UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Introducció a la programació lineal: Mètode Símplex i el problema del transport

Autor: Josep Baradat Marí

Directora: Dra. Susana Romano Rodríguez

Realitzat a: Departament de Matemàtica Aplicada

Barcelona, 13 de juny de 2022

Abstract

The following article presents the necessary knowledge to understand and solve linear programming problems.

Linear programming was born in 1947, when G.B. Dantzig suggested the simplex method in order to solve planification problems of the United States Air Forces. In this article it is explained how this method works and a series of examples are solved by using a program which has the method's algorithm implemented.

To do this, the definitions and basic concepts of linear programming are explained to understand the linear programming problems from an algebraic point of view. There are also explanations of the necessary concepts and definitions of convex analysis to understand this type of problems from a geometric point of view.

On the other hand, the concept of duality and the relation between primal linear programming problems and dual linear programming problems are explained.

Finally, this article presents the transportation problem, a linear programming practical case, and the algorithm in order to solve it. This algorithm has been implemented in a program and has been used to solve an example shown in this article.

Resum

En aquest treball es presenten els coneixements necessaris per a entendre i resoldre els problemes de programació lineal.

La programació lineal neix com a tal al 1947 quan G.B. Dantzig va proposar el mètode símplex per a resoldre problemes de planificació de les Forces Aèries dels Estats Units. En aquest treball, s'explica com funciona aquest mètode i es resolen un seguit d'exemples utilitzant un programa en què hi ha implementat l'algorisme del mètode.

Per a fer això, s'expliquen abans les definicions i els conceptes bàsics sobre la programació lineal per a entendre els problemes de programació lineal des d'un punt de vista algebraic. També s'expliquen els conceptes i les definicions necessaris de l'anàlisi convexa per a entendre aquests problemes des d'un punt de vista geomètric.

D'altra banda, s'introdueix el concepte de dualitat i la relació entre els problemes de programació lineal primals i duals.

Finalment, es planteja el problema del transport, un cas pràctic d'un problema de programació lineal, i l'algorisme per a resoldre'l. S'ha implementat aquest algorisme en un programa i s'ha utilitzat per a resoldre un exemple exposat en aquest treball.

2020 Mathematics Subject Classification. 90C05, 90C08

Agraïments

Amb aquest treball posaré fi, si tot va bé, a l'etapa estudiantil de la meva vida. Per a qui no ho sàpiga, abans d'estudiar matemàtiques vaig fer economia i abans d'això havia intentat estudiar matemàtiques a una altra universitat.

Han estat vuit anys plens d'experiències, d'estrès, de dubtes, d'alegries, d'algunes decepcions i de molt d'esforç. I vull remarcar aquest esforç i la meva força de voluntat, claus per a seguir endavant i arribar al final de la carrera. No sé on em poratrà la vida o si m'hauria portat al mateix lloc si no hagués estudiat matemàtiques. El que sí que sé és que per a mi el fet d'acabar aquesta carrera és un èxit personal que, per molt que pugui compartir amb molta gent que d'alguna manera o d'altra n'ha set partícip, no crec que ningú valori tant com jo.

Dit això, vull agrair a tots el professors que he tingut l'ajuda que m'han donat i la dedicació que han demostrat en la seva feina. En especial, vull agrair a la doctora Susana Romano l'ajuda i la paciència prestades a l'hora de fer aquest treball. També vull agrair als meus pares el suport econòmic que m'han donat tots aquests anys i també a tots els amics que m'han fet costat en algun moment al llarg d'aquesta etapa.

Finalment, vull agrair al professor Antonio Alegre l'empenta que em va donar en el meu pitjor moment perquè tornés a estudiar matemàtiques.

Índex

1 Introducció

La programació lineal forma part d'una branca de les matemàtiques aplicades, concretament la de la programació matemàtica. Aquesta és alhora una branca de la teoria d'optimització en la qual es busca minimitzar o maximitzar una funció objectiu d'n variables $x_1, ..., x_n$ subjecta a un nombre finit de restriccions, que poden ser igualtats o desigualtats. Aquest valor mínim o màxim serà el valor òptim que s'assolirà quan les variables prenen uns valors determinats.

Als segles XVII i XVIII matemàtics com Isaac Newton, Gottfried Wilhelm Leibniz, Daniel Bernoulli i Joseph-Louis Lagrange ja es plantejaven com trobar màxims i mínims de funcions subjectes a restriccions. No va ser fins el 1826 que Jean-Baptiste Joseph Fourier va fer els primers avenços en aquest camp en plantejar el mètode d'eliminació de Fourier per a resoldre un problema d'aquest tipus.

Ja en el segle XX, Charles-Jean de la Vallée Poussin va proposar un mètode els principis del qual serien anàlegs als de la futura programació lineal. Anys més tard, en 1939, Leonid Vitaliyevich Kantorovich va proposar un algorisme per a resoldre problemes de programació lineal. Malauradament, els seus raonaments no van sortir de la Unió Soviètica fins que la programació lineal ja s'havia convertit en una teoria ben formalitzada gràcies a George Bernard Dantzig i d'altres.

La programació lineal com a tal neix al 1947 quan G.B. Dantzig dissenya el mètode símplex per a resoldre problemes de planificació de les Forces Aèries dels Estats Units per al desplegament de tropes, entrenament i diferents temes logístics. El nom de programació lineal l'adopta Tjalling Charles Koopmans i fa referència a aquesta planaificació i no a la programació informatica com es podria pensar.

Tot i que no s'abordarà en aquest treball, arran de la programació lineal naixerà la programació lineal entera al 1958 gràcies a Ralph Edward Gomory. La diferencia principal amb la programació lineal és que es restringirà com a mínim una de les variables a prendre valors enters.

Ja en 1975, L.V. Kantorovich i T.C. Koopmans van rebre el Premi Nobel d'economia per la seva contribució a la teoria de l'assignació òptima de recursos. Paradoxalment, sembla ser que G.B. Dantzig no va rebre cap Premi Nobel tot i ser considerat el pare de la programació lineal perquè es va considerar el seu treball com a massa matemàtic.

En 1979, Leonid Genrikhovich Khachiyan va dissenyar l'algorisme de l'el·lipsoide, el qual permet demostrar que els problemes de programació lineal són resolubles de manera eficient.

Per acabar, caldria destacar que en 1984 Narendra Krishna Karmarkar introdueix el mètode del punt interior per a resoldre problemes de programació lineal, el qual va permetre dur a terme grans avenços teòrics i pràctics.

Ja des d'un principi, l'ús de la programació lineal i del mètode símplex es va estendre a molts altres camps, com per exemple el del sector industrial. Fins aquell moment, les desicions en la producció en aquest sector es prenien a través de l'experiència i la intuïció. La introducció de la programació lineal en aquesta àrea va permetre la millora de l'eficiència en la producció, gràcies a una optimització en la planificació. Això va suposar un canvi de paradigma, ja que fins llavors aquestes millores en l'eficiència havien estat motivades tan sols per millores tecnològiques. A més, va provocar que fos inviable ignorar els mètodes i els models matemàtics en entorns competitius, com és el lliure mercat.

La programació lineal també va tenir un profund impacte en l'economia. Ja en 1947, T.C. Koopmans va assenyalar que la programació lineal podria servir per a analitzar més profundament teories clàssiques, com el sistema proposat en 1874 per Marie-Esprit-Léon Walras. D'altra banda, també va ajudar a relacionar diferents teories purament matemàtiques prèviament existents com la geometria de conjunts convexos, problemes de combinatòria o la teoria dels jocs de dues persones.

Estructura de la Memòria

Després d'aquesta introducció, basada en les fonts [5] i [9], es farà una breu introducció sobre el que és la programació lineal i la no lineal al capítol 2.

Tot seguit, al capítol 3, s'explicaran un seguit de conceptes d'anàlisi convexa per a entendre la interprectació dels problemes de programació lineal des d'un punt de vista geomètric.

Llavors, al capítol 4, s'explicarà com funciona el mètode sìmplex, el qual permet resoldre problemes de programació lineal. Per a entendre millor aquest mètode i tot el que s'explicarà, es presentaran un seguit d'exemples.

A continuació, al capítol 5, s'explicaran els conceptes bàsics sobre la dualitat, així com la relació entre els problemes de programació lineal primals i duals.

Després, al capítol 6, s'introduirà el problema del transport, s'explicarà un mètode per a resoldre'l i es donarà un exemple per a facilitar-ne la comprensió.

Finalment, al capítol 7 s'exposen les conclusions del treball.

Objectius

L'objectiu d'aquest treball de fi de grau serà donar una base de coneixements necessària per a entendre què és la programació lineal, com plantejar els problemes que se'n deriven i com resoldre aquests problemes amb el mètode símplex. Per a fer això, es presentaran diferents resultats teòrics necessaris per a entendre aquest mètode des d'un punt de vista algebraic i geomètric.

Finalment, s'aprofundirà en una aplicació pràctica de la programació lineal com és el problema del transport.

2 Programació lineal i no lineal

En aquest capítol, les principals fonts consultades han set [3], [7] i [8]. També s'ha consultat [1] per a la part relativa a la programació no lineal.

La programació lineal i la no lineal són dues branques de la programació matemàtica.

Definició 2.1. *Funció objectiu: Funció que es vol maximitzar o minimitzar en un problema de programació lineal o no lineal.*

Definició 2.2. *Una funció $f(x)$ es diu que és lineal si és una funció de valors reals definida en un espai vectorial de dimensió n tal que per a qualsevol vector d'aquest espai $x = \alpha u + \beta v$, on u i v també són vectors d'aquest espai vectorial i α i β són escalars, es té que $f(x) = f(\alpha u + \beta v) = \alpha f(u) + \beta f(v)$*

La programacó lineal consisteix en la resolució de problemes en què s'ha de maximitzar o minimitzar una funció objectiu subjecta a unes restriccions i en què tant la funció objectiu com les restriccions són lineals.

En canvi, els problemes de programació no lineal poden tenir restriccions o funcions objectiu que no siguin lineals o fins i tot no tenir restriccions. Això es deu al fet que, a vegades, afegir restriccions limita el problema de tal manera que no representa la realitat. Un exemple de problema de programació no lineal seria quan es tinguessin economies d'escala (el cost unitari de producció decreix quan augmenta la quantitat produïda).

Pel que fa a la mida dels problemes, tant en programació lineal com no lineal, es pot determinar segons el nombre d'incògnites:

- Problemes a petita escala, si tenen 5 variables o menys
- Problemes a escala intermèdia, si tenen entre 5 i 100 variables
- Problemes a gran escala, si tenen 100 variables o més

Finalment, cal parlar de la teoria de la convergència. Aquesta prediu el temps que trigarà un algorisme a convergir cap a la solució del problema. Això permetrà determinar si un problema està resolt i comparar diferents algorismes per a un mateix problema.

2.1 Programació lineal

Aquest treball es centrarà en els problemes de programació lineal, així com en el problema del transport, una aplicació concreta.

Els problemes de programació lineal poden ser de maximització o de minimització. També poden estar en la seva *forma estàndard* o en la *forma canònica*. En resum, per a $i = 1, ..., m$ i $j = 1, ..., n$ es té:

	Minimització		Maximització	
Forma estàndard	Minimitzar	$\sum_{j=1}^{n} c_j x_j$	Maximitzar	$\sum_{j=1}^{n} c_j x_j$
	Subjecte a	$\sum_{j=1}^{n} a_{ij} x_j = b_i,$ $i = 1, ..., m$ $x_j \geq 0,$ $j = 1, ..., n$	Subjecte a	$\sum_{j=1}^{n} a_{ij} x_j = b_i,$ $i = 1, ..., m$ $x_j \geq 0,$ $j = 1, ..., n$
Forma canònica	Minimitzar	$\sum_{j=1}^{n} c_j x_j$	Maximitzar	$\sum_{j=1}^{n} c_j x_j$
	Subjecte a	$\sum_{j=1}^{n} a_{ij} x_j \geq b_i,$ $i = 1, ..., m$ $x_j \geq 0,$ $j = 1, ..., n$	Subjecte a	$\sum_{j=1}^{n} a_{ij} x_j \leq b_i,$ $i = 1, ..., m$ $x_j \geq 0,$ $j = 1, ..., n$

En aquest cas, la funció objectiu que es vol optimitzar és:

$$\sum_{j=1}^{n} c_j x_j$$

D'altra banda, per a $i = 1, ..., m$, les restriccions seran:

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \sum_{j=1}^{n} a_{ij} x_j \geq b_i, \sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

Està clar que n és el nombre de variables i m, el nombre de restriccions. Òbviament, $n \leq m$, perquè, si no, seria un sistema amb més restriccions que variables i, per tant, tindria una única solució o no en tindria cap.

Un problema d'optimització es un problema de programació lineal, si i només si es compleixen els suposits de:

- Proporcionalitat: Tota variable x_j contribueix al valor de la funció objectiu segons $c_j x_j$ i en la i-èssima restricció segons $a_{ij} x_j$.
- Activitat: El valor total de la funció objectiu és la suma dels valors $c_j x_j$ i el valor de la i-èssima restricció és la suma dels valors $a_{ij} x_j$.
- Divisibilitat: Les variables x_j poden prendre valors no enters.

Forma estàndard del problema

A partir d'ara s'utilitzarà la següent notació matricial per a facilitar-ne l'escriptura i la comprensió:

$$\begin{array}{rl} \text{Minimitzar:} & c^T x \\ \text{Subjecte a:} & Ax = b \\ & x \geq 0 \end{array}$$

En aquest cas:

- x és un vector d'incògnites n-dimensional
- A és una matriu de dimensió $m \cdot n$
- b és un vector m-dimensional
- c és un vector n-dimensional

Tots els problemes de programació lineal es poden transformar en problemes de minimització en forma estàndard.

Si el problema és de maximització en lloc de minimització, només cal multiplicar la funció objectiu per -1 i així el problema passa a ser de minimització. Després només caldria multiplicar per -1 el valor mínim que s'obtingués per a saber el valor màxim que s'assoliria.

Quan es passi el problema a un problema de minimització en forma estàndard, sempre es voldrà que els termes independents de les restriccions, el vector b, siguin positius. Si en alguna restricció no es compleix això es multiplicarà la igualtat o desigualtat per -1 a banda i banda. Cal destacar que en fer això s'invertiria la desigualtat.

Definició 2.3. *Variables de separació: Són variables fictícies que s'introdueixen a les restriccions d'un problema de progamació lineal que són desigualtats per tal de convertir-les en igualtats. Igual que les variables x_j, seran majors o iguals que zero.*

En introduir variables de separació en un problema de programació lineal, la matriu A tindrà en aquest cas tantes noves columnes com variables de separació s'hi afegeixin. Es poden distingir dos tipus de variables de separació diferents:

Cas 1: Si es té una restricció:

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \leq b_i$$

Suposem que $b_i \geq 0$, s'hi afegirà una variable de separació $y_i \geq 0$, anomenada en aquest cas variable de folgança, de manera que:

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n + y_i = b_i$$

Cas 2: Si la restricció és:

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \geq b_i$$

Suposem que $b_i \geq 0$, s'hi afegirà una variable de separació $y_i \geq 0$, anomenada en aquest cas variable excedent, de manera que:

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n - y_i = b_i$$

En la forma estàndard del problema sempre es complirà que $x_j \geq 0$ per a $j = 1, ..., n$. Si hi hagués una restricció de la forma $x_j \geq u_j$ per a algun j en concret, es canviaria questa variable per $x'_j = x_j - u_j \geq 0$. Si hi hagués una restricció que fos $x_j \leq u_j$ per a algun j en concret, es canviaria aquesta variable per $x'_j = u_j - x_j \geq 0$.

Si alguna de les variables x_j no ha de complir necessàriament que $x_j \geq 0$, aleshores es diu que x_j és una variable lliure. En aquest cas hi ha dues opcions:

- Fer el canvi $x_j = u_j - v_j$, amb $u_j \geq 0$ i $v_j \geq 0$. En aquest cas el problema passarà a tenir $n + 1$ variables.

