ÎNTRODUCERE ÎN OPTIMIZAREA LINIARĂ

4.1. Noţiuni despre mulţimi şi funcţii convexe

4.1.1. Mulţimi convexe

O mulţime $M, M \subset R^n$, se numeşte *mulţime conexă* dacă, luând oricare două puncte ale sale $x^1, x^2 \in M$, segmentul de dreaptă ce le uneşte aparţine mulţimii date. Altfel spus, oricare ar fi $x^1, x^2 \in M$ are loc

$$x(\alpha) = \alpha x^1 + (1-\alpha)x^2 \in M$$

pentru orice număr real α, $0 \le \alpha \le 1$ (fig.4.1).

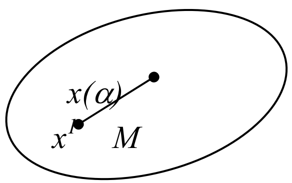


Fig. 4.1.Mulţime convexă.

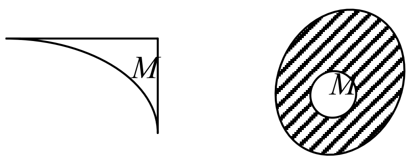


Fig. 4.2. Exemple de mulţimi care nu sunt convexe.

În particular, mulţimea vidă, mulţimea formată dintr-un singur element şi tot spaţiul n dimensional R^n, sunt mulţimi convexe. În fig. 4.2 sunt aduse exemple de mulţimi în R^2 care nu sunt convexe.

Se constată cu uşurinţă că intersecţia unei familii de mulţimi convexe este o mulţime convexă.

Fie $a, x \in R^n, a \neq 0, b \in R$. Mulţimea

$$\Gamma = \{x \mid a^T x = b\} = \left\{ x \;\middle|\; \sum_{i=1}^{n} a_i x_i = b \right\}$$

formează un *hiperplan* din R^n.

Semispaţiile determinate de hiperplanul Γ sunt

$$S^- = \{x \mid a^T x \le b\},$$

$$S^+ = \{x \mid a^T x \ge b\}.$$

Evident, $\Gamma = S^- \cap S^+$. Mulţimile Γ, S^- şi S^+ sunt convexe.

Fie *A* o matrice de dimensiunile $m \times n$, $x \in R^n$ iar $b \in R^m$. Mulţimea

$$T = \{x \mid Ax \le b\} \equiv \left\{ x \;\middle|\; \sum_{j=1}^{n} a_{ij} x_j \le b_i, i = 1,2,...,m \right\}$$

se numeşte *tronson.* Tronsonul reprezintă intersecţia unui număr finit de semispaţii

$$S_i = \left\{ x \;\middle|\; \sum_{j=1}^{n} a_{ij} \le b_i \right\}, \quad i=1,2,\ldots,m,$$

şi, deci, este o mulţime convexă.

Un tronson mărginit se numeşte *poliedru convex.*

Combinaţia liniară a elementelor $x^i \in R^n$, $i=1,2,\ldots,m$ de forma $\alpha_1 x^1 + \alpha_2 x^2 + ... + \alpha_m x^m$ în care $\alpha_i \ge 0$, $i = 1,2,...,m$ şi $\alpha_1 + \alpha_2 + ... + \alpha_m = 1$ se numeşte *combinaţie convexă* a elementelor $x^1, x^2, ..., x^m$.

Teorema 4.1. O mulţime convexă conţine orice combinaţie convexă construită cu elemente ale sale.

Demonstrație (prin inducție). Fie M o mulțime convexă, $M \subset R^n$. Deoarece M este o mulțime convexă, orice combinație convexă, construită cu două elemente din M aparține lui M. Să presupunem că M conține orice combinație a cel mult $m-1, m \geq 3,$ elemente din M. Considerăm combinația convexă $z = \alpha_1 x^1 + \alpha_2 x^2 + \alpha_m x^m$, unde $x^i \in M$, i=1,2,…,m, şi scalarii $\alpha_i > 0$, i=1,2,…,m, astfel ca $\alpha_1 + \alpha_2 + ... + \alpha_m = 1$. Evident, cel puțin un indice i avem $\alpha_i < 1$.Putem presupune că $\alpha_1 < 1$. Atunci $y = \beta_2 x^2 + \beta_3 x^3 + ... + \beta_m x^m$, unde $\beta_i = \dfrac{a_i}{1-a_i}$, i=2,3,…,m, este o combinație convexă a elementelor $x^2 x^3, ..., x^m$ deoarece $\beta_i > 0$ şi $\beta_2 + \beta_3 + ... + \beta_m = 1$. Conform ipotezei inductive că $y \in M$. Cum $z = \alpha_1 x^1 + (1-\alpha_1) y$ şi M este o mulțime convexă rezultă $z \in M$, ceea ce ***demonstrează teorema***.

Se numeşte *proiecția* punctului v pe mulțimea M un punct $p \in M$ astfel încât

$$\|p - v\|_2 \leq \|x - v\|_2$$

pentru orice $x \in M$.

Dacă M este o mulțime convexă închisă atunci există şi este unică proiecția unui punct pe M.

Fie $M \subset R^n$ o mulțime nevidă şi convexă. Vom spune că $x \in M$ este un *punct extrem* al mulțimii M dacă nu există $y, z \in M$, $\alpha \in (0,1)$, $y \neq z$ aşa încât să avem

$$x = \alpha y + (1-\alpha) z.$$

Cu alte cuvinte punctul extrem x aparține intervalului *(y,z)* oricare ar fi $y,z \in M$, $y \neq z$.

Aducem două exemple de puncte extreme ale mulțimilor convexe în R^2. Prin E se notează mulțimea punctelor extreme ale mulțimii M .

1. $M = \{x \mid x_1^2 + x_2^2 \leq 1\}$, $E = \{x \mid x_1^2 + x_2^2 = 1\}$.
2. $M = \{x \mid 2x_1 - x_2 \leq 4\}$, $x_1 + x_2 \leq 5, x_1 \geq 0, x_2 \geq 0$ $\}$.,
$E = \{ (0,0)^T, (0,5)^T, (2,0)^T, (3,2)^T \}$.

În fig. 4.3 sunt reprezentate aceste două mulțimi convexe pentru care sunt evidențiate mulțimea punctelor extreme *E*.

Un hiperplan sau un semispațiu nu are puncte extreme.

O mulțime închisă şi mărginită în R^n se numeşte *compactă.*

Orice mulțime convexă compactă în R^n are puncte

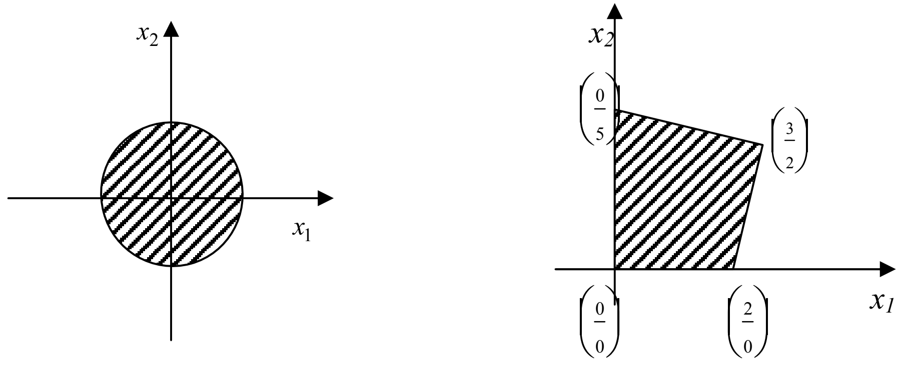


Fig.4.3 Puncte extreme.

extreme. Dacă M este o mulțime convexă compactă în R^n şi are un număr finit de puncte extreme atunci orice element al mulțimii

M se poate exprima ca o combinație convexă a punctelor sale extreme.

Fie tronsonul

$$T = \{x | Ax \le b\},$$

unde $x \in R^n$, A este o matrice de dimensiune $m \times n, m \ge n$, $b \in R^m$. Notăm cu $\overline{A}$ o submatrice de ordinul $n \times n$ a lui A. Presupunem că $\det(\overline{A}) \ne 0$. În acest caz sistemul de ecuații $\overline{A}x = \overline{b}$, unde $\overline{b}$ conține elemente din b (după o eventuală renumerotare), admite o soluție unică $\overline{x}$. Dacă $\overline{x} \in T$ atunci $\overline{x}$ se numeşte *vârf* al tronsonului T.

Se verifică imediat că tronsonul T are cel mult $C_m^n = m!/n!(m-n)!$ vârfuri şi aceste vârfuri sunt punctele extreme ale sale.

Exemplu. Fie tronsonul T definit de

$$\begin{cases} 2x_1 - x_2 \le 4, \\ x_1 + x_2 \le 5, \\ -x_1 \le 0, \\ -x_2 \le 0. \end{cases}$$

Aici $m = 4, n = 2$. Vom avea vârfurile (vezi fig. 4.3)

$$x^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x^2 = \begin{pmatrix} 0 \\ 5 \end{pmatrix}, \quad x^3 = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad x^4 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

date de sistemul de ecuații liniare:

$$\begin{cases} x_1 = 0, \\ x_2 = 0, \end{cases} \quad \begin{cases} x_1 + x_2 = 5, \\ x_1 \qquad = 0, \end{cases} \quad \begin{cases} 2x_1 - x_2 = 4, \\ x_1 + x_2 = 5, \end{cases} \quad \begin{cases} 2x_1 - x_2 = 4, \\ x_2 = 0. \end{cases}$$

4.1.2. *Funcții convexe*

Fie M o mulțime convexă în R^n. Funcția $f(x)$ definită pe mulțimea M se numeşte *funcție convexă* dacă

$$f(\alpha x + (1-\alpha)y \le \alpha f(x) + (1-\alpha) f(y) \tag{4.1}$$

pentru orice $x,y \in M$ şi orice $\alpha \in [0,1]$. Dacă inegalitatea (4.1) este strictă pentru orice $x,y \in M$, $x \ne y$ şi orice $\alpha \in (0,1)$, atunci se spune că funcția $f(x)$ este o funcție *strict convexă.*

Funcția $f(x)$ se numeşte *concavă (strict concavă)* dacă funcția $-f(x)$ este convexă (strict convexă).

Dacă există $\rho \in R$, $\rho > 0$, astfel ca pentru orice $x,y \in M$ şi orice $\alpha \in [0,1]$, are loc inegalitatea

$$f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha) f(y) - \rho\alpha(1-\alpha)\|x-y\|_2^2, \tag{4.2}$$

se spune că funcția $f(x)$ este o funcție *tare convexă.*

Se constată imediat că orice funcție tare convexă este totodată o funcție convexă.

Exemplul 4.1. *Funcția* $f(x) = (x,x) = \|x\|_2^2$, $x \in R$, este o funcție tare convexă. Pentru această funcție inegalitatea (4.2) se transformă în egalitate cu constanta $\rho = 1$:

$$\|\alpha x + (1-\alpha)y\|_2^2 = \alpha\|x\|_2^2 + (1-\alpha)\|y\|_2^2 - \alpha(1-\alpha)\|x-y\|_2^2$$

oricare ar fi $x, y \in R^n$ şi orice $\alpha \in [0,1]$.

Exemplul 4.2. Orice funcție liniară $f(x) = (c,x)$, $x,c \in R^n$, este o funcție convexă şi în acelaşi timp o funcție concavă, deoarece

$$f(\alpha x + (1-\alpha)y) = \alpha f(x) + (1-\alpha) f(y).$$

Fie funcția $f(x), x \in R^n$, admite derivate parțiale

$$\frac{\partial f(x)}{\partial x_i}, \quad \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \quad i, j = 1,2,\ldots,n.$$

Vectorul, componentele căruia sunt derivatele parțiale,

$$\nabla f(x)=\begin{pmatrix}\dfrac{\partial f(x)}{\partial x_1}\\ \dfrac{\partial f(x)}{\partial x_2}\\ \vdots\\ \dfrac{\partial f(x)}{\partial x_n}\end{pmatrix}$$

se numeşte *gradientul* funcției f. Se mai notează prin $grad\, f(x)$ sau $f'(x)$.

Matricea de dimensiunea $n\times n$

$$\nabla f^2(x)=\begin{bmatrix}\dfrac{\partial f^2(x)}{\partial x_1^2} & \dfrac{\partial f^2(x)}{\partial x_1\partial x_2} & \dots & \dfrac{\partial f^2(x)}{\partial x_1\partial x_n}\\ \dfrac{\partial f^2(x)}{\partial x_2\partial x_1} & \dfrac{\partial f^2(x)}{\partial x_2^2} & \dots & \dfrac{\partial f^2(x)}{\partial x_2\partial x_1}\\ \dots\ \dots & \dots & \dots & \dots\\ \dfrac{\partial f^2(x)}{\partial x_n\partial x_1} & \dfrac{\partial f^2(x)}{\partial x_n\partial x_2} & \dots & \dfrac{\partial f^2(x)}{\partial x_n^2}\end{bmatrix}$$

se numeşte *matricea Hesse* a funcției f sau *matricea hessiană*. Se mai notează prin *H(x)* sau $f''(x)$.

În cursul de analiză matematică se demonstrează că, dacă funcția *f(x)* are derivate parțiale de ordinul întâi continue în R^n, atunci

$$f(x+y)=f(x)+(\nabla f(x),y)+o(\|y\|). \qquad (4.3)$$

Dacă *f(x)* admite derivate parțiale de ordinul doi în R^n, atunci

$$f(x+y)=f(x)(\nabla f(x),y)+\frac{1}{2}(\nabla^2 f(x)y,y)+o(\|y\|^2). \qquad (4.4)$$

Teorema 4.2. Fie *f(x)* o funcție care admite derivate parțiale de ordinul întâi continue în R^n. Funcția *f(x)* este convexă, dacă şi numai dacă are loc

$$f(x+y)\ge f(x)+(\nabla f(x),y),\quad \forall x,y\in R^n.$$

Funcția *f(x)* este stric convexă dacă şi numai dacă

$$f(x+y)> f(x)+(\nabla f(x),y),\quad \forall x,y\in R^n,\ y\ne 0.$$

Funcția *f(x)* este tare convexă dacă şi numai dacă

$$f(x+y)\ge f(x)+(\nabla f(x),y)+\frac{\rho}{2}\|y\|_2^2,\quad \forall x,y\in R^n.$$

Cu alte cuvinte graficul funcției convexe (strict convexe) este situat deasupra (strict deasupra) hiperplanului tangent, iar pentru funcții tare convexe graficul este situat deasupra unui oarecare paraboloid (fig. 4.4).

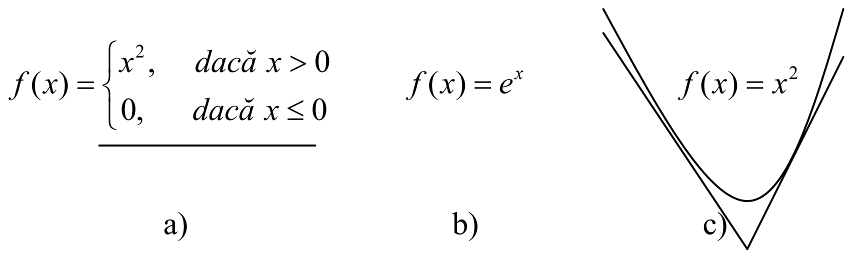


Fig.4.4. Tipurile de convexitate: a) funcție convexă; b) funcție strict convexă; b) funcție tare convexă.

Teorema 4.3. Dacă funcția *f(x)*, $x\in R^n$ are derivatele parțiale de ordinul doi continue în R^n, atunci funcția *f(x)* este convexă dacă şi numai dacă matricea Hesse $\nabla^2 f(x)$ este pozitiv semidefinită pentru orice $x\in R^n$, adică

$$(\nabla^2 f(x)y,y)\ge 0,\quad \forall y\in R^n.$$

O condiţie necesară şi suficientă ca funcţia *f(x)* să fie tare convexă este ca să existe $m \in R, m > 0$, astfel încât

$$(\nabla^2 f(x)y, y) \geq m\|y\|^2, \quad \forall x, y \in R^n. \qquad (4.5)$$

Demonstraţia acestei teoreme poate fi găsită în [13,14,19,21].

Exemplul 4.3. Fie $x \in R^2$ şi funcţia

$$f(x) = x_1^2 - x_1x_2 + 5x_1 - 8x_2 + 3.$$

Avem

$$\nabla f(x) = \begin{pmatrix} 2x_1 - x_2 + 5 \\ -x_1 - 8 \end{pmatrix}, \qquad \nabla^2 f(x) = \begin{pmatrix} 2 & -1 \\ -1 & 0 \end{pmatrix}.$$

Matricea Hesse $\nabla^2 f(x)$ este pozitiv semidefinită şi deci putem afirma că funcţia considerată este convexă.

Să observăm că funcţia din exemplul de mai sus 4.3 poate fi scrisă astfel :

$$f(x) = \frac{1}{2}(x_1, x_2)\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + (5,-8)\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + 3$$

sau

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c = \frac{1}{2}(Ax, x) + (b, x) + c,$$

unde $A = \nabla^2 f(x)$, $b = (5,-8)^T$ şi c=3. De asemenea, mai reţinem că suma vectorilor Ax şi b ne dă gradientul $\nabla f(x)$.

Fie A o matrice simetrică de dimensiune $n \times n$, $b \in R^n$ şi $c \in R$. Funcţia $f(x) = \frac{1}{2}(Ax, x)$ se numeşte *formă pătratică* pe R^n, iar funcţia

$$f(x) = \frac{1}{2}(Ax, x) + (b, x) + c \qquad (4.6)$$

se numeşte *funcţie pătratică.* Se verifică imediat că

$$f(x + y) = f(x) + (Ax + b, y) + \frac{1}{2}(Ay, y), \quad \forall x, y \in R^n. \qquad (4.7)$$

Deoarece $|(Ay, y)| \leq \|A\|_2 \|y\|_2^2$ avem $\frac{1}{2}(Ay, y) = o(\|y\|)$. Prin urmare putem scrie

$$f(x + y) = f(x) + (Ax + b, y) + o(\|y\|)$$

şi conform (4.3) gradientul funcţiei pătratice (4.6) este

$$\nabla f(x) = Ax + b.$$

Acum din (4.4) şi (4.7) rezultă imediat că matricea Hesse $\nabla^2 f(x)$ a unei funcţii pătratice (4.6) este constantă şi este egală cu matricea *A*.

Funcţiile pătratice joacă un rol important în problemele de optimizare. Aceasta se explică prin faptul că în jurul punctelor de minim (vezi capitolul V) funcţiile au o comportare pătratică. Mai mult, dezvoltarea unei funcţii până la aproximaţii de ordinul doi dă o funcţie pătratică.

În cele ce urmează vom arăta, că dacă matricea *A* este pozitiv definită, atunci funcţia pătratică (4.6) este tare convexă. Într-adevăr, fie $(Ax,x) > 0$ pentru $\forall\, x \in R^n$, $x \neq 0$. Notăm prin

$$S = \{z \mid z \in R^n, \|z\| = 1\}.$$

Mulţimea *S* este închisă şi mărginită (este un compact). Cum forma pătratică $g(z) = \frac{1}{2}(Az, z)$ este continuă, există $z^* \in S$ astfel încât $g(z^*) \leq g(z)$ pentru orice $z \in S$. În ipoteza că matricea *A* este pozitiv definită avem

$$\rho = g(z^*) = \frac{1}{2}(Az^*, z^*) > 0.$$

Evident oricare ar fi $x \neq 0$, $x \in R^n$ rezultă că $z = \frac{x}{\|x\|} \in S$ şi

$$\frac{1}{2}(Ax,x)=\frac{\|x\|^2}{2}(A\frac{x}{\|x\|},\frac{x}{\|x\|})=\|x\|^2\cdot\frac{1}{2}(Az,z)\geq\rho\|x\|^2,$$

ceea ce arată că matricea Hesse $\nabla^2 f(x)=A$ satisface inegalitatea (4.5) cu $m=2\rho>0$.

Subliniem faptul că atunci când matricea Hesse este pozitiv definită:

$$\nabla^2 f(x)y,y)>0,\quad \forall x,y\in R^n,\ \ y\neq 0, \qquad (4.8)$$

putem afirma, în caz general, doar că funcția $f(x)$ este strict convexă (dar nu şi tare convexă). Cu alte cuvinte relațiile (4.5) şi (4.8) nu sunt echivalente în cazul funcțiilor nepătratice. Mai mult ca atât, condiția (4.8) este numai o condiție suficientă ca o funcție oarecare (nepătratică) să fie strict convexă.

