilon_\theta$, on (aplicant que $u \cdot v = \|u\|_2 \|v\|_2 \cos \theta_{(u,v)}$) podem escriure:

$$\cos \theta = \frac{\hat{\beta}_2^t \hat{D}dy_{2_i}}{\|\hat{\beta}_2\|_2 \|\hat{D}dy_{2_i}\|_2}$$

Donat que el vector residu r_i calculat a A8.1 verifica que $\hat{D}dy_{2_i} + r_i = \hat{\beta}_2$, podem realitzar el càlcul anterior de forma més eficient com:

$$\cos \theta = \frac{\hat{\beta}_2^t (\hat{\beta}_2 - r_i)}{\|\hat{\beta}_2\|_2 \|(\hat{\beta}_2 - r_i)\|_2}$$

A la implementació realitzada, inicialment $\epsilon_\theta = 10^{-6}$, i a cada iteració de l'algorisme primal-dual és actualitzat de la següent manera:

$$\epsilon_\theta := \begin{cases} \max\{0.95 \epsilon_\theta, 10^{-8}\} & \text{si el } \textit{gap} \text{ dual} \leq 10^{-4} \\ 10^{-8} & \text{si el } \textit{gap} \text{ dual} > 10^{-4} \end{cases}$$

Capítol 9

Experiència computacional amb el mètode primal-dual multiarticle.

Al capítol anterior s'han establert les bases d'un algorisme primal-dual especialitzat per resoldre el problema (**FML**). En aquest capítol es presentaran els resultats computacionals obtinguts amb una implementació d'aquell algorisme. A l'igual que es va fer al capítol 6, en primer lloc es descriuran molt breument alguns aspectes del codi desenvolupat, per passar tot seguit a presentar els problemes test usats i els resultats computacionals obtinguts. Finalment, es farà un breu comentari tot comparant la implementació primal-dual per a problemes multiarticle, amb la implementació del particionament primal lineal presentada als capítols previs.

9.1 Aspectes del codi desenvolupat.

El codi desenvolupat, el qual serà referit com a IPM (*Interior Point Multicommodity*), es basa en el mètode primal-dual especialitzat per a la resolució del problema (**FML**), tal i com s'ha descrit al capítol 8. IPM ha estat codificat en ANSI-C, i consta d'un total de 2785 línies de codi. S'ha dissenyat en forma de llibreria per poder ser usat com a rutina dins d'una aplicació més gran. L'usuari només ha de subministrar la informació relativa a les dimensions del problema (nombre de nodes, arcs i articles), costos lineals per arc i article, capacitats dels arcs per article, injeccions als nodes per article, capacitat mútua dels arcs, i topologia de la xarxa. A més, no s'ha de proporcionar cap vector addicional per ser usat com espai de treball, donat que l'assignació de memòria és completament dinàmica.

IPM usa la llibreria SPARSPAK [29] per realitzar les operacions d'obtenció d'una reordenació simètrica d'una matriu per l'algorisme del mínim grau (rutina GENQMD), factorització simbòlica d'una matriu simètrica i definida positiva (rutina SMBFCT), factorització de Cholesky d'una matriu simètrica i definida positiva (s'usa una variant de GSFCT), i solució d'un sistema simètric i definit positiu (GSSLV). Donat que SPARSPAK està escrit en Fortran-77, la component primera d'un vector sempre ocupa la posició número 1 (numeració Fortran). Per la seva banda, IPM considera que la primera component ocupa la posició número 0 (numeració C), donat que està implementat en ANSI-C. Aquest fet s'ha hagut de tenir molt present a l'hora de desenvolupar el codi IPM. D'igual forma, a l'hora de subministrar la informació del problema a IPM, aquesta s'ha de passar considerant numeració C (això només afecta a la declaració de la topologia de la xarxa, on el primer node serà el node 0 —no el node 1—, i el darrer node serà el node $m - 1$ —en comptes del node m).

A diferència del paquet PPRN (introduït al capítol 6), que va ser presentat com un codi relativament robust i, en principi, definitiu, IPM encara està subjecte a modificacions, i s'espera, en un futur, poder-li afegir opcions algorísmiques addicionals (consideració d'arcs amb capacitat il·limitada, detecció de problemes infactibles i il·limitats, etc.).

9.2 Problemes test usats.

Els problemes test usats corresponen a un subconjunt del total de problemes lineals presentats a §6.2.1. Només s'han pogut executar aquells problemes que no tenen constriccions a banda (per tant, els problemes de coordinació hidrotèrmica a llarg i curt termini — $L_j^{(6)}$ $j=1, \dots, 5$ i $L_j^{(7)}$ $j=1, 2$ — no han estat executats). De la resta de problemes, tots ells sense constriccions a banda, tampoc s'han executat aquells generats amb el generador Mnetgen ($L_j^{(5)}$ $j=1, \dots, 6$). Això és degut a que en els problemes generats amb Mnetgen apareixen arcs sense capacitat (a banda dels arcs arrel), i aquests no són suportats encara per IPM (i, tal i com es va dir a §8.1.1, considerar un límit arbitràriament gran —per ex. 10^{20} — no és una bona solució, donat que es provoquen greus problemes de convergència).

Per tant, els problemes test usats amb IPM corresponen als obtinguts a partir dels quatre generadors DIMACS: Rmfgen, Grid-on-torus, Gridgraph, i Gridgen, (denotats per $L_j^{(1)}$, $L_j^{(2)}$, $L_j^{(3)}$, i $L_j^{(4)}$, $j=1, \dots, 8$, respectivament). Les característiques d'aquests problemes van ser mostrades a les taules 6.I–6.IV.

9.3 Resultats computacionals.

Les taules 9.I–9.IV presenten els resultats computacionals obtinguts amb IPM amb cada un dels problemes test esmentats a la secció anterior. Totes les execucions s'han realitzat sobre una SunSparc 10/41 sota UNIX, amb un únic processador RISC de 40Mhz i aproximadament 20 Mflops i 100 Mips, i amb 64 Mbytes de memòria (32 Mbytes reals i 32 Mbytes mapejats a disc). La informació presentada a les taules per a cada problema test és la següent:

- “ $f(x^*)_P$ ”: Valor de la funció objectiu del problema primal al punt solució.
- “ $f(x^*)_D$ ”: Valor de la funció objectiu del problema dual al punt solució.
- “Fact $_P$ ”: Valor de la condició d'aturada que controla la factibilitat primal, calculada segons (7.68).
- “Fact $_D$ ”: Valor de la condició d'aturada que controla la factibilitat dual, calculada segons (7.69).
- “Gap $_D$ ”: Valor de la condició d'aturada que controla el *gap* dual, calculat segons (7.70).
- “It.”: Nombre d'iteracions de l'algorisme primal-dual.
- “GCP It.”: Nombre d'iteracions total del mètode del gradient conjugat preconditionat (calculat acumulant el nombre d'iteracions del gradient conjugat a cada iteració de l'algorisme primal-dual).

Taula 9.I. Resultats obtinguts amb els problemes Rmfgn (usant IPM).

test	$f(x^*)_P$	$f(x^*)_D$	Fact $_P$	Fact $_D$	Gap $_D$	It.	GCP It.	CPU sec.
L ₁ ¹⁾	$3.7567566 \cdot 10^5$	$3.7567467 \cdot 10^5$	$4.97 \cdot 10^{-15}$	$1.46 \cdot 10^{-10}$	$2.64 \cdot 10^{-6}$	30	0	397.4
L ₂ ¹⁾	$2.0272907 \cdot 10^6$	$2.0272792 \cdot 10^6$	$2.29 \cdot 10^{-9}$	$7.30 \cdot 10^{-11}$	$5.68 \cdot 10^{-6}$	50	554	3001.5
L ₃ ¹⁾	$4.5062711 \cdot 10^6$	$4.5062531 \cdot 10^6$	$8.08 \cdot 10^{-10}$	$7.30 \cdot 10^{-11}$	$4.00 \cdot 10^{-6}$	55	898	7095.4
L ₄ ¹⁾	$9.8704456 \cdot 10^6$	$9.8704228 \cdot 10^6$	$4.16 \cdot 10^{-9}$	$8.53 \cdot 10^{-11}$	$2.31 \cdot 10^{-6}$	61	1187	16737.9
L ₅ ¹⁾	$1.1839356 \cdot 10^7$	$1.1839303 \cdot 10^7$	$1.26 \cdot 10^{-6}$	$2.31 \cdot 10^{-10}$	$4.46 \cdot 10^{-6}$	46	2253	178.6
L ₆ ¹⁾	$2.7150868 \cdot 10^7$	$2.7150762 \cdot 10^7$	$2.45 \cdot 10^{-6}$	$3.49 \cdot 10^{-10}$	$3.91 \cdot 10^{-6}$	59	7431	1036.0
L ₇ ¹⁾	$3.9835381 \cdot 10^7$	$3.9835268 \cdot 10^7$	$1.33 \cdot 10^{-6}$	$5.41 \cdot 10^{-10}$	$2.86 \cdot 10^{-6}$	66	8660	1839.9
L ₈ ¹⁾	$5.4343457 \cdot 10^7$	$5.4343768 \cdot 10^7$	$2.39 \cdot 10^{-6}$	$5.63 \cdot 10^{-10}$	$5.72 \cdot 10^{-6}$	67	5679	1710.0

Taula 9.II. Resultats obtinguts amb els problemes Grid-on-torus (usant IPM).