- Eliminar la variable x_j juntament amb alguna de les restriccions. Això es podrà fer per a qualsevol restricció tal que el coeficient a_{ij} sigui diferent a zero. El que es farà és aïllar:

  $$x_j = \frac{(a_{i1}x_1 + ... + a_{i(j-1)}x_{j-1} + a_{i(j+1)}x_{j+1} + ... + a_{i(n)}x_n)}{a_{ij}}$$

  Aleshores es podrà descartar aquesta restricció i aquesta variable. Així el problema passarà a tenir $m - 1$ restriccions i $n - 1$ variables. Un cop trobada la solució del nou problema, amb aquesta igualtat es podrà trobar quin valor ha de prendre x_j en el problema original.

Solucions

En el problema estàndard de programació lineal, les restriccions venen donades pel sistema $Ax = b$. Cal recordar que A és una matriu $m \cdot n$ i se suposarà que $m < n$, perquè, si no, seria un sistema indeterminat o amb una única solució. A més, si les m files de A són linealment independents, es dirà que es compleix la hipòtesi del rang complet i $rang(A) = m$.

Definició 2.4. *Una solució d'un problema de programació lineal és una solució factible si es compleix la hipòtesi que $x \geq 0$.*

Definició 2.5. *Sigui B una submatriu de A de dimensió $m \cdot m$ formada per columnes de A no singular. Llavors, si les $n - m$ variables x_j no associades a les columnes de A que conformen B s'igualen a zero, queda un sistema determinat $Bx = b$. La solució d'aquest sistema s'anomena solució bàsica respecte de la base B, i les m variables x_j associades a les columnes de A que conformen B s'anomenen variables bàsiques. Les altres variables seran les variables no bàsiques i la matriu A es pot representar com $A = (B|D)$, on D és la submatriu de A formada per les columnes associades a las variables no bàsiques.*

Sota la hipòtesi del rang complet abans introduïda, el sistema tindrà solució i com a mínim una solució bàsica. D'ara endavant, se suposarà que es compleix aquesta hipòtesi per a simplificar els enunciats. Si no és així, és fàcil modificar el problema perquè es compleixi.

Definició 2.6. *Si una de les variables bàsiques és igual a zero en la solució, es dirà que és una solució bàsica degenerada.*

Si la solució bàsica és no degenerada, es podran identificar immediatament les variables bàsiques amb les components positives de la solució.

Definició 2.7. *Una solució bàsica serà una solució òptima si és una solució del problema. La solució òptima pot ser o no ser degenerada, però segur que serà factible i bàsica.*

Teorema fonamental de la programació lineal

Per a buscar una solució òptima per a un problema de programació lineal només cal considerar les solucions factibles bàsiques. Cal tenir en compte que per a un problema de n variables i m restriccions hi haurà com a molt $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ solucions bàsiques. Això correspon a les possibles formes de seleccionar m de les n columnes que conformen A.

Teorema 2.8. *Teorema fonamental de la programació lineal:*

Donat un problema de programació lineal en forma estàndard:

$$\begin{array}{rc} \textit{Minimitzar:} & c^T x \\ \textit{Subjecte a:} & Ax = b \\ & x \geq 0 \end{array}$$

On la matriu A és una matriu $m \cdot n$ de rang m (com s'ha explicat abans, es parteix de la hipòtesi de rang complet), es tindrà que:

1) Si hi ha una solució factible, hi haurà una solució factible bàsica.

2) Si hi ha una solució factible òptima, hi haurà una solució factible bàsica òptima.

Demostració de 1)**:**

Siguin $a_1, ..., a_n$ les columnes de la matriu A. Suposem que es té una solució factible $x = (x_1, ..., x_n)^T$. Aleshores, per ser x una solució factible se sap que:

$$x_1 a_1 + ... + x_n a_n = b \text{ i } x \geq 0$$

Suposem ara que p variables x_j són estrictament majors que zero. Sense pèrdua de generalitat, suposem que ho són les p primeres. Aleshores:

$$x_1 a_1 + ... + x_p a_p = b \text{ i } x \geq 0$$

Aleshores, se'n deriven dos casos en funció de si les columnes a_j amb $j = 1, ..., n$ són linealment independents o no:

1) Se sap que $p \leq m$. Si les columnes $a_1, ..., a_p$ són linealment independents i si $p = m$ la solució és bàsica i la demostració està completa.

Si $p < m$, com que A té rang m, es poden trobar $m-p$ columnes de les $a_{p+1}, ..., a_n$ tal que siguin linealment independents de les p primeres. Aleshores assignant a les variables x_j d'aquests $m - p$ columnes el valor zero i agafant el valor $x_1, ..., x_p$ per a les p primeres variables, s'obté una solució bàsica factible (degenerada, en aquest cas).

2) Si $a_1, ..., a_p$ són linealment dependents, aleshores hi haurà una combinació lineal no trivial d'aquests vectors igual a zero:

$$y_1a_1 + ... + y_pa_p = 0$$

On les y_i són constants, no totes iguals a zero i almenys una d'elles és positiva (si no, multiplicant per -1 a banda i banda ja es té alguna y_i positiva).

Aleshores, es multiplica aquesta equació per ε i es resta a $x_1a_1 + ... + x_pa_p = b$. Així s'obté:

$$(x_1 - \varepsilon y_1)a_1 + ... + (x_p - \varepsilon y_p)a_p = b$$

Aquesta igualtat és certa per a qualsevol ε i $x - \varepsilon y$ serà per a qualsevol ε una solució del problema, tot i que no serà una solució factible per a qualsevol ε.

A mesura que augmenta ε, les components $x_j - \varepsilon y_j$ creixen si y_j és negativa; es queden iguals si y_j és zero i decreixen si y_j és positiva. S'ha vist que com a mínim hi ha una constant y_j que és positiva; per tant la component associada a aquesta constant decreixerà a mesura que augmenta ε.

S'agafa $\varepsilon = min(\frac{x_j}{y_j} \| y_j > 0)$, que és el primer ε pel qual una o més components passen a ser zero. Aleshores, per a aquest ε es té una solució factible amb p-1 variables positives (com a molt).

Aquest procés es va repetint fins a tenir una solució factible amb les columnes a_j corresponents linealment independents. Un cop arribats en aquest punt, es pot aplicar el que s'ha vist en el cas de columnes linealment independents. $\square$

Demostració de 2)**:**

Sigui $x = (x_1, ..., x_n)^T$ una solució factible òptima, suposem sense pèrdua de generalitat que les p primeres variables $x_1, ..., x_p$ són positives.

En el cas que $a_1, ..., a_p$ siguin linealment independents, el raonament és igual que en la demostració de 1). En el cas que siguin linealment dependents, també es demostra igual que en el cas 1), però ara cal demostrar que la solució és òptima per a qualsevol ε.

S'observa que per a qualsevol ε el valor que pren la funció objectiu és $c^T x - \varepsilon c^T y$, on y =(y_1,...,y_p, 0,...,0)^T. Per a ε suficientment petit, $x - \varepsilon y$ és una solució factible per a valors positius i negatius de ε. Per tant, $c^T y = 0$, ja que si no fos així existiria un ε prou petit i amb signe apropiat tal que $c^T x - \varepsilon c^T y$ seria menor que $c^T x$ cosa que conservaria la factibilitat però violaria el supòsit d'òptima de la solució x.

Un cop establert que la nova solució factible amb menys components també és òptima, es procedeix com en el cas 1). $\square$

3 Convexitat

En aquest capítol, les principals fonts consultades han set [3], [5], [7] i [8].

Fins ara, s'ha parlat dels problemes de programació lineal des d'un punt de vista algebraic, però també es poden interpretar des d'un punt de vista geomètric. Per a entendre aquesta relació, cal tenir clares un seguit de definicions i proposicions de l'anàlisi convexa que s'exposen a continuació.

Definició 3.1. *Un conjunt $C \subset \mathbb{R}^n$ és convex si per a tot $x_1, x_2 \in C$ i per a tot $\alpha \in \mathbb{R}$ tal que $0 \leq \alpha \leq 1$, el punt $\alpha x_1 + (1-\alpha)x_2 \in C$*

Definició 3.2. *Un punt x d'un conjunt convex C es diu punt extrem de C si no hi ha dos punts diferents x_1 i x_2 de C tals que $x = \alpha x_1 + (1-\alpha)x_2$ per a algun $0 < \alpha < 1$*

Definició 3.3. *Un conjunt $V \in \mathbb{R}^n$ es diu que és una varietat lineal si donats $x_1, x_2 \in V$ es té que $\lambda x_1 + (1-\lambda)x_2 \in V$ per a tot $\lambda \in \mathbb{R}$*

Definició 3.4. *Un hiperplà de $\mathbb{R}^n$ és una varietat lineal (n-1)-dimensional*

Proposició 3.5. *Sigui $a \in \mathbb{R}^n$ tal que $a \neq 0$ i sigui $c \in \mathbb{R}$, el conjunt:*

$$H = \{x \in \mathbb{R}^n \| a^T x = c\}$$

És un hiperplà de $\mathbb{R}^n$

Demostració:

Com que $a^T x = c$ és una equació lineal, està clar que H és una varietat lineal. Sigui x_1 qualsevol vector de H, traslladant per a $-x_1$ es considera $M = H - x_1$, que serà un subespai lineal de $\mathbb{R}^n$. Aquest subespai està format per tots els vectors x que satisfan que $a^T x = 0$, és a dir, tots els vectors ortogonals al vector a. Això és un subespai (n-1)-dimensional, i per tant, H també ho serà. $\square$

Proposició 3.6. *Sigui H un hiperplà de $\mathbb{R}^n$, aleshores existeixen $a \in \mathbb{R}^n$ tal que $a \neq 0$ i $c \in \mathbb{R}$ tal que $H = \{x \in \mathbb{R}^n \| a^T x = c\}$*

Demostració:

Sigui $x_1 \in H$ i es considera $M = H - x_1$. Com que H és un hiperplà, M serà un subespai (n-1)-dimensional. Sigui a qualsevol vector diferent a zero ortogonal a aquest subespai, és a dir, a pertany al subespai unidimensional $M^\perp$. És evident que $M = \{x \| a^T x = 0\}$. En fer $c = a^T x_1$ s'observa que si $x_2 \in H$ s'obté $x_2 - x_1 \in M$, i així $a^T x_2 - a^T x_1 = 0$, el qual implica que $a^T x_2 = c$. Per tant, $H \subset \{x \| a^T x = c\}$. Però com que H i $\{x \| a^T x = c\}$ són de dimensió n-1 per la proposició 3.5., està clar que han de ser iguals. $\square$

Definició 3.7. *Sigui $a \in \mathbb{R}^n$ tal que $a \neq 0$ i sigui $c \in \mathbb{R}$, es considera l'hiperplà $H = \{x \in \mathbb{R}^n \| a^T x = c\}$. Aleshores, el semiespai tancat positiu serà $H_+ = \{x \in \mathbb{R}^n \| a^T x \geq c\}$ i l'obert serà $H_+^o = \{x \in \mathbb{R}^n \| a^T x > c\}$. Igualment, el semiespai tancat negatiu serà $H_- = \{x \in \mathbb{R}^n \| a^T x \leq c\}$ i l'obert serà $H_-^o = \{x \in \mathbb{R}^n \| a^T x < c\}$. Els semiespais són conjunts convexos i $H_+ \cup H_- = \mathbb{R}^n$.*

Definició 3.8. *Un conjunt que es pot expressar com la intersecció d'un nombre finit de semiespais tancats es diu plítop convex.*

Definició 3.9. *Un polítop fitat no buit s'anomena poliedre.*

3.1 Equivalència entre punts extrems i solucions bàsiques

Per a entendre l'equivalència entre punts extrems i solucions bàsiques, primer cal demostrar que el conjunt de solucions factibles (aquelles per a les quals $x \geq 0$) que inclou les solucions bàsiques, és convex.

Teorema 3.10. *El conjunt K de solucions factibles d'un problema de programació lineal és convex.*

Demostració:

S'ha de veure que qualsevol combinació convexa de solucions factibles és una solució factible. És obvi que si aquest conjunt només té un element, el teorema és cert. Si en té més d'un, per exemple x_1 i x_2 (dues solucions bàsiques), aleshores:

$$Ax_1 = b \text{ amb } x_1 \geq 0$$

$$Ax_2 = b \text{ amb } x_2 \geq 0$$

Aleshores, per a $0 \leq \alpha \leq 1$, sigui $x = \alpha x_1 + (1-\alpha)x_2$ una combinació convexa de x_1 i x_2. És obvi que $x \geq 0$ perquè $x_1 \geq 0$, $x_2 \geq 0$, $\alpha \geq 0$ i $(1-\alpha) \geq 0$. Aleshores, es comprova fàcilment que x també és una solució factible:

$$Ax = A(\alpha x_1 + (1-\alpha)x_2) = \alpha Ax_1 + (1-\alpha)Ax_2 = \alpha b + (1-\alpha)b = b$$

Efectivament, x és una solució factible i, per tant, K és convex. □

Un cop introduïts els conceptes de solucions bàsiques i les definicions, les proposicions i el teorema anteriors, es pot entendre el següent teorema que permet relacionar les explicacions geomètriques i algebraiques de la programació lineal.

Teorema 3.11. *Equivalència de punts extrems i solucions bàsiques*

Sigui A una matriu $m \cdot n$ de rang m i b un m-vector. Es considera el polítop convex K format per tots els n-vectors x que satisfan $Ax = b$ i $x \geq 0$.

Un vector x és un punt extrem de K si i només si x és una solució factible bàsica de $Ax = b$ amb $x \geq 0$.

Demostració:

Sense pèrdua de generalitat, suposem que $x = (x_1, ..., x_m, 0, ..., 0)$ és una solució factible bàsica de $Ax = b$ amb $x \geq 0$. Aleshores:

$$x_1a_1 + ... + x_ma_m = b$$

On $a_1, ..., a_m$ són les m primeres columnes de la matriu A i són linealment independents.

Suposem que x es pot formular com una combinació convexa de dos punts de K: $x = \alpha y + (1-\alpha)z$ amb $0 < \alpha < 1$, $y \neq z$ i $y, z \in K$. Totes les components de x, y i z són positives i α també ho és. Aleshores, com que les $m-n$ components de x són zero, les $m-n$ components de y i z també ho seran. Per tant, es té:

$$y_1a_1 + ... + y_ma_m = b$$

$$z_1a_1 + ... + z_ma_m = b$$

Però com que els vectors $a_1, ..., a_m$ són linealment independents, això implica que $x = y = z$. Per tant, x és un punt extrem de K.

Per a demostrar el reciproc, suposem que x és un punt extrem de K. Suposem, sense pèrdua de generalitat, que les components de x diferents de zero són les k primeres i són positives. Aleshores:

$$x_1a_1 + ... + x_ka_k = b$$

Per a demostrar que x és una solució factible bàsica, s'ha de demostrar que els vectors $a_1, ..., a_k$ són linealment independents.

Per a demostrar-ho, es farà reducció a l'absurd. Suposem que fossin linealment dependents. Aleshores, existiria una combinació $y_1a_1 + ... + y_ka_k = 0$ amb no tots els $y_1, ..., y_k$ iguals a zero. Per tant, es considera el n-vector $y = (y_1, ..., y_k, 0, ..., 0)$. Com que per al vector x, $x_i > 0$ per a $1 \leq i \leq k$, es pot seleccionar ε tal que $x + \varepsilon y \geq 0$ i $x - \varepsilon y \geq 0$.

Està clar que $x = \frac{1}{2}(x + \varepsilon y) + \frac{1}{2}(x - \varepsilon y)$. Per tant, x és una combinació convexa de dos vectors de K. S'ha arribat a una contradicció, ja que x és un punt extrem de K.