Exemplul 4.4. Considerăm în spațiul bidimensional R^2 funcția strict convexă (verificați aceasta !) definită prin

$$f(x)=x_1^4+x_2^4.$$

Avem

$$\nabla^2 f(x)=\begin{pmatrix}12x_1^2 & 0\\ 0 & 12x_2^2\end{pmatrix}$$

Matricea Hesse $\nabla^2 f(x)$ nu satisface condiției (4.8) în cazul când

$$x=(0\,,0)^T.$$

4.2. Formularea problemelor de programare liniară

În industrie, comerț, transport, administrație, economie, în aproape orice domeniu al activității umane apar probleme care conduc la alegerea uneia dintre toate posibilitățile de acțiune într-o situație dată şi această alegere trebuie făcută astfel încât să fie asigurată realizarea unui scop bine determinat. Astfel de probleme

se numesc *probleme de decizie.* Un rol important în problemele de decizie îl ocupă *problemele de optimizare* care constau în aflarea maximumului sau minimumului unei funcții de mai multe variabile legate între ele prin anumite relații. Un caz particular al problemelor de optimizare este *programarea liniară* care include un sistem de ecuații şi (sau) inecuații liniare, numite restricțiile problemei, precum şi o funcție liniară care reprezintă scopul dorit, scop definit prin valoarea maximă sau minimă a acesteia.

4.2.1. Problema generală de programare liniară

O problemă generală de programare liniară constă în determinarea numerelor reale $x_1,x_2,\ldots x_n$ care maximizează (sau minimizează) o funcție liniară

$$f=c_1x_1+c_2x_2+\cdots+c_nx_n$$

şi care verifică inegalitățile şi egalitățile liniare:

$$\begin{cases} a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n\leq b_1\ ,\\ a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n\leq b_2\ ,\\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\\ a_{r1}x_1+a_{r2}x_2+\ldots+a_{rn}x_n\leq b_r\ ,\\ a_{r+1,1}x_1+a_{r+1,2}x_2+\ldots+a_{r+1,n}x_n\leq b_{r+1}\ ,\\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\ \ldots\\ a_{s1}x_1+a_{s2}x_2+\ldots+a_{sn}x_n\leq b_s\ ,\\ x_k\geq 0,\ k\in K,\end{cases}$$

unde c_j, a_{ij}, b_i, $i=1,2,\ldots s, j=1,2,\ldots,n$ sunt numere reale date; K este o submulțime de indici ai mulțimii $\{1,2,\ldots,n\}$.

Cu notațiile matriceale:

$$A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{r1} & a_{r2} & \cdots & a_{rn} \end{pmatrix}, \qquad b=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_r \end{pmatrix}, \qquad x=\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

$$\overline{A}=\begin{pmatrix} a_{r+1.1} & a_{r+1,2} & \cdots & a_{r+1,n} \\ a_{r+2,1} & a_{r+2,2} & \cdots & a_{r+2,n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{s1} & a_{s2} & \cdots & a_{sn} \end{pmatrix}, \qquad \overline{b}=\begin{pmatrix} b_{1+r} \\ b_{2+r} \\ \vdots \\ b_s \end{pmatrix}, \qquad c=\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix},$$

problema generală de programare liniară poate fi rescrisă astfel:

$$f(x)=(c,x)\longrightarrow \max \text{ (sau min)}$$

$$\begin{cases} Ax \le b, \\ \overline{A}x=\overline{b}, \\ x_k \ge 0, \ k \in K. \end{cases}$$

Notăm prin

$$T=\left\{x \in R^n \,\middle|\, Ax \le b, \quad \overline{A}x=\overline{b}, \quad x_k \ge 0, \quad k \in K \right\}.$$

Mulțimea T se numeşte *domeniu de soluții admisibile* ale problemei. Domeniul soluțiilor admisibile este o mulțime convexă (vezi paragraful 4.1.1.). Condițiile $Ax \le b$, $\overline{A}x=\overline{b}$, $x_k \ge 0$ se numesc *restricții* ale problemei de programare liniară. Coeficienții a_{ij} şi termenii liberi b_i sunt mărimi constante date.

Funcția $f(x)$ care se maximizează (sau se minimizează) se numeşte *funcție obiectiv,* sau *funcție criteriu,* sau *funcție scop,* sau *funcție de eficiență.*

O soluție admisibilă $x^* \in T$ pentru care avem $f(x^*) \ge f(x), \forall x \in T$, în cazul problemei de maxim sau $f(x^*) \le f(x), \forall x \in T$, în cazul problemei de minim, se va numi *soluție optimă* a problemei de programare liniară.

4.2.2. Exemple de probleme de programare liniară

1. Problema dietei (a meniului optim). Presupunem că dispunem de *n* alimente (pâine, brânză, zahăr etc.) care conțin *m* substanțe nutritive (vitamine, proteine, albumine, grăsimi etc.). Problema dietei constă în determinarea cantităților care trebuie consumate din fiecare aliment, astfel încât să se satisfacă anumite cerințe în substanțe nutritive şi ca să se asigure necesarul biologic la un preț de consum minim.

Fie că o unitate din elementul i conține a_{ij} unități de substanță nutritivă *i.* Mai presupunem că necesarul organismului în substanțe nutritive j este de b_j, iar costul unei unități din alimentul j este c_j. Pentru a formula matematic problema să notăm cu x_i numărul de unități de alimente *i* ce urmează a fi consumate. Problema dietei se va scrie sub forma:

$$f(x)=c_1x_1+c_2x_2+\ldots+c_nx_n \longrightarrow \min$$

cu restricțiile

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \geq b_1 , \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \geq b_2 , \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \geq b_m , \\ x_1 \geq 0, \quad x_2 \geq 0, \ldots , x_n \geq 0. \end{cases}$$

Funcţia liniară $f(x) = \sum_{i=1}^{n} c_i x_i$ reprezintă costul total al meniului, al cărui minimum se caută, iar restricţiile problemei exprimă condiţiile de satisfacere a nevoilor organismului în substanţe nutritive. Evident că variabilele $x_i,\ i = 1,2,\ldots,n,$ au o semnificaţie pentru problema dietei numai dacă sunt negative.

Se pot cerceta şi alte criterii, ca de exemplu maximizarea cantităţii totale de calorii sau minimizarea volumului produselor alimentare.

Problema dietei este un caz particular al unei *probleme de amestec.* Orice problemă în care se doreşte ca din cantităţile $x_i,\ i = 1,2,\ldots,n,$ de produse (alimente, sau cărbuni, sau benzină, sau mijloace de producţie etc.), cu anumite proprietăţi, să se facă o combinaţie care să conţină anumite elemente $j,\ j = 1,2,\ldots,m,$ în cantităţile b_j, dacă se cunosc cantităţile a_{ij} de elemente j conţinute într-o unitate din x_i ,cu cheltuială minimă, se numeşte problemă de amestec (vezi [14]).

2. Problema utilizării optime a resurselor disponibile. Într-o întreprindere se fac *n* feluri de produse notate $P_1, P_2, \ldots, P_n$, care necesită *M* feluri de resurse de materie primă $R_1, R_2, \ldots, R_m$, având la dispoziţie b_i, unităţi din R_i, i=1,2,…,*m*. Pentru o unitate de produs P_i, se consumă a_{ij} unităţi din R_j de materie primă. Mai

cunoaştem că fiecare unitate de produs P_i, dă un beneficiu egal cu c_i lei. Se cere să se afle câte unităţi de fiecare produs sunt necesare pentru a obţine un beneficiu maxim. Astfel spus să se determine un plan de producţie optim, adică un vector $x \in R^n$ astfel încât

$$f(x) = c_1x_1 + c_2x_2 + \quad \ldots \quad + c_nx_n \longrightarrow \max$$

în următoarele condiţii:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 , \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 , \\ \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m , \\ x_1 \geq 0, \quad x_2 \geq 0, \ldots , x_n \geq 0. \end{cases}$$

Aici $x_1, x_2, \ldots x_n$ reprezintă respectiv cantitatea de unităţi de $P_1, P_2, \ldots, P_n$ fabricate în întreprindere.

3. Problema de transport. Să determinăm că un anumit produs este stocat în *m* centre de depozitare $A_1, A_2, \ldots A_m$ în cantităţile $a_1, a_2, \ldots a_n$. Acest produs trebuie transportat la *n* centre de consum $B_1, B_2, \ldots B_n$ în cantităţile $b_1, b_2, \ldots b_n$. Se ştie costul c_{ij} de transport al unei unităţi de produs din centrul A_i la centrul de consum B_j. Se presupune că, $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$ adică rezerva totală din depozite este egală cu cererea totală a centrelor de consum. Se cere să se întocmească planul optim de transport, adică să se determine cantităţile x_{ij} de produs transportate de la A_i la B_j astfel încât costul total de transport să fie minim.

Costul transportului din centrul A_i în centrul B_j este $c_{ij}\, x_{ij}$ unităţi financiare, iar costul total al transportărilor va fi:

$$f = c_{11}x_{11} + c_{12}x_{12} + \cdots + c_{1n}x_{1n} +$$
$$+ c_{21}x_{21} + c_{22}x_{22} + \cdots + c_{2n}x_{2n} +$$
$$+ \cdots + c_{m1}x_{m1} + c_{m2}x_{m2} + \cdots + c_{mn}x_{mn} = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}.$$

Restricțiile problemei se scriu astfel:

*a)*cantitatea transportată din depozitul A_i celor n centre de consum trebuie să fie egal cu disponibilul din A_i , adică

$$x_{i1} + x_{i2} + \ldots + x_{in} = a_i,\ i = 1,2,\ldots,n;$$

*b)*cantitatea transportată în centrul de consum B_i din toate centrele de depozitare este egală cu cantitatea de produs b_i de care are nevoie, adică:

$$x_{1j} + x_{2j} + \ldots + x_{mj} = b, \quad j = 1,2,\ldots,n;$$

c)cantitățile x_{ij} trebuie să fie mărimi nenegative.

Formularea matematică a problemei de transport poate fi prezentată astfel:

Să se determine valoare minimă a funcției

$$f = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}.$$

în condițiile:

$$\begin{cases} \sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1,2,\ldots,m, \\ \sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,2,\ldots,n, \\ x_{ij} \geq 0, \quad i = 1,2,\ldots,m, \quad j = 1,2,\ldots,n. \end{cases}$$

În modelul de mai sus al problemei s-a presupus că este satisfăcută condiția $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$, numită *condiție de echilibru.* În cazul în care condiția de echilibru nu este realizată, problema respectivă se numeşte problema de transport neechilibrată care prin introducerea unui centru fictiv se poate reduce la o problemă de transport echilibrată.

4.2.3. Forme ale unei probleme de programare liniară

Se spune că o problemă de programare liniară are forma *standard* dacă este de forma

$$f(x) = (c,x) \longrightarrow \max$$

cu restricțiile

$$\begin{cases} Ax = b, \\ x \geq 0, \end{cases}$$

unde $x \in R^n$,$c \in R^n$,$b \in R^m$, A este o matrice de dimensiune $m \times n$, , $rang(A) = m < n.$ Deci în problema de programare liniară sub formă standard sistemul de restricții este un sistem de egalități şi tuturor necunoscutelor li se impun condiții de nenegativitate.

Orice problemă de programare liniară poate fi adusă la forma standard. Se verifică imediat că

$$\min_{x \in T} f(x) = -\min_{x \in T}(-f(x)).$$

Prin urmare cazul problemei de minimum se reduce la problema de maximum.

Dacă $Ax \leq b$, $x \geq 0$, atunci putem scrie

$$Ax + y = b,$$
$$x \geq 0, \quad y \geq 0.$$

unde $y = (y_1, y_2, \ldots, y_m)^T$.

Dacă $Ax \geq b$, $x \geq 0$, atunci

$$Ax - y = b,$$
$$x \geq 0, \quad y \geq 0.$$

Variabilele $y_1, y_2, \ldots, y_m$ se numesc *variabile de compensare* sau *variabile ecart.* Menționăm că variabilele de compensare nu apar în funcția obiectiv.

Dacă uneia din variabile x_i nu i se impune condiția de nenegativitate atunci putem înlocui în problema considerată $x_i = u_i - v_i$, unde u_i și v_i sunt alte două variabile noi astfel ca $u_i \geq 0$, $v_i \geq 0$.

Exemplu. Fie problema de programare liniară:

$$f(x) = x_1 - 2x_2 + x_3 \to \min$$

cu restricțiile

$$\begin{cases} 2x_1 + x_2 + x_3 \leq 8, \\ x_1 - 3x_2 + x_3 \geq 1, \\ x_1 \geq 0 \text{ , } x_2 \geq 0. \end{cases}$$

Să aducem această problemă la forma standard.

1. Minimizarea funcției f este echivalentă maximizării lui
$$g(x) = -f(x) = -x_1 + 2x_2 - x_3.$$
2. Substituim $x_3 = x_4 - x_5$, unde $x_4 \geq 0$, $x_5 \geq 0$.
3. Introducem variabilele de compensare $x_6 \geq 0$ și $x_7 \geq 0$

pentru a transforma restricțiile – inecuații în restricții – ecuații:

$$2x_1 + x_2 + x_3 + x_6 = 8,$$
$$x_1 - 3x_2 + x_3 - x_7 = 1,$$
$$x_6 \geq 0 \text{ , } x_7 \geq 0.$$

Deci problema considerată se reduce la următoarea problemă de programare liniară sub forma standard:

$$g(x) = -x_1 + 2x_2 - x_4 + x_5 \longrightarrow \max$$

cu restricțiile

$$\begin{cases} 2x_1 + x_2 + x_4 - x_5 + x_6 = 8, \\ x_1 - 3x_2 + x_4 - x_5 + x_7 = 1, \\ x_i \geq 0 \text{ , } i = 1,2,4,5,6,7. \end{cases}$$

O altă formă de prezentare a problemelor de programare liniară este forma canonică. Dacă toate restricțiile sunt inegalități de acelaşi sens şi toate necunoscutele sunt nenegative, atunci se spune că problema de programare liniară are *forma canonică*:

$$f(x) = (c, x) \longrightarrow \max \text{ (sau min)}$$

cu restricțiile

$$\begin{cases} Ax \leq b, \\ x \geq o, \end{cases} \quad \text{sau} \quad \begin{cases} Ax \geq b, \\ x \geq o. \end{cases}$$

Subliniem că, dacă sensul inegalității este "≤", funcția obiectiv se maximizează, iar pentru "≥" funcția obiectiv se minimizează.

Să observăm că orice restricție de forma

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n = b_i$$

este echivalentă cu sistemul de două restricții

$$\begin{cases} a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \leq b_i, \\ -a_{i1}x_1 - a_{i2}x_2 - \ldots - a_{in}x_n \leq -b_i. \end{cases}$$

Constatăm că orice problemă de programare liniară poate fi adusă la forma standard, sau la forma canonică.

4.2.4. Interpretarea geometrică a problemelor de programare liniară cu două variabile

Considerăm o problemă de programare liniară de formă canonică în spaţiul bidimensional cu funcţia obiectiv $f = c_1x_1 + c_2x_2$ şi sistemul de restricţii:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 \le b_1\ , \\ a_{21}x_1 + a_{22}x_2 \le b_2\ , \\ \dots \quad \dots \quad \dots \quad \dots \\ a_{m1}x_1 + a_{m2}x_2 \le b_m\ , \\ x_1 \ge 0, \quad x_2 \ge 0 \ge 0. \end{cases}$$

Să desenăm în sistemul cartezian de coordonate x_1ox_2 dreptele d_i, ecuaţiile cărora sunt

$$a_{11}x_1 + a_{12}x_2 = b_{i\,,}\ i = 1,2,\dots,m$$

Notăm prin T domeniul din plan pentru care sunt satisfăcute restricţiile problemei considerate de programare liniară şi fie acest domeniu, poligonul *ABCDEF* (fig. 4.5). După cum am văzut în paragraful 4.1.1 T este un poligon convex (tronson).

Pentru a rezolva problema de programare liniară trebuie să căutăm punctele $M(x_1, x_2)$, aparţinând mulţimii T, pentru care funcţia $f = c_1x_1 + c_2x_2$ ia cea mai mare valoare.

Să reprezentăm dreapta d_0 de ecuaţie $c_1x_1 + c_2x_2 = 0$. Această dreaptă trece prin origine şi este perpendiculară vectorului $n = (c_1c_2)^T$. După cum se ştie, vectorul n este îndreptat în sensul crescător al valorilor funcţie f. Fie v o valoare oarecare a funcţiei obiectiv $f = c_1x_1 + c_2x_2$. Atunci ecuaţia $c_1x_1 + c_2x_2 = v$ reprezintă o familie de drepte paralele cu dreapta d_0; când mărimea v variază în intervalul $(-\infty, \infty)$ dreapta $c_1x_1 + c_2x_2 = v$ se deplasează, rămânând perpendiculară pe vectorul n.

Deoarece $M(x_1, x_2)$ trebuie să aparţină mulţimii T, rezultă

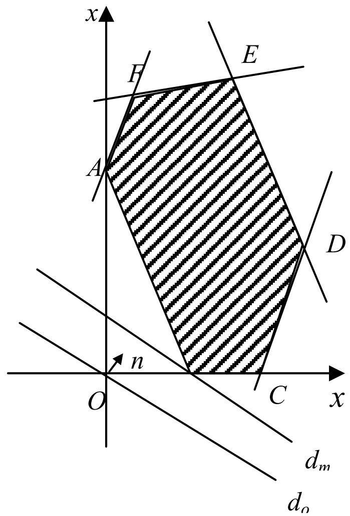


Fig. 4.5

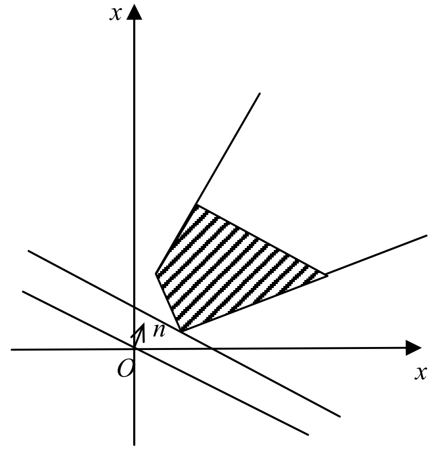


Fig. 4.6

că maximumul funcţiei $f = c_1x_1 + c_2x_2$ se atinge în vârful E a lui T. Într-adevăr, vectorul n arată direcţia în care trebuie să deplasăm dreapta d_0, pentru ca să majorăm valoarea funcţiei obiectiv, deoarece funcţia liniară $f = c_1x_1 + c_2x_2$ este crescătoare în direcţia lui n. Deci valoarea maximă se obţine pentru coordonatele punctului E care se află la intersecţia dreptelor d_2 şi d_3 (vezi fig.4.5). Din fig. 4.5 se vede de asemenea că minimumul funcţiei obiectiv se atinge în vârful B al poligonului T. Dacă dreapta d_0 nu este paralelă cu nici o latura a poligonului soluţiilor admisibile atunci există o singură soluţie optimă

În cazul când una din laturile poligonului soluţiilor admisibile este paralelă cu dreapta d_0, atunci problema admite o infinitate de soluții optime care vor fi date de coordonatele tuturor punctelor situate pe această latură.

O altă situație care poate apărea în problema de programare, este aceea când funcția obiectiv nu are maximum (sau minimum) finit (vezi fig. 4.6). Acest lucru se întâmplă atunci când tronsonul *T* nu este mărginit.

Exemplu. Să se determine valorile lui x_1 și x_2 astfel încât funcția $f = x_1 + x_2$ să fie maximă în următoarele condiții:

$$\begin{cases} x_1 + 2x_2 \geq 4, \\ -x_1 + x_2 \leq 2, \\ 3x_1 + x_2 \leq 3, \\ \quad x_2 \leq 3, \\ x_1 \geq x_2 \geq 0. \end{cases}$$

Pentru aceasta reprezentăm grafic dreptele:

$d_1 : x_1 + 2x_2 = 4,$

$d_2 : -x_1 + x_2 = 2,$

$d_3 : 3x_1 + x_2 = 9,$

$d_4 : x_2 = 3,$

$d_0 : x_1 + x_2 = 0.$

Se obține un poligon convex *ABCD* (fig. 4.7). Dând dreptei d_0 de ecuație $x_1 + x_2 = 0$ o deplasare paralelă cu ea însăși în direcția n se obține că coordonatele

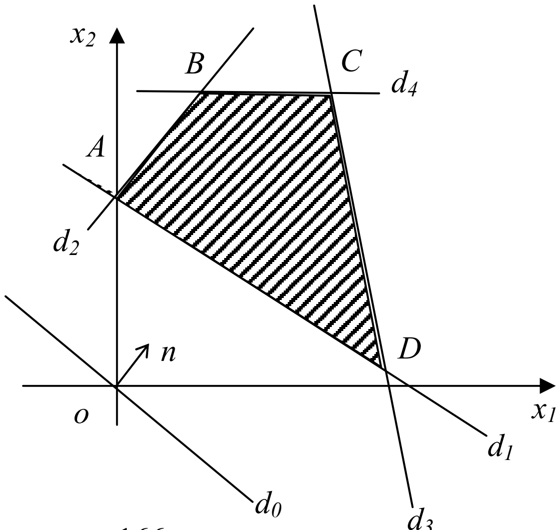


Fig .4.7

punctului *C* realizează maximumul funcției liniare $f = x_1 + x_2$. Determinăm coordonatele acestui punct, care este intersecția dreptelor d_3 și d_4, rezolvând sistemul:

$$\begin{cases} 3x_1 + x_2 = 9, \\ \quad x_2 = 3. \end{cases}$$

Avem $x_1^* = 2, x_2^* = 3$ și max *f*=2+3=5. Rezultă că soluția optimă a unei probleme de programare liniară cu două variabile se atinge într-unul din vârfurile tronsonului *T*. Aceste vârfuri, după cum am văzut în paragraful 4.1, nu sunt altceva decât punctele extreme ale mulțimii *T*.

