Universitat de Barcelona

Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

OPTIMITZACIÓ LINEAL: ALGUNS EXEMPLES

Autor: Jaume Delgado Costa

Director: Dr. Miquel Bosch Gual

Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, 24 de gener de 2022

Abstract.

Optimization (or programming) is a wide branch of Mathematics that analyzes complex decision or location problems and studies algorithms or strategies to find the best solution.

We are goint to study the basis of linear programming, the simplex method, and we are also going to consider two problems in particular: the transport problem and the minimum cost flow problem.

In addition, we are developing two computer programs that theoretically solve each kind of problem.

Resum.

L'optimització (o programació) és una àmplia branca de les matemàtiques que analitza problemes complexos de decisió o ubicació, i estudia algoritmes o estratègies per trobar la millor solució.

En aquest treball, estudiem les bases de la programació lineal, el mètode símplex, i també abordem dos problemes específics: el problema del transport i el problema de cost mínim.

A més, desenvoluparem dos programes informàtics que teòricament resolen cada tipus de problema.

2020 Mathematics Subject Classification. 90C05, 90C10, 90C46, 90C35, 49N15, 49M05, 49M29, 65K05

Agraïments.

Vull agrair al meu tutor, el Dr. Miquel Bosch Gual, per guiar-me durant la realització d'aquest treball. Tant els seus comentaris com les seves indicacions, a les nostres reunions, m'ha resultat de molta utilitat.

D'altra banda, vull agrair a la meva família i amics per tot el suport que m'han donat quan ho he necessitat. De manera especial, vull destacar la tasca feta per la meva família, que ha estat un pilar fonamental i m'ha donat l'oportunitat d'estudiar aquest grau.

Índex

1 Introducció.

L'optimització és una àrea de les matemàtiques força extensa.

En aquest treball estudiarem els conceptes necessaris per entendre què és la programació lineal, i resoldrem un parell de problemes de forma pràctica. A més plantejarem els algoritmes aplicables a aquests problemes i obtindrem una sèrie de resultats.

L'enunciat del problema general de la programació matemàtica pot ser:

$$\begin{aligned} &\text{minimitzar } f(\mathbf{x}) \\ &\text{subjecte a } \; h_i(\mathbf{x}) = 0, \;\; i = 1, 2, \ldots, m \\ &\qquad\qquad\quad g_j(\mathbf{x}) \leq 0, \;\; j = 1, 2, \ldots, r \\ &\qquad\qquad\quad \mathbf{x} \in S. \end{aligned}$$

En aquesta formulació, $\mathbf{x}$ és un vector d'incògnites n-dimensional $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, i f, h_i, $i = 1, 2, \ldots, m$ i g_j, $j = 1, 2, \ldots, r$, són funcions amb valors reals de les variables $(x_1, x_2, \ldots, x_n)$. El conjunt S és un subconjunt d'un espai n-dimensional. La funció f és la *funció objectiu* del problema i les equacions, les desigualtats i el conjunt de restricció són les *restriccions*. A aquest tipus de problema també l'anomenarem *programa*.

En un *programa lineal* la funció objectiu és necessàriament lineal en les incògnites i les restriccions consten d'igualtats i desigualtats lineals.

Una mesura de la complexitat d'un problema de programació és la seva mida, que es mesura en funció del nombre de variables incògnites i del nombre de restriccions. La mida dels problemes que es poden resoldre amb efectivitat, ha anat augmentant amb l'avenç de la tecnologia de computació i també gràcies al progrés teòric.

La teoria inicial associada amb la programació, es proposa obtenir condicions necessàries i suficients que es satisfan a un punt solució, més que qüestions de computació. Però, si sabem des d'un principi que un problema donat és massa gran i complex s'abandona la idea de resoldre el conjunt de condicions necessàries, en favor d'un procediment més directe de cerca de punts cada vegada millors fins a arribar a l'òptim.

Per resoldre problemes grans s'utilitzen, en general, algoritmes iteratius, i és requerit l'ús de computadors. Com a norma, al cercar un vector que resolgui el problema d'optimització, s'eligeix un vector inicial $\mathbf{x}_0$ i l'algoritme genera un vector millorat $\mathbf{x}_1$. Aquest procés es repeteix i es troba una solució millor $\mathbf{x}_2$. Continuant d'aquesta manera, es troba una successió de punts cada vegada més apropiats $\mathbf{x}_0$, $\mathbf{x}_1$,..., $\mathbf{x}_k$,..., que tendeix cap a un punt solució $\mathbf{x}^*$. En problemes de programació lineal, la successió que es genera és finita, obtenint un punt solució en un nombre finit de passos.

La teoria dels algoritmes iteratius pot dividir-se en tres aspectes. El primer es refereix a la creació dels algoritmes propiament dits, que es basen en un estudi creatiu del programa d'optimització. El segon aspecte és l'anàlisi de convergència global, la qual estudia si l'algoritme, iniciat amb un vector inicial allunyat del punt solució, acabarà convergint cap aquest. El tercer aspecte es refereix a l'anàlisi de la convergència local, i tracta de la proporció en què la successió generada de vectors convergeix cap a la solució, ja que no només és important el fet de convergir a la solució, sinó també fer-ho ràpidament.

1.1 Objectius.

Ara anem a introduir l'objectiu principal i alguns objectius específics que es volen assolir en la realització d'aquest treball.

L'objectiu principal d'aquest treball és estudiar la branca lineal de la programació matemàtica, enfocada a dos problemes en específic: el problema del transport i el problema del flux de cost mínim.

Com a objectius específics tenim:

- Veure les bases de la programació lineal.
- Plantejar el problema del transport i veure la seva resolució mitjançant el mètode símplex.
- Estudiar el problema del flux de cost mínim i veure la seva resolució amb l'algoritme de les desviacions.
- Elaborar programes informàtics que apliquin els algoritmes estudiats.

1.2 Estructura de la memòria.

L'estructura d'aquesta memòria es divideix en les següents parts:

- A la secció 1, expliquem els conceptes comuns a la programació matemàtica i definim els conceptes que hi intervenen i exposem la teoria dels algoritmes iteratius.
- A la secció 2 introduïm la programació lineal i es presenten els conceptes bàsics amb els quals treballarem. En aquesta secció també s'estudia el mètode símplex.
- A la secció 3 es presenta el problema del transport. S'estudia la seva estructura i diversos algoritmes per arribar a una solució factible inicial. Posteriorment, s'estudia la dualitat i s'explica la resta de l'algoritme de resolució.
- A la secció 4 es veuen els conceptes necessaris per a l'estudi posterior del problema del flux de cost mínim, es veu l'algoritme de les desviacions que el resol i s'estudia la convergència del programa realitzat.
- A la secció 5 es presentaran les conclusions del treball.

2 Programació lineal.

2.1 Introducció.

Per a la redacció d'aquest capítol, he utilitzat principalment com a referència els elements [7] i [12] de la bibliografia.

Un problema de programació lineal és un programa matemàtic en el qual la funció objectiu és lineal en les incògnites i les restriccions consten d'igualtats i desigualtats lineals. La forma d'aquestes restriccions pot variar d'uns problemes a altres, però qualsevol programa lineal es pot convertir a la següent forma estàndard:

$$\begin{array}{ll}
\text{minimitzar} & c_1x_1 + c_2x_2 + ... + c_nx_n \\
\text{subjecte a} & a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\
 & a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\
 & \quad . \qquad\qquad\qquad\qquad\quad . \\
 & \quad . \qquad\qquad\qquad\qquad\quad . \\
 & \quad . \qquad\qquad\qquad\qquad\quad . \\
 & a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n = b_m \\
\text{i} & x_1 \geq 0,\ x_2 \geq 0, ...,\ x_n \geq 0,
\end{array}$$

On les b_j, c_i i a_{ji} són constants reals fixes i les x_i són valors reals a determinar, $j = 1, 2, \ldots, m$, $i = 1, 2, \ldots, n$.

En una notació vectorial més compacta, aquest problema comú pot expresar-se com:

$$\begin{aligned}
\text{minimitzar } & \mathbf{c^T x} \\
\text{subjecte a } & \mathbf{Ax} = \mathbf{b} \\
& \mathbf{x} \geq 0
\end{aligned} \tag{2.1}$$

Aquí $\mathbf{x}$ és un vector columna n-dimensional, $\mathbf{c^T}$ és un vector fila n-dimensional, $\mathbf{A}$ és una matriu de $m \times n$, i $\mathbf{b}$ és un vector columna m-dimensional. La desigualtat vectorial $\mathbf{x} \geq 0$ vol dir que totes les components de $\mathbf{x}$ són positives.

Solucions bàsiques i Teorema Fonamental de la Programació Lineal.

Considerem el sistema d'igualtats

$$\mathbf{Ax} = \mathbf{b}, \tag{2.2}$$

on $\mathbf{x}$ és un n-vector, $\mathbf{b}$ un m-vector i $\mathbf{A}$ una matriu de $m \times n$.

Suposem que de les n columnes d'$\mathbf{A}$ es selecciona un conjunt de m columnes linealment independents (existeix si el rang d'$\mathbf{A}$ és m).

Suposem que es seleccionen les primeres m columnes d'$\mathbf{A}$, i $\mathbf{B}$ és la matriu $m \times m$ determinada per aquestes columnes. Aleshores, $\mathbf{B}$ és no singular i l'equació

$$\mathbf{Bx_B} = \mathbf{b} \tag{2.3}$$

només es pot resoldre per al m-vector $\mathbf{x_B}$. Al fer $\mathbf{x} = (\mathbf{x_B}, \mathbf{0})$, s'obté una solució per a (2.2), el que planteja la següent definició:

Definició 2.1. *Donat el conjunt* (2.2) *de m equacions lineals simultànies de n incògnites, sigui* $\boldsymbol{B}$ *qualsevol submatriu de* $m \times m$ *no singular formada per columnes d'*$\mathbf{A}$. *Aleshores si totes les* $n-m$ *components de x no associades a les columnes de* $\boldsymbol{B}$ *s'igualen a zero, la solució del conjunt resultant d'equacions s'anomena solució bàsica de (2.2) respecte a la base* $\boldsymbol{B}$. *Les components de* $\boldsymbol{x}$ *associades a columnes de* $\boldsymbol{B}$ *es denominen variables bàsiques.*

A menys que s'inidiqui el contrari, en aquesta secció treballarem a partir d'ara baix la següent hipòtesi:

Hipòtesi del rang complet. La matriu $\mathbf{A}$ de m×n té m<n, i les m files d'$\mathbf{A}$ són linealment independents

Baix aquesta hipòtesi, el sistema (2.2) sempre tindrà solució i, de fet, sempre tindrà una solució bàsica.

Definició 2.2. *Si una o més variables bàsiques d'una solució bàsica són igual a zero, es diu que aquesta solució és una solució bàsica degenerada .*

Fins ara, només s'ha tractat la restricció d'igualtat (2.2) i no s'ha fet resferència a la restricció de la posivitat sobre les variables. Amb aquestes restriccions, tenim definicions similars. Considerem ara el sistema de restriccions

$$\begin{aligned} \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq 0 \end{aligned} \tag{2.4}$$

que representa les restriccions d'un programa lineal en forma estàndar.

Definició 2.3. *Un vector* $\boldsymbol{x}$ *que satisfaga* (2.4)*, es diu que es factible per a aquestes restriccions. Una solució factible per a les restriccions* (2.4)*, que també sigui bàsica, es denomina solució factible bàsica, si, a més, aquesta solució és una solució bàsica degenerada, es denomina solució factible bàsica degenerada.*

El teorema fonamental de la programació lineal estableix la importància primordial de les solucions factibles bàsiques per a resoldre problemes de programació lineal. El teorema demostra que al cercar una solució optimal per a un programa lineal només és necessari tenir en compte les solucions factibles bàsiques, donat que en aquest tipus de solució sempre es consegueix el valor optimal. Considerem un programa lineal en la forma estàndard (2.1) una solució factible per a les restriccions que assoleixi el valor mínim en la funció objectiu subjecte a aquestes restriccions, es denomina *solució factible optimal.* Si aquesta solució és bàsica, és una *solució factible bàsica optimal.*

Teorema 2.4 (**Teorema fonamental de la programació lineal**). *Donat un programa lineal en la forma estàndard* (2.1)*, on* $\mathbf{A}$ *es una matriu de m×n de rang m,*

1. *si hi ha una solució factible, hi ha una solució factible bàsica;*
2. *si hi ha una solució factible optimal, hi ha una solució factible bàsica optimal.*

Demostració. De 1. Siguin $\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_n}$ les columnes d'$\mathbf{A}$. Suposem que $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Llavors, en funció de les columnes d'$\mathbf{A}$, aquesta solució satisfà:

$$x_1\mathbf{a_1} + x_2\mathbf{a_2} + \ldots + x_n\mathbf{a_n} = \mathbf{b}$$

Suposem que exactament p de les variables x_i són majors que zero, i, per conveniència, que són les primeres p variables. Llavors,

$$x_1\mathbf{a_1} + x_2\mathbf{a_2} + \ldots + x_p\mathbf{a_p} = \mathbf{b} \tag{2.5}$$

Ara hi ha dos casos, que responen a si el conjunt $\mathbf{a_1}, \mathbf{a_2}, \ldots \mathbf{a_p}$ és linealment independent o linealment dependent.

Cas 1: Suposem que $\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_p}$ són linealment independents. Llavors, és evident que $p \leq m$. Si $p = m$, la solució és bàsica i la demostració està completa. Si $p < m$, com $\mathbf{A}$ té rang m, per la hipòtesi del rang complet, es poden trobar $m - p$ vectors dels restants $n-p$ vectors, pel que el conjunt resultant de m vectors és linealment independent. L'assignació del valor zero a les $m - p$ variables corresponents dóna una solució factible bàsica (degenerada).

Cas 2: Suposem que $\mathbf{a_1}, \mathbf{a_2}, \ldots \mathbf{a_p}$ són linealment dependents. Llavors, hi ha una combinació lineal no trivial d'aquests vectors que és zero. Així, hi ha un conjunt de constants $y_1, y_2, \ldots, y_p$, una de les quals, almenys, es pot suposar que és positiva, de manera que

$$y_1\mathbf{a_1} + y_2\mathbf{a_2} + \cdots + y_p\mathbf{a_p} = 0$$

Multiplicant aquesta equació per un escalar ε i restant-la de (2.5), s'obté

$$(x_1 - \varepsilon y_1)\mathbf{a_1} + (x_2 - \varepsilon y_2)\mathbf{a_2} + \cdots + (x_p - \varepsilon y_p)\mathbf{a_p} = \mathbf{b}$$

Aquesta equació és valida per a tot ε, i per cada ε les components $x_i - \varepsilon y_i$ corresponen a una solució de les equacions lineal, tot i que poden violar $x_i - \varepsilon y_i \geq 0$ Fent $\mathbf{y} = (y_1, y_2, \ldots, y_p, 0, 0, \ldots, 0)$, s'observa que per qualsevol ε

$$\mathbf{x} - \varepsilon\mathbf{y} \tag{2.6}$$

és una solució per a les igualtats. A mesura que augmenta ε des de zero, les components augmenten, disminueixen o romanen constants depenent si la component y_i és negativa, positiva o nul·la, respectivament. Com hem suposat que almenys una y_i és positiva, com a mínim una component decreixerà quan creix ε. Incrementem ε fins al primer punt on una o més components són zero. Específicament, fem

$$\varepsilon = \min\{x_i/y_i : y_i > 0\} \tag{2.7}$$

Per a aquest valor de ε, la solució donada per (2.6) és factible i té com a molt $p - 1$ variables positives. Al repetir aquest procés, en cas necessari, es podríen eliminar les variables positives fins tenir una solució factible amb les columnes corresponents que són linealment independents. En aquest punt és aplicable el cas 1. □

Demostració. De 2. Sigui $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ una solució factible optimal. Suposem, com abans, que hi ha exactament p variables positives, $x_1, x_2, \ldots, x_p$.

De nou hi ha dos casos; el cas 1, corresponent a la independència lineal, exactament igual que abans. El cas 2 també es desenvolupa igual que l'anterior, però s'ha de demostrar que per a qualsevol ε la solució (2.6) és optimal. Per a provar-ho, observem que el valor de la funció objectiu és

$$\mathbf{c^T x} - \varepsilon\mathbf{c^T y} \tag{2.8}$$

Per a ε de maginitud suficientment petita, $\mathbf{x} - \varepsilon\mathbf{y}$ és una solució factible per a valors positius o negatius de ε. Així, es conclou que $\mathbf{c^T y} = 0$. Doncs, si $\mathbf{c^T y} \neq 0$, pot determinar-se una ε de magnitud petita i signe apropiat que manté (2.8) més petita que $\mathbf{c^T x}$, conservant la factibilitat. Això violaria la hipòtesi d'optimalitat de $\mathbf{x}$, pel que hem de tenir $\mathbf{c^T y} = 0$.

Una vegada establert que la nova solució factible amb menys components positius també és optimal, la resta de la demostració pot completar-se com al cas *1.*, referent a la solució factible bàsica. □

2.2 El mètode simplex.

El mètode símplex va ser desenvolupat pel matemàtic George B. Dantzig, considerat el pare de la programació lineal. La idea del mètode simplex és passar d'una solució factible bàsica del conjunt de restricció d'un problema en forma estàndard a una altra solució bàsica, de manera que el valor de la funció objectiu decreixi continuament fins a arribar a un mínim. Hem vist, a 2.4, que per cercar una solució factible optimal és suficient considerar només les solucions factibles bàsiques. Aquest mètode va provant diferents solucions bàsiques fins trobar la solució minimal.

El conjunt d'equacions lineals simultànies:

$$\begin{array}{ccc} a_{11}x_1 & + & a_{12}x_2 + ... + a_{1n}x_n = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 + ... + a_{2n}x_n = & b_2 \\ \vdots & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 + ... + a_{mn}x_n = & b_m \end{array} \tag{2.9}$$

on $m \leq n$. En forma matricial, això s'expressa

$$\mathbf{Ax} = \mathbf{b} \tag{2.10}$$

Aquests conjunts d'equacions (2.9) i (2.10) es poden interpretar en E^m com una equació vectorial. Si es denoten les columnes d'$\mathbf{A}$ per $\mathbf{a_1}, \mathbf{a_2}, \cdots \mathbf{a_n}$, (2.9) es pot expressar com:

$$x_1\mathbf{a_1} + x_2\mathbf{a_2} + \cdots + x_n\mathbf{a_n} = \mathbf{b} \tag{2.11}$$

En aquesta interpretació es pretén expressar $\mathbf{b}$ com combinació lineal de les $\mathbf{a_i}$.