Aleshores, $a_1, ..., a_k$ són linealment independents i x és una solució factible bàsica (tot i que si $k < m$, és una solució factible bàsica degenerada). □

D'aquest teorema es deriven un seguit de corol·laris sobre el polítop convex K que defineixen el conjunt de restriccions d'un problema de programació lineal.

Corołari 3.12. *Si el conjunt $K = \{x | Ax = b, x \geq 0\}$ és no buit, aleshores té almenys un punt extrem.*

Demostració:

Com s'ha demostrat en el teorema 3.11., veure que K té algun punt extrem és equivalent a veure que existeixen solucions bàsiques factibles per al problema que defineix. Com s'ha demostrat en el teorema 2.8., si hi ha una solució factible per al problema que defineix, hi haurà una solució bàsica factible. Sota la hipòtesi de rang complet, que suposem sempre certa, se sap que existeix alguna solució bàsica factible. □

Corołari 3.13. *Si hi ha una solució finita a un problema de programació lineal, hi haurà una solució òptima finita que és un punt extrem del conjunt de restriccions.*

Corołari 3.14. *El conjunt de restriccions K corresponent a $Ax = b$ i $x \geq 0$ té com a molt un nombre finit de punts extrems.*

Corołari 3.15. *Si el polítop convex K corresponent a $Ax = b$ i $x \geq 0$ és fitat, aleshores K és un poliedre convex, és a dir, K es compon de punts que són combinacións convexes d'un nombre finit de punts.*

3.2 Interpretació geomètrica del problema

Amb el que s'ha vist fins ara, ja es pot entendre la interpretació geomètrica dels problemes de programació lineal.

En $\mathbb{R}^n$, una igualtat de la forma $a_1x_1 + ... + a_nx_n = h$, on els x_i són variables i h i els a_i són constants, determina un hiperplà, de dimensió $n-1$ per definició. En canvi, una desigualtat de la forma $a_1x_1 + ... + a_nx_n \leq h$ divideix l'espai $\mathbb{R}^n$ en dos semiespais.

Aleshores, donat un problema de programació lineal, el conjunt de les restriccions del problema delimitarà un polítop convex a l'espai $\mathbb{R}^n$. Si aquest polítop convex és fitat, serà un poliedre. Llavors, pel teorema 3.11., només caldrà mirar els punts extrems del polítop convex, que seran vèrtexs si és un poliedre, per a trobar la solució factible òptima del problema.

Exemple 3.16. Un problema en $\mathbb{R}^3$ podria ser:

$$\begin{array}{rc} \text{Maximitzar:} & 3x_1 + 2x_2 + 5x_3 \\ \text{Subjecte a:} & 2x_1 + x_2 \leq 4 \\ & x_3 \leq 5 \\ & x_1, x_2, x_3 \geq 0 \end{array}$$

Cada una de les restriccions, divideix l'espai $\mathbb{R}^3$ en dos subespais. En un subespai es complirà la restricció i en l'altre no.

En aquest cas, les cinc restriccions determinaran un prisma com a la figura 3.17. Cada igualtat de cada restricció determinarà una cara del prisma. En concret:

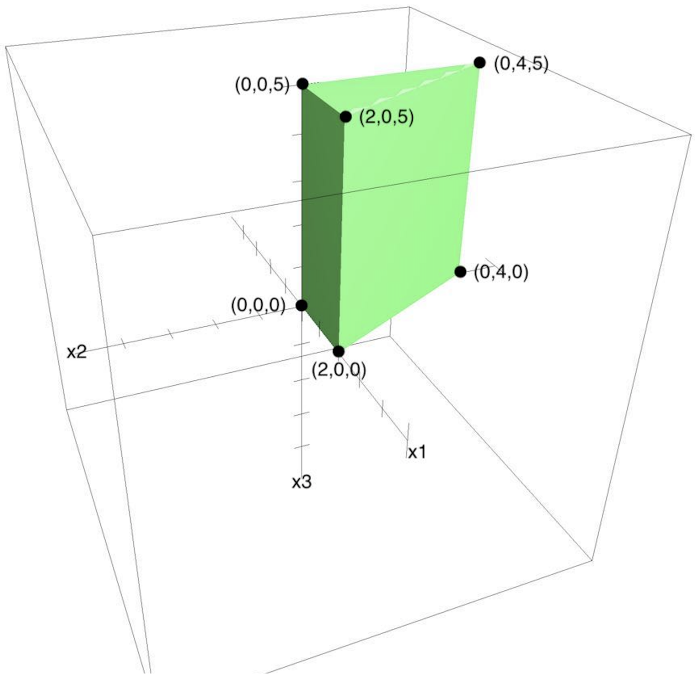
- $x_3 = 0$ correspon a la cara de sota.
- $x_3 = 5$ correspon a la cara de sobre.
- $x_2 = 0$ determina la cara sobre l'eix x_1.
- $x_1 = 0$ determina la cara sobre l'eix x_2.
- $2x_1 + x_2 = 4$ determina la cara que falta.

En aquest cas, com es veu a la figura 3.17., el polítop convex és fitat, per tant, és un poliedre. Aleshores, pel teorema 3.11., se sap que la solució òptima serà un dels vèrtexs. En la següent taula es veuen quins valors pren la funció objectiu per a cada vèrtex:

Vèrtex	Valor de la funció
$(0,0,5)$	25
$(0,0,0)$	0
$(0,4,5)$	33
$(2,0,5)$	31
$(0,4,0)$	8
$(2,0,0)$	26

Per tant, està clar que la solució òptima serà $(0,4,5)$, on la funció objectiu prendrà el valor de 33.

Figura 3.17.



Cal destacar que no sempre hi haurà solucions òptimes per als problemes de programació lineal. En l'exemple anterior, si es considerés la mateixa funció objectiu i les mateixes restriccions excepte $2x_1 + x_2 \leq 4$, es tindria un polítop no fitat. En aquest cas, no hi hauria una solució òptima. Per exemple, es podria fixar $x_1 = 1$ i $x_3 = 1$ i llavors agafar x_2 tan gran com es volgués, fent que el valor de la funció objectiu tendís a infinit.

4 Mètode sìmplex

En aquest capítol les principals fonts consultades han set [3], [4], [5], [8], [9] i [10].

Com s'ha explicat abans, tot problema de programació lineal es pot transformar en un problema de minimització en forma estàndard. Aleshores, el mètode sìmplex consisteix a passar d'una solució factible bàsica a una altra mentre, el valor de la funció objectiu vagi decreixent. Vist des d'un punt de vista geomètric, seria el mateix que anar passant d'un punt extrem a un altre.

Cal destacar que aquest mètode és el més important de la programació lineal, però encara no existeix una teoria de la convergència que se li pugui aplicar. Tal com s'explica a la referencia [4], com que el mètode sìmplex va passant per les diferents solucions bàsiques, se sap que no hi haurà més de $\binom{m}{n}$ pivotacions. Aquesta fita pot ser molt gran, però a través de l'observació de diferents casos, es va formular la conjectura de Hirsch (1957 - William Morris Hirsch). Aquesta conjectura ve a dir que:

- 3(m-n) pivotacions haurien de ser suficients per a resoldre qualsevol problema de programació lineal.
- $\frac{3}{2}$(m-n) sol ser la mitjana de pivotacions que es fan.
- Per a qualsevol problema es pot trobar una variant de mètode simplex que el resol amb m-n pivotacions.

Per a entendre aquest mètode, es partirà de la suposició de no degeneració: tota solució factible bàsica és no degenerada. Això simplifica molts els arguments. Si no fos així, existeixen mètodes per a corregir-ho.

Finalment, tal com s'explica a la referència [9], és important destacar dos altres mètodes que no s'estudiaran en aquest treball:

- El mètode de l'el·lipsoide: No pot competir amb el mètode símplex a la pràctica, però és important perquè va ser el primer mètode de programació lineal pel que es va demostrar que funcionava en temps polinomial (és a dir, que a partir d'un seguit de variables pròpies del problema es pot fitar el temps de convergència).
- El mètode del punt interior: És, de fet, un conjunt de mètodes. Per a alguns ha estat demostrat que funcionen en temps polinomial, mentre que per d'altres no. Depenent del problema, pot ser que algun d'aquests mètodes funcioni millor que el mètode símplex.

4.1 Pivotatge

El procediment de pivotatge, és el primer pas per a entendre el mètode símplex. Aquest procediment es pot interpretar des de dos punts de vista, segons la dualitat que s'explica més endavant.

Primera interpretació

Sigui el sistema $Ax = b$ de les restriccions:

$$\begin{array}{cccc} a_{11}x_1 & +...+ & a_{1n}x_n & = & b_1 \\ \vdots & & \vdots & & \vdots \\ a_{m1}x_1 & +...+ & a_{mn}x_n & = & b_m \end{array}$$

Es pot escriure això com un conjunt de m equacions de la forma:

$$a^1x = b_1,\ ...,\ a^mx = b_m$$

En el cas que $m < n$, no hi hauria una solució única, sinó un conjunt de solucions. Però afegint un seguit d'equacions de la forma $e^kx = 0$, linealment independents entre elles i a les que ja es tenen, on e^k és el vector k-éssim de la base canònica, es tindrà una solució bàsica.

En aquest cas, com que les equacions són linealment independents, es pot substituir qualsevol d'elles per una combinació lineal de l'equació triada i de la resta de les equacions. Això és al, cap i a la fi, fer eliminació gaussiana. Això permetrà convertir el sistema en:

$$\begin{array}{ccccccccccc} x_1 & & & & & + & y_{1(m+1)}x_{m+1} & +...+ & y_{1n}x_n & = & y_{10} \\ & x_2 & & & & + & y_{2(m+1)}x_{m+1} & +...+ & y_{2n}x_n & = & y_{20} \\ & & x_3 & & & + & y_{3(m+1)}x_{m+1} & +...+ & y_{3n}x_n & = & y_{30} \\ & & & \ddots & & & \vdots & & \vdots & & \vdots \\ & & & & x_m & + & y_{m(m+1)}x_{m+1} & +...+ & y_{mn}x_n & = & y_{m0} \end{array}$$

I per tant $x = (y_{10}, ..., y_{m0}, 0, ..., 0)^T$ és una solució bàsica del sistema.

Definició 4.1. *Quan es tinguin m variables bàsiques cadascuna de les quals només apareixen en una equació, es dirà que la taula està en forma canònica. No s'ha de confondre això amb que un problema estigui en forma canònica.*

El pivotatge consistirà en que si es vol substituir la variable bàsica x_p amb $1 \leq p \leq m$ per una variable no bàsica x_q amb $m < q \leq n$, es podrà fer si el coeficient y_{pq} és diferent de zero. El que es farà serà dividir la fila p per y_{pq}, per tal d'obtenir un 1 en la posició on abans hi havia y_{pq}. Tot seguit, es restarà a les altres files la fila p multiplicada per $-y_{iq}$ per a $i = 1, ..., p-1, p+1, ..., m$ per tal de que la columna q sigui tota de zeros (excepte l'element de la fila p que serà 1). Si es denota y'_{ij} als nous coeficients, les fórmules que els donaran són:

$$y'_{pj} = \frac{y_{pj}}{y_{pq}}$$

$$y'_{ij} = y_{ij} - \frac{y_{pj}}{y_{pq}}y_{iq} \text{ per a } i \neq p$$

Segona interpretació

Una altra interpretació del sistema per a entendre el pivotatge és veure'l com una equació vectorial $x_1 a_1 + ... + x_n a_n = b$, on els a_i amb $i = 1, ..., n$ són les columnes de la matriu A.

Si $m < n$, els vectors a_i formen una base de $\mathbb{R}^m$ i no hi ha una solució única del sistema. Però com que la matriu A té rang m, se sap que es pot expressar b com una combinació lineal de m vectors a_i. Les variables x_i associades a aquests vectors seran variables bàsiques. La resta de les variables s'igualaran a zero, i així s'obté una solució bàsica. Si es torna a tenir la taula en forma canònica, es tindrà:

$$\begin{array}{cccccccc|c}
1 & 0 & ... & 0 & y_{1(m+1)} & \cdots & y_{1n} & y_{10} \\
0 & 1 & ... & 0 & y_{2(m+1)} & \cdots & y_{2n} & y_{20} \\
\vdots & \vdots & \ddots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & ... & 1 & y_{m(m+1)} & \cdots & y_{mn} & y_{m0}
\end{array}$$

Aleshores, està clar que es pot expressar qualsevol columna de A com una combinació lineal de les m primeres:

$$a_j = y_{1j} a_1 + y_{2j} a_2 + ... + y_{mj} a_m \text{ per } j = 1, ..., n$$

Per tant, es poden entendre les columnes de la taula com els coeficients de la combinació lineal dels vectors de la base que formen la columna.

Un altre cop, es vol substituir a_p amb $1 \leq p \leq m$ per a_q amb $m < q \leq n$. Es podrà fer si i només si els vectors que conformen el que serà la nova base són linealment independents. Això passarà si i només si $y_{pq} \neq 0$.

Qualsevol vector a_q es pot expressar com:

$$a_q = \sum_{i=1 i \neq p}^{m} (y_{iq} a_i) + y_{pq} a_p$$

D'on es pot aïllar a_p. Substituint aquesta a_p en la combinaciò lineal d'a_j anterior, s'obté que:

$$a_j = \sum_{i=1 i \neq p}^{m} (y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj}) a_i + \frac{y_{pj}}{y_{pq}} a_q$$

I d'això s'obtenen les equacions per a calcular y'_{pj} i y'_{ij} per $i \neq p$, que són iguals que en la primera interpretació.

4.2 Variable a incorporar

Partint d'una solució factible bàsica, es vol passar a una altra que doni un resultat millor. El primer pas per a fer això és triar quina variable incorporar a la base.

Teorema 4.2. *Millora de la solució factible bàsica:*

Donada una solució factible bàsica no degenerada que aplicada a la funció objectiu dona z_0, sent $z_0 = c_1x_1 + ... + c_nx_n$. Si per a algun $1 \leq j \leq n$ es té que $c_j - z_j < 0$, on $z_j = y_{1j}c_1 + ... + y_{mj}c_m$. Aleshores, hi ha una solució factible amb valor objectiu $z' < z_0$.

Si la columna a_j es pot substituir per un vector de la base original i generar una nova solució factible bàsica, aquesta nova solució tindrà valor $z' < z_0$ en la funció objectiu. En cas contrari, el conjunt K de solucions no està fitat i la funció objectiu es pot fer arbitràriament petita.

Demostració:

Sense pèrdua de generalitat, sigui $x = (x_0, ..., x_m, 0, ..., 0)^T$ una solució factible bàsica tal que $z_0 = c^Tx$. Sense pèrdua de generalitat, suposem que $c_{m+1} - z_{m+1} < 0$. Aleshores, es pot construir una solució factible de la forma $x' = (x'_1, ..., x'_m, x'_{m+1}, 0, ..., 0)^T$ amb $x'_{m+1} > 0$. Aleshores, com que $z' = c^Tx'$, es té que $z' - z_0 = (c_{m+1} - z_{m+1})x'_{m+1} < 0$. Per tant, $z' < z_0$.

Es desitja agafar el x'_{m+1} més gran possible. En augmentar aquesta variable, les altres es queden igual, disminueixen o augmenten. Per tant, s'augmentarà aquesta variable fins que $x'_i = 0$ on $1 \leq i \leq m$, on s'obtindrà una nova solució factible bàsica. Si x'_{m+1} pot augmentar tant com es vulgui sense que cap variable bàsica original passi a ser zero, voldrà dir que el conjunt de solucions no està fitat. □

Teorema 4.3. *Condició d'optimitat:*

Si per a alguna solució factible bàsica es té que $c_j - z_j \geq 0$ per a tot j, aleshores és una solució òptima.