test	$f(x^*)_P$	$f(x^*)_D$	Fact $_P$	Fact $_D$	Gap $_D$	It.	GCP It.	CPU sec.
L ₁ ²⁾	$3.6896888 \cdot 10^4$	$3.6896784 \cdot 10^4$	$2.28 \cdot 10^{-14}$	$4.46 \cdot 10^{-11}$	$2.82 \cdot 10^{-6}$	31	0	313.1
L ₂ ²⁾	$1.8796212 \cdot 10^5$	$1.8796027 \cdot 10^5$	$3.24 \cdot 10^{-7}$	$2.62 \cdot 10^{-11}$	$9.80 \cdot 10^{-6}$	66	4111	4816.1
L ₃ ²⁾	$1.1970551 \cdot 10^6$	$1.1970437 \cdot 10^6$	$8.01 \cdot 10^{-7}$	$3.86 \cdot 10^{-11}$	$9.52 \cdot 10^{-6}$	70	6496	12296.3
L ₄ ²⁾	Memòria insuficient per executar aquest problema							
L ₅ ²⁾	$5.2076448 \cdot 10^6$	$5.2076110 \cdot 10^6$	$2.55 \cdot 10^{-6}$	$3.35 \cdot 10^{-10}$	$6.49 \cdot 10^{-6}$	76	3958	287.0
L ₅ ²⁾	$1.2922777 \cdot 10^7$	$1.2922692 \cdot 10^7$	$9.05 \cdot 10^{-6}$	$3.44 \cdot 10^{-10}$	$6.64 \cdot 10^{-6}$	102	9915	1323.7
L ₆ ²⁾	$2.2663309 \cdot 10^7$	$2.2663088 \cdot 10^7$	$6.47 \cdot 10^{-6}$	$4.19 \cdot 10^{-10}$	$9.73 \cdot 10^{-6}$	120	22416	4352.4
L ₈ ²⁾	$3.6829211 \cdot 10^7$	$3.6828959 \cdot 10^7$	$8.43 \cdot 10^{-5}$	$3.89 \cdot 10^{-10}$	$6.86 \cdot 10^{-6}$	139	47508	11974.4

- “CPU sec.”: Temps de CPU, en segons, requerit per a l’execució del problema.

Observant les taules 9.I–9.IV poden fer-se alguns comentaris. En primer lloc, cal observar que dos casos (L₄²⁾ i L₄⁴⁾) no van ser executats per problemes de memòria. De fet, aquests problemes requerien menys de 64 Mbytes (que correspon a la capacitat de l’estació de treball on van ser executats els problemes), però, donat que 32 Mbytes es trobaven mapejats a disc, la major part del temps el sistema es dedicava a paginar en comptes de fer avançar l’execució de IPM. Això, junt amb el fet de que ambdós models ja eren per ells mateixos prou costosos, provocava execucions excessivament lentes, les quals van ser abortades.

Pot també comprovar-se com, en totes les execucions, el *gap* dual va ser reduït per sota del valor desitjat (a la implementació realitzada $\epsilon_g = 10^{-5}$). Pel que fa a la factibilitat dual, pot observar-se com aquesta sempre és assolida amb un valor molt acurat, mentre que la factibilitat primal no s’aconsegueix amb tanta precisió (a la majoria de casos l’execució s’atura en un punt on el valor de factibilitat primal és major que la precisió requerida $\epsilon_p =$

Taula 9.III. Resultats obtinguts amb els problemes
Gridgraph (usant IPM).

test	$f(x^*)_P$	$f(x^*)_D$	Fact $_P$	Fact $_D$	Gap $_D$	It.	GCP It.	CPU sec.
L ₁ ³⁾	9.4212976·10 ⁷	9.4212402·10 ⁷	8.05·10 ⁻¹⁴	9.21·10 ⁻¹²	6.10·10 ⁻⁶	24	0	9.7
L ₂ ³⁾	3.5588489·10 ⁸	3.5588447·10 ⁸	1.03·10 ⁻⁵	1.84·10 ⁻¹¹	1.18·10 ⁻⁶	50	12594	1644.3
L ₃ ³⁾	1.2874324·10 ⁸	1.2874302·10 ⁸	6.93·10 ⁻⁶	3.70·10 ⁻¹¹	1.73·10 ⁻⁶	64	10771	2818.3
L ₄ ³⁾	2.5361669·10 ⁸	2.5361551·10 ⁸	5.23·10 ⁻⁶	4.63·10 ⁻¹¹	4.62·10 ⁻⁶	79	30082	16485.5
L ₅ ³⁾	2.7853260·10 ⁷	2.7853004·10 ⁷	3.18·10 ⁻⁵	1.14·10 ⁻¹⁰	9.16·10 ⁻⁶	45	2563	236.3
L ₆ ³⁾	6.5144571·10 ⁷	6.5144494·10 ⁷	1.18·10 ⁻⁵	5.26·10 ⁻¹⁰	1.18·10 ⁻⁶	62	10132	1722.2
L ₇ ³⁾	2.7066800·10 ⁷	2.7066655·10 ⁷	9.66·10 ⁻⁶	8.74·10 ⁻¹⁰	5.37·10 ⁻⁶	61	3075	962.9
L ₈ ³⁾	3.7965012·10 ⁷	3.7964873·10 ⁷	9.85·10 ⁻⁶	1.20·10 ⁻⁹	3.67·10 ⁻⁶	67	5533	2083.1

Taula 9.IV. Resultats obtinguts amb els problemes
Gridgen (usant IPM).

test	$f(x^*)_P$	$f(x^*)_D$	Fact $_P$	Fact $_D$	Gap $_D$	It.	GCP It.	CPU sec.
L ₁ ⁴⁾	5.5419879·10 ⁶	5.5419722·10 ⁶	4.20·10 ⁻¹³	1.75·10 ⁻¹¹	2.84·10 ⁻⁶	24	0	251.1
L ₂ ⁴⁾	2.3223542·10 ⁷	2.3223404·10 ⁷	2.12·10 ⁻⁶	1.74·10 ⁻¹¹	5.93·10 ⁻⁶	39	2642	2853.6
L ₃ ⁴⁾	6.1791906·10 ⁷	6.1791943·10 ⁷	3.42·10 ⁻⁶	2.54·10 ⁻¹¹	6.00·10 ⁻⁷	44	9692	12216.2
L ₄ ⁴⁾	Memòria insuficient per executar aquest problema							
L ₅ ⁴⁾	1.4094746·10 ⁶	1.4094654·10 ⁶	2.96·10 ⁻⁷	2.00·10 ⁻¹⁰	6.55·10 ⁻⁶	51	463	114.1
L ₆ ⁴⁾	2.9402226·10 ⁶	2.9402110·10 ⁶	9.99·10 ⁻⁷	2.30·10 ⁻¹⁰	3.94·10 ⁻⁶	59	697	315.2
L ₇ ⁴⁾	4.6149909·10 ⁶	4.6149597·10 ⁶	2.04·10 ⁻⁶	2.29·10 ⁻¹⁰	6.75·10 ⁻⁶	68	1004	584.8
L ₈ ⁴⁾	6.4403968·10 ⁶	6.4403712·10 ⁶	3.24·10 ⁻⁶	2.47·10 ⁻¹⁰	3.97·10 ⁻⁶	70	1339	915.2

10⁻⁶). Aquests resultats es troben en consonància amb tot el dit a §8.3, on es van tractar les condicions d'aturada de l'algorisme.

També pot ser observat que, en alguns casos on la factibilitat primal i dual és assolida (per ex., L₁²⁾), la funció objectiu primal sempre és major que la funció objectiu dual (tal i com ha de ser). Tanmateix, en els casos on la factibilitat primal no s'aconsegueix de forma acurada, pot ocórrer que el valor de la funció objectiu dual superi al de la primal (per ex., L₁⁸⁾).

Els problemes L₁ⁱ⁾ i=1,...,4 són simples problemes de fluxos en xarxes amb un únic article. Per aquest tipus de problemes l'algorisme primal-dual descrit al capítol anterior també pot ser aplicat. En aquest cas, però, el càlcul de la direcció dy es redueix només a aplicar l'equació (8.28) (no cal solucionar (8.29)), i, per tant, no cal fer ús del mètode del gradient conjugat preconditionat. Això pot comprovar-se a les taules observant la columna "GCP it.", que en aquests exemples val 0. Així mateix, en no usar el mètode del gradient conjugat, el càlcul de dy es fa de forma exacta, eliminant errors de precisió. De nou això queda clar observant que en aquests problemes la factibilitat primal s'assoleix de forma molt acurada. Val a dir que, per aquest tipus de problemes de xarxes amb un únic

article, en comptes de factoritzar la matriu $AS^{(1)}A^t$ per solucionar (8.28), hi ha d'altres alternatives més eficients, com ara emprar un gradient conjugat amb preconditionament, usant un arbre d'expansió màxima com a preconditionador, tal i com es descriu a [65,68].

Un altre dels fets que cal remarcar és que, per a les quatre taules de resultats, en general els quatre primers casos L_j^i , $j=1, \dots, 4$, corresponents a xarxes grans i pocs articles, són força més costosos que els quatre darrers, associats a xarxes petites i molts articles. Això es deu precisament a la dimensió de la xarxa de cada tipus de problema, donat que les k matrius que cal factoritzar a cada iteració són del tipus $AS^{(i)}A^t$, i la seva dimensió correspon al nombre de nodes de la xarxa. Aleshores, com més gran sigui la xarxa, més costoses seran les factoritzacions de Cholesky que s'han de realitzar, i més costosa la solució de sistemes amb aquestes factoritzacions (sistemes que cal solucionar a cada iteració del mètode del gradient conjugat).