Fenomenul acesta este general: orice soluție optimă a unei probleme de programare liniară cu un număr oarecare de variabile și restricții corespunde unui punct extrem al tronsonului soluțiilor admisibile.

4.3. Metoda Simplex

Așa cum s-a constatat în paragraful 4.2, problema de programare liniară revine la determinarea punctelor extreme (vârfurilor) ale poliedrului convex și alegerea vârfului pentru care funcția obiectiv ia valoarea optimă. *Metoda simplex* a fost elaborată în anul 1947 de B. G. Dantzing și constă în parcurgerea vârfurilor poliedrului convex în așa fel încât funcția obiectiv are o valoare mai bună în sensul optimului decât în anterior de fiecare dată. Când îmbunătățirea funcției obiectiv nu mai este posibilă, se ajunge la o soluție optimă. Metoda simplex se poate prezenta sub diverse forme (vezi, de exemplu, [6,13,14,19,21,31,36]). În toate cazurile metoda simplex se desfășoară în următoarele etape:

- se determină o soluție inițială de bază;

- printr-un procedeu iterativ se înlocuiește soluția de bază inițială cu alte soluții de bază, până se ajunge la situația că valoarea funcției obiectiv nu mai crește sau la concluzia că problema nu admite un optim finit.

Vom da în continuare descrierea metodei simplex [31]. Să considerăm problema de programare liniară dată sub forma standard:

$$\left.\begin{aligned} f=(c,x)&\longrightarrow \max\\ Ax&=b,\\ x&\geq 0,\end{aligned}\right\} \tag{4.9}$$

unde $c=(c_1,c_2,\ldots,c_n)^T$, $x=(x_1,x_2,\ldots,x_n)^T$, $b=(b_1,b_2,\ldots,b_n)^T$, iar A este o matrice de dimensiunea $m\times n$ cu $m<n$. Presupunem că liniile matricei A sunt liniar independente, adică $rang(A)=m$.

4.3.1. Soluție admisibilă de bază

Notăm cu B submatricea matricei A constituită din m coloane astfel că B este o matrice nesingulară $(\det B\neq 0)$ de ordinul $m\times m$ iar cu N submatricea de ordinul $m\times(n-m)$ ce conține restul coloanelor din A. Deci avem desfacerea

$$A=\left[B|N\right].$$

Matricea *B* se numește matrice bazică, iar *N*-matrice nebazică.

În mod analog se separă vectorii *c* și *x*:

$$c=\left[\frac{c_B}{c_N}\right],\quad x=\left[\frac{x_B}{c_N}\right]$$

unde c_B și x_B se compun din acele *m* componente ce determină desfacerea matricei A. Atunci sistemul $Ax=b$ devine

$$Bx_B+Nx_N=b.$$

Acest sistem poate fi scris sub forma

$$Bx_B=b-Nx_N,$$

sau

$$x_B=B^{-1}b-B^{-1}Nx_N. \tag{4.10}$$

Componentele vectorului x_B se numesc *variabile bazice* (sau *de bază*), iar componentele vectorului x_N sunt numite *variabile nebazice.*

Un sistem de ecuații de forma (4.10) se spune că este *un sistem sub formă explicită.* Soluția acestui sistem dată prin

$$x_B=B^{-1}b,\ x_N=0$$

se numește soluție de bază a sistemului $Ax=b.$

Dacă $B^{-1}b\geq 0$ atunci soluția de bază verifică condițiile de nenegativitate:

$$x=\left[\begin{array}{cc}\dfrac{x_B}{x_N} & \begin{array}{c}(\geq 0)\\(=0)\end{array}\end{array}\right]\geq 0$$

și se numește *soluție de bază admisibilă* a problemei de programare (4.9).

O soluție de bază admisibilă pentru problema (4.9) se numește *nedegenerată,* dacă numărul componentelor sale nenule este egal cu numărul ecuațiilor sistemului $Ax=b$, adică $B^{-1}b>0$.

Se demonstrează (vezi, de exemplu, [13,14]) că soluția de bază admisibilă nu este altceva decât un punct extrem al mulțimii

$$T=\left\{x\middle|x\in R^n,\ Ax=b,\ x\geq 0\right\}$$

și reciproc.

4.3.2. Criteriul de optimalitate

Fie B o matrice bazică și respectiv $c^T=\left[c_B^T\middle|c_N^T\right]$. Atunci funcția obiectiv $f=(c,x)$ devine

$$f = c^T x = \left[c_B^T \middle| c_N^T \right] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = c_B^T x_B + c_N^T x_N =$$

$$= c_B^T (B^{-1}b - B^{-1}Nx_N) + c_N^T x_N = c_B^T B^{-1}b - (c_B^T B^{-1}N - c_N^T)x_N.$$

Problema de programare liniară (4.9) poate fi scrisă sub forma

$$\left.\begin{aligned} f = (c_B, B^{-1}b) - (N^T(B^T)^{-1}c_B - c_N, x_N) &\longrightarrow \max \\ x_N = B^{-1}b - B^{-1}Nx_N, & \\ x_N \geq 0, x_B \geq 0. & \end{aligned}\right\} \qquad (4.11)$$

Această formă se numeşte *formă explicită* a problemei de programare liniară (4.9).

Notăm $\beta = B^{-1}b$ şi $\gamma = (B^T)^{-1}c_B$. Atunci forma explicită poate fi scrisă în felul următor

$$\left.\begin{aligned} f = (c_B, \beta) - (N^T\gamma - c_N, x_N) &\longrightarrow \max \\ x_N = \beta - B^{-1}Nx_N, & \\ x_N \geq 0, x_B \geq 0. & \end{aligned}\right\}$$

Vectorul $\Delta = N^T\gamma - c_N$ reprezintă *criteriul de optimalitate,* deoarece ne spune dacă soluția de bază admisibilă este o soluție optimă de programare liniară (4.9).

Fie z_j, $(c_N)_j$ şi $(x_N)_j$ $j = 1,2,\ldots,n-m$, componentele vectorilor $z = N^T\gamma$, c_N şi respectiv x_N. Constatăm că dacă $\Delta_j = z_j - (c_N)_j \geq 0$, $j = 1,2,\ldots,n-m$ maximul funcției obiectiv

$$f = (c_B, \beta) - \Delta_1(x_N)_1 - \Delta_2(x_N)_2 - \ldots - \Delta_{n-m}(x_N)_{n-m}$$

se atinge pentru $x_N = 0$. Pentru orice altă soluție admisibilă x' cel puțin o componentă $(x'_N)_j > 0$ şi la trecerea de la o soluție admisibilă de bază la alta valoarea funcției obiectiv devine mai mică.

Astfel, *dacă avem o soluție de bază admisibilă pentru care*

$$\Delta_j \geq 0, \quad j = 1,2,\ldots,n-m,$$

atunci ea este o soluție optimă.

Vom nota prin a_j, $j = 1,2,\ldots,n-m$, coloana corespunzătoare din matricea *N*. Se observă că $z_j = a_j^T\gamma = (a_j, \gamma)$. Dacă există *j* astfel încât $\Delta_j < 0$ iar vectorul $\alpha_j = B^{-1}a_j$ are toate componentele mai mici sau egale cu zero ($(\alpha_j)_i \leq 0$ pentru $j = 1,2,\ldots,m$) atunci funcția obiectiv *f* nu este mărginită superior pe mulțimea soluțiilor admisibile. Într-adevăr, se verifică imediat că pentru orice $t \in T, t \geq 0$, vectorul

$$x' = \begin{bmatrix} x'_B \\ x'_N \end{bmatrix}$$

definit prin

$$x'_B = \beta - t\alpha_j, (x'_N)_j = t, \quad (x'_N)_i = 0, \quad i \neq j, \quad i = 1,2,\ldots,m,$$

este de asemenea o soluție admisibilă a problemei de programare liniară considerată. Alegând un t suficient de mare, constatăm că problema are soluții admisibile nemărginite ($Ax' = b, \quad x' \geq 0$ oricare ar fi $t \geq 0$ şi $x' \to \infty$ dacă $t \to \infty$), deci nu admite un maxim finit:

$$f = (c, x') = (c_B, \beta) - t\Delta_j \to \infty \text{ pentru } t \to \infty.$$

Este posibil şi următorul caz. Printre componentele vectorului $\alpha_j = B^{-1}a_j$ (corespunzător diferenței $\Delta_j < 0$) sunt mărimi pozitive. Fie $\Delta_s < 0$ şi $(\alpha_s)_i > 0$, unde *i* este un număr natural, egal cu unul din numerele $1,2,\ldots,m$. În acest caz $(x_N)_s$ nu poate lua valori arbitrare, deoarece pentru $i \neq s$ avem $(x_N)_i = 0$ şi $x_B = \beta - \alpha_s(x_N)_s$ iar $f = (c_B, \beta) - \Delta_s(x_N)_s$.

Din condiția de admisibilitate $x_B \geq 0$ rezultă

$$(x_N)_s \le \frac{\beta_i}{(\alpha_s)_i}, \quad \forall i .$$

Alegând

$$(x_N)_s = \min_{i|(\alpha_s)_i>0} \frac{\beta_i}{(\alpha_s)_i} = \frac{\beta_k}{(\alpha_s)_k}, \tag{4.12}$$

vom obține o altă soluție admisibilă cu

$$(x_B)_s = 0, \quad (x_N)_s = \frac{\beta_k}{(\alpha_s)_k} > 0 .$$

Se poate demonstra (vezi, de exemplu, [13,14]) că această soluție evidențiază o soluție de bază admisibilă corespunzătoare unei noi matrice bazice B'. Matricea B' se obține din matricea precedentă B prin înlocuirea coloanei sale k prin vectorul a_s. Pentru soluția admisibilă de bază obținută, valoarea funcției obiectiv devine mai mare decât valoarea în soluția precedentă.

Această operație de trecere la o soluție admisibilă de bază la alta se numeşte *pivotare* sau *iterație simplex,* iar elementul $(\alpha_s)_k > 0$ se numeşte *pivot.* La această etapă se exclude variabila $(x_B)_s$ din mulțimea de variabile de bază, introducând în locul ei variabila $(x_N)_s$. Având o nouă matrice bazică B', obținem o altă formă explicită a sistemului $Ax = b$:

$$x_{B'} = (B')^{-1}b - (B')^{-1}N'x_{N'} .$$

Alegerea (4.12) ne asigură că soluția de bază

$$x' = \begin{bmatrix} x_{B'} \\ x_{N'} \end{bmatrix} = \begin{bmatrix} (B')^{-1}b \\ 0 \end{bmatrix} \tag{4.13}$$

este o soluție de bază admisibilă, adică $(B')^{-1}b \ge 0$.

Prin urmare, sunt posibile următoarele cazuri:

1. Dacă $\Delta_j \ge 0, \quad j = 1,2,\dots,n-m,$ atunci soluția de bază admisibilă dată prin $x_B = \beta, x_N = 0$ este o soluție optimă a problemei (4.11).
2. Dacă există un indice j astfel încât $\Delta_j < 0$ şi toate elementele de pe coloana lui $\alpha_j = B^{-1}a_j$ sunt mai mici sau egale cu zero, atunci problema de programare liniară (4.11) nu admite un maxim finit.
3. Dacă există cel puțin un s pentru care $\Delta_s < 0$ şi vectorii α_s au componente pozitive, se va determina o nouă soluție admisibilă de bază cu ajutorul formulei (4.13).

4.3.3. Algoritmul simplex

În cele demonstrate mai sus, în paragraful 4.3.2, organizarea calculelor se poate face în diferite moduri. Astfel se obțin diverse variante ale metodei simplex. Ne vom opri asupra uneia din ele [31]. Această variantă a algoritmului simplex, descrisă mai jos, se utilizează în majoritatea programelor speciale pentru rezolvarea problemelor de programare liniară cu ajutorul mijloacelor tehnice de calcul. În practică problemele cu caracter concret au un număr, în general, destul de mare de variabile şi de restricții şi pot fi rezolvate numai cu ajutorul calculatoarelor electronice.

Fie problema de programare liniară în formă explicită (4.11), adică se cunoaşte o matrice bazică B astfel încât vectorul

$$x = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix}$$

este o soluție de bază admisibilă a problemei de programare liniară dată. Atunci paşii algoritmului simplex sunt următorii:

Pasul 1. Se determină vectorul

$$\gamma = (B^T)^{-1} c_B$$

Calculul acestui vector este echivalent cu rezolvarea sistemului de ecuații liniare

$$(B^T)^{-1}\gamma = C_B .$$

Pasul 2. Se calculează produsul scalar

$$z_j = (a_j, \gamma)$$

şi diferențele

$$\Delta_j = z_j - (c_N)_j$$

pentru $j = 1,2,\dots,n-m$. Aici a_j este coloana j din matricea N.

Pasul 3. Se cercetează semnele diferențelor Δ_j.

- Dacă pentru toți j, avem $\Delta_j \geq 0$, atunci soluția de bază disponibilă este o soluție optimă. STOP.
- Dacă există cel puțin un j pentru care $\Delta_j < 0$, atunci se calculează $\Delta_s = \min\limits_{\Delta_j < 0} \Delta_j$ şi se trece la pasul următor.
- **Pasul 4.** Se determină vectorul $\alpha_s = B^{-1} a_s$, adică se rezolvă sistemul de ecuații liniare

$$B\alpha_s = a_s .$$

Pasul 5. Se studiază componentele vectorului α_s.

- Dacă $(\alpha_s)_i \leq 0, i = 1,2,\dots,m,$ atunci problema dată are funcția obiectiv nemărginită superior pe mulțimea soluțiilor sale admisibile. STOP.
- În caz contrar se determină

$$\frac{\beta_k}{(\alpha_s)_k} = \min_{i|(\alpha_s)_i > 0} \left\{ \frac{\beta_i}{(\alpha_s)_i} \right\}$$

şi se trece la pasul 6 al algoritmului.

Pasul 6. Se formează o nouă matrice bazică B' astfel încât în locul coloanei k a matricei B să apară coloana a_s a matricei N şi se calculează soluția de bază admisibilă corespunzătoare noii matrice bazice B', în care

$$(x_N)_s = \frac{\beta_k}{(\alpha_s)_k} \text{ şi } (x_B)_s = 0 .$$

Se revine la primul pas al algoritmului. Algoritmul se repetă până când se obține o soluție optimă (Pasul 3) sau se stabileşte că funcția obiectiv nu este mărginită superior pe mulțimea soluțiilor admisibile (pasul 5, a).

Exemplu. Să folosim algoritmul simplex pentru rezolvarea problemei de programare liniară

$$g = x_2 - x_1 \longrightarrow \min$$

în condițiile următoare:

$$\left.\begin{aligned} -2x_1 + x_2 &\leq 2 \\ x_1 - 2x_2 &\leq 2 \\ x_1 + x_2 &\leq 5 \\ x_1 \geq 0, x_2 &\geq 0. \end{aligned}\right\}$$

În prealabil aducem această problemă la forma standard, adică:

$$f = x_1 - x_2 \longrightarrow \min$$

$$\left.\begin{array}{l} -2x_1 + x_2 + x_3 = 2, \\ x_1 - 2x_2 + x_4 = 2, \\ x_1 + x_2 + x_5 = 5, \\ x_i \geq 0, i = 1,2,3,4,5. \end{array}\right\}$$

Matricea sistemului de ecuaţii este

$$A = \begin{pmatrix} -2 & 1 & 1 & 0 & 0 \\ 1 & -2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

şi are $rang(A) = 3$.

Alegem

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad N = \begin{pmatrix} -2 & 1 \\ 1 & -2 \\ 1 & 1 \end{pmatrix},$$

deoarece $\det(B) = 1 \neq 0$ şi deci B este o matrice bazică. Subliniem faptul că $B^{-1}b = b \geq 0$.

Ţinând cont de aceasta, vom avea

$$c_B = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad x_B = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 5 \end{pmatrix} = \beta.$$

$$c_N = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad x_N = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

$$\alpha_1 = \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix}, \quad \alpha_2 = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}.$$

Pasul 1.

$$\gamma = (B^T)^{-1} c_B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Pasul 2.

$$z_1 = z_2 = 0, \quad \Delta_1 = -(c_N)_1 = -1, \quad \Delta_2 = -(c_N)_2 = 1.$$

Pasul 3. Aşadar $\Delta_1 < 0$ şi deci soluţia de bază admisibilă $x^{(1)} = (0,0,2,2,5)^T$ nu este optimă. Ea se poate îmbunătăţi prin schimbarea matricei bazice *B*.

Pasul 4.

$$\alpha_1 = B^{-1} a_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix}.$$

Pasul 5.

$$\min_{i|(\alpha_1)_i > 0} \left\{ \frac{\beta_i}{(\alpha_1)_i} \right\} = \min \left\{ \frac{2}{1}, \frac{5}{1} \right\} = \frac{\beta_2}{(\alpha_1)_2}.$$

Pasul 6.

$$B' = \begin{pmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad N' = \begin{pmatrix} 1 & 0 \\ -2 & 1 \\ 1 & 0 \end{pmatrix}.$$

Matricea B' se obţine din *B,* înlocuind coloana a doua cu vectorul a_1.

Se calculează

$$(B')^{-1} = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \quad \beta = \begin{pmatrix} 6 \\ 2 \\ 3 \end{pmatrix} = x_{B'} = \begin{pmatrix} x_3 \\ x_1 \\ x_5 \end{pmatrix},$$

$$x_{N'} = \begin{pmatrix} x_4 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad c_{N'} = \begin{pmatrix} -1 \\ 0 \end{pmatrix},$$

$$c_{B'}=\begin{pmatrix}0\\1\\0\end{pmatrix},\quad a_1=\begin{pmatrix}1\\-2\\1\end{pmatrix},\quad a_2=\begin{pmatrix}0\\1\\0\end{pmatrix}$$

şi se revine la primul pas cu $B=B'$.

Pasul 1.

$$\gamma=(B^T)^{-1}c_B=\begin{pmatrix}1&0&0\\2&1&-1\\0&0&1\end{pmatrix}\begin{pmatrix}0\\1\\0\end{pmatrix}=\begin{pmatrix}0\\1\\0\end{pmatrix}.$$

Pasul 2.

$$z_1=(a_1,\gamma)=-2,\quad z_2=(a_2,\gamma)=1,$$
$$\Delta_1=z_1-(c_N)_1=-2-(-1)=-1,$$
$$\Delta_2=z_2-(c_N)_2=1-0=1.$$

Pasul 3. Deoarece $\Delta_1<0$, soluția de bază admisibilă $x^{(2)}=(2,0,6,0,3)^T$ nu este optimă.

Pasul 4.

$$\alpha_1=B^{-1}a_1=\begin{pmatrix}1&2&0\\0&1&0\\0&-1&1\end{pmatrix}\begin{pmatrix}1\\-2\\1\end{pmatrix}=\begin{pmatrix}-3\\-2\\3\end{pmatrix}.$$

Pasul 5. Se calculează

$$\min_{i|(\alpha_1)_i>0}\left\{\frac{\beta_i}{(\alpha_1)_i}\right\}=\frac{3}{3}=\frac{\beta_3}{(\alpha_1)_3}.$$

Pasul 6. În locul coloanei a treia a matricei B apare vectorul α_1

$$B'=\begin{pmatrix}1&-2&1\\0&1&-2\\0&1&1\end{pmatrix},\quad N'=\begin{pmatrix}0&0\\0&1\\1&0\end{pmatrix}.$$

Se obțin datele

$$(B')^{-1}=\begin{pmatrix}1&1&1\\0&1/3&2/3\\0&-1/3&1/3\end{pmatrix},\quad \beta=\begin{pmatrix}9\\4\\1\end{pmatrix}=x_{B'}=\begin{pmatrix}x_3\\x_1\\x_2\end{pmatrix},$$

$$x_{N'}=\begin{pmatrix}x_4\\x_5\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix},\qquad c_{N'}=\begin{pmatrix}0\\0\end{pmatrix},,$$

$$c_{B'}=\begin{pmatrix}0\\1\\-1\end{pmatrix},\qquad a_1=\begin{pmatrix}0\\0\\1\end{pmatrix},\qquad a_2=\begin{pmatrix}0\\1\\0\end{pmatrix}.$$

Reluând algoritmul simplex cu $B=B'$ avem

Pasul 1.

$$\gamma=(B^T)^{-1}c_{B'}=\begin{pmatrix}1&0&0\\1&1/3&-1/3\\1&2/3&1/3\end{pmatrix}\begin{pmatrix}0\\1\\-1\end{pmatrix}=\begin{pmatrix}0\\2/3\\1/3\end{pmatrix}.$$

Pasul 2.

$$z_1=(a_1,\gamma)=\frac{1}{3},\qquad z_2=(a_2,\gamma)=\frac{2}{3},$$

$$\Delta_1=z_1-(c_N)_1=\frac{1}{3}-0=\frac{1}{3},$$

$$\Delta_2=z_2-(c_N)_2=\frac{2}{3}-0=\frac{2}{3}.$$

Pasul 3. Deoarece $\Delta_1 > 0$, $\Delta_2 > 0$, soluția de bază admisibilă obținută $x^{(3)} = (4,1,9,0,0)^T$ este optimă. Avem $\max(f) = 3$ pentru $x_1^* = 4$, $x_2^* = 1$ şi deci $\min(g) = -3$.