Demostració:

Tal com es demostra a la pàgina 41 de la referència [8], s'observa que:

$$z = c^Tx = z_0 + (c_{m+1} - z_{m+1})x_{m+1} + ... + (c_n - z_n)x_n$$

Per a les variables bàsiques està clar que $c_i - z_i$ és zero. Aleshores, si per a totes les variables no bàsiques es té que $c_j - z_j \geq 0$, llavors canviant qualsevol variable bàsica per una no bàsica el valor z obtingut augmentaria. Per tant, és una solució òptima. □

A partir d'ara s'utilitzarà la nomenclatura $r_j = c_j - z_j$. Aquest r_j és el coeficient de cost relatiu. Una interpretació econòmica d'això és que es mesura el cost d'una variable en relació a la base donada. En aquesta interpretació, c_j seria el cost d'augmentar la variable x_j i z_j seria l'estalvi en augmentar-la. Per tant, com més negatiu sigui r_j, més a compte sortirà incloure la variable x_j a la base.

En resum, es triarà la variable a incorporar a la base en funció del coeficient de cost relatiu. En el mètode símplex original, es va proposar triar aquella variable que tingués un coeficient de cost relatiu més negatiu. Si aquesta variable no es pot substituir a la base, voldrà dir que el conjunt de solucions no està fitat. Si la variable es pot substituir a la base, es repetirà aquest procés fins que tots els coeficients de cost relatius siguin majors o iguals a zero. En aquest cas s'haurà arribat a una solució òptima.

Cal destacar que, a part de triar la variable amb el coeficient de cost relatiu més negatiu, hi han altres maneres de triar la variable que entra a la base. En 1977 Robert Gary Bland va proposar un refinament del mètode símplex , en qu¨, donat un ordre arbitrari de les variables que es manté al llarg de tot el mètode, la variable no bàsica a incorporar a la base serà la primera que es trobi amb un coeficient de cost relatiu negatiu, seguint l'ordre donat. A més, també s'utilitzarà aquesta regla per a triar la variable que surt de la base en cas que se'n trobin dues igual de vàlides (es parlarà de com triar la variable que surt de la base a la següent secció). Se'n pot trobar una demostració a la referència [5], pàgina 37. Tot i això, s'ha demostrat a través de diferents tests empírics que la regla de Bland és més ineficient que altres mètodes posteriors.

4.3 Variable a treure

Suposem que es té una solució factible bàsica $(x_1, ..., x_m, 0, ..., 0)$ i que s'ha decidit incorporar la variable associada a la columna a_q amb $q > m$. Aquesta columna expressada en la base actual és:

$$a_q = y_{1q}a_1 + ... + y_{mq}a_m \tag{4.1}$$

Com que es té que:

$$x_1a_1 + .. + x_ma_m = b \tag{4.2}$$

Restant la igualtat (4.1) multiplicada per ε a la igualtat (4.2) s'obté:

$$(x_1 - \varepsilon y_{1q})a_1 + ... + (x_m - \varepsilon y_{mq})a_m + \varepsilon a_q = b$$

Així, per a qualsevol $\varepsilon \geq 0$ es pot expressar b com una combinació lineal de $m + 1$ vectors i s'obté una solució factible, tot i que no totes aquestes solucions són bàsiques. En funció de si els y_{iq} són positius, negatius o zero, els factors $(x_i - \varepsilon y_{iq})$ decreixen, creixen o es queden igual. Cal destacar que si el problema té m variables bàsiques que apareixen cada una només en una equació, es tindrà que $x_i = y_{i0}$. També cal destacar que si la taula està en forma canònica es tindrà que $x_i = y_{i0}$.

Aleshores, si s'agafa $\varepsilon = \min_i(\frac{y_{i0}}{y_{iq}}|y_{iq} > 0)$ s'obté una nova solució factible.

Si per a l'ε anterior $i = p$, vol dir que la variable a treure ha de ser la x_p. Si amb aquest ε dos factors es convertissin en zero alhora, es podria treure de la base qualsevol de les dues variables que hi estan associades. Si tots els $y_{iq} \leq 0$, vol dir que es podria incrementar ε tant com es volgués que s'obtindria una solució factible. Aleshores, voldria dir que el conjunt de solucions K no està fitat.

En resum, la variable que es traurà de la solució bàsica és x_i tal que i minimitza $\frac{y_{i0}}{y_{iq}}$ amb $y_{iq} > 0$.

4.4 Solució bàsica inicial

Ara que ja se sap com triar la variable a incloure en la nova solució bàsica i també com triar quina variable treure, només cal una solució bàsica inicial per a iniciar el mètode símplex.

Si les restriccions del problema són de la forma $Ax \leq b$, amb la introducció de variables de separació es pot obtenir una solució bàsica des d'on iniciar el mètode. Si, en canvi, les restriccions són de la forma $Ax = b$ una manera de trobar aquesta solució bàsica inicial podria ser utilitzant el mètode de dues fases o el mètode Big-M.

Mètode de dues fases

Definició 4.4. *Les variables artificials seran y_i amb $i = 1, ..., m$ i són variables que inicialment no formaven part del problema i que, si el problema té solució òptima, seran iguals a zero en aquesta solució.*

La primera fase d'aquest mètode consistirà en introduir variables artificials i plantejar un nou sistema. Concretament, consistirà en utilitzar les restriccions $Ax = b$ i $x \geq 0$, per a plantejar el problema:

$$
\begin{array}{rc}
\text{Minimitzar:} & \sum_{i=1}^{m} y_i \\
\text{Subjecte a:} & Ax + y = b \\
& x \geq 0 \\
& y \geq 0
\end{array}
$$

Aleshores, es trobarà una solució factible bàsica òptima utilitzant el mètode símplex com s'ha explicat fins ara, partint de la solució bàsica $y = b$. Si aquesta solució òptima té alguna variable artificial diferent de zero, voldrà dir que el problema original no té solució factible. Això és així perquè si el problema original tingués alguna solució factible bàsica x' tal que $Ax' = b$, llavors $Ax' + y = b$ implicaria que $y = (y_1, ..., y_m) = 0$ i la funció objectiu en aquest cas es minimitzaria prenent valor zero.

En el cas que les variables artificials siguin totes zero en acabar la primera fase, s'iniciarà la segona fase consistent en utilitzar la solució factible bàsica trobada com a punt de partida del mètode símplex per al problema original.

Mètode Big-M

Aquest mètode, també anomenat mètode de penalització, consisteix en introduir variables artificials i sumar-les a la funció objectiu multiplicades per un coeficient M positiu suficientment gran. Així el problema:

$$\begin{array}{rc} \text{Minimitzar:} & c^T x \\ \text{Subjecte a:} & Ax = b \\ & x \geq 0 \end{array}$$

Passaria a ser:

$$\begin{array}{rc} \text{Minimitzar:} & c^T x + \sum_{i=1}^{m} M y_i \\ \text{Subjecte a:} & Ax + y = b \\ & x \geq 0 \\ & y \geq 0 \end{array}$$

La idea d'aquest mètode és que amb un coeficient $M \geq 0$ prou gran, el mètode símplex trobarà una solució factible bàsica òptima que no inclou cap de les variables y_i per $i = 1, ..., m$, ja que incloure'n alguna suposaria sumar My_i, un valor gran, al valor de la funció objectiu.

Aleshores, si la solució del problema no està fitada, però totes les variables artificials són zero, voldrà dir que que el problema original té una solució òptima no fitada. D'altra banda, si la solució del problema no està fitada i alguna de les variables artificials és diferent de zero, voldrà dir que el problema original no és factible.

Si el nou problema té alguna solució òptima finita i totes les variables artificials són zero, aquesta solució tambè serà una solució òptima del problema original. Si el nou problema té alguna solució òptima finita, però alguna variable artificial és diferent a zero, voldrà dir que el problema original no té cap solució factible.

Tot i que aquest mètode pot resultar més atractiu que el de dues fases perquè només cal aplicar un cop el mètode símplex, té dos grans problemes:

- És difícil determinar com de gran ha de ser M.
- El valor gran de M dominarà completament al dels altres coeficients i això pot provocar errors d'arrodoniment.

4.5 Cicles

Abans s'ha dit que es partiria de la hipòtesi de no degeneració. En aquesta secció s'estudiarà què passa si no es parteix d'aquesta hipòtesi.

Si les solucions bàsiques factibles són no degenerades, el mètode símplex convergirà en un nombre finit de passos, ja sigui concloent que hi ha una solició òptima o concloent que la solució no està fitada. En cas contrari, és possible que el mètode símplex entrés en un cicle on passés sempre per unes mateixes solucions bàsiques que donen el mateix valor a la funció objectiu. Cal destacar que si hi ha solucions bàsiques degenerades, pot haver-n'hi més d'una associades al mateix punt extrem (cosa que no passarà si són totes no degenerades).

Per a evitar això existeixen diferents mètodes per a solucionar-ho. Per exemple, la regla de Bland explicada a l'apartat 4.2. serveix per a evitar-ho. Un altre mètode és la regla de sortida, que estableix un criteri per a triar la variable que surt de la base en cada iteració i assegura que no es repetirà cap base durant el desenvolupament del mètode símplex.

Regla de sortida

Sigui $x = \{x_B, x_D\}$ una solució bàsica, x_B les variables bàsiques d'aquesta i X_D les no bàsiques. Sigui x_q una variable no bàsica que es vol incorporar a la solució bàsica. Aleshores, tal com s'ha explicat abans, es calcula el conjunt de subíndexs de variables candidates a abandonar la base:

$$I_0 = \{i | \tfrac{y_{i0}}{y_{iq}} = \min_{1 \le r \le m} \tfrac{y_{r0}}{y_{rq}} : y_{rq} > 0\}$$

Si el conjunt I_0 té un únic element, s'agafa aquest subíndex. Si en té més d'un, es calcula:

$$I_1 = \{i | \tfrac{y_{i1}}{y_{iq}} = \min_{r \in I_0} \tfrac{y_{r1}}{y_{rq}}\}$$

Un altre cop, si I_1 té un únic element, es tria aquest com a subíndex. En cas contrari, es calcula I_2 i així successivament. En general:

$$I_j = \{i | \tfrac{y_{ij}}{y_{iq}} = \min_{r \in I_{j-1}} \tfrac{y_{rj}}{y_{rq}}\} \text{ amb } j \le m$$

És segur que per a algun $j \le m$, I_j tindrà un únic element, que serà el subíndex de la variable que sortirà de la base. Així s'evitarà repetir bases i, per tant, la formació de cicles. No es farà una demostració formal d'aquest fet, però se'n pot trobar una a la referència [3], pàgina 172.

4.6 Taula

A continuació s'explicarà com es procedeix a resoldre un problema de programació lineal amb el mètode símplex tenint en compte el que s'ha explicat fins ara.

Suposem que es té un problema en forma estàndard per al qual s'ha trobat una solució bàsica (per exemple utilitzant variables artificials). El que es farà és col·locar en forma de taula les columnes $a_1, ..., a_n$, juntament amb la columna b. Sota cada columna es col·locarà els coeficients c_i^T i sota la columna b s'hi posarà 0.

Es farà reducció gaussiana, incloent també l'última fila, de tal manera que els vectors columna de A associats a les variables bàsiques siguin tot zeros excepte un 1. D'aquesta manera, a l'última fila quedaran els coeficients de cost relatiu (que són zero en el cas de les variables bàsiques) i $-z_0$ sota la columna b. Així s'obtindrà una taula de l'estil:

$$\begin{array}{cccccccc|c}
1 & 0 & ... & 0 & y_{1,m+1} & y_{1,m+2} & \cdots & y_{1,n} & y_{1,0} \\
0 & 1 & ... & 0 & y_{2,m+1} & y_{2,m+2} & \cdots & y_{2,n} & y_{2,0} \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
0 & 0 & ... & 1 & y_{m,m+1} & y_{m,m+2} & \cdots & y_{m,n} & y_{m,0} \\
\hline
0 & 0 & ... & 0 & r_{m+1} & r_{m+2} & \cdots & r_n & -z_0
\end{array}$$

Una altra manera d'entendre això és a partir de la forma matricial del problema. Sigui B la submatriu $m \cdot m$ de A associada a la solució bàsica i D la submatriu $m \cdot (n-m)$ associada a la resta de variables. Es pot reformular el problema inicial definint:

- A=[B,D]
- $x = (x_B, x_D)$
- $c^T = (c_B^T, c_D^T)$

Així es tindrà el problema:

$$\begin{array}{rc}
\text{Minimitzar:} & c_B^T x_B + c_D^T x_D \\
\text{Subjecte a:} & Bx_B + Dx_D = b \\
 & x_B \geq 0 \\
 & x_D \geq 0
\end{array}$$

Aleshores, la solució bàsica corresponent a B seria $x = (B^{-1}b, 0)$. D'altra banda, a partir de la restricció es veu que:

$$x_B = B^{-1}b - B^{-1}Dx_D$$

Llavors, està clar que el valor de la funció a optimitzar en qualsevol cas serà:

$$z = c_B^T B^{-1}b + (c_D^T - c_B^T B^{-1}D)x_D$$

Llavors, per a la solució bàsica inicial, es tindria que:

$$z_0 = c_B^T B^{-1}b$$

Ja que s'ha vist que en aquest cas $x_D = 0$.

Aleshores, en crear la taula com s'ha explicat abans, quedaria:

A	b
c^T	0

Que és igual a:

B	D	b
c_B^T	c_D^T	0

Finalment, utilitzant la matriu B com a base, es pot passar a la forma canònica desitjada. El que es farà és multiplicar els blocs de dalt per B^{-1} i aleshores restar als blocs de baix els blocs de dalt multiplicats per $-c_B^T$. Així s'obté:

Id	$B^{-1}D$	$B^{-1}b$
0	$c_D^T - c_B^T B^{-1} D$	$-c_B^T B^{-1} b$

Aquesta taula està en la forma canònica: sota les columnes associades a les variables no bàsiques hi ha els coeficients de cost relatiu (ja que $r_D^T = c_D^T - c_B^T B^{-1} D$) i sota l'última columna tenim l'element $-z_0$ tal com es volia.

Aleshores s'aniran iterant els següents passos:

- Pas 1: Un cop formada la taula com s'ha explicat, si tots els coeficients de cost relatiu ($r_j = c_j - z_j$) són majors o iguals a zero es para; la solució factible bàsica és òptima.
- Pas 2: Si hi ha algun coeficient de cost relatiu estrictament negatiu, es tria q tal que r_q sigui el més negatiu dels coeficients de cost relatiu. La variable no bàsica x_q serà la que es convertirà en bàsica.
- Pas 3: Es calculen els quocients $\frac{y_{i0}}{y_{iq}}$ per a $y_{iq} > 0$ i $i = 1, 2, ..., m$. Si cap y_{iq} és positiu, el problema no està fitat. En cas contrari, es tria p tal que $\frac{y_{p0}}{y_{pq}}$ és el mínim de tots els quocients.
- Pas 4: Es pivota sobre el pq-èssim element, actualitzant tota la taula i es torna al pas 1.

A continuació, per a facilitar la comprensió, s'exposen dos exemples en què s'aplica el mètode símplex. En tots dos s'utilitza un programa *Símplex* inclòs en l'annex A. Aquest programa utilitza el mètode símplex original, triant la variable a incorporar a la base segons el coeficient de cost relatiu més negatiu, i requereix abans haver passat la taula a la forma canònica.