A §8.2.3 es va comentar que, a mida que ens apropàvem al punt solució del nostre problema, el preconditionador escollit deixava de ser una bona aproximació de la inversa de la matriu del sistema a resoldre. Això s'ha pogut comprovar experimentalment a les execucions realitzades, tal i com mostren les figures Fig. 9.1 i Fig. 9.2, corresponents a dos problemes concrets, un amb una xarxa gran i pocs articles i l'altre amb una xarxa petita i molts articles. La Fig. 9.1 mostra el nombre d'iteracions del gradient conjugat preconditionat (eix y) a cada una de les iteracions del l'algorisme primal-dual (eix x) pel

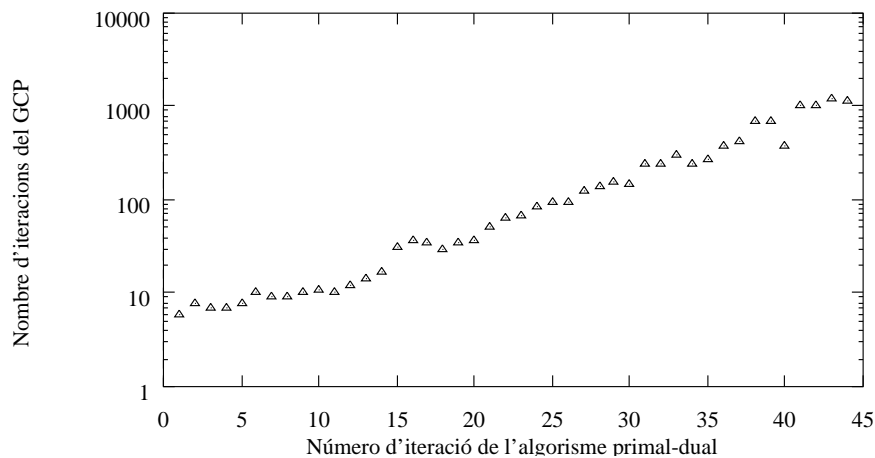


Figura 9.1. Nombre d'iteracions del GCP per iteració primal-dual del problema L_3^4 .

problema L_3^4 . La Fig. 9.2 mostra la mateixa informació, però pel problema L_8^1 . L'eix y ha estat escalat logarímicament en ambdós casos. S'observa clarament la tendència que hi ha a incrementar el nombre d'iteracions del gradient conjugat preconditionat a mida que ens apropem al punt solució. També pot comprovar-se com el nombre d'iteracions de gradient conjugat requerit és major pel problema L_3^4 que pel problema L_8^1 . Això es deu a que el sistema a solucionar amb el gradient conjugat és de dimensió n , on n és el nombre

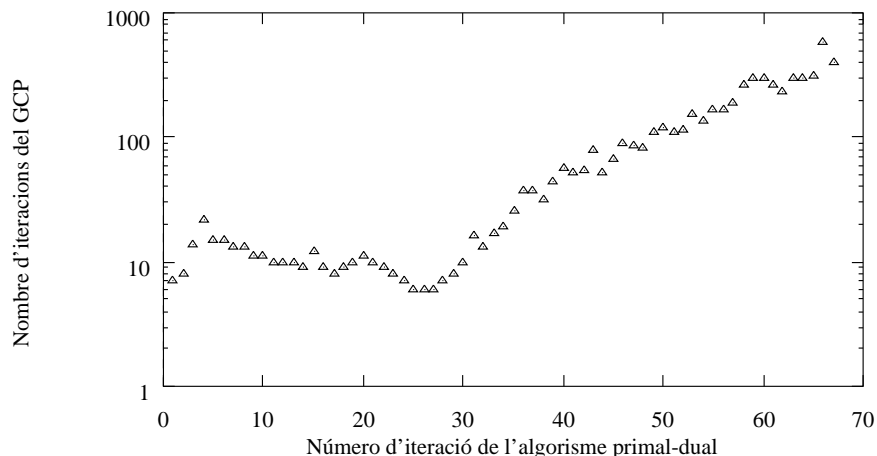


Figura 9.2. Nombre d'iteracions del GCP per iteració primal-dual del problema L_8^1 .

d'arcs de la xarxa, i en aquest cas el problema L_3^4 té una xarxa amb 7808 arcs, mentre que la del problema L_8^1 és de 496.

9.4 Comparació de IPM amb codis alternatius.

A la secció anterior s'han presentat els resultats obtinguts amb IPM usant un subconjunt dels problemes lineals presentats a §6.2.1. Donat que aquests problemes van ser executats amb diferents codis al capítol 6, és interessant poder establir comparacions entre l'actuació d'aquells i IPM. Concretament, hi ha dos casos d'especial interès. El primer consisteix en comparar IPM amb un codi de punt interior general (no especialitzat per a fluxos multiarticle). El segon serà veure com es comporta IPM respecte un codi basat en el símplex especialitzat per a problemes multiarticle. En el primer cas la comparació que es farà serà entre IPM i LoQo [77], mentre que en el segon s'analitzaran els resultats obtinguts amb IPM i PPRN.

A §6.3.1 es van presentar els resultats obtinguts amb els problemes test lineals amb el paquet LoQo. En aquell cas es va observar que LoQo requeria una gran quantitat de memòria per poder solucionar els problemes multiarticle (això pot ser explicat, en part, per algun resultat equivalent al de la proposició P8.1, donat que LoQo no explota l'estructura del problema i ha de fer una factorització de la matriu obtinguda amb el mètode general). La taula 9.v mostra els resultats obtinguts amb LoQo i IPM, només per als casos on LoQo va poder finalitzar la seva execució. La informació presentada és la següent:

- columnes “It.”: nombre d'iteracions, tant per a IPM com per LoQo.
- columnes “CPU sec.”: temps de CPU (en segons) requerit per realitzar l'execució del problema, per a IPM i LoQo.
- “It. ratio”: quocient del nombre d'iteracions de LoQo i IPM $\left(\frac{\text{It. (LoQo)}}{\text{It. (IPM)}}\right)$.

Taula 9.v. Comparació entre IPM i LoQo.

test	IPM		LoQo		ratios comparatius	
	It.	CPU sec.	It.	CPU sec.	It. ratio	CPU ratio
L ₁ ¹⁾	30	397.4	20	499.6	0.67	1.26
L ₅ ¹⁾	46	178.6	36	3402.8	0.78	19.05
L ₁ ²⁾	31	313.1	25	410.4	0.81	1.31
L ₅ ²⁾	76	287.0	49	5211.1	0.64	18.16
L ₁ ³⁾	24	9.7	32	29.8	1.33	3.07
L ₂ ³⁾	50	1644.3	37	874.1	0.74	0.53
L ₅ ³⁾	45	236.3	39	3180.4	0.87	13.46
L ₁ ⁴⁾	24	251.1	22	362.1	0.92	1.44

- “CPU ratio”: quocient del temps de CPU de LoQo i IPM $\left(\frac{\text{CPU. sec (LoQo)}}{\text{CPU. sec (IPM)}}\right)$.

Pot comprovar-se com LoQo realitza menys iteracions que IPM, però, en general, també requereix molt més temps de càlcul. Aquest darrer fet és sobre tot notable en els exemples amb molts articles. Cal tenir present, però, que les condicions d’aturada de IPM són menys restrictives que les de LoQo, i, en aquest sentit, el punt assolit solució assolit per LoQo és més acurat. Tanmateix, i a banda del comentari anterior, es pot deduir dels resultats de la taula 9.v que, a l’igual que succeeix amb el símplex, l’especialització per a problemes multiarticle en mètodes de punt interior permet obtenir un guany considerable en temps de càlcul (i en molts casos ens permet solucionar problemes que aplicant un mètode de punt interior general no podrien ser executats per requerir una gran quantitat de recursos —especialment memòria). El fet de que LoQo requereixi menys iteracions que IPM pot ser degut a que aquell usa un mètode predictor-corrector per solucionar el sistema d’equacions que ens proporciona dy .

La segona tasca a fer consistia en comparar el comportament de IPM amb PPRN, una implementació especialitzada per a problemes multiarticle basada en el símplex. S’ha escollit PPRN en comptes de MCNF85 donat que, en general, va tenir un millor rendiment en la majoria de resultats presentats a §6.3.1, i, a més, va poder solucionar tots els problemes (MCNF85 no va tenir un comportament tan robust). La taula 9.vi presenta la comparació del rendiment de IPM i PPRN. Per a cada problema test es dona (a la columna “CPU ratio”) el quocient del temps de CPU entre PPRN i IPM $\left(\frac{\text{CPU. sec (PPRN)}}{\text{CPU. sec (IPM)}}\right)$. La taula s’ha particionat en quatre subtaules verticals, cada una d’elles associada a un dels quatre tipus de problemes DIMACS considerats. Horitzontalment, es marca clarament la separació entre els problemes que tenen una xarxa gran i pocs articles (L_j^i , $j=1, \dots, 4$), i els que tenen una xarxa petita i molts articles (L_j^i , $j=5, \dots, 8$). Per a dos problemes (L_4^2 i L_4^4) no es mostra cap valor donat que l’execució amb IPM no va ser realitzada.