Si $m < n$ i els vectors $\mathbf{a_i}$ s'extenen fins E^m, aleshores no hi ha una solució única, sinó tota una família de solucions. Tot i així, el vector $\mathbf{b}$ té una representació única, combinació lineal d'un subconjunt linealment independent donat d'aquests vectors. La solució bàsica corresponent a $n - m$ variables x_i igualades a zero, és una solució bàsica de (2.9).

Suposem ara que es comença amb un sistema en forma canònica, és a dir, entre les n variables, hi ha m bàsiques amb la propietat que cada una apareix amb coeficient 1 en només una equació i mai apareixen dues d'aquestes variables en una mateixa equació, corresponent a la taula:

$$\begin{array}{ccccccccc} 1 & 0 & \cdots & 0 & y_{1,m+1} & y_{1,m+2} & \cdots & y_{1,n} & y_{10} \\ 0 & 1 & \cdot & 0 & y_{2,m+1} & y_{2,m+2} & \cdots & y_{2,n} & y_{20} \\ 0 & 0 & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & 1 & y_{m,m+1} & y_{m,m+2} & \cdots & y_{m,n} & y_{m0} \end{array} \tag{2.12}$$

En aquest cas, els m primers vectors formen una base. A més, qualsevol altre vector representat en la taula pot expressar-se com una combinació lineal d'aquests vectors bàsics llegint els coeficients de la columna corresponent.

$$\mathbf{a_j} = y_{1j}\mathbf{a_1} + y_{2j}\mathbf{a_2} + \cdots + y_{mj}\mathbf{a_m} \tag{2.13}$$

Pot interpretar-se que la taula proporciona les representacions dels vectors $\mathbf{a_i}$ en funció de la base, la i-èsima columna de la taula és la representació del vector $\mathbf{a_i}$. En concret, l'expressió per a $\mathbf{b}$ en funció de la base està donada per la última columna.

Si volem substituir un membre de la base per un altre que encara no estigui en ella, per exemple, es vol substituir el vector de la base $\mathbf{a_p}$ $1 \leq p \leq m$, pel vector $\mathbf{a_q}$ (amb $q > m$) . Sempre que els m primers vectors, una vegada feta la substitució de $\mathbf{a_p}$ per $\mathbf{a_q}$, siguin linealment independents, aquests vectors contitueixen una base, i tot vector podrà formular-se com una combinació lineal d'aquesta nova base. Per trobar les noves representacions, hi ha que actualitzar la taula. La condició d'independència lineal es compleix si, i només si, $y_{pq} \neq 0$.

Qualsevol vector $\mathbf{a_j}$ pot expressar-se en funció de la taula (2.12). Per a $\mathbf{a_q}$, resulta

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

on es pot aïllar $\mathbf{a_p}$,

$$\mathbf{a_p} = \frac{1}{y_{pq}}\mathbf{a_q} - \sum_{\substack{i=1 \\ i \neq p}}^{m} \frac{y_{iq}}{y_{pq}}\mathbf{a_i} \tag{2.14}$$

Substituint (2.14) en (2.13), resulta:

$$\mathbf{a_j} = \sum_{\substack{i=1 \\ i \neq p}}^{m} (y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj})\mathbf{a_i} + \frac{y_{pj}}{y_{pq}}\mathbf{a_q} \tag{2.15}$$

Al denotar per y'_{ij} els coeficients de la nova taula, que proporciona les combinacions lineals, a partir de (2.15) resulta directament

$$\begin{aligned} y'_{ij} &= y_{ij} - \frac{y_{iq}}{y_{pq}} y_{pj}, \qquad \text{i} \neq \text{p} \\ y'_{pj} &= \frac{y_{pj}}{y_{pq}} \end{aligned}$$

Aquestes són les equacions pivot, que solen presentar-se en programació lineal. Es diu que l'element y_{pq} és l'*element pivot* del sistema original. D'aquesta manera hem realitzat una operació pivot la qual ha generat una solució bàsica nova partint d'una solució bàsica inicial. Però, ens hem d'assegurar que la solució a la qual arribem sigui essent factible.

Durant la resta de secció, per simplificar els arguments, introduirem la següent suposició:

Suposició de no degenaració . *Tota solució factible bàsica de* (2.4) *és una solució factible bàsica no degenerada.*

Quan no es compleix, el mètode símplex pot no funcionar si no es corregeix correctament.

Vegem com s'incorpora un vector de la matriu $\mathbf{A}$ a la base. Suposem que disposem d'una solució factible bàsica $\mathbf{x} = (x_1, x_2, \ldots, x_m, 0, 0, \ldots, 0)$, o la representació

$$x_1\mathbf{a_1} + x_2\mathbf{a_2} + \cdots + x_m\mathbf{a_m} = \mathbf{b}. \tag{2.16}$$

Per la suposició de no degenació, $x_i > 0$ $i = 1, 2, \ldots, m$. Si ara decidim incorporar el vector $\mathbf{a_q}$ a la respresentació, $q > m$. Disposem d'una representació de $\mathbf{a_q}$ en funció de la base actual

$$\mathbf{a_q} = y_{1q}\mathbf{a_1} + y_{2q}\mathbf{a_2} + \cdots + y_{mq}\mathbf{a_m} \tag{2.17}$$

Multiplicant (2.17) per una variable $\varepsilon \geq 0$ i restant de (2.16), tenim

$$(x_1 - \varepsilon y_{1q})\mathbf{a_1} + (x_2 - \varepsilon y_{2q})\mathbf{a_2} + \cdots + (x_m - \varepsilon y_{mq})\mathbf{a_m} + \varepsilon\mathbf{a_q} = \mathbf{b} \tag{2.18}$$

Per qualsevol $\varepsilon \geq 0$, (2.18) expressa $\mathbf{b}$ com una combinació lineal de m+1 vectors com a màxim. Quan s'incrementa el valor de ε a partir de zero, el coeficient de $\mathbf{a_q}$ va augmentat i els d'altres vectors creixeran, decreixeran o romandran iguals mentre s'incrementa ε. Si decreix algun, es pot igualar ε al valor corresponent a la primera posició on un (o més) dels coeficients s'anul·la. És a dir,

$$\varepsilon = \min_i \{x_i/y_{iq} : y_{iq} > 0\} \tag{2.19}$$

En aquest cas, es té una nova solució factible bàsica, amb el vector $\mathbf{a_q}$ substituint al vector $\mathbf{a_p}$, on p correspon a l'índex minimitzador de (2.19). Si s'assoleix el mínim de (2.19) per més d'un sol índex i, aleshores la nova solució és degenerada, i qualsevol dels vectors amb component zero es pot considerar com el que deixa la base.

Si cap dels y_{iq} són positius, s'observa que hi ha solucions factibles per (2.4) que tenen coeficients arbitriàriament grans. Per tant, el conjunt K de solucions factible no està acotat.

Resumint, tenim que donada una solució factible bàsica i un vector arbitrari $\mathbf{a_q}$, existeix una nova solució factible bàsica amb $\mathbf{a_q}$ a la seva base i un dels vectors originals eliminats, o el conjunt de solucions factibles no està acotat.

El mètode símplex es basa en seleccionar la columna de manera que la nova solució factible bàsica resultant proudeixi un valor per a la funció objectiu menor que l'anterior, determinant amb senzills càlculs quin vector ha de sortir de la base i quin ha d'entrar a aquesta, conservant la factibilitat.

Si tenim una solució factible bàsica

$(\mathbf{x_B}, 0) = (y_{10}, y_{20}, \ldots, y_{m0}, 0, \ldots, 0)$ junt amb la taula del problema en la forma canònica:

$$\begin{array}{ccccccccc}
\mathbf{a_1} & \mathbf{a_2} & \cdots & \mathbf{a_m} & \mathbf{a_{m+1}} & \mathbf{a_{m+2}} & \cdots & \mathbf{a_n} & \mathbf{b} \\
1 & 0 & \cdots & 0 & y_{1,m+1} & y_{1,m+2} & \cdots & y_{1,n} & y_{10} \\
0 & 1 & \cdot & 0 & y_{2,m+1} & y_{2,m+2} & \cdots & y_{2,n} & y_{20} \\
0 & 0 & \cdot & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
0 & 0 & \cdot & 1 & y_{m,m+1} & y_{m,m+2} & \cdots & y_{m,n} & y_{m0}
\end{array} \tag{2.20}$$

El valor de la funció objectiu a qualsevol solució $\mathbf{x}$ és

$$z = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n, \tag{2.21}$$

pel que per la solució bàsica, el valor corresponent és

$$z_0 = \mathbf{c_B^T x_B} \tag{2.22}$$

on $\mathbf{c_B^T} = [c_1, c_2, \ldots, c_m]$

És evident que si assignem valors arbitraris a $x_{m+1}, x_{m+2}, \ldots, x_n$, es poden aïllar sense problema les variables restants, d'aquesta manera:

$$\begin{aligned} x_1 &= y_{10} - \textstyle\sum_{j=m+1}^{n} y_{1j} x_j \\ x_2 &= y_{20} - \textstyle\sum_{j=m+1}^{n} y_{2j} x_j \\ &\vdots \\ x_m &= y_{m0} - \textstyle\sum_{j=m+1}^{n} y_{mj} x_j \end{aligned} \tag{2.23}$$

Utilitzant (2.23) es poden eliminar $x_1, x_2, \ldots, x_m$ de la fórmula general (2.22). Al fer-ho s'obté.

$$z = \mathbf{c^T x} = z_0 + (c_{m+1} - z_{m+1}) x_{m+1} + (c_{m+2} - z_{m+1}) x_{m+2} + \cdots + (c_n - z_n) x_n \tag{2.24}$$

on

$$z_j = y_{1j} c_1 + y_{2j} c_2 + \cdots + y_{mj} c_m \tag{2.25}$$

que és la relació necessària per a determinar la columna pivot. Aquesta equació ens dóna els valors de la funció objectiu z per a qualsevol solució de $\mathbf{Ax} = \mathbf{b}$ en funció de les variables $x_{m+1}, \ldots, x_n$. D'allí es pot determinar si té algun avantatge canviar la solució bàsica introduint una de les variables no bàsiques. Observem que si $c_j - z_j$ és negatiu per algun j, $m + 1 \leq j \leq n$, al augmentar x_j des de zero fins a cert valor positiu, disminuiria el cost total, obtenint una solució millor per al problema. Les fórmules (2.24) i (2.25) tenen en compte els canvis que es necessitarien en els valors de les variables bàsiques $x_1, x_2, \ldots, x_m$ per a adoptar el canvi en x_j. Vist això tenim la condició de millora, deduïda de l'anterior observació

Teorema 2.5 (**Millora de la solució factible bàsica**). *Donada una solució factible bàsica no degenerada amb valor objectiu corresponent a z_0, suposem que per alguna j es compleix $c_j - z_j < 0$. Llavors, hi ha una solució factible amb valor objectiu $z < z_0$. Si la columna $\mathbf{a_j}$ es pot substituir per un vector de la base original i generar una nova solució factible bàsica, aquesta nova solució tindrà $z < z_0$. Si $\mathbf{a_j}$ no es pot substituir per a generar una solució bàsica, llavors el conjunt solució K no està acotat i la funció objectiu pot fer-se arbitràriament petita (tendint a menys infinit).*

Si en algun moment $c_j - z_j < 0$ per a alguna j, x_j es pot fer positiva i disminuir la funció objectiu. També tenim:

Teorema 2.6 (**Teorema de la condició d'optimalitat**). *Si existeix una solució factible bàsica tal que $c_j - z_j \leq 0$ per qualsevol j, llavors aquesta solució és optimal.*

Com les constants $c_j - z_j$ tenen un paper tan important en le mètode símplex, introduirem la notació $r_j = c_j - z_j$ i anomenarem a les r_j *coeficients de cost relatiu.*

Anem a veure ara com és l'algortime del mètode símplex. Suposem que es parteix d'una solució factible bàsica i que la taula corresponent a $\mathbf{Ax} = \mathbf{b}$ està en la forma canònica per la solució.

A la part baixa de la fila afegirem una fila formada pels coeficients de cost relatiu i el valor negatiu del cost actual. El resultat l'anomenem *taula símplex*

$$\begin{array}{cccccccccccc}
\mathbf{a_1} & \mathbf{a_2} & \cdots & \mathbf{a_m} & \mathbf{a_{m+1}} & \mathbf{a_{m+2}} & \cdots & \mathbf{a_j} & \cdots & \mathbf{a_n} & \mathbf{b} \\
\hline
1 & 0 & \cdots & 0 & y_{1,m+1} & y_{1,m+2} & \cdots & y_{1,j} & \cdots & y_{1,n} & y_{10} \\
0 & 1 & \cdots & 0 & y_{2,m+1} & y_{2,m+2} & \cdots & y_{2,j} & \cdots & y_{2,n} & y_{20} \\
. & \cdots & . & . & . & . & . & . & . & . & . \\
. & \cdots & . & . & . & . & . & . & . & . & . \\
0 & 0 & \cdots & 0 & y_{i,m+1} & y_{i,m+2} & \cdots & y_{i,j} & \cdots & y_{i,n} & y_{i0} \\
. & \cdots & . & . & . & . & . & . & . & . & . \\
. & \cdots & . & . & . & . & . & . & . & . & . \\
0 & 0 & \cdots & 1 & y_{m,m+1} & y_{m,m+2} & \cdots & y_{m,j} & \cdots & y_{m,n} & y_{m0} \\
\hline
0 & 0 & \cdots & 0 & r_{m+1} & r_{m+2} & \cdots & r_j & \cdots & r_n & -z_0
\end{array} \tag{2.26}$$

La solució bàsica corresponent a aquesta taula és

$$x_i = \begin{cases} y_{i0} & 0 \leq i \leq m \\ 0 & m+1 \leq i \leq n \end{cases} \tag{2.27}$$

la qual es presuposa factible, per tant, $y_{i0} \geq 0$, $i = 1, 2, \ldots, m$. Els coeficients del cost relatiu r_j indiquen si el valor de la funció objectiu creixerà o decreixerà quan es pivota x_j sobre la solució.

L'algoritme es podria resumir en els següents passos:

- *Pas 0.* Es forma una taula com la (2.26) corresponent a una solució factible bàsica. Els coeficients de cost relatiu es poden trobar amb la reducció de files.
- *Pas 1.* Si tots els $r_j \geq 0$, aturem; la present solució factible bàsica és optimal.
- *Pas 2.* Si no es optimal, es selecciona un q tal que $r_q < 0$, per definir quina variable no bàsica es tornarà en bàsica.
- *Pas 3.* Es calulen els quocients y_{i0}/y_{iq} per a $y_{iq} > 0$ $i = 1, 2, \ldots, m$. Si cap $y_{iq} > 0$, aturem; el problema no està acotat. Si no és així, es selecciona p com l'índex i corresponent al mínim quocient.
- *Pas 4.* Es pivota sobre el pq-èssim element, actualitzant les files, inclosa la última. Es torna al *Pas 1.*

3 Problema del transport.

3.1 Estructura del problema i solució factible inicial.

Al problema del transport hi ha m orígens, $O_1, O_2, \ldots, O_m$, els quals tenen certes quantitats d'un producte que ha d'enviar-se a n destins $D_1, D_2, \ldots, D_n$.

A aquest problema, l'origen i conté una quantitat a_i i el destí j necessita una quantitat b_j.

Suposem que els nombres a_i i b_j, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$, són no negatius.

Hi ha un cost unitari c_{ij} relacionat amb l'enviament del producte de l'origen O_i al destí D_j.

El problema és trobar el model d'enviament entre orígens i destins que satisfaga les necesitats i minimitzi el cost total de l'enviament.

En termes matemàtics, el problema anterior pot ésser expressat com la cerca d'un conjunt de x_{ij}, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$. El model lineal corresponent és:

$$\text{minimitzar } \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{3.1}$$

$$\begin{aligned} \text{subjecte a } \sum_{j=1}^{n} x_{ij} &\leq a_i \text{ per a } i = 1, 2 \ldots, m \\ \sum_{i=1}^{m} x_{ij} &\geq b_j \text{ per a } j = 1, 2 \ldots, n \\ x_{ij} &\geq 0 \text{ per a tot } i \text{ i } j. \end{aligned}$$

Llavors, el problema és determinar x_{ij}, el nombre d'unitats transportades de O_i a D_j, de manera que els subministraments siguin consumits i les demandes satisfetes al mínim cost total possible.

Les primeres m restriccions corresponen als límits de subministrament, i expressen que les unitats que subministren a cada origen no poden ser excedides. Les següents n restriccions asseguren que les demandes dels destins seran satisfetes. Les variables de decisió x_{ij} són definides positives, ja que representen el nombre d'unitats transportades.