Exemple 4.5. Per a il·lustrar la selecció de les variables a treure i a incorporar a la base en cada pas del mètode símplex es proposa el següent exemple. Sigui el problema:

$$\begin{array}{rc} \text{Maximitzar:} & 3x_1 + x_2 + 3x_3 \\ \text{Subjecte a:} & 2x_1 + x_2 + x_3 \leq 2 \\ & x_1 + 2x_2 + 3x_3 \leq 5 \\ & 2x_1 + 2x_2 + x_3 \leq 6 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \end{array}$$

Aleshores, el primer que es faria seria introduir tres variables de separació x_4, x_5 i x_6, juntament amb les restriccions $x_4 \geq 0$, $x_5 \geq 0$ i $x_6 \geq 0$, per a transformar les desigualtats en igualtats. Tot seguit, es multiplicaria la funció a optimitzar per -1. Així el problema passaria de ser de maximització a ser de minimització. Cal destacar que en aquest cas la solució bàsica de la qual es podria partir és $(0, 0, 0, 1, 1, 1)^T$, és a dir, incloent a la solució bàsica només les variables de separació. Com que amb aquesta solució bàsica el valor de la funció a minimitzar seria zero, la taula quedaria com:

$$\begin{array}{c|cccccc|c} x_4 & 2 & 1 & 1 & 1 & 0 & 0 & 2 \\ x_5 & 1 & 2 & 3 & 0 & 1 & 0 & 5 \\ x_6 & 2 & 2 & 1 & 0 & 0 & 1 & 6 \\ \hline & -3 & -1 & -3 & 0 & 0 & 0 & 0 \end{array}$$

Cal destacar que per a facilitar la comprensió, s'ha afegit una columna extra al principi de la taula on es marquen quines són les variables bàsiques.

Mirant els coeficients de cost relatiu, es decidiria incorporar la variable amb el coeficient més negatiu. En aquest cas, podrien ser x_1 o x_3. És indiferent si se'n tria una o l'altra, així que es tria la primera. Aleshores, per als elements de la primera columna, es calcularien els quocients $y_{i,0}/y_{i,1}$. En aquest cas, com que $y_{1,1} > 0$, $y_{2,1} > 0$ i $y_{3,1} > 0$, es calculen tres quocients, que serien: $2/2 = 1$, $5/1 = 5$ i $6/3 = 3$. Com que 1 és el coeficient més petit d'aquests, es pivota respecte del primer element de la primera columna. En fer el primer pas del mètode símplex, la taula quedaria com:

$$\begin{array}{c|cccccc|c} x_1 & 1 & 0.5 & 0.5 & 0.5 & 0 & 0 & 1 \\ x_5 & 0 & 1.5 & 2.5 & -0.5 & 1 & 0 & 4 \\ x_6 & 0 & 1 & 0 & -1 & 0 & 1 & 4 \\ \hline & 0 & 0.5 & -1.5 & 1.5 & 0 & 0 & 3 \end{array}$$

Aleshores, l'únic coeficient de cost relatiu negatiu que hi ha és -1.5, cosa que indica que s'haurà de pivotar per a un element de la tercera columna. En aquest cas, com que $y_{3,3} = 0$, només es calcularien dos quocients: $1/0.5 = 2$ i $4/2.5 = 1.6$.

Està clar que s'haurà de pivotar respecte del segon element de la tercera columna, que és el que té el quocient associat més petit sense ser negatiu. Aleshores, la taula que queda en fer el segon pas de mètode símplex és:

x_1	1	0.2	0	0.6	-0.2	0	0.2
x_3	0	0.6	1	-0.2	0.4	0	1.6
x_6	0	1	0	-1	0	1	4
	0	1.4	0	1.2	0.6	0	5.4

Aleshores, com que tots els coeficients de cost relatiu són positius, el mètode símplex ha acabat. La solució bàsica òptima és $(0.2, 0, 1.6, 0, 0, 4)^T$ i el valor que assolirà la funció objectiu per a aquesta solució és 5.4.

Exemple 4.6. Per a il·lustrar l'ús de variables artificials, es proposa el següent exemple. Si es té el problema:

$$\begin{array}{rc} \text{Minimitzar:} & 4x_1 + x_2 + x_3 \\ \text{Subjecte a:} & 2x_1 + x_2 + 2x_3 = 4 \\ & 3x_1 + 3x_2 + x_3 = 3 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \end{array}$$

No hi ha una solució factible bàsica aparent, per la qual cosa s'haurà d'utilitzar el mètode de les dues fases de les variables artificials. S'introdueixen així les variables artificials $x_4 \geq 0$ i $x_5 \geq 0$. Es canvia el problema a:

$$\begin{array}{rc} \text{Minimitzar:} & x_4 + x_5 \\ \text{Subjecte a:} & 2x_1 + x_2 + 2x_3 + x_4 = 4 \\ & 3x_1 + 3x_2 + x_3 + x_5 = 3 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \\ & x_4 \geq 0 \\ & x_5 \geq 0 \end{array}$$

Aleshores, quedaria la taula:

2	1	2	1	0	4
3	3	1	0	1	3
0	0	0	1	1	0

On l'última fila són els coeficients de la funció a optimitzar. En passar aquesta taula a la forma canònica, quedaria:

x_4	2	1	2	1	0	4
x_5	3	3	1	0	1	3
	-5	-4	-3	0	0	-7

On l'última fila són els coeficients r^T juntament amb el valor negatiu z_0 que pren la funció objectiu amb aquesta solució bàsica. En aplicar el mètode símplex en aquest cas, s'obtindria en la primera iteració la taula:

x_4	0	-1	4/3	1	-2/3	2
x_1	1	1	1/3	0	1/3	1
	0	1	-4/3	0	5/3	-2

En fer una altra iteració s'obtindria la taula final de la primera fase:

x_3	0	-3/4	1	3/4	-1/2	3/2
x_1	1	5/4	0	-1/4	1/2	1/2
	0	0	0	1	1	0

Aleshores, s'haurien canviat les variables artificials de la solució bàsica de la qual es partia per variables no artificials i s'hauria aconseguit que la funció a minimitzar en aquest cas prengués valor zero. Així, es tindria una solució bàsica factible $x_1 = 3/2$, $x_2 = 0$ i $x_3 = 1/2$ per al problema original.

Llavors la taula pel al problema original seria:

0	-3/4	1	3/2
1	5/4	0	1/2
4	1	1	0

On a l'última fila s'han posat els coeficients de la funció objectiu original i un zero al final. En passar aquesta taula a la forma canònica queda:

x_3	0	-3/4	1	3/2
x_1	1	5/4	0	1/2
	0	-13/4	0	-7/2

En aplicar el mètode símplex a aquesta taula es farà una única iteració i s'obtindrà:

x_3	3/5	0	1	9/5
x_2	4/5	1	0	2/5
	13/5	0	0	-11/5

Per tant, la solució òptima serà: $x_1 = 0$, $x_2 = 2/5$ i $x_3 = 9/5$. En aquest cas, la funció objectiu prendria el valor 11/5.

4.7 Interpretació geomètrica del mètode simplex

Com s'ha explicat, el mètode símplex va passant d'un punt extrem a un altre, sempre que es vagi reduint el valor de la funció objectiu. Pel teorema 3.11., se sap que les solucions factibles es trobaran en un dels punts extrems. En concret, la solució òptima estarà en un punt extrem. Per tant, si s'arriba a un punt extrem pel qual no es pot passar a cap altre on el valor de la funció objectiu disminueixi (o augmenti en el cas de maximització), aquest serà el punt equivalent a la solució optima. Per a il·lustrar això es recupera l'exemple 3.16., que s'ha resolt amb el programa *Símplex* inclòs en l'annex. Cal destacar, que en aquest cas es vol maximitzar la funció objectiu. Per tant, el valor d'aquesta funció en els punts extrems (vèrtexs en aquest cas) anirà creixent.

Exemple 4.7. Si es té el problema:

$$\begin{array}{rc} \text{Maximitzar:} & 3x_1 + 2x_2 + 5x_3 \\ \text{Subjecte a:} & 2x_1 + x_2 \leq 4 \\ & x_3 \leq 5 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \end{array}$$

El primer que es fa en aquest cas és introduir les variables de separació x_4 i x_5 amb les quals el problema passa a ser:

$$\begin{array}{rc} \text{Maximitzar:} & 3x_1 + 2x_2 + 5x_3 \\ \text{Subjecte a:} & 2x_1 + x_2 + x_4 = 4 \\ & x_3 + x_5 = 5 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \\ & x_3 \geq 0 \\ & x_4 \geq 0 \\ & x_5 \geq 0 \end{array}$$

Aleshores, la solució bàsica inicial amb la qual es començaria el mètode símplex seria $(0,0,0,4,5)^T$, per la qual la funció objectiu prendria valor 0. Aquesta solució bàsica correspondria al vèrtex $(0,0,0)$ i la taula associada seria:

x_4	2	1	0	1	0	4
x_5	0	0	1	0	1	5
	-3	-2	-5	0	0	0

Des d'aquest vèrtex es podria passar al $(0,0,5)$, al $(2,0,0)$ o al $(0,4,0)$. Fent una iteració del mètode símplex, la taula resultant seria:

x_4	2	1	0	1	0	4
x_3	0	0	1	0	1	5
	-3	-2	0	0	5	25

Per tant, el resultat seria passar al vèrtex $(0,0,5)$, associat a la solució bàsica $(0,0,5,4,0)^T$ i valor de la funció objectiu 25. Des d'aquest vèrtex es pot passar al $(0,0,0)$, al $(2,0,5)$ o al $(0,4,5)$. Fent una altra iteració del mètode símplex, s'obtindria la taula:

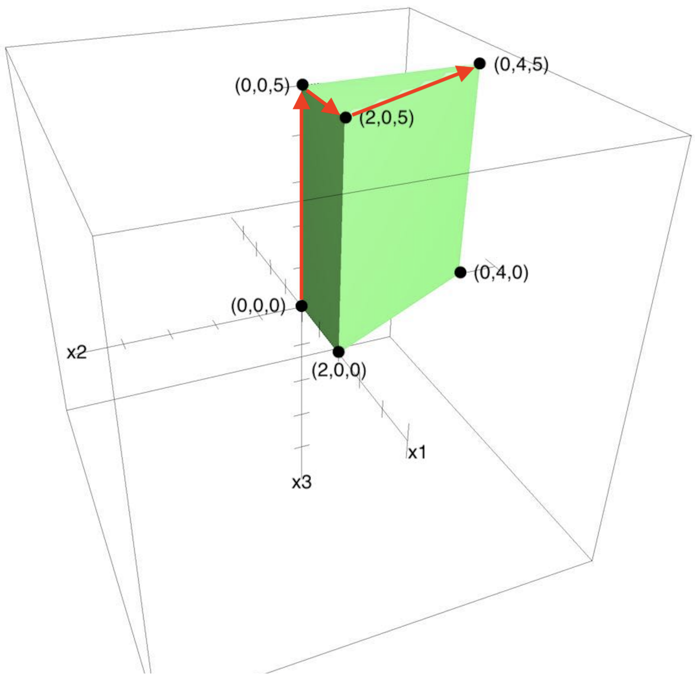
x_1	1	0.5	0	0.5	0	2
x_3	0	0	1	0	1	5
	0	-0.5	0	1.5	5	31

Per tant, el resultat seria passar al vèrtex $(2,0,5)$, associat a la solució bàsica $(2,0,5,0,0)^T$ i valor de la funció objectiu 31. Finalment, des d'aquest vèrtex es pot passar al $(0,0,5)$, al $(0,4,5)$ o al $(2,0,0)$. Fent l'última iteració del mètode símplex s'obtindria la taula:

x_2	2	1	0	1	0	4
x_3	0	0	1	0	1	5
	1	0	0	2	5	33

Per tant, el resultat seria passar al vèrtex $(0,4,5)$, associat a la solució bàsica $(0,4,5,0,0)^T$ i valor de la funció objectiu 33. Com que tots els coeficients de cost relatiu són positius, es conclou que s'ha arribat a la solució òptima. El recorregut resultant de les iteracions que s'han fet es pot veure en la figura 4.8.

Figura 4.8.



5 Dualitat

En aquest capítol les principals fonts consultades han set [3], [6], [8] i [10].

Tot problema de programació lineal, o problema primal, té associat un problema dual. Els dos problemes es construeixen a partir de les mateixes restriccions i coeficients de cost relatiu, però si un és de minimització, l'altre és de maximització.

Es poden interpretar les variables del problema dual com a preus associats a les restriccions del problema primal.

Cal destacar que donats un problema primal i el seu dual, el problema dual del problema dual original serà el primal original.

5.1 Programes lineals duals

Donat un problema primal en forma canònica, el problema dual associat que determina la forma canònica de la dualitat serà:

$$\begin{array}{rc||rc}
 & \text{Primal} & & \text{Dual} \\
\text{Minimitzar} & c^T x & \text{Maximitzar} & \lambda^T b \\
\text{Subjecte a} & Ax \geq b & \text{Subjecte a} & \lambda \geq 0 \\
 & x \geq 0 & & \lambda^T A \leq c^T
\end{array}$$

Si A és una matriu $m \cdot n$, x és un vector n-dimensional, b és un vector m-dimensional i c^T és un vector fila n-dimensional, llavors λ^T serà un vector fila m-dimensional. El vector x serà el de les variables del problema primal i el vector λ^T serà el de les variables del problema dual. Cal destacar que existeix una variable dual per cada restricció primal.

D'altra banda, donat un problema en forma estàndard, la forma estàndard de la dualitat serà:

$$\begin{array}{rc||rc}
 & \text{Primal} & & \text{Dual} \\
\text{Minimitzar} & c^T x & \text{Maximitzar} & \lambda^T b \\
\text{Subjecte a} & Ax = b & \text{Subjecte a} & \lambda^T A \leq c^T \\
 & x \geq 0 & &
\end{array}$$

Cal destacar que no sempre es partirà d'un problema en forma estàndard o en forma canònica, però ja s'ha explicat en l'apartat 2 com modificar els problemes perquè siguin d'aquestes formes.

5.2 Relacions primal-dual

Hi ha una relació entre les variables i les restriccions dels problemes primals i duals. Sigui quin sigui el problema de maximització i el de minimització:

- Una variable en el problema de minimització està restringida a ser ≥ 0 si i només si la restricció corresponent en el problema de maximització és una desigualtat $\leq$.
- Una variable en el problema de minimització està restringida a ser ≤ 0 si i només si la restricció corresponent en el problema de maximització és una desigualtat $\geq$.
- Una variable en el problema de minimització no està restringida si i només si la restricció corresponent en el problema de mazimització és una igualtat.
- Una restricció en el problema de minimització és una desigualtat de la forma $\geq$ si i només si la variable corresponent en el problema de maximització està restringida a ser ≥ 0.
- Una restricció en el problema de minimització és una desigualtat de la forma $\leq$ si i només si la variable corresponent en el problema de maximització està restringida a ser ≤ 0.
- Una restricció en el problema de minimització és una igualtat si i només si la variable corresponent en el problema de maximització no està restringida.

Per a entendre millor aquestes relacions es proposa el següent exemple:

Exemple 5.1. Donat el problema primal:

$$\begin{array}{rc} \text{Maximitzar:} & 8x_1 + 3x_2 \\ \text{Subjecte a:} & x_1 - 6x_2 \geq 2 \\ & 5x_1 + 7x_2 = -4 \\ & x_1 \leq 0 \\ & x_2 \geq 0 \end{array}$$

Aleshores, com que en el problema primal es té que la primera restricció és $\geq$, en el dual $\lambda_1 \leq 0$. Com que la segona restricció del primal és una igualtat, està clar que λ_2 no estarà restringida en el dual.

Llavors, com que la primera variable del primal compleix $x_1 \leq 0$, la primera restricció del dual serà una desigualtat $\leq$. Igualment, com que la segona variable del primal compleix $x_2 \geq 0$, la segona restricció del dual serà una desigualtat $\geq$.

Finalment, els termes independents del primal $(2, -4)$ seran els coeficients de la funció objectiu del dual. D'altra banda, els coeficients de la funció objectiu primal $(8, 3)$ seran els termes independents del problema dual. Llavors, els coeficients de les restriccions del primal determinen els coeficients de les restriccions del dual, tenint en compte que en el primal les restriccions venien determinades per Ax i en el dual vindran determinades per $\lambda^T A$.