Observant 9.vi poden ser extretes unes conclusions força interessants. A la secció anterior es va comentar que IPM tenia un pitjor comportament per als casos on la mida de la xarxa era gran, donat que calia factoritzar les k matrius $AS^{(i)}A^t$, les quals en aquest cas

Taula 9.vi. Comparació entre IPM i PPRN.

test	CPU ratio	test	CPU ratio	test	CPU ratio	test	CPU ratio
L ₁ ¹⁾	0.02	L ₁ ²⁾	0.02	L ₁ ³⁾	0.45	L ₁ ⁴⁾	0.02
L ₂ ¹⁾	0.03	L ₂ ²⁾	0.11	L ₂ ³⁾	0.49	L ₂ ⁴⁾	0.12
L ₃ ¹⁾	0.10	L ₃ ²⁾	0.40	L ₃ ³⁾	0.50	L ₃ ⁴⁾	0.28
L ₄ ¹⁾	0.41	L ₄ ²⁾	—	L ₄ ³⁾	0.86	L ₄ ⁴⁾	—
L ₅ ¹⁾	1.54	L ₅ ²⁾	0.59	L ₅ ³⁾	1.54	L ₅ ⁴⁾	0.35
L ₆ ¹⁾	2.65	L ₆ ²⁾	1.23	L ₆ ³⁾	2.48	L ₆ ⁴⁾	0.61
L ₇ ¹⁾	4.39	L ₇ ²⁾	1.75	L ₇ ³⁾	4.65	L ₇ ⁴⁾	0.71
L ₈ ¹⁾	9.01	L ₈ ²⁾	1.86	L ₈ ³⁾	5.63	L ₈ ⁴⁾	1.39

tenien una dimensió elevada. Aquest fet queda també ratificat de forma clara observant la taula 9.vi. Pot comprovar-se com, per les execucions $L_j^{i)}$ $j=1, \dots, 4$ (xarxa gran i pocs articles), IPM té un rendiment molt inferior a PPRN. Per la seva banda, a la majoria de problemes amb molts articles i xarxa petita ($L_j^{i)}$ $j=5, \dots, 8$) IPM es comporta de forma més eficient que PPRN (en alguns casos el guany de temps és considerable, com a $L_8^{1)}$). Caldria matisar de nou, però, que IPM no troba l'òptim de forma exacta (per problemes de precisió lligades a l'ús del gradient conjugat amb preconditionament), sinó que obté una solució propera a aquest.

Un altre dels fets clau, i de molta rellevància, fa referència al comportament entre ambdós paquets a mida que creix la grandària del problema. Pot observar-se com, per a cada un dels quatre tipus de problemes DIMACS, i dins d'aquests, per a cada un dels dos tipus de casos (xarxa gran-pocs articles i xarxa petita-pocs articles), la seqüència de quocients presentada és creixent en tots els casos (per ex., per als problemes $L_j^{1)}$, $j=1, \dots, 4$, aquesta seqüència és: 0.02, 0.03, 0.10 i 0.41). Això implica que IPM, a mida que creix la dimensió del problema, té un millor comportament que PPRN. Aleshores, per a problemes amb un gran nombre de variables, sembla que IPM seria una millor alternativa que PPRN. Aquest resultat ve a ratificar una de les propietats bàsiques dels mètodes de punt interior i el mètode del símplex: els primers són algorismes polinòmics, mentre que el símplex té un cost exponencial.

Capítol 10

Conclusions i àrees de futura recerca.

Després d'introduir al llarg dels capítols precedents dues tècniques per solucionar problemes de fluxos multiarticle (particionament primal i mètode primal-dual de punt interior), en aquest capítol es presentaran, a mode de resum, les principals conclusions que poden ser extretes del treball desenvolupat, així com les possibles ampliacions i les línies de futura recerca. Es finalitzarà fent una exposició de les publicacions conseqüència del treball de tesi desenvolupat.

10.1 Conclusions.

Les principals conclusions que poden ser extretes del treball desenvolupat poden ser desglossades en els següents punts:

- El mètode del particionament primal s'ha mostrat com un mètode eficient per a la resolució de problemes multiarticle considerant constriccions a banda addicionals en la formulació del problema. De fet aquest mètode ha pogut ser estès de forma natural per tal de tractar constriccions a banda addicionals.
- D'igual forma, el mètode del particionament primal s'ha mostrat també com un mètode vàlid per solucionar eficientment problemes de fluxos multiarticle (amb o sense constriccions a banda) considerant funcions objectiu no lineals. En aquest cas, l'algorisme del conjunt de constriccions actives per a programació no lineal general ha pogut ser adaptat per realitzar certes manipulacions algebraïques a través del particionament primal.
- La implementació realitzada del particionament primal (PPRN), per a funcions lineals ha tingut un millor rendiment que paquets generals d'optimització (MINOS [59], LoQo [77]), i, en la majoria de casos, també s'ha mostrat més eficient que un altre paquet especialitzat per problemes multiarticle (MCNF85 [46]), sobre tot per a problemes on el nombre d'articles és molt elevat. Així mateix, PPRN s'ha mostrat com un codi més robust que MCNF85 en la resolució de diferents problemes. Cal observar que aquesta comparació no s'ha pogut realitzar amb un altre codi especialitzat multiarticle amb constriccions a banda addicionals, ja que es desconeix l'existència d'un paquet alternatiu.
- Pel que fa als problemes no lineals, PPRN només s'ha pogut comparar amb un paquet d'abast general, ja que no es coneix cap altra implementació d'un codi de fluxos no lineals multiarticle. En aquest cas, PPRN s'ha comparat amb MINOS, i en tot moment va tenir un millor comportament que aquest. Tanmateix, per al cas no lineal el guany

de temps no és tan considerable com al cas lineal, degut a que els càlculs on pot ser explotada l'estructura del problema no suposen un tant per cent tan elevat respecte del procés total, com ocorria al cas lineal (per ex., cal fer avaluacions addicionals de la funció objectiu no lineal).

- El fet de considerar tres fases en l'algorisme desenvolupat s'ha mostrat com una alternativa molt eficient, sobre tot pel que fa al nombre d'iteracions requerit per obtenir un punt inicial factible. Això s'ha pogut comprovar en les diferents execucions realitzades, on, en general, PPRN requeria moltes menys iteracions que la resta de codis per obtenir punts inicials factibles.
- El càlcul i actualització eficient de la matriu de treball desenvolupat difereix lleugerament del proposat al mètode original del particionament primal, i, a més, tracta el cas de considerar constriccions a banda addicionals. Part de l'eficiència relativa de PPRN i MCNF85 pot ser explicada gràcies a aquesta diferent actualització de la matriu de treball.
- L'algorisme primal-dual de punt interior desenvolupat per poder tractar problemes de caràcter general amb totes les variables afitades inferiorment i només un subconjunt d'elles afitades superiorment, s'ha mostrat com una alternativa vàlida respecte altres codis capdavanters d'optimització lineal (ha tingut un rendiment millor que MINOS i pitjor que LoQo —el millor rendiment de LoQo es deu a l'ús d'un algorisme predictor-corrector [54] per obtenir la direcció dy de canvi de les variables). Això ha fet que pogués servir de base per desenvolupar un algorisme de punt interior especialitzat per a problemes de fluxos multiarticle (lineals).
- El mètode primal-dual desenvolupat per solucionar problemes de fluxos multiarticle s'ha mostrat com una alternativa eficient als mètodes clàssics per solucionar aquest tipus de problemes. La seva eficiència rau especialment en l'ús d'un gradient conjugat amb preconditionament, usant un preconditionador específic per al problema multiarticle. Aquest preconditionador ha tingut un molt bon comportament a les primeres iteracions de l'algorisme, i, a mida que ens apropem al punt solució esdevé menys eficient. Globalment, però, els resultats obtinguts podrien ser qualificats de bons.
- L'especialització del mètode primal-dual per a la resolució de problemes de fluxos multiarticle s'ha mostrat més eficient que un codi de punt interior d'abast general (LoQo). Per tant, explotar l'estructura del problema usant mètodes de punt interior permet accelerar l'execució del problema, tal i com passava amb els mètodes clàssics basats en el símplex. A més, l'especialització usant tècniques de punt interior permet reduir considerablement la quantitat de recursos (entengui's memòria) requerits per l'algorisme, permetent de solucionar problemes amb major dimensió.
- Els tests realitzats amb la implementació primal-dual per a problemes multiarticle han demostrat que aquesta tècnica és més eficient que les tècniques clàssiques per a problemes amb un gran nombre d'articles i amb una dimensió de la xarxa no gaire

elevada. Per a problemes amb pocs articles i xarxa gran la implementació del particionament primal ha tingut un millor rendiment. Tanmateix, a mida que augmentava la mida del problema, aquest rendiment es feia menor. Sembla que, per a problemes molt grans, la implementació primal-dual pot millorar les execucions que puguin ser obtingudes amb tècniques basades en el símplex.