Observația 1. Dacă în cursul aplicării algoritmului simplex soluția de bază admisibilă obținută este degenerată (conține componente nenule mai puține decât numărul restricțiilor), atunci este posibilă o ciclare; se poate reveni la o soluție de bază găsită anterior. Subliniem faptul că degenerarea nu implică neapărat ciclarea. Pentru înlăturarea ciclării se procedează într-un mod special (se aplică tehnica de perturbație sau metoda lexicografică; vezi lucrările [13,14,21,31,36]). Deşi multe probleme practice sunt degenerate, puține din ele ciclează. În literatura de specialitate se cunosc exemple de ciclare, dar ele se construiesc cu dificultate.

Observația 2. Algoritmul simplex asigură convergența către soluția optimă într-un număr finit de paşi. Într-adevăr, din faptul că o problemă de programare liniară are numai un număr finit de soluții de bază admisibile nedegenerate după un număr finit de iterații se va ajunge fie la o soluție optimă, fie la concluzia că funcția obiectiv nu este mărginită superior pe mulțimea soluțiilor sale admisibile.

Observația 3. În momentul în care $\Delta_j \geq 0$, $\forall j$, rezultă că funcția obiectiv nu mai poate creşte. Dacă pentru o variabilă nebazică x_k avem $\Delta_k = 0$, atunci în cazul când pe coloana "*k*" este posibil să alegem un pivot $(\alpha_s)_i > 0$ la pasul următor vom obține o nouă soluție optimă care dă aceeaşi valoare pentru funcția obiectiv. Făcând o combinație liniară convexă a soluțiilor optime găsite, se obține mulțimea tuturor soluțiilor optime.

4.3.4. Tabele simplex

Dacă problema de programare liniară nu este mare, se pot organiza calculele după algoritmul simplex cu ajutorul unor tabele numite *tabele simplex.*

Fie dată o problemă de programare liniară în forma explicită (vezi paragraful 4.3.2.):

$$\left.\begin{aligned} & f = (c_{B,}\beta - (N^T\gamma - c_N, x_N) \to \max \\ & x_B = \beta - B^{-1}Nx_N \\ & x_B \geq 0,\ x_N \geq 0 \end{aligned}\right\}$$

unde matricea bazică B se presupune egală cu matricea unitate I. Vom mai presupune că termenii liberi $b_1, b_2, ..., b_m$ sunt negativi. In cazul de față $\beta = b \geq 0$, $\alpha_1 = a$, $\alpha_2 = a_2, ..., \alpha_{n-m} = a_{n-m}$, soluția admisibilă de bază este $(x_B)_1 = b_1$, $(x_B)_2 = b_2$, ... , $(x_B)_m = b_m$, $(x_N)_1 = 0$, $(x_N)_2 = 0$, ..., $(x_N)_{n-m} = 0$, iar funcția obiectiv va avea valoarea $f = (c_B, \beta)$. Cu aceste date putem construi un tabel care se numeşte *tabel simplex* (vezi Tabelul 4.1. din pagina 182).

Fiecare linie a acestui tabel corespunde ecuației care reprezintă variabilele de bază $(x_B)_1$, ..., $(x_B)_m$ în raport cu variabilele $(x_N)_1$, $(x_N)_2$, ..., $(x_N)_{n-m}$. Ultima linie corespunde diferențelor $\Delta_j = z_j - c_j$, unde $z_j = (c_{B,} a_j)$, $j = 1, 2, ..., n\text{-}m.$ Pentru fiecare coloană din matricea bazică vom avea $\Delta_j = 0$.

Dacă toate diferențele Δ_j sunt nenegative, soluția de bază x_B va fi o soluție optimă. În caz contrar se alege drept coloană pivot coloana α_s a variabilei nebazice $(x_N)_s$ pentru care Δ_s este cea mai mică:

$$\Delta_s = \min_{j|(\Delta_j < 0} \Delta_j .$$

Tabelul 4.1. Tabelul simplex

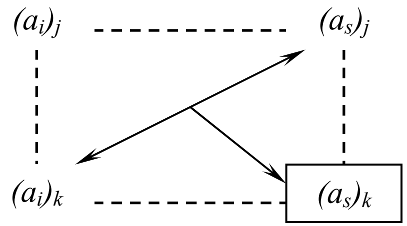
Variabile de bază	β	$(x_B)_1$	$(x_B)_2$	…	$(x_B)_m$	$(x_N)_1$	…	$(x_N)_s$	…	$(x_N)_{n-m}$	$\frac{\beta_i}{(\alpha_s)_i}$
$(x_B)_1$	b_1	1	0	…	0	$(\alpha_1)_1$	…	$(\alpha_s)_1$	…	$(\alpha_{n-m})_1$	
$(x_B)_2$	b_2	0	1	…	0	$(\alpha_1)_2$	…	$(\alpha_s)_2$	…	$(\alpha_{n-m})_2$	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
$(x_B)_k$	b_k	0	0	…	0	$(\alpha_1)_k$		$(\alpha_s)_k$	…	$(\alpha_{n-m})_k$	$\frac{\beta}{(\alpha_{n-m})_k}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
$(x_B)_m$	b_m	1	0	…	1	$(\alpha_1)_m$	…	$(\alpha_s)_m$	…	$(\alpha_{n-m})_m$	
$\Delta_j = z_j - c_j$	f	0	0	…	0	Δ_1	…	$\Delta_s \uparrow$	…	Δ_{n-m}	

Se studiază componentele vectorului α_s. Dacă $(\alpha_s)_i \le 0$ pentru $i = 1,2,...,m$, atunci problema de programare liniară nu admite un maxim finit. Dacă vectorul α_s are componente pozitive, se va alcătui raportul $\beta_i/(\alpha_s)_i$, cu acei i care $(\alpha_s)_i > 0$. Vom alege ca element pivot $(\alpha_s)_k$ (în tabel este notat $[(\alpha_s)_k]$) dacă

$$\frac{\beta_k}{(\alpha_S)_k} = \min_{i|(\alpha_s)i>0}\left\{\frac{\beta_i}{(\alpha_s)_i}\right\}.$$

Linia k se va numi *linie pivot*. Dacă valoarea minimă este atinsă pentru câțiva indici k, atunci se alege unul din ei.

La această etapă se exclude variabila $(x_B)_k$ din mulțimea de variabile de bază, introducând în locul ei variabila $(x_N)_s$ și se trece la tabelul următor. Trecerea de la un tabel la altul constă în calculul componentelor vectorilor care nu fac parte din bază și a diferențelor corespunzătoare $\Delta_j = z_j - (c_N)_j$. Notăm elementele din tabelul următor cu $b_i^{'}, f^{'}, (\alpha_j^{'})_i, \Delta_j^{'}$. Aceste elemente pot fi obținute ușor după așa-numita "*regulă a dreptunghiului*": se formează un dreptunghi, având un vârf în elementul care se calculează, iar elementul pivot este situat în vârful diametral opus.



Valoarea nouă a elementului calculat va fi egală cu produsul dintre valoarea veche și elementul pivot, din care se scade produsul celorlalte vârfuri ale dreptunghiului, rezultatul împărțindu-se la pivot:

$$(\alpha_i^{'})_j = \frac{(\alpha_i)_j.(\alpha_s)_k - (\alpha_i)_k.(\alpha_s)_j}{(\alpha_s)_k} = (\alpha_i)_j - \frac{(\alpha_i)_k.(\alpha_s)_j}{(\alpha_s)_k}.$$

Elementele de pe linia pivot se împart cu coeficientul $(\alpha_s)_k$, iar pentru elementele situate pe coloana pivotului avem $(\alpha_s^{'})_i = 0$ pentru $i \ne k$ și $(\alpha_s^{'})_k = 1$.

Exemplu. Considerăm următoarea problemă de programare liniară

$$f = -x_1 + x_2 + 3x_3 \to \max$$

cu restricțiile

$$\left.\begin{aligned} &2x_1 - x_2 + x_3 \le 1,\\ &4x_1 - 2x_2 + x_3 \ge -2,\\ &3x_1 \qquad\; + x_3 \le 5,\\ &x_i \ge 0; \quad i = 1,2,3 \end{aligned}\right\}$$

Utilizând variabilele de compensare $x_4 \ge 0,\ x_5 \ge 0,\ x_6 \ge 0,$ problema de programare liniară o aducem la forma standard cu termenii liberi nenegativi:

$$\left.\begin{aligned} f = -x_1 + x_2 + 3x_3 + 0\cdot x_4 + 0\cdot x_5 + 0\cdot x_6 \to \max\\ 2x_1 - x_2 + x_3 + x_4 \qquad\qquad = 1,\\ -4x_1 + 2x_2 - x_3 \qquad + x_5 \quad = 2,\\ 3x_1 \qquad + x_3 \qquad\qquad x_6 = 5,\\ x_i \ge 0, i = 1,2,\ldots,6. \end{aligned}\right\}$$

Se observă că sistemul de ecuații din problema standard este explicit în raport cu necunoscutele $x_4,\ x_5,$ şi x_6. Se poate scrie deci primul tabel simplex:

Variabile de bază	β	x_1	x_2	x_3	x_4	x_5	x_6	$\beta_i/(\alpha_s)_i$
x_4	1	2	-1	1	1	0	0	$1:1 \leftarrow$
x_5	2	-4	2	-1	0	1	0	-
x_6	5	3	0	1	0	0	1	$5:1$
Δ_j	0	1	-1	$-3\uparrow$	0	0	0	

În acest tabel ultima linie conține două diferențe negative: $\Delta_2 = -1 < 0$ şi $\Delta_3 = -3 < 0$. Aceasta înseamnă că soluția inițială de bază $x_4 = 1,\ x_5 = 2,\ x_6 = 5,\ x_1 = x_2 = x_3 = 0$ nu este optimă.

Se alege drept coloană pivot, coloana formată din coeficienții $(\alpha_1)_1 = 1,\ (\alpha_1)_2 = -1,\ (\alpha_1)_3 = 1$ ai variabilei nebazice x_3, pentru care $\Delta_3 = -3$ este cea mai mică. Alcătuim rapoartele $\beta_i/(\alpha_1)_i$ pentru care $(\alpha_1)_i > 0 : \beta_1/(\alpha_1)_1 = 1$, $\beta_3/(\alpha_1)_3 = 5$. Prin urmare elementul pivot va fi $(\alpha_1)_1 = 1$.

Aplicând metoda descrisă mai sus vom obține în continuare următoarea succesiune de trei tabele simplex.

Etapa	VB	β	x_1	x_2	x_3	x_4	x_5	x_6	$\frac{\beta_i}{(\alpha_s)_i}$
II	x_3	1	2	-1	1	1	0	0	
	x_5	3	-2	[1]	0	1	1	0	$\frac{3}{1} \leftarrow$
	x_6	4	1	1	0	-1	0	1	$\frac{4}{1}$
	Δ_j	3	7	$-4\uparrow$	0	3	0	0	
III	x_3	4	0	0	1	2	1	0	-
	x_2	3	-2	1	0	1	1	0	-

	x_6	1	[3]	0	0	-2	-1	1	$\frac{1}{3}$
	Δ_j	15	-1↑	0	0	7	4	0	
IV	x_3	4	0	0	1	2	1	0	
	x_2	11/3	0	1	0	-1/3	1/3	2/3	
	x_1	1/3	1	0	0	-1/2	-1/3	1/3	
	Δ_j	46/3	0	0	0	19/3	11/3	1/3	

Toate diferențele Δ_j ale ultimului tabel sunt nenegative, fapt ce semnalează că algoritmul simplex a ajuns la etapa finală. Prin urmare, avem soluția optimă $x_1^* = 1/3, x_2^* = 11/3, x_3^* = 4$ și valoarea optimă a funcției obiectiv $f_{\max} = 46/3$.

Observație. Dacă există mai mult de un raport minim

$$\frac{\beta_k}{(\alpha_s)_k} = \frac{\beta_r}{(\alpha_s)_r} = \min_{i|(\alpha_s)_i>0}\left\{\frac{\beta_i}{(\alpha_s)_i}\right\}$$

atunci putem alege ca element pivot $(\alpha_s)_k$ sau $(\alpha_s)_r$. Alegând elementul pivot $(\alpha_s)_k$ vom obține $(x_B)_r = 0$. Soluția de bază obținută va fi degenerată. În acest caz este posibil ca valoarea funcției obiectiv să nu se modifice în cursul câtorva etape succesive şi să revenim la una din soluțiile de bază admisibile prin care am trecut deja. Această situație se numeşte *ciclare*; algoritmul simplex poarte continua la infinit fără a conduce la soluții. Iată de

ce se presupune nedegenerarea în aplicarea algoritmului simplex. Finitudinea algoritmului simplex este asigurată pentru probleme de programare liniară nedegenerată (având toate soluțiile de bază nedegenerate). Subliniem încă o dată faptul că degenerarea (existența a cel puțin unei soluții de bază degenerate) nu implică neapărat ciclarea. Deşi multe probleme practice sunt degenerate, nici una nu a ciclat până acum şi exemplele de ciclare au fost construite cu dificultate. În cazul degenerării se procedează într-un mod special pentru a evita fenomenul de ciclare, utilizând fie tehnica de perturbare, fie metoda lexicografică (vezi de exemplu [13,21,31,36]).

4.3.5. Determinarea soluției inițiale de bază

După cum s-a stabilit în paragraful 4.3.3, pentru rezolvarea unei probleme de programare liniară se pleacă de la o soluție de bază inițială şi folosind algoritmul simplex, se ajunge la o soluție optimă. Nu orice matrice *B* din desfacerea $A = [B\ N]$ ne asigură că $x_B = B^{-1}b \geq 0$ şi prin urmare trebuie să continuăm cu altă matrice *B*. Numărul de probe poate fi destul de mare. Mai mult, în cazul când mulțimea soluțiilor admisibile este vidă nu există matricea *B* ce ne-ar da o soluție de bază admisibilă.

Considerăm o problemă de programare liniară în formă standard

$$\begin{aligned} &f = (c, x) \to \max \\ &\left.\begin{aligned} Ax = b,\\ x \geq 0 \end{aligned}\right\} \end{aligned} \tag{4.14}$$

unde *A* este o matrice de dimensiune $m \times n$, $x \in R^n$, $c \in R^n$, $b \in R^m$. Vom presupune că $b \geq 0$; în caz contrar putem înmulți linia *i* în care $b_i < 0$ cu –1. Asociem problemei (4.14) următoarea problemă de programare liniară

$$g = y_1 + y_2 + ... + y_m \to \min$$
$$\left.\begin{array}{r} Ax + y = b, \\ x \geq 0,\ y \geq 0 \end{array}\right\} \qquad (4.15)$$

Variabilele $y_1, y_2, ..., y_m$ se numesc *variabile artificiale.* Problema (4.15) are forma explicită cu matricea *B*=*I* unde *I* este matricea unitate de dimensiune $m \times m$ şi cu soluția de bază admisibilă $y_1 = b_1 \geq 0, ..., y_m = b_m \geq 0$, $x_1 = x_2 = ... = x_n = 0$. Prin urmare, putem aplica algoritmul simplex pentru rezolvarea problemei (4.15).

Dacă problema (4.15) are o soluție optimă (x^*, y^*) şi $\min(g) = 0$, cum $y_i^* \geq 0$, rezultă că toți $y_i^* = 0$.

Sunt posibile două cazuri [13, 14]:

1. În urma aplicării algoritmului simplex nici una din variabilele artificiale $y_1, y_2, ..., y_m$ nu este variabilă bazică. Atunci, înlăturând variabilele artificiale şi coeficienții acestora se obține un tabel simplex corespunzător unei forme explicite a sistemului $Ax = b$ în care termenii liberi sunt nenegativi.
2. În tabelul simplex final unele dintre variabilele artificiale $y_1, y_2, ..., y_m$ sunt variabile bazice. Aceasta este posibil dacă problema (4.15) este degenerată sau dacă $rang(A) = r < m$. Dacă $rang(A) = m$ şi problema (4.15) este degenerată, atunci în loc de variabila bazică $y_i = 0$ introducem variabila nebazică x_i fără a afecta nenegativitatea termenilor liberi şi nici optimalitatea soluției pentru problema (4.15), deoarece soluția de bază admisibilă nou obținută este aceeaşi cu soluția precedentă. În cazul

$rang(A) = r < m$, pot rămâne printre variabilele bazice $y_i = 0$, dar liniile lor se exclud din sistem fără a se modifica mulțimea soluțiilor admisibile pentru problema (4.14).

În acest mod se pot elimina succesiv dintre variabilele bazice din tabelul simplex final toate variabilele artificiale, după care se poate continua algoritmul simplex cu soluția de bază admisibilă x^* pentru problema (4.14).

Dacă $\min(g) > 0$, în acest caz cel puțin un $y_i^* > 0$ şi, deci, problema (4.14) nu are soluții admisibile şi deci nici soluție optimă.

Această metodă se mai numeşte *metoda celor două faze.* O altă metodă este *metoda lui M – mare* (*metoda penalizării*), care constă în rezolvarea următoarei probleme

$$f = (c, x) - M\sum_{i=1}^{m} y_i \to \max$$
$$\left.\begin{array}{r} Ax + y = b, \\ x \geq 0,\ y \geq 0 \end{array}\right\} \qquad (4.16)$$

unde *M* este un număr real suficient de mare (de aici şi denumirea metodei).

Problema (4.16) are o soluție de bază admisibilă $y_i = b_i \geq 0$, $i = 1,2,...,m$; $x_i = 0$, $i = 1,2,...,n$. Dacă în urma aplicării algoritmului simplex problema (4.16) are o soluție optimă (x^*, y^*) pentru care $y^* = 0$, atunci x^* este soluția optimală a problemei (4.14). În cazul când $y^* \neq 0$ problema (4.14) nu are soluții admisibile. Dacă problema (4.16) are funcția obiectiv nemărginită superior atunci problema (4.14) nu are soluții optime (nu are soluții admisibile sau funcția obiectiv nu este mărginită superior).

4.4. Dualitatea în programarea liniară

4.4.1. Probleme duale simetrice

Fie o problemă de programare liniară de forma

$$\left.\begin{array}{l} f(x)=(c,x)\to \max \\ Ax\le b, \\ x\ge 0 \end{array}\right\} \qquad (P)$$

în care $c\in R^n$, $b\in R^m$, $x\in R^n$, iar A este o matrice de dimensiune $m\times n$.

Să-i asociem problemei (*P*) o altă problemă de programare liniară definită prin

$$\left.\begin{array}{l} g(y)=(b,y)\to \min \\ A^T y\le c, \\ y\ge 0 \end{array}\right\} \qquad (D)$$

în care $y\in R^m$, iar *A*, *b* şi *c* sunt datele problemei (P).

Fiind dată problema (*P*), numită *problema primală*, *problema* (*D*) se va numi *problema duală* asociată problemei (*P*).

Exemplu. Să se scrie problema duală problemei:

$$f(x)=2x_1-3x_2+x_3+4x_4\to \max$$

$$\left.\begin{array}{l} 3x_1-2x_2-3x_3+x_4\le 4, \\ x_1+5x_2-4x_3+6x_4\le 8, \\ 2x_1+x_2+x_3+x_4\le 5, \\ x_i\ge 0,\ i=1,2,3,4, \end{array}\right\}$$

Avem problema duală

$$g(y)=4y_1+8y_2+5y_3\to \min$$

cu restricţiile

$$\left.\begin{array}{l} 3y_1+\ y_2+2y_3\ge 2, \\ -2y_1+5y_2+\ y_3\ge -3, \\ y_1-4y_2+y_3\ge 1, \\ y_1+6y_2+y_3\ge 4, \\ y_i\ge 0,\quad i=1,2,3. \end{array}\right\}$$

Dacă transformăm problema (*D*) într-o problemă de minim, scriind-o sub forma

$$\left.\begin{array}{l} -g(y)=-(b,y)\to \max \\ -A^t y\le -c, \\ y\le 0, \end{array}\right\} \qquad (D')$$

si construim duala problemei (D') atunci problema duală asociată acesteia va fi problema (*P*). Putem spune că problemele (*P*) şi (*D*)constituie o pereche de probleme duale una alteia. Aceste probleme (*P*), (*D*) se mai zic *duale simetrice.*

4.4.2. Teoreme duale ale programării liniare

Între problemele primală şi duală pot fi stabilite unele relaţii care se dovedesc a fi de multe ori deosebit de utile atât din punct de vedere practic cât şi din punct de vedere teoretic.