Aleshores, el problema dual serà:

$$
\begin{array}{rc}
\text{Minimitzar:} & 2\lambda_1 - 4\lambda_2 \\
\text{Subjecte a:} & \lambda_1 + 5\lambda_2 \leq 8 \\
 & -6\lambda_1 + 7\lambda_2 \geq 3 \\
 & \lambda_1 \leq 0 \\
 & \lambda_2 \text{ no restringit}
\end{array}
$$

També hi ha una relació entre els valors òptims de les funcions objectiu del problema primal i del dual.

Lema 5.2. *Dualitat dèbil: Si x i λ són solucions factibles per als problemes primal i dual respectivament (en forma estàndard, sent el primal el de minimització), llavors $c^T x \geq \lambda^T b$.*

Demostració:

Com que $\lambda^T b = \lambda^T Ax$, si $x \geq 0$ i $\lambda^T A \leq c^T$ llavors es té que $\lambda^T Ax \leq c^T x$. □

Per tant, el valor objectiu d'una solució factible en el problema de minimització és sempre major o igual que el valor objectiu d'una solució factible en el problema de maximització. D'això se'n deriven dos corol·laris i un lema.

Corołari 5.3. *Si x_0 i λ_0 són solucions factibles dels problemes primal i dual respectivament i es compleix que $cx_0 = \lambda_0 b$, llavors són solucions òptimes dels seus respectius problemes (Aquest corol·lari es coneix també com teorema de la dualitat forta, tal com s'explica a la referència [7]).*

Corołari 5.4. *Si un dels dos problemes té un valor objectiu no fitat, aleshores l'altre problema no té cap solució factible.*

Cal destacar que el recíproc del corol·lari 5.4. no és cert: que un problema no sigui factible no implica que l'altre no estigui fitat.

Lema 5.5. *Si un dels dos problemes té una solució óptima, aleshores els dos problemes tindran solució óptima i els dos valors objectius óptims seran iguals.*

Aleshores, es pot resumir tot això en el següent teorema.

Teorema 5.6. *Dualitat de la programació lineal:*

Si el problema primal o dual té una solució òptima finita, l'altre problema també la tindrà. A més, els valors corresponents de les funcions objectius són iguals. Si un dels dos problemes no té fitat el seu valor objectiu, l'altre no tindrà una solució factible.

Demostració: Es pot trobar una demostració formal d'aquest teorema a la referència [8], pàgina 88. □

6 Problema del transport

En aquest capítol les principals fonts consultades han set [2], [3], [8], [10] i [11].

Una de les aplicacions de la programació lineal és el problema del transport. En aquest problema, se suposa que hi ha m orígens amb unes quantitats no negatives d'un producte a_i amb $i = 1, ..., m$ i n destins amb unes demandes no negatives d'aquest producte b_j amb $j = 1, ..., n$. Aquestes quantitats poden ser nombres reals, però per a facilitar la comprensió del que s'explicarà, els exemples proposats en aquest capítol utilitzen nombres enters. Se suposa, a més, que la suma de la quantitat demandada total és igual a la de la quantitat oferida total. És a dir:

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

També se suposa que hi ha un cost associat a enviar una unitat del producte des de l'origen a_i al destí b_j, que serà c_{ij} amb $i = 1, ..., m$ i $j = 1, ..., n$. Llavors, el problema del transport serà:

$$\begin{array}{ll} \text{Minimitzar} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{Subjecte a} & \displaystyle\sum_{j=1}^{n} x_{ij} = a_i \text{ amb } i = 1, ..., m \\ & \displaystyle\sum_{i=1}^{m} x_{ij} = b_j \text{ amb } j = 1, ..., n \\ & x_{ij} \geq 0 \text{ per a tot i,j} \end{array}$$

Cal destacar que si la demanda total és inferior a l'oferta total, es pot suposar l'existència d'un nou destí la demanda del qual és la diferència entre l'oferta total i la demanda total i pel que tots els costs unitaris seran zero. Això serà equivalent a que alguns orígens es quedin amb alguna quantitat del producte sense repartir. Fent això, es pot solucionar el problema del transport com segueix.

Les variables x_{ij} representen les quantitats enviades des de l'origen a_i al destí b_j. Les restriccions també es poden escriure com:

$$\begin{array}{ccc} x_{11} + x_{12} + ... + x_{1n} & = & a_1 \\ x_{21} + x_{22} + ... + x_{2n} & = & a_2 \\ \vdots & \vdots & \vdots \\ x_{m1} + x_{m2} + ... + x_{mn} & = & a_m \\ x_{11} + x_{21} + ... + x_{m1} & = & b_1 \\ x_{12} + x_{22} + ... + x_{m2} & = & b_2 \\ \vdots & \vdots & \vdots \\ x_{1n} + x_{2n} + ... + x_{mn} & = & b_n \end{array}$$

Es pot veure que el problema del transport és factible i sempre té una solució òptima. Sigui:

$$S = \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

És evident que si es defineix $x_{ij} = \frac{a_i b_j}{S}$ per $i = 1, ..., m$ i $j = 1, ..., n$, llavors aquests x_{ij} compleixen les restriccions del problema, per tant, es tracta d'una solució factible. A més, el valor que pot prendre la variable x_{ij} està fitat:

$$0 \leq x_{ij} \leq min\{a_i, b_j\}$$

Per tant, el problema del tranport sempre tindrà una solució òptima, ja que tot programa lineal fitat que té una solució factible té una solució òptima.

D'altra banda, tal com es demostra a la pàgina 344 de la referencia [3], es pot observar que el conjunt de restriccions és redundant. Per tant, es pot eliminar una de les restriccions sense canviar el conjunt de les solucions factibles. Aquesta restricció eliminada es podrà reconstruir a partir de la resta. Per tant, el problema es pot considerar de $m + n - 1$ restriccions i una base per a aquest problema tindrà $m + n - 1$ vectors.

6.1 Solució factible bàsica inicial

Igual que amb el mètode símplex, per a trobar la solució d'un problema del transport cal trobar una solució bàsica factible inicial.

Alguns mètodes per a fer això són el mètode de Vogel o el del cost mínim. Un altre mètode, el qual es presenta a continuació, és el de la regla de la cantonada nord-est. Per a aquest mètode s'utilitza una taula com la següent:

x_{11}	x_{12}	x_{13}	...	x_{1n}	a_1
x_{21}	x_{22}	x_{23}	...	x_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
x_{m1}	x_{m2}	x_{m3}	...	x_{mn}	a_m
b_1	b_2	b_3	...	b_n	

Aleshores, una cel·la buida indica que la variable corresponent pren el valor zero. Començant amb totes les cel·les buides se segueix recursivament el següent procediment:

- Pas 1: Es comença a la cel·la de la cantonada superior esquerra.
- Pas 2: Es col·loca el mínim entre el que falta per complir del requeriment de la columna i el que falta per complir del requeriment de la fila.
- Pas 3: Si es compleix totalment el requeriment de la fila, es passa a la cel·la de sota. Si es compleix totalment el requeriment de la columna, es passa a la cel·la de la dreta. Si es compleixen tots els requeriments, es para. Si no, es torna al pas 2.

Per a entendre millor aquest procediment s'explica pas a pas amb el següent exemple.

Exemple 6.1. Es consideren els vectors $a = (30, 80, 10, 60)$ i $b = (10, 50, 20, 80, 20)$. Aleshores, la taula que s'obtindria seria:

10	20				30
	30	20	30		80
			10		10
			40	20	60
10	50	20	80	20	

El que s'ha fet és, partint de la cel·la superior esquerra, col·locar-hi el mínim entre el requeriment de la fila (30) i el de la columna (10). Com que el mínim requeriment és el de la columna, es posa 10 en aquesta cel·la i es passa a la de la dreta.

Aleshores, en aquesta es posa el mínim per a complir el requeriment de la fila (30-10=20) i el de la columna (50). S'hi posa 20, i com que s'ha complert el requeriment de la fila es passa a la cel·la de sota.

Se segueix aquest procediment fins a arribar al resultat final. Aleshores, la solució bàsica inicial serà:

$$\begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\ x_{41} & x_{42} & x_{43} & x_{44} & x_{45} \end{pmatrix} = \begin{pmatrix} 10 & 20 & 0 & 0 & 0 \\ 0 & 30 & 20 & 30 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 40 & 20 \end{pmatrix}$$

6.2 Propietats estructurals del problema del transport

Una propietat clau del problema del transport, la qual és útil per a trobar-ne la solució, és la que s'exposa en el següent teorema.

Definició 6.2. *Una base és triangular si la matriu bàsica corresponent es pot representar com una matriu triangular inferior a través d'una permutació de les seves files i columnes.*

Teorema 6.3. *Triangularitat de les bases*

Totes les bases del problema del transport són triangulars.

Demostració:

Considerant les restriccions:

$$
\begin{array}{rcl}
x_{11} + x_{12} + ... + x_{1n} & = & a_1 \\
x_{21} + x_{22} + ... + x_{2n} & = & a_2 \\
\vdots & \vdots & \vdots \\
x_{m1} + x_{m2} + ... + x_{mn} & = & a_m \\
x_{11} + x_{21} + ... + x_{m1} & = & b_1 \\
x_{12} + x_{22} + ... + x_{m2} & = & b_2 \\
\vdots & \vdots & \vdots \\
x_{1n} + x_{2n} + ... + x_{mn} & = & b_n
\end{array}
$$

Aleshores, es canvia el signe de la meitat superior del sistema. La matriu dels coeficients del sistema constarà només de 0, +1 i -1. Per la redundància, es pot eliminar una de les restriccions. Aleshores, a partir de la matriu de coeficients resultant, es forma una base B triant un subconjunt no singular de m+n-1 columnes.

Cada columna de B té com a molt dos coeficients diferents de zero, que poden ser +1 o -1. Per tant, hi ha com a molt $2(m+n-1)$ coeficients diferents a zero en B. Però si totes les columnes tinguessin dos coeficients diferents de zero, la suma de totes les files seria zero, contradient la no singularitat de B. Per tant, almenys una columna tindrà només un coeficient diferent de zero. Per tant, el nombre total de coeficients diferents de zero de B serà menor a $2(m+n-1)$, per tant, hi haurà una fila de B que només té un coeficient diferent de zero. És per això que es pot dur a terme el següent procediment per a demostrar que la matriu és triangular:

- Pas 1: Trobar una fila de B amb només un coeficient diferent de zero.
- Pas 2: Formar una submatriu traient de la que es té la fila triada al pas 1 i la columna corresponent a l'únic coeficient diferent de zero. Tornar al pas 1.

Com que es podrà seguir aquest procediment fins a haver eliminat totes les files i columnes, queda demostrat que B és una matriu triangular. $\square$

Com ha quedat demostrat, tota base del problema del transport és triangular i tots els elements diferents de zero seran +1 (o -1 en cas de fer algun canvi de signe). Aleshores, d'això se'n deriva el següent corol·lari:

Corołari 6.4. *Si les sumes de fila i de columna d'un problema de transport són enteres, aleshores les variables bàsiques de qualsevol solució bàsica són enteres.*

6.3 Algorisme per al problema del transport

Gràcies a la triangularitat de les bases abans explicada, és fàcil aplicar el mètode símplex al problema del transport. Per a fer-ho, s'utilitzaran el que es coneixen com a multiplicadors símplexs, que formaran el vector $\lambda = (u_1, ..., u_m, v_1, ..., v_n)$.

El que es farà és considerar una matriu C de dimensió $m \cdot n$, formada pels coeficients de cost unitaris c_{ij}. Llavors, s'aniran afegint, a mesura que es vagin trobant, els multiplicadors símplexs u_i per $i = 1, ..., m$ a la dreta de la matriu C i els multiplicadors símplexs v_j per $j = 1, ..., n$ a sota. Com que una de les restriccions és redundant, es pot assignar un valor arbitrari a un multiplicador qualsevol.

Per a trobar els multiplicadors símplexs, s'utilitzarà que per a les variables x_{ij} que són bàsiques aquests multiplicadors compliran que $u_i + v_j = c_{ij}$. D'altra banda, es complirà que els coeficients de cost relatiu seran $r_{ij} = c_{ij} - u_i - v_j$ per a $i = 1, 2, ..., m$ i $j = 1, 2, ..., n$.

Al final, s'obtindrà una taula com la següent:

$c_1 1$	$\dots$	$c_1 n$	u_1
$\vdots$		$\vdots$	$\vdots$
$c_m 1$	$\dots$	$c_m n$	u_m
v_1	$\dots$	v_n	

Corol·lari 6.5. *Si els costs unitaris c_{ij} d'un problema de transport són tots enters, llavors (suposant que el valor arbitrari que s'assigna a un multiplicador símplex també és enter) els multiplicadors símplex associats a qualsevol base també són enters.*

Aleshores, per a trobar aquests multiplicadors, se segueix el següent algorisme:

- Pas 1: S'assigna un valor arbitrari a un multiplicador. Per exemple, el valor zero a v_n.
- Pas 2: Es busquen els elements c_{ij} corresponents a les variables x_{ij} que són bàsiques pels que u_i o v_j (però no tots dos) ja està determinat.
- Pas 3: Per als elements del pas 2, es calcula el multiplicador no determinat utilitzant que $c_{ij} = u_i + v_j$. Si tots els multiplicadors estan determinats, es para. En cas contrari, es torna al pas 2.

La triangularitat de les bases garanteix que es pot utilitzar aquest algorisme fins a trobar tots els multiplicadors símplexs.

Exemple 6.6. A l'exemple 6.1., es proposaven els vectors $a = (30, 80, 10, 60)$ i $b = (10, 50, 20, 80, 20)$. Aleshores, la taula que s'obtindria seria:

10	20				30
	30	20	30		80
			10		10
			40	20	60
10	50	20	80	20	

Si se suposa ara que els costs unitaris són:

3	4	6	8	9
2	2	4	5	5
2	2	2	3	2
3	3	2	4	2

Aleshores, s'afegeix una fila a sota corresponent al vector $v = (v_1, v_2, v_3, v_4, v_5)$ i una columna a la dreta corresponent al vector $u = (u_1, u_2, u_3, u_4)$. Si s'assigna el valor zero a v_5, es busca quins elements de la columna 5 corresponen a una variable bàsica. En aquest cas, només serà c_{45}. Per tant, com que $c_{45} = u_4 + v_5$ i com que $c_{45} = 2$ i $v_5 = 0$, està clar que $u_4 = 2$. Aleshores, es busca quins altres elements de la fila 4 corresponen a una variable bàsica. En aquest cas, a part de c_{45}, només serà c_{44}. Com que $c_{44} = u_4 + v_4$ i com $c_{44} = 4$ i $u_4 = 2$, està clar que $v_4 = 2$. Seguint l'algorisme s'arriba a:

3	4	6	8	9	5
2	2	4	5	5	3
2	2	2	3	2	1
3	3	2	4	2	2
-2	-1	1	2	0	

Aleshores, els multiplicadors símplexs són: $u = (5, 3, 1, 2)$ i $v = (-2, -1, 1, 2, 0)$.

Un cop se sap com calcular els multiplicadors simplexs, se sabrà calcular els coeficients de cost relatiu. Com passa amb el mètode símplex general, una variable no bàsica serà candidata a entrar a la base sempre que el seu coeficient de cost relatiu sigui negatiu. A mesura que es va incrementant el valor de la variable que s'afegeix a la base, alguna de les variables bàsiques anteriors disminuirà tendint a zero.

Per la propietat triangular de les bases en aquest problema es pot comprovar que en introduir una variable nova a la base amb valor θ, el valor de les variables bàsiques anteriors es queda igual, augmenta en $+\theta$ o disminuiex en $-\theta$.

En la taula de les variables x_{ij}, totes les variables bàsiques estaran connectades. Això vol dir que agafant les cel·les de les variables bàsiques com a vèrtexs i unint aquests vèrtexs amb arestes horitzontals i verticals (unint només els vèrtexs més pròxims que estan en la mateixa fila o columna), es podrà passar de qualsevol vèrtex a un altre. Aquests vèrtexs i arestes definiran un arbre sense cicles. Si l'arbre contingués un cicle, es podria fer una combinació lineal dels vectors de la base que fos igual a zero. Per tant, els vectors de la base no serien linealment independets i, per tant, no seria una base.