10.2 Ampliacions i futura recerca.

La tasca desenvolupada no està, òbviament, tancada, i pot ser ampliada afegint nous aspectes que millorin, tant l'eficiència, com la robustesa dels mètodes presentats. En aquest sentit, els següents punts detallen les possibles ampliacions i actuacions futures (algunes d'elles únicament viables a llarg termini):

- Pel que fa al mètode del particionament primal, la tasca desenvolupada permet, en principi, ampliacions amb un nombre de graus de llibertat no gaire gran. Una de les tasques que podrien ser realitzades consistiria en una comparació exhaustiva entre l'actualització de la matriu de treball presentada, la implementada per MCNF85, i la usada i proposada a [49] per a la resolució de problemes uniaarticle amb constriccions a banda. Una anàlisi exhaustiva d'aquestes tres metodologies podria permetre l'obtenció d'una nova actualització que pogués explotar més l'estructura del problema, autoadaptant-se l'actualització en funció del nombre d'articles, la dimensió de la xarxa, el nombre de constriccions a banda, i el percentatge d'aquestes que són d'igualtat o de desigualtat.
- Un altre aspecte que podria ser introduït consistiria en permetre el fet de considerar constriccions a banda no lineals en la formulació del problema solucionat. Caldria ampliar, aleshores, el codi PPRN per tal de tractar aquest tipus de constriccions (usant alguna de les possibles tècniques per tractar aquest tipus de constriccions: lagrangians augmentats, lagrangians projectats, etc.).
- Pel que fa al segon mètode presentat (de punt interior) per solucionar els problemes multiarticle lineals, aquí sí que s'obre un ventall més ampli de possibilitats. Un dels primers aspectes que caldria considerar seria l'obtenció d'un mètode per identificar la faceta òptima d'un problema de fluxos multiarticle un cop ens trobem a prop del punt solució. D'aquesta forma, un cop estiguéssim a prop del punt òptim, podríem evitar continuar iterant (amb el consegüent estalvi de temps, donat que a prop de l'òptim el preconditionador presentat no té un bon comportament) tot identificant la solució òptima (aquesta idea és presentada a [67] per al problema de fluxos amb un únic article).
- Un altre del temes a tractar seria l'addició de constriccions a banda en la implementació realitzada. Això implicaria disposar d'un preconditionador que es comportés més o menys bé en considerar una estructura arbitrària de constriccions a banda.

Aquesta és una de les vies d'estudi que podrien ser considerades com més ambiciosos (i, per tant, pot ser més a llarg termini).

- Un nou punt d'interès consistiria en intentar millorar el comportament de l'algorisme evitant de solucionar els sistemes amb les matrius $AS^{(i)}A^t$ mitjançant la seva factorització de Cholesky. En comptes d'això, es podria provar de solucionar-los a través d'un gradient conjugat preconditionat (usant com a preconditionador un arbre d'expansió màxima, tal i com es descriu a [68]).
- El mètode presentat per calcular la direcció dy del mètode primal-dual (basat en la descomposició de dominis) permet que la resolució dels k sistemes amb les matrius $AS^{(i)}A^t$ pugui ser feta en paral·lel. En aquest sentit, una altra via de futura recerca consistiria en adaptar el codi desenvolupat per poder ser executat en ordinadors amb processament paral·lel, el qual reduiria notablement el procés de càlcul.
- També seria d'interès estendre les comparacions entre algorismes alternatius de fluxos multiarticle amb problemes de més alta dimensió utilitzant ordinadors amb més capacitat de memòria que l'utilitzat fins ara. Això permetria confirmar tendències observades en el canvi d'eficiència dels diferents mètodes segons la mida del problema.
- Anant més enllà, l'algorisme desenvolupat podria ser ampliat pel cas de tractar problemes quadràtics, tot explotant l'estructura de xarxa inherent. I, un cop aconseguit aquest primer objectiu, es podria continuar l'extensió pel cas de problemes no lineals. Naturalment, aquestes dues tasques impliquen un canvi qualitatiu en els algorismes descrits. Tanmateix, actualment aquest camp està essent objecte d'una intensa recerca, i no pot descartar-se l'obtenció de nous i prometedors resultats en un futur proper.

10.3 Publicacions conseqüència del treball de tesi.

Durant el temps dedicat al desenvolupament d'aquest treball diverses publicacions han estat elaborades, les quals han estat la base de la memòria aquí presentada. Tot i que aquestes ja han estat esmentades al llarg dels diferents capítols de la memòria, a continuació són recollides i llistades per ordre cronològic (a diferència de l'apartat posterior de Bibliografia, on apareixen per ordre alfabètic).

- Castro, J., "Càlcul i actualització eficient de la matriu de treball del problema de fluxos multiarticle amb constriccions a banda a través del particionament primal", DR 93/03 Dept. d'Estadística i Investigació operativa, Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1993.
- Castro, J., and Nabona, N., "Computational tests of a linear multicommodity network flow code with linear side constraints through primal partitioning", DR 94/02 Statistics and Operations Research Dept., Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1994.

- Castro, J., and Nabona, N., “Nonlinear multicommodity network flows through primal partitioning and comparison with alternative methods”, in: J. Henry and J.-P. Yvon (eds.), *System Modelling and Optimization. Proceedings of the 16th IFIP-TC7 Conference*, Springer-Verlag, 1994, 875–884.
- Castro, J., and Nabona, N., “Computational tests of a nonlinear multicommodity network flow code with linear side constraints through primal partitioning”, DR 94/05 Statistics and Operations Research Dept., Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1994.
- Castro, J., “PPRN 1.0, User’s Guide”, DR 94/06 Statistics and Operations Research Dept., Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1994.
- Castro, J., “Implementació d’un algorisme primal-dual de punt interior amb fites superiors a les variables”, *Qüestió* 19 (1995).
- Castro, J., and Nabona, N., “An implementation of linear and nonlinear multicommodity network flows”, accepted for publication in the *European Journal of Operational Research*, 1995.
- Castro, J., and Nabona, N., “Primal-dual interior point method for multicommodity network flows with side constraints and comparison with alternative methods”, accepted for presentation at 17th IFIP TC7 Conference, Prague, Czech Republic, July 1995.

Bibliografia.

- [1] Adler, I., Resende, M.G.C., and Veiga, G., “An implementation of Karmarkar’s algorithm for linear programming”, *Mathematical Programming* 44 (1989) 297–335.
- [2] Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., *Network Flows*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [3] Ali, A., Helgason, R.V., Kennington, J.L., and Lall, H., “Computational comparison among three multicommodity network flow algorithms”, *Operations Research* 28 (1980) 995–1000.
- [4] Ali, A., and Kennington, J.L., “MNETGEN program documentation”, Technical Report 77003, Department of Industrial Engineering and Operations Research, Southern Methodist University, Dallas, TX 75275, USA, 1977.
- [5] Arbel, A., *Exploring Interior-Point Linear Programming. Algorithms and Software*. The MIT Press, Cambridge, Massachusetts 02142, 1993.
- [6] Barnes, E.R., “A variation on Karmarkar’s algorithm for solving linear programming problems”, *Mathematical Programming* 36 (1986) 174–182.
- [7] Beckmann, M., McGuire, C.B., and Winsten, C.B., *Studies in the Economics of Transportation*, Yale University Press, New Haven CT, 1956.
- [8] Bellmore, M., Bennington, G., and Lubore, S., “A multivehicle tanker scheduling problem”, *Transportation Science*, 5 (1971) 36–47.
- [9] Bradley, G.H., Brown, G.G., and Graves, G.W., “Design and implementation of large scale primal transshipment algorithms”, *Management Science* 24(1) (1977) 1–34.
- [10] Castro, J., “Càlcul i actualització eficient de la matriu de treball del problema de fluxos multiarticle amb constriccions a banda a través del particionament primal”, DR 93/03 Dept. d’Estadística i Investigació operativa, Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1993.
- [11] Castro, J., “PPRN 1.0, User’s Guide”, DR 94/06 Statistics and Operations Research Dept., Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1994.
- [12] Castro, J., “Implementació d’un algorisme primal-dual de punt interior amb fites superiors a les variables”, *Qüestió* 19 (1995).
- [13] Castro, J., and Nabona, N., “Nonlinear multicommodity network flows through primal partitioning and comparison with alternative methods”, in: J. Henry and J.-P. Yvon (eds.), *System Modelling and Optimization. Proceedings of the 16th IFIP-TC7 Conference*, Springer-Verlag, 1994, 875–884.
- [14] Castro, J., and Nabona, N., “Computational tests of a linear multicommodity network flow code with linear side constraints through primal partitioning”, DR 94/02 Statistics and Operations Research Dept., Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1994.

- [15] Castro, J., and Nabona, N., “Computational tests of a nonlinear multicommodity network flow code with linear side constraints through primal partitioning”, DR 94/05 Statistics and Operations Research Dept., Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1994.
- [16] Castro, J., and Nabona, N., “An implementation of linear and nonlinear multicommodity network flows”, accepted for publication in the *European Journal of Operational Research*, 1995.
- [17] Castro, J., and Nabona, N., “Primal-dual interior point method for multicommodity network flows with side constraints and comparison with alternative methods”, accepted for presentation at 17th IFIP TC7 Conference, Prague, Czech Republic, July 1995.
- [18] Choi, I.C., and Goldfarb, D., “Solving multicommodity network flow problems by an interior point method”, *SIAM proceedings on Applied Mathematics* 46 (1990) 58–69, Philadelphia, PA, USA.
- [19] Choi, I.C., and Goldfarb, D., “Exploiting special structure in a primal-dual path-following algorithm”, *Mathematical Programming* 58 (1993) 33–52.
- [20] Crainic, T., Ferland, J.A., and Rousseau, J.M., “A tactical planning model for rail freight transportation”, *Transportation Science* 18 (1984) 165–184.
- [21] Dantzig, G.B., and Wolfe, P., “Decomposition principle for linear programming”, *Operations Research* 8 (1960) 101–111.
- [22] Dembo, R.S., Mulvey, J.M., and Zenios, S.A., “Large Scale Nonlinear Network Models and their Application”, Report EES-86-18 Dept. of Civil Engineering and Operations Research, Princeton University, Princeton NJ 08544, USA, 1987.
- [23] Dembo, R.S., and Steihaug, T., “Truncated-Newton algorithms for large-scale unconstrained optimization”, *Mathematical Programming* 26 (1983) 190–212.
- [24] Dikin, I.I., “Iterative solution of problems of linear and quadratic programming”, *Soviet Mathematics Doklady* 8(3) (1967) 674–675.
- [25] DIMACS, “The first DIMACS international algorithm implementation challenge: The bench-mark experiments”, Technical report, DIMACS, New Brunswick, NJ, USA, 1991.
- [26] Duff, I.S., Erisman A.M., and Reid, J.K., *Direct Methods for Sparse Matrices* Oxford University Press, New York, 1986.
- [27] Ford, L.R., and Fulkerson, D.R., “A suggested computations for maximal multicommodity network flows”, *Management Science* 5 (1958) 97–101.
- [28] Gay, D.M., “Electronic mail distribution of linear programming test problems”, *Mathematical Programming Society COAL Newsletter* 13 (1985) 10–12.
- [29] George, J.A., and Liu, J.W.H., *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1981.
- [30] Gill, P.E., W. Murray, and Wright, M.H., *Practical Optimization*, Academic Press, London, UK, 1981.