Així, en particular, el problema del transport en forma estàndard és el següent:

$$\text{minimitzar } \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{3.2}$$

$$\begin{aligned} \text{subjecte a } \sum_{j=1}^{n} x_{ij} &= a_i \text{ per a } i = 1, 2 \ldots, m \\ \sum_{i=1}^{m} x_{ij} &= b_j \text{ per a } j = 1, 2 \ldots, n \\ x_{ij} &\geq 0 \text{ per a tot } i \text{ i } j. \end{aligned}$$

L'estructura d'aquest problema s'obseva millor escrivint les equacions de restriccions de manera estàndard:

$$
\begin{array}{ccccccccc}
x_{11} & +x_{12}+\cdots+x_{1n} & & & & & & = & a_1 \\
& & x_{21} & +x_{22}+\cdots+x_{2n} & & & & = & a_2 \\
& & & \vdots & & & & & \vdots \\
& & & & x_{m1} & +x_{m2}+\cdots+x_{mn} & & = & a_m \\
\hline
x_{11} & & +x_{21} & \ldots & +x_{m1} & & & = & b_1 \\
x_{12} & & & +x_{22} & \ldots & +x_{m2} & & = & b_2 \\
& & & \vdots & & & & & \vdots \\
x_{1n} & & & +x_{2n} & \ldots & & +x_{mn} & = & b_n
\end{array}
\tag{3.3}
$$

Per tant, la matriu $\mathbf{A}$ dels coeficients del sistema d'equacions anterior s'expressa en notació vector-matriu de la següent manera:

$$
\mathbf{A} = \begin{pmatrix}
1^T & & & & & \\
& 1^T & & & & \\
& & \cdot & & & \\
& & & \cdot & & \\
& & & & \cdot & \\
& & & & & 1^T \\
\mathbf{I} & \mathbf{I} & & \cdots & & \mathbf{I}
\end{pmatrix}
\tag{3.4}
$$

On $1^T = (1, 1, \ldots, 1)$ és n-dimensional i cada $\mathbf{I}$ és una matriu identitat de $n \times n$.

Les dades importants per a resoldre el problema del transport es poden representar utilitzant l'anomenada *taula dels costs de transport*, la qual presenta els origens amb el seu subministrament, els destins amb la seva demanda i el cost per transport unitari d'un origen i a un destí j :

	D_1	D_2	$\cdots$	D_n	subministrament
O_1	c_{11}	c_{12}	$\cdots$	c_{1n}	a_1
O_2	c_{21}	c_{22}	$\cdots$	c_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
O_m	c_{m1}	c_{m2}	$\cdots$	c_{mn}	a_m
Demanda	b_1	b_2	$\cdots$	b_n	

Podem aprofitar l'estructura especial que presenta el problema del transport per adaptar l'algoritme del mètode símplex i d'aquesta manera obtenir d'una manera més eficient una solució.

Ara presentarem una sèrie de definicions i teoremes que ens permetran entendre millor el mètode per solucionar els problemes del transport.

Teorema 3.1. *La condició necessària i suficient que garanteix l'existència d'una solució al problema del transport és que la demanda total sigui igual al subministrament total.*

D'aquest teorema sorgeix la següent definició:

Definició 3.2. *Un problema del trasport es diu que està equilibrat si* $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Si el problema del transport no està equilibrat, hem de convertir-lo en un d'equilibrat abans de solucionar-lo. Es presenten dos casos possibles:

1r cas. La demanda supera el subministrament, $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$

En aquest cas, no és possible satisfer la demanda total amb el subministrament existent. Per tant, s'afegeix un origen fictici O_{m+1} per equilibrar el model. El seu subministrament i cost per unitat transportada són els següents:

$$a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i.$$
$$c_{m+1,j} = 0, j = 1, \ldots, n.$$

Com el subministrament que ve de O_{m+1} no existeix, un destí que satisfà la seva demanda amb unitats d'aquest origen fictici realment no la satisfarà, i per això en alguns casos pràctics de vegades en comptes de posar el cost $c_{m+1,j}$ a zero, s'afegeix una penalització.

2n cas. El subministrament supera la demanda, $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$

En aquest cas, anàlogament, afegim un destí fictici D_{n+1} al problema, de manera que la seva demanda i cost per unitat transportada són:

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j.$$
$$c_{i,n+1} = 0, i = 1, \ldots, m.$$

Aquí, el cost per unitat transportada és zero perquè no corresponen a productes reals que estan sent transportats. Tot i així, es pot posar un altre cost si es tenen en consideració altres costs, com el de emmagatzemar els productes.

Teorema 3.3. *Un problema del transport equilibrat sempre té una solució factible.*

Demostració. Considerem la forma estàndard del problema del tranport equilibrat i considerem

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

Aleshores podem veure fàcilment que si $x_{ij} = \frac{a_i b_j}{S}$, es verifiquen les restriccions per $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. A més, és una solució factible ja que $x_{ij} \geq 0$ per a tot $i = 1, 2, \ldots, m$, j =$1, 2, \ldots, n$. □

Veiem que a (3.3) hi ha m equacions corresponents a restriccions d'origen i n equacions corresponents a restriccions de destí: un total de $n + m$. Tanmateix, podem observar que la suma de les equacions d'origen és

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} a_i,$$

i la suma de les equacions de destí és

$$\sum_{j=1}^{n} \sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{n} b_j,$$

Els dos costats esquerres d'aquestes equacions són iguals. Com es formen amb dues combinacions lineals diferents de les equacions originals, resulta que les equacions del sistema original no són independents. Baix la suposició que el sistema està equilibrat, els dos costats drets de les esquacions són iguals. Tenim doncs que les dues equacions són consistents, però també veiem que el sistema original d'equacions és redundant. És a dir, una de les restriccions originals pot eliminar-se sense canviar el conjunt de solucions factibles. El següent teorema amplia i recull el que hem vist fins ara.

Teorema 3.4. *Un problema del transport equilibrat sempre té una solució, però hi ha exactament una restricció d'igualtat redundant. Quan s'elimina qualsevol restricció d'igualtat, el sistema restant de n + m - 1 restriccions d'igualtat és linealment independent.*

Demostració. Ja hem establert anteriorment l'existència d'una solució i d'una redundància. La suma de totes les restriccions d'origen menys la suma de totes les restriccions de destí és igual a zero. D'aquí resulta que qualsevol restricció es pot expressar com una combinació lineal de les altres, per la qual cosa podem eliminar qualsevol restricció.

Suposem que eliminem la darrera equació i que hi hagués una combinació lineal de les equacions restants que fos idènticament zero.

Siguin $\alpha_i, i = 1, 2, \ldots, m$ i $\beta_j, j = 1, 2, \ldots, n-1$ els coeficients d'aquesta combinació. Tenint en compte (3.3), s'observa que cada x_{in} només apareix en la i-èsima equació, ja que hem eliminat la darrera equació. Per tant tenim que $\alpha_i = 0$, per a $i = 1, 2, \ldots, m$.

En les equacions restants, x_{ij} apareix en només una equació. D'aquí podem deduir que $\beta_j = 0$ per a $j = 1, 2, \ldots, n-1$.

Així doncs, l'única combinació lineal que dóna zero és la que té per coeficients tot zeros, i el sistema d'equacions és linealment independent. □

Definició 3.5. *Es diu que una matriu quadrada no singular **M** és triangular si, per una permutació de les seves files i columnes, es pot representar en forma de matriu triangular inferior. Una matriu triangular compleix:*

1. *La matriu té una fila que conté exactament un valor no nul.*
2. *La submatriu, formada eliminant la fila que conté un únic valor no nul i la columna que contenia aquest valor, compleix la propietat (1). Aquest procediment es pot repetir fins que totes les files i columnes s'eliminen.*

Una propietat estructural important del problema del transport és la següent:

Teorema 3.6 (**Teorema de la triangularitat de la base**). *Totes les bases del problema del transport són triangulars.*

Demostració. Considerem el sistema de restriccions (3.3). Es canvia el signe de la meitat superior del sistema. Llavors la matriu **A** dels coeficients del sistema consta de registres que són +1, -1 ó 0.

A partir del resultat del teorema (3.4), suprimim qualsevol de les equacions per a eliminar la redundància. A partir de la matriu de coeficients resultant, formem una base **B** seleccionant un subconjunt no singular de $m + n - 1$ columnes.

Cada columna de **B** conté com a màxim dos registres diferents de zero, un +1 i un -1. Així, com a molt, hi ha $2(m + n - 1)$ registres diferents de zero a la base. Tanmateix, si tota columna conté dos registres diferents de zero, llavors la suma de totes les files seria zero, contradient, la no singularitat de **B**.

Aleshores, almenys una columna de **B** ha de contindre només un registre diferent de zero. Això significa que el nombre total de registres diferents de zero en **B** és menor que $2(m+n-1)$. D'on resulta que ha d'haver una fila amb només un registre diferent de zero, perquè si cada fila tingués dos o més registres diferents de zero, el nombre total seria com a mínim $2(m+n-1)$. Això significa que es compleix el primer pas del procediment per verificar la triangularitat.

Un argument anàleg pot aplicarse a la submatriu de **B** obtinguda eliminant la fila amb l'únic registre diferent de zero i la columna corresponent a aquest registre. Aquesta submatriu també ha de tenir una fila amb un sol registre diferent de zero. Aquest argument es pot continuar fins a establir la triangularitat de **B**. □

A l'hora de resoldre el problema, com al mètode símplex, primer hem de trobar una solució bàsica factible inicial.

Un procés per calcular-la es realitza a una taula amb les mateixes dimensions que la taula de costs, anomenada *taula solució*, on cada posició (i,j) s'associa amb la variable x_{ij}, o sigui, el nombre d'unitats del producte a trasportar de l'origen O_i al destí D_j.

Cada posició (i,j) s'anomena *cel·la* i el conjunt de totes les cel·les representen una solució. Una cel·la buida representa que x_{ij} és zero.

	D_1	D_2	$\cdots$	D_n	Subministrament
O_1	x_{11}	x_{12}	$\cdots$	x_{1n}	a_1
O_2	x_{21}	x_{22}	$\cdots$	x_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
O_m	x_{m1}	x_{m2}	$\cdots$	x_{mn}	a_m
Demanda	b_1	b_2	$\cdots$	b_n	

Un mètode utilitzat per obtindre aquesta solució factible bàsica inicial s'anomena: *regla de la cantonada noroest.* Aquest mètode estableix que donat un problema equilibrat i una taula solució amb totes les cel·les (i,j) buides, si seguim els següents passos troba la solució factible bàsica inicial:

1. *Pas 1.* Comencem a la cel·la de la cantonada superior esquerra, $[(i,j) = (1,1)]$.
2. *Pas 2.* Assignem a la variable x_{ij} la menor quantitat del subministrament restant a_i o de demanda restant b_j i actualitzem aquests valors, restant aquesta quantitat.
3. *Pas 3.* Ens desplacem una cel·la cap a la dreta si encara el subministrament és positiu. En cas contrari, ens desplacem una cel·la cap a baix. E s torna al *Pas 2.*

Existeix la posibilitat que en un pas intermig es compleixin alhora els requeriments de la fila i de la columna corresponents a una cel·la. Llavors, el registre següent serà un zero, aquest zero podrà estar a la fila següent o a la columna següent de la posició on es satisfan alhora els requeriments, el qual indica una solució bàsica degenerada.

Es poden aconseguir diferents solucions factibles bàsiques mitjançant aquest mètode realitzant una permutació de files i columnes de la taula abans d'aplicar el procediment.

També es pot modificar el procés començant a una cel·la arbitrària i tractant les files i les columnes successives en un ordre arbitrari.

Aquest procediment és molt simple i troba una solució bàsica factible inicial pel problema del transport. Tanmateix, els costs per unitat transportada (c_{ij}) no es contemplen en aquest mètode. Ara presentarem una altra manera per obtindre una solució bàsica factible que sí que té en compte els costs, per la qual cosa sol obtenir una solució més propera a la optimal, dit mètode és el que utilitzarem al nostre programa que resol el sistema del transport.

És l'anomenat *mètode d'aproximació de Vogel.* El mètode d'aproximació de Vogel utilitza també la taula solució del problema, i calcula les diferències de files i les diferències de columnes per seleccionar una cel·la. La diferència de fila i la de la columa es defineix:

- DF_i = la diferència entre el més petit i el següent més petit cost unitari c_{ij} que encara està baix consideració a la fila i, $i = 1, 2, \ldots, m$.
- DC_j = la diferència entre el més petit i el següent més petit cost unitari c_{ij} que encara està baix consideració a la columna j, $j = 1, 2, \ldots, n$.

Aquestes diferències ens permeten fer una millor selecció de la cel·la que utilitzarem per emplenar la taula solució. Donat un problema del transport equilibrat, i començant amb la taula solució amb les cel·les buides, el mètode d'aproximació de Vogel consisteix en els següents passos:

1. *Pas 1.* Calculem per cada fila i i per cada columna j considerades, les diferències DF_i i DC_j. Entre les files i columnes cosiderades, trobem la que té la major diferència, la cel·la (i, j) d'aquesta amb el menor cost per unitat serà la que s'omplirà.
2. *Pas 2.* Assignem a la variable x_{ij} la quantitat màxima factible consistent amb les restriccions de la suma de la fila i de la suma de la columna que s'incloguin en aquesta cel·la. (Es complirà almenys un d'aquest requeriments). És a dir,

   $$x_{ij} = \min(a_i, b_j)$$

   Si tenim que a_i és el mínim, aleshores el subministrament de l'origen O_i es torna zero, i la fila i no es considerarà en iteracions futures. La demanda b_j s'actualitza a $b_j - a_i$.

   Anàlogament, si b_j és el mínim, aleshores la demanda del destí D_j es torna zero, i la columna j no es considerarà en iteracions futures. La demanda a_i s'actualitza a $a_i - b_j$.

   Si $a_i = b_j$, aleshores s'actualitzen els valors del subministrament a_i i de la demanda b_j a zero. La fila i i la columna j no es consideraran en iteracions posteriors.
3. *Pas 3.* Tenim dos casos:
   - Si només una fila o una columna està sent considerada, llavors les cel·les restants (i, j), són seleccionades i els subministraments restants són asignats a aquestes. Aturem.
   - Altrament, tornem al *Pas 1.*

Una vegada tenim aquesta solució bàsica factible inicial, ara ens dedicarem a millorar-la iterativament. Per fer-ho, utilitzarem el problema del transport dual, però primerament, anem a veure què és un programa lineal dual.

3.2 Dualitat i continuació de l'algoritme de resolució

La dualitat es defineix amb els dos problemes següents:

$$\begin{aligned} &\textit{Primal} \\ \text{minimitzar } &\mathbf{c^T x} \\ \text{subjecte a } &\mathbf{Ax} \geq \mathbf{b} \\ &\mathbf{x} \geq 0 \end{aligned} \tag{3.5}$$

$$\begin{aligned} &\textit{Dual} \\ \text{maximitzar } &\lambda^{\mathbf{T}}\mathbf{b} \\ \text{subjecte a } &\lambda^{\mathbf{T}}\mathbf{A} \leq \mathbf{c^T} \\ &\lambda \geq 0 \end{aligned} \tag{3.6}$$

Si $\mathbf{A}$ és una matriu de $m \times n$, llavors $\mathbf{x}$ és un vector columna n-dimensional, $\mathbf{b}$ és un vector columna m-dimensional, $\mathbf{c^T}$ és un vector fila n-dimensional i $\lambda^{\mathbf{T}}$ és un vector fila m-dimensional. El vector $\mathbf{x}$ és la variable del problema primal i λ és la variable del problema dual. Els dos problemes es denominen *forma simètrica* de la dualitat.

És important destacar que les funcions dels problemes dual i primal es poden invertir. Així, quan estudiem el procés d'obtenció del dual a partir del primal, l'intercanvi dels vectors de cost i restricció, la transposició dels coeficients de la matriu, la inversió de les desigualtats de restricció i el canvi de minimització a maximització, s'observa que aplicant aquest mateix procés al dual produeix el primal. En altres paraules, si es transforma el dual, multiplicant per menys un la funció objectiu i les restriccions, de manera que tengui l'estructura del primal (expressat en funció de λ), el seu dual corresponent seria equivalent al primal original.

Donat un problema lineal en forma estàndard (2.1), aquest s'expressa de manera equivalent:

$$\begin{aligned} \text{minimitzar } &\mathbf{c^T x} \\ \text{subjecte a } &\mathbf{Ax} \geq \mathbf{b} \\ &-\mathbf{Ax} \geq -\mathbf{b} \\ &\mathbf{x} \geq 0 \end{aligned} \tag{3.7}$$

El qual està en forma primal, però amb la matriu de coeficients $\left[\frac{\mathbf{A}}{-\mathbf{A}}\right]$

Usant un vector dual dividit om $(\mathbf{u}, \mathbf{v})$, el dual corresponent és

$$\begin{aligned} \text{maximitzar } &\mathbf{u^T b} - \mathbf{v^T b} \\ \text{subjecte a } &\mathbf{u^T A} - \mathbf{v^T A} \leq \mathbf{c^T} \\ &\mathbf{u} \geq 0 \\ &\mathbf{v} \geq 0 \end{aligned} \tag{3.8}$$

Fent $\lambda = \mathbf{u} - \mathbf{v}$, es pot simplificar la representació del problema dual per a obtenir els problemes següents:

$$\begin{aligned} &\textit{Primal} \\ \text{minimitzar } &\mathbf{c^T x} \\ \text{subjecte a } &\mathbf{Ax} = \mathbf{b} \\ &\mathbf{x} \geq 0 \end{aligned} \tag{3.9}$$

$$\begin{aligned} &\textit{Dual} \\ \text{maximitzar } &\lambda^{\mathbf{T}} \mathbf{b} \\ \text{subjecte a } &\lambda^{\mathbf{T}} \mathbf{A} \leq \mathbf{c^T} \end{aligned} \tag{3.10}$$

$$\tag{3.11}$$

Aquesta és la *forma asimètrica* de la relació de dualitat. En aquesta forma, el vector dual λ (que no és més que una composició d'$\mathbf{u}$ i $\mathbf{v}$) no està restringit a ser no negatiu.

Lema 3.7 (Lema de la dualitat feble)**.** *Si* $\mathbf{x}$ *i* λ *són factibles per a* (3.9) *i* (3.10) *respectivament, llavors* $\mathbf{c^T x} \geq \lambda^{\mathbf{T}} \mathbf{b}$

Demostració. Es té

$$\lambda^{\mathbf{T}} \mathbf{b} = \lambda^{\mathbf{T}} \mathbf{Ax} \leq \mathbf{c^T x},$$

Essent vàlida la darrera desigualtat, perquè $\mathbf{x} \geq 0$ □

Aquest lema demostra que un vector factible per a un dels problemes dóna lloc a una cota en el valor de l'altre. Els valor associats al primal són majors que els associats al dual. Com el problema primal cerca un mínim, i el dual, un màxim, cadascun tracta d'arribar a l'altre. D'aquí es deriva un corol·lari important.