Teorema 4.1. Fie $\overline{x}$ o soluţie admisibilă a problemei primale (P), iar $\overline{y}$ o soluţie admisibilă a problemei duale(D). Atunci are loc relaţia

$$f(\overline{x})\le g(\overline{y}).$$

Demonstraţie. Să presupunem că $\overline{x}\ge 0, \overline{y}\ge 0$ sunt soluţii admisibile ale problemelor (P), (D). Înmulţim $A\overline{x}\le b$ din (P) cu $\overline{y}\ge 0$ şi $A^T\overline{y}\ge c$ din(D) cu $\overline{x}\ge 0$. Obţinem

$$(c,\overline{x})\le (A^T\overline{y},\overline{x})=(\overline{y},A\overline{x}),\qquad (b,\overline{y})\ge (A\overline{x},\overline{y})=(\overline{y},A\overline{x}),$$

de unde rezultă că $g(\bar{y}) \geq f(\bar{x})$ şi ***teorema este demonstrată***.

Teorema 4.2. Dacă x^* este o soluţie admisibilă a problemei (P) iar y^* este o soluţie admisibilă a problemei (D) şi are loc

$$(b, y^*) = (c, x^*)$$

atunci y^* şi x^* sunt soluţii optime a problemei (D) şi respectiv problemei (P).

Demonstraţie. Fie x^* o soluţie admisibilă a problemei (P). Din teorema 4.1 rezultă că pentru orice soluţie admisibilă $\bar{y}$ a problemei (D) are loc

$$(b, \bar{y}) \geq (c, x^*)$$

şi având în vedere că $(c, x^*) = (b, y^*)$ urmează

$$(b, y^*) \geq (b, \bar{y}).$$

Deci y^* este o soluţie optimă a problemei (D). În mod analog se arată că x^* este soluţie optimă a problemei (P). ***Teorema este demonstrată***.

Teorema 4.3 (Teorema dualităţii). Dacă problema primală (P) are soluţie optimă atunci şi problema duală ei (D) de asemenea are soluţie optimă, iar valorile optime ale funcţiilor obiectiv din cele două probleme sunt egale: $f_{\max} = g_{\min}$.

Demonstraţie. Să presupunem că problema (P) admite o soluţie optimă x^*. În acest caz soluţia x^* ar putea fi obţinută rezolvând cu algoritmul simplex problema standard asociată problemei (P):

$$(c, x) + (0, x_s) \to \max$$

$$Ax + Ix_s = b,$$
$$x \geq 0,\ x_s \geq 0,$$

în care $0 = (0,0,...,0)^T \in R^m$, $x_s \in R^m$, iar I este matricea unitate de ordin m.

Dacă notăm cu B submatricea matricei $[A\ I]$ construită din acele coloane ale acesteia pentru care soluţia optimă va fi

$$x^* = \begin{bmatrix} x_B^* \\ 0 \end{bmatrix} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix},$$

atunci $\Delta_j = (c_B, B^{-1}a_j) - (c_N)_j \geq 0$, unde c_B este vectorul constituit din acele componente ale vectorului c care corespund necunoscutelor x_B^* din ultimul tabel simplex asociat problemei standard de mai sus.

Putem scrie că

$$(c_B, B^{-1}a_j) \geq (c_N)_j,\ j = 1,2,...,n-m.$$

Deoarece

$$c_B^T B^{-1} - c_B^T = 0^T,$$

rezultă

$$c_B^T B^{-1} a_j = (c_B)_j$$

şi

$$c_B^T B^{-1}[A\ I] \geq [c^{Tt}\ 0^T].$$

Dacă notăm $(y^*)^T = c_B^T B^{-1}$ din ultima relaţie rezultă

$$(y^*)^T A \geq c^T \text{ şi } (y^*)^T I \geq 0$$

sau

$$A^T y^* \geq c \text{ şi } y^* \geq 0$$

şi deci y^* este o soluţie admisibilă a problemei duale (D). În plus putem scrie

$$(b, y^*) = (y^*)^T b = c_B^T B^{-1} b = c_B^T x_B^* = c^T x^* = (c, x^*).$$

Dar x^* este o soluție admisibilă (chiar optimă) a problemei (P), iar y^* este o soluție admisibilă a problemei (D) şi totuşi folosind teorema 4.1 constatăm că

$$(b, y) \geq (c, x^*) = (b, y^*)$$

pentru orice soluție admisibilă *y*, adică y^* este o soluție optimă a problemei (D). ***Teorema este demonstrată***.

Problema primală (P) poate fi considerată ca duala problemei (D) şi atunci din teorema 4.3 rezultă imediat că şi în cazul când problema duală (D) are o soluție optimă urmează că problema primală (P) are de asemenea soluție optimă şi valorile optime ale funcțiilor obiectiv sunt egale.

Teorema 4.4 (Teorema ecarturilor complementare). Pentru ca două soluții admisibile x^* şi y^* ale problemelor duale (P) şi respectiv (D) să fie soluții optime este necesar şi suficient ca aceste soluții să verifice relațiile:

$$(y^*, Ax^* - b) = 0,$$
$$(x^*, A^T y^* - c) = 0.$$

Demonstrație. *Necesitatea.* Fie x^* şi y^* soluții optime ale problemei duale (P) şi respectiv (D). Putem scrie

$$(b, y^*) = (c, x^*) \text{ şi } (y^*, Ax^*) = (A^T y^*, x^*).$$

De unde

$$(y^*, b - Ax^*) + (x^*, A^T y^* - c) = 0.$$

Ținând cont de faptul că $y^* \geq 0, x^* \geq 0, b \geq Ax^*$ şi $A^{tT} y^* \geq c$, din relația de mai sus rezultă

$$(y^*, b - Ax^*) = 0,$$
$$(x^*, A^T y^* - c) = 0.$$

Suficiența. Reciproc, dacă x^* este admisibilă pentru problema primală, iar y^* pentru cea duală, din $(y^*, b - Ax^*) = 0$

şi $(x^*, A^T y^* - c) = 0$, cum avem $(A^T y^*, x^*) = (y^*, Ax^*)$, rezultă $(b, y^*) = (c, x^*)$. Adică x^* este o soluție optimă a problemei primale (P) iar y^* soluție optimă a problemei duale (D). ***Teorema este demonstrată***.

4.4.3. Algoritmul simplex dual

Aplicarea algoritmului simplex la rezolvarea problemei duale ne conduce la un algoritm nou de rezolvare a problemelor de programare liniară, numit *algoritmul simplex dual.* Algoritmul simplex dual construieşte o succesiune de soluții de bază ale problemei primale astfel încât prima soluție de bază admisibilă este o soluție optimă a problemei. Pentru început vom descrie procedeul de aflare a soluției optime cu ajutorul metodei simplex, corespunzătoare soluției optime a problemei primale.

Fie x^* soluția optimă a problemei primale

$$f(x) = (c, x) \to \max$$
$$Ax \leq b, y \geq 0.$$

Din demonstrația teoremei dualității (vezi paragraful 4.4.2.) se deduce imediat că vectorul

$$y^* = \left(B^{-1}\right)^T c_B$$

este o soluție optimă a problemei duale

$$g(y) = (b, y) \to \min$$
$$A^T y \geq c, \ y \geq 0.$$

În limbajul tabelelor simplex aceasta înseamnă că coordonatele vectorului y^* sunt situate în ultima linie a tabelului simplex corespunzător vectorului unitate.

Exemplu. Fiind dată problema primală:

$$f(x) = x_1 + x_2 + 2x_3 \to \max$$

cu restricțiile:

$$\left.\begin{array}{l} x_1 + \frac{1}{2}x_2 + x_3 \le 1, \\ x_1 + 2x_2 + 3x_3 \le 2, \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0 \end{array}\right\} ,$$

problema duală va fi:

$$g(y) = y_1 + 2y_2 \to \min$$

cu restricțiile:

$$\left.\begin{array}{l} y_1 + y_2 \ge 1, \\ \frac{1}{2}y_1 + 2y_2 \ge 1, \\ y_1 + 3y_2 \ge 2, \\ y_1 \ge 0, \quad y_2 \ge 0 \end{array}\right\}.$$

Aplicăm algoritmul simplex la problema primală adusă la forma standard şi obținem următoarea succesiune de tabele simplex:

Etapa	Variabile de bază	β	x_1	x_2	x_3	x_4	x_4	$\frac{\beta_i}{(\alpha_s)_i}$
I	x_4	1	1	1/2	1	1	0	1:1
	x_5	2	1	2	[3]	0	1	$2:3 \leftarrow$
	Δ_j	0	-1	-1	-2↑	0	0	
II	x_4	1/3	[2/3]	1/6	0	1	-1/3	$1:3 \leftarrow$
	x_4	2/3	1/3	2/3	1	0	1/3	2
	Δ_j	4/3	-1/3↑	1/2	0	0	2/3	
III	x_4	1/2	1	1/4	0	3/2	-1/2	
	x_4	1/2	0	7/12	1	-1/2	1/2	
	Δ_j	3/2	0	7/12	0	1/2	1/2	

Soluția optimă a problemei primale este $x_1^* = 1/2$, $x_2^* = 0$, $x_3^* = 1/2$. Conform celor spuse mai sus, soluția optimă a problemei duale se găseşte pe ultima linie a tabelelor simplex: $y_1^* = 1/2$, $y_2^* = 1/2$. Pentru ambele probleme, funcția de eficiență are valoarea

$$f_{\max} = g_{\min} = 3/2 .$$

Să trecem acum la descrierea algoritmului simplex dual. Considerăm o problemă de programare liniară în formă explicită (cu notațiile precedente): să se determine valoarea maximă a funcției

$$f(x) = (c_B, \beta) - \Delta_1 (x_N)_1 - \Delta_2 (x_N)_2 - \quad \cdots \quad - \Delta_{n-m} (x_N)_{n-m}$$ în

următoarele condiții

$$x_B = \beta - \alpha_1 (x_N)_1 - \alpha_2 (x_N)_2 - \quad \cdots \quad - \alpha_{n-m} (x_N)_{n-m}$$

$$x_N \ge 0, \quad x_B \ge 0 .$$

Se spune că problema de programare liniară este *dual admisibilă* dacă este în formă explicită şi diferențele $\Delta_1, \ \Delta_2, ..., \Delta_{n-m}$ sunt negative. Atunci când matricea bazică B coincide cu matricea unitate I, problema este dual admisibilă dacă coeficienții funcției obiectiv sunt nepozitivi.

Să presupunem că $\Delta_j \ge 0$ pentru orice $j = 1,2,...,n-m$. Soluția de bază

$$x = \begin{bmatrix} x_B \\ 0 \end{bmatrix} = \begin{bmatrix} \beta \\ 0 \end{bmatrix}$$

se va numi *soluție dual admisibilă* a problemei.

Algoritmul simplex dual este algoritmul simplex aplicat dualei fără însă a construi problema duală. Acest algoritm execută ciclic următoarea succesiune de paşi.

Pasul 1. *Alegerea liniei pivot p*.

Se determină indicele p astfel încât $\beta_p = \min\limits_{i|\beta_i<0} \beta_i$.

În cazul că $\beta_i \geq 0, \forall i$, atunci soluția de bază disponibilă este soluție optimă a problemei de programare liniară. STOP.

Pasul 2. *Alegerea coloanei pivot*.

Pe linia p se caută elementele $(\alpha_j)_p < 0$ şi se determină acel indice q pentru care

$$-\frac{\Delta_q}{(\alpha_q)_p} = \min_{j|(\alpha_j)_p<0}\left\{-\frac{\Delta_j}{(\alpha_j)_p}\right\}$$

şi se decide eliminarea dintre componentele soluției de bază a necunoscutei x_p şi înlocuirea acesteia cu variabila x_q.

Dacă toți $(\alpha_j)_p \geq 0, \forall j$, atunci problema considerată nu are soluție admisibilă. STOP.

Pasul 3. Se execută transformările elementare necesare în tabelul simplex asociat problemei, cu pivotul $(\alpha_q)_p$, după regulile de la algoritmul simplex şi se revine la pasul 1.

Exemplu. Fie problema de programare liniară

$$x_1 + 3x_2 + x_3 \to \min$$

cu restricțiile

$$\left.\begin{aligned} 2x_1 + x_2 - x_3 &\geq 1,\\ x_1 - x_2 &\geq 2,\\ 3x_1 + x_3 &\geq 0,\\ x_2 + 2x_3 &\geq 1,\\ x_i \geq 0, \quad i &= 1,2,3 \end{aligned}\right\}$$

Aducând această problemă la forma standard obținem problema de programare liniară

$$f(x) = -x_1 - 3x_2 - x_3 \to \max$$

în condițiile

$$\left.\begin{aligned} -2x_1 - x_2 - x_3 + x_4 &= -1,\\ -x_1 + x_2 + x_5 &= -2,\\ -3x_1 - x_3 + x_6 &= 0,\\ -x_2 - 2x_3 + x_7 &= -1,\\ x_i \geq 0, \quad i &= 1,2,\ldots,7. \end{aligned}\right\}$$

Această problemă este sub formă dual admisibilă şi aplicându-i algoritmul simplex dual obținem următoarea succesiune de tabele simplex:

Etapa	Variabile de bază	β	x_1	x_2	x_3	x_4	x_5	x_6	x_7	min β_i
	x_4	-1	-2	-1	1	1	0	0	0	
	x_5	-2	[-1]	1	0	0	1	0	0	-2 ←
	x_6	0	-3	0	-1	0	0	1	0	
I	x_7	-1	0	-1	-2	0	0	0	1	
	Δ_j	0	1	3	1	0	0	0	0	
	$-\Delta_j/(\alpha_j)_p$		1↑							
II	x_4	3	0	-3	1	1	-2	0	0	
	x_1	2	1	-1	0	0	-1	0	0	
	x_6	6	0	-3	-1	0	-3	1	0	
	x_7	-1	0	-1	[-2]	0	0	0	1	-1 ←

	Δ_j	-2	0	4	1	0	1	0	0	
	$-\Delta_j/(\alpha_j)_p$		4		1/2↑					
III	x_4	5/2	0	5/2	0	1	-2	0	1/2	
	x_1	2	1	-1	0	0	-1	0	0	
	x_6	13/2	0	7/2	0	0	-3	1	-1/2	
	x_3	1/2	0	1/2	1	0	0	0	-1/2	
	Δ_j	-5/2	0	7/2	0	0	1	0	1/2	

Prin urmare soluția optimă a problemei considerate este $x_1^* = 2, x_2^* = 0, \quad x_3^* = 1/2$ iar $f_{\max} = -5/2$. Soluția optimă a problemei duale este $y_1^* = 0, \quad y_2^* = 1, \; y_3^* = 1/2$.

Înainte de a încheia acest paragraf, vom sublinia faptul că algoritmul simplex sau algoritmul simplex dual nu constituie unicele metode de rezolvare a problemelor de programare liniară. Cu alte metode de rezolvare puteți lua cunoştință în lucrările [13,14,19,21,24,25,31,33,36].

Algoritmul simplex a fost creat în anul 1947 de George Dantzing (S.U.A.) şi după cum s-a văzut constă în parcurgerea vârfurilor poliedrului soluțiilor admisibile, apropiindu-se mereu de soluția optimă, până o atinge într-un vârf al poliedrului. Prin anii şaptezeci s-a constatat că în anumite cazuri algoritmul simplex lucrează în timp nepolinomial (timpul de calcul nu este o funcție polinomială). În anul 1979 savantul L. G. Khachian a propus un algoritm numit metoda elipsoidului. În acest algoritm se parcurge un şir finit de elipsoizi, astfel încât centrul ultimului elipsoid corespunde unei soluții optime şi timpul său de calcul este polinomial. Ulterior, în anul 1984, a fost lansat de către N. Karmarkar (savant indian care locuieşte în SUA) o metodă originală de rezolvare a problemelor de programare liniară, care

considerabil întrece prin rapiditate metodele vechi. Strategia lui Karmarkar constă în reformularea problemei de fiecare dată astfel încât se obține un şir finit de puncte situate în centrul poliedrelor soluțiilor admisibile. Pentru detalii privind metodele lui Khachian şi Karmarkar recomandăm cititorului monografia [36], precum şi lucrarea lui Karmarkar N.A. *New Polinomial-Time Algorithm for Linear Programming. Bell AT&T Laboratories 1984, 38 p.*

4.5. Rezolvarea problemelor de transport

4.5.1. Preliminarii

Primele aplicații ale metodei programării liniare în rezolvarea problemelor de planificare le-au constituit problemele de transport. În paragraful 4.2.2. a fost expusă problema de transport ca un exemplu de programare liniară. Astfel de probleme se întâlnesc foarte des în practică cu ocazia distribuirii mărfurilor, a repartiției între diverşi consumatori etc. Se spune că o problemă de optimizare este o *problemă de tip transport,* dacă modelul său matematic poate fi prezentat astfel: să se determine valoarea minimă a funcției $f = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$ în condițiile:

$$\begin{cases} \sum_{j=1}^{n} x_{ij} = a_i, i = 1,2,\ldots,m; \\ \sum_{i=1}^{m} x_{ij} = b_j, j = 1,2,\ldots,n; \\ x_{ij} \geq 0, \quad i = 1,2,\ldots,m; \quad j = 1,2,\ldots,n; \\ \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j. \end{cases}$$

Datele unei probleme de transport pot fi prezentate sub forma următorului tabel numit matrice de transport:

B_j \ A_i	B_1	B_2	$\dots$	B_{n1}	a_i
A_1	c_{11} x_{11}	c_{12} x_{12}	$\dots$	c_{1n} x_{1n}	a_1
A_2	c_{21} x_{21}	c_{22} x_{22}	$\dots$	c_{2n} x_{2n}	a_2
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
A_m	c_{m1} x_{m1}	c_{m2} x_{m2}	$\dots$	c_{mn} x_{mn}	a_m
b_j	b_1	b_2	$\dots$	b_n	T

În tabel prin T s-a notat valoarea comună din $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.

Această condiție se numeşte *condiție de echilibru*. Dacă ea nu este realizată, atunci se poate introduce un centru fictiv şi reduce astfel problema la o problemă de transport echilibrată. Într-adevăr, fie

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

Atunci introducem o linie $(m+1)$ cu $a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i$ şi $c_{m+1,j} = 0$ pentru $j = 1,2,\dots,n.$ Pentru cazul

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

se va introduce o coloană $(n+1)$ cu

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j \ \text{ şi } \ c_{i,n+1} = 0, \quad i = 1,2,\dots,m\,.$$

Astfel în ambele situații ajungem la formă echilibrată.

Problema de transport este un caz particular de programare liniară, care se poate rezolva prin algoritmul simplex. Datorită formei ei particulare rezolvarea se poate face prin metode speciale. În continuare vom arăta o variantă a algoritmului simplex numită *metoda potențialelor*. Înainte de a trece la descrierea acestei metode vom arăta procedee de determinare a soluției inițiale de bază.

4.5.2. Determinarea soluției inițiale de bază

Într-o problemă de transport, sistemul ecuațiilor de condiții are $m+n$-1 ecuații independente. Într-adevăr, numărul de restricții este $m+n$. Rezolvăm fiecare ecuație, începând cu a doua, a sistemului $\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1,2,\dots,m$ în raport cu variabila x_{i1}:

$$x_{i1} = a_i - \sum_{j=2}^{n} x_{ij}, \quad i = 2,3,\dots,m.$$

În mod analog rezolvăm fiecare ecuație, începând cu a doua, a sistemului $\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,2,\dots,n.$, în raport cu variabila x_{1j}:

$$x_{1j} = b_j - \sum_{i=2}^{m} x_{ij}, \quad j = 2,3,\dots,n.$$

Din primele ecuații ale sistemelor menționate avem

$$x_{11} = a_1 - \sum_{j=2}^{n} x_{1j},$$

$$x_{11} = b_1 - \sum_{i=2}^{m} x_{i1},$$

sau

$$x_{11} = a_1 - \sum_{j=2}^{n}\left(b_j - \sum_{i=2}^{m} x_{ij}\right) = b_1 - \sum_{i=2}^{m}\left(a_i - \sum_{j=2}^{n} x_{ij}\right),$$

de unde conchidem că

$$a_1 - \sum_{j=2}^{n} b_j = b_1 - \sum_{i=2}^{m} a_i,$$

adică obținem condiția de echilibru.

Prin urmare, din condiția de echilibru rezultă că sistemul de ecuații – restricții poate fi rezolvat în raport cu $m+n$-1 variabile. Aşadar o soluție de bază are cel mult $m+n$-1 componente $x_{ij} > 0$, celelalte fiind nule. Dacă avem mai puțin de $m+n$-1 componente pozitive, soluția de bază este degenerată.