-
- [31] Golden, B.L., “A minimum cost multicommodity network flow problem concerning imports and exports”, *Networks* 5 (1975) 331–356.
- [32] Goldfarb, D., and Grigoriadis, M.D., “A Computational Comparison of the Dinic and Network Simplex Methods for Maximum Flow”, *Annals of Operations Research* 13 (1988) 83–128.
- [33] Golub, G., and Ortega, J.M., *Scientific Computing. An introduction with parallel computing*, Academic Press, San Diego CA, 1993.
- [34] Grigoriadis, M.D., “An Efficient Implementation of the Network Simplex Method”, *Mathematical Programming Study* 26 (1986) 83–111.
- [35] Grigoriadis, M.D., and White, W.W., “A partitioning algorithm algorithm for the multicommodity network flow problem”, *Mathematical Programming* 3 (1972) 157–177.
- [36] Hellerman, E., and Rarick, D., “Reinversion with the preassigned pivot procedure”, *Mathematical Programming* 1 (1971) 195–216.
- [37] Heredia, F.J., i Nabona, N., “Programa FXCB de fluxos lineals amb xarxes amb constriccions a banda lineals”, RR 90/06. Facultat d’Informàtica de Barcelona, Universitat Politècnica de Catalunya, 08071 Barcelona, Spain, 1990.
- [38] Heredia, F.J., and Nabona, N., “Numerical implementation and computational results of nonlinear network optimization with linear side constraints”, in: P. Kall (ed.), *System Modelling and Optimization. Proceedings of the 15th IFIP Conference*, Springer-Verlag, 1992, 301–310.
- [39] Heredia, F.J., and Nabona, N., “Optimum Short-Term Hydro-thermal Scheduling with Spinning Reserve through Network Flows”, accepted by IEEE PES for publication in *IEEE Trans. on Power Systems*, 1995.
- [40] Held, M., Wolfe, P., and Crowder, H., “Validation of subgradient optimization”, *Mathematical Programming* 6 (1974) 62–88.
- [41] Jewell, W.S., ‘Warehousing and distribution of a seasonal product’, *Naval Research Logistic* 4 (1957), 29–34
- [42] Kamath, A.P, Karmarkar, N.K., and Ramakrishnan, K.G., “Computational and Complexity Results for an Interior Point Algorithm on Multicommodity Flow Problems”. Technical Report TR-21/93, Dipartimento di Informatica, Università di Pisa, Italy, 1993. Extended abstracts of Netflow’93.
- [43] Kapoor, S., and Vaidya, P.M., “Fast Algorithms for Convex Quadratic Programming and Multicommodity Flows”, in *Proceedings of 18th Annual ACM Symposium on the Theory of Computing*, 1986, 147–159.
- [44] Karmarkar, N.K., “A new polynomial time algorithm for linear programming”, *Combinatorica* 4 (1984) 373–395.
- [45] Karmarkar, N.K., and Ramakrishnan, K.G., “Computational results of an interior point algorithm for large scale linear programming”, *Mathematical Programming* 52 (1991) 555–586.

- [46] Kennington, J.L., “Solving multicommodity transportation problems using a primal partitioning simplex technique”, Technical Report CP 75013, Department of Computer Science and Operations Research, Southern Methodist University, Dallas, TX 75275, USA, 1975.
- [47] Kennington, J.L., “A Survey of Linear Cost Multicommodity Network Flows”, *Operations Research* 26(2) (1978) 209–236.
- [48] Kennington, J.L., “A primal partitioning code for solving multicommodity flow problems (version 1)”, Technical Report 79008, Department of Industrial Engineering and Operations Research, Southern Methodist University, Dallas, TX 75275, USA, 1979.
- [49] Kennington, J.L., and Helgason, R.V., *Algorithms for network programming*, John Wiley & Sons, New York, 1980.
- [50] Kennington, J.L., and Whisman, A., “NETSIDE User’s Guide”, TR86-OR-01 (revised April 1988), Southern Methodist University, Dallas, TX 75275, USA, 1988.
- [51] Khachiyan, G., “A polynomial algorithm in linear programming”, *Doklady Akademii Nauk SSSR* 244(S) (1979) 1093–1096, traduït en *Soviet Mathematics Doklady*, 20(1) (1979) 191–194.
- [52] Leighton, T., Makedon, F., Plotkin, S., Stein, C., Tardos, É, and Tragoudas, S., “Fast approximation algorithms for multicommodity flow problems”, in: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 1990, 310–321.
- [53] Luenberger, D.G., *Linear and Nonlinear Programming*, Addison-Wesley, Menlo Park, California, 1984.
- [54] Mehrotra, S., “On the implementation of a (primal-dual) interior point method”, Technical Report 90-03 Dept. of Industrial Engineering and Management Science, Northwestern University, Evanston, IL, 1990.
- [55] Monma, C.L., and Morton, A.J., “Computational experience with a dual affine variant of Karmarkar’s method for linear programming”, *Operations Research Letters* 6 (1987) 261–267.
- [56] Monteiro, R.D.C., and Adler, I., “Interior path following primal-dual algorithms. Part I: linear programming”, *Mathematical Programming* 44 (1989) 27–41.
- [57] Monteiro, R.D.C., and Adler, I., “Interior path following primal-dual algorithms. Part II: quadratic programming”, *Mathematical Programming* 44 (1989) 43–66.
- [58] Murtagh, B.A., and Saunders, M.A., “Large-scale linearly constrained optimization”, *Mathematical Programming* 14 (1978) 41–72.
- [59] Murtagh, B.A., and Saunders, M.A., “MINOS 5.0. User’s guide”, Dept. of Operations Research, Stanford University, CA 9430, USA, 1983.
- [60] Nabona, N., “Descripció del programa LEXA”, Document intern, Dept. d’Estadística i Investigació operativa, Facultat d’Informàtica de Barcelona, 1985.
- [61] Nabona, N., “Implementació i extensions dels fluxos multiarticle en xarxes per descomposició dictada per preus”, Report de Recerca RR88/26 Facultat d’Informàtica de Barcelona, Universitat Politècnica de Catalunya, Barcelona 08071, Spain, 1988.

-
- [62] Nabona, N., “Multicommodity network flow model for long-term hydrogeneration optimization”, *IEEE Trans. on Power Systems* 8(2) (1993) 395–404.
- [63] Nabona, N., and Verdejo, J.M., “Numerical implementation of nonlinear multicommodity network flows with linear side constraints through price-directive decomposition”, in: P. Kall (ed.), *System Modelling and Optimization. Proceedings of the 15th IFIP Conference*, Springer-Verlag, Zurich, 1992, 311–320.
- [64] Ortega, J.M., *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, 1988.
- [65] Portugal, L., Resende, M., Veiga, G., and Júdice, J., “A truncated primal-infeasible dual-feasible network interior point method”, Technical Report, AT&T Bell Laboratories, Murray Hill NJ, USA, 1994.
- [66] Resende, and Pardalos, P., “Interior point algorithms for network flow problems”, to appear in: J.E. Beasley (ed.), *Advances in linear and integer programming*, Oxford University Press, 1995.
- [67] Resende, M., Tsuchiya, T., and Veiga, G., “Identifying the optimal face of a network linear program with a globally convergent interior point method”, in: W. Hager, D. Hearn, and P. Pardalos (eds.), *Large scale optimization: State of the art*, Kluwer Academic Publishers, 1994, 362–387.
- [68] Resende, M., and Veiga, G., “An efficient implementation of a network interior point method”, in: D.S. Johnson and C.C. McGeoch (eds.), *Network Flows and Matching: First DIMACS implementation challenge*, v. 12 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1993, 299–348.
- [69] Resende, M., and Veiga, G., “An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite uncapacitated networks”, *SIAM Journal of Optimization* 3 (1993) 516–537.
- [70] Robacker, J.T., “Notes on linear programming: Part xxxvii Concerning multicommodity networks”, RM-1799, The Rand Corporation, Santa Monica, California, 1956.
- [71] Schrage, L., “A More Portable FORTRAN Random Number Generator”, *ACM Transactions on Mathematical Software* June 1979.
- [72] Sheffi, Y., *Urban Transportation Networks. Equilibrium Analysis with Mathematical Methods*, Prentice Hall, Englewood Cliffs, New Jersey, 1985.
- [73] Swoveland, C., “Decomposition algorithms for the multi-commodity distribution problem”, Working Paper No. 184, Western Management Science Institute, University of California, Los Angeles, 1971.
- [74] Toint, Ph.L., and Tuyttens, D., “On large scale nonlinear network optimization”, *Mathematical Programming Series B* 48(1) (1990) 125–159.
- [75] Tomlin, J.A., “Minimum-cost multicommodity network flows”, *Operations Research* 14 (1966) 45–51.
- [76] Vanderbei, R.J., “Affine-scaling for linear programs with free variables”, *Mathematical Programming* 43 (1989) 31–44.