Corol·lari 3.8. *Si* $\mathbf{x_0}$ *i* λ_0 *són factibles per a* (3.9) *i* (3.10) *respectivament, i si* $\mathbf{c^T x_0} = \lambda_{\mathbf{0}}^{\mathbf{T}} \mathbf{b}$, *llavors* $\mathbf{x_0}$ *i* λ_0 *són optimals per als seus respectius problemes.*

El següent teorema estableix que la inversa de l'anterior corol·lari també és certa

Teorema 3.9 (Teorema de la dualitat de la programació lineal)**.** *Si un dels dos problemes* (3.9) *o* (3.10) *té una solució optimal finita, també la té l'altre, i els valors de les funcions objectiu són iguals. Si un problema té un objectiu no acotat, l'altre no té una solució factible.*

És a dir entre els dos resultats anteriors tenim:

Teorema 3.10. *Siguin* $\mathbf{x}$ *i* λ *solucions factibles de* (3.9) *o* (3.10). *Aleshores* $\mathbf{x}$ *i* λ *són optimals pels seus respectius problemes si i només si compleixen la condició* $(\mathbf{c^T} - \lambda^T \mathbf{A})\mathbf{x} = 0$

Suposem que tenim una solució bàsica optimal $\mathbf{x} = (\mathbf{x_B}, \mathbf{0})$ del problema (3.9) amb la base $\mathbf{B}$ corresponent. Es determinarà una solució del problema dual (3.10) en funció de $\mathbf{B}$.

Teorema 3.11. *Sigui el problema de programació lineal* (3.9) *amb solució factible bàsica optimal corresponent a la base* $\boldsymbol{B}$*. Llavors el vector* λ *que satisfaga* $\lambda^{\mathbf{T}} = \mathbf{c}_{\mathbf{B}}^{\mathbf{T}}\mathbf{B}^{-\mathbf{1}}$ *és una solució optimal per al problema dual* (3.10)*. Els valors optimals de ambdós problemes són iguals.*

Sigui $\mathbf{B}$ la submatriu de la matriu original $\mathbf{A}$ formada per les m columnes d'$\mathbf{A}$ que corresponen a les variables bàsiques. Aquestes columnes són linealment independents i, per tant, les columnes de $\mathbf{B}$ formen una base per E^m. Suposem que $\mathbf{B}$ es compon de les primeres m columnes d'$\mathbf{A}$. Aleshores, al separar $\mathbf{A}$, $\mathbf{x}$ i $\mathbf{c}^{\mathbf{T}}$ com:

$$\begin{aligned} \mathbf{A} &= [\mathbf{B}, \mathbf{D}] \\ \mathbf{x} &= (\mathbf{x}_{\mathbf{B}}, \mathbf{x}_{\mathbf{D}}) \\ \mathbf{c}^{\mathbf{T}} &= (\mathbf{c}_{\mathbf{B}}^{\mathbf{T}}, \mathbf{c}_{\mathbf{D}}^{\mathbf{T}}) \end{aligned}$$

el problema de programació lineal estàndard és

$$\begin{aligned} \text{minimitzar } & \mathbf{c}_{\mathbf{B}}^{\mathbf{T}}\mathbf{x}_{\mathbf{B}} + \mathbf{c}_{\mathbf{D}}^{\mathbf{T}}\mathbf{x}_{\mathbf{D}} \\ \text{subjecte a } & \mathbf{B}\mathbf{x}_{\mathbf{B}} + \mathbf{D}\mathbf{x}_{\mathbf{D}} = \mathbf{b} \\ & \mathbf{x}_{\mathbf{B}} \geq \mathbf{0},\ \mathbf{x}_{\mathbf{D}} \geq \mathbf{0}. \end{aligned} \tag{3.12}$$

La solució bàsica, que també es suposa factible, corresponent a la base $\mathbf{B}$ és $\mathbf{x} = (\mathbf{x}_{\mathbf{B}}, \mathbf{0})$, on $\mathbf{x}_{\mathbf{B}} = \mathbf{B}^{-\mathbf{1}}\mathbf{b}$. La solució bàsica resulta de fer $\mathbf{x}_{\mathbf{D}} = \mathbf{0}$. Tanmateix, per a qualsevol valor de $\mathbf{x}_{\mathbf{D}}$, el valor necessari de $\mathbf{x}_{\mathbf{B}}$ pot calcular-se a partir de (3.12) com segueix:

$$\mathbf{x}_{\mathbf{B}} = \mathbf{B}^{-\mathbf{1}}\mathbf{b} - \mathbf{B}^{-\mathbf{1}}\mathbf{D}\mathbf{x}_{\mathbf{D}} \tag{3.13}$$

i al substituir aquesta expressió a la funció de cost, resulta

$$\begin{aligned} z &= \mathbf{c}_{\mathbf{B}}^{\mathbf{T}}(\mathbf{B}^{-\mathbf{1}}\mathbf{b} - \mathbf{B}^{-\mathbf{1}}\mathbf{D}\mathbf{x}_{\mathbf{D}}) + \mathbf{c}_{\mathbf{D}}^{\mathbf{T}}\mathbf{x}_{\mathbf{D}} \\ &= \mathbf{c}_{\mathbf{B}}^{\mathbf{T}}\mathbf{B}^{-\mathbf{1}}\mathbf{b} + (\mathbf{c}_{\mathbf{D}}^{\mathbf{T}} - \mathbf{c}_{\mathbf{B}}^{\mathbf{T}}\mathbf{B}^{-\mathbf{1}}\mathbf{D}\mathbf{x}_{\mathbf{D}})\mathbf{x}_{\mathbf{D}}, \end{aligned} \tag{3.14}$$

que representa el cost de qualsevol solució de (3.12) en funció de $\mathbf{x}_{\mathbf{D}}$. Sent

$$\mathbf{r}_{\mathbf{D}}^{\mathbf{T}} = \mathbf{c}_{\mathbf{D}}^{\mathbf{T}} - \mathbf{c}_{\mathbf{B}}^{\mathbf{T}}\mathbf{B}^{-\mathbf{1}}\mathbf{D} \tag{3.15}$$

el vector de cost relatiu (per a les variables no bàsiques). En el qual ens basem per a determinar el vector que s'ha d'incorporar a la base o si la solució és òptima.

Ara és possible escriure la taula símplex en forma matricial de la forma:

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{b} \\ \hline \mathbf{c}^{\mathbf{T}} & 0 \end{array}\right] = \left[\begin{array}{c|c|c} \mathbf{B} & \mathbf{D} & \mathbf{b} \\ \hline \mathbf{c}_{\mathbf{B}}^{\mathbf{T}} & \mathbf{c}_{\mathbf{B}}^{\mathbf{T}} & 0 \end{array}\right] \tag{3.16}$$

Si la matriu $\mathbf{B}$ s'utilitza com a base, multiplicant per la matriu $\left[\begin{array}{c|c} \mathbf{B}^{-\mathbf{1}} & \mathbf{0} \\ \hline -\lambda^{\mathbf{T}} & 1 \end{array}\right]$ tendriem

$$\left[\begin{array}{c|c|c} \mathbf{I} & \mathbf{B}^{-\mathbf{1}}\mathbf{D} & \mathbf{B}^{-\mathbf{1}}\mathbf{b} \\ \hline -\lambda^{\mathbf{T}}\mathbf{B} + \mathbf{c}_{\mathbf{B}}^{\mathbf{T}} & -\lambda^{\mathbf{T}}\mathbf{D} + \mathbf{c}_{\mathbf{D}}^{\mathbf{T}} & -\lambda^{T}\mathbf{b} \end{array}\right] \tag{3.17}$$

que correspon a un pas de pivotatge a la taula símplex amb λ^T definit com $\lambda^T = \mathbf{c_B^T B^{-1}}$. El vector λ s'anomena *multiplicador símplex.* Si substituim el seu valor en (3.17) tenim:

$$\left[\begin{array}{c|c|c} \mathbf{I} & \mathbf{B^{-1}D} & \mathbf{B^{-1}b} \\ \hline \mathbf{0} & \mathbf{c_D^T} - \mathbf{c_B^T B^{-1} D} & -\mathbf{c_B^T B^{-1} b} \end{array}\right] \tag{3.18}$$

Observem que $(\mathbf{c^T} - \lambda^T \mathbf{A})\mathbf{x} = (\mathbf{c_B^T} - \lambda^T \mathbf{B})\mathbf{x_B} + (\mathbf{c_D^T} - \lambda^T \mathbf{D})\mathbf{x_D} = 0$ ja que, per la definició de λ, els valors dels coeficients de cost reduits de les variables bàsiques $\mathbf{x_B}$ són 0 i les variables no bàsiques $\mathbf{x_D}$ són 0. Pel teorema 3.10 la solució actual $\mathbf{x}$ és optimal si i només si λ és factible al problema dual. Això és així si $\lambda^T \mathbf{A} \leq \mathbf{c^T}$, que significa que $\lambda^T \mathbf{B} \leq \mathbf{c_B^T}$ i $\lambda^T \mathbf{D} \leq \mathbf{c_D^T}$. Per construcció $\lambda^T \mathbf{B} = \mathbf{c_B^T}$, per tant la primera desigualtat es compleix. Aleshores, la condició necessària i suficient per la optimalitat es redueix a $\mathbf{c_D^T} - \lambda^{\mathbf{T}} \mathbf{D} = \mathbf{c_D^T} - \mathbf{c_B^T B^{-1} D} \geq 0$. És a dir, que els coeficients de cost reduit de les variables no bàsiques $\mathbf{r_D^T}$ han de ser més grans o igual a zero, perquè sigui una solució optimal.

Les solucions optimals dels problemes primal i dual satisfan una relació adicional. Aquesta relació es pot formular per a un parell qualsevol de programes lineals duals, però aquí s'expressa només per a les formes simètriques i asimètriques dels problemes vistes anteriorment.

Teorema 3.12 (Folga complementària (forma asimètrica))**.** *Siguin* $\mathbf{x}$ *i* λ *solucions factibles per als problemes primal i dual, respectivament, en el parell asimètric. Una condició necessària i suficient per a que les dues solucions siguin optimals és que, per a tota i*

1. $x_i > 0 \implies \lambda^{\mathbf{T}} \mathbf{a_i} = c_i$
2. $x_i = 0 \Longleftarrow \lambda^{\mathbf{T}} \mathbf{a_i} < c_i$

Demostració. Si es compleixen les condicions formulades, és evident que $(\lambda^{\mathbf{T}} \mathbf{A} - \mathbf{c^T})\mathbf{x} = 0$. Així $\lambda^{\mathbf{T}} \mathbf{b} = \mathbf{c^T x}$, i pel corol·lari (3.8) les dues solucions són optimals. A la inversa, si les dues solucions són optimals ha de ser cert, pel teorema de dualitat, que $\lambda^{\mathbf{T}} \mathbf{b} = \mathbf{c^T x}$, d'on $((\lambda^{\mathbf{T}} \mathbf{A} - \mathbf{c^T})\mathbf{x} = 0$. Com cada component de $\mathbf{x}$ és no negativa i cada component de $\lambda^{\mathbf{T}} \mathbf{A} - \mathbf{c^T}$ és no positiva, han de donar-se les condicions *1.* i *2.*. $\square$

Teorema 3.13 (Folga complementària (forma simètrica))**.** *Siguin* $\mathbf{x}$ *i* λ *solucions factibles per als problemes primal i dual, respectivament, en el parell simètric. Una condició necessària i suficient per a que les dues solucions siguin optimals és que, per a tota i i j*

1. $x_i > 0 \implies \lambda^{\mathbf{T}} \mathbf{a_i} = c_i$
2. $x_i = 0 \Longleftarrow \lambda^{\mathbf{T}} \mathbf{a_i} < c_i$
3. $\lambda_j > 0 \implies \mathbf{a^j x} = b_j$
4. $\lambda_j = 0 \Longleftarrow \mathbf{a^j x} > b_j$

(on $\mathbf{a^j}$ *és la j-èsima fila d'*$\boldsymbol{A}$*).*

Una vegada vista aquesta teoria, considerem el problema del transport equilibrat estàndard (3.2).

Denotem per $u_1, \dots, u_m$ i $v_1, \dots, v_n$, les variables duals, que formen el problema dual del transport:

$$\begin{aligned} \text{maximitzar } & \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j \\ \text{subjecte a } u_i + v_j \leq c_{ij}, \ & \text{per a } i = 1, 2 \dots, m, \ j = 1, 2 \dots, n \\ & u_i, v_j : \text{sense restriccions.} \end{aligned} \tag{3.19}$$

El mètode utilitzat per calcular la solució optimal per un problema del transport és una adaptació directa del mètode símplex. Començant amb una solució inicial factible bàsica, el mètode calcula una altra amb un menor valor de la funció objectiu. Per fer-ho ens movem d'una base a una altra, per tant, hem de determinar quina variable entra a la base i quina surt. Aquest canvi es basa en el teorema de millora (2.5).

Notem que les variables del model del problema del transport estan denotades per x_{ij}, els costs unitaris de transport per c_{ij} i els vectors columna de la matriu $\mathbf{A}$ per a_{ij}, $i = 1, 2, \dots, m; j = 1, 2, \dots, n$.

Seguint el teorema de millora, hem de calcular els coeficient de cost reduït $r_{ij} = z_{ij} - c_{ij}$ per així decidir si la solució que tenim pot ésser millorada:

$$z_{ij} - c_{ij} = c_B^T B^{-1} a_{ij} - c_{ij}$$

com $c_B^T B^{-1}$ és el vector de les variables duals:

$$c_B^T B^{-1} = (u_1, \dots, u_m, v_1, \dots, v_n) \tag{3.20}$$

llavors,

$$z_{ij} - c_{ij} = (u_1, \dots, u_m, v_1, \dots, v_n) a_{ij} - c_{ij} \tag{3.21}$$

Com a_{ij} consisteix en uns i zeros, i només té dos 1 a la i-èsima i la $(m+j)$-èsima posició, tenim:

$$z_{ij} - c_{ij} = u_i + v_j - c_{ij} \tag{3.22}$$

Per tant aquests coeficients de cost reduït es calculen per mitjà de les variables duals. Com tenim que $z_{ij} - c_{ij} = 0$ per a les variables bàsiques x_{ij}, i com hi ha $m + n - 1$ variables bàsiques, tenim un sistema de $m + n - 1$ equacions del tipus (3.22)

La solució del sistema ens permet saber el valor de les $m + n$ variables duals, aquest sistema es pot resoldre assignant a una variable dual un valor arbitrari.

Una vegada sabem els valors d'aquestes variables duals, els coeficients de cost reduït per les variables no bàsiques es calculen. Tindrem aleshores dos casos:

- Si $z_{ij} - c_{ij} \leq 0$ per a qualsevol $i = 1, \dots, m$, $j = 1, \dots, n$ aleshores la solució és òptima.
- Si hi ha un o més $z_{ij} - c_{ij} > 0$, es pot millorar la solució, i la variable amb el màxim coeficient de cost reduït es selecciona per entrar a la base.

Una vegada seleccionada la variable que entrarà a la base, hem d'identificar la variable que sortirà de la base actual. Però primerament, necessitem saber les següents nocions.

Definició 3.14. *Un **cicle** és una seqüència ordenada d'almenys quatre cel·les que compleixen:*

1. *Dues cel·les consecutives estan o a la mateixa fila o a la mateixa columna.*
2. *No hi ha tres o més cel·les consecutives a la mateixa fila o columna.*
3. *La darrera cel·la a la seqüència té la fila o la columna en comú amb la primera cel·la de la seqüència.*

Teorema 3.15. *En un problema del transport equilibrat, un conjunt de $m+n-1$ variables és bàsic si i només si les cel·les corresponents a la taula de transport no contenen cicles.*

Definició 3.16. *Un subconjunt de cel·les és un θ-cicle si valors de $+\theta$ i $-\theta$ son posats alternativament dins les cel·les del cicle, de tal manera que si una columna o una fila conté una cel·la del cicle amb un valor $+\theta$, també conté una cel·la amb un valor $-\theta$.*

Definició 3.17. *Un conjunt de cel·les de la taula de transport és un conjunt minimal linealment dependent si i només si:*

1. *És linealment dependent.*
2. *Cap subconjunt és linealment dependent.*

Per la definició de θ-cicle, és clar que un θ-cicle és un conjunt minimal linealment dependent. El següent teorema, defineix el criteri per trobar l'únic θ-cicle.

Teorema 3.18 (θ-cicle en $\mathbf{B} \cup$ (p,q))**.** *Suposem que $\boldsymbol{B}$ és un conjunt de m + n -1 cel·les bàsiques de la taula $m \times n$ de transport i (p,q) una cel·la no bàsica. Llavors, el conjunt $\boldsymbol{B} \cup$ (p,q) conté exactament un θ-cicle i aquest θ-cicle conté una cel·la no bàsica.*

Demostració. Com $\mathbf{B}$ és un conjunt bàsic, $\mathbf{B}$ és linealment independent, i per tant, no pot contenir un θ-cicle. Llavors si hi ha un θ-cicle a $\mathbf{B} \cup$ (p,q), ha de contenir (p,q). Com el rang d'$\mathbf{A}$ pel teorema (3.4), és $m+n-1$, cap subconjunt de $m+n$ cel·les és linealment independent. Aleshores, $\mathbf{B} \cup$ (p,q) és linealment dependent, i per tant, conté almenys un θ-cicle. D'Àlgebra Lineal, sabem que un conjunt que conté una base i exactament un vector que no pertany a la base, conté un únic conjunt minimal linealment dependent. Per tant, el conjunt de vectors contingut en $\mathbf{B} \cup$ (p,q), conté exactament un θ-cicle. □

Els θ-cicles poden ser usats per canviar la base, afegint la variable no bàsica i eliminant una de la base, com es veu al següent teorema.