Pentru determinarea unei soluții inițiale de bază pe lângă metodele cunoscute din programarea liniară, se cunosc şi metode specifice: procedeul colțului nord-vest, procedeul elementului minim (din tabel, linie sau coloană). Aceste procedee nu cer o teoretizare specială şi le vom ilustra cu aplicarea lor în cazul rezolvării problemei concrete de transport cu trei centre de producție A_1, A_2 şi A_3 în care se găsesc cantitățile *90 t, 70 t* şi respectiv *50 t* şi care trebuie transportate la patru centre de consum B_1, B_2, B_3 şi B_4 în cantitățile *80t, 60 t, 40 t* şi respectiv *30 t*.

Costul transporturilor de la centrele A_i la B_j sunt prezentate în matricea de transport.

$A_i \backslash B_j$	B_1	B_2	B_3	B_4	a_i
A_1	2 80	1 10	3 -	2 -	90
A_2	2 -	3 50	3 20	1 -	70
A_3	3 -	3 -	2 20	1 30	50
b_j	80	60	40	30	210

Procedeul colțului nord-vest de determinare a soluției inițiale de bază constă în următoarele. În pătratul, căruia i se mai spune căsuță, (1,1) din colțul nord-vest al matricei de transport se trece valoarea 80 egală cu cel mai mic dintre a_1 şi b_1 adică $x_{11} = \min(80,90)$. Deoarece avem repartizată o cantitate de produs necesară, în prima coloană se trec $x_{21} = 0, x_{31} = 0$. În matricea de transport în aceste căsuțe vom scrie simbolul "-". Căsuțele în care este trecut acest simbol se numesc *căsuțe libere*.

În continuare ne fixăm atenția asupra lui x_{12} din colțul din stânga de sus al tabelului format cu coloanele B_2 şi B_3 şi procedăm în mod analog ca şi pentru x_{11}, adică vom lua $x_{12} = \min\{b_2, a_1 - b_1\} = \min\{60,10\} = 10$. Deoarece întreaga cantitate $a_1 = 90$ a fost repartizată centrelor B_1 şi B_2 vom avea $x_{13} = 0, x_{14} = 0$.

Rămâne un tabel format din liniile A_2, A_3 şi coloanele B_2, B_3 şi B_4. Pentru acest tabel se aplică din nou procedeul descris:

$x_{22} = \min\{50,70\} = 50,\ x_{32} = 0$. Se trece la $x_{23} = \min\{20,40\} = 20,\ x_{24} = 0$.

În fine $x_{33} = \min\{50,20\} = 20$, $x_{34} = \min\{30,30\} = 30$.

Soluția inițială de bază este $x_{11} = 80$, $x_{12} = 10$, $x_{22} = 50$, $x_{23} = 20$, $x_{33} = 20$, $x_{34} = 30$. Valoarea funcției de eficiență este:

$$f_1 = 2\cdot 80 + 1\cdot 10 + 3\cdot 50 + 3\cdot 20 + 2\cdot 20 + 1\cdot 30 = 450.$$

Determinarea soluției inițiale de bază cu ajutorul *procedeului elementului minim* (din tabel, linie sau coloană) se face în mod asemănător, numai că valorile nu se mai trec în colțul nord-vest, ci în căsuțele cu cost minim de fiecare dată (din tabel, linie corespunzător coloană).

De exemplu, folosind procedeul elementului minim pe linie în problema de transport precedentă, obținem următorul tabel:

B_j \ A_i	B_1	B_2	B_3	B_4	a_i
A_1	30 [2]	60 [1]	- [3]	- [2]	90
A_2	40 [2]	- [3]	- [3]	30 [1]	70
A_3	10 [3]	- [3]	40 [2]	- [1]	50
b_j	80	60	40	30	210

Se alege în fiecare linie elementul c_{ij} cel mai mic şi se repartizează în căsuța corespunzătoare cea mai mică cantitate dintre cantitățile disponibile a_i şi cele necesare $b_j : x_{ij} = \min\{a_i, b_j\}$. În linia întâi elementul cel mai mic este $c_{12} = 1$. Se va destina pentru

$$x_{12} = \min\{a_1, b_2\} = \min\{90,60\} = 60.$$

Deci $x_{22} = 0$, $x_{32} = 0$. În A_1 au mai rămas 90-60-30 t. Vom căuta în valoarea lui c_{1j} imediat superioară lui c_{12}. Această valoare este dată de $c_{11} = c_{14} = 2$. Constatăm că putem lua arbitrar oricare din ele. De exemplu, dacă alegem căsuța (1,1), atunci se va destina pentru x_{11} valoarea 30 egală cu $\min\{90-60,80\}$.

În continuare se procedează în mod similar cu celelalte linii rămase. Se obține următoarea soluție inițială de bază: $x_{11} = 30$, $x_{12} = 60$, $x_{21} = 40$, $x_{24} = 30$, $x_{31} = 10$, $x_{33} = 40$. Valoarea funcției devine:

$$f_2 = 2\cdot 30 + 1\cdot 60 + 2\cdot 40 + 1\cdot 30 + 3\cdot 10 + 2\cdot 40 = 340.$$

Procedeul elementului minim pe coloană este asemănător procedeului elementului minim pe linie, dar valorile $x_{ij} = \min\{a_i, b_j\}$ se determină ținându-se cont de valoarea minimală a elementelor c_{ij} aflate în fiecare coloană. Rezultatul aplicării acestui procedeu duce la soluția de bază inițială arătată în tabelul următor:

B_j \ A_i	B_1	B_2	B_3	B_4	a_i
A_1	80 [2]	10 [1]	- [3]	- [2]	90
A_2	- [2]	50 [3]	- [3]	20 [1]	70
A_3	- [3]	- [3]	40 [2]	- [1]	50
b_j	80	60	40	30	210

Valoarea funcției de eficiență este:

$$f_3 = 2 \cdot 80 + 1 \cdot 10 + 3 \cdot 50 + 1 \cdot 20 + 2 \cdot 40 + 1 \cdot 10 = 430 .$$

Procedeul elementului minim din tabel constă în repartizarea cantităților destinatarilor după criteriul elementului c_{ij} minim din tabel. Aplicarea acestui procedeu ne dă o soluție de bază inițială care coincide cu cea obținută prin procedeul elementului minim pe linie sau cu cea indicată în tabelul următor:

$A_i \setminus B_j$	B_1	B_2	B_3	B_4	a_i
A_1	2 30	1 60	3 -	2 -	90
A_2	2 50	3 -	3 20	1 -	70
A_3	3 -	3 -	2 20	1 30	50
b_j	80	60	40	30	210

Valoarea funcției de eficiență va fi:

$$f_4 = 2 \cdot 30 + 1 \cdot 60 + 2 \cdot 50 + 3 \cdot 20 + 2 \cdot 20 + 1 \cdot 30 = 350.$$

Soluțiile de bază inițiale obținute prin cele patru procedee nu coincid și

$$f_1 = 450, f_2 = -340, f_3 = 430, f_4 = 350 .$$

Rezultatele obținute pentru problema considerată nu pot constitui o concluzie în avantajul utilizării unuia sau altuia din procedeele descrise.

Observație: În exemplul dat $m = 3, n = 4$. Prin procedeele prezentate mai sus am obținut $m + n - 1 = 6$ valori strict pozitive pentru x_{ij}, iar celelalte $m \times n - (m + n - 1) = 12 - 6 = 6$ sunt egale

cu zero. Se demonstrează (vezi, de exemplu, [14]) că, în caz general, pentru *m* centre de depozitare și *n* centre de destinație cu ajutorul procedeelor descrise mai sus putem determina cel mult *m* + *n* – 1 valori ale necunoscutelor x_{ij} strict mai mare decât zero. Aceste variabile se numesc *variabile bazice* și determină o soluție inițială de bază. De la această soluție inițială de bază stabilită prin unul din procedeele prezentate se poate trece la o altă soluție de bază și așa mai departe până vom obține soluția optimă. În continuare vom descrie metoda numită *metoda potențialelor* și care este o variantă a algoritmului simplex.

4.5.3. Metoda potențialelor

Teorema 4.5. Fie $x_{ij}^*, i = 1,2,\ldots,m; j = 1,2,\ldots n$ o soluție optimă a problemei de transport. Fiecărei linii i a matricei de transport îi corespunde un număr u_i^* și fiecărei coloane j - un număr v_j^* astfel, încât se satisfac relațiile:

$$u_i^* + v_j^* = c_{ij} \quad \text{pentru } x_{ij}^* > 0,$$

$$u_i^* + v_j^* \le c_{ij} \quad \text{pentru } x_{ij}^* = 0,$$

$$i = 1,2,\ldots,m; \quad j = 1,2,\ldots,n.$$

Demonstrație. Problema de transport poate fi considerată ca o problemă duală a următoarei probleme de programare liniară:

$$g = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j \to \max$$

cu restricțiile

$$u_i + v_j \le c_{ij},$$

$$i = 1,2,\ldots,m;$$

$$j = 1,2,\ldots,n.$$

Dacă $u_i^*, v_j^*, i = 1,2,\dots,m; j = 1,2,\dots,n$ este o soluție optimă a problemei de mai sus atunci, conform teoremei ecarturilor complementare (vezi paragraful 4.2, teorema 4.4), condiția necesară şi suficientă ca, x_{ij}^*, u_i^* şi v_j^* să fie optime este dată de relațiile

$$\left(u_i^* + v_j^* - c_{ij}\right)\cdot x_{ij}^* = 0, i = 1,2,\dots m; j = 1,2,\dots,n .$$

Din faptul că u_i^* şi v_j^* sunt soluții admisibile pentru problema duală urmează că

$$u_i^* + v_j^* - c_{ij} \le 0, i = 1,2,\dots m; j = 1,2,\dots,n$$

relații care împreună cu cele de mai sus ***demonstrează teorema***.

Mărimile u_i^* şi v_j^* se numesc *potențiale* ale centrelor A_i şi B_j.

Să reluăm exemplul din paragraful 4.5.2 şi să cercetăm soluția inițială de baza obținută prin procedeul colțului nord-vest pentru cazul optim. Pentru aceasta centrului de depozitare $A_i, i = 1,2,3,$ îi atribuim potențialul u_i, iar centrului de consum $B_j, j = 1,2,3,4$ – potențialul v_i.

$A_i \backslash B_j$	B_1	B_2	B_3	B_4	a_i
A_1	2 80	1 10	3 -	2 -	90
A_2	2 -	3 50	3 20	1 -	70
A_3	3 -	3 -	2 20	1 30	50
b_j	80	60	40	30	210

Conform teoremei demonstrate mai sus pentru căsuțele ocupate avem:

$$\left.\begin{aligned} u_1 + v_1 &= 2,\\ u_1 + v_2 &= 1,\\ u_2 + v_2 &= 3,\\ u_2 + v_3 &= 3,\\ u_3 + v_3 &= 2,\\ u_3 + v_4 &= 1.\end{aligned}\right\}$$

Deci, pentru a determina potențialele trebuie să rezolvăm un sistem de şase ecuații cu şapte necunoscute, care are o infinitate de soluții. Luând în acest sistem de ecuații, de exemplu, $u_1 = 0$, vom obține $v_1 = 2, v_2 = 1, u_2 = 2, u_3 = 1, v_3 = 1, v_4 = 0$.

Calculăm diferențele $u_i + v_j - c_{ij}$ pentru căsuțele libere ale matricei de transport:

- pentru căsuța (1,3): $u_1 + v_3 - c_{13} = -2 < 0$;
- pentru căsuța (1,4): $u_1 + v_4 - c_{14} = 0 \le 0$;
- pentru căsuța (2,1): $u_2 + v_1 - c_{21} = 2 > 0$;
- pentru căsuța (2,4): $u_2 + v_4 - c_{24} = 1 > 0$;
- pentru căsuța (3,1): $u_3 + v_1 - c_{31} = 0 \le 0$;
- pentru căsuța (3,2): $u_3 + v_2 - c_{32} = -1 < 0$.

Observăm că $u_2 + v_1 > c_{21}$ şi $u_2 + v_4 > c_{24}$ $u_2 + v_4 > c_{24}$, prin urmare, soluția inițială de bază nu este optimă.

Pentru a îmbunătăți valoarea funcției de eficiență, va trebui să întocmim un nou tabel, care reprezintă tot o soluție de bază admisibilă şi care să conducă la o valoare mai mică pentru funcția de eficiență.

Determinăm

$$\max_{i,j}\{u_i + v_j - c_{ij}\} = u_2 + v_1 - c_{21} = 2\,.$$

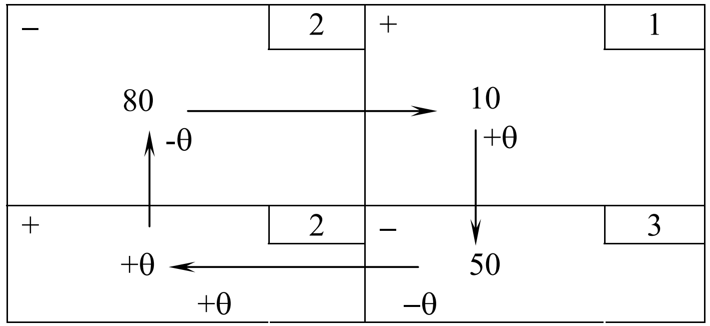
Aceasta înseamnă că în căsuța (2.1) vom introduce o cantitate oarecare θ. Introducerea trebuie făcută în aşa fel încât pe linia a doua dintr-un $x_{2j}, j = 2,3,4$ trebuie scăzut θ pentru a nu modifica disponibilul a_2, iar pe coloana *j* trebuie adăugat θ pentru a nu modifica necesarul b_j şi aşa mai departe, până se închide ciclul în căsuța(2,1).

Vom numi *ciclu* în matricea de transport o linie frântă închisă, vârfurile căreia sunt situate în căsuțele ocupate, iar segmentele – de-a lungul liniilor sau coloanelor ei. Din orice vârf al ciclului putem nimeri în orice alt vârf de-a lungul segmentelor liniei frânte.

Se numeşte *ciclu de recalculare* a căsuței libere (*i,j*) un astfel de ciclu, pentru care unul din vârfuri este situat în căsuțele ocupate. Fiecărui vârf al ciclului de recalculare i se atribuie semnul ″+″ sau ″–″ după următoarea regulă: căsuței libere (*i,j*) aleasă i se atribuie semnul ″+″, căsuței-vârf învecinate – semnul ″–″, următorului vârf – semnul ″+″ ş.a.m.d.

Pentru a construi o nouă soluție de bază în ciclul de recalculare în căsuțele cu semnul ″+″ adăugăm valoarea θ, iar în căsuțele cu semnul ″–″ vom scădea θ. Pentru ca toate valorile x_{ij} situate în căsuțele – vârf să fie nenegative, θ se alege egal cu valoarea minimă din căsuțele – vârf cu semnul atribuit ″–″.

Revenim la exemplul de mai sus şi studiem ciclul de recalculare pentru căsuța liberă (2,1). Avem următorul ciclu de recalculare cu căsuțele – vârfuri (1,1); (1,2); (2,1); (2,2) :



Valoarea lui θ este egală cu $\min(80,50) = 50$. Prin urmare, avem următorul tabel:

A_i \ B_j	B_1	B_2	B_3	B_4	a_i
A_1	2 30	1 60	3 -	2 -	90
A_2	2 50	3 -	3 20	1 -	70
A_3	3 -	3 -	2 20	1 30	50
b_j	80	60	40	30	210

Obținem o nouă soluție de bază:

$$x_{11} = 30, x_{12} = 60,\ x_{21} = 50,\ x_{23} = 20,\ x_{33} = 20,\ x_{34} = 30.$$

În raport cu această soluție, valoarea funcției de eficiență este

$$f = 350.$$

Verificăm dacă această soluție este optimă. Avem următorul sistem:

$$\left.\begin{aligned} u_1+v_1&=2,\\ u_1+v_2&=1,\\ u_2+v_1&=2,\\ u_2+v_3&=3,\\ u_3+v_3&=2,\\ u_3+v_4&=1. \end{aligned}\right\}$$

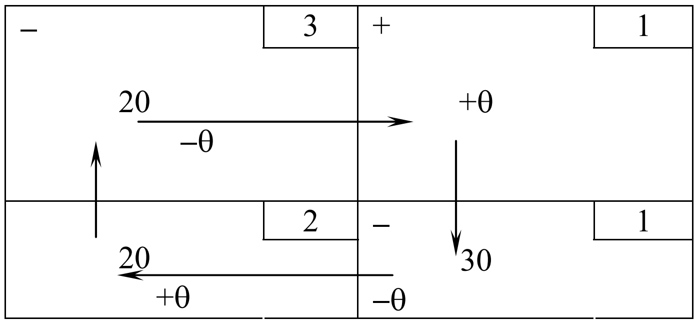
Luând $u_1 = 0$, obținem $u_2 = 0$, $u_3 = -1$, $v_1 = 2$, $v_2 = 1$, $v_3 = 3$, $v_4 = 2$.

Calculăm $\max\limits_{i,j}\{u_i + v_j - c_{ij}\}$ pentru căsuțele libere (1,3), (1,4), (2,2), (2,4), (3,1), (3,2). Avem o singură căsuță (2,4) cu:

$$u_2 + v_4 - c_{24} = 1 > 0.$$

Deci această soluție nu este optimă.

Trecem la o nouă soluție de bază. Pentru aceasta efectuăm o repartizare, folosind un nou ciclu de recalculare pentru căsuța (2,4) cu θ = min(20,30) = 20:



Avem tabelul următor:

B_j / A_i	B_1	B_2	B_3	B_4	a_i
A_1	2 30	1 60	3 -	2 -	90
A_2	2 50	3 -	3 -	1 -	70
A_3	3 -	3 -	2 40	1 30	50
b_j	80	60	40	30	210

Aşadar obținem o nouă soluție de bază $x_{11} = 30$, $x_{12} = 60$, $x_{21} = 50$, $x_{24} = 20$, $x_{33} = 40$, $x_{34} = 10$ cu costul total $f = 330$.

Verificăm dacă această soluție este optimă. Conform celor expuse mai sus, avem următorul sistem:

$$\left.\begin{aligned} u_1+v_1&=2,\\ u_1+v_2&=1,\\ u_2+v_1&=2,\\ u_2+v_4&=1,\\ u_3+v_3&=2,\\ u_3+v_4&=1. \end{aligned}\right\}$$

În acest caz $u_1 = 0$, $u_2 = 0$, $u_3 = 0$, $v_1 = 2$, $v_2 = 1$, $v_3 = 2$, $v_4 = 1$. Constatăm că pentru căsuțele libere

$$\begin{aligned} u_1+v_3&=2<3,\\ u_1+v_4&=1<2,\\ u_2+v_2&=1<3,\\ u_2+v_3&=2<3,\\ u_3+v_1&=2<3,\\ u_3+v_2&=1<3. \end{aligned}$$

Deci soluția obținută este optimă .

Metoda pe care am descris-o este cunoscută sub denumirea de *metoda potențialelor.* Numărul iterațiilor necesare a obține soluția optimă depinde de soluția inițială de bază, adică de procedeul de determinare a ei.

Observație importantă. Metoda potențialelor descrisă în varianta de mai sus necesită ca toate soluțiile de bază să fie nedegenerate, adică să avem de fiecare dată exact $m+n$-1 căsuțe ocupate. Dacă la un moment dat obținem o soluție de bază degenerată (cu mai puține de $m+n$-1 căsuțe ocupate), atunci există căsuțe libere pentru care nu se poate determina nici un ciclu de recalculare a lor. În continuare, vom arăta cum trebuie să procedăm în cazul apariției unei soluții de bază degenerate. Sunt posibile două cazuri.

Cazul 1. Soluția de bază degenerată apare pe parcursul aplicării metodei potențialelor, de exemplu, având o soluție de bază nedegenerată, putem obține un ciclu de recalculare, astfel încât prin înlocuirea valorii θ se anulează deodată mai multe componente ale soluției noi. Atunci vom lăsa liberă numai una dintre aceste căsuțe, iar în celelalte se va scrie câte un zero şi se vor considera ocupate. Asemenea zerouri se numesc *zerouri esențiale.* Zerourile esențiale sunt considerate variabile bazice ca şi variabile diferite de zero. Prin urmare, putem asigura că întotdeauna numărul căsuțelor ocupate să fie egal cu $m+n$-1. Când se verifică dacă soluția de bază este optimă sau nu, în sistemul $u_i + v_j = c_{ij}$ (pentru x_{ij}>0) se includ şi ecuațiile corespunzătoare valorilor egale cu zerouri esențiale. Dacă soluția obținută nu este optimă, atunci se construieşte un ciclu nou de recalculare pentru căsuța respectivă liberă, ciclu în care căsuțele cu zerourile esențiale pot fi folosite ca vârfuri.