- [77] Vanderbei, R.J., “LOQO User’s Manual”, Princeton University, Princeton NJ 08544, USA, 1992.
- [78] Vanderbei, R.J., “An interior point code for quadratic programming”, Princeton University, Princeton NJ 08544, USA, 1994.
- [79] Vanderbei, R.J., and Carpenter, T.J., “Symmetric Indefinite systems for interior point methods”, *Mathematical Programming* 58 (1993) 1–32.
- [80] Vanderbei, R.J., Duarte, A., and Yang, B., “An Algorithmic and Numerical Comparison of Several Interior-Point Methods”, Technical Report SOR-94-5, Dept. of Civil Engineering and Operations Research, Princeton University, Princeton NJ 08544, 1994.
- [81] Vanderbei, R.J., M.S. Meketon, and Freedman, B.A., “A modification of Karmarkar’s linear programming algorithm”, *Algorithmica* 1 (1986) 395–407.
- [82] White, W.W., “Mathematical Programming, Multicommodity Flows, and Communication Nets”, *Proceedings of the Symposium on Computer-Communications Networks and Teletraffic*, Polytechnic Institute of Brooklyn, 1972, 325–334.
- [83] Wright, M.H., “Interior Methods for constrained optimization”, *Acta Numerica*, (1991) 341–407.

Glossari de símbols.

S'indica entre parèntesis la pàgina on es defineix (o apareix per primer cop) cada símbol.

- $|\mathcal{C}|$: cardinalitat del conjunt qualsevol \mathcal{C} . (20)
- $\|\cdot\|_p$: norma p d'un vector. (66)
- $\underline{0}$: capacitats mínimes dels arcs. (8)
- $\mathbb{1}^{cb}$: matriu de constriccions associada a les variables f_{cb} durant la fase 1. (50)
- $\mathbb{1}^{cm}$: matriu de constriccions associada a les variables f_{cm} durant la fase 1. (50)
- $\mathbb{1}$: matriu identitat. (9)
- A : matriu amb les equacions de xarxa. (7)
- \mathcal{A} : conjunt de constriccions de capacitat mútua i a banda actives. (20)
- \tilde{A} : matriu de constriccions del problema de programació (lineal o no lineal) considerat. (9)
- \hat{A} : matriu de constriccions actives. (58)
- \mathcal{A}_{cb} : conjunt de constriccions a banda actives. (20)
- \mathcal{A}_{cm} : conjunt de constriccions de capacitat mútua actives. (20)
- A_l : submatriu de \tilde{A} associada a les variables no afitades superiorment. (110)
- $A_{\bar{a}}$: submatriu de \tilde{A} que no contempla l'arc arrel. (130)
- A_u : submatriu de \tilde{A} associada a les variables afitades superiorment. (110)
- B : matriu bàsica de \tilde{A} . (20)
- \underline{b}_{cb} : límits inferiors de les constriccions a banda. (9)
- \underline{b}_{cb_u} : límits superiors de les constriccions a banda afitades superior i inferiorment. (133)
- \bar{b}_{cb} : límits superiors de les constriccions a banda. (9)
- \bar{b}_{cb_e} : terme independent de les constriccions a banda d'igualtat. (133)
- \bar{b}_{cb_l} : límits superiors de les constriccions a banda afitades superiorment. (133)
- \bar{b}_{cb_u} : límits superiors de les constriccions a banda afitades superior i inferiorment. (133)
- b_{cm} : vector amb les capacitats mútues als arcs. (8)
- $B^{(i)}$: base de la matriu de xarxa A de l'article i . (20)
- $b^{(i)}$: vector amb les injeccions als nodes per a l'article i . (8)
- $c^{(i)}$: vector de costos lineals de l'article i . (7)
- c_l : components de c associades a les variables no afitades superiorment. (110)
- c_u : components de c associades a les variables afitades superiorment. (110)
- det : determinant d'una matriu. (24)
- dim : dimensió d'una matriu. (24)
- \mathcal{E} : conjunt d'arcs de la xarxa definida per \mathcal{G} . (7)
- f : folgues primals del mètode primal-dual. (110)
- F : matriu diagonal formada per les folgues $\bar{x}_{u_j} - x_{u_j}$ $j=1, \dots, n_u$. (112)

- $f(x)$: funció objectiu real que té per variables els fluxos dels articles. (8)
- (F0)** : problema considerat a la fase 0. (48)
- f_{cb} : variables artificials introduïdes a la fase 1 per les constriccions a banda. (49)
- f_{cm} : variables artificials introduïdes a la fase 1 per les constriccions de capacitat mútua. (49)
- (FML)** : problema de fluxos multiarticle lineals. (7)
- (FMLCB)** : problema de fluxos multiarticle lineals amb constriccions a banda. (9)
- (FMN)** : problema de fluxos multiarticle no lineals. (9)
- (FMNCB)** : problema de fluxos multiarticle no lineals amb constriccions a banda. (9)
- \mathcal{G} : graf dirigit que defineix la xarxa del problema. (7)
- $g(x)$: gradient de la funció objectiu $f(x)$ ($g(x) = \nabla f(x)$). (57)
- g_z : vector gradient reduït (o gradient projectat). (59)
- $H(x)$: hessiana de la funció objectiu $f(x)$ ($H(x) = \nabla^2 f(x)$). (57)
- H_z : matriu hessiana reduïda (o hessiana projectada). (59)
- k : nombre d'articles que circulen per la xarxa. (7)
- \mathcal{K} : conjunt d'articles de la xarxa definida per \mathcal{G} . (7)
- L_1 : submatriu de B associada a les constriccions de xarxa i arcs bàsics. (21)
- L_2 : submatriu de B associada a les constriccions de capacitat mútua i a banda actives, per als arcs bàsics. (22)
- L_3 : submatriu de B associada a les constriccions de capacitat mútua i a banda inactives, per als arcs bàsics. (22)
- m : nombre de nodes de la xarxa. (7)
- \bar{M} : matriu estesa de la matriu qualsevol M . (32)
- \tilde{m} : nombre total de constriccions del problema de programació (lineal o no lineal) considerat. (9)
- n : nombre d'arcs de la xarxa. (7)
- (NCL)** : problema d'optimització no lineal amb constriccions lineals. (57)
- N : matriu no bàsica de \tilde{A} . (20)
- \tilde{n} : nombre total de variables del problema de programació (lineal o no lineal) considerat. (9)
- p_B : direcció de descens per les variables bàsiques. (59)
- \mathcal{P}_j : conjunt d'arcs que formen el cicle de l'arc complementari j dins l'arbre bàsic associat. (25)
- p_N : direcció de descens per les variables no bàsiques. (59)
- p_S : direcció de descens per les variables superbàsiques. (59)
- (PZÀ)** : problema d'optimització al subespai de les variables superbàsiques. (59)
- Q : matriu de treball del mètode del particionament primal. (23)
- Q_{cb} : submatriu de Q associada a les constriccions a banda actives. (25)
- Q_{cm} : submatriu de Q associada a les constriccions de capacitat mútua actives. (25)
- R : matriu de la factorització de $Z^t H Z$ ($R^t R = Z^t H Z$). (67)
- r : vector auxiliar usat en la resolució del sistema lineal al mètode primal-dual. (115)

- R_1 : submatriu de B associada a les constriccions de xarxa i arcs complementaris. (21)
- $R_1^{(i)}$: submatriu de R_1 associada als arcs complementaris de l'article i . (43)
- $\mathcal{R}_1^{(i)}$: conjunt d'arcs complementaris de l'article i . (44)
- R_2 : submatriu de B associada a les constriccions de capacitat mútua i a banda actives, per als arcs complementaris. (22)
- R_3 : submatriu de B associada a les constriccions de capacitat mútua i a banda inactives, per als arcs complementaris. (22)
- $r^{(i)}$: recursos del mètode de la descomposició dictada per recursos. (15)
- r_l : components de r associades a les variables no afitades superiorment. (115)
- r_u : components de r associades a les variables afitades superiorment. (115)
- S : matriu diagonal auxiliar usada en la resolució del sistema lineal al mètode primal-dual. (115)
- S : matriu superbàsica de \tilde{A} (al mètode de Murtagh i Saunders). (58)
- s : nombre de variables superbàsiques. (57)
- s_{cb} : folgues de les constriccions a banda. (9)
- s_{cb_l} : folgues de les constriccions a banda afitades superiorment. (132)
- s_{cb_u} : folgues de les constriccions a banda afitades superior i inferiorment. (132)
- s_{cm} : folgues de les constriccions de capacitat mútua. (9)
- S_l : submatriu de S associada a les variables no afitades superiorment. (115)
- S_u : submatriu de S associada a les variables afitades superiorment. (115)
- t : nombre de constriccions a banda. (9)
- t_e : nombre de constriccions a banda d'igualtat. (132)
- $T^{(i)}$: matriu amb les constriccions a banda de l'article i . (9)
- $T_e^{(i)}$: matriu amb les constriccions a banda d'igualtat per a l'article i . (132)
- $T_l^{(i)}$: matriu amb les constriccions a banda afitades només superiorment per a l'article i . (132)
- $T_u^{(i)}$: matriu amb les constriccions a banda afitades superior i inferiorment per a l'article i . (132)
- t_l : nombre de constriccions a banda afitades superiorment. (132)
- t_u : nombre de constriccions a banda afitades superior i inferiorment. (132)
- \mathcal{V} : conjunt de vèrtex de la xarxa definida per \mathcal{G} . (7)
- w : folgues duals associades a les fites superiors de les variables del mètode primal-dual. (110)
- W : matriu diagonal formada per les components de w . (112)
- X : matriu diagonal formada per X_u i X_l . (112)
- x_B : variables bàsiques. (57)
- $x^{(i)}$: vector de fluxos de l'article i . (7)
- $x_{(j)}^{(i)}$: punts extrems de $\Lambda^{(i)}$. (14)
- $x_a^{(i)}$: arc arrel de l'article i . (130)
- $\bar{x}^{(i)}$: vector amb les capacitats de cada arc per a l'article i . (8)