Teorema 3.19. *Sigui $\boldsymbol{B}$ una base d'$\boldsymbol{A}$ (ignorant una fila) i sigui una altra columna $\mathbf{d}$ corresponent a una variable no bàsica què entra a la base. Llavors el vector $\mathbf{y} = \mathbf{B^{-1}d}$ donarà els canvis a les actuals variables bàsiques quan entri la nova variable. Les components del vector $\mathbf{y}$ són 0, +1 o -1.*

Demostració. Sigui $\mathbf{y}$ la solució de l'equació $\mathbf{By} = \mathbf{d}$. Aleshores $\mathbf{y}$ és la representació de $\mathbf{d}$ en funció de la base. Aquesta equació pot ser resolta per la regla de Cramer com segueix:

$$\mathbf{y_k} = \frac{\det(\mathbf{B_k})}{\det(\mathbf{B})}$$

On $\mathbf{B_k}$ és es la matriu obtinguda substituint la k-èsima columna de $\mathbf{B}$ per $\mathbf{d}$. $\mathbf{B}$ i $\mathbf{B_k}$ són submatrius de la matriu de restriccions $\mathbf{A}$ original. La matriu $\mathbf{B}$ pot expresar-se en forma triangular amb tots els elements de la diagonal iguals a +1, amb combinacions i intercanvis de files i columnes. Per tant, $\det(B) = +1$ ó -1. Com qualsevol submatriu quadrada d'$\mathbf{A}$ pot contenir només entrades de 0 o 1 amb un màxim de dos 1s per columna, per com és $\mathbf{A}$, tot determinant de qualsevol submatriu d'$\mathbf{A}$ tindrà un valor de +1, -1 ó 0, per tant, $\det(B_k) = 0$, +1, ó -1. Aleshores, $y_k = 0$, +1 ó -1. □

El resultat anterior implica que quan s'afegeix una nova variable a la solució a nivell unitari, les variable bàsiques actuals canviaran en +1, -1 ó 0. Si la nova variable té un valor θ, llavors les variables bàsiques canvien en $+\theta$, $-\theta$ ó 0.

L'únic θ-cicle es troba procedint com segueix: La cel·la corresponent a la variable entrant es marca amb un signe +, per indicar que se li donarà un valor positiu (θ) a la variable entrant. Després, deixem de considerar les columnes i les files amb només una cel·la bàsica, considerant que la que està marcada amb un + ho és. Primer eliminem les files, per exemple, i després les columnes, després les files i així successivament. Al final del procés, les cel·les bàsiques que no han estat eliminades, així com la marcada, formen l'únic θ-cicle que busquem.

Per resumir, per canviar la base, havent elegit la variable no bàsica x_{pq} que entrarà a la base. Trobem el θ-cicle al conjunt $\mathbf{B} \cup (p, q)$. Asignem $+\theta$ a la cel·la (p,q) i valors $-\theta$ i $+\theta$ alternativament recorrent les cel·les del cicle, de manera que al final tindrem a cada fila i columna del cicle un valor $-\theta$ i un valor $+\theta$. Sigui $\tilde{x}$ la representació de la solució factible bàsica actual. Llavors, els valors de la nova solució factible bàsica són $x_{ij} = \tilde{x}_{ij} + \theta$, $\tilde{x}_{ij}$ - θ o $\tilde{x}_{ij}$. Totes les altres variables no bàsiques tenen valor zero. Ara, determinem el valor de θ, aquest es fixa de manera que porti una de les variables bàsiques anteriors a zero. D'entre les variables marcades amb un menys, les quals són les que disminuiran el seu valor al introduir x_{pq} a la base, igualem θ a la menor magnitud d'aquestes variables. Actualitzem la taula amb els canvis comentats, i obtenim una nova solució factible bàsica.

3.3 Descripció del programa que resol el problema del transport.

El programa que resol el problema del transport està desenvolupat amb el llenguatge C.

La creació d'aquest programa ha cercat veure a la pràctica la resolució del problema del transport.

Aquest algoritme comença demanant a l'usuari que introudeixi el nombre d'origens, m, i el nombre de destins, n, els quals han de ser positius.

Posteriorment, es generen aleatòriament el nombre d'unitats de cada destí i cada origen. Cada un tindrà un subministrament o una demanda d'un nombre d'unitats a l'interval $[100, 999]$.

Llavors, si escau, s'equilibra el problema, i es crea un origen fictici si la demanda supera al subministrament, o un destí fictici si el subministrament supera la demanda. Posteriorment, s'inicialitza la matriu de costs on el valor de cada cost c_{ij} estarà a l'interval $[10, 100]$, excepte si s'ha equilibrat el problema. En aquest cas, els costos des de l'origen fictici o cap al destí fictici seran zero.

A continuació, després d'assignar la memòria necessària per desar les variables x_{ij} i les variables duals, es troba la solució factible inicial amb el mètode d'aproximació de Vogel.

Després, mentre no es trobi una solució degenerada, es repeteix el següent bucle:

1. Es calculen els valors de les variables duals u's i v's.
2. Es cerca la variable que entrarà a la solució
3. Es troba el θ-cicle i s'introdueix la variable a la solució, i surt de la base una altra variable.

Si es troba una solució degenerada, el nostre programa s'acaba, i indica que s'ha degenerat la solució.

Altrament, es troba la solució optimal del problema i apareix en pantalla el cost total minimitzat.

4 Problema del flux de cost mínim i xarxes.

Ara comencem l'estudi d'una branca diferent en la programació lineal: la de grafs i fluxes en xarxes. Tanmateix, veurem que ofereix una visió diferent per al problema del transport. Primerament, veurem la part de la terminologia i conceptes bàsics de grafs i xarxes necessaris per a la resolució dels problemes. Ens centrarem en els grafs dirigits.

Definició 4.1. Un *graf dirigit* G consta d'un conjunt finit d'elements anomenats *nodes* $N = \{1, 2, \ldots, n\}$ i d'un subconjunt de parells ordenats de nodes anomenats *arcs* $S = \{(i, j), (k, l), \ldots (s, t)\}$.

Com veiem, els nodes solen estar numerats. Veiem un arc entre dos nodes i i j es representa per un parell ordenat de nodes $(i,\ j)$ i es diu que l'arc és *incident* amb els nodes i i j i està dirigit del node i al node j. A la respresentació d'un graf dirigit els nodes estan representats per cercles, amb un nombre al seu interior, representant el nombre del node; i els arcs $(i,\ j)$ es representen com una fletxa que va d'i a j. A un graf dirigit es pot tenir que dos nodes i i j poden estar conectats en ambdues direccions, amb els arcs (i,j) i (j,i).

Hi ha altres definicions útils per descriure l'estructura d'un graf dirigit. Un *camí* (del node i_0 al node i_p) és una successió d'arcs $P = \{(i_0, i_1), (i_1, i_2), \ldots, (i_{p-1}, i_p)\}$ en els que el node inicial de cada arc és el mateix que el node terminal de l'arc que el precedeix a la successió. Així doncs, cada arc al camí està dirigit "cap a" i_p i "cap a fora" de i_0.

Una *cadena* és una estructura similar a un camí amb la diferència que no tots els arcs estan dirigits cap a i_p.

Un *circuit* és un camí en la que $i_0 = i_p$, per tant, un circuit és un camí tancat. Un *cicle* és una cadena tancada. Aleshores tenim que tot camí és una cadena, però no recíprocament i que cada circuit és un cicle però no inversament.

Definició 4.2. Diem que un graf dirigit G és *connex* si existeix una cadena de cada node de G a qualsevol altre node de G.

Un *arbre* és un graf connex que no conté cicles. Un *arbre d'expansió* és un arbre que conté tots els nodes del graf. Considerant un arbre dins d'un graf G, és fàcil comprovar que G és connex si, i només si, conté un arbre d'expansió.

A part de la representació visual d'un graf dirigit, un altre mètode comú de representació és en funció d'una matriu d'incidència node-arc del graf, que es construeix relacionant verticalment els nodes, i horitzontalment, els arcs. Llavors, en la columna baix l'arc $(i,\ j)$ es coloca +1 a la posició (fila) corresponent al node i i -1 a la corresponent al node j.

Quan existeix la possibilitat de *flux* a través dels arcs, el graf dirigit s'anomena *xarxa*. En algunes aplicacions, la xarxa pot representar un sistema de transport, una xarxa de comunicació o simplement una representació utiltzada per visualitzar un problema matemàtic.

Un *flux* en un arc dirigit $(i,\ j)$ donat és un nombre $x_{ij} \geq 0$. Els fluxes en els arcs de la xarxa han de satisfer conjuntament un criteri de conservació a cada node: el flux no pot crear-se o eliminar-se en un node; el flux total que entra a un node, ha d'ésser igual al flux total que surt del node, a menys que el node sigui una *font* o un *embornal*, segons veurem a continuació. D'aquesta manera a cada node i d'una xarxa dirigida amb n nodes tenim:

$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = 0$$

El primer sumatori és el flux total "cap a fora de" i, i el segon, el flux total "cap a" i. (Òbviament, si tenim que no hi ha arc entre i i j, x_{ij} no existeix). Per a que existeixin fluxes diferents de zero en una xarxa sense fonts ni embornals, aquesta ha de contenir un cicle.

En molts de problemes, uns nodes es designen com a fonts, i altres com embornals (també anomenats nodes d'oferta i nodes de demanda, respectivament). El flux net *que surt* d'una font pot ser positiu, anàlogament, el flux net *que entra* a un embornal pot ser positiu.

Una vegada vists aquests conceptes previs, anem a veure el problema del flux del cost mínim, que generalitza el problema del transport. Considerem una xarxa G amb n nodes. Per a cada node i en G hi ha un nombre b_i que representa l'*oferta* o els recursos disponibles d'un article al node (si $b_i > 0$) o la *demanda* requerida de l'article al node (si $b_i < 0$). A vegades, els nodes i amb $b_i > 0$ s'anomenen *orígens* i els nodes amb $b_i < 0$ s'anomenen *destins*. Si $b_i = 0$, aleshores cap article està disponible al node i i cap es requereix; en aquest cas es sol denominar i com node *intermedi* (o de *transbord*). Com ja hem dit, al ser un problema de flux de cost mínim, associat amb cada arc (i, j) es tenen el nombre x_{ij}, el qual representa la quantitat de flux sobre l'arc (es suposa que $x_{ij} \geq 0$); i el nombre c_{ij}, que és el cost unitari de transport al llarg de l'arc. Es suposarà que la xarxa està *equilibrada*, o sigui, que l'oferta total a G és igual a la demanda total, aleshores tindrem:

$$\sum_{i=1}^{n} b_i = 0$$

Així doncs, el problema del flux de cost minimal a una xarxa G amb n nodes i m arcs, consisteix en determinar fluxes $x_{ij} \geq 0$ a cada arc de la xarxa, per a que el flux net que entra a cada node i sigui b_i, minimitzant el cost total. Expressat com un programa lineal tenim:

$$\begin{aligned} \text{minimitzar} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{subjecte a} \quad \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} &= b_i,\ i = 1, 2, \ldots, n \qquad (4.1) \\ x_{ij} &\geq 0,\ i, j = 1, 2, \ldots, n \end{aligned}$$

Les restriccions (4.1) s'anomenen equacions de *conservació de flux* o equacions de *Kirchoff* i indiquen que, a la xarxa, no es pot crear ni destruir flux. Aquestes equacions requereixen que el flux net que surt del node i, o sigui, $\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki}$ sigui igual a b_i.

Un problema d'aquest tipus es pot originar, per exemple, a una xarxa logística en la qual persones y matèria es mouen entre diversos punts de la terra, o també hi ha problemes de flux amb cost mínim quan s'analitzen i es disenyen sistemes de comunicació, oleoductes, entre d'altres.

Una manera de resoldre aquest problema, és utilitzar l'algoritme símplex però veurem més endavant una altra resolució ja que ja hem estudiat aquesta amb anterioritat. El problema del transport que hem vist és un cas especial d'aquest problema, corresponent a una xarxa amb arcs que van dels nodes d'oferta als de demanda. Al problema del transport, els enviaments estan restringits a anar directament als nodes de demanda mentre que al de flux de cost minimal el flux des d'un node d'oferta pot passar per diversos nodes intermedis o de transbord fins a arribar als nodes de demanda. Per això aquest problema a vegades es denomina també *problema del transbord*.

Abans d'estudiar l'estructura del problema de flux de cost mínim, anem a veure les condicions de Kuhn-Tucker lineals en les quals es basa el nostre algoritme.

Considerem el següent problema de programació lineal:

$$\begin{aligned} \text{minmitzar } & \mathbf{c^T x} \\ \text{subjecte a } & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

on $\mathbf{c^T}$ és un n-vector, $\mathbf{b}$ és un m-vector, i $\mathbf{A}$ és una matriu $m \times n$. Les condicions Kuhn-Tucker es poden enunciar com segueix. El vector $\mathbf{x}$ és una solució òptima del problema anterior si existeix un n-vector $\mathbf{v}$ i un m-vector $\mathbf{w}$ tals que satisfacin les tres condicions següents. Recíprocament, si les tres condicions següents es satisfan, aleshores $\mathbf{x}$ és una solució òptima del problema anterior:

$$\mathbf{Ax} \geq \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0} \tag{4.2}$$

$$\mathbf{c^T} - \mathbf{w^T A} - \mathbf{v^T} = \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}, \quad \mathbf{v} \geq \mathbf{0} \tag{4.3}$$

$$\mathbf{w^T}(\mathbf{Ax} - \mathbf{b}) = \mathbf{0}, \qquad \mathbf{v^T x} = \mathbf{0} \tag{4.4}$$

Abans de demostrar-ho, anem a veure què signifiquen aquestes tres condicions d'optimalitat. La primera condició (4.2) estableix simplement que el punt candidat ha d'ésser factible, és a dir, ha de satisfer les restriccions del problema. Aquesta condició es coneix usualment com *factibilitat primal*. La segona condició (4.3), es coneix normalment com *factibilitat dual*, doncs correspon a la factibilitat del problema dual. Aquí $\mathbf{w}$ i $\mathbf{v}$ es coneixen com *multiplicadors de Lagrange* o *variables duals* corresponents a les restriccions $\mathbf{Ax} \geq \mathbf{b}$ i $\mathbf{x} \geq \mathbf{0}$, respectivament. Darrerament, la tercera condició (4.4) es sol conèixer com *folga complementària*. Donat que $\mathbf{w^T} \geq \mathbf{0}$ i $\mathbf{Ax} \geq \mathbf{b}$, llavors $\mathbf{w^T}(\mathbf{Ax} - \mathbf{b}) = \mathbf{0}$ si, i només si, ó $\mathbf{w_i}$ és 0, ó bé, la i-èsima variable de folga és 0. D'igual manera, $\mathbf{v^T x} = \mathbf{0}$ si, i només si, la coordenada $\mathbf{x_j}$ és nul·la, ó bé, $\mathbf{v_j} = 0$

Anem a demosrar primer que les condicions de Kuhn-Tucker són suficients per a tenir l'optimalitat. Suposem que $\mathbf{x}$ és una solució factible del problema i que existeixen vectors $\mathbf{w}$ i $\mathbf{v}$ tals que les condicions (4.2), (4.3) i (4.4) es compleixen. Veurem que, de fet, $\mathbf{x}$ és una solució òptima. Sigui $\mathbf{x}'$ qualsevol punt factible que satisfaci $\mathbf{Ax}' \geq \mathbf{b}$ i $\mathbf{x}' \geq \mathbf{0}$. De l'equació (4.3), $\mathbf{c^T} - \mathbf{w^T A} - \mathbf{v^T} = \mathbf{0}$ i s'obté:

$$0 = (\mathbf{c^T} - \mathbf{w^T A} - \mathbf{v^T})(\mathbf{x} - \mathbf{x}') = (\mathbf{c^T x} - \mathbf{c^T x}') - \mathbf{w^T A x} - \mathbf{v^T x} + \mathbf{w^T A x}' + \mathbf{v^T x}'$$

De la condició (4.4), $\mathbf{w^T A x} = \mathbf{w^T b}$ i $\mathbf{v^T x} = 0$. Per tant, donat que

$$0 = (\mathbf{c^T x} - \mathbf{c^T x}') + \mathbf{w^T}(\mathbf{Ax}' - \mathbf{b}) + \mathbf{v^T x}' \tag{4.5}$$

$\mathbf{w} \geq 0$, $\mathbf{Ax}' - \mathbf{b} \geq 0$, $\mathbf{v} \geq 0$ i $\mathbf{x}' \geq 0$, l'equació anterior implica que $\mathbf{c^T x}' \geq \mathbf{c^T x}$. Donat que això es compleix per a qualsevol solució factible, llavors $\mathbf{x}$ és, en efecte, una solució òptima al problema. Això demostra que les condicions de Kuhn-Tucker són suficients per a l'optimalitat.

Per veure la demostració que les condicions (4.2), (4.3) i (4.4) són necessàries, es pot acudir a [2] de la bibliografia.

Ara anem a estudiar en profunditat l'estructura d'aquest problema. Com hem comentat amb anterioritat, la matriu de coeficients $\mathbf{A}$ de les restriccions de flux és la matriu d'incidència node-arc de la xarxa. La matriu $\mathbf{A}$ té una fila per cada node de la xarxa i una columna per cada arc, $\mathbf{A}$ presenta per la columna respectiva a l'arc (i, j) un registre $+1$ a la fila i i un registre -1 a la fila j, i tots els elements restants són zero. Per tant, les columnes d'$\mathbf{A}$ estan donades per

$$\mathbf{a_{ij}} = \mathbf{e_i} - \mathbf{e_j} \tag{4.6}$$

On $\mathbf{e_i}$ i $\mathbf{e_j}$ són vectors unitaris en E^n, amb 1's en la i-èsima i la j-èsima posicions respectivament.

Com tenim que la suma de totes les files és el vector zero, la matriu $\mathbf{A}$ té com a màxim un rang de $n-1$ i qualsevol fila d'$\mathbf{A}$ pot eliminar-se per obtenir una matriu de coeficients del mateix rang que l'original. Ara es demostrarà, utilitzant els conceptes de grafs i xarxes vists més amunt, que el rang de la matriu de coeficients és realment $n-1$ partint d'una suposició de connectivitat simple a la xarxa.

Per a formular la suposició necessària de manera precisa, es defineix el graf H no dirigit de la xarxa. Cada arc de la xarxa s'inclou en H, independentment de la seva direcció. No considerem l'orientació dels arcs ja que només ens interessen les propietats lineals d'$\mathbf{A}$. S'ha de suposar que H és connex, que implica que aquest graf no dirigit conté almenys un arbre d'expansió.