Com s'ha explicat, una solució bàsica del problema del transport tindrà exactament $m + n - 1$ variables diferents a zero. Llavors, en introduir una nova variable a la solució bàsica, el conjunt de les $m + n$ cel·les no buides formarà un cicle, tal com s'explica a la referència [9].

Aleshores, un cop triada la variable no bàsica que s'introduirà a la base, es determina quin cicle es formaria en aquest cas. Llavors, com que el valor d'aquesta variable que s'introdueix a la base passa de zero a θ, s'escriu + a la cel·la corresponent que abans estava en blanc. Per a les cel·les de les variables bàsiques que formen part del cicle que comença i acaba en la cel·la on ara hi ha un +, es va posant alternada-ment - i +. Totes les altres variables bàsiques no es modificaran. Finalment, θ serà el mínim valor corresponent a una de les cel·les amb un -, cosa que farà que almenys una variable bàsica passi a ser zero i, per tant, abandoni la base. Llavors, se suma θ a les cel·les amb un + i es resta a les que tenen un -. Aquest mètode es diu *stepping-stone* i permet trobar noves solucions bàsiques factibles a partir d'una altra prèviament coneguda.

Per a trobar aquest cicle, s'ha de tenir en compte que a cada fila i columna, hi haurà exactament un $+\theta$ i un $-\theta$ o no hi haurà res. Aleshores, en posar + a la cel·la corresponent a la variable que entra a la base, se sap que en aquella fila i columna hi haurà un $-\theta$. A més, si hi ha alguna variable bàsica que és l'única que queda sense assignar a la seva fila o columna, se sap que no canviarà.

Exemple 6.7. Aquest exemple s'ha resolt amb el programa *Transport* de l'annex. Tornant als exemples 6.1. i 6.6., s'havia vist que els multiplicadors símplexs eren:

3	4	6	8	9	5
2	2	4	5	5	3
2	2	2	3	2	1
3	3	2	4	2	2
-2	-1	1	2	0	

Per tant, calculant els coeficients de cost relatiu, s'obté:

0	0	0	1	4
1	0	0	0	2
3	2	0	0	1
3	2	-1	0	0

Per tant, l'única variable no bàsica candidata a entrar a la base és x_{43}, ja que serà l'única amb un coeficient de cost relatiu associat negatiu. Aquesta variable que entra a la base formaria un cicle amb les variables x_{44}, x_{24} i x_{23}. Llavors, s'assignen + i - de la següent manera:

10	20				30
	30	20^-	30^+		80
			10		10
		+	40^-	20	60
10	50	20	80	20	

Per a trobar aquest cicle, el que s'ha fet és posar + a la cel·la d'x_{43}. Aleshores, se sap que en la fila 4 i en la columna 3 hi haurà exactament un -. Com que a la columna 3 només queda la variable $x_{23} = 20$, se sap que a aquesta se li restarà θ.

Aleshores, es veu que la variable $x_{11} = 10$ es l'única de la columna 1, per tant, no canviarà. Com que a la fila 1 hi ha 2 variables, però una és $x_{11} = 10$, que no canvia, es dedueix que l'altra, $x_{12} = 20$, tampoc canviarà. Com que a la columna 2 hi ha dues variables i una és $x_{12} = 20$, que no canvia, es dedueix que l'altra, $x_{22} = 30$, tampoc canviarà. Aleshores, a la fila 2 hi ha una variable que no canvia, $x_{22} = 30$, i una a la que se li restrà θ, $x_{23} = 20$. Per tant, a l'altra variable d'aquesta fila, $x_{24} = 30$, que és l'última opció que queda, serà a la que se li sumarà θ. La variable $x_{34} = 10$ està sola a la fila 3, per tant, no canviarà. A la columna 4 hi ha una variable que no canvia, $x_{34} = 10$, i una a la que se li suma θ, $x_{24} = 30$. Per tant, l'altra variable de la columna 4, $x_{44} = 40$, serà a la que se li restarà θ. Finalment, com que a la fila 4 ja hi ha una variable a la que se li suma θ, x_{43}, i una altra a la que se li resta, $x_{44} = 40$, la variable $x_{45} = 20$ no canviarà.

Com que el mínim entre 20 i 40, corresponent a les variables bàsiques amb un -, és 20, queda clar que θ serà 20. Llavors, es suma i es resta θ segons correspongui i s'obté la nova solució bàsica:

10	20				30
	30		50		80
			10		10
		20	20	20	60
10	50	20	80	20	

Si es calculessin ara els nous multiplicadors símplexs i els nous coeficients de cost relatiu, s'observaria que tots els coeficients de cost relatiu nous serien no negatius. Per tant, s'hauria arribat a la solució òptima.

En definitiva l'algorisme per a resoldre el problema del transport serà:

- Pas 1: Es calcula una solució bàsica inicial utilitzant la regla de la cantonada nord-est o qualsevol altre mètode.
- Pas 2: Es calculen els multiplicadors símplexs i els coeficients de cost relatiu. Si tots els coeficients de cost relatiu són no negatius, la solució és òptima i es para. En cas contrari, es passa al pas 3.
- Pas 3: Es selecciona la variable no bàsica a introduir a la base, normalment la que està associada al coeficient de cost relatiu més negatiu. Es calcula θ i es canvien les variables bàsiques del cicle. Es torna al pas 2.

Finalment, cal destacar que en el cas que existeixi degeneració (alguna variable bàsica té valor zero), es pot seguir el mateix procediment. El problema en aquest cas és que l'arbre que formen les variables bàsiques deixarà de ser connex. Per a solucionar això, l'únic que caldrà fer és donar a la variable bàsica degenerada el valor ε que es considerarà sempre suficientment petit. Així, es podrà seguir l'algorisme explicat anteriorment i, si la variable bàsica degenerada deixa la base, es podrà eliminar ε.

Aquest problema de la degeneració apareixerà quan, per a un subconjunt de files i columnes, l'oferta total i la demanda total siguin iguals.

Problema de l'assignació

És interessant parlar d'aquest problema, que es pot interpretar com una variació del problema del transport.

En aquest cas, hi haurà tants orígens com destins. Una manera d'interpretar-ho és que s'han d'assignar n treballadors a n tasques. Cada treballador donarà un rendiment diferent en cada tasca. Per tant, es vol maximitzar el rendiment total dels treballadors, subjecte a que tots facin alguna tasca.

En aquest cas, les variables prendran només el valor 0 o 1. Hi haurà exactament n variables bàsiques amb valor 1. En un problema del transport normal, hi hauria $2n-1$ variables bàsiques diferents de zero. Per tant, aquest és un cas molt degenerat.

Aquest problema es podria resoldre amb l'algorisme abans proposat, però resultaria bastant pesat perquè s'hauria de tractar amb moltes variables degenerades. És per això que en aquest cas s'utilitza un altre mètode basat en la relació primal-dual abans introduïda.

6.4 Exemple

A continució es proposa un exemple del problema del transport. Per a resoldre'l, s'ha utilitzat un programa inclòs en l'annex B.

Exemple 6.8. Per a aquest exemple se suposa que hi ha 3 orígens i 4 destins. Les unitats de producte oferides des de cada origen són $a = (400, 700, 100)$ i les unitats de producte demandades des de cada destí són $b = (500, 400, 100, 200)$. És evident que l'oferta i la demanda són iguals a 1200 unitats. Aleshores, se suposa que els costs unitaris d'enviar una unitat des de l'origen a_i al destí b_j venen donats per la següent taula:

8	11	5	7
9	5	6	11
12	4	8	10

Aleshores, s'introdueix aquesta informació al programa *Transport* de l'annex i s'obté que la primera solució bàsica, trobada amb el mètode de la cantonada nord-oest, és:

400				400
100	400	100	100	700
			100	100
500	400	100	200	

És evident que aquesta solució bàsica compleix tant els requeriments de les files com els de les columnes.

Aleshores, en calcular els multiplicadors símplexs, s'obté:

8	11	5	7	10
9	5	6	11	11
12	4	8	10	10
-2	-6	-5	0	

Els multiplicadors símplexs seran l'última fila i l'última columna. Amb ells es podran calcular els coeficients de cost relatiu. L'únic coeficient de cost relatiu es trobarà en l'última posició de la primera fila i serà $7 - 10 - 0 = -3$. Això determina quina variable entrarà en la base. Aleshores, la matriu de les variables serà:

400^-			$+$	400
100^+	400	100	100^-	700
			100	100
500	400	100	200	

Com que θ serà el mínim dels elements als quals es resta θ, està clar que serà 100. Per tant, la nova solució bàsica serà:

300			100	400
200	400	100		700
			100	100
500	400	100	200	

Llavors, es tornen a calcular els multiplicadors símplexs:

8	11	5	7	7
9	5	6	11	8
12	4	8	10	10
1	-3	-2	0	

Es pot veure que l'únic coeficient de cost relatiu negatiu estaria a la segona posició de l'última fila i valdria $4 - 10 - (-3) = -3$. Aleshores, això determinaria quina variable entraria a la base. La matriu de les variables seria:

300^-			100^+	400
200^+	400^-	100		700
	$+$		100^-	100
500	400	100	200	

Un cop definit el cicle, θ serà el mínim dels elements als quals es resta θ. En aquest cas està clar que serà 100. Llavors, la nova solució bàsica serà:

200			200	400
300	300	100		700
	100			100
500	400	100	200	

Finalment, si es tornen a calcular els multiplicadors símplexs:

8	11	5	7	7
9	5	6	11	8
12	4	8	10	7
1	-3	-2	0	

Com que tots els coeficients de cost relatius són no negatius, s'ha arribat a la solució bàsica òptima. En aquest cas, el cost associat a satisfer totes les demandes dels destins és:

$$200 * 8 + 200 * 7 + 300 * 9 + 300 * 5 + 100 * 6 + 100 * 4 = 8200$$

Finalment, cal destacar que, com que el cost relatiu associat a la variable en la tercera posició de la primera fila és zero, existeix una solució bàsica òptima alternativa.

7 Conclusions

Amb aquest treball s'ha complert l'objectiu inicial, que era establir una base de coneixements necessaris per a entendre què és la programació lineal, com plantejar els problemes que se'n deriven i com resoldre aquests problemes amb el mètode sìmplex.

Amb la introducció que s'ha fet dels problemes de programació lineal, ha quedat clara la seva interpretació algebraica. D'altra banda, gràcies a les explicacions i demostracions dels conceptes de l'anàlisi convexa, ha quedat clara la seva interpretació geomètrica.

S'ha aprofundit en el mètode símplex ja que és la base de la programació lineal. Tot i que aquest mètode va ser ideat al 1947, la seva importància segueix estant vigent avui en dia. S'ha explicat l'algorisme que s'utilitza en el mètode símplex i s'ha implementat aquest algorisme en un programa que es pot trobar en l'annex.

També s'han introduït les nocions de dualitat necessaries per a entendre interpretacions equivalents d'un mateix problema.

D'altra banda, també s'ha complert l'altre objectiu del treball, que era aprofundir en una aplicació pràctica de la programació lineal, com és el problema del transport. S'ha explicat l'algorisme que s'utilitza per a resoldre aquest problema i s'ha implementat aquest algorisme en un programa que es pot trobar a l'annex.

7.1 Propers passos

Donada aquesta base de coneixements sobre aquesta branca de les matemàtiques aplicades, es podrà ampliar en diverses direccions.

D'una banda, amb els coneixements establerts, serà més fàcil entendre la programació no lineal, en què alguna de les restriccions o la funció objectiu no serà una funció lineal. També servirà com a base per a introduir-se en la programació entera, en què es restringiran algunes variables (o totes) a prendre valors enters.

D'altra banda, com ja s'ha comentat en el treball, existeixen altres mètodes a part del símplex per a resoldre problemes de programació lineal. Tot i que la importància del mètode símplex és evident, existeixen altres mètodes com el de l'el·lipsoide o el del punt interior que s'haurien d'estudiar si es desitja seguir aprofundint en aquest coneixement. Però també es podria aprofundir més en el mètode símplex, per exemple estudiant el mètode símplex revisat, que és mès òptim computacionalment.

Pel que fa a la dualitat, es podria aprofundir més en la interpretació econòmica dels problemes duals i també es podria estudiar la sensibilitat dels problemes de programació lineal: com varien els resultats en variar algun paràmetre.

Finalment, pel que fa al problema del transport, es podria aprofundir en altres mètodes per a trobar solucions inicials, com el mètode de Vogel o el del cost mínim. També es podrien estudiar altres aplicacions pràctiques de la programació lineal com el problema de la dieta o els problemes de la motxilla i de localització (que són problemes de programació entera).

Referències

[1] Avriel, M.: *Nonlinear programming analysis and methods*. Mineola, Nova York: Dover Publications Inc., 2003

[2] Bársov, A.S.: *Qué es la programación lineal*. Moscú, URSS: Mir, 1982

[3] Bazaraa, M.; Jarvis, J.; Hernandez Lerma, O.: *Programación lineal y flujo de redes*. Mexic: Limusa, 1981

[4] Borgwardt, K.H.: *The simplex method A probabilistic analysis*. Berlín, Heidelberg, Alemanya: Springer-Verlag, 1987

[5] Chvátal, V.: *Linear programming*. Nova York: W.H. Freeman and company, 1983

[6] García Solera, M.; Muñoz Vaquer, C.; Soldevilla Senat, C.: *Programación lineal Teoría y aplicaciones*. Universitat de Barcelona, Barcelona, Catalunya: Ediciones Gráficas Rey S.L., 1998

[7] Gass, S.I.: *Linear programming*. Estats Units: McGraw-Hill book comapny, 1969

[8] Luenberger, D.; López Mateos, M.: *Programación lineal y no lineal*. Willmington, Delaware: Addison-Wesley Iberoamericana, 1989

[9] Matoušek, J.; Gärtner, B.: *Understanding and using linear programming*. Berlín, Heildelberg, Alemanya: Springer-Verlag, 2007

[10] Murty, K.G.: *Linear programming*. Estats Units: John Wiley & Sons, 1983

[11] Ríos Insua, S.; Ríos Insua, D.; Mateos Caballero, A.; Martín Jiménez, J.: *Programación lineal y aplicaciones. Ejercicios resueltos*. Madrid, Espanya: RA-MA, 1997

Annexos

A Simplex

Per a poder escriure els programes amb LaTeX, s'han tret els accents de les paraules. També s'han separat algunes línies de text per a que càpiguen en una pàgina.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void pivotar(int d, int s, int n, int m, double **A);

int main(void){
    int m, n, i, j, acabat, s, d, iter;
    double u;
    printf("Escriu␣m,␣el␣nombre␣de␣restriccions:\n");
    scanf("%d", &m);
    printf("Escriu␣n,␣el␣nombre␣de␣variables:\n");
    scanf("%d", &n);
    double **A;
    A=(double **)malloc((m+1)*sizeof(double *));
    for(i=0; i<=m; ++i){
        A[i]=(double *)malloc((n+1)*sizeof(double));
    }
    printf("Escriu␣els␣elements␣de␣la␣matriu␣A␣de
␣␣␣␣restriccions␣un␣per␣un␣per␣files:\n");
    for(i=0; i<m; ++i){
        for(j=0; j<n; ++j){
            scanf("%lf", &A[i][j]);
        }
    }
    printf("Escriu␣el␣vector␣b␣dels␣termes␣independents
␣␣␣␣de␣les␣restriccions␣un␣per␣un:\n");
    for(i=0; i<m; ++i){
        scanf("%lf", &A[i][n]);
    }
    printf("Escriu␣el␣vector␣r␣dels␣coeficients␣de␣cost
␣␣␣␣relatiu␣un␣per␣un:\n");
    for(i=0; i<n; ++i){
        scanf("%lf", &A[m][i]);
    }
    printf("Escriu␣el␣valor␣que␣pren␣la␣funcio␣objectiu
␣␣␣␣amb␣la␣solucio␣basica␣inicial:\n");
    scanf("%lf", &A[m][n]);
    acabat=0;
```