Cazul 2. Soluția degenerată apare la început ca soluție inițială de bază determinată prin unul din procedeele descrise în paragraful 4.5.2. Această soluție se completează cu zerouri esențiale pentru a avea $m+n$-1 căsuțe ocupate. Alegerea căsuțelor, în care se pun zerourile esențiale, trebuie făcută cu atenție, astfel încât vectorii corespunzători tuturor căsuțelor ocupate sa fie liniar independenți (să formeze o matrice bazică). În acest scop, în procedeele de determinare a soluțiilor inițiale de bază, descrise în paragraful 4.5.2, vom acționa în felul următor. Dacă la un pas oarecare s-a obținut că avem epuizată cantitatea disponibila a_i şi este repartizată cantitatea necesara de produs b_i, atunci în coloana sau linia respectiva *i* se va trece în zero esențial, iar în celelalte căsuțe ale coloanei şi liniei date se va pune simbolul menționat deja ″–″.

Exemplu. Fie o problemă de transport şi fie soluția inițială de bază, calculată prin procedeul colțului nord-vest, este cea din tabelul următor:

B_j \ A_i	B_1	B_2	B_3	a_i
A_1	3 30	2 20	1 –	50
A_2	3 –	5 –	6 40	40
b_j	30	20	40	90

Avem numai trei căsuțe ocupate, adică o soluție de bază degenerată. Pentru a obține $m+n$-1=2+3-1=4 căsuțe ocupate vom pune un zero esențial în căsuța (1,3) sau în căsuța (2,2). Alegerea căsuței, în care se pune un zero esențial, poate fi făcută comparând costurile de transport (se poate face şi în

mod arbitrar, mai ales când prin comparare nu se poate decide).
Vom obține următorul tabel:

B_j \ A_i	B_1	B_2	B_3	a_i
A_1	3 30	2 20	1 **0**	50
A_2	3 –	5 –	6 40	40
b_j	30	20	40	90

$$x_{11} = 30,\ x_{12} = 20,\ x_{23} = 40,\ f = 370.$$

Pentru a determina potențialele u_i, v_j, rezolvăm sistemul de ecuații:

$$\left.\begin{aligned} u_1 + v_1 &= 3,\\ u_1 + v_2 &= 2,\\ u_1 + v_3 &= 1,\\ u_2 + v_3 &= 6. \end{aligned}\right\}$$

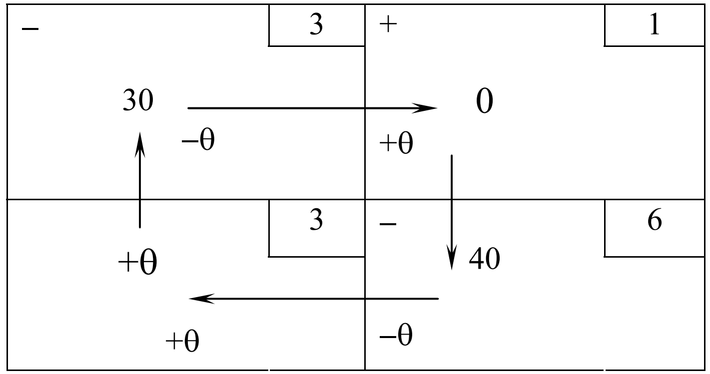
Obținem

$$u_1 = 0,\ u_2 = 5,\ v_1 = 3,\ v_2 = 2,\ v_3 = 1.$$

Calculând $u_i + v_j - c_{ij}$ pentru căsuțele libere, avem:

$$u_2 + v_1 - 3 = 8 - 3 = 5 > 0,$$
$$u_2 + v_2 - 5 = 7 - 5 = 2.$$

Pentru căsuța (2,1) construim ciclul de recalculare cu căsuțele vârfuri (1,1), (1,3), (2,3), (2,1):



Valoarea lui θ este egală cu min{30,40}=30. Se obține astfel tabelul:

B_j \ A_i	B_1	B_2	B_3	a_i
A_1	3 30	2 20	1 **0**	50
A_2	3 –	5 –	6 40	40
b_j	30	20	40	90

$$x_{12} = 20,\ x_{13} = 30,\ x_{21} = 30,\ x_{23} = 10,\ f = 220.$$

Continuarea calculelor o lăsăm pentru cititor, indicându-i că soluția optimă este

$$x^*_{12} = 10,\ x^*_{13} = 40,\ x^*_{21} = 30,\ x^*_{22} = 10 \text{ şi } f_{\min} = 200.$$

Mai subliniem aici, că daca în cercetarea unei soluții optime găsim una sau mai multe diferențe nule pentru căsuțele libere:

$$u_i + v_j - c_{ij} = 0,$$

atunci putem afirma că există şi alte soluţii optime de bază care dau aceeaşi valoare pentru funcţia obiectiv. Daca dorim să le determinăm, putem continua aplicarea metodei potenţialelor, ocupând căsuţa libera, pentru care diferenţa respectiva este nula.

Cu alte metode de rezolvare a problemelor de transport puteţi lua cunoştinţă în lucrarea [14].

4.6. Programarea liniară în numere întregi

Să considerăm o problemă de programare liniară:

$$\left.\begin{aligned} f(x) = (c,x) &\to \max \\ Ax &= b, \\ x &\geq 0, \end{aligned}\right\}$$

în care toate sau o parte dintre variabilele $x_1, x_2, \ldots, x_n$ sunt numere întregi. Dacă toate variabilele trebuie să ia valori întregi, atunci problema considerată se va numi problemă de programare liniară *total în numere întregi*; în caz contrar se va numi problemă de programare *liniară parţial în numere întregi*.

Astfel de probleme apar în cazul multor modele matematice în care variabilele x_i reprezintă mărimi fizice ce nu pot lua decât valori întregi. Multe probleme de extrem cu caracter combinatoric pot fi de asemenea modelate ca probleme de programare liniară în numere întregi (în majoritatea cazurilor variabilele pot lua una din valorile 0,1). Vom da în continuare câteva exemple de probleme de programare în numere întregi.

Problema rucsacului(raniţei). Un călător care urmează să plece într-o excursie intenţionează să ia într-un rucsac n obiecte utile. Să presupunem că există m restricţii asupra rucsacului şi obiectelor în discuţie, de exemplu volumul rucsacului, greutatea obiectelor ş.a.m.d. Fie a_{ij} j caracteristica i a obiectului de tipul j,

$i=1,2,\ldots,m;$ $j=1,2,\ldots,n,$ iar $b_i,$ $i=1,2,\ldots,m$ sunt volumul, greutatea ş.a.m.d. Notăm cu x_j numărul de obiecte de tipul j planificate de a fi puse în rucsac, $j=1,2,\ldots,n.$ Se consideră că se cunoaşte valoarea utilă c_j a unui obiect de tipul j. Atunci modelul matematic al problemei rucsacului are forma:

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

cu restricţiile

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1,2,\ldots,m,$$

$$x_j \geq 0, x_j \text{ - numere întregi, } j = 1,2,\ldots,n.$$

Multe probleme practice se reduc la problema rucsacului. De exemplu, în calitate de rucsac poate fi considerat un vapor sau un avion de bombardament, iar rolul obiectelor îl preia o anumită marfă ambalată în lăzi sau bombe de volume şi greutăţi date.

Problema de afectare. Fie că trebuie executate m lucrări de către n persoane, fiecare dintre aceste persoane urmând a executa exact una din cele n lucrări. Se cunoaşte beneficiul c_{ij} obţinut în urma execuţiei lucrării j de către persoana i. Se pune problema determinării unei repartizări a celor n persoane pentru execuţia al celor n lucrări, astfel încât beneficiul realizat să fie maxim.

Dacă notăm cu x_{ij}, $i,j = 1,2,\ldots,n$ astfel încât $x_{ij} = 1$, atunci când persoana i este repartizată să execute lucrarea j şi $x_{ij} = 0$ în caz contrar, atunci putem scrie

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \to \max$$

cu restricţiile

$$\sum_{i=1}^{n} x_{ij} = 1, j = 1,2,\dots,n,$$

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1,2,\dots,n,$$

$$x_{ij} = 0 \text{ sau } 1, i, j = 1,2,\dots,n.$$

Restricțiile acestei probleme determină ca fiecare persoană să execute o singură lucrare şi ca fiecare lucrare să fie efectuată de o singură persoană.

Problema voiajorului comercial. Să presupunem că avem oraşele notate cu numerele 0,1,2,...,*n*. Voiajorul comercial trebuie să treacă o singură dată prin fiecare oraş şi să se întoarcă acasă, în oraşul 0, astfel încât tot drumul să aibă cost minim.

Fie variabilele $x_{ij}, i, j = 1,2,\dots,n.$ astfel încât

$$x_{ij} = \begin{cases} 1, & \text{dacă drumul trece din} \\ & \text{orasul } i \text{ în orasul } j, \quad i \neq j, \\ 0, & \text{în caz contrar.} \end{cases}$$

Dacă c_{ij} este costul drumului din oraşul *i* în oraşul *j*, atunci problema voiajorului comercial se reduce la rezolvarea problemei de programare liniară cu variabile întregi:

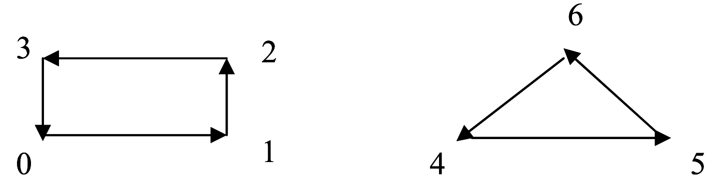
$$\sum_{i=0}^{n} \sum_{j=o}^{n} c_{ij} x_{ij} \to \min$$

astfel încât să fie îndeplinite condițiile

$$\sum_{i=0}^{n} x_{ij} = 1, j = 1,2,\dots,n,$$

$$\sum_{j=0}^{n} x_{ij} = 1, i = 1,2,\dots,n,$$

$$x_{ij} = 0 \text{ sau } 1, i, j = 1,2,\dots,n.$$



S-a constatat că condițiile problemei nu garantează în totdeauna conexiunea drumului, de exemplu, drumul ilustrat în desenul de mai sus satisface relațiilor menționate mai sus, dar nu şi problemei voiajorului comercial.

Se demonstrează (vezi, de exemplu, lucrarea *Ашманов С.А. Линейное программирование. М.: Наука,1981,p.240 - 242*) că, pentru a garanta conexiunea drumului, este necesar a adăuga la restricțiile deja menționate următoarele inegalități:

$$u_i - u_j + nx_{ij} \leq n - 1, \quad i, j = 1,2,\dots,n,$$

unde u_i sunt variabile suplimentare, care nu participă în funcția obiectiv.

Problema de afectare şi problema voiajorului comercial au diverse aplicații în probleme de organizare, dispecerate, relee de televiziune etc.

Faptul că într-o problemă de programare liniară unele sau toate variabilele trebuie să fie întregi, implică mari greutăți în rezolvarea acestor probleme. Se disting două categorii de metode de rezolvare a problemelor de programare liniară în numere întregi:

- metode secționare;
- metode de ramificare şi mărginire (branch and bound).

Orice metodă de secționare constă în următoarele:

1. Se rezolvă problema de programare liniară obținută din problema dată prin renunțarea la restricțiile de integritate (numită *problema relaxată*);
2. Dacă soluția optimă a problemei relaxate are componentele întregi, atunci ea este soluția optimă întreagă căutată;

3. Dacă nu, atunci se construieşte noua problemă relaxată obţinută din problema relaxată anterioară prin adăugarea unei restricţii suplimentare.

Astfel problema de programare liniară în numere întregi se reduce la rezolvarea unui şir de probleme de relaxare. Regulile după care sunt construite restricţiile suplimentare determină diferite metode secţionare. Una dintre cele mai simple metode de secţionare este *algoritmul lui Gomory.*

În cele ce urmează, vom prezenta primul algoritm al lui Gomory de rezolvare a problemei de programare liniară integral în numere întregi. Considerăm problema de programare liniară în numere întregi:

$$(c,x)\to \max$$

cu restricţiile

$$\left.\begin{aligned} &Ax=b,\\ &x\ge 0,\\ &x-\textit{întreg}.\end{aligned}\right\}$$

Vom nota prin $[a]$ partea întreagă a numărului a, iar prin $\{a\}$ partea fracţionară a numărului a:

$$\{a\} = a - [a],\quad 0 \le \{a\} < 1.$$

Se rezolvă problema relaxată

$$(c,x)\to \max$$

cu restricţiile

$$\left.\begin{aligned} &Ax=b,\\ &x\ge 0,\end{aligned}\right\}$$

folosind algoritmul simplex. Se obţine soluţia optimă x^*. Fie x_i^* prima componentă a lui x_i^*, ne întreagă, în ordinea crescătoare (după indice) din ultimul tabel simplex:

Var de bază	β	$(x_B)_1$	…	$(x_B)_i$	…	$(x_B)_m$,	$(x_N)_1$	…	$(x_N)_j$	..	$(x_N)_{n-m}$
$(x_B)_1$	x_1^*	1	…	0	…	0	$(a_1)_1$	…	$(a_j)_1$	..	$(a_{n-m})_1$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$(x_B)_i$	x_i^*	0	…	1	…	0	$(a_1)_j$	…	$(a_j)_i$	..	$(a_{n-m})_i$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$(x_B)_m$	x_m^*	0	…	0	…	1	$(a_1)_m$	…	$(a_j)_m$	..	$(a_{n-m})_m$

Se calculează părţile fracţionare $\{(a_s)_i\}$ pentru linia corespunzătoare variabilei $(x_B)_i$ şi se construieşte o nouă problemă de programare liniară prin adăugarea la restricţiile date a unei restricţii noi de forma

$$\{(a_1)_i\}\cdot(x_N)_1 + \{(a_2)_i\}\cdot(x_N)_2 + \ldots + \{(a_{n-m})_i\}\cdot(x_N)_{n-m} \ge \{x_i^*\}.$$

Se demonstrează (vezi, de exemplu, [13]) că inecuaţia de mai sus defineşte o secţionare admisibilă, astfel încât soluţia optimă x^* a problemei relaxate nu verifică această inecuaţie, dar orice soluţie admisibilă a problemei de programare liniară în numere întregi verifică această inecuaţie.

În continuare se rezolvă următoarea problemă relaxată:

$$(c,x)\to \max$$

cu restricţiile

$$\left.\begin{aligned} &Ax=b,\ x\ge 0,\\ &\sum_{j=1}^{n-m}\{(a_j)_i\}\cdot(x_N)_j \ge \{x_i^*\}\end{aligned}\right\}$$

Dacă soluţia optimă a acestei probleme nu are componentele întregi, atunci se va construi o nouă problemă de programare liniară prin adăugarea la restricţiile date a unei noi secţionări obţinute prin procedeul descris mai sus. Se poate arăta că în condiţii destul de slabe din punct de vedere practic primul algoritm

Gomory rezolvă problema de programare liniară în numere întregi într-un număr finit de paşi. În cazul când $\{(a_s)_i\}=0$ pentru $s = 1,2,...,n-m,$ constatăm că problema de programare liniară integral în numere nu admite soluții admisibile.

Exemplu. Fie că problema de programare liniară integral în numere întregi:

$$f = 7x_1 + 9x_2 \to \max$$

în condițiile

$$\left.\begin{array}{l} -x_1 + 3x_2 \le 6, \\ 7x_1 + x_2 \le 35, \\ x_1 \ge 0, x_2 \ge 0, \\ x_1, x_2 \text{ - întregi.} \end{array}\right\}$$

Aplicând algoritmul simplex pentru rezolvarea problemei relaxate asociate problemei date scrisă în formă standard

$$f = 7x_1 + 9x_2 \to \max$$

$$\left.\begin{array}{l} -x_1 + 3x_2 + x_3 = 6, \\ 7x_1 + x_2 + \qquad x_4 = 35, \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, x_4 \ge 0. \end{array}\right\}$$

Obținem următorul tabel simplex final:

Variabile de bază	β	x_1	x_2	x_3	x_4
x_2	$7/2$	0	1	$7/22$	$1/22$
x_1	$9/2$	1	0	$-1/22$	$3/22$
Δ_j	63	0	0	$28/11$	$15/11$

Soluția optimă $x_1^* = 9/2, x_2^* = 7/2$ pentru problema relaxată nu este întreagă. Vom construi o nouă problemă de programare liniară prin adăugarea la restricțiile anterioare a unei secțiuni, de exemplu x_2:

$$\{7/22\}x_3 + \{1/22\}x_4 \ge \{x_2^*\}$$

sau

$$\frac{7}{22}x_3 + \frac{1}{22}x_4 \ge \frac{1}{2}.$$

Prin urmare avem următoarea problemă relaxată (adusă la forma standard):

$$f = 7x_1 + 9x_2 \to \max$$

cu restricțiile

$$\left.\begin{array}{l} -x_1 + 3x_2 + x_3 \qquad\qquad = 6, \\ 7x_1 + x_2 + \qquad x_4 \qquad = 35, \\ \qquad -\frac{7}{22}x_3 - \frac{1}{22}x_4 + x_5 = -\frac{1}{22}, \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, x_4 \ge 0, x_5 \ge 0. \end{array}\right\}$$

Se observă că problema obținută este dual admisibilă. Folosind algoritmul simplex dual obținem următorul tabel simplex**:**

Variabile de bază	β	x_1	x_2	x_3	x_4	x_5
x_2	3	0	0	0	0	1
x_1	$32/7$	1	1	0	$1/7$	$-1/7$
x_3	$11/7$	0	0	1	$1/7$	$-22/7$
Δ_j	59	0	0	0	1	8

Deoarece soluția optimă $x_1^* = 32/7, x_2^* = 3,$ nu este total întreagă, vom continua procesul de secționare. Variabilei x_1 îi corespunde secționarea generată de restricția

$$\{1/7\}x_4 + \{-1/7\}x_5 \geq \{x_1^*\}$$

sau, cum partea fracționară

$$\{-1/7\} = -1/7 - [-1/7] = -1/7 + 1 = 6/7,$$

$$\frac{1}{7}x_4 + \frac{6}{7}x_5 \geq \frac{4}{7}.$$

Formăm o nouă problemă de programare liniară duală admisibilă căreia - i aplicăm în continuare algoritmul simplex dual. Obținem următorul tabel

Variabile de bază	β	x_1	x_2	x_3	x_4	x_5	x_6
x_2	3	0	1	0	0	1	0
x_1	4	1	0	0	0	-1	1
x_3	1	0	0	1	0	-4	1
x_4	4	0	0	0	1	6	-7
Δ_j	55	0	0	0	0	2	7

care dă soluția optimă întreagă $x_1^* = 4, x_2^* = 3$ şi $f_{\max} = 55$. Observăm că eficiența scade de la 63 la 55 unități.

Subliniem faptul că restricția adăugată problemei anterioare face ca problema relaxată, după ce este adusă la forma

standard, să fie într-o formă dual admisibilă. În acest mod putem aplica algoritmul simplex dual. În aceasta şi constă partea importantă a algoritmului Gomory.

Primul algoritm al lui Gomory, descris mai sus, rezolvă problema de programare liniară total în numere întregi. Cel de-al doilea algoritm al lui Gomory poate fi utilizat atât pentru rezolvarea problemelor total în numere întregi, cât şi pentru rezolvarea problemelor de programare liniară parțial în numere întregi. Descrierea acestui algoritm şi a altor metode de rezolvare a problemelor de programare liniară în numere întregi (metode de ramificare şi mărginire) poate fi găsită în lucrările [13,14,36], la care şi-l trimitem pe cititorul dornic de a face cunoştință cu ele.

4.7. Programarea liniar-fracționară

În cazul când funcția obiectiv este raportul as două funcții liniare şi domeniul soluțiilor admisibile reprezintă un tronson, atunci se spune că avem o *problemă de programare liniar-fracționară.* Modelul matematic al unei probleme de programare liniar-fracționară este de forma:

$$f = \frac{(p,x)+\alpha}{(q,x)+\beta} \to \max$$

în condițiile

$$\left.\begin{array}{l} Ax \leq b, \\ x \geq 0, \end{array}\right\}$$

unde vectorii $p, q \in R^n, b \in R^m$, A este o matrice de dimensiune $m \times n$, iar α şi β sunt scalari.

Astfel de probleme sunt reductibile la probleme de programare liniară. Există diferite procedee de reducere (vezi, de exemplu, lucrarea [19]). Procedeul ce-l vom descrie mai jos a fost propus în anul 1962 de către Charnes şi Cooper.

Să presupunem că domeniul soluţiilor admisibile

$$D = \{x \in R^n : Ax \le b, x \ge 0\}$$

este o mulţime mărginită (adică un poliedru) şi $(q, x) + \beta > 0$ pentru orice soluţie admisibilă $x \in D$.