Per a continuar amb la demostració que el rang d'$\mathbf{A}$ és $n-1$, es selecciona qualsevol fila aleatòriament per a eliminar-la d'$\mathbf{A}$, la matriu resultant d'aquesta eliminació la denotem $\mathbf{A}^{'}$. Considerem qualsevol arbre d'expansió $\mathbf{T}$ del graf H. Aquest arbre constarà de $n-1$ arcs sense cicle. El node corresponent a la fila que hem eliminat d'$\mathbf{A}$ s'expressa com l'*arrel* de l'arbre. Sigui $\mathbf{A_T}$ la submatriu $(n-1)\times(n-1)$ d'$\mathbf{A}^{'}$, composada per les $n-1$ columnes corresponents als arcs de l'arbre. Almenys dos nodes de l'arbre, han de tenir un únic arc de $\mathbf{T}$ que els uneixi, i almenys un d'ells no és l'arrel. Això vol dir que la fila corresponent d'$\mathbf{A_T}$ té només un registre diferent de zero.

Si ara eliminem la fila i la columna corresponents a aquest registre. Que en termes de l'arbre, correspon a eliminar dit node i l'arc incident en aquest. Els $n-2$ arcs restants de $\mathbf{T}$ formen un arbre per a la xarxa reduïda de $n-1$ nodes, incloent l'arrel. Per tant, el procediment pot repetir-se successivament, eliminant tots els nodes, excepte l'arrel, fins que desapareguin totes les files d'$\mathbf{A_T}$.

Aquest procediment equival al procés de triangulació que hem vist anteriorment. Per tant, $\mathbf{A_T}$ és una matriu triangular no singular de $(n-1)\times(n-1)$. Aleshores $\mathbf{A}$ té rang igual $n-1$.

Anem a veure ara com és l'estructura d'una base del problema del flux de cost mínim. Ja hem demostrat que un arbre d'expansió de G correspon a una base, doncs defineix una submatriu $\mathbf{A}^{'}$ no singular. Ara demostrarem la inversa.

Una base correspon a elegir de $n-1$ columnes linealment independents d'$\mathbf{A}$. Cada columna correspon a un arc de la xarxa, de manera que una elecció d'una base equival a una tria de $n-1$ arcs. Es vol demostrar que aquests arcs han de formar necessàriament un arbre d'expansió. Suposem que el conjunt d'arcs corresponents a la base conté un cicle que consta de m arcs. Quan es disposen en forma de cicle, els arcs tenen la següent forma $(i_1, i_2)(i_2, i_3)(i_3, i_4)\ldots(i_m, i_1)$. Respectant aquest ordre, alguns arcs poden mantenir la seva orientació original i altres poden aparèixer amb el seu sentit invertit. Considerem ara les columnes corresponents a aquests arcs $\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}, \ldots, \mathbf{a_m}$ d'$\mathbf{A}$. Efectuem ara la combinació lineal $\pm\mathbf{a_1} \pm \mathbf{a_2} \pm \mathbf{a_3} \ldots \pm \mathbf{a_m}$, on, en cada cas, el coeficient és $+$ si l'orientació de l'arc és la mateixa en el cicle que en el graf original, i $-$, altrament. El j-èssim vector columna d'aquesta combinació (després de tenir en compte el signe del coeficient) correspon a l'arc (i_j, i_{j+1}) del cicle i té +1 a la fila corresponent a i_j i -1 a la fila corresponent a i_{j+1}. Com a resultat, tots els +1 i -1 es neutralitzen a la combinació. D'aquesta manera, la combinació és el vector zero, el que contradiu a la independència lineal de les columnes $\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}, \ldots, \mathbf{a_m}$. Com hi ha $n-1$ arcs i n nodes, concluim que els arcs han de formar un arbre d'expansió.

Aleshores, es conclou que hi ha una correspondència biunívoca entre els arcs (columnes) d'una base i els arbres d'expansió. Com hem vist, una base és triangular; per tant, les característiques del problema del transport es conserven al problema més general del flux de cost mínim.

Podríem estudiar el mètode símplex per al problema de cost mínim de flux en xarxes, però al ser força similar al que ja hem vist pel problema del transport, estudiarem un altre algoritme, l'anomenat *algoritme de les desviacions*.

4.1 Algoritme de les desviacions

Per conveniència, la forma del problema de flux amb cost mínim amb la qual treballarà l'algoritme és la següent:

$$\begin{aligned} &\text{minimitzar } \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subjecte a } &\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = 0 \quad i = 1, \ldots, n \\ &x_{ij} \geq l_{ij} \quad i, j = 1, \ldots, n \\ &x_{ij} \leq u_{ij} \quad i, j = 1, \ldots, n \end{aligned} \tag{4.7}$$

on s'entèn que les sumes i desigualtats es prenen únicament sobre els arcs existents. Anomenarem *flux circulatori* a qualsevol flux, (o sigui, qualsevol secció dels x_{ij} que satisfà les restriccions (4.7). Un flux circulatori que satisfà les restriccions restants $l_{ij} \leq x_{ij} \leq u_{ij}$ és un *flux factible*. Suposarem que c_{ij}, l_{ij}, i u_{ij} són enters i que $0 \leq l_{ij} \leq u_{ij}$.

Ara anem a veure el dual d'un problema de flux d'una xarxa amb aquestes restriccions i les seves propietats.

Si associem una variable dual w_i amb cada equació de consevació de flux dels nodes (4.7), una variable dual h_{ij} amb cada restricció $x_{ij} \leq u_{ij}$ i una variable dual v_{ij} amb la restricció $x_{ij} \geq l_{ij}$, aleshores el dual del programa lineal anterior està donat per:

$$\begin{aligned} \text{maximitzar } &\sum_{i=1}^{n}\sum_{j=1}^{n} l_{ij}v_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} u_{ij}h_{ij} \\ \text{subjecte a } &w_i - w_j + v_{ij} - h_{ij} = c_{ij} \quad i, j = 1, \ldots, n \\ &h_{ij}, v_{ij} \geq 0 \quad i, j = 1, \ldots, n \\ &w_i \text{ no restringida } i = 1, \ldots, n \end{aligned}$$

on, recordem, les sumes i les restriccions es prenen sobre els arcs existents. El problema dual presenta una estructura interessant. Suposem que se selecciona qualsevol conjunt de les w_i (suposant que w_i són nombres enters). Llavors la restricció dual per a l'arc $(i,\ j)$ és:

$$v_{ij} - h_{ij} = c_{ij} - w_i + w_j, \quad h_{ij} \geq 0, \quad v_{ij} \geq 0$$

i pot ser satisfeta per

$$\begin{aligned} v_{ij} &= \max\{0, \quad c_{ij} - w_i + w_j\} \\ h_{ij} &= \max\{0, -(c_{ij} - w_i + w_j)\} \end{aligned}$$

Per tant, donat qualsevol conjunt de les w_i, el problema dual sempre té una solució factible. De fet, les seleccions anteriors de v_{ij} i h_{ij} donen els valors òptims de v_{ij} i h_{ij} per a un conjunt fixat de les w_i, com veurem a continuació.

Anem a veure les condicions de folga complementària per a l'optimalitat (Kuhn-Tucker) de la formulació de desviacions, són les següents:

$$(x_{ij} - l_{ij})v_{ij} = 0 \quad i, j = 1, 2, \ldots, n \tag{4.8}$$

$$(u_{ij} - x_{ij})h_{ij} = 0 \quad i, j = 1, 2, \ldots, n \tag{4.9}$$

Definim $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$. Llavors, per la definició de v_{ij} i h_{ij} es té

$$v_{ij} = \max\{0, -(z_{ij} - c_{ij})\} \tag{4.10}$$

$$h_{ij} = \max\{0, z_{ij} - c_{ij}\} \tag{4.11}$$

Donat un conjunt de les w_i es poden calcular $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$. De les equacions (4.10) i (4.11), es veu que les condicions de folga complementària (4.8) i (4.9) es compleixen només si

$$\begin{aligned} z_{ij} - c_{ij} < 0 \implies v_{ij} > 0 \implies x_{ij} = l_{ij} \quad i, j = 1, 2, \ldots, n \\ z_{ij} - c_{ij} > 0 \implies h_{ij} > 0 \implies x_{ij} = u_{ij} \quad i, j = 1, 2, \ldots, n \end{aligned}$$

Incloem la condició adicional òbvia

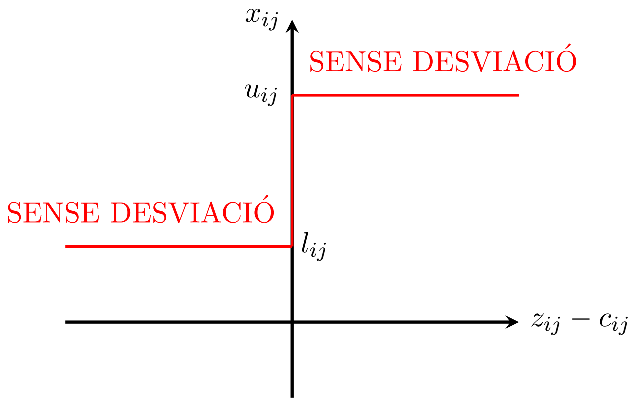
$$z_{ij} - c_{ij} = 0 \implies l_{ij} \leq x_{ij} \leq u_{ij} \quad i, j = 1, 2, \ldots, n$$

Qualsevol flux circulatori que satisfaci les tres condicions anteriors serà òptim. Aleshores, el problema consisteix en cercar entre els valors de les w_i i entre les x_{ij} fins que les tres condicions anteriors es satisfacin.

Diem que un arc $(i,\ j)$ és *sense desviació* si concorda amb alguna de les tres condicions anteriors i diem que és *amb desviació* altrament, per exemple, si tenim que $z_{ij} - c_{ij} > 0$ i $x_{ij} < u_{ij}$, llavors tindriem que l'arc $(i,\ j)$ és amb desviació o que té desviació. D'aquí ve el nom de l'algoritme de les *desviacions*. El que fa aquest algoritme es modificar els fluxes o les w_i de manera que van minvant aquestes desviacions. Es diu que quan els arcs estan sense desviació, han adquirit la *conformabilitat*.

Durant la part primal de l'algoritme de desviacions es canviaran els x_{ij}, intentant eliminar la desviació dels arcs. Durant la fase dual, es canvien les w_i per assolir la conformabilitat.

	$z_{ij} - c_{ij} < 0$	$z_{ij} - c_{ij} = 0$	$z_{ij} - c_{ij} > 0$
$x_{ij} > u_{ij}$	DESVIACIÓ	DESVIACIÓ	DESVIACIÓ
$x_{ij} = u_{ij}$	DESVIACIÓ	**NO DESVIACIÓ**	**NO DESVIACIÓ**
$l_{ij} < x_{ij} < u_{ij}$	DESVIACIÓ	**NO DESVIACIÓ**	DESVIACIÓ
$x_{ij} = l_{ij}$	**NO DESVIACIÓ**	**NO DESVIACIÓ**	DESVIACIÓ
$x_{ij} < l_{ij}$	DESVIACIÓ	DESVIACIÓ	DESVIACIÓ



Notem que un arc és sense desviació si $l_{ij} \leq x_{ij} \leq u_{ij}$ i les condicions (4.8) i (4.9) es satisfan. A mesura que es canvia el flux sobre l'arc (i, j), el punt es mou cap a amunt o cap a avall sobre la vertical específica a la gràfica, depenent si augmenta o disminueix el valor de x_{ij}. Quan canviem w_i ó w_j, el punt es mou a l'esquerra o a la dreta a una horitzontal determinada. Per a assegurar que l'algoritme convergirà, és necessària una mesura de "distància"a l'optimalitat. Si es construeix un algoritme que periòdicament (en intervals finits) redueixi la distància a l'optimalitat, llavors aquest algoritme convergirà finalment a una solució òptima, això ho veurem més endavant.

Hi ha moltes mesures de distància diferents per aquest algoritme. Nosaltres utilitzarem l'anomenada desviació K_{ij}: per a un arc (i, j) la distància es defineix com el mínim canvi de flux requerit sobre l'arc per a que l'arc sigui sense desviació. Com es mostra a la següent taula:

	$z_{ij} - c_{ij} < 0$	$z_{ij} - c_{ij} = 0$	$z_{ij} - c_{ij} > 0$
$x_{ij} > u_{ij}$	$\lvert x_{ij} - l_{ij} \rvert$	$\lvert x_{ij} - u_{ij} \rvert$	$\lvert x_{ij} - u_{ij} \rvert$
$x_{ij} = u_{ij}$	$\lvert x_{ij} - l_{ij} \rvert$	**0**	**0**
$l_{ij} < x_{ij} < u_{ij}$	$\lvert x_{ij} - l_{ij} \rvert$	**0**	$\lvert x_{ij} - u_{ij} \rvert$
$x_{ij} = l_{ij}$	**0**	**0**	$\lvert x_{ij} - u_{ij} \rvert$
$x_{ij} < l_{ij}$	$\lvert x_{ij} - l_{ij} \rvert$	$\lvert x_{ij} - l_{ij} \rvert$	$\lvert x_{ij} - u_{ij} \rvert$

Tenim que la desviació K_{ij} d'un arc és sempre no negativa, sent zero quan l'arc és sense desviació. Observem també com es mesura la desviació i quan s'adquireix la conformabilitat depenent del valor de $z_{ij} - c_{ij}$.

Els passos generals de l'algoritme de les desviacions són els següents:

1. Es comença amb un flux circulatori, per exemple cada $x_{ij} = 0$, i una solució factible pel dual, per exemple cada $w_i = 0$, amb v_{ij} i h_{ij}definits com a les equacions (4.10) i (4.11). S'identifiquen els estats conformables i es calculen les desviacions.

2. Si la xarxa té un arc amb desviació, es fa una fase primal de l'algoritme. Durant aquesta fase es selecciona un arc amb desviació i s'intenta construir un nou flux circulatori de tal manera que no augmenti la desviació de cap arc i que disminueixi la desviació de l'arc seleccionat.

3. Quan s'ha determinat que no es pot construir un flux millor a la fase primal, l'algoritme construeix una nova solució dual de forma que no augmenti cap desviació i es repeteix el pas 2.

4. Iterant els pasos 2 i 3, l'algoritme construirà una solució òptima, o determinarà que no existeix una solució factible.

Durant la fase primal, l'algoritme de desviacions tracta de minvar la desviació d'un arc que no ha adquirit la conformabilitat canviant els fluxes de tal manera que no augmenti la desviació de cap arc. Per exemple, si tenim l'estat sense conformabilitat $x_{ij} > u_{ij}$ i $z_{ij} - c_{ij} < 0$, es pot disminuir x_{ij} tant com $|x_{ij} - l_{ij}|$ abans que l'arc sigui sense desviació; si minvés més x_{ij}, l'arc també tendria desviació, cosa que no volem; i tampoc es permet en aquest cas augmentar el x_{ij} inicial. En els casos que $z_{ij} - c_{ij} = 0$ si per exemple tenim $x_{ij} > u_{ij}$ podem disminuir el flux fins a $|x_{ij} - l_{ij}|$, ja que tot i que només necessitem disminuir $|x_{ij} - u_{ij}|$, una quantitat menor, per adquirir la conformabilitat, es pot continuar disminuint si es desitja, fet que sol ajudar a altres arca a adquirir estats sense desviació. De la mateixa manera si estem en el cas $l_{ij} < x_{ij} < u_{ij}$ i $z_{ij} - c_{ij} = 0$ el flux es pot augmentar o disminuir i seguir mantenint la conformabilitat. Els altres casos són anàlegs.

	$z_{ij} - c_{ij} < 0$	$z_{ij} - c_{ij} = 0$	$z_{ij} - c_{ij} > 0$
$x_{ij} > u_{ij}$			
$x_{ij} = u_{ij}$			
$l_{ij} < x_{ij} < u_{ij}$			
$x_{ij} = l_{ij}$			
$x_{ij} < l_{ij}$			

A la taula anterior surten, depenent del flux i $z_{ij} - c_{ij}$, el que està permès disminuir (fletxa blava) o augmentar (fletxa vermella) com a màxim el flux.

Ara que ja s'ha determinat quant es pot canviar el flux individual sobre un arc, encara s'ha de determinar quina és la combinació de fluxes que es pot canviar per mantindre un flux circulatori. Si $\bar{\mathbf{x}}$ és el vector de fluxes circulatoris (actuals), llavors l'equació (4.7) es pot escriure com $\mathbf{A}\bar{\mathbf{x}} = 0$, on $\mathbf{A}$ és la matriu d'incidència node-arc. Si Δ és un vector de canvis de flux, llavors s'ha de tenir

$$\mathbf{A}(\bar{\mathbf{x}} + \boldsymbol{\Delta}) = 0 \text{ ó } \mathbf{A}\boldsymbol{\Delta} = 0$$

Si $\mathbf{A}\boldsymbol{\Delta} = 0$ per a Δ diferent de zero, llavors les columnes d'$\mathbf{A}$ corresponents a les components diferents de zero de Δ han de ser linealment dependets. Sabem que cada columna d'$\mathbf{A}$ té exactament un +1 i un -1, i les components diferents de zero de Δ han de correspondre a un cicle o a un conjunt de cicles. Per tant, els fluxes han de canviar al llarg d'un cicle o d'un conjunt de cicles per mantenir les equacions de conservació.

Així doncs, donat un arc amb desviació, s'ha de contruir un cicle que contengui aquest arc. Aquest cicle ha de tenir la propietat que al ser dotat d'orientació i afegir-li un flux, no s'empitjori la desviació de cap arc.

Un mètode per fer això consisteix en construir una nova xarxa $G^{'}$, a partir de l'original seguint els canvis que hem comentat anteriorment. Primerament, cada node de la xarxa original està en la nova xarxa. Posteriorment, si un arc $(i,\ j)$ està a la xarxa original i el flux es pot incrementar, llavors l'arc $(i,\ j)$ formarà part de la nova xarxa amb un canvi de flux permès adequat a les directrius que hem comentat a dalt. Finalment, si un arc $(i,\ j)$ està a la xarxa original i el flux es pot disminuir, llavors l'arc $(j,\ i)$ formarà part de la nova xarxa amb el canvi de flux seguint els mateixos criteris. Els arcs a la xarxa original amb $l_{ij} < x_{ij} < u_{ij}$ i $z_{ij} - c_{ij} = 0$ produiran dos arcs $(i,\ j)$ i $(j,\ i)$ amb els diferents canvis de flux permesos a la nova xarxa. Els arcs que no tenen permès un canvi de flux s'ometen completament a $G^{'}$.