```
for(i=0; i<n; ++i){
     if(A[m][i]<0){
          acabat=1;
     }
}
iter=0;
while(acabat==1){
     ++iter;
     printf("Iteracio␣%d:\n", iter);
     /*Trobem la possicio del primer coeficient de
     cost relatiu negatiu*/
     s=0;
     u=0;
     for(i=0; i<n; ++i){
          if(A[m][i]<u){
               u=A[m][i];
               s=i;
          }
     }
     /*Trobem l'element sobre el que pivotar*/
     /*Primer mirem si tots els elements de la columna
     son no positius, en aquest cas, la solucio no esta acotada*/
     d=-1;
     for(i=0; i<m; ++i){
          if(A[i][s]>0){
               d=i;
               i=m+1;
          }
     }
     if(d==-1){
          printf("La␣solucio␣no␣esta␣acotada\n");
          for(i=0; i<=m; ++i){
               free(A[i]);
          }
          free(A);
          return 0;
     }
     u=A[d][n]/A[d][s];
     /*Un cop vist que no tots els elements de la columna
     son no positius, busquem el coeficient mes petit
     associat a un element positiu*/
     for(i=0; i<m; ++i){
          if(A[i][s]>0 && A[i][n]/A[i][s]<u){
             u=A[i][n]/A[i][s];
             d=i;
          }
```

```
        }
        /*Pivotem sobre l'element trobat*/
        pivotar(d, s, n, m, A);
        printf("\n");
        for(i=0; i<=m; ++i){
            for(j=0; j<=n; ++j){
                printf(" %lf ", A[i][j]);
            }
            printf("\n");
        }
        printf("\n");
        /*Comprovem si ja hem acabat*/
        acabat=0;
        for(i=0; i<n; ++i){
            if(A[m][i]<0){
                acabat=1;
            }
        }
    }
    for(i=0; i<=m; ++i){
        free(A[i]);
    }
    free(A);
    return 0;
}

void pivotar(int d, int s, int n, int m, double **A){
    int i, j;
    double num=A[d][s];
    double den;
    for(i=0; i<=n; ++i){
        A[d][i]=A[d][i]/num;
    }
    for(i=0; i<=m; ++i){
        den=A[i][s];
        for(j=0; j<=n; ++j){
            if(i!=d){
                A[i][j]=A[i][j]-(den/A[d][s])*A[d][j];
                /*Ja hem transformat el pivot, per tant
                A[d][s] es 1 i no influeix*/
            }
        }
    }
    return;
}
```

B Transport

Per a poder escriure els programes amb LaTeX, s'han tret els accents de les paraules. També s'han separat algunes línies de text per a que càpiguen en una pàgina.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*Aquest programa resol el problema del transport,
per a nombres enters, utilitzant tres matrius*/

/*La matriu X sera la matriu de les variables i el valor que prenen.
A mes, inclou una fila i una columna extra on es guarden les
quantitats demandades i oferides*/

/*La matriu B serveix per a guardar com variaran les variables
basiques a cada iteracio. Un 2 vol dir que la variables basica no
canvia, un 1 vol dir que se li suma theta i un -1 que se li resta
theta. A les variables no basiques els hi correspon un zero*/

/*La matriu C guarda els costs unitaris i te dos files i columnes
extra. En una fila i en una columna extra es guardaran els
multiplicadors simplexs. En l'altra fila i columna extra es guardara
un 1 si el multiplicador simplex esta determinat i un 0 altrament.*/

int main(void){
    int m, n, i, j, p, sum, faltacol, faltafil, theta, mincofreli,
    mincofrelj, elembase, solf, solc, bdif;
    int mincofrel=-1;
    int **X;
    int **C;
    int **B;
    printf("Escriu␣m,␣el␣nombre␣d'origens:\n");
    scanf("%d", &m);
    printf("Escriu␣n,␣el␣nombre␣de␣destins:\n");
    scanf("%d", &n);
    X=(int **)malloc((m+1)*sizeof(int *));
    for(i=0; i<=m; ++i){
        X[i]=(int *)malloc((n+1)*sizeof(int));
    }
    B=(int **)malloc((m)*sizeof(int *));
    for(i=0; i<m; ++i){
        B[i]=(int *)malloc((n)*sizeof(int));
    }
    for(i=0; i<=m; ++i){
        for(j=0; j<=n; ++j){
```

```
                X[i][j]=0;
            }
        }
        C=(int **)malloc((m+2)*sizeof(int *));
        for(i=0; i<=m+1; ++i){
            C[i]=(int *)malloc((n+2)*sizeof(int));
        }
        /*Posem una fila i columna mes a C per a controlar
        quins multiplicadors estan be (1) i quins no (0)*/
        printf("Escriu␣els␣cost␣unitaris,␣un␣per␣un,␣d'enviar
    ␣␣␣␣una␣unitat␣des␣de␣l'origen␣i␣al␣destÃ␣j:\n");
        for(i=0; i<m; ++i){
            for(j=0; j<n; ++j){
                scanf("%d", &C[i][j]);
            }
        }
        printf("Escriu␣les␣quantitats␣oferides␣per␣a␣cada␣origen:\n");
        for(i=0; i<m; ++i){
            scanf("%d", &X[i][n]);
        }
        printf("Escriu␣les␣quantitas␣demandades␣per␣a␣cada␣destÃ:\n");
        for(i=0; i<n; ++i){
            scanf("%d", &X[m][i]);
        }
        /*Metode de la cantonada nord oest*/
        i=0;
        j=0;
        while(i<m && j<n){
            sum=0;
            for(p=0; p<n; ++p){
                sum=sum+X[i][p];
            }
            faltafil=X[i][n]-sum;
            sum=0;
            for(p=0; p<m; ++p){
                sum=sum+X[p][j];
            }
            faltacol=X[m][j]-sum;
            if(faltacol<faltafil){
                X[i][j]=faltacol;
                ++j;
            }else{
                X[i][j]=faltafil;
                ++i;
            }
        }
```

```
printf("Matriu␣X␣amb␣primera␣base:\n");
for(i=0; i<=m; ++i){
     for(j=0; j<=n; ++j){
          printf("%d␣", X[i][j]);
     }
     printf("\n");
}
/*Costs relatius i nova solucio basica*/
while(mincofrel<0){
     /*++iter;*/
     mincofrel=0;
     /*Borrem els multiplicadors simplexs anteriors*/
     for(i=0; i<m; ++i){
          C[i][n]=0;
          C[i][n+1]=0;
     }
     for(j=0; j<n; ++j){
          C[m][j]=0;
          C[m+1][j]=0;
     }
     C[m+1][n-1]=1; /*Marquem que el multiplicador C[m][n-1]=0
     ja esta ben colocat*/
     elembase=m+n-1;
     /*Netejem la matriu B*/
     for(i=0; i<m; ++i){
          for(j=0; j<n; ++j){
               B[i][j]=0;
          }
     }
     /*Calculem els multiplicadors simplex*/
     while(elembase>0){
          i=m-1;
          j=n-1;
          while(i>=0 || j>=0){
               if(j>=0){
                    for(p=m-1; p>=0; --p){
                         if(X[p][j]!=0 && C[p][n+1]==0 &&
                         C[m+1][j]==1){
                              C[p][n]=C[p][j]-C[m][j];
                              C[p][n+1]=1;
                              --elembase;
                         }
                    }
               }
               if(i>=0){
                    for(p=n-1; p>=0; --p){
```

```
                    if(X[i][p]!=0 && C[m+1][p]==0 &&
                    C[i][n+1]==1){
                        C[m][p]=C[i][p]-C[i][n];
                        C[m+1][p]=1;
                        --elembase;
                    }
                }
            }
            --i;
            --j;
        }
    }
    printf("Matriu␣C:\n");
    for(i=0; i<=m+1; ++i){
        for(j=0; j<=n+1; ++j){
            printf("%d␣", C[i][j]);
        }
        printf("\n");
    }
    /*Busquem si algun coeficient de cost relatiu es negatiu,
    en cas que n'hi hagi algun, ens quedem amb el mes negatiu*/
    for(i=0; i<m; ++i){
        for(j=0; j<n; ++j){
            if((C[i][j]-C[i][n]-C[m][j])<mincofrel){
                mincofrel=(C[i][j]-C[i][n]-C[m][j]);
                mincofreli=i;
                mincofrelj=j;
            }
        }
    }
    printf("El␣minim␣ccr␣es␣%d␣i␣esta␣a␣la␣posicio␣%d␣,
␣␣␣␣␣␣␣␣␣%d\n", mincofrel, mincofreli, mincofrelj);
    /*Busquem el cicle i actualitzem la base, si cal*/
    if(mincofrel!=0){
        elembase=m+n-1;
        /*Netejem la matriu B*/
        for(i=0; i<m; ++i){
            for(j=0; j<n; ++j){
                B[i][j]=0;
            }
        }
        /*Marquem que sumem theta a la posicio de la variable
        que entra a la base*/
        B[mincofreli][mincofrelj]=1;
        /*Busquem els elements de la base que estan sols a una
        fila o columna; els marquem perque es quedaran iguals*/
```

```
X[mincofreli][mincofrelj]=1; /*Canviem momentaniament
aquesta variable per a trobar el cicle*/
solf=1;
solc=1;
for(i=0; i<m; ++i){
    for(j=0; j<n; ++j){
        if(X[i][j]!=0){
            for(p=0; p<m; ++p){
                if(X[p][j]!=0 && p!=i){
                    solc=0;
                }
            }
            for(p=0; p<n; ++p){
                if(X[i][p]!=0 && p!=j){
                    solf=0;
                }
            }
            if(solf==1 || solc==1){
                B[i][j]=2;
                solf=1;
                solc=1;
                --elembase;
            }else{
                solf=1;
                solc=1;
            }
        }
    }
}
printf("Matriu␣B:\n");
for(i=0; i<m; ++i){
    for(j=0; j<n; ++j){
        printf("%d␣", B[i][j]);
    }
    printf("\n");
}
X[mincofreli][mincofrelj]=0;
/*Marquem com canviarem tots els altres elements*/
while(elembase>0){
    for(i=0; i<m; ++i){
        for(j=0; j<n; ++j){
            if(X[i][j]!=0 && B[i][j]==0){
                /*Si en una fila o columna hi ha un +-1 i
                nomÃ©s queda un lloc on posar el -+1, el
                /*Columna*/
                bdif=0;
```

```
for(p=0; p<m; ++p){
     if(B[p][j]!=0 && B[p][j]!=2 && p!=i){
          ++bdif;
     }
}
if(bdif==1){
     for(p=0; p<m; ++p){
          if(B[p][j]!=0 && B[p][j]!=2
          && p!=i){
               if(B[p][j]==1
               && B[i][j]==0){
                    B[i][j]=-1;
                    --elembase;
               }
               if(B[p][j]==-1
               && B[i][j]==0){
                    B[i][j]=1;
                    --elembase;
               }
          }
     }
}
/*Fila*/
bdif=0;
for(p=0; p<n; ++p){
     if(B[i][p]!=0 && B[i][p]!=2 && p!=j){
          ++bdif;
     }
}
if(bdif==1){
     for(p=0; p<n; ++p){
          if(B[i][p]!=0 && B[i][p]!=2
          && p!=j){
               if(B[i][p]==1 && B[i][j]==0){
                    B[i][j]=-1;
                    --elembase;
               }
               if(B[i][p]==-1
               && B[i][j]==0){
                    B[i][j]=1;
                    --elembase;
               }
          }
     }
}
/*Si en una fila o columna hi ha un 1 i
```

```
un -1 i queden elements sense assignar,
els marquem com 2*/
bdif=0;
for(p=0; p<m; ++p){
    if(B[p][j]==1 || B[p][j]==-1){
        ++bdif;
    }
}
if(bdif==2){
    for(p=0; p<m; ++p){
        if(X[p][j]!=0 && B[p][j]==0){
            B[p][j]=2;
            --elembase;
        }
    }
}
bdif=0;
for(p=0; p<n; ++p){
    if(B[i][p]==1 || B[i][p]==-1){
        ++bdif;
    }
}
if(bdif==2){
    for(p=0; p<n; ++p){
        if(X[i][p]!=0 && B[i][p]==0){
            B[i][p]=2;
            --elembase;
        }
    }
}
/*Nomes queda un element per assignar
a una fila o columna i no pot ser +-1*/
/*Columna*/
/*Mirem primer que nomes queda un
element sense assignar*/
bdif=0;
for(p=0; p<m; ++p){
    if(X[p][j]!=0 && B[p][j]==0){
        ++bdif;
    }
}
if(bdif==1){
    /*Mirem que hi hagi exactament un 1
    i un -1 a la columna o que no n'hi
    hagi cap*/
    bdif=0;
```

```
    for(p=0; p<m; ++p){
        if(B[p][j]==1 || B[p][j]==-1){
            ++bdif;
        }
    }
    if(bdif==2 || bdif==0){ /*bdif==2
    descarta la posibilitat que l'elemt
    que queda pugui ser +-1*/
        /*Busquem l'element que falta i
        li posem 2*/
        for(p=0; p<m; ++p){
            if(X[p][j]!=0 && B[p][j]==0){
                B[p][j]=2;
                --elembase;
            }
        }
    }
}
/*Fila*/
/*Mirem primer que nomes queda un
element sense assignar*/
bdif=0;
for(p=0; p<n; ++p){
    if(X[i][p]!=0 && B[i][p]==0){
        ++bdif;
    }
}
if(bdif==1){
    /*Mirem que hi hagi exactament un 1
    i un -1 a la fila o que no n'hi hagi
    cap*/
    bdif=0;
    for(p=0; p<n; ++p){
        if(B[i][p]==1 || B[i][p]==-1){
            ++bdif;
        }
    }
    if(bdif==2 || bdif==0){
    /*bdif==2 descarta la posibilitat que
    l'elemt que queda pugui ser +-1*/
        /*Busquem l'element que falta i
        li posem 2*/
        for(p=0; p<n; ++p){
            if(X[i][p]!=0 && B[i][p]==0){
                B[i][p]=2;
                --elembase;
```

```
                                }
                            }
                        }
                    }
                }
            }
        }
        printf("Matriu␣B:\n");
        for(i=0; i<m; ++i){
            for(j=0; j<n; ++j){
                printf("%d␣", B[i][j]);
            }
            printf("\n");
        }
    }
    /*Busquem theta*/
    for(i=0; i<m; ++i){
        for(j=0; j<n; ++j){
            if(B[i][j]==-1){
                theta=X[i][j];
                i=m;
                j=n;
            }
        }
    }
    for(i=0; i<m; ++i){
        for(j=0; j<n; ++j){
            if(B[i][j]==-1 && X[i][j]<theta){
                theta=X[i][j];
            }
        }
    }
    /*Actualitzem la base*/
    for(i=0; i<m; ++i){
        for(j=0; j<n; ++j){
            if(B[i][j]==-1){
                X[i][j]=X[i][j]-theta;
            }
            if(B[i][j]==1){
                X[i][j]=X[i][j]+theta;
            }
        }
    }
}
printf("Matriu␣X:\n");
for(i=0; i<=m; ++i){
```

```
            for(j=0; j<=n; ++j){
                printf("%d␣", X[i][j]);
            }
            printf("\n");
        }
    }
    for(i=0; i<=m; ++i){
        free(X[i]);
    }
    free(X);
    for(i=0; i<=m+1; ++i){
        free(C[i]);
    }
    free(C);
    for(i=0; i<m; ++i){
        free(B[i]);
    }
    free(B);
    return 0;
}
```