Dacă notăm

$$z = \frac{1}{(q, x) + \beta} \quad \text{şi} \quad y = z \cdot x,$$

şi ţinem cont de faptul că $z>0$, atunci problema se reduce la următoarea problemă de programare liniară:

$$f = (p, y) + \alpha z \to \max$$

în condiţiile

$$\left.\begin{array}{l} Ay - bz \le 0, \\ (q, y) + \beta z = 1, \\ y \ge 0, z \ge 0. \end{array}\right\}$$

Rezolvând această problemă, obţinem soluţia optimă y^*, z^*. Atunci $x^* = y^*/z^*$ va fi o soluţie optimă a problemei de programare liniar-fracţionară. Într-adevăr, imediat se constată că

$$Ax^* \le b \text{ si } x^* \ge 0,$$

adică x^* este o soluţie admisibilă. Putem scrie

$$(p, y^*) + \alpha z^* \ge (p, y) + \alpha z$$

pentru orice

$$y = \frac{x}{(q, x) + \beta} \text{ şi } z = \frac{1}{(q, x) + \beta}$$

şi orice $x \in D$, ceea ce este echivalent cu

$$\frac{(p, x^*) + \alpha}{(q, x^*) + \alpha} \ge \frac{(p, x) + \alpha}{(q, x) + \beta}, \forall x \in D$$

deoarece $(q, y^*) + \beta z^* = 1$.

Dacă $(q, x) + \beta < 0$ pentru $\forall x \in D$, notând

$$-z = \frac{1}{(q, x) + \beta} \text{ şi } y = z \cdot x,$$

obţinem următoarea problemă de programare liniară:

$$f = -(p, y) - \alpha z \to \max$$

în condiţiile

$$\left.\begin{array}{l} Ay - bz \le 0, \\ -(q, y) - \beta z = 1, \\ y \ge 0, z \ge 0. \end{array}\right\}$$

În mod similar se arată că, dacă y^*, z^* este o soluţie optimă a acestei probleme, atunci $x^* = y^*/z^*$ este o soluţie optimă a problemei de programare liniar-fracţionară considerată.

Prin urmare, în situaţia când D este o mulţime mărginită şi $(q, x) + \beta > 0$ sau $(q, x) + \beta < 0$ pentru $\forall x \in D$, putem construi o problemă liniară, care fiind rezolvată cu ajutorul algoritmului simplex ne dă soluţia optimă a problemei de programare liniar-fracţionară.

Dacă însă există punctele admisibile $x^1, x^2 \in D$ astfel încât

$$(q, x^1) + \beta > 0 \text{ şi } (q, x^2) + \beta < 0,$$

atunci funcţia obiectiv din problema de programare liniar-fracţionară este nemărginită pe mulţimea D.

Exemplu. Fie problema de programare liniar-fracţionară

$$f = \frac{2x_1 - x_2}{x_1 + 2x_2 + 1} \to \max$$

cu restricţiile

$$\left.\begin{array}{l} 2x_1 + x_2 \le 6, \\ x_1 - 2x_2 \le 2, \\ x_1 \ge 0, x \ge 0. \end{array}\right\}$$

Se observă că $x_1 + 2x_2 + 1 > 0$ pentru $\forall x_1 \geq 0, x_2 \geq 0$. Notând

$$z = \frac{1}{x_1 + 2x_2 + 1}, \quad y_1 = zx_1, \quad y_2 = zx_2,$$

reducem această problemă la problema de programare liniară:

$$f = 2y_1 - y_2 \rightarrow \max$$

cu restricțiile

$$\left.\begin{array}{l} 2y_1 + y_2 - 6z \leq 0, \\ y_1 - 2y_2 - 2z \leq 0, \\ y_1 + 2y_2 + z = 1, \\ y_1 \geq 0, y_2 \geq 0, z \geq 0. \end{array}\right\}$$

Aplicând algoritmul simplex, se obține soluția optimă

$$y_1^* = 2/3, y_2^* = 0, z^* = 1/3.$$

Prin urmare $x_1^* = 2, x_2^* = 0$ și $f_{\max} = 4/3$.

4.8. Reoptimizarea şi parametrizarea în programarea liniară

Problemele de programare liniară au un caracter static, însă reoptimizarea şi parametrizarea acestor probleme permit abordarea dinamică a fenomenelor economice, elementelor ataşate fenomenelor economice.

Reoptimizarea sau postoptimizarea constă în recalcularea soluției optime a unei probleme de programare liniară în cazul când se schimbă condițiile inițiale ale acesteia. Importanța unui astfel de studiu duce la economisirea timpului, deoarece se utilizează informația de care se dispune – adică soluția optimă cunoscută.

În procesul de reoptimizare se pot aborda situațiile următoare:

- modificarea vectorului termenilor liberi;
- modificarea coeficienților funcției obiectiv;
- modificarea unui vector linie din *A*;
- modificarea numărului restricțiilor sistemului;
- modificarea numărului de variabile.

Reoptimizarea se efectuează parcurgând următorii paşi:

1. Verificarea optimalității soluției în condițiile modificării datelor problemei;
2. Aflarea optimului problemei modificate (în cazul când soluția problemei inițiale nu mai este optimă).

Prin parametrizarea unei probleme de programare se urmăreşte aflarea variantelor posibile de soluții optime în cazul când unele dintre "constantele" modelului variază liniar sau neliniar în funcție de unul sau mai mulți parametri dați.

Principalele cazuri de programare parametrică sunt:

- parametrizarea vectorului termenilor liberi;
- parametrizarea vectorului coeficienților funcției obiectiv;
- parametrizarea vectorului coloană sau linie din matricea ***A*.**

Soluția problemelor de acest tip cunoaşte următorii paşi:

1. Aflarea soluției optime pentru o valoare determinată a parametrului.
2. Odată aflată soluția optimă pentru o valoare fixată a parametrului, se face studiul sensibilității soluției optime la variația parametrului.

4.8.1. *Modificarea vectorului termenilor liberi*

Se porneşte de la formula de verificare în cadrul metodei simplex:

$$\overline{x}_B = B^{-1} \cdot b$$

$$\overline{a}_j = B^{-1} \cdot a_j, \text{pentru } j = 1,2,...,n$$

unde x_B şi a_j reprezintă formulele finale din tabelul de optim (sau alt tabel). Vom reaminti că ,matricea B^{-1} se citeşte în vechiul tabel de final, coloanele ei corespund succesiv coloanelor matricei unitate iniţiale. În cazul când se modifică vectorul b, el transformându-se în b', ultimul tabel de optim rămâne valabil, cu excepţia coloanei x_B, care se recalculează astfel:

$$\overline{x'} = B^{-1} \cdot b'$$

Aici apar două situaţii:

- noul vector al soluţiei $\overline{x'}$ are toate componentele nenegative, iar $\overline{x'}$ este însăşi soluţia optimă, deoarece în ultimul tabel simplex diferenţele $\Delta_j = z_j - c_j$) nu depind de b;
- noul vector al soluţiei $\overline{x'}$ are componente negative, atunci aplicăm algoritmul simplex-dual şi aflăm soluţia optimă.

Vom analiza următorul exemplu:

$$f(x) = 2x_n + 2x_2 + x_3 \to (\max)$$

cu restricţiile

$$\begin{cases} 2x_1 + x_2 + 3x_3 \le 4, \\ x_1 + x_2 + x_3 \le 3, \\ x_1 + 2x_2 + 2x_3 \le 5. \end{cases}$$

Soluţia optimă este $f(x) = 6, x^* = (1,2,0)^T$, iar

$$B^{-1} = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 1 & -3 & 1 \end{pmatrix}.$$

1. Fie că avem prima situaţie. Noile componente ale vectorului $b' = (5, 3, 5)^T$. Verificăm pe x'_B:

$$x'_B = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 1 & -3 & 1 \end{pmatrix}\begin{pmatrix} 5 \\ 3 \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

Toate componentele lui x'_B sunt nenegative, deci soluţia este şi

$$f^*(x'_B) = 2 \cdot 2 + 2 \cdot 1 + 1 = 7.$$

2. Fie că vectorul $b' = (4, 4, 5)^T$ atunci avem:

$$x'_B = B^{-1} \cdot b = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & 0 \\ 1 & -3 & 1 \end{pmatrix}\begin{pmatrix} 4 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ -3 \end{pmatrix},$$

având componentă negativă, continuăm iteraţiile, pornind de la ultimul tabel, deoarece ea este o soluţie dual-realizabilă:

VB	β	x_1	x_2	x_3	x_4	x_5	x_6
x_1 x_2 x_6	0 4 -3	1 0 0	0 1 0	2 -1 2	1 -1 1	-1 2 -3	0 0 1
Δ_j	8	0	0	-1	0	2	0
x_1 x_2 x_5	1 2 1	1 0 0	0 1 0	1,33 0,(3) 2/3	0,667 $-0,(3)$ $-1/3$	0 0 1	$-0,(3)$ 0,(6) -1
Δ_j	6	0	0	$-2,(3)$	$-0,(6)$	0	$-0,(6)$

Am obținut soluția optimă: $x_B'^{*} = (1,2,0)$ şi $f(x_B'^{*}) = 6$.

4.8.2. Modificarea coeficienților funcției obiectiv

Reoptimizarea funcției obiectiv se face pe baza analizei noilor diferențe, deosebindu-se 2 cazuri:

1) Dacă diferențele $\Delta_j = z_j - c_j$ corespunzătoare vectorilor, care nu aparțin bazei *B*, păstrează semnul pe care l-au avut înainte de modificarea vectorului *c*, atunci înseamnă că noua soluție va fi de asemenea optimă.
2) Dacă cel puțin una din diferențele $\Delta_j = z_j - c_j$, care corespund unui vector ce nu aparține bazei, este pozitivă, în cazul problemei de minimum şi negativ, în cazul problemei de maximum, atunci se vor continua iterațiile prin aplicarea algoritmului simplex primar pornind de la ultimul simplex.

Remarcă. Reoptimizarea vectorului *c* se poate realiza de asemenea prin scrierea dualei problemei date; caz în care vectorul *c* devine *b* şi se procedează aşa cum s-a arătat în paragraful anterior.

Vom analiza pe baza aceluiaşi exemplu ambele cazuri:

I. Fie coeficienții funcției obiectiv au devenit respectiv $c = (3 \quad 2 \quad 1)^T$.

Aşadar se cere de rezolvat:

$$f(x) = 3x_1 + 2x_2 + x_3 \to \max$$

cu restricțiile

$$\begin{cases} 2x_1 + x_2 + 3x_3 \le 4, \\ x_1 + x_2 + x_3 \le 3, \\ x_1 + 2x_2 + 2x_3 \le 5, \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{cases}$$

Observăm că s-a modificat numai prima componentă a funcției scop, care corespunde x_1 şi care nu aparține bazei *B*, deci soluția rămâne optimă, Δ_j nu-şi schimbă semnul. Deci avem $x'^{*} = (1,2,0)^T$, iar $f(x'^{*}) = 3 \cdot 1 + 2 \cdot 2 = 7$

II. Fie că avem modelul

$$f(x) = 2x_1 + 2x_2 + 3x_3 \to \max$$

cu restricțiile

$$\begin{cases} 2x_1 + x_2 + 3x_3 \le 4, \\ x_1 + x_2 + x_3 \le 3, \\ x_1 + 2x_2 + 2x_3 \le 5, \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{cases}$$

Aici observăm că s-a schimbat componenta lui x_3, vectorul căruia intră în baza *B*, deci vom continua iterațiile pornind de la soluția obținută în paragraful 4.8.1.

4.8.3. *Adăugarea unui vector coloană în matricea A*

În cazul extinderii matricei A cu unul sau mai mulți vectori coloane este necesar de asemenea să se verifice criteriul de optimalitate, adică diferențele $\Delta_j = z_j - c_j$, care corespund vectorilor introduşi. Se distring 2 cazuri:

1) Dacă condițiile de optimalitate se verifică, atunci soluția rămâne optimă şi pentru matricea extinsă.
2) Dacă există cel puțin un vector a_m, m – unul din mulțimea indicilor introduşi, – care nu verifică criteriul de optimalitate, atunci se continuă iterațiile până la determinarea soluției optime.

4.8.4. *Modificarea unui vector coloană al matricei A*

Vom presupune că vectorul a_1 din matricea A s-a schimbat, după ce s-a aflat soluția optimă x_B. Vom avea de studiat 2 cazuri, în dependență de apartinerea sau nu a acestui vector bazei B:

1. Dacă a_1 nu aparține bazei B, atunci se verifică testul de optimalitate $\Delta_j = z_j - c_j \geq 0$. În cazul în care soluția nu este optimă se continuă iterațiile.
2. Dacă a_1 aparține bazei optime B, atunci el capătă forma a_j. Deci, se modifică chiar o coloană a matricei B, ceea ce duce şi la schimbarea lui B^{-1}, care va trebui recalculată. Fie noua inversă notată cu $\widetilde{B}^{-1}$. Cu ajutorul ei se reface – coloană cu coloană – tot tabelul final, folosind formulele:

$$\overline{\widetilde{X}_{\widetilde{B}}} = \widetilde{B}^{-1} \cdot b; \quad \overline{\widetilde{a}_l} = \widetilde{B}^{-1} \cdot a_l, l = 1,2,..,m$$

În cazul algoritmului simplex primal şi dual, poate să se întâmple ca unele componente ale vectorului $\overline{\widetilde{X}_{\widetilde{B}}}$ să aibă componente negative, astfel să nu verifice testul de optimalitate. În asemenea cazuri se reia problema de la iterația anterioară intrării vectorului a_j în bază sau se măreşte matricea A cu o coloană, caz analizat în punctul precedent.

4.8.5. *Adăugarea de noi restricții*

Fie că după determinarea soluției optime x_B se adaugă noi restricții (linii) la sistemul $Ax = b$. Vom obține problema de programare liniară:

$$\begin{pmatrix} A \\ A_1 \end{pmatrix} \cdot x = \begin{pmatrix} b \\ b_1 \end{pmatrix}; \; A_1(1,n); b(1,1); x \geq 0; \; \max f(x) = (c, x).$$

Vom alcătui extins:

$$\begin{pmatrix} A & O \\ A_1 & I \end{pmatrix} \cdot \begin{pmatrix} x \\ x_1 \end{pmatrix} = \begin{pmatrix} b \\ b_1 \end{pmatrix}; I(l,n); \; x \geq 0, x_1 \geq 0$$

$$\max g(x) = (c, x) - \lambda \sum_{l=1} x_l$$

Dacă B_1 este baza optimă pentru problema de programare liniară inițială şi B_1^{-1} inversa ei, atunci matricea

$$B = \left(\begin{array}{c|c} B_1 & O \\ B_2 & I \end{array}\right)$$

este nesingulară (B_2 este matricea formată din componentele restricțiilor suplimentare corespunzătoare bazei B_I în soluția optimă inițială).

Dacă notăm cu b_1 vectorul termenilor liberi ai problemei extinse, atunci:

$$x_1^{opt} = B^{-1} \cdot b$$

4.9. Exerciții

1. Să se arate că, dacă X şi Y sunt mulțimi convexe atunci mulțimea $Y = X \cap Y$ de asemenea este convexă.
2. Să se demonstreze că vârfurile tronsonului $T = \{x | Ax \le b\}$ în care A este o matrice de dimensiuni sunt puncte extreme ale sale.
3. Să se determine punctele extreme ale tronsonului T definit de

$$\begin{cases} x_1 + 2x_2 \le 12, \\ x_1 + x_2 \le 18, \\ x_1 \le 12, \\ x_2 \le 9, \\ x_1 \ge 0, \\ x_2 \ge 0. \end{cases}$$

4. Să se arate că funcțiile

$$f_1(x) = \sum_{k=1}^{n} |x_k|, x \in R^n$$

$$f_2(x) = -\sqrt{x}, x \ge 0, x \in R^1,$$

$$f_3(x) = 2x_1^2 + x_2^2 - 2x_1x_2, x \in R^2,$$

$$f_4(x) = x_1^4 + 2x_2^2 + 3x_3^2 - 4x_1 - 4x_2x_3, x \in R^3$$

sunt convexe. Care din ele sunt strict convexe şi care sunt tare convexe?

5. Fie $f(x), x \in R^n$, o funcție convexă. Să se arate că dacă $f(x) \ge 0, \forall x \in R^n$ şi $p \ge 1, p \in R$ atunci funcția

$$g(x) = [f(x)]^p$$

este o funcție convexă.

6. Fie

$$f(x) = \|Ax - b\|_2^2, x \in R^n,$$

unde A este o matrice de dimensiuni $m \times n, b \in R^m$. Să se arate că funcția *f(x)* este convexă în R^n.

7. Să se arate că dacă funcțiile $f_i(x), i = 1,2,...,m, \in R^n$ sunt convexe atunci va fi convexă şi funcția

$$g(x) = \max_{1 \le i \le m} f_i(x).$$

8. Să se arate că pentru orice funcție $f(x), x \in R^n$, care are un punct de minim global în $T \subset R^n$, are loc

$$\min_{x \in T} f(x) = -\max_{x \in T}(-f(x)).$$

9. Să se aducă la forma standard următoarea problemă de programare liniară:

$$f(x) = 3x_1 - 2x_2 + 4x_3 \to \min$$

în condițiile

$$\left.\begin{aligned} 3x_1 - 6x_2 + 9x_3 &\ge -4, \\ 2x_1 + x_2 + 3x_3 &\le 5, \\ x_1 - 4x_2 + 5x_3 &= 2, \\ x_1 \ge 0, x_3 &\ge 0. \end{aligned}\right\}$$

10. Să se rezolve cu ajutorul metodei grafice problema de programare liniară

$$f(x) = 4x_1 + 6x_2 \to \min$$

în următoarele condiții

$$\left.\begin{array}{l} x_1 + 2x_2 \geq 8, \\ x_1 + 6x_2 \geq 12, \\ 3x_1 + x_2 \geq 9, \\ x_1 \geq 0, x_2 \geq 0. \end{array}\right\}$$

11. Se dă sistemul de ecuații liniare

$$\left.\begin{array}{l} x_1 + 3x_2 - x_3 \quad = 4, \\ 2x_1 + x_2 \quad - x_4 = 3, \\ x_i \geq 0, i = 1,2,3,4. \end{array}\right\}$$

a) Să se scrie acest sistem sub forma explicită;
b) Să se determine o soluție de bază;
c) Să se determine o soluție admisibilă de bază.

12. Să se rezolve problema de programare liniară

$$f(x) = x_1 - 2x_2 - 2x_3 \to \min$$

$$\left.\begin{array}{l} x_1 + x_2 + 5x_3 = 2, \\ x_1 - x_2 + 2x_2 = 0, \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{array}\right\}$$

utilizând algoritmul simplex.

13. Să se demonstreze că duala unei probleme duale de programare liniară este problema inițială.
14. Să se rezolve atât problema primală, cât şi problema duală. Să se aducă interpretarea geometrică.

a) $$f(x) = 2x_1 + x_2 \to \min$$

$$\left.\begin{array}{l} -x_1 + 3x_2 \geq 5, \\ 3x_1 - x_2 \geq 1, \\ x_1 \geq 0, x_2 \geq 0; \end{array}\right\}$$

b) $$f(x) = x_1 + x_2 \to \max$$

$$\left.\begin{array}{l} -x_1 + 3x_2 \leq 6, \\ 3x_1 - x_2 \leq 6, \\ x_1 \geq 0, x_2 \geq 0; \end{array}\right\}$$

c) $$f(x) = 2x_1 + x_2 \to \max$$

$$\left.\begin{array}{l} x_1 + 3x_2 \leq 2, \\ -2x_1 + x_2 \leq 2, \\ x_1 \geq 0, x_2 \geq 0. \end{array}\right\}$$

15. Să se demonstreze că dacă problema primală are funcția obiectiv nemărginită pe mulțimea soluțiilor admisibile, atunci problema duală nu are soluții admisibile.
16. Folosind algoritmul simplex dual, să se rezolve problema de programare liniară

$$f(x) = x_1 + 2x_2 + 3x_3 \to \min$$

în condițiile

$$\left.\begin{array}{l} -x_1 + x_2 + 4x_3 \geq 3, \\ x_1 + x_2 - 2x_3 \geq 6, \\ 2x_1 + 2x_2 - x_3 \geq 2, \\ 2x_1 + x_2 - 2x_3 \geq 3. \end{array}\right\}$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$$

17. Să se rezolve problema de transport cu următoarele date

A_i \ B_j	B_1	B_2	B_3	B_4	B_4	a_i
A_1	1	3	2	4	5	50
A_2	3	1	5	3	2	40
A_3	4	2	1	5	1	60
A_4	2	3	1	2	4	21
b_j	15	25	36	10	85	171

18. Utilizând algoritmul Gomory, să se rezolve problema de programare liniară integral în numere întregi

$$f(x) = -x_1 + x_2 + 3x_3 \to \max$$

în condițiile

$$\left.\begin{array}{r} 3x_1 \qquad + x_3 \le 5, \\ -4x_1 + 2x_2 - x_3 \le 2, \\ 2x_1 + \ x_2 + x_3 \le 1, \\ x_1, x_2, x_3 - \textit{întregi} \end{array}\right\}$$

$$x_1 \ge 0, x_2 \ge 0, x_3 \ge 0.$$

19. Să se rezolve problema de programare liniar-fracționară

$$f = \frac{x_1 - x_2 + x_3}{2x_1 + x_3 + 1} \to \max$$

cu restricțiile

$$\left.\begin{array}{r} x_1 - x_2 + 3x_3 = 8, \\ -x_1 + 2x_2 - x_3 = 4, \\ x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{array}\right\}$$