Una vegada s'ha construit la nova xarxa $G^{'}$ i que s'ha seleccionat un arc $(p,\ q)$ amb desviació, es troba un circuit (camí tancat) que contengui aquest arc en $G^{'}$. Aquest circuit en $G^{'}$ correspon a un cicle a G. El flux del cicle en G es canvia seguint l'orientació donada pel circuit a $G^{'}$. La quantitat de canvi està especificada pel menor canvi de flux permès per qualsevol arc que forma part del circuit de $G^{'}$. Si a $G^{'}$ no existeix cap circuit que contengui un arc amb desviació seleccionat, es passa a la fase dual de l'algoritme.

Quan ja no és possible construir un circuit en $G^{'}$ que contengui un arc amb desviació específic, s'hauran de canviar els $z_{ij} - c_{ij}$ de manera que no s'empitjori cap desviació i amb l'esperança que s'introdueixin en $G^{'}$ nous arcs que permetin trobar un circuit que conteingui l'arc no conformable en consideració.

Donat que $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$, s'han de canviar les w_i per poder canviar els $z_{ij} - c_{ij}$. Sigui $(p,\ q)$ un arc amb desviació i sigui X el conjunt de nodes en $G^{'}$ que es poden assolir desde q al llarg d'algun *camí* en $G^{'}$. Sigui $\overline{X} = N - X$, on $N = \{1, 2, \ldots, n\}$ Notem que ni X ni $\overline{X}$ són buit s doncs $q \in X$ i $p \in \overline{X}$ quan es passa a la fase dual.

Es desitja canviar les w_i de tal manera que no empitjori cap desviació i que periòdicament, el conjunt X es faci més gran. Si algun altre node entra a X a intervals finits, llavors finalmet p entrarà en X i es crearà un circuit en $G^{'}$. Hem suposat que X no es farà més petit. Per assegurar que això no succeeixi, s'han de canviar les w_i de manera que tots els arcs amb ambdós extrems en X es conservin a la gràfica modificada.

Considerem $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$. Si w_i i w_j canvien en la mateixa quantitat llavors $z_{ij} - c_{ij}$ no canviarà. Per tant, es pot assegurar que el conjunt X contindrà almenys tots els nodes que contenia després de canviar la variable dual si es canvien totes les w_i en X en la mateixa quantitat θ. Suposem que no es canvien les w_i en $\overline{X}$. Aleshores els únics arcs que es veuran afectats seran els arcs de X a $\overline{X}$ i de $\overline{X}$ a X. Específicament, si $\theta > 0$ i es canvien les w_i d'acord amb

$$w_i = \begin{cases} w_i + \theta & i \in X \\ w_i & i \in \overline{X} \end{cases}$$

llavors, com $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$ tenim:

$$(z_{ij} - c_{ij})^{'} = \begin{cases} z_{ij} - c_{ij} & \text{si } i \in X, j \in X \text{ ó } i \in \overline{X}, j \in \overline{X} \\ (z_{ij} - c_{ij}) + \theta & \text{si } i \in X, j \in \overline{X} \\ (z_{ij} - c_{ij}) - \theta & \text{si } i \in \overline{X} \text{ i } j \in X \end{cases}$$

Per tant, els arcs de X a $\overline{X}$ tindran els seus $z_{ij} - c_{ij}$ augmentats en θ, i aquells arcs de $\overline{X}$ a X disminuits en θ. A la pràctica, aquest fet ens evita la necessitat de treballar amb les variables duals w_i, treballarem només amb els valors $z_{ij} - c_{ij}$. A continuació anem a identificar els arcs que poden estar en el conjunt $(X, \overline{X})$ i en el conjunt $(\overline{X}, X)$. La notació $(X,\ Y)$ representa el conjunt $S = \{(x,y); x \in X\ , y \in Y\}$

El conjunt $(X, \overline{X})$ no pot contindre un arc associat amb l'estat $x_{ij} < l_{ij}$ i $z_{ij} - c_{ij} < 0$, doncs dit arc $(i,\ j)$ en G vendria a ser un arc en $G^{'}$ amb el resultat de que si i pot ser assolit per mitjà d'un camí des de q, aleshores j pot ser assolit des de q, i es tendria que $j \in X$, que és una contradicció. Així doncs, hem vist que els arcs de $(X, \overline{X})$ en G veuen augmentats els seus $z_{ij} - c_{ij}$; a la gràfica que em vist abans els punts aniran a la dreta, per tant, els únics candidats d'aquest conjunt son els arcs que a la gràfica es situen a un punt que al anar cap a la dreta disminuiran en algun moment la seva desviació K_{ij}. Anem a veure cas a cas, un arc de X a $\overline{X}$ en G que compleix $x_{ij} > u_{ij}$ i $z_{ij} - c_{ij} < 0$, es veu de la gràfica que si θ creix, K_{ij} disminueix de $K_{ij} = |x_{ij} - l_{ij}|$ a $K_{ij} = |x_{ij} - u_{ij}|$ i d'aquí endavant permaneix constant. Així doncs, per tal arc, es pot incrementar θ tant com es vulgui i la desviació mai creixerà (el límit superior de θ és ∞), el mateix límit tenim si $x_{ij} = u_{ij}$ i $z_{ij} - c_{ij} < 0$, ja que a partir d'un cert θ, $K_{ij} = 0$. Però si estem en el cas que l'arc de X a $\overline{X}$ en G té $l_{ij} < x_{ij} < u_{ij}$ i $z_{ij} - c_{ij} < 0$ es veu que la desviació K_{ij} quan θ creix s'anul·lara però si segueix creixent aquesta desviació deixarà de ser nul·la. Per tant, aquí s'ha de imposar un límit de $|z_{ij} - c_{ij}|$ sobre θ. El mateix límit s'imposa en el cas $l_{ij} = x_{ij}$ i $z_{ij} - c_{ij} < 0$.

Una anàlisi similar per a arcs de $\overline{X}$ a X en G dóna que els punts reprepresentants dels arcs d'aquest grup es trobaran a la zona anàloga de la gràfica.

En el que respecta l'empitjorament de les desviacions, hem vist que només és necessari calcular θ basat en els arcs de X a $\overline{X}$ i $x_{ij} < u_{ij}$ i en els arcs de $\overline{X}$ a X amb $x_{ij} > l_{ij}$. Però si definissim un mètode per calcular θ basant-nos en aquestes consideracions, surgirien dificultats en el cas $\theta = \infty$. Les coses es simplifiquen si admitim les desigualtats dèbils sobre el flux (és a dir, $x_{ij} \leq u_{ij}$ i $x_{ij} \geq l_{ij}$). Explicarem aquesta raó quan estudiem la convergència de l'algoritme.

L'anàlisi anterior respecte als límits de θ considerant les desviacions i les propietats de convergència (que veurem més endavant) porta al següent procediment per a calcular θ. En G, definim S_1 i S_2 com

$$S_1 \equiv \left\{(i,j) : i \in X, j \in \bar{X}, z_{ij} - c_{ij} < 0, x_{ij} \leq u_{ij}\right\}$$

i

$$S_2 \equiv \left\{(i,j) : i \in \bar{X}, j \in X, z_{ij} - c_{ij} > 0, x_{ij} \geq l_{ij}\right\}$$

Siguin

$$\begin{aligned} \theta_1 &= \min_{(i,j)\in S_1} \{|z_{ij} - c_{ij}|\} \\ \theta_2 &= \min_{(i,j)\in S_2} \{|z_{ij} - c_{ij}|\} \\ \theta &= \min\{\theta_1, \theta_2\} \end{aligned}$$

on $\theta_i = \infty$ si S_i és buit. Per tant, θ és un enter estrictament positiu o ∞. A continuació analitzem els dos casos.

Cas 1: $0 < \theta < \infty$. En aquest cas, es fan els canvis apropiats en w_i (és a dir, $w_i^{'} = w_i + \theta$ si $i \in X$, i $w_i^{'} = w_i$ si $i \in \overline{X}$) i es passa a la fase primal de l'algoritme.

Cas 2: $\theta = \infty$. En aquest cas el problema no té solució factible (ho demostrarem a continuació).

Això completa la fase dual de l'algoritme de les desviacions.

Anem a veure la no factibilitat del problema quan $\theta = \infty$. Suposem que estem a aquest cas $\theta = \infty$. Quan això ocorre s'ha de tenir $S_1 = S_2 = \emptyset$. Donat que si $S_1 = \emptyset$, llavors revisant la definició d'aquest es conclou que $i \in X$ i $j \in \overline{X}$ implica un dels següents casos:

1. $z_{ij} - c_{ij} < 0$ i $x_{ij} > u_{ij}$

2. $z_{ij} - c_{ij} = 0$

3. $z_{ij} - c_{ij} > 0$

I donat que $i \in X$ i $j \in \overline{X}$, la possibilitat (2) o la (3) es pot complir només si $x_{ij} \geq u_{ij}$. Per tant, $S_1 = \emptyset$ es compleix només si $x_{ij} \geq u_{ij}$ per a $i \in X$ i $j \in \overline{X}$. De igual manera $S_2 = \emptyset$ es satisfà només si $i \in \overline{X}$ i $j \in X$ implica que $x_{ij} \leq l_{ij}$. En conseqüència, si $S_1 = S_2 = \emptyset$ implica que

$$x_{ij} \geq u_{ij} \text{ si } i \in X, j \in \overline{X} \tag{4.12}$$

i

$$x_{ij} \leq l_{ij} \text{ si } i \in \overline{X}, j \in X \tag{4.13}$$

En particular, considerem l'arc amb desviació $(p,\ q)$ en G'. Si $(p,\ q)$ està en G, llavors per la desigualtat (4.13) $x_{pq} \leq l_{pq}$. Suposem que $x_{pq} = l_{pq}$, com $(p,\ q)$ és amb desviació, llavors $z_{pq} - c_{pq} > 0$, el qual viola la suposició que $S_2 = \emptyset$, per tant, $x_{pq} < l_{pq}$ Si, per altra banda, $(q,\ p)$ està en G, per un argument similar podem demostrar $x_{qp} > u_{qp}$. Per tant, almenys una de les desigualtats (4.12) o (4.13) és estricta. Sumant aquestes dues desigualtats s'obté

$$\sum_{\substack{i \in X \\ j \in \overline{X}}} x_{ij} - \sum_{\substack{i \in \overline{X} \\ j \in X}} x_{ij} > \sum_{\substack{i \in X \\ j \in \overline{X}}} u_{ij} - \sum_{\substack{i \in \overline{X} \\ j \in X}} l_{ij} \tag{4.14}$$

Donat que el flux actual donat pels x_{ij} és circulatori, l'equació (4.7) es satisfà. Observem que el conjunt de nodes consisteix de $X \cup \overline{X}$ i $X \cap \overline{X} = \emptyset$, llavors l'equació (4.7) es pot escriure com

$$\sum_{j \in X} x_{ij} + \sum_{j \in \overline{X}} x_{ij} - \sum_{j \in X} x_{ji} - \sum_{j \in \overline{X}} x_{ji} = 0 \qquad i = 1, 2, \ldots, n$$

Sumant aquestes equacions sobre $i \in X$, obtenim

$$\sum_{\substack{i \in X \\ j \in X}} x_{ij} + \sum_{\substack{i \in X \\ j \in \overline{X}}} x_{ij} - \sum_{\substack{j \in X \\ i \in X}} x_{ji} - \sum_{\substack{j \in \overline{X} \\ i \in X}} x_{ji} = 0$$

Notant que

$$\sum_{\substack{i \in X \\ j \in X}} x_{ij} = \sum_{\substack{j \in X \\ i \in X}} x_{ji}$$

i que

$$\sum_{\substack{i \in X \\ j \in \overline{X}}} x_{ji} = \sum_{\substack{i \in \overline{X} \\ j \in X}} x_{ij}$$

l'equació anterior es redueix a

$$\sum_{\substack{i\in X\\ j\in\overline{X}}} x_{ij} - \sum_{\substack{i\in\overline{X}\\ j\in X}} x_{ij} = 0 \tag{4.15}$$

Substituin en la desigualtat (4.14) s'obté

$$0 > \sum_{\substack{i\in X\\ j\in\overline{X}}} u_{ij} - \sum_{\substack{i\in\overline{X}\\ j\in X}} l_{ij} \tag{4.16}$$

Suposem, per contradicció, que existeix un flux factible representat per $\hat{x_{ij}}$ per a $i, j = 1, 2, \ldots, n$. Llavors $u_{ij} \geq \hat{x_{ij}}$ i $-l_{ij} \geq -\hat{x_{ij}}$ i per tant, la desigualtat (4.16) dóna

$$0 > \sum_{\substack{i\in X\\ j\in\overline{X}}} u_{ij} - \sum_{\substack{i\in\overline{X}\\ j\in X}} l_{ij} \geq \sum_{\substack{i\in X\\ j\in\overline{X}}} \hat{x_{ij}} - \sum_{\substack{i\in\overline{X}\\ j\in X}} \hat{x_{ij}} \tag{4.17}$$

Però com els x_{ij} representen un flux factible, resulta que han de ser circulatoris. En una forma similar a l'equació (4.15), és clar que el costat dret de la desigualtat (4.17) és igual a zero. En conseqüència, la desigualtat (4.17) implica que $0 > 0$, el qual és impossible. Aquesta contradicció demostra que si $\theta = \infty$, aleshores no pot haver-hi un flux factible.

Notem que si haguèssim definit S_1 i S_2 mitjançant desigualtats estrictes per a x_{ij}, no tindríem la desigualtat requerida en (4.14).

Anem a veure ara la convergència finita d'aquest algoritme. Per fer-ho suposarem que els vectors $\mathbf{l}$, $\mathbf{u}$ i $\mathbf{c}$ prenen valors enters.

Anem a veure algunes propietats de l'algoritme. Primer, cada cop que es constueix en $G^{'}$ un circuit que conté un arc amb desviació, la desviació d'aquest i de la xarxa total es redueix en un enter. Pot construir-se només un nombre finit de circuits que contenen arcs amb desviació abans d'obtenir una solució òptima. Segon, després que canvia cada variable dual, l'estat de conformabilitat de cada arc en G que té ambdós extrems en X roman sense canviar. Per tant, si $(p,\ q)$ és amb desviació aleshores després d'un canvi en una variable dual, cada node en X abans del canvi està en X després del canvi. Existeixen dues possibilitats. Una possibilitat es que un nou node k pot quedar en X perquè es va afegir un arc en $G^{'}$ des d'algun node en X al node k. Cada vegada que això succeeix, el conjunt X creix almenys un node. Això pot ocórrer com a molt un nombre finit de vegades abans que el node p sigui membre de X i creï un circuit que conté a $(p,\ q)$. Per tant, si l'algoritme no és finit, aleshores ocorren un nombre infinit de canvis en variables duals sense que el conjunt X creixi ó $\theta = \infty$. Es provarà que el primer no pot ocórrer.

Suposem que després d'un canvi en variable dual, cap nou node passa a ser membre de X. Llavors al passar a la següent fase dual, prenem els mateixos conjunts X i $\overline{X}$ i els mateixos x_{ij}. A més, per cada arc de X a $\overline{X}$ s'ha incrementat el seu $z_{ij} - c_{ij}$ i per cada arc de $\overline{X}$ a X s'ha disminuït el seu $z_{ij} - c_{ij}$. Per tant, després del canvi a la variable dual els nous conjunts $S_1^{'}$ i $S_2^{'}$ satisfan que

$$S_1^{'} \subseteq S_1 \text{ i } S_2^{'} \subseteq S_2$$

A més a més, per la selecció del valor (finit) de θ, almenys un arc ha estat eliminat de S_1 o de S_2. Llavors, almenys una de les inclusions anteriors és propia. Ara bé, S_1 i S_2 poden decréixer com a molt un nombre finit de vegades abans que $S_1 \cup S_2 = \emptyset$ i $\theta = \infty$, cas on l'algoritme acaba.

Això completa un argument de convergència finita de l'algoritme de desviacions.

4.2 Descripció del programa que resol el problema del cost de flux mínim

El programa que he creat per resoldre el problem l'he desenvolupat en Python 3.8 [11], mitjançant l'entorn integrat de desenvolupament PyCharm.

Els objectius de la creació d'aquest programa són, a banda de veure l'aplicació de l'algoritme de les desviacions d'acord amb la base teòrica exposada, fer un posterior estudi de la seva complexitat per veure què tan eficient és.

Aquest programa comença demanant a l'usuari que introdueixi en aquest ordre:

1. El nombre de nodes, denotat per n, el qual ha de ser major que 1.
2. El valor màxim possible de la cota inferior d'un eix, que denotarem per l_M, la qual ha de ser major que 0.
3. El valor màxim possible de la cota superior d'un eix, denotat per u_M, el qual ha de ser major que l'anterior valor l_M.

A continuació, s'inicialitzen a zero les variables que contindran la informació de la suma total dels temps d'execució de l'algoritme. Aquests temps es calcularan gràcies a la llibreria *time* que permet calcular el temps que triga en trobar la solució. Els recomptes referents al nombre de solucions trobades i al nombre instàncies que no tenen solució trobades també s'inicialitzen a zero.

Seguidament, comença un bucle que fins que no s'han trobat 100 solucions o 500 instàncies generades que no tenen solució, es realitzen les següents accions:

1. Es generen les dades del problema.
2. Es guarden les dades al diccionari que conté totes les dades del problema, s'initzialitzen els fluxes a $x_{ij} = 0$ per a tot eix (i, j), i també els valors de $z_{ij} - c_{ij}$.
3. Es du a terme l'algoritme de les desviacions mentre la solució no és òptima o es veu que no hi ha solució factible possible. Aquest algoritme es divideix, com hem vist, en dues parts:
   - Pas primal.
   - Pas dual.