- X_l : matriu diagonal formada per les components de x_l . (112)
- x_N : variables no bàsiques. (57)
- $x_a^{(i)}$: vector format per les components de $x^{(i)}$ excloent aquella associada a l'arc arrel. (130)
- x_S : variables superbàsiques. (57)
- x_l : components de x associades amb les variables no afitades superiorment. (110)
- x_u : components de x associades amb les variables afitades superiorment. (110)
- X_u : matriu diagonal formada per les components de x_u . (112)
- $\hat{x}^{(i)}$: punts pseudofactibles obtinguts en solucionar la fase 0. (49)
- y : variables duals del mètode primal-dual. (110)
- z : folgues duals associades a la matriu \tilde{A} del mètode primal-dual. (110)
- Z : matriu del subespai nul de la matriu de constriccions actives (al mètode del conjunt de constriccions actives). (59)
- Z : matriu diagonal formada per Z_u i Z_l (al mètode primal-dual). (112)
- Z_l : matriu diagonal formada per les components de z_l . (112)
- Z_u : matriu diagonal formada per les components de z_u . (112)
- α_d : pas realitzat per les variables duals al mètode primal-dual. (117)
- α_p : pas realitzat per les variables primals al mètode primal-dual. (117)
- ε_M : precisió de la màquina. (66)
- λ : multiplicadors de Lagrange de les fites de les variables no bàsiques del problema no lineal. (60)
- $\Lambda^{(i)}$: conjunts convexos del mètode de la descomposició dictada per preus. (14)
- $\lambda_{(j)}^{(i)}$: coeficients convexos dels punts extrems de $\Lambda^{(i)}$. (14)
- μ : coeficient de la barrera logarísmica del primal-dual. (111)
- π : multiplicadors de Lagrange de les constriccions del problema no lineal. (60)
- ρ : coeficient usat per allunyar variables de la seva fita al mètode primal-dual. (122)
- $\rho(H)$: radi espectral d'una matriu qualsevol H . (142)
- σ : factor de reducció de μ al mètode primal-dual. (118)
- ϕ : grau del preconditionador de l'algorisme primal-dual per a fluxos multiarticls. (143)

Índex de matèries.

- #nblocs, 51
- #nmb, 51
- ANSI-C, 77, 123, 149
- aplicacions, 10
- aproximació tangencial, 16
- arc
 - arrel, 8, 19, 130, 132, 134
 - bàsic, 21
 - complementari, 21, 29, 30, 35, 38, 40, 43–45, 47
- barrera logarísmica, 111, 118, 122, 145
- BFGS, actualització de l'hessiana, 68
- Bunch-Parlett, factorització, 109, 117
- Cholesky, factorització, 109, 119, 120, 136–138, 146, 160
- complement de Schur, 138
- complementarietat, 113
- conjunt de constriccions actives, mètode, 57, 61
- coordinació hidrotèrmica, 88, 93, 95, 150
 - curt termini, 90, 93
 - llarg termini, 88, 93
- descomposició
 - de Dantzig-Wolfe, 14
 - dictada per preus, 14, 16, 18
 - dictada per recursos, 15, 16
- descomposició de dominis, 137, 138
- DIMACS, 81, 82, 150, 155
- el·lipsoid, mètode de, 17
- escalat afí, mètodes de, 17, 109
- escalat afí dual, mètode de, 17
- estacionari, punt, 60
- estructures de dades, 48, 53, 64, 74
- exploració lineal, 59
- factibilitat dual, 113, 121, 122, 145, 150, 151
- factibilitat primal, 113, 122, 145, 150
- factorització simbòlica, 109, 119, 121, 146
- fase
 - 0, 48, 49, 64, 78, 80, 95, 102
 - 1, 48, 49, 52, 64, 74, 78, 80, 102
 - 2, 48, 50, 52, 64, 72, 74, 78, 80, 102
- flux concurrent, problema de, 18
- formulació
 - arc-camí, 8
 - arc-node, 8
 - problema (**F0**), 48
 - problema (**F1**), 49
 - problema (**FML**), 7
 - problema (**FMLCB**), 9
 - problema (**FMN**), 9
 - problema (**FMNCB**), 9
 - problema (**NCL**), 57
 - problema (**PZÂ**), 59
- Fortran-77, 77, 149
- freecycle, punter, 54
- gap dual, 118, 122, 145, 150, 151
- generador automàtic
 - Grid-on-torus, 81, 83, 95, 150
 - Gridgen, 81, 85, 95, 150
 - Gridgraph, 81, 84, 95, 150
 - Mnetgen, 81, 86, 95, 150
 - Rmfgen, 81, 82, 95, 150
- gradient conjugat preconditionat, mètode, 139, 140, 143–146, 150, 152, 153
- gradient conjugat, mètode, 65, 139
- gradient projectat, vector (veure “gradient reduït”)
- gradient reduït, vector, 59, 61–63, 68, 71–73
- gradient, vector, 57, 60

- grau del preconditionador, 143, 146
- Hellerman i Rarick, algorisme de, 53
- hessiana projectada, matriu (veure “hessiana reduïda”)
- hessiana reduïda, matriu, 59, 61, 64, 65
- hessiana, matriu, 57
- implementació
 - particionament primal, 4, 48, 64
 - primal-dual, 118
 - punt interior multiarticle, 4, 145
- IP, 123, 128
- IPM, 149–151, 154–156
- Karush-Kuhn-Tucker, cond. d’optimalitat, 60, 112, 113
- Lagrange, multiplicadors, 60, 61, 111
- Lagrangiana, 111, 112, 122
- llibreria
 - Harwell, 53
 - NAG, 53
 - SPARSPAK, 119, 138, 146, 149
- LoQo, 95, 96, 100, 123, 128, 154, 157, 158
- LU, descomposició, 52, 53
- matriu
 - de constriccions actives, 58, 60, 61
 - de Givens, 68
 - de treball, 23, 24, 28, 29, 48, 51, 52, 55, 63, 78, 80, 96, 103, 138, 158
 - estesa, 31, 32
 - eta, 32, 37, 38, 42
 - eta columna, 32, 33, 40, 52
 - eta fila, 32, 33, 39, 40, 47
 - incidències arcs-nodes, 7
- MCNF85, 95, 96, 102, 155, 157
- MINOS, 69, 79, 95, 96, 102, 103, 107, 123, 128, 157
- Netlib, 123
- Newton, mètode, 114
- Newton-truncat, mètode de, 65, 70, 79, 105
- optimització subgradient, 16
- ordenació
 - del mínim grau, 119, 138, 146, 149
 - del mínim ompliment, 119
- orientació
 - inversa, 25–28
 - normal, 25–28
- P3, algorisme, 53
- particionament primal, 3, 16, 18–20, 22, 23, 29, 36, 48, 50, 57, 62, 64, 65, 74, 77, 95, 96, 108, 109
- pivotació degenerada, 49
- PPRN, 77–80, 95, 96, 102, 103, 107, 150, 154–158
- predictor-corrector, mètode, 109, 117, 128, 155, 158
- preordre, ordenació en, 53
- primal-dual, mètodes, 17, 95, 109, 110, 130, 133, 144
- problema
 - (**F0**), 48, 49, 64
 - (**F1**), 64
 - (**FML**), 19, 145, 149
 - (**FMLCB**), 19, 49, 50, 77, 129, 130, 133
 - (**FMNCB**), 19, 62, 77
 - (**NCL**), 60
- pseudofactible, punt, 49
- punt interior, mètodes de, 16, 158, 159
- punxa, 53
- quasi-actives, variables superbàsiques, 73
- quasi-Newton, actualització, 68
- quasi-Newton, mètode, 67, 70, 79, 105
- radi espectral, 142
- relaxació lagrangiana, 17
- rutina
 - F01BRF, 53
 - GENQMD, 119, 138, 146, 149
 - GETPTC, 79
 - GSFCT, 149

- GSSLV, 149
- MA28, 53
- SMBFCT, 149
- símplex, 3, 4, 13, 16, 17, 20, 22, 48, 65, 102, 154, 155
- subespai nul, 59, 61
- superbàsiques, variables, 57–64, 66, 67, 70, 71, 73–75, 79, 93, 108
- taxació, 48, 50, 61
- trànsit, problema d'assignació, 94
- transformació projectiva, 16
- UNIX, 95, 123, 150
- valor propi, 142
- vector
 - actives, 55, 74
 - asa, 119
 - cicles, 54
 - diag, 119
 - eta, 39, 42
 - folguescb, 55
 - folguescm, 55
 - iab, 53
 - ifillin, 119
 - ilnz, 119
 - inicicle, 54
 - ipf, 53
 - ipr, 53, 54
 - iti, 53
 - iab, 53
 - ipf, 53
 - ipr, 53
 - iti, 53
 - itv, 53
 - jple, 54
 - jsup, 74, 75
 - ka, 120
 - kple, 54
 - ksup, 74, 75
 - la, 120
 - lnz, 119
 - longcicle, 54
 - mnk, 53, 54
 - mnl, 53, 54
 - statarc, 74
 - stslcb, 74
 - stslcm, 74
 - va, 120