Quan es generen les dades utilitzant la llibreria *random*, es generen aleatòriament $n \times (n-1)$ arestes dirigides de la xarxa que compleixen:

- El node de destí i el d'origen no poden ser el mateix.
- Si a la xarxa ja està present l'aresta (i, j), no pot estar l'aresta (j, i).
- Not pot haver-hi dues arestes (i, j).

Cada aresta tendrà aleatòriament una cota inferior amb valor a l'interval $[0, l_M]$, una cota superior amb valor a l'interval $[l_M + 1, u_M]$ i un cost per unitat de flux amb valor a l'interval $[0, c_M]$.

El graf de la xarxa G, i el graf de la xarxa creada al pas primal G', es creen utilitzant la llibreria *networkx*.

Al pas primal mirem si hem arribat a una solució òptima. Si en canvi, trobem una aresta (p, q) amb desviació, la seleccionem i creem el graf G'. Si l'aresta (p, q) requereix menys flux per assolir la conformabilitat s'inverteix l'aresta. Després, mitjançant l'algoritme de cerca en profunditat amb una *deque* de la llibreria *collections*, es cerca un camí de q a p. Si existeix aquest camí, tindrem un cicle, i es canvia el flux de la xarxa per aquest. Si no, es passa al pas dual.

Al pas dual creem els conjunts $Xi\overline{X}$, i recorrem els eixos de G per calcular el valor de θ_1 i θ_2. Si θ és infinit vol dir que no hi ha solució factible i l'algoritme s'acaba, i es suma una unitat al recompte de solucions no factibles. Altrament, s'actualitzen els valors pertinents de $z_ij - c_ij$ i es passa al pas primal.

Una vegada la solució és òptima, es calcula quan de temps ha trigat i es suma al total actual de temps d'execució. També es suma una unitat al recompte de solucions factibles òptimes trobades.

Si es troben 100 solucions factibles òptimes, es calcula la mitjana de temps i s'imprimeix per pantalla.

4.3 Temps d'execució experimental de l'algoritme de les desviacions

Sabem que un algoritme és un procediment per resoldre un problema. Per problema ens referim a un model generic com el problema del cost mínim. Una instància és un cas especial d'un problema amb dades especificades per tots els paràmetres del problema. El nombre de passos fets per un algoritme, que en extensió determina el temps que requereix, diferirà d'una instància a una altra. Mentre a una instància "bona", l'algoritme pot resoldre el problema ràpidament, aquest mateix algoritme pot trigar molt en una instància "dolenta".

Per estudiar els algoritmes compararem dos enfocaments diferents:

- *Anàlisi empírica.* L'objectiu d'aquesta anàlisi és estimar com es comporten els algoritmes a la pràctica. Amb un programa informàtic per l'algoritme, es prova el rendiment del programa amb algunes classes d'instàncies.
- *Anàlisi al pitjor cas.* Aquesta anàlisi proveeix una cota superior en el nombre de passos pot realitzar l'algoritme en qualsevol instància del problema. D'aquesta manera garanteix que en un nombre de passos es resoldra el problema.

Ambdós tenen avantatges i desventatges. L'anàlisi empírica té aquestes desventatges: un algoritme rendeix diferent depenent del llenguatge de programació, compilador i ordinador utilitzat per dur a terme els experiments, així com també les habilitats del programador; la comparació d'algoritmes sol ser normalment no concloent en el sentit que un algoritme rendeix millor en diferents classes d'instàncies del problema i diferents estudis empírics donen resultats diferents. L'anàlisi en el pitjor cas és independent de l'entorn de computació i és definitiu a l'hora de determinar si un algoritme és millor a un altre. Però té un gran inconvenient, permet determinar el rendiment de l'algoritme a les instàncies "dolentes", tot i que aquestes poden ser molt infreqüents a la pràctica. Pel seu gran avantatge però és el més àmpliament utilitzat a l'estudi científic dels algoritmes.

Per expressar el temps requerit d'un algoritme, també anomenat *temps d'execució*, haurem d'introduir alguna mesura de complexitat de les instàncies del problema que ens trobarem, normalment a mesura que augmenta la mida del problema, augmenta el temps d'execució del programa que resol aquest. Una *funció de complexitat temporal* és una funció la qual depèn de la mida del problema i especifica la quantitat màxima de temps necessari per resoldre qualsevol instància d'un problema determinat, és a dir, en el pitjor cas. Per representar la complexitat de l'algoritme utilitzarem la notació "O gran", aquesta es defineix així:

Definició 4.3. *Un algoritme es diu que executa en un temps $\mathcal{O}(f(n))$ si per certs nombres c i n_0 el temps que triga l'algoritme és com a molt $cf(n)$ per tot $n \geq n_0$.*

Per tant, la complexitat d'un algoritme és una cota superior del temps d'execució de l'algoritme per suficientment grans valors de n. A més amb aquesta notació només s'indica el terme dominant en el temps d'execució, perquè per valors grans de n, els termes amb un grau de creixement menor, són insignificants comparats amb els d'un ordre superior. Per exemple, si el temps d'execució d'un algoritme és $100n + n^2 + 0.00001n^3$, aleshores per tot $n \geq 100$ domina el terme de segon grau sobre el de primer grau, mentre que per $n \geq 10000$, el tercer terme domina sobre el segon; aleshores tenim que la complexitat de l'algoritme és $\mathcal{O}(n^3)$. Tot i que a la definició només es considera un paràmetre n, podem incorporar altres paràmetres de mida que tengui el problema, com pot ser al nostre cas m, el nombre d'eixos o U la capacitat màxima d'un arc. A vegades, assumirem que U està polinòmicament acotada per n (i.e., $U = \mathcal{O}(n^k)$ per alguna constant k).

En general, diem que un algoritme és bo, si està acotat per una funció polinomial dels paràmetres del sistema, és a dir, en funció de n, m i $\log U$, en aquest cas diem que és un *algoritme de temps polinòmic*, per exemple $\mathcal{O}(m + n \log U)$. Un *algoritme de temps exponencial* si el seu temps d'execució en el pitjor cas creix com una funció que no pot ser polinòmicament acotada per la mida (o mides) de les dades, per exemple $\mathcal{O}(2^n)$. Un *algoritme de temps pseudo-polinòmic* és aquest que la seva funció de complexitat temporal està acotada per n, m i U. Aquests algoritmes són una subclasse important dels algortimes exponencials, un exemple seria $\mathcal{O}(m + nU)$; veiem que per problemes que satisfan l'assumpció anterior, aquests algoritmes són algoritmes de temps polinòmic però aquests no siran molt atractius si U és un polinomi de grau gran respecte n.

Segons (referència), on veiem una implementació amb un enfocament diferent del nostre algoritme, tenim que s'executa en un temps $\mathcal{O}(mUS(n, m, nC))$ on n és $S(n, m, nC)$ és el temps requerit per resoldre el problema del camí més curt amb longitud d'arc no negativa, per tant si s'utilitza Dijkstra genèric tindríem $S(n, m) = \mathcal{O}(n^2)$ i aleshores aquest algoritme de les deviacions en un temps de l'ordre $\mathcal{O}(mUn^2)$,és a dir, és pseudo-polinòmic, on m és el nombre d'arestes, n és el nombre de nodes, U en la implementació de la referència, és la desviació màxima que un arc pot tenir inicialment. Al nostre programa, com ja em dit, seguim l'enfocament de [2], per tant l'algoritme que he realitzat fa uns passos diferents, arribant a la mateixa solució.

A continuació, estudiarem el temps que triga en executar el nostre programa de l'algoritme de les desviacions. Aquest programa, donat un nombre n de nodes, la cota inferior de flux màxima d'un eix, l_M; la cota superior de flux màxima d'un eix, u_M; i un cost màxim per unitat de flux positiu, c_M.

Al nostre estudi, només modificarem el nombre de nodes (i, per tant, el d'arestes). Les dades inicials seran doncs n, i fixarem l_M, u_M i c_M.

Per veure el temps d'execució el que fem és calcular la mitjana del temps d'execució de 100 solucions.

Els resultats obtinguts han estat:

n	**m**	$\mathbf{l_M}$	$\mathbf{u_M}$	$\mathbf{c_M}$	**Temps d'execució**
40	780	3	10	10	0.18s
56	1540	3	10	10	0.62s
79	3081	3	10	10	2.28s
111	6105	3	10	10	9.45s
136	9180	3	10	10	24.74s
157	12246	3	10	10	47.40s
176	15400	3	10	10	79.45s

Al nostre enfocament de l'algoritme de les desviacions, ja hem vist anteriorment que s'assegura la convergència finita de l'algoritme, però en aquest cas no tenim un nombre màxim d'iteracions en les quals es resolgui el problema, per tant, i vists els resultats obtinguts, aquest algoritme entraria dintre de la categoria d'*algoritme de temps exponencial*.

5 Conclusions

L'objectiu principal d'aquest treball de fi de grau ha estat estudiar la programació lineal aplicada a problemes concrets. I, d'acord amb aquest objectiu he tractat un parell d'exemples de tècniques d'optimització lineal que poden ajudar a solucionar: el problema del transport i el problema de flux de cost mínim.

En primer lloc, he exposat la teoria que sustenta la programació lineal, una sèrie de definicions elementals, i el Teorema fonamental de la programació lineal. Tot això, és la base sobre la qual es construeixen els diferents mètodes de resolució d'aquests problemes.

En segon lloc s'ha vist en profunditat el mètode símplex, per la seva importància històrica i perquè constitueix els fonaments de l'algoritme que resolel problema del transport.

En tercer lloc, he introduït conceptes clau com el de la millora de la solució factible bàsica i les condicions que ha de tenir una solució per a garantir l'optimalitat, a part del concepte de pivotatge que s'utilitza en els algoritmes que resolen els problemes vistos.

En quart lloc, he plantejat el problema del transport de forma concreta, i he mostrat la seva estructura especial. Així, després d'estudiar-lo en profunditat, he plantejat diverses formes de representar-lo i estudiar-lo. També he presentat les condicions suficents d'existència de solucions i el fet que quan el problema és equilibrat, hi ha una equació al sistema que és redundant. A més, he explicat l'algoritme de resolució en dues parts:

- A la primera part, que consisteix en obtindre una solució bàsica factible inicial, he tractat dos mètodes per obtenir-la: el de la cantonada noroest i el mètode d'aproximació de Vogel.
- A la segona part, he tractat la millora de solució, la qual es basa en la teoria de la dualitat dels problemes de programació lineal. En aquesta part de dualitat, he mostrat el canvi del problema primal al problema dual, i les condicions d'optimalitat relacionades amb la relació d'aquests dos problemes.

Finalment, he dissenyat un programa informàtic que simula tot el que s'ha plantejat teòricament en els punts anteriors.

A continuació, he vist la teoria de folga complementària, utiltitzada a l'algoritme de les desviacions.

Posteriorment, he treballat conceptes bàsics de grafs i xarxes de flux, i he introduït, entre d'altres, les definicions de circuit i graf dirigit, que són crucials per al posterior plantejament del problema del flux de cost mínim i el desenvolupament del programa.

També he aprofundit en l'algoritme de les desviacions d'acord amb la teoria que el sustenta, amb conceptes com la desviació d'un arc.

Després, he presentat pas a pas el sistema de resolució del problema del flux de cost mínim. I he estudiat les dues parts en què es divideix l'algoritme en profunditat:

- El pas primal, on es cerca reduir la desviació d'un arc mitjançant un canvi de flux per un circuit que contengui aquest arc.
- El pas dual, on es canvia el valor de les variables duals dividint els nodes en dos conjunts i estudiant els arcs que conecten aquests dos conjunts.

Seguidament, he fet un estudi dels conceptes sobre l'anàlisi del rendiment dels algoritmes, i he exposat conceptes com l'anàlisi empírica o l'anàlisi en pitjor cas.

Finalment, també he desenvolupat que s'ha utilitzat el programa realitzat per a veure el rendiment de l'algoritme estudiat a la pràctica, i els resultats indiquen que és de temps exponencial.

Personalment, puc dir que els coneixements adquirits en els meus estudis de matemàtiques i informàtica en aquests quatre anys, m'han permès desenvolupar uns algoritmes de càlcul de solucions a problemes de minimització de costos. Aquests programes informàtics els he dissenyat en llenguatges apresos durant la carrera, Python 3 i C. Així, he pogut constatar com des d'una base teòrica es pot crear una aplicació que pot resultar beneficiosa per a empreses, administracions públiques, etc. I de com es poden modelar unes circumstàncies de la vida real en un model matemàtic.

En definitiva, després d'aquesta tasca d'investigació puc concloure que les matemàtiques i la informàtica en general, i la programació lineal i els algoritmes informàtics en particular, són eines indispensables per trobar solucions en l'àmbit de l'economia i la logística. I que continuar en la recerca d'aquestes branques del coneixement i trobar eines que ens permetin trobar solucions en aquest àmbits, ens farà arribar a un ús més eficient dels recursos escasos disponibles que tenim.

6 Referències

Referències

[1] Ahuja, Ravindra K.; Magnanti, Thomas L.; Orlin, James B: *Network Flows. Theory, algorithms and applications.*

[2] Bazaraa, Mokhtar S.; Jarvis, John J.: *Programación lineal y flujo en redes*, 1a edició: México, Editorial Limusa, 1981. ISBN 968-18-1326-X.

[3] Dantzig, George B.; Thapa, Mukund N.: *Linear Programming 1: Introduction*, Springer, 1997.

[4] Durbin, E. P.; Kroeke D. M.: *The out-of-kilter algorithm: a premier*, Estats Units: The Rand Corporation, 1967.

[5] Fulkerson, Delbert R.: *An Out-of-Kilter Method for Minimal-Cost Flow Problems*, Journal of the Society for Industrial and Applied Mathematics , Mar., 1961, Vol.9, No. 1 (Mar., 1961), pp. 18-27, Society for Industrial and Applied Mathematics.

[6] Günther Hämmerlin; Hoffman Karl-Heinz: *Numerical Mathematics*, Springer, 1991.

[7] Luenberger, David E.: *Programación lineal i no lineal*, Addison-Wesley Iberoamericana, 1989. ISBN 0-201-64408-8.

[8] Matlab code for MODI or U-V Method. 17 de novembre de 2017. `https://www.youtube.com/watch?v=ffN7R7w9gOQ` Accés el 11 de desembre de 2021.

[9] Nishant-Gobi `https://github.com/Nishanth-Gobi/VAM-MODI` Accés el 26 de desembre de 2021.

[10] Surtastic, Alissa M.: *An Efficient Implementation of the Transport Problem*, University of North Florida, 1999.

[11] Van Rossum, G.; Drake, F. L. (2019). *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace.

[12] Zelaia Jauregi, Ana: *Operations Research. Linear Programming [2013/04][Eng]*, 2013, UPV/EHU, `https://ocw.ehu.eus/course/view.php?id=170`

Annexes

A Possible entorn per l'execució del programa del problema del transport.

Juntament amb aquesta memòria s'adjunta l'arxiu del programa del problema del transport anomenat *transportjaume.c*.

Per executar aquest programa en C, el més senzill que es pot fer és utilitzar el sistema operatiu Ubuntu. En el cas de no tenir aquest sistema operatiu natiu, com és el meu cas, es pot crear una màquina virtual utilitzant VirtualBox.

Tot i que es pot executar el programa de diverses maneres, jo ho he fet de la forma següent:

Primerament, s'ha d'accedir a la pàgina oficial de VirtualBox:

`https://www.virtualbox.org/wiki/Downloads`

I s'ha de descarregar la versió corresponent al teu sistema operatiu. En el meu cas he utilitzat la versió 6.1.30 de Windows.

Posteriorment, s'ha de descarregar a la web `https://ubuntu.com/download/desktop` la versió corresponent d'Ubuntu LTS. Jo he utilitzat la Ubuntu-20.04.3-amd64.

Després d'instalar VirtualBox, he creat una màquina virtual de tipus Linux amb la versió Ubuntu (versió 64 bits).

Seguint amb la creació de la màquina virtual he assignat la memòria RAM i de disc dur.

A continuació he iniciat la màquina virtual creada i he seleccionat com a unitat de disc la versió d'Ubuntu descarregada amb anterioritat.

Afegeixo un enllaç a un tutorial per aclarir les passes anteriors: `https://osl.ugr.es/2020/09/29/como-ins`

Una vegada la instal·lació d'Ubuntu a la màquina virtual s'hagi completat, ens descarreguem el programa *transportjaume.c* des de la màquina virtual i accedim des de l'aplicació Terminal a la carpeta de descàrregues, realitzan la comanda:

cd Downloads

Downloads és el nom de la carpeta contenidora del programa.

Després, instal·lem des de la terminal el compilador GNU C, necessari per executar el nostre programa, amb la comanda:

sudo apt install gcc

Posteriorment, realitzem les comandes:

gcc -c -ansi -Wall transportjaume.c gcc tranportjaume.o -o transportjaume.exe -lm I, finalment, executem el programa amb la comanda *./transportjaume.exe*.

B Entorn per l'execució del programa del problema del flux de cost mínim.

Com he exposat en un paràgraf anterior, per desenvolupar i executar el meu programa he utilitzat PyCharm amb la versió 3.8 de Python.

Juntament amb aquesta memòria, s'adjunta comprimit el projecte de PyCharm ProblemaCostFluxMinim.zip

En primer lloc, per instal·lar Pycharm, hem d'accedir a la pàgina:

`https://www.jetbrains.com/es-es/pycharm/download`

En segon lloc, seleccionem el nostre sistema operatiu i descarreguem la versió Community.

En tercer lloc, d'acord amb les instruccions per defecte de l'instal·lador, instal·lem Pycharm. En quart lloc hem de descomprimir el projecte adjunt i obrir amb PyCharm la carpeta del projecte.

En cinquè lloc cal s'instal·lar la versió 3.8 de Python. PyCharm t'ho permet fer fàcilment.

Finalment, a l'arxiu main.py del projecte, cliquem la fletxa verda per executar el nostre programa.

Afegeixo un enllaç a un tutorial per aclarir les passes anteriors:

`https://www.youtube.com/watch?v=lM5Y7BnP56k